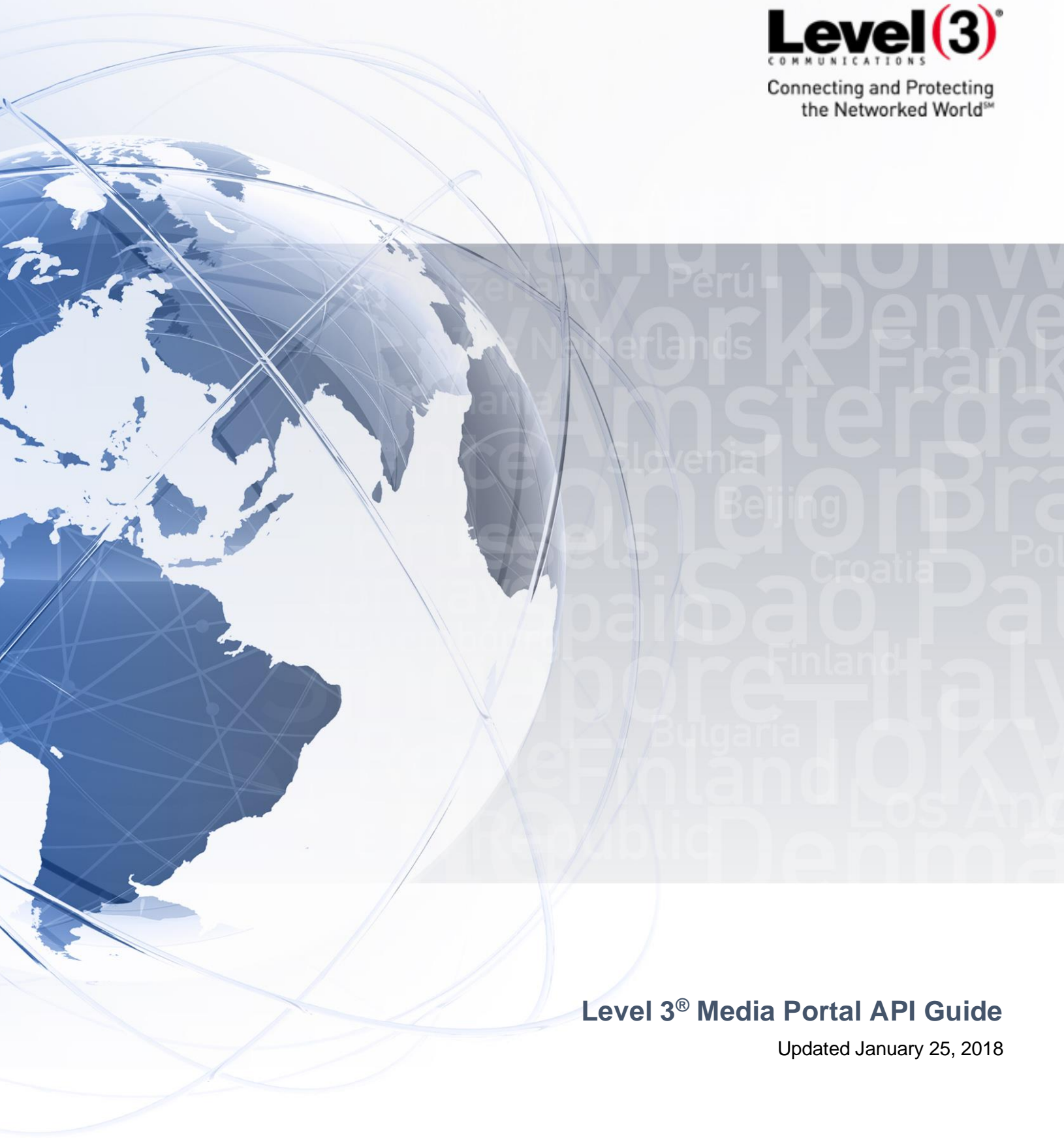




Connecting and Protecting
the Networked WorldSM



Level 3[®] Media Portal API Guide

Updated January 25, 2018

Contents

- Media Web Services (API) 5**
 - Overview 5
 - Usage Style 6
 - API Interface Style 6
 - Definitions 7
- Get Started with Media Portal APIs..... 8**
 - Get Set up to Use Media Portal APIs 8
 - Determine the Access Group ID..... 9
 - API Interfaces 9
 - Host..... 9
 - Interface Examples..... 10
 - API Call Details..... 10
 - API Security 10
 - Sign Your API Requests 11
- API Security Keys 12**
 - What is the API Security Keys Menu Option? 12
 - The API Security Keys Interface 12
 - Add a New Key 13
 - View or Edit an Existing Key 14
 - Security Hierarchy 14
 - Create a New Secret 15
 - Disable or Enable an API Key..... 15
- API Security Framework 16**
 - API Key Characteristics 16
 - Key Statuses..... 17
 - API Request Acceptance 17
 - API Authentication Overview 18
 - Request Signature Form..... 19
 - Unauthenticated Requests..... 19
 - Authorization: Roles and Permissions 20

- Rate Limits 20
- API Tutorial..... 21**
 - Use Case: New User Setting Up Services 21
 - Add Property Definition – Media Portal UI.....21
 - Add Property Definition – API.....22
 - Configure Cache Control | Add a Service Rule – Media Portal UI22
 - Configure Cache Control | Add a Service Rule – API23
 - Add a Property Rule – Media Portal UI23
 - Add a Property Rule – API.....24
- API Specification..... 25**
 - Access Control 25
 - Retrieve Access Control Configuration Group.....26
 - Create Access Control Configuration Group27
 - Update Access Control Configuration.....29
 - Access Group Hierarchy 30
 - Aliases 31
 - Retrieve Aliases31
 - Add Alias32
 - Remove Aliases33
 - Cache Control..... 34
 - Retrieve Cache Control (CCHO) Configuration Groups35
 - Create Cache Control (CCHO) Configuration Group36
 - Update Cache Control (CCHO) Configuration Group37
 - Configuration Groups 39
 - Delete Configuration Group39
 - Retrieve All Configuration Groups40
 - Content Analytics Summary & Trend 41
 - Content Manipulation 47
 - Retrieve Content Manipulation (QSHMode) Configuration Group.....47
 - Create Content Manipulation (QSHMode) Configuration Group48
 - Update Content Manipulation (QSHMode) Configuration Group49
 - Geo Definitions 50
 - Create Geo Definition50
 - Retrieve all Geo Definitions.....51
 - Retrieve a Specified Geo Definition52
 - Update Geo Definition53
 - Delete Geo Definition.....54

- Get Configuration Information 55
- HTTP Delivery 56
- Invalidations 59
 - Invalidations: Request Invalidation (POST).....59
 - Invalidations: Request Origin Invalidation (POST).....64
 - Invalidations: Check status (GET)66
- Key (Deprecated) 70
- Mobile Login 71
- Origin Definition 72
 - Retrieve Origin Definition72
 - Update Origin Definition.....73
- Origin Storage 74
 - Create New Virtual Host75
 - Create New User Account.....77
 - Update User Account.....78
 - Delete User Account.....79
- Order Status 80
- Properties 81
 - Create a New Property82
 - Retrieve a Property.....84
 - Deactivate a Property85
 - Reactivate a Property86
- Resource Groups 87
 - Create Resource Group.....88
 - Retrieve Resource Groups.....89
 - Retrieve Individual Resource Group.....90
 - Update Resource Group Order91
 - Update Resource Group.....92
 - Delete Resource Group93
- RTM (Caching) 94
- Services Hierarchy (Partially Deprecated) 98
- Token Definitions..... 103
 - Create Token Definition 104
 - Retrieve all Token Definitions..... 105
 - Retrieve an Individual Token Definition..... 106
 - Delete a Token Definition..... 107
- Usage Reporting..... 108
 - Usage Reporting – Data Interval 113

- Usage by Access Group 117
- Vyvx Reservations 119
 - Vyvx Reservation – Get Active Status 119
 - Vyvx Reservation – Get Service Status 120
 - Vyvx Reservation – Get Service History 121
- Appendix: Error Responses 122
- Appendix: Usage Granularity Rules 125
 - Caching 125
 - Streaming 125
 - Origin Storage 125
- Appendix: Vyvx State and Ingress State Response Definitions 126
 - Possible State Values 126
 - Ingress State Calculation 126
- Using Media Portal’s API Debugger 127**
 - Installing the Application 127
 - Building a Request 128
 - Using Output or Input Tabs 128
- API Sample Code 129**
 - Terms of Use 130

Media Web Services (API)

Overview

Media Portal provides a web-based interface to the CDN services and features you use to operate your content business. For some customers, however, the ability to request data directly about the services for use in their own processing is more useful. These pages describe Media Web Services made available to customers through Media Portal.

The API provides access to these areas:

- Service configuration
- List of services
- Historical reports
- Real-time monitoring
- Invalidations
- Content Analytics reporting
- Client-side statistics reporting

These pages help you answer these questions:

- **What is an API key?**

An API key is a numerical ID and secret provided by the Media Portal that is used to construct an API request. See "What is the API Key?" **Error! Bookmark not defined..**

- **How do I get an API key?**

You can create a key through the Media Portal interface for use in building your request signature. See "Add a New Key" **Error! Bookmark not defined..**

- **How do I construct an API key signature?**

The header of each request must include the request signature, which includes all the parts necessary to validate the request, authenticate the sender's ID, and authorize access to the API interface. See "API Authentication Overview" **Error! Bookmark not defined..**

- **How do I construct an API request?**

Each API request must conform to an interface specification. See "API Specification" on page 21 to find the interface you want to use.

- **Where do I send the API request?**

Each API interface has a specific URL. See "API Specification" on page 21 and find the interface you want to use.

- **What do I get back for my request?**

Depending on how the request is built, the data includes different levels of detail. See "API Specification" on page 21 to find the interface you want to use.

Usage Style

The API interface style used is representational state transfer (REST). The hierarchical organization of the CDN service tree makes the REST style a good fit for the CDN APIs. The service tree hierarchy for the purposes of calling the APIs is: Access Group > SCID > Network Identifier.

API Interface Style

The API architecture is stateless:

- Every HTTPS API request must include all of the information required for the server to fulfill that request.
- Each request is constructed to be complete, unless information is provided back to the server from the client within the new request.
- Some API functions may be submissions for a process which will not be completed for several minutes, such as cache invalidation requests. For an invalidation request, the server might immediately accept the request and return a transaction ID. The client may then poll the server with the transaction ID for an update on the state of the request.

For more information, see: *RESTful Web Services* by Leonard Richardson and Sam Ruby, Chapter 4: “The Resource-Oriented Architecture: Stateless”, p. 86, and Chapter 8: “REST and ROA Best Practices: Why Statelessness Matters”, p. 222.

Definitions

These terms are used in the API documentation pages:

Term	Definition
API	Application programming interface (API) is an interface that defines the ways by which an application program may request services from libraries and/or operating systems. An API determines the vocabulary and calling conventions the programmer should employ to use the services. It may include specifications for routines, data structures, object classes and protocols used to communicate between the requesting software and the library.
REST	Representational state transfer (REST) is a style of software architecture for distributed systems such as the World Wide Web.
RESTful Web Service	A RESTful web service (also called a RESTful web API) is a simple web service implemented using HTTP and the principles of REST. The book "RESTful Web Services" by Leonard Richardson and Sam Ruby, identifies a REST architecture called "RESTful, Resource Oriented Architecture". See the description in Chapter 1: The Competing Architectures, page 13, and the more detailed explanation in Chapter 8: REST and ROA Best Practices.
API Key	Also referred to generically as 'key'. It is created within the Media Portal, associated to a single Access Group, with associated permissions (authorized APIs), and is used to authenticate API requests to the Level 3 Media API infrastructure.
URI	A RESTful, resource-oriented service exposes a URI for every piece of data the client might want to operate on.

Get Started with Media Portal APIs

Media Portal is Level 3's customer portal for CDN services. Through Media Portal, you can configure user accounts, group your services for reporting purposes, view historical and real-time usage, and order and configure new services.

While a browser-based portal is ideal for many users, some situations require access to data feeds so that customers may incorporate Level 3 CDN data directly into their own tools and systems. Media Portal APIs are designed to meet this need. An API (application programming interface) provides a means for developers to build software that interacts directly with Level 3's CDN portal data without the use of a browser-based graphical user interface. Over time, our goal is to provide an API equivalent for each functional area available in Media Portal.

Get Set up to Use Media Portal APIs

The basic steps required to use Media Portal APIs are:

1. **Acquire an API Key.** An API Key consists of a numeric key ID and a secret. Keys can be created through the Media Portal *API Security Key Management* screen.
2. **Determine the Access Group ID.** Since the access group name is editable, Media Portal assigns an access group ID that does not change. The access group ID is required as part of the scope for each API request and the Key API is used to locate this ID. For more information, see: *Determine the Access Group ID*.
3. **Determine the SCID or Network Identifier.** You will need to include those values in your URI syntax by using the Services Hierarchy API. For more information, see: *Services Hierarchy (Partially Deprecated)*.
4. **Determine which API interfaces you need to call,** then write code to form the request and to handle the response. Brief descriptions of the available API interfaces are included in this document. For more information, see: *API Interfaces* or *API Specification*.
5. **Implement code to sign the requests you send.** Each request must be signed in order to ensure the request can be authenticated. For more information, see: *API Security*.

Determine the Access Group ID

After you have the API Key and Secret, you can then locate the Access Group ID, which is used to develop your API request. Each API request requires the Access Group ID as part of the scope.

To determine the ID:

1. Send a request using the API Key or Access Group interface. See *API: Key (Deprecated)* on page 59 or *API: Access Group Hierarchy*. (The *API: Key* returns the top-level access group, while the *API: Access Group Hierarchy* returns all access groups, including the top-level group.)
2. Locate the access group ID in the response. The ID is circled below:

```
<apikey id="14816" xsi:noNamespaceSchemaLocation="https://ws.level3.com/schema/keyv1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <apiCorrelationId>API-421d15cd-0338-4eca-b644-8c85c9c44277</apiCorrelationId>
  <assignedAccessGroup id="1" name="Level3 Internal"/>
  <contact id=" " name=" " @level3.com"/>
  <role id="5" name="Admin"/>
  <status>Active</status>
  <credits>4999843</credits>
  <next-top-up/>
</apikey>
```

3. Use this ID as part of your selected API request.

API Interfaces

Media Portal APIs include a selection of interfaces that enable you to query your CDN service hierarchy, perform caching invalidations and retrieve real-time monitoring information for Streaming services, for example. Other interfaces cover additional key functional areas of the browser-based Media Portal.

Media Portal APIs follow the REST style (representational state transfer). RESTful web services leverage the design principles of HTTP 1.1. The common HTTP methods are used to denote the nature of the action to be taken (the “verb”). For example, the GET method is used only to retrieve data, never to delete it. The URI contains the scope information that defines the “nouns” – what the call will act upon. URIs are constructed logically, reflecting natural hierarchies of resources and are designed to be easily human-readable.

The APIs are called by making an HTTP request with a selected method to the required URI. Responses include XML data in the request response body providing the requested data or result of the operation. Each request must be signed in order to be authenticated the process.

Host

The default host for all API calls is:

<https://ws.level3.com>

Interface Examples

This table includes a list of some of the specific interfaces:

Interface	Description
/key	Basic information about the API Key used to make the call
/accessGroups	The access group hierarchy addressable by the calling API Key
/services	The service hierarchy addressable by the calling API Key
/rtm	Real-time monitor data for both caching and streaming services
/usage	Historical usage data for caching, streaming and storage services
/invalidations	Support for creating and querying status of caching content invalidation requests

API Call Details

Full syntax for all of the available web services, along with example calls and the returned data, is available in the "API Specification" on page 21.

API Security

In order to protect access to your CDN services, Media Portal APIs include a robust security mechanism. In addition to using HTTPS to ensure that requests and response contents are encrypted, every request presented to the API web services must be "signed". A valid signature confirms that the request has been sent by an authenticated API Key.

Note: The secret provided as part of your API Key is used to sign requests. If a third party gains knowledge of your API Key, it is your responsibility to disable or suspend the key, or to inform Level 3 so that the key may be disabled on your behalf. Failure to disable a compromised key could result in unauthorized access to your CDN service information and configuration.

Sign Your API Requests

The signature mechanism used by Media Portal APIs is HMAC (hash-based message authentication code) and the SHA-1 cryptographic hash function. The process works as follows:

1. Selected information from your request, including certain HTTP request header fields, are combined into a string.
2. A digest of that string is produced using the secret associated with your API Key: this is the signature.
3. The signature and the API Key ID (which is public) are placed in the HTTP Authorization header and the request is sent.
4. Upon receiving the request, Media Portal inspects the Authorization header and extracts the API Key ID.
5. Media Portal looks up the secret associated with that API Key – remember, the secret is known only by the API Key owner and Level 3.
6. Media Portal gathers the other inputs to the digest and builds its own signature.
7. If the two signatures match, then the request is authenticated.

Note: In testing your connectivity to Media Portal APIs, an HTTP debugger (such as [Charles](#)) may be useful. HTTP debuggers allow you to inspect outgoing requests and ensure that the HTTP headers that are sent over the wire match the values you used to construct your request signature. It is essential that the input values to the signature match what is sent to the API host exactly.

API Security Keys

What is the API Security Keys Menu Option?

The API Security Keys menu option is used to generate an API key for use in constructing the signature required to authenticate an API request. The API key is an input to the authorization procedure that is applied to all API requests.

An API request requires four basic things to successfully operate:

- A numeric ID and secret text string. These two generated parts of the API key are used to authenticate the request. Only you as the creator and Level 3 have access to the secret.
- The access group and role. These two assignments determine what operations the API key is authorized to perform.

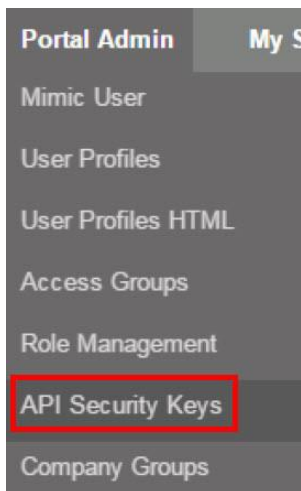
Each of these are either created or selected on the API Security Keys screen. For more information about these items:

- The contents of a security key, see "API Key Characteristics" **Error! Bookmark not defined.**
- How a signature is built, see "Request Signature Form" **Error! Bookmark not defined.**

Using roles in Access Groups, see *Role Management*.

The API Security Keys Interface

If your role includes the Admin permission, the *API Security Keys* menu item is available to you:



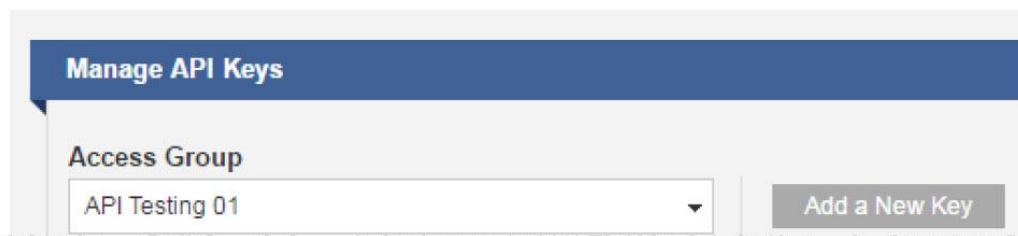
Add a New Key

An API key is needed to construct an API request.

To add a new key:

1. Select the **API Security Key** menu item.
2. Select the Access Group that will use the key.

API Security Keys



3. Click the **Add a New Key** button.

Note: There is a limit of 5 keys per Access Group. If you reach this limit, this button becomes disabled.

4. Read and accept the Terms of Use.
5. Enter a memorable Key Name (one that matches the name of your application, for example).
6. Select a Role that assigns the proper authorization level for this key. For more information on roles, see Role Management.

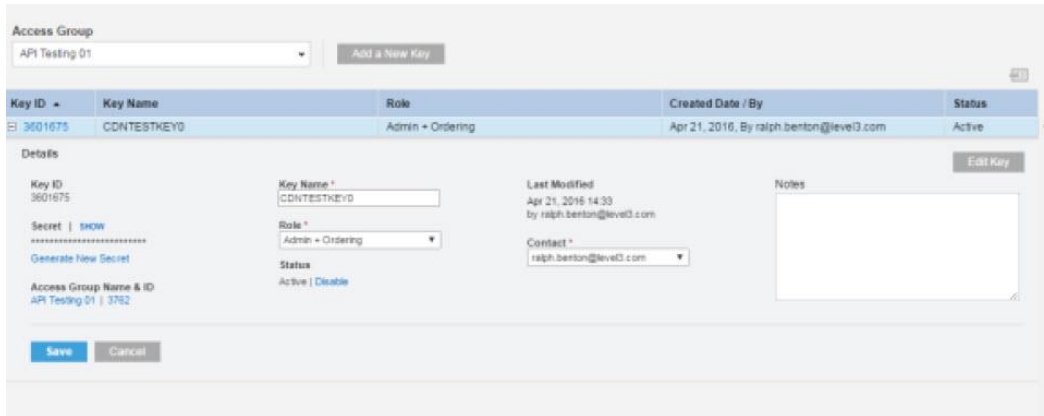
Note: Only Media Portal default roles can be assigned to an API Key. Custom roles are not displayed in the list of available roles to be assigned.

7. Click **Save**.

View or Edit an Existing Key

You can view the details of an existing key and, if necessary, edit informational fields.

1. API Key ID to expand the section.



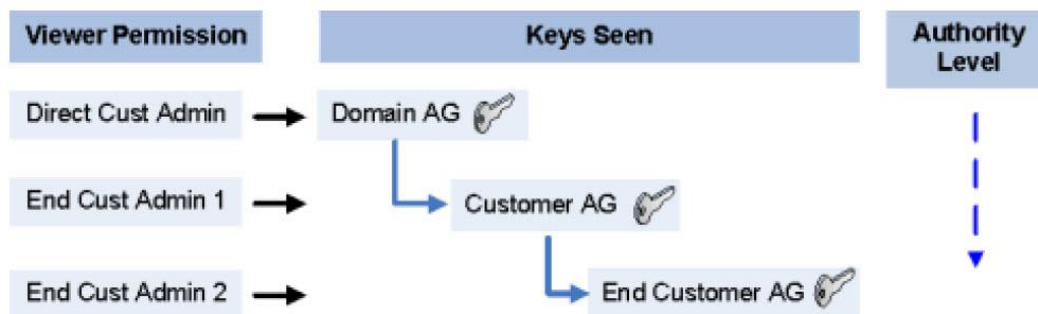
2. Click on **Edit Key** to make changes to Key Name, Role, Contact, or Notes, then click **Save**.
 - If needed, you can click on **Generate New Secret**. Once confirmed, you do not need to save.

Security Hierarchy

Depending on a user's role and permissions, Media Portal may or may not display the API keys. For example, in the illustration below:

- The Direct Customer Administrator can see keys in the Domain Access Group and any below.
- End Customer Administrator 1 can see the keys in Customer Access group and End Customer Access Group, but not in the Domain Access group.

Although the permissions assigned to an administrator in different access group levels might be the same, the authority is lower the further down the access group tree.



Create a New Secret

If you need to change the secret used in an API key because, for example, the existing key's security has been compromised, you can request that Media Portal generate a new secret for an existing key.

Note: Changing the secret immediately cancels the validity of the existing secret. Any requests made with the old secret in the signature are rejected until the new secret is inserted.

To create a new secret:

1. Select the API Key menu item.
2. Find and click the existing key in the Access Group Key list.
3. Click on the Key ID to expand the table row.
4. Click on **Edit Key**.
5. Click on **Generate New Secret** and confirm your request.
6. Click on **Show** to confirm the change.

Disable or Enable an API Key

Simply expand the table row for the relevant Key. Under the Status section, click on **Active** or **Disable**. Email notifications are sent when the status changes.

API Security Framework

In Media Portal, an API key is a 5-digit numeric ID plus an alphanumeric Secret. These two parts are used to construct a request signature that is used to authenticate all Media Portal API requests. To create the key, see the "API Security Keys" on page 12.

The Secret element of the API key is used to construct a hash-based message authentication code from unique elements of the request, using a RFC2104 HMAC-SHA1 hash.

API Key Characteristics

An API key has these characteristics:

- Every API Key is assigned a unique numerical ID by Media Portal. This key is public and does not need to be kept secret.
- The secret is assigned by Media Portal and can be regenerated if necessary. The secret is 160 bits in length and conforms to the requirements of constructing a RFC2104 HMAC-SHA1 digest.

Note: If the secret becomes compromised, you may want to generate a new one. See "Creating a New Secret" **Error! Bookmark not defined.**

- The API key name is optional and can be updated by the administrator. Use this feature to simplify managing your keys. For example, assign the key the name of the application that will use it.
- An API Key is associated to a single Access Group.

An API Key is authorized to access specific APIs based on the role assigned to the key. For more information about roles, see Role Management.

Key Statuses

An API key can have one of four statuses:

Status	Reason
Active	Functioning.
Disabled	Assigned to this key on the Edit screen by customer, or by Level 3 due to some other important reason. Key is still valid, but requests from this key are rejected. If the key has been disabled by an admin in a parent access group, the enable key is not active in child access groups.
Suspended	Assigned to all keys within the access group, which stops all API request processing from requests within the access group and any child access groups. Key is still valid, but requests from this key are rejected. In addition, no new keys can be created by any user in the access group or any of its child access groups.

You can view the status of any key by selecting it on the API Security Keys table.

Note: An administrator that disables or suspends keys can enable or resume those same keys within their access group. However, they cannot take those actions on keys disabled or suspended by an admin of a parent access group if they have not been assigned to the group.

API Request Acceptance

Each time Media Portal receives an API request, it goes through these steps:

1. Authentication:
 - Is the API Key ID recognized? If not, reject the request.
 - Is the request signature valid? If not, reject the request.
2. Enabled status: Is the API Key ID currently disabled? If so, reject the request.
3. Suspension: Is the Access Group ID to which the API Key ID is assigned currently suspended? If so, reject the request.
4. Request rate: Is the rate of requests from this API Key ID higher than the specified threshold? If so, reject the request. The rate is 10 requests per minute.

Media Portal returns a HTTP status code if the request is rejected for one of these reasons. This table lists the codes:

Description	Response Code	Entity Body Returned to Client
Authentication failure	403	None. No entity body is returned to the caller to limit exposing data to a potentially malicious request.
Request timestamp is too old	403	mpeRequestTooOld
API Key is disabled	403	mpeAPIKeyDisabled
Access Group API privileges suspended	403	mpeAPIPrivilegesSuspended
API Key request rate too high	503	mpeRequestRateTooHigh

API Authentication Overview

API authentication is bi-directional between Level 3 and the requestor.

- Level 3's authentication is established by using a signed SSL certificate, allowing the caller to establish communications with the web services via HTTPS. The traffic exchanged is encrypted, preventing snooping of both CDN service-related data and the parameters required to construct the request signature.
- The requester is authenticated by providing a signed HTTPS request to a specific URI/API, using the secret to generate a request signature.

Once the request is authenticated, the API key's authorization level is evaluated using the assigned role and rate limits.

Request Signature Form

The Authorization HTTP request header field expected from clients is:

MPA [API Key ID]:[signature]

where MPA (Media Portal Authentication) is the authentication scheme and signature is a value that is properly constructed as described below.

Note: If an accept header is set in the request, the only valid value is "text/xml". Any other value will receive a 406 response.

This signature is constructed in the form of a RFC2104 HMAC-SHA1 digest. Create a string formed as follows:

- [Date] + "\n" + [RelativePath] + "\n" + [Content-Type] + "\n" + [HTTP-Verb] + "\n" + [Content-MD5]
- "\n" is a line feed
 - [Date] is the value of the Date request header field formatted as, for example, Wed, 29 Apr 2015 +0000 using Java SimpleDateFormat, use: "EEE, dd MMM yyyy HH:mm:ss +0000" Java SimpleDateFormat using Locale.US), using Locale.US for the current UTC time. See the sample code for examples.
 - [URI or RelativePath] is the path of the request including request scope if applicable (access group, service, network IDs). The RelativePath should include the first forward slash (/) but should not include query string parameters. Examples:
/key/v1.0
/usage/v1.0/1234/BBB1234/my.property.com
 - [Content-Type] is the value of the Content-Type request header field. For example:
text/xml
application/json
 - [HTTP-Verb] is the HTTP method used for the request, e.g. "GET", "PUT", "POST", "DELETE"
 - Optional: [Content-MD5] is the value of the Content-MD5 request header field, an MD5 digest of the request body. See RFC 2616 Section 14.15. If this request header is set, then it must be included in the signature string.

Encode this string as UTF8, construct an HMAC-SHA1 digest (using the secret) and then Base64 encode the result. The output of these steps is the signature. For implementation examples, see the code samples.

Unauthenticated Requests

Unauthenticated requests are rejected and a HTTP status is sent. These include:

- un-signed requests
- requests not made over HTTPS
- requests where the Date header value is older than 15 minutes

Media Portal logs un-authenticated requests (IP address, requested URI, key ID, date/time).

Authorization: Roles and Permissions

Each API key is assigned a role. Roles contain permissions that determine access to features within Media Portal. Roles are assigned when the key is created. For more information, see: Role Management.

For example, if you want to use an API key for invalidations, assign the key the default Configuration role because it includes Invalidation, or assign a custom role that includes at least the Invalidation permission.

Keys can only be assigned a single role. The permissions in the role define which APIs the key is authorized to use.

As with users, API keys inherit authorizations from parent to child access groups down the hierarchy. However, keys created in a child access group do not have authorization for actions in any higher-level, parent access group.

If a request fails authorization, Media Portal sends a response code to the requester and log the request (IP address, requested URI, key ID, date and time). For more information, see: *Appendix: Error Responses*.

Rate Limits

Media Portal limits the rate and number of requests per API key per minute. The current rate limit is 10 per minute.

If the rate of requests is higher than the defined rate, further requests are denied until another time period begins. Requests over the rate amount create a log event.

Rate limits are enforced after the request is authenticated.

API Tutorial

Use Case: New User Setting Up Services

This sample case describes a hypothetical Level 3 customer who has a website that contains images and headings that need to be refreshed daily, videos that are uploaded weekly, and a shopping cart that allows visitors to make purchases (except in three European countries).

The customer has a website called **www.LetsLearnHowToPaint.com**.

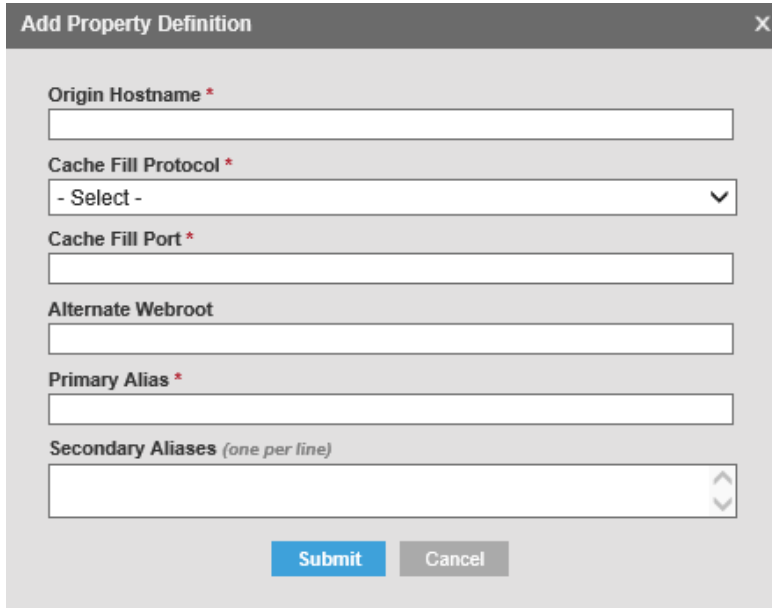
The customer creates an access group called **LetsPaint**.

The steps to setting up their website include the following:

1. **Add a property definition**—define an access group
2. **Add a service rule**—refresh content using CCHOMODE
3. **Add a property rule**—define geoblocking

Add Property Definition – Media Portal UI

1. Click **My Services > Edit Service Configuration**.
2. Select a Service Component ID.
3. Click **Add Property**.



Add Property Definition [X]

Origin Hostname *

Cache Fill Protocol *

Cache Fill Port *

Alternate Webroot

Primary Alias *

Secondary Aliases (one per line)

Submit **Cancel**

Add Property Definition – API

```
POST /serviceConfiguration/v1.0/1234/BBB5678
{
  "originserver" : {
    "host" : "LetsLearnHowToPaint.hostedorigin.com",
    "port" : "8001",
    "protocol" : "http",
  "aliases" : [
    "www.LetsLearnHowToPaint.com",
    "LetsLearnHowToPaint.com"
  ]
}
```

Configure Cache Control | Add a Service Rule – Media Portal UI

1. Click **My Services > Edit Service Configuration**.
2. Select a Service Component ID.
3. Click **Add Service Rule**.

Add Service Rule

URL Path Filter
 URL Path Filter
Path Filter From *

Cache Control Enabled

Query String Handling Enabled

Token Authentication Enabled

Geo Blocking Enabled

Submit **Cancel**

Configure Cache Control | Add a Service Rule – API

```

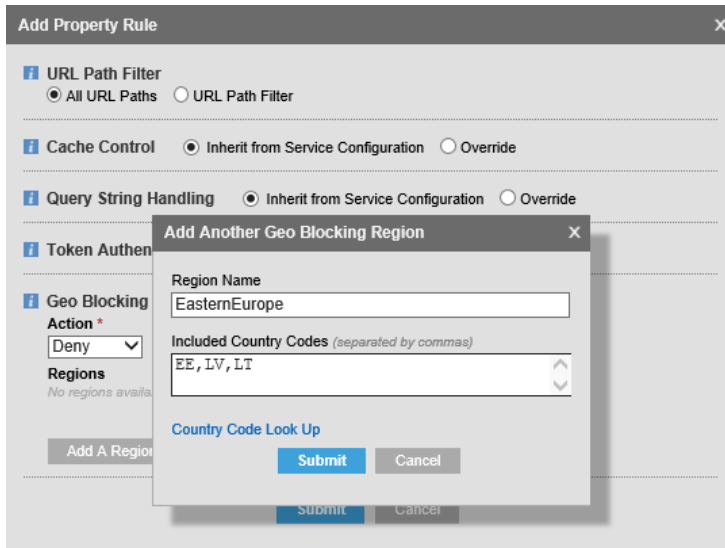
POST /serviceConfiguration/v1.0/1234/BBB5678/resourceGroups
{
  "rgid" : "videos_s",
  "rgtype" : "path"
  "rgdef" : [ "*.mp4"],
}
POST /serviceConfiguration/v1.0/1234/BBB5678/ConfigurationGroups
{
  "CacheControl" : {
    "cchomode" : {
      "int" : "1d"
    }
  }
}

```

Add a Property Rule – Media Portal UI

Use geo-blocking to restrict shopping from three Eastern European countries (Estonia, Latvia, and Lithuania).

1. Click **My Services > Edit Service Configuration**.
2. Select a Service Component ID.
3. Click **Add Property Rule**.



The screenshot shows the 'Add Property Rule' dialog box in the Media Portal UI. The 'Geo Blocking' section is active, with the 'Action' set to 'Deny'. A modal titled 'Add Another Geo Blocking Region' is open, showing the following fields:

- Region Name:** EasternEurope
- Included Country Codes (separated by commas):** EE, LV, LT

Buttons for 'Submit' and 'Cancel' are visible at the bottom of the modal.

Add a Property Rule – API

```
POST /serviceConfiguration/v1.0/1234/BBB5678/[URL]/GeoDefs
```

```
{  
  "geoid" : "EasternEurope",  
  "cc" : "EE,LV,LT"
```

```
}
```

```
POST /serviceConfiguration/v1.0/1234/BBB5678/[URL]ResourceGroups/  
shoppingcart/ConfGroups/AccessControl
```

```
{  
  "AccessControl": {  
    "geoblocking": {  
      "geoid": [ " EasternEurope " ],  
      "type": "deny"
```

```
    }
```

```
  }
```

```
}
```

API Specification

The API Specification is a collection of interfaces that are designed to accept requests from authorized, authenticated senders. When all of the conditions are met in a properly constructed request, Media Portal provides a response. Follow the steps listed in *Getting Started with Media Portal APIs* on page 8 to create a key.

The APIs described below are for the Level 3 Media Portal API Version 1.0.

Access Control

The Access Control section includes the operations for setting up Geographic Location Blocking (geoblocking) and Token Authentication.

Once a geoblocking rule is configured, it causes an edge server to use the client IP address to determine a geo location and evaluate against the defined allow or deny list. The configuration of a geoblocking object includes the following:

geoid—array of geo IDs as defined in GeoDefs

Type either "allow" or "deny".

Token Authentication allows customers to protect content from URL tampering or unauthorized re-use or re-publication via email forwarding or deep linking to content. Using shared secrets defined in the tokens, a URL signature appended to the query string of the resource URL can be validated by the CDN before serving content without contacting the customer environment for authentication.

The configuration of the tokenauth object applies to all tokens defined in the Tokens section and includes the following:

action Currently limited to "fail"

Note: Because token authentication requires the addition of query string parameters to the URL, Content Manipulation qshmode for assets to be protected using token authentication should typically be set to action: "ignore".

Retrieve Access Control Configuration Group

Base URI	https://ws.level3.com
Method	GET
Description	Retrieves the Access Control Configuration Group of a given Resource Group for service component or property.
URI Syntax	<p>/serviceConfiguration/(version)/(scope)/ResourceGroups/(rgid)/ConfGroups/AccessControl</p> <p>version [CDATA[]] Values: "v1.0" (required)</p> <p>scope [CDATA[]] (AG)/(SCID)/[(ALIAS)]</p> <p>AG = Access Group</p> <p>SCID = Service Component Identifier</p> <p>ALIAS = Optional Alias identifies Property</p>
Sample Request	<p>URI: https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/test1.caching.cdn.level3.net/ResourceGroups/videos/ConfGroups/AccessControl</p> <p>https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/ResourceGroups/videos/ConfGroups/ AccessControl</p>
Sample Response	<pre>{ "AccessControl": { "tokenauth": { "action": "fail" }, "geoblocking": { "geoid": ["NorthAmerica", "Asia"], "type": "deny" } } }</pre>
Possible Status and Error Messages Returned to Client	<p>200: Request Successful.</p> <p>See Appendix A – Status Codes & Error Messages for additional return codes and messages.</p>

Create Access Control Configuration Group

Base URI	https://ws.level3.com
Method	POST
Description	Creates a new AccessControl group of a service component or a property. An AccessControl group can include a geoblocking and a tokenauth object. A geoblocking setting includes an array of geo ids and a type specifying an allow or deny filter. The tokenauth setting includes token ID and the action attribute, which is currently limited to "fail".
Body Syntax	<pre>{ "AccessControl": {"geoblocking": {# optional "geoid": [" <geoid>", ...] # required "action": ("deny" "allow") # required}, "tokenauth": { # optional "action": "fail" # required } } }</pre>
URI Syntax	<p>/serviceConfiguration/(version)/(scope)ResourceGroups/(rgid)/ConfGroups/AccessControl</p> <p>version [CDATA[]] Values: "v1.0" (required)</p> <p>scope [CDATA[]] (AG)/(SCID)/[(ALIAS)]</p> <p>AG = Access Group</p> <p>SCID = Service Component Identifier</p> <p>ALIAS = Optional Alias identifies Property</p>
Sample Request	<p>URI: https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/test1.caching.cdn.level3.net/ResourceGroups/videos/ConfGroups/AccessControl</p> <p>https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/ResourceGroups/videos/ConfGroups/AccessControl</p> <p>Body:</p> <pre>{ "AccessControl": { "geoblocking": { "geoid": ["NorthAmerica"], "type": "allow" } } }</pre>
Sample Response	<pre>{ "AccessControl": { "geoblocking": { "geoid": ["NorthAmerica"], "type": "allow" } } }</pre>

Possible Status and Error Messages Returned to Client 201: Created.
See Appendix A – Status Codes & Error Messages for additional return codes and messages.

Update Access Control Configuration

Base URI	https://ws.level3.com
Method	PUT
Description	<p>Updates an AccessControl group of a service component or a property. An AccessControl group can include a geoblocking and a tokenauth object. A geoblocking setting includes an array of geo IDs and a type specifying an allow or deny filter. The tokenauth setting comprises an array of token IDs and an action attribute. Action is currently limited to "fail". The update applies to adding or removing the geoblocking or tokenauth objects or changes to such objects themselves.</p>
URI Syntax	<p>/serviceConfiguration/(version)/(scope)/ResourceGroups/(rgid)/ConfGroups/AccessControl</p> <p>version [CDATA[]] Values: "v1.0" (required)</p> <p>scope [CDATA[]] (AG)/(SCID)/[(ALIAS)]</p> <p>AG = Access Group</p> <p>SCID = Service Component Identifier</p> <p>ALIAS = Optional Alias identifies Property</p>
Body Syntax	<pre>{ "AccessControl": { "geoblocking": { # optional "geoid": ["<geoid>", ...] # required "action": ("deny" "allow") # required }, "tokenauth": { # optional "type": "fail" # required } } }</pre>
Sample Request	<p>URI: https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/test1.caching.cdn.level3.net/ResourceGroups/videos/ConfGroups/AccessControl https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/ResourceGroups/videos/ConfGroups/AccessControl</p> <p>Body:</p> <pre>{ "AccessControl": { "geoblocking": { "geoid": ["NorthAmerica", "Asia"], "type": "allow" }, "tokenauth": { "action": "fail" } } }</pre>
Sample Response	Empty
Possible Status and Error Messages Returned to Client	<p>204: Success no Content.</p> <p>See Appendix A – Status Codes & Error Messages for additional return codes and messages.</p>

Access Group Hierarchy

Base URI	https://ws.level3.com
Method	GET
Description	Default call: returns hierarchical representation of the Access Group associated with the calling API key
Typical Use	Determine part of the scope for any API call. Lists every access group and child access group that is assigned to the user.
Schema Location	https://ws.level3.com/schema/accessGroups/v1.0
URI Syntax	/accessGroups/(version) accessGroups—Returns list of access groups within this key. version—Required version.
Sample Request	https://ws.level3.com/accessGroups/v1.0
Sample Response	<pre><?xml version="1.0" encoding="UTF-8"?> <accessGroup id="12345" name="My Access Group" xsi:noNamespaceSchemaLocation="https://ws.level3.com/schema/accessGroups/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <apiCorrelationId>CDNPortal-1270071184562- 8650</apiCorrelationId> <serviceResource>/12345</serviceResource> <description>AG description</description> <createdDate>2010-01-22 01:23 -0700</createdDate> <modifiedDate>2010-01-22 01:23 -0700</modifiedDate> <createdUser>user@domain.com</createdUser> <modifiedUser>user@domain.com</modifiedUser> <domainId>894</domainId> <parentId>894</parentId> <accessGroups> <accessGroup .../> </accessGroups> </accessGroup></pre>
Possible Status and Error Messages Returned to Client	See "Appendix: Error Responses" on page 122.
Cost Per Call	See Appendix: Cost per Call.

Aliases

Retrieve Aliases

Base URI	https://ws.level3.com
Method	GET
Description	Retrieves the array of aliases for the property.
Schema Location	https://ws.level3.com/schema/serviceConfiguration/v1.0
URI Syntax	/serviceConfiguration/(version)/(scope)/Aliases version [CDATA[]] Values: "v1.0" (required) scope [CDATA[]] (AG)/(SCID) AG = Access Group SCID = Service Component Identifier
Sample Request	URI: https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/test1.caching.cdn.level3.net/Aliases
Sample Response	["test1.alias.cdn.level3.net", "test2.alias.cdn.level3.net"]
Possible Status and Error Messages Returned to Client	200: Request Successful. See Appendix A – Status Codes & Error Messages for additional return codes and messages.

Add Alias

Base URI	https://ws.level3.com
Method	POST
Description	<p>Adds a new alias to the array of aliases for the property.</p> <div style="background-color: #e0ffe0; padding: 5px; border: 1px solid black;"> <p>Note: This service does not require or accept a content body. Please ensure that the Content-Length HTTP header is set to 0.</p> </div>
URI Syntax	<p>/serviceConfiguration/(version)/(scope)/Aliases/(ALIAS)</p> <p>version [CDATA[]] Values: "v1.0" (required)</p> <p>scope [CDATA[]] (AG)/(SCID)</p> <p>AG = Access Group</p> <p>SCID = Service Component Identifier</p>
Sample Request	URI: https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/test1.caching.cdn.level3.net/Aliases/mynew.alias.caching.cdn.level3.net
Sample Response	["mynew.alias.caching.cdn.level3.net"]
Possible Status and Error Messages Returned to Client	<p>201: Created</p> <p>See Appendix A – Status Codes & Error Messages for additional return codes and messages.</p>

Remove Aliases

Base URI	https://ws.level3.com
Method	DELETE
Description	Removes an alias from the array of aliases for the property.
	<div style="background-color: #e0ffe0; padding: 5px;"> <p>Note: This service does not require or accept a content body. Please ensure that the Content-Length HTTP header is set to 0.</p> </div>
URI Syntax	<p>/serviceConfiguration/(version)/(scope)Aliases/(ALIAS)</p> <p>version [CDATA[]] Values: "v1.0" (required)</p> <p>scope [CDATA[]] (AG)/(SCID)</p> <p>AG = Access Group</p> <p>SCID = Service Component Identifier</p>
Sample Request	URI: https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/test1.caching.cdn.level3.net/Aliases/mynew.alias.caching.cdn.level3.net
Sample Response	EMPTY
Possible Status and Error Messages Returned to Client	<p>200: Request successful.</p> <p>See Appendix A – Status Codes & Error Messages for additional return codes and messages.</p>

Cache Control

The Cache Control section includes the operations for setting up the Cache Control Header Override (CCHO).

A Cache Control Header Override setting consists of the following:

- An internal Cache Control policy specified as TTL defaulting to seconds or optionally followed by one of the following letters to indicate the units:
 - s (seconds)
 - m (minutes)
 - h (hours)
 - d (days)
 - w (weeks)
 - y (years)

Note: Alternatively, the internal policy can be set to "as-is", "no-cache", or "no-store."

- An external Cache Control policy, following the same rules as the internal policy.

A force attribute with possible values "yes" and "no". This is an optional value for the external Cache Control policy only.

Retrieve Cache Control (CCHO) Configuration Groups

Base URI	https://ws.level3.com
Method	GET
Description	Retrieves the Cache Control Configuration Group of a given Resource Group for service component or property.
URI Syntax	<p>/serviceConfiguration/(version)/(scope)/ResourceGroups/(rgid)/ConfGroups/CacheControl</p> <p>version [CDATA[]] Values: "v1.0" (required)</p> <p>scope [CDATA[]] (AG)/(SCID)/[(ALIAS)]</p> <p>AG = Access Group</p> <p>SCID = Service Component Identifier</p> <p>ALIAS = Optional Alias identifies Property</p>
Sample Request	<p>URI:</p> <p>https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/test1.caching.cdn.level3.net/ ResourceGroups/videos/ConfGroups/CacheControl</p> <p>https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/ResourceGroups/videos/ConfGroups/CacheControl</p>
Sample Response	<pre>{ "CacheControl": { "cchomode": { "ext": "no-store", "force": "yes", "int": "1234" } } }</pre>
Possible Status and Error Messages Returned to Client	<p>200: Request Successful.</p> <p>See Appendix A – Status Codes & Error Messages for additional return codes and messages.</p>

Create Cache Control (CCHO) Configuration Group

Base URI	https://ws.level3.com
Method	POST
Description	Creates a new Cache Control Group. Supports a cchomode object, which includes an internal and external policy identifier and an optional force attribute (for details, see Cache Control category summary section)
Body Syntax	{ "CacheControl": { "cchomode": { "int": "<TTL String>", # required "ext": "<TTL String>", # optional "force": ("yes" "no") # optional, only applicable for ext } } # TTL String: Integer with optional time unit (s/m/h/d/w/y). Default is seconds. # Minimum TTL is 30 seconds.
URI Syntax	/serviceConfiguration/(version)/(scope)/ResourceGroups/(rgid)/ConfGroups/CacheControl version [CDATA[]] Values: "v1.0" (required) scope [CDATA[]] (AG)/(SCID)/[(ALIAS)] AG = Access Group SCID = Service Component Identifier ALIAS = Optional Alias identifies Property
Sample Request	<pre>{ "CacheControl": { "cchomode": { "int": "<TTL String>", # required "ext": "<TTL String>", # optional "force": ("yes" "no") # optional, only applicable for ext } } }</pre> <p># TTL String: Integer with optional time unit (s/m/h/d/w/y). Default is seconds. # Minimum TTL is 30 seconds.</p>
Sample Response	<pre>{ "CacheControl": { "cchomode": { "ext": "no-store", "force": "yes", "int": "120m" } } }</pre>
Possible Status and Error Messages Returned to Client	201: Created. See Appendix A – Status Codes & Error Messages for additional return codes and messages.

Update Cache Control (CCHO) Configuration Group

Base URI	https://ws.level3.com
Method	PUT
Description	Updates a Cache Control Group. See Create operation above for detail on CacheControl object. Note that all required attribution needs to be provided in the update body.
URI Syntax	/serviceConfiguration/(version)/(scope)/ResourceGroups/(rgid)/ConfGroups/CacheControl version [CDATA[]] Values: "v1.0" (required) scope [CDATA[]] (AG)/(SCID)/(ALIAS) AG = Access Group SCID = Service Component Identifier ALIAS = Optional Alias identifies Property
Body Syntax	<pre> { "CacheControl": { "cchomode": { "int": ("as-is" "no-cache" "no-store" "<TTL String>") , # required "ext": ("as-is" "no-cache" "no-store" "<TTL String>") , # optional "force": ("yes" "no") # optional, only applicable for ext } } } # TTL String: Integer with optional time unit (s/m/h/d/w/y) </pre>
Sample Request	URI: https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/ test1.caching.cdn.level3.net/ResourceGroups/videos/ConfGroups/CacheControl https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/ ResourceGroups/videos/ConfGroups/CacheControl Body: <pre> { "CacheControl": { "cchomode": { "ext": "no-cache", "force": "yes", "int": "120m" } } } </pre>
Sample Response	<pre> { "CacheControl": { "cchomode": { "ext": "no-store", "force": "yes", "int": "120m" } } } </pre>

}

Possible Status and Error Messages Returned to Client	204: Success no Content. See Appendix A – Status Codes & Error Messages for additional return codes and messages.
--	--

Configuration Groups

Configuration Groups provide rules that apply to the associated Resource Group filters. Currently, this allows the following categories:

- **Cache Control**—set CCHO Mode
- **Content Manipulation**—specify query string handling mode
- **Access Control**—specify geoblocking and token authentication

Most operations are specific to the types above and are detailed in subsequent sections, with the exceptions of one common Get operation to retrieve all Configuration Groups for a Resource Group and a Delete, which applies to the specified category.

Delete Configuration Group

Base URI	https://ws.level3.com
Method	DELETE
Description	Removes a resource group for service component or property by rgid.
URI Syntax	/serviceConfiguration/(version)/(scope)/ResourceGroups/(rgid)/ConfGroups/(group) version [CDATA[]] Values: "v1.0" (required) scope [CDATA[]] (AG)/(SCID)/[(ALIAS)] AG = Access Group SCID = Service Component Identifier ALIAS = Optional Alias identifies Property group [CDATA[]] CacheControl AccessControl ContentManipulation
Sample Request	URI: https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/test1.caching.cdn.level3.net/ResourceGroups/videos/ConfGroups/CacheControl https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/ResourceGroups/videos/ConfGroups/AccessControl
Sample Response	Empty
Possible Status and Error Messages Returned to Client	200: Request Successful. See Appendix A – Status Codes & Error Messages for additional return codes and messages.

Retrieve All Configuration Groups

Base URI	https://ws.level3.com
Method	GET
Description	Retrieves the array of Configuration Groups for a given Resource Group of a service component or property.
URI Syntax	/serviceConfiguration/(version)/(scope)/ResourceGroups/(rgid)/ConfGroups version [CDATA[]] Values: "v1.0" (required) scope [CDATA[]] (AG)/(SCID)/[(ALIAS)] AG = Access Group SCID = Service Component Identifier ALIAS = Optional Alias identifies Property
Sample Request	URI: https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/ test1.caching.cdn.level3.net/ResourceGroups/videos/ConfGroups https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/ResourceGroups/videos / ConfGroups
Sample Response	<pre> { "ContentManipulation": { "qshmode": { "action": "ignore", "type": "string" } }, "CacheControl": { "cchomode": { "ext": "no-store", "force": "yes", "int": "1234" } }, "AccessControl": { "geoblocking": { "geoid": ["NorthAmerica", "type": "allow" } } } </pre>
Possible Status and Error Messages Returned to Client	200: Request Successful. See Appendix A – Status Codes & Error Messages for additional return codes and messages.

Content Analytics Summary & Trend

Base URI	https://ws.level3.com												
Method	GET												
Description	Returns summary data for the specified collection, or summary and trend data for the remaining report types and date range. Hourly data is stored for 30 days. Daily data is stored for 12 months.												
Typical Use	This is the second step in querying Content Analytics data. <div style="background-color: #e0f0e0; padding: 5px; border: 1px solid #ccc;"> <p>Note: The first step requires using the "Services Hierarchy (Partially Deprecated)" to determine the Content Analytics hierarchy and collections.</p> </div> <p>Finds content analytics data for enabled network identifiers and their collection detail in these types of data: Collection, Requestor, Referer, Server and ASN. For more information about these types of analytics data, see Viewing Reports.</p>												
Schema Location	https://ws.level3.com/schema/contentAnalytics/v1.0 https://ws.level3.com/schema/contentAnalyticsDataInterval/v1.0												
URI Syntax	<p>/contentAnalytics/(version)/(AG)/(SCID)/(NI)/(collectionID)?groupBy=(reporttype)&dateFrom=<yyyymmddhhmm>&dateTo=<yyyymmddhhmm>&id=(value)&dataInterval=(dataInterval)</p> <table border="1"> <tr> <td>version</td> <td>Required version.</td> </tr> <tr> <td>scope /(AG)/(SCID¹)/(NI)/</td> <td>Scope must retain sequence that reflects hierarchy. AG is Access Group² ID. NI³ is network identifier name.</td> </tr> <tr> <td>collectionID</td> <td>Required. Returned with "Services Hierarchy (Partially Deprecated)" Error! Bookmark not defined. Same as "Slot" on Media Portal Content Analytics Collections screen.</td> </tr> <tr> <td>groupBy "collection" "requestor" "referer" "server" "asn" "urlDetails" "statusCodes"</td> <td>Required. Report data type.</td> </tr> <tr> <td>dateFrom yyyymmddhhnn</td> <td>Required. Date range – starting date/time</td> </tr> <tr> <td>dateTo</td> <td>Required. Date range – end date/time</td> </tr> </table>	version	Required version.	scope /(AG)/(SCID ¹)/(NI)/	Scope must retain sequence that reflects hierarchy. AG is Access Group ² ID. NI ³ is network identifier name.	collectionID	Required. Returned with "Services Hierarchy (Partially Deprecated)" Error! Bookmark not defined. Same as "Slot" on Media Portal Content Analytics Collections screen.	groupBy "collection" "requestor" "referer" "server" "asn" "urlDetails" "statusCodes"	Required. Report data type.	dateFrom yyyymmddhhnn	Required. Date range – starting date/time	dateTo	Required. Date range – end date/time
version	Required version.												
scope /(AG)/(SCID ¹)/(NI)/	Scope must retain sequence that reflects hierarchy. AG is Access Group ² ID. NI ³ is network identifier name.												
collectionID	Required. Returned with "Services Hierarchy (Partially Deprecated)" Error! Bookmark not defined. Same as "Slot" on Media Portal Content Analytics Collections screen.												
groupBy "collection" "requestor" "referer" "server" "asn" "urlDetails" "statusCodes"	Required. Report data type.												
dateFrom yyyymmddhhnn	Required. Date range – starting date/time												
dateTo	Required. Date range – end date/time												

¹Service Component Identification. Unique ID number in the order entry system associated to a billable component of a service. See Service.

²An administration method used to create and manage groups or sub-accounts that are then used to grant and limit access by service. Administration of those sub-accounts can be delegated to business units or customers.

³A common reference to the name used in the CDN. In Caching, the primary alias or "property". In Streaming, the Streaming ID, while in FMS 3.5, the primary supename plus Streaming ID. In Origin Storage, the VHost name.

yyyymmddhhnn

id	Provides collection data for groupByentity. Returned with Summary call.
dataInterval "day" "hour"	Required for groupBy argument. Used with id for collection data.

Sample 1 Request

```
https://ws.level3.com/contentAnalytics/v1.0/12345/BBBN12345/cdn.exempleni.com/1?groupBy=collection&dateFrom=201008010000&dateTo=201008300000&dataInterval=day
```

Sample 1 Response - Collection Trend for AG 12345 for August 1 to 4, 2010

```
<data xsi:noNamespaceSchemaLocation="https://ws.level3.com/schema/contentAnalyticsDataInterval/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <apiCorrelationId>CDNPortal-1283897706255-9806</apiCorrelationId>
  <point id="08/02/10">
    <item serviceResource="/12345/BBBN12345/cdn.exempleni.com/1">
      <time>08/02/10</time>
    </item>
  </point>
  <point id="08/03/10">
    <item serviceResource="/12345/BBBN12345/cdn.exempleni.com/1">
      <time>08/03/10</time>
    </item>
  </point>
</data>
```

Sample 2 Request

```
https://ws.level3.com/contentAnalytics/v1.0/12345/BBBN12345/cdn.exempleni.com/1?groupBy=requestor&dateFrom=201008010000&dateTo=201008300000
```

Sample 2 Response - Requestor Summary for AG 12345 for April 1 to April 5, 2010

```
<accessGroup id="12345" name="12345/BBBN12345"
xsi:noNamespaceSchemaLocation="https://ws.level3.com/schema/contentAnalytics/v1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <apiCorrelationId>CDNPortal-1283984445602-0592</apiCorrelationId>
  <dataInterval/>
  <serviceResource>/170</serviceResource>
  <services>
    <service id="BBBN12345">
      <serviceResource>/12345/BBBN12345</serviceResource>
      <product>CACHING</product>
      <networkIdentifiers>
        <ni id="cdn.exempleni.com">
          <serviceResource>/12345/BBBN12345/cdn.exempleni.com
        </serviceResource>
      </networkIdentifiers>
      <collections>
        <collection id="1">
          <serviceResource>/12345/BBBN12345/cdn.exempleni.com/1
          </serviceResource>
        </collection>
      </collections>
    </service>
  </services>
  <summaryData>
```

```

        <id>107</id>
        <area>CHN</area>
        <areaName>CHINA</areaName>
        <bytes>5174496.00</bytes>
        <country>CHN</country>
        <countryName>CHINA</countryName>
        <region>APAC</region>
        <requests>5174496</requests>
        <summary>0</summary>
        <type>country</type>
    </summaryData>
    <...summaryData>
    (remaining summaries in collection...)
    <.../summaryData>
</collection>
</collections>
</ni>
</networkIdentifiers>
</service>
</services>
</accessGroup>

```

Sample 3 Request <https://ws.level3.com/contentAnalytics/v1.0/12345/BBBN12345/cdn.exempleni.com/1?groupBy=requestor&dateFrom=201004010000&dateTo=201004040000&id=170&dataInterval=day>

**Sample 3
Response -
Requestor Trend
for AG 12345 for
April 1 to April 4,
2010**

```

<data xsi:noNamespaceSchemaLocation=
  "https://ws.level3.com/schema/contentAnalyticsDataInterval/v1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <apiCorrelationId>CDNPortal-1283986736514-8796</apiCorrelationId>
    <point id="04/01/10">
      <item serviceResource="/12345/BBBN12345/cdn.exempleni.com/1">
        <bytes>1036800.00</bytes>
        <name>CHINA</name>
        <requests>1036800</requests>
        <time>04/01/10</time>
        <value>1036800.00</value>
      </item>
    </point>
    <point id="04/02/10">
      <item serviceResource="/12345/BBBN12345/cdn.exempleni.com/1">
        <bytes>1027296.00</bytes>

```

```

    <name>CHINA</name>

    <requests>1027296</requests>

    <time>04/02/10</time>

    <value>1027296.00</value>

  </item>

</point>

<point id="04/03/10">

  <item serviceResource="/12345/BBBN12345/cdn.exempleni.com/1">

    <bytes>1036800.00</bytes>

    <name>CHINA</name>

    <requests>1036800</requests>

    <time>04/03/10</time>

    <value>1036800.00</value>

  </item>

</point>

<point id="04/04/10">

  <item serviceResource="/12345/BBBN12345/cdn.exempleni.com/1">

    <bytes>1036800.00</bytes>

    <name>CHINA</name>

    <requests>1036800</requests>

    <time>04/04/10</time>

    <value>1036800.00</value>

  </item>

</point>
</data>

```

Sample 4 Request - URL Details <https://ws.level3.com/contentAnalytics/v1.0/12345/BBBN56179/cdn.exempleni.com/51?groupBy=urlDetails&dateFrom=201101010000&dateTo=201102010000>

Sample 4 Response - URL Details `<accessGroup id="12345" name="Level 3 - Internal Provisioning CDN" xsi:noNamespaceSchemaLocation="https://ws.level3.com/schema/contentAnalytics/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <apiCorrelationId>CDNPortal-1300479099140-1078</apiCorrelationId>
 <dataInterval/>`

```

<serviceResource>/12345</serviceResource>
<services>
  <service id="BBBN56179">
    <serviceResource>/12345/BBBN56179</serviceResource>
    <product>CACHING</product>
    <networkIdentifiers>
      <ni id="cdn.level3.com">
        <serviceResource>/12345/BBBN56179/cdn.level3.com

      </serviceResource>
    </networkIdentifiers>
    <collections>
      <collection id="51">
        <serviceResource>/12345/BBBN56179/cdn.level3.com/51
        </serviceResource>
        <summaryData>
          <id>905243</id>
          <url>/downloads/Fair_Opportunity_Process.pdf</url>
          <bytes>9.66</bytes>
          <requests>100</requests>
          <statusCodes code="2xx">
            <bytes>9.66</bytes>
            <requests>100</requests>
            <statusCode code="200">
              <bytes>9.66</bytes>
              <requests>100</requests>
            </statusCode>
          </statusCodes>
        </summaryData>
        <summaryData>
          ...
        </summaryData>
      </collection>
    </collections>
  </ni>
</networkIdentifiers>
</service>
</services>
</accessGroup>

```

Sample 5 Request - Status Codes <https://ws.level3.com/contentAnalytics/v1.0/12345/BBBN56179/cdn.exempleni.com/51?groupBy=statusCodes&dateFrom=201101010000&dateTo=201102010000>

Sample 5 Response - Status Codes

```

<accessGroup id="12345" name="Level 3 - Internal Provisioning CDN"
xsi:noNamespaceSchemaLocation=
"https://ws.level3.com/schema/contentAnalytics/v1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <apiCorrelationId>CDNPortal-1300479099140-1078</apiCorrelationId>
  <dataInterval/>
  <serviceResource>/12345</serviceResource>
  <services>
    <service id="BBBN56179">
      <serviceResource>/12345/BBBN56179</serviceResource>
      <product>CACHING</product>
      <networkIdentifiers>

```

```

<ni id="cdn.level3.com">
  <serviceResource>/12345/BBBN56179/cdn.level3.com

  </serviceResource>
  <collections>
    <collection id="51">
      <serviceResource>/12345/BBBN56179/cdn.level3.com/51
      </serviceResource>
      <statusCodes code="2xx">
        <bytes>37609.52</bytes>
        <requests>37900</requests>
        <statusCode code="200">
          <bytes>37601.89</bytes>
          <requests>37400</requests>
          <urlList>
            <summaryData>
              <id>40971</id>
              <url>/downloads/Rural_Broadband_Stimulus.pdf</url>
              <bytes>36961.58</bytes>
              <requests>36200</requests>
            </summaryData>
            ...
          </urlList>
          ...
        </statusCode>
      </statusCodes>
    </collection>
  </collections>
</ni>
</networkIdentifiers>
</service>
</services>
</accessGroup>

```

**Possible Status
and Error
Messages
Returned to Client**

See "Appendix: Error Responses" on page 122.

Cost Per Call

See Appendix: Cost per Call.

Content Manipulation

The Content Manipulation section includes the operations for setting up the Query String Handling mode (QSHMode).

By default, resources are cached using their complete URL, including the query string. Elements of the query string can be unique for each user, resulting in frequent cache misses. QSHMode allows removing the query string or portions of the query string from the cache key.

Currently, the API only allows suppressing the full query string, by setting the type attribute to the value "string". The action attribute determines if the query string is included in the cache key ("honor") or ignored ("ignore").

Retrieve Content Manipulation (QSHMode) Configuration Group

Base URI	https://ws.level3.com
Method	GET
Description	Retrieves the Content Manipulation Configuration Group with the QSHMode settings of a given Resource Group for service component or property.
URI Syntax	/serviceConfiguration/(version)/(scope)/ResourceGroups/(rgid)/ConfGroups/ContentManipulation version [CDATA[]] Values: "v1.0" (required) scope [CDATA[]] (AG)/(SCID)/[(ALIAS)] AG = Access Group SCID = Service Component Identifier ALIAS = Optional Alias identifies Property
Sample Request	URI: https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/test1.caching.cdn.level3.net/ResourceGroups/videos/ConfGroups/ContentManipulation https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/ResourceGroups/videos/ConfGroups/ContentManipulation
Sample Response	<pre>{ "ContentManipulation": { "qshmode": { "action": "ignore", "type": "string" } } }</pre>
Possible Status and Error Messages Returned to Client	200: Request Successful. See Appendix A – Status Codes & Error Messages for additional return codes and messages.

Create Content Manipulation (QSHMode) Configuration Group

Base URI	https://ws.level3.com
Method	POST
Description	Creates a new Content Manipulation Group. Supports a qshmode object, which includes a type identifier (only "string" supported) and the action attribute ("honor" or "ignore").
Body Syntax	<pre>{ "ContentManipulation": { "qshmode": { "type": "string", # required "action": ("honor" "ignore") # required } } }</pre>
URI Syntax	<p>/serviceConfiguration/(version)/(scope)/ResourceGroups/(rgid)/ConfGroups/ContentManipulation</p> <p>version [CDATA[]] Values: "v1.0" (required)</p> <p>scope [CDATA[]] (AG)/(SCID)/[(ALIAS)]</p> <p>AG = Access Group</p> <p>SCID = Service Component Identifier</p> <p>ALIAS = Optional Alias identifies Property</p>
Sample Request	<p>URI: https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/test1.caching.cdn.level3.net/ResourceGroups/videos/ConfGroups/ContentManipulation</p> <p>https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/ResourceGroups/videos/ConfGroups/ContentManipulation</p> <p>Body:</p> <pre>{ "ContentManipulation": { "qshmode": { "action": "ignore", "type": "string" } } }</pre>
Sample Response	<pre>{ "ContentManipulation": { "qshmode": { "action": "ignore", "type": "string" } } }</pre>
Possible Status and Error Messages Returned to Client	<p>201: Created.</p> <p>See Appendix A – Status Codes & Error Messages for additional return codes and messages.</p>

Update Content Manipulation (QSHMode) Configuration Group

Base URI	https://ws.level3.com
Method	PUT
Description	Updates a Content Manipulation Group. Supports a qshmode object, which includes a type identifier (only "string" supported) and the action attribute ("honor" or "ignore").
URI Syntax	<p>/serviceConfiguration/(version)/(scope)/ResourceGroups/(rgid)/ConfGroups/ContentManipulation</p> <p>version [CDATA[]] Values: "v1.0" (required)</p> <p>scope [CDATA[]] (AG)/(SCID)/[(ALIAS)]</p> <p> AG = Access Group</p> <p> SCID = Service Component Identifier</p> <p> ALIAS = Optional Alias identifies Property</p>
Body Syntax	<pre>{ "ContentManipulation": { "qshmode": { "type": "string", # required "action": ("honor" "ignore") # required } } }</pre>
Sample Request	<p>URI: https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/test1.caching.cdn.level3.net/ResourceGroups/videos/ConfGroups/ContentManipulation</p> <p>https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/ResourceGroups/videos/ConfGroups/ContentManipulation</p> <p>Body:</p> <pre>{ "ContentManipulation": { "qshmode": { "action": "honor", "type": "string" } } }</pre>
Sample Response	Empty
Possible Status and Error Messages Returned to Client	<p>204: Success no Content.</p> <p>See Appendix A – Status Codes & Error Messages for additional return codes and messages.</p>

Geo Definitions

Create Geo Definition

Base URI	https://ws.level3.com	
Method	POST	
Description	Creates a new Geo Definition. Requires a geoid and a list of country codes.	
	<div style="border: 1px solid black; background-color: #e0ffe0; padding: 5px;"> <p>Note: Geo services manage geo definitions at both the service component and property level (service component if no alias is provided, or a property when an alias has been specified). A Geo Definition is a combination of a unique ID (geoid) and a list of country codes.</p> </div>	
URI Syntax	/serviceConfiguration/(version)/(scope)/GeoDefs version [CDATA[]] Values: "v1.0" (required) scope [CDATA[]] AG)/(SCID)/[(ALIAS)] AG = Access Group SCID = Service Component Identifier ALIAS = Optional Alias identifies Property	
Sample Request	<pre>{ "geoid" : "<String containing [A-Za-z0-9_]>", # required "cc" : ["<country code>", ...] # required }</pre>	
Sample Response	<pre>{ "geoid": "NorthAmerica", "cc": ["CA", "US"] }</pre>	
Possible Status and Error Messages Returned to Client	200: Request Successful. 404: Not Found. See Appendix A – Status Codes & Error Messages for additional return codes and messages.	

Retrieve all Geo Definitions

Base URI	https://ws.level3.com
Method	GET
Description	Retrieves the array of geo definitions for service component or property. <div style="background-color: #e0ffe0; padding: 5px; margin-top: 10px;"> <p>Note: Geo services manage geo definitions at both the service component and property level (service component if no alias is provided, or a property when an alias has been specified). A Geo Definition is a combination of a unique ID (geoid) and a list of country codes.</p> </div>
URI Syntax	<p>/serviceConfiguration/(version)/(scope)/GeoDefs</p> <p>version [CDATA[]] Values: "v1.0" (required)</p> <p>scope [CDATA[]] (AG)/(SCID)</p> <p>AG = Access Group</p> <p>SCID = Service Component Identifier</p> <p>ALIAS = Optional Alias identifies Property</p>
Sample Request	<p>URI: https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/test1.caching.cdn.level3.net/GeoDefs</p> <p>https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/GeoDefs</p>
Sample Response	<pre>[{ "cc": "GE, FR", "geoid": "Europe" }, { "cc": "CA, US", "geoid": "NorthAmerica" }]</pre>
Possible Status and Error Messages Returned to Client	<p>200: Request successful.</p> <p>See Appendix A – Status Codes & Error Messages for additional return codes and messages.</p>

Retrieve a Specified Geo Definition

Base URI	https://ws.level3.com
Method	GET
Description	Retrieves a single geo definition by geoid. The results include the order attribute, indicating the position of this geo definition in the array of definitions.
	<div style="background-color: #e0ffe0; padding: 5px;"> <p>Note: Geo services manage geo definitions at both the service component and property level (service component if no alias is provided, or a property when an alias has been specified). A Geo Definition is a combination of a unique ID (geoid) and a list of country codes.</p> </div>
URI Syntax	<p>/serviceConfiguration/(version)/(scope)/GeoDefs/(geoid)</p> <p>version [CDATA[]] Values: "v1.0" (required)</p> <p>scope [CDATA[]] AG)/(SCID)/(ALIAS]</p> <p>AG = Access Group</p> <p>SCID = Service Component Identifier</p> <p>ALIAS = Optional Alias identifies Property</p>
Sample Request	<p>URI: https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/test1.caching.cdn.level3.net/GeoDefs/NorthAmerica</p> <p>https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/GeoDefs/NorthAmerica</p>
Sample Response	<pre>{ "cc": "CA, US", "geoid": "NorthAmerica", "order": 0 }</pre>
Possible Status and Error Messages Returned to Client	<p>200: Request Successful.</p> <p>404: Not Found.</p> <p>See Appendix A – Status Codes & Error Messages for additional return codes and messages.</p>

Update Geo Definition

Base URI	https://ws.level3.com
Method	PUT
Description	Update the country codes for an existing Geo Definition. <div style="background-color: #e0ffe0; padding: 5px; margin-top: 10px;"> <p>Note: Geo services manage geo definitions at both the service component and property level (service component if no alias is provided, or a property when an alias has been specified). A Geo Definition is a combination of a unique ID (geoid) and a list of country codes.</p> </div>
URI Syntax	<p>/serviceConfiguration/(version)/(scope)/GeoDefs</p> <p>version [CDATA[]] Values: "v1.0" (required)</p> <p>scope [CDATA[]] AG)/(SCID)/[(ALIAS)]</p> <p>AG = Access Group</p> <p>SCID = Service Component Identifier</p> <p>ALIAS = Optional Alias identifies Property</p>
Body Syntax	<pre>{ "geoid" : "<String containing [A-Za-z0-9_]>", # required "cc" : ["<country code>", ...] # required }</pre>
Sample Request	<p>URI: https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/test1.caching.cdn.level3.net/GeoDefs/NorthAmerica</p> <p>https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/GeoDefs/NorthAmerica</p> <p>Body:</p> <pre>{ "geoid": "NorthAmerica", "cc": ["CA", "US", "MX"] }</pre>
Sample Response	EMPTY
Possible Status and Error Messages Returned to Client	<p>204: Success no Content.</p> <p>See Appendix A – Status Codes & Error Messages for additional return codes and messages.</p>

Delete Geo Definition

Base URI	https://ws.level3.com
Method	DELETE
Description	Removes an existing Geo Definition identified by geoid. <div style="background-color: #e0ffe0; padding: 5px; margin-top: 10px;"> <p>Note: Geo services manage geo definitions at both the service component and property level (service component if no alias is provided, or a property when an alias has been specified). A Geo Definition is a combination of a unique id (geoid) and a list of country codes.</p> </div>
URI Syntax	<p>/serviceConfiguration/(version)/(scope)/GeoDefs</p> <p>version [CDATA[]] Values: "v1.0" (required)</p> <p>scope [CDATA[]] AG)/(SCID)/(ALIAS)]</p> <p>AG = Access Group</p> <p>SCID = Service Component Identifier</p> <p>ALIAS = Optional Alias identifies Property</p>
Sample Request	<p>URI: https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/test1.caching.cdn.level3.net/GeoDefs/NorthAmerica</p> <p>https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/GeoDefs/NorthAmerica</p>
Sample Response	EMPTY
Possible Status and Error Messages Returned to Client	<p>200: Request successful.</p> <p>See Appendix A – Status Codes & Error Messages for additional return codes and messages.</p>

Get Configuration Information

Base URI	https://ws.level3.com
Method	GET
Description	Return configuration information for Level 3 CDN service(s) associated to an Access Group, Service Component Identifier, or Network Identifier.
	<div style="background-color: #e0ffe0; padding: 5px;"> <p>Note: Although the serviceConfiguration service returns service component and network identifier information for streaming (Adobe FMS) service, no provisioning or configuration of FMS service is supported.</p> </div>
Schema Location	https://ws.level3.com/schema/serviceConfiguration/v1.0
URI Syntax	<p>/serviceConfiguration/(version)/(scope)?serviceType=(serviceType)</p> <p>/serviceConfiguration/(version)/(scope)?serviceType=(serviceType)</p> <p>version [CDATA[]] Values: "v1.0" (required)</p> <p>scope [CDATA[]] Must retain sequence that reflects hierarchy and cannot have an optional middle value.</p> <p>/(AG/ AG = Access Group</p> <p>or</p> <p>/(AG)/(SCID/ SCID = Service Component Identifier</p> <p>or</p> <p>/(AG)/(SCID)/(NI)/serviceType NI = Network Identifier serviceType</p> <p>Values:</p> <p>"storage" "o"</p> <p>"download" "d"</p> <p>"streaming" "s"</p> <p>Required for Scope to AG and SCID. Optional for NI.</p>
Sample URI	<p>https://ws.level3.com/serviceConfiguration/v1.0/12345</p> <p>https://ws.level3.com/serviceConfiguration/v1.0/12345/BBBN5678</p> <p>https://ws.level3.com/serviceConfiguration/v1.0/12345/BBBN5678/testaccount.origin.cdn.level3.net</p>
Sample Request	<p>https://ws.level3.com/serviceConfiguration/v1.0/12345/BBBN5678/testaccount.origin.cdn.level3.net</p> <p>Returns all streams in Access Group 12345 including those with no data.</p>
Sample Response	<pre><?xml version="1.0" encoding="UTF-8"?> <accessGroup id="12345" name="My Access Group" xsi:noNamespaceSchemaLocation= "https://ws.level3.com/schema/serviceConfiguration/v1.0"</pre>


```

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<apiCorrelationId>CDNPortal-1270071184562-8650
</apiCorrelationId>
<time>2010-02-01 00:00 +0000</time>
<serviceResource>/12345</serviceResource>
<services>
<service id="BBBN56789">
<serviceResource>/12345/BBBN56789</serviceResource>
<product>STORAGE</product>
<networkIdentifiers>
<ni id="testaccount">
<serviceResource>/12345/BBBN56789/testaccount.origin.cdn.level3.net
</serviceResource>

</networkIdentifiers>
</service>
</services>
</accessGroup>
    
```

Possible Status and Error Messages Returned to Client See Appendix A – Status Codes & Error Messages for additional return codes and messages.

HTTP Delivery

Category	Function	Scope
Property Origin Aliases	Retrieve, add, inactivate/reactivate a property	Service or Property
Origin Aliases	Retrieve and update the origin server section of a property	Property
Geo Definition	Retrieve, add, and remove aliases for a property	Property
Token	Retrieve, create, update, and delete a Geo definition for Geo Blocking	Service or Property
Resource Groups	Retrieve, add, update, and remove Token definitions for Token Authentication	Service or Property
Configuration Groups	Retrieve, add, update, and remove Resource Groups (typically path based URI filters)	Service or Property
Propagation of Configuration Changes	Retrieve, add, update, and remove Configuration Groups for Access Control, Cache Control, and Content Manipulation rules associated with a Resource Group filter.	Service or Property
	The Service Configuration services for HTTP Delivery potentially affect the operation of hundreds or thousands of CDN devices globally. To efficiently manage	

Category	Function	Scope
<p>Property</p>	<p>configuration across this global network, there is a multi-tier architecture for managing network configuration where changes are queued and propagate in batch. As a result, you may experience some delay before configuration changes take effect. In particular:</p> <ul style="list-style-type: none"> • New properties provisioned to the network may take up to 15 minutes until they are visible through the API and are available for additional service configuration. • Configuration changes may be confirmed immediately through the API. However, it may take up to 15 minutes to propagate to the edge servers and effect network behavior. <p>API Services for properties provide the means to create, retrieve, deactivate and reactivate properties. The create and get operations include only the basic attribution of a property, limited to origin and alias information.</p>	
<p>Adding and Editing Properties Adding and Updating Aliases</p>		<ul style="list-style-type: none"> • "Retrieve All Properties" on page 59 • "Retrieve Aliases" on page 31
<p>Creating and updating Match Pattern / URL Path (Resource Groups)</p>		<ul style="list-style-type: none"> • "Add Alias" on page 32 • "Remove Aliases" on page 33 • "Create Resource Group" on page 88 • "Retrieve Resource Groups" on page 89 • "Retrieve Individual Resource Group" on page 90 • "Retrieve Individual Resource Group" on page 90 • "Update Resource Group Order" on page 91 • "Update Resource Group " on page 92 • "Delete Resource Group" on page 93
<p>Creating and Updating Configuration Rules (Resource Groups)</p>		<ul style="list-style-type: none"> • "Retrieve Individual Resource Group" on page 90 • "Delete Configuration Group" on page 39
<p>Creating and Updating Geo Definitions</p>		<ul style="list-style-type: none"> • "Create Geo Definition" on page 50 • "Retrieve Geo Definition" on page 51 • "Retrieve Specified Geo Definition" on page 52 • "Update Geo Definition" on page 53 • "Delete Geo Definition" on page 54
<p>Creating and Updating Token Definitions</p>		<ul style="list-style-type: none"> • "Create Token Definition" on page 104 • "Retrieve Token Definitions" on page 105 • "Retrieve Individual Token Definition" on page 106 • "Delete Token Definition" on page 107
<p>Manage Access for Geo and Token Definitions</p>		<ul style="list-style-type: none"> • "Access Control Overview" on page 25 <ul style="list-style-type: none"> ▪ "Retrieve Access Control Configuration Group" on page 26 ▪ "Create Access Control Configuration Group" on page 27 ▪ "Update Access Control Configuration" on page 29
<p>Creating and Updating Cache</p>		<ul style="list-style-type: none"> • "Retrieve Cache Control (CCHO) Configuration Groups" on page 35

Control Management

Creating and Updating Query String Handling Mode (Content Manipulation)

- "Create Cache Control (CCHO) Configuration Group" on page 36
- "Update Cache Control (CCHO) Configuration Group" on page 37
- "Content Manipulation" on page 39
- "Retrieve Content Manipulation (QSHMode) Configuration Group" on page 47
- "Create Content Manipulation (QSHMode) Configuration Group" on page 48
- "Update Content Manipulation (QSHMode) Configuration Group" on page 49

Invalidations

Invalidations: Request Invalidation (POST)

Base URI	https://ws.level3.com										
Method	POST										
Description	Request to invalidate content										
Typical Use	Send invalidation requests for caching properties or streaming IDs.										
Schema Location	https://ws.level3.com/schema/invalidations/v1.0										
URI Syntax	<p>/invalidations/(version)/(scope)?[force=true][&ignoreCase=true][notification=name@example.com]</p> <table border="1"> <tr> <td>version</td> <td>Required version.</td> </tr> <tr> <td>scope /(AG)/(SCID⁴)/(NI) /originInvalidation</td> <td>Required. AG is Access Group⁵ ID. NI⁶ is network identifier name. originInvalidation is the origin name.</td> </tr> <tr> <td>force "true" 1 "false" 0 (not specified)</td> <td>Optional. Defines Invalidation Type. "true" matches Forced; "false" matches Normal radio buttons in the Media Portal interface. See Submitting invalidation requests.</td> </tr> <tr> <td>ignoreCase "true" 1 "false" 0 (not specified)</td> <td>Optional. Defines how URLs are matched to those in cache. See Submitting invalidation requests.</td> </tr> <tr> <td>notification "name@domain.com, name2@domain.com, name3@domain.com"</td> <td>Optional. Send notification to one or more email addresses when the invalidation process is complete. Use up to 250 characters total, including comma separators, with no spaces between addresses.</td> </tr> </table> <p>For more information about wildcards and valid path characters, see Path Attributes.</p>	version	Required version.	scope /(AG)/(SCID ⁴)/(NI) /originInvalidation	Required. AG is Access Group ⁵ ID. NI ⁶ is network identifier name. originInvalidation is the origin name.	force "true" 1 "false" 0 (not specified)	Optional. Defines Invalidation Type. "true" matches Forced; "false" matches Normal radio buttons in the Media Portal interface. See Submitting invalidation requests.	ignoreCase "true" 1 "false" 0 (not specified)	Optional. Defines how URLs are matched to those in cache. See Submitting invalidation requests.	notification "name@domain.com, name2@domain.com, name3@domain.com"	Optional. Send notification to one or more email addresses when the invalidation process is complete. Use up to 250 characters total, including comma separators, with no spaces between addresses.
version	Required version.										
scope /(AG)/(SCID ⁴)/(NI) /originInvalidation	Required. AG is Access Group ⁵ ID. NI ⁶ is network identifier name. originInvalidation is the origin name.										
force "true" 1 "false" 0 (not specified)	Optional. Defines Invalidation Type. "true" matches Forced; "false" matches Normal radio buttons in the Media Portal interface. See Submitting invalidation requests.										
ignoreCase "true" 1 "false" 0 (not specified)	Optional. Defines how URLs are matched to those in cache. See Submitting invalidation requests.										
notification "name@domain.com, name2@domain.com, name3@domain.com"	Optional. Send notification to one or more email addresses when the invalidation process is complete. Use up to 250 characters total, including comma separators, with no spaces between addresses.										
Example 1 Request	<p>https://ws.level3.com/invalidations/v1.0/12345/BBBN56789/cdn.exempleni.com?notification=name@domain.com,name2@domain.com</p> <p>Invalidate path(s) specified in the body of the POST request, force is false, ignoreCase is false. Send email notification to two addresses when complete.</p>										
Example 1 Body of POST	<p>Example 1 (uses wildcards):</p> <pre><paths> <path>/images/*.jpg</path> </paths></pre> <p>Invalidate ALL jpg objects (*.jpg) in the /images directory, including any subdirectories.</p>										

⁴Service Component Identification. Unique ID number in the order entry system associated to a billable component of a service. See Service.

⁵An administration method used to create and manage groups or sub-accounts that are then used to grant and limit access by service. Administration of those sub-accounts can be delegated to business units or customers.

⁶A common reference to the name used in the CDN. In Caching, the primary alias or "property". In Streaming, the Streaming ID, while in FMS 3.5, the primary supename plus Streaming ID. In Origin Storage, the VHost name.

Example 1 Response

```
<accessGroup id="12345" name="Level 3 - Internal Provisioning CDN"
  xsi:noNamespaceSchemaLocation=
    "https://ws.level3.com/schema/invalidations/v1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <apiCorrelationId>CDNPortal-1295308737747-3910</apiCorrelationId>
  <serviceResource>/12345</serviceResource>
  <services>
    <service id="BBBN56789">
      <serviceResource>/12345/BBBN56789</serviceResource>
      <product>CACHING</product>
      <networkIdentifiers>
        <ni id="cdn.level3.com">
          <serviceResource>/12345/BBBN56789/cdn.level3.com

          </serviceResource>
          <product>CACHING</product>
          <invalidations>
            <invalidation id="DAG_12345/9774@12345-13776-1295308738596"
              path="/images.*jpg"/>
          </invalidations>
        </ni>
      </networkIdentifiers>
    </service>
  </services>
</accessGroup>
```

Example 2 Request

https://ws.level3.com/invalidations/v1.0/12345/BBBN56789/cdn.exempleni.com?force=true
 Invalidate path(s) specified in the body of the POST request, force is true, ignoreCase is false.

Example 2 Body of POST

Example 2 (does not use wildcards):

```
<paths>
<path>/directory/structure</path>
<path>/directory/structure02</path>
</paths>
```

Invalidates two specific objects identified by their paths.

Path—Required. Up to 200 <path> elements are allowed in a single invalidation request. Can include wildcards. If a wildcard is used, only one <path> element can be specified in the request.

Note: The path must include a leading slash "/".

Example 2 Response

```
<?xml version="1.0" encoding="UTF-8"?>
<accessGroup id="12345" name="My Access Group"

  xsi:noNamespaceSchemaLocation=
    "https://ws.level3.com/schema/invalidations/v1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <apiCorrelationId>CDNPortal-1270071184562-8650</apiCorrelationId>
  <serviceResource>/12345</serviceResource>
  <services>
    <service id="BBBN56789">
      <serviceResource>/12345/BBBN56789</serviceResource>
      <product>CACHING</product>
```

```

<networkIdentifiers>
  <ni id="cdn.exempleni.com">
    <serviceResource>/12345/BBBN56789/cdn.exempleni.com

    </serviceResource>
  </ni>
</networkIdentifiers>
</services>
</accessGroup>

```

Example 3 Request <https://ws.level3.com/invalidations/v1.0/12345?ignoreCase=true&force=true>
 Invalidate path(s) specified in the body of the POST request, force is true, ignoreCase is true.

Example 3 Body of POST Example 3 (does not use wildcards):

```

<properties>
  <property>
    <name>exampleproperty1</name>
    <paths>
      <path>/example1.jpg</path>
      <path>/example2.jpg</path>
    </paths>
  </property>
  <property>
    <name>exampleproperty2</name>
    <paths>
      <path>/example3.jpg</path>
    </paths>
  </property>
</properties>

```

Invalidates multiple properties with one API request.

Example 3 Response <accessGroup id="12345" name="Level 3 - Internal Provisioning CDN" xsi:noNamespaceSchemaLocation="https://ws.level3.com/schema/invalidations/v1.0"

```

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <apiCorrelationId>CDNPortal-1295372721524-1835</apiCorrelationId>
  <serviceResource>/12345</serviceResource>
  <networkIdentifiers>
    <ni id="exampleproperty2">
      <serviceResource>/12345/BBBN56789/exampleproperty1

      </serviceResource>
      <product>CACHING</product>
      <invalidations>
        <invalidation id="DAG_12345/9774@12345-21470-1295372734654"
          path="/example3.jpg"/>
      </invalidations>
    </ni>
    <ni id="exampleproperty2">

```

```

    <serviceResource>/12345/BBBN56789/exampleproperty2
  </serviceResource>
  <product>CACHING</product>
  <invalidations>
    <invalidation id="DAG_12345/9774@12345-13776-1295372734654"
      path="/example1.jpg"/>
    <invalidation id="DAG_12345/9774@12345-13776-1295372734654"
      path="/example2.jpg"/>
  </invalidations>
</ni>
</networkIdentifiers>
</accessGroup>

```

Example 4 Request <https://ws.level3.com/invalidations/v1.0/12345>
 Invalidate multiple properties with a single request.

Example 4 Body of POST Example 4 (uses wildcards):
 XML:

```

<properties>
  <property>
    <name>exampleproperty1</name>
    <paths>
      <path>/example*.jpg</path>
    </paths>
  </property>
  <property>
    <name>exampleproperty2</name>
    <paths>
      <path>/example*.jpg</path>
    </paths>
  </property>
</properties>

```

JSON:

```

[
  {"name":"exampleproperty1",
  "paths":["example*.jpg"]},
  {"name":"exampleproperty2",
  "paths":["example*.jpg"]}
]

```

Example 4 Response

```

<accessGroup id="12345" name="Level 3 - Internal Provisioning CDN"
  xsi:noNamespaceSchemaLocation=
    "https://ws.level3.com/schema/invalidations/v1.0"

  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <apiCorrelationId>CDNPortal-1295374895751-0173</apiCorrelationId>
  <serviceResource>/12345</serviceResource>
  <networkIdentifiers>
    <ni id="exampleproperty1">
      <serviceResource>/12345/BBBN56789/www.level3test.com
    </serviceResource>
    <product>CACHING</product>
    <invalidations>
      <invalidation id="DAG_12345/9774@12345-21470-1295374897118"
        path="/example*.jpg"/>
    </invalidations>
  </ni>

```

```
<ni id="exampleproperty2">
  <serviceResource>/12345/BBBN56789/cdn.level3.com
  </serviceResource>
  <product>CACHING</product>
  <invalidations>
    <invalidation id="DAG_12345/9774@12345-13776-1295374897118"
      path="/example*.jpg"/>
  </invalidations>
</ni>
</networkIdentifiers>
</accessGroup>
```

Possible Status and Error Messages Returned to Client	See "Appendix: Error Responses" on page 122.
--	--

Cost Per Call	See Appendix: Cost per Call.
----------------------	------------------------------

Invalidations: Request Origin Invalidation (POST)

Base URI	https://ws.level3.com
Method	POST
Description	Request to invalidate content at the origin storage server
Typical Use	Send invalidation requests for caching properties or streaming IDs on the origin server.
Schema Location	https://ws.level3.com/schema/originInvalidations/v1.0
URI Syntax	<p>/originInvalidations/(version)/(scope)?[force=true][&ignoreCase=true][notification=name@example.com]</p> <p>version Required version.</p> <p>scope Required. AG is Access Group⁸ ID. ORIGIN is the origin storage server name. /(AG)/(SCID⁷)/(ORIGIN)</p> <p>force Optional. Defines Invalidation Type. "true" matches Forced; "false" matches Normal radio buttons in the Media Portal interface. See Submitting invalidation requests. "true" 1 "false" 0 (not specified)</p> <p>ignoreCase Optional. Defines how URLs are matched to those in cache. See Submitting invalidation requests. "true" 1 "false" 0 (not specified)</p> <p>notification Optional. Send notification to one or more email addresses when the invalidation process is complete. Use up to 250 characters total, including comma separators, with no spaces between addresses. "name@domain.com, name2@domain.com, name3@domain.com"</p> <p>For more information about wildcards and valid path characters, see Path Attributes.</p>
Example Request	<p>https://ws.level3.com/originInvalidations/v1.0/12345/BBBN56789/cdn.exampleorigin.com?notification=name@domain.com,name2@domain.com</p> <p>Invalidate path(s) specified in the body of the POST request. Send email notification to two addresses when complete.</p>
Example Body of POST	<p>Example (does not use wildcards):</p> <pre><paths> <path>/path1/example1.jpg</path> <path>/path1/example2.jpg</path> </paths></pre> <p>Invalidates two properties with one API request.</p>
Example Response	<pre><accessGroup id="12345" name="Level 3 - Internal Provisioning CDN"> <apiCorrelationId>CDNPortal-1289931176093-7961</apiCorrelationId></pre>

⁷Service Component Identification. Unique ID number in the order entry system associated to a billable component of a service. See Service.

⁸An administration method used to create and manage groups or sub-accounts that are then used to grant and limit access by service. Administration of those sub-accounts can be delegated to business units or customers.

```

<serviceResource>/202</serviceResource>
<services>
  <service id="BBBN56789">
    <serviceResource>/12345/BBBN56789</serviceResource>
    <product>CACHING</product>
    <networkIdentifiers>
      <ni id="cdn.level3.com">
        <serviceResource>/12345/BBBN56789/cdn.level3.com
        </serviceResource>
        <product>CACHING</product>
        <invalidations>
          <invalidation id="DAG_12345/9931@12345-13776-1289931178391"
            path="/path1/test1.jpg"/>
          <invalidation id="DAG_12345/9931@12345-13776-1289931178391"
            path="/path1/test2.jpg"/>
        </invalidations>
      </ni>
    </networkIdentifiers>
  </service>
</services>
</accessGroup>

```

Possible Status and Error Messages Returned to Client

See "Appendix: Error Responses" on page 122.

Cost Per Call

See Appendix: Cost per Call.

Invalidations: Check status (GET)

Base URI	https://ws.level3.com																
Method	GET																
Description	Request to see invalidation status																
Schema Location	https://ws.level3.com/schema/invalidationStatus/v1.0																
URI Syntax	<p>/invalidations/(version)/(scope)?[id=(invalidation tracking ID)]</p> <table border="1"> <tr> <td>version</td> <td>Required version.</td> </tr> <tr> <td>scope</td> <td>Required.</td> </tr> <tr> <td>/(AG)/</td> <td>Scope must retain sequence that reflects hierarchy.</td> </tr> <tr> <td>or</td> <td>Scope cannot have an optional middle value. AG is Access Group¹⁰</td> </tr> <tr> <td>/(AG)/(SCID⁹)/</td> <td>ID. NI¹¹ is network identifier name.</td> </tr> <tr> <td>or</td> <td></td> </tr> <tr> <td>/(AG)/(SCID)/(NI)/</td> <td></td> </tr> </table> <p>id (resource) Optional. Invalidation id – status for that id Left blank – status for all invalidations submitted within the scope</p> <p>Query string variables</p> <table border="1"> <tr> <td>verbose true 1 false 0</td> <td>Optional (defaults to false). If set to true or 1, all properties within scope will display regardless of whether they have invalidation status notifications. If left unspecified or set to false or 0, only those properties within the scope that have invalidation IDs will be returned.</td> </tr> </table>	version	Required version.	scope	Required.	/(AG)/	Scope must retain sequence that reflects hierarchy.	or	Scope cannot have an optional middle value. AG is Access Group ¹⁰	/(AG)/(SCID ⁹)/	ID. NI ¹¹ is network identifier name.	or		/(AG)/(SCID)/(NI)/		verbose true 1 false 0	Optional (defaults to false). If set to true or 1, all properties within scope will display regardless of whether they have invalidation status notifications. If left unspecified or set to false or 0, only those properties within the scope that have invalidation IDs will be returned.
version	Required version.																
scope	Required.																
/(AG)/	Scope must retain sequence that reflects hierarchy.																
or	Scope cannot have an optional middle value. AG is Access Group ¹⁰																
/(AG)/(SCID ⁹)/	ID. NI ¹¹ is network identifier name.																
or																	
/(AG)/(SCID)/(NI)/																	
verbose true 1 false 0	Optional (defaults to false). If set to true or 1, all properties within scope will display regardless of whether they have invalidation status notifications. If left unspecified or set to false or 0, only those properties within the scope that have invalidation IDs will be returned.																
Sample URIs	<p>https://ws.level3.com/invalidations/v1.0/12345</p> <p>Returns properties within Access Group 12345 for which there are invalidation IDs.</p>																
Sample 1 Request	<p>https://ws.level3.com/invalidations/v1.0/12345/BBBN56789/cdn.exempleni.com</p> <p>Returns status for all invalidations submitted for a given property.</p>																
Sample 1 Response	<pre><?xml version="1.0" encoding="UTF-8"?> <accessGroup id="12345" name="My Access Group" xsi:noNamespaceSchemaLocation= "https://ws.level3.com/schema/invalidationStatus/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <apiCorrelationId>CDNPortal-1270071184562-8650</apiCorrelationId> <serviceResource>/12345</serviceResource> <services> <service id="BBBN56789"></pre>																

⁹Service Component Identification. Unique ID number in the order entry system associated to a billable component of a service. See Service.

¹⁰An administration method used to create and manage groups or sub-accounts that are then used to grant and limit access by service. Administration of those sub-accounts can be delegated to business units or customers.

¹¹A common reference to the name used in the CDN. In Caching, the primary alias or "property". In Streaming, the Streaming ID, while in FMS 3.5, the primary supename plus Streaming ID. In Origin Storage, the VHost name.

```

<serviceResource>/12345/BBBN56789</serviceResource>
<product>CACHING</product>
<networkIdentifiers>
  <ni id="cdn.exempleni.com">
    <serviceResource>/12345/BBBN56789/cdn.exempleni.com
    </serviceResource>
    <product>CACHING</product>
    <invalidations>
      <invalidation id="DAG_12345/6789@12345-24674-1264192808754"
      percentComplete="55">
</paths>
      <path>/sample/01.jpg</path>
      <path>/sample/02.jpg</path>
</paths>
    </invalidation>
    <invalidation .../>
  </invalidations>
</ni>
<ni .../>
</networkIdentifiers>
</service>
</services>
</accessGroup>

```

**Sample 2
Request**

<https://ws.level3.com/invalidations/v1.0/12345/BBBN56789>
Returns status of invalidations within Access Group 12345 and SCID BBN56789.

**Sample 2
Response**

```

<accessGroup id="12345" name="Level 3 - Internal Provisioning CDN"

xsi:noNamespaceSchemaLocation=
"https://ws.level3.com/schema/invalidationStatus/v1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <apiCorrelationId>CDNPortal-1295369074375-2228</apiCorrelationId>
  <serviceResource>/12345</serviceResource>
  <services>
    <service id="BBBN56789">
      <serviceResource>/12345/BBBN56789</serviceResource>
      <product>CACHING</product>
      <networkIdentifiers>
        <ni id="cdn.level3.com">
          <serviceResource>/12345/BBBN56789/cdn.level3.com

          </serviceResource>
          <product>CACHING</product>
          <invalidations>
            <invalidation id="DAG_12345/9774@12345-13776-1295308738596"
            percentComplete="100">
              <paths>
                <path>/images.*cpp</path>
              </paths>
            </invalidation>
            <invalidation id="DAG_12345/9774@DAG_12345-13776-1295307
            828333" percentComplete="100">
              <paths>
                <path>/images/*.cpp</path>

```

```

        </paths>
      </invalidation>
    </invalidations>
  </ni>
</networkIdentifiers>
</service>
</services>
</accessGroup>

```

Example 3 Request

https://ws.level3.com/invalidations/v1.0/12345/BBBN56789/cdn.exempleni.com?id=DAG_12345/9774@12345-13776-1295374897118
Returns status for a specific invalidation tracking ID.

Example 3 Response

```

<accessGroup id="12345" name="Level 3 - Internal Provisioning CDN"

xsi:noNamespaceSchemaLocation=
"https://ws.level3.com/schema/invalidationStatus/v1.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <apiCorrelationId>CDNPortal-1295381524017-9511</apiCorrelationId>
  <serviceResource>/12345</serviceResource>
  <services>
    <service id="BBBN56789">
      <serviceResource>/12345/BBBN56789</serviceResource>
      <product>CACHING</product>
      <networkIdentifiers>
        <ni id="cdn.exempleni.com">
          <serviceResource>/12345/BBBN56789/cdn.exempleni.com

          </serviceResource>
          <product>CACHING</product>
          <invalidations>
            <invalidation id="DAG_12345/9774@12345-13776-1295374897118
              percentComplete="100">
              <paths>
                <path>/[eE][xX][aA][mM][pP][lL][eE]*.[jJ][pP][gG]</path>
              </paths>
            </invalidation>
          </invalidations>
        </ni>
      </networkIdentifiers>
    </service>
  </services>
</accessGroup>

```

Example 4 Request

<https://ws.level3.com/invalidations/v1.0/12345?verbose=true>
Returns the full node for Access Group 12345, including properties with invalidation IDs and those without.

Example 4 Response

```

<accessGroup id="12345" name="My Access Group">
  <apiCorrelationId>CDNPortal-1295389069002-0969</apiCorrelationId>
  <serviceResource>/12345</serviceResource>
  <services>
    <service id="BBBM54321">

```

```

<serviceResource>/12345/BBBM54321</serviceResource>
<product>STREAMING</product>
<networkIdentifiers>
  <ni id="exampleni1">
    <serviceResource>/12345/BBBM54321/exampleni1</serviceResource>
    <product>STREAMING</product>
    <type>On Demand</type>
  </ni>
</networkIdentifiers>
</service>
<service id="BBBN56789">
  <serviceResource>/12345/BBBN56789</serviceResource>
  <product>CACHING</product>
  <networkIdentifiers>
    <ni id=" exampleni2">
      <serviceResource>/12345/BBBN56789/exampleni2</serviceResource>
      <product>CACHING</product>
      <invalidations>
        <invalidation id="DAG_12345/9774@12345-13776-1295374489244"
          percentComplete="100">
          <paths>
            <path>/[eE] [xX] [aA] [mM] [pP] [lL] [eE]1.[jJ] [pP] [gG]</path>
          </paths>
        <invalidation id="DAG_12345/9774@DAG_12345-13776-129530
          7828333" percentComplete="100">
          <paths>
            <path>/images/*.png</path>
          </paths>
        </invalidation>
      </invalidations>
    </ni>
  </networkIdentifiers>
</service>

```

Continued, next page.

**Example 4
Response, cont.**

```

<service id="BBBP12346">
  <serviceResource>/12345/BBBP12346</serviceResource>
  <product>CACHING</product>
  <networkIdentifiers>
    <ni id="exampleni3">
      <serviceResource>/12345/BBBP12346/exampleni3</serviceResource>
      <product>CACHING</product>
    </ni>
  </networkIdentifiers>
</service>
</services>
</accessGroup>

```

**Possible Status
and Error
Messages
Returned to
Client**

See "Appendix: Error Responses" on page 122.

Cost Per Call

See Appendix: Cost per Call.

Key (Deprecated)

Base URI	https://ws.level3.com
Method	GET
Description	(Deprecated)Returns details of current credit balance for API key and date of next top-up
Schema Location	https://ws.level3.com/schema/key/v1.0
URI Syntax	/key/(version) version—Required version.
Sample Request	https://ws.level3.com/key/v1.0
Sample Response	<pre><?xml version="1.0" encoding="UTF-8"?> <apikey id="55555" xsi:noNamespaceSchemaLocation= "https://ws.level3.com/schema/key/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <apiCorrelationId>CDNPortal-1270071184562-8650 </apiCorrelationId> <assignedAccessGroup id="12345" name="My Access Group"/> <contact id="2345" name="contactemail@yourcompany.com"/> <role id="30" name="Reporting"/> <status>Active</status> <credits>702</credits> <next-top-up>2009-12-21 23:43:54 +0000</next-top-up> </apikey></pre>
Possible Status and Error Messages Returned to Client	See "Appendix: Error Responses" on page 122.
Cost Per Call	See Appendix: Cost per Call.

Mobile Login

Base URI	https://https://ws.level3.com/api/
Method	POST
Description	Returns a user-based key/secret combination
Typical Use	Provides already-authorized Media Portal users with a way to request credentials for API requests through a mobile application.
	Note: Exclude the API Key ID and Secret from your request.
Schema Location	https://https://ws.level3.com/api/v1.0/schema/key
URI Syntax	/(version)/mobileLogin version—Required version.
Example 1 Request	https://https://ws.level3.com/api/v1.0/mobileLogin User credentials specified in the body of the POST request.
Example 1 Body of POST	<pre><user> <username>{media portal username - email address}</username> <password>{media portal password}</password> </user></pre>
Example 1 Response	<pre><apikey xsi:noNamespaceSchemaLocation= "http://https://ws.level3.com/api/v1.0/schema/mobileLogin" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <apiCorrelationId>CDNPortal-1300303970131-6830</apiCorrelationId> <userApiKey>54321</userApiKey> <userApiSecret>QYsM3YqMeYxXbMTYDNRr</userApiSecret> <contact id="2121" name=""/> <status>Active</status> </apikey></pre>
Possible Status and Error Messages Returned to Client	See "Appendix: Error Responses" on page 122.
Cost Per Call	See Appendix: Cost per Call.

Origin Definition

Retrieve Origin Definition

Base URI	https://ws.level3.com
Method	GET
Description	Retrieves the origin server definition for the property.
Schema Location	https://ws.level3.com/schema/serviceConfiguration/v1.0
URI Syntax	/serviceConfiguration/(version)/(scope)/Origin version [CDATA[]] Values: "v1.0" (required) scope [CDATA[]] (AG)/(SCID) AG = Access Group SCID = Service Component Identifier
Sample Request	URI: https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/test1.caching.cdn.level3.net/Origin
Sample Response	<pre>{ "webroot": "/", "port": "8001", "protocol": "http", "host": "test.origin.com" }</pre>
Possible Status and Error Messages Returned to Client	200: Request Successful. See Appendix A – Status Codes & Error Messages for additional return codes and messages.

Update Origin Definition

Base URI	https://ws.level3.com
Method	PUT
Description	Update the origin server definition for the property. Only the property that requires changing needs to be sent.
Schema Location	https://ws.level3.com/schema/serviceConfiguration/v1.0
URI Syntax	/serviceConfiguration/(version)/(scope)/Origin version [CDATA[]] Values: "v1.0" (required) scope [CDATA[]] (AG)/(SCID) AG = Access Group SCID = Service Component Identifier
Body Syntax	<pre>{ "originserver" : { "host" : "<FQDN of the origin server>", # required "port" : "<port number of origin server>", # required "protocol" : ("http" "https"), # required "webroot" : "<root path>" # optional, default: "/" } }</pre>
Sample Request	URI: https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/test1.caching.cdn.level3.net/Origin <pre>{ "port" : "8007" }</pre>
Sample Response	Empty
Possible Status and Error Messages Returned to Client	204: Request with no Content See Appendix A – Status Codes & Error Messages for additional return codes and messages.

Origin Storage

Supported services for Origin Storage include the ability to create and delete storage Virtual Hosts and the ability to create, modify, and delete storage user FTP/SFTP accounts:

Service	Method	Function
serviceConfiguration	GET	Retrieve configuration details
serviceConfiguration	POST	Create new network identifier
serviceConfiguration	DELETE	Delete network identifier
storageUser	POST	Create new storage user for FTP/SFTP access to storage content
storageUser	PUT	Update storage user password
storageUser	DELETE	Delete storage user
orderStatus	GET	Check order status for service changes and new service orders

Create New Virtual Host

Base URI	https://ws.level3.com
Method	POST
Description	Create a new Virtual Host for the Level 3 Origin Storage service associated to an Access Group and Service Component Identifier.
Schema Location	https://ws.level3.com/schema/serviceConfiguration/v1.0
URI Syntax	<p>/serviceConfiguration/(version)/(scope)?requestor=(requestor)</p> <p>version [CDATA[]] Values: "v1.0" (required)</p> <p>scope [CDATA[]] Must retain sequence that reflects hierarchy and cannot have an optional middle value.</p> <p>/(AG)/(SCID)/(VH)/ AG = Access Group SCID = Service Component Identifier VH = Virtual Hostname</p> <p>requestor [CDATA[]] Dataset Values:</p> <pre>{ uploadLocation: "US East" "US West" "EU East" "EU West" vhostPrefix: example "mystorage" uploadProtocol: "Default" userNames: Dataset of user credentials { userId: "username-value", password: "password-value" } }</pre>
Sample URI	<p>https://ws.level3.com/serviceConfiguration/v1.0/12345/BBBN5678/testaccount.origin.cdn.level3.net</p> <p>Body:</p> <pre>{ "uploadLocation": "US East", "vhostPrefix": "testaccount", "uploadProtocol": "Default", "usernames": { "userId": "bobsmith", "password": "letmein" } }</pre>
Sample Request	<p>https://ws.level3.com/serviceConfiguration/v1.0/12345/BBBN5678/testaccount.origin.cdn.level3.net?requestor=bob.smith@customer.com</p> <p>Creates the Origin Storage virtual host "testaccount.origin.cdn.level3.net" as a child of SCID BBBN5678 giving access to user "bobsmith" with the password "letmein".</p>
Sample Response	Order ID

Possible Status and Error Messages Returned to Client	<p>201: Request Complete.—the Origin Storage virtual host account has been successfully provisioned and is available for use.</p> <p>202: Request Accepted.—no further action needed. The Level 3 CDN may need additional time to complete the provisioning of the service. Please use the Order Status API under General Services, to check for the completion of your request.</p> <p>See Appendix A – Status Codes & Error Messages for additional return codes and messages.</p>
--	--

Update User Account

Base URI	https://ws.level3.com
Method	PUT
Description	Update information of existing Users registered to a Level 3 Origin Storage service.
Schema Location	https://ws.level3.com/schema/storageUser/v1.0
URI Syntax	/storageUser/(version)/(scope) version [CDATA[]] Values: "v1.0" (required) scope [CDATA[]] Must retain sequence that reflects hierarchy and cannot have an optional middle value. /(AG)/(SCID)/(VH)/ AG = Access Group SCID = Service Component Identifier VH = Virtual Hostname
Sample URI	https://ws.level3.com/storageUser/v1.0/12345/BBBN5678/testaccount.origin.cdn.level3.net
Sample Request	https://ws.level3.com/storageUser/v1.0/12345/BBBN5678/testaccount.origin.cdn.level3.net Body: <pre>{ "userId": "janesmith", "password": "letmelogin" }</pre> Updates password to existing "janesmith" user in the "testaccount.origin.cdn.level3.net" account under SCID BBN5678, assigning the new password "letmelogin".
Sample Response	Order ID
Possible Status and Error Messages Returned to Client	200: Request Successful. See Appendix A – Status Codes & Error Messages for additional return codes and messages.

Delete User Account

Base URI	https://ws.level3.com
Method	DELETE
Description	Delete an existing User registered to a Level 3 Origin Storage service. <div style="border: 1px solid black; background-color: #e0ffe0; padding: 5px; margin-top: 10px;"> <p>Note: This service does not require or accept a content body. Please ensure that the Content-Length HTTP header is set to 0.</p> </div>
Schema Location	https://ws.level3.com/schema/storageUser/v1.0
URI Syntax	<p>/storageUser/(version)/(scope)</p> <p>version [CDATA[]] Values: "v1.0" (required)</p> <p>scope [CDATA[]] Must retain sequence that reflects hierarchy and cannot have an optional middle value.</p> <p>/(AG)/(SCID)/(VH)/</p> <p>AG = Access Group SCID = Service Component Identifier VH = Virtual Hostname</p>
Sample URI	https://ws.level3.com/storageUser/v1.0/12345/BBBN5678/testaccount.origin.cdn.level3.net
Sample Request	<p>https://ws.level3.com/storageUser/v1.0/12345/BBBN5678/testaccount.origin.cdn.level3.net</p> <p>Body:</p> <pre>{ "userId": "bobsmith", }</pre> <p>Deletes "janesmith" User from the "testaccount.origin.cdn.level3.net" account under SCID BBN5678</p>
Sample Response	Order ID
Possible Status and Error Messages Returned to Client	<p>200: Request Successful.</p> <p>See Appendix A – Status Codes & Error Messages for additional return codes and messages.</p>

Order Status

Base URI	https://ws.level3.com	
Method	GET	
Description	Get the status of a submitted API request and in the queue for processing.	
Schema Location	https://ws.level3.com/schema/orderStatus/v1.0	
URI Syntax	/ordersStatus/(version)/(scope)/(orderID)	
	version [CDATA[]]	Values: "v1.0" (required)
	scope [CDATA[]]	Must retain sequence that reflects hierarchy and cannot have an optional middle value.
	/(AG)/(SCID)/	AG = Access Group SCID = Service Component Identifier
	orderID	Values: Order ID value returned by the corresponding API request that was submitted for processing. (required)
Sample URI	https://ws.level3.com /orderStatus/v1.0/12345/BBBN5678/CHG-BBBR5678-12345	
Sample Request	https://ws.level3.com /orderStatus/v1.0/12345/BBBN5678/CHG-BBBR5678-12345 Returns status of Order ID CHG-BBBR5678-12345 for the corresponding API request submitted for processing.	
Sample Response	"Completed"	

Properties

Base URI	https://ws.level3.com
Method	GET
Description	Retrieves a list of available resources at the network identifier level wrapped in an access group and SCID object. The access group returns all available services (SCIDs) under the services array. Caching services will be identified as such by the product property ("CACHING"). The networkIdentifiers array will return a list of properties identified by their primary alias for every property of the given SCID. A status attribute will be returned for each property.
URI Syntax	/serviceConfiguration/(version)/(scope) version [CDATA[]] Values: "v1.0" (required) scope [CDATA[]] (AG)/(SCID) AG = Access Group SCID = Service Component Identifier
Sample Request	https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678
Sample Response	<pre> { "accessGroup": { "id": 1234, "name": "Customer Corporation", "apiCorrelationId": "API-XXXXXXXXXXXXXXXXXX", "serviceResource": "/1234", "services": [{ "id": "BBKR84567", "serviceResource": "/1234/BBBN5678", "product": "CACHING", "networkIdentifiers": [{ "id": " prop1.alias.cdn.level3.net", "serviceResource": "/1234/BBBN5678/prop1.alias.cdn.level3.net", "status": "active" }, { "id": " prop2.alias.cdn.level3.net ", "serviceResource": "/1234/BBBN5678/prop2.alias.cdn.level3.net", "status": "active" }] }] } } </pre>
Possible Status / Error Messages Returned to Client	200: Request Successful. See Appendix A – Status Codes & Error Messages for additional return codes and messages.

Create a New Property

Base URI	https://ws.level3.com
Method	POST
Description	<p>Create a new caching property under the given SCID and returns the newly created property object. The new property information will be returned in the success response.</p> <p>Note: At least one alias must be specified for each new property. That alias will be designated the primary alias. To reference the property in the Service Configuration API or any other media API, use the primary alias to specify scope. This is also the name that will be displayed in Media Portal for usage reporting, invalidation, and trouble ticketing. This primary alias cannot be deleted or changed.</p>
URI Syntax	<p>/serviceConfiguration/(version)/(scope)</p> <p>version [CDATA[]] Values: "v1.0" (required)</p> <p>scope [CDATA[]] (AG)/(SCID)</p> <p>AG = Access Group</p> <p>SCID = Service Component Identifier</p>
Body Syntax	<pre>{ "originserver" : { "host" : "<FQDN of the origin server>", # required "port" : "<port number of origin server>", # required "protocol" : ("http "https"), # required "webroot" : "<root path>" # optional, default: "/" }, "aliases" : ["<Footprint alias>", # list of aliases, minimum 1 ...] }</pre>
Sample Request	<p>URI: https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678</p> <p>Body:</p> <pre>{ "originserver" : { "host" : " test.origin.com", "port" : "8001", "protocol" : "http" }, "aliases" : ["test1.caching.cdn.level3.net", "test2.caching.cdn.level3.net"] }</pre>
Sample Response	<pre>{ "originserver" : { "host" : test.origin.com",</pre>

```
"port" : "8001",  
"protocol" : "http"  
},  
"cosid" : "111111", # Internal code, can be ignored  
"aliases" : [  
"test1.caching.cdn.level3.net",  
"test2.caching.cdn.level3.net"  
]  
}
```

**Possible Status
and Error
Messages
Returned to
Client**

201: Created.
See Appendix A – Status Codes & Error Messages for additional return codes and messages.

Retrieve a Property

Base URI	https://ws.level3.com
Method	GET
Description	Returns the property definition (origin and aliases) for the property. This includes the status ("active" or "inactive").
URI Syntax	/serviceConfiguration/(version)/(scope) version [CDATA[]] Values: "v1.0" (required) scope [CDATA[]] (AG)/(SCID) AG = Access Group SCID = Service Component Identifier ALIAS = Alias identifies Property
Sample Request	URI: https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/test1.caching.cdn.level3.net
Sample Response	<pre>{ "originserver" : { "host" : "test.origin.com", "port" : "8001", "protocol" : "http" }, "aliases" : ["test1.caching.cdn.level3.net", "test2.caching.cdn.level3.net"], "status": "active" }</pre>
Possible Status and Error Messages Returned to Client	200: Request Successful. See Appendix A – Status Codes & Error Messages for additional return codes and messages.

Deactivate a Property

Base URI	https://ws.level3.com
Method	DELETE
Description	Removing a property uses the Delete operation. The status changes to "inactive" and the property is deactivated but not permanently removed. It can be reactivated via a PUT operation.
URI Syntax	/serviceConfiguration/(version)/(scope) version [CDATA[]] Values: "v1.0" (required) scope [CDATA[]] (AG)/(SCID)/(ALIAS)
Sample URI	URI: https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/test1.caching.cdn.level3.net
Sample Request	URI: https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/test1.caching.cdn.level3.net
Sample Response	
Possible Status and Error Messages Returned to Client	200: Request Successful. See Appendix A – Status Codes & Error Messages for additional return codes and messages.

Reactivate a Property

Base URI	https://ws.level3.com
Method	PUT
Description	Reinstate a deactivated property. Status changes to "active".
URI Syntax	/serviceConfiguration/(version)/(scope) version [CDATA[]] Values: "v1.0" (required) scope [CDATA[]] (AG)/(SCID)/(ALIAS)
Sample URI	URI: https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/test1.caching.cdn.level3.net
Sample Request	URI: https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/test1.caching.cdn.level3.net
Sample Response	Empty
Possible Status and Error Messages Returned to Client	204: Success with no Content. See Appendix A – Status Codes & Error Messages for additional return codes and messages.

Resource Groups

Resource Groups are filters defining the scope for a group of configuration rules (via an associated Configuration Group, see next category). A given service component or property can have multiple Resource Groups defined, each identified by a unique ID (rgid).

The order of Resource Groups is important, as it is used to determine precedence for overlapping matches. Individually added Resource Groups are kept in the order they were added, each one given a sequence number starting from 0.

For any HTTP request, the matching Resource Group with the lowest sequence number governs the behavior for that request.

In addition to the standard Add, Update, and Delete operations, Resource Groups supports rearranging the order via a PUT operation.

A Resource Group consists of the following elements:

- **rgid**—string in [A-Za-z0-9_] format
- **rgtype**—resource group type. Currently limited to "path"
- **rgdef**—array of string identifiers defining the Resource Group

A request path is considered a match for the Resource Group if any of the expressions in rgdef match the request. Supported wild cards for rgdef include:

- '*' matches 0 or more characters
- '+' matches 1 or more non-/ (slash, aka directory separator) characters

Create Resource Group

Base URI	https://ws.level3.com
Method	POST
Description	Creates a new Resource Group. Requires a resource group id, a type, and a definition in the form of a string array. The response echoes the supplied resource group definition on success.
Body Syntax	<pre>{ "rgid": "<string with characters [A-Za-z0-9_]>", # required "rgtype": "path" # required "rgdef": ["<identifier>", ...], # required }</pre>
URI Syntax	/serviceConfiguration/(version)/(scope)/ResourceGroups version [CDATA[]] Values: "v1.0" (required) scope [CDATA[]] (AG)/(SCID)/[(ALIAS)] AG = Access Group SCID = Service Component Identifier ALIAS = Optional Alias identifies Property
Sample Request	URI: https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/test1.caching.cdn.level3.net/ResourceGroups https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/ResourceGroups Body: <pre>{ "rgdef": ["*.flv", "*.mp4", "*.f4p"], "rgtype": "path", "rgid": "videos" }</pre>
Sample Response	<pre>{ "rgdef": ["*.flv", "*.mp4", "*.f4p"], "rgtype": "path", "rgid": "videos" }</pre>
Possible Status and Error Messages Returned to Client	201: Created. See Appendix A – Status Codes & Error Messages for additional return codes and messages.

Retrieve Resource Groups

Base URI	https://ws.level3.com
Method	GET
Description	Retrieves the array of resource groups for service component or property.
URI Syntax	/serviceConfiguration/(version)/(scope)/ResourceGroups version [CDATA[]] Values: "v1.0" (required) scope [CDATA[]] (AG)/(SCID)/[(ALIAS)] AG = Access Group SCID = Service Component Identifier ALIAS = Optional Alias identifies Property
Sample Request	URI: https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/test1.caching.cdn.level3.net/ResourceGroups https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/ResourceGroups
Sample Response	<pre>[{ "rgdef": ["*.flv", "*.mp4", "*.f4p"], "order": 0, "rgtype": "path", "rgid": "videos" }, { "rgdef": ["*.jpg", "*.png"], "order": 1, "rgtype": "path", "rgid": "images" }]</pre>
Possible Status and Error Messages Returned to Client	200: Request Successful. See Appendix A – Status Codes & Error Messages for additional return codes and messages.

Retrieve Individual Resource Group

Base URI	https://ws.level3.com
Method	GET
Description	Retrieves a resource group for service component or property by rgid.
URI Syntax	/serviceConfiguration/(version)/(scope)/ResourceGroups/(rgid) version [CDATA[]] Values: "v1.0" (required) scope [CDATA[]] (AG)/(SCID)/[(ALIAS)] AG = Access Group SCID = Service Component Identifier ALIAS = Optional Alias identifies Property
Sample Request	URI: https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/test1.caching.cdn.level3.net/ResourceGroups/videos https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/ResourceGroups/videos
Sample Response	<pre>{ "rgdef": ["*.flv", "*.mp4", "*.f4p"], "order": 0, "rgtype": "path", "rgid": "videos" }</pre>
Possible Status and Error Messages Returned to Client	200: Request Successful. 404: Not Found. See Appendix A – Status Codes & Error Messages for additional return codes and messages.

Update Resource Group Order

Base URI	https://ws.level3.com
Method	PUT
Description	Change the order of Resource Groups. The order for a given resource group is specified via the order attribute. The numbering is consecutive and starts with 0. Values outside of the range for order will be mapped to such consecutive numbering (for example, if 3 groups are present and the order for one is specified as 8, it is converted to 2 (highest consecutive number starting from 0). The request does not require that all Resource Groups are included.
URI Syntax	/serviceConfiguration/(version)/(scope)/ResourceGroups/Order version [CDATA[]] Values: "v1.0" (required) scope [CDATA[]] (AG)/(SCID)/[(ALIAS)] AG = Access Group SCID = Service Component Identifier ALIAS = Optional Alias identifies Property
Body Syntax	<pre>[{ "rgid": "<string with characters [A-Za-z0-9_]>", # required "order": <integer> # required }, ...]</pre>
Sample Request	URI: https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/test1.caching.cdn.level3.net/ResourceGroups/Order https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/ResourceGroups/Order Body: <pre>[{ "rgid": "videos", "order": 0 }, { "rgid": "images", "order": 1 }]</pre>
Sample Response	Empty
Possible Status and Error Messages Returned to Client	204: Success no Content. See Appendix A – Status Codes & Error Messages for additional return codes and messages.

Update Resource Group

Base URI	https://ws.level3.com
Method	PUT
Description	Update a Resource Group. Only the rgdef attribute can be updated.
URI Syntax	/serviceConfiguration/(version)/(scope)/ResourceGroups/(rgid) version [CDATA[]] Values: "v1.0" (required) scope [CDATA[]] (AG)/(SCID)/(ALIAS)] AG = Access Group SCID = Service Component Identifier ALIAS = Optional Alias identifies Property
Body Syntax	<pre> { "rgid": "<string with characters [A-Za-z0-9_]>", # required "rgtype": "path" # optional "rgdef": ["<identifier>", ...], # required } </pre>
Sample Request	URI: https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/ test1.caching.cdn.level3.net/ResourceGroups https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/ ResourceGroups Body: <pre> { "rgdef": ["*.mp4], "rgtype": "path", "rgid": "videos" } </pre>
Sample Response	Empty
Possible Status and Error Messages Returned to Client	204: Success no Content. See Appendix A – Status Codes & Error Messages for additional return codes and messages.

Delete Resource Group

Base URI	https://ws.level3.com
Method	DELETE
Description	Removes a resource group for service component or property by rgid.
URI Syntax	/serviceConfiguration/(version)/(scope)/ResourceGroups/(rgid) version [CDATA[]] Values: "v1.0" (required) scope [CDATA[]] (AG)/(SCID)[/(ALIAS)] AG = Access Group SCID = Service Component Identifier ALIAS = Optional Alias identifies Property
Sample Request	URI: https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/test1.caching.cdn.level3.net/ResourceGroups https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/ResourceGroups Body: { "rgdef": ["*.mp4], "rgtype": "path", "rgid": "videos" }
Sample Response	Empty
Possible Status and Error Messages Returned to Client	204: Success no Content. See Appendix A – Status Codes & Error Messages for additional return codes and messages.

RTM (Caching)

Base URI	https://ws.level3.com
Method	GET
Description	Returns RTM data for given access group.
Schema Location	https://ws.level3.com/schema/cachingRTM/v1.0
URI Syntax	/rtm/cdn/(version)/(scope)?serviceType=(servicetype)[&geo=(geo)] [&property=true]
cdn	Required. Names CDN API as the correct engine.
version	Required version.
scope /(AG/ or /(AG)/(SCID ¹²)/ or /(AG)/(SCID)/(NI)/	Required. Scope for caching is by Access Group ID. AG is Access Group ¹³ ID. Scope cannot have an optional middle value. AG is Access Group ¹⁴ ID. NI ¹⁵ is network identifier name.
property=true	Optional. Provides RTM data at the NI level.
serviceType "caching" "c"	Required service type when using caching (as caching RTM is provided on the Access Group level).
geo "none" "region" "metro" "clientRegion"	Optional. Specify the level of granularity for geographic reporting. None reports at Access Group level, region at continent levels, and metro at metropolitan area levels.

Note: Queries without this parameter return metro data.

Sample 1 Request https://ws.level3.com/rtm/cdn/v1.0/12345?serviceType=c
Returns caching RTM data for AG 12345.

Sample 1 Response

```
<?xml version="1.0" encoding="UTF-8"?>
<accessGroup id="12345" name="My Access Group"
  xsi:noNamespaceSchemaLocation=
"https://ws.level3.com/schema/cachingRTM/v1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <apiCorrelationId>CDNPortal-1270071184562-8650</apiCorrelationId>
  <time>2010-02-01 00:00 +0000</time>
  <serviceResource>/12345</serviceResource>
  <metros>
```

¹²Service Component Identification. Unique ID number in the order entry system associated to a billable component of a service. See Service.

¹³An administration method used to create and manage groups or sub-accounts that are then used to grant and limit access by service. Administration of those sub-accounts can be delegated to business units or customers.

¹⁴An administration method used to create and manage groups or sub-accounts that are then used to grant and limit access by service. Administration of those sub-accounts can be delegated to business units or customers.

¹⁵A common reference to the name used in the CDN. In Caching, the primary alias or "property". In Streaming, the Streaming ID, while in FMS 3.5, the primary supernetname plus Streaming ID. In Origin Storage, the VHost name.

```

<metro name="Atlanta, GA" region="North America" latitude="33.44"
longitude="-84.23">
  <Mbps>184</Mbps>
  <requestsPerSecond>295.83</requestsPerSecond>
  <missMbps>0.13</missMbps>
  <missPerSecond>55.94</missPerSecond>
  <status404PerSec>1.54</status404PerSec>
  <status503PerSec>0</status503PerSec>
  <status504PerSec>6.59</status504PerSec>
  <status5xxPerSec>6.59</status5xxPerSec>
  <ptPerSec>1.28</PTPerSec>
  <hitRatePercentage>86.25</hitRatePercentage>
  <authPercentage>0.00</authPercentage>
  <signedPercentage>0.00</signedPercentage>
</metro>
<metro name="Vienna, AT" region="Europe" lat="48.14" long="16.2">
  <Mbps>1.57</Mbps>
  <requestsPerSecond>2.93</requestsPerSecond>
  <missMbps>0</missMbps>
  <missPerSecond>2.4</missPerSecond>
  <status404PerSec>0</status404PerSec>
  <status503PerSec>0</status503PerSec>
  <status504PerSec>0</status504PerSec>
  <status5xxPerSec>0</status5xxPerSec>
  <ptPerSec>0.03</PTPerSec>
  <hitRatePercentage>45.43</hitRatePercentage>
  <authPercentage>0.00</authPercentage>
  <signedPercentage>0.00</signedPercentage>
</metro>
</metros>
</accessGroup>

```

Sample 2 Request

<https://ws.level3.com/rtm/cdn/v1.0/12345?serviceType=c&geo=region>
Returns caching RTM data for AG 12345 at the region (continent) level.

Sample 2 Response - Geo

```

<?xml version="1.0" encoding="UTF-8"?>
<accessGroup id="12345" name="My Access Group"
xsi:noNamespaceSchemaLocation=
"https://ws.level3.com/schema/cachingRTM/v1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <accessGroup id="12345" name="My Access Group">
    <apiCorrelationId>CDNPortal-1292452796752-1022</apiCorrelationId>
    <serviceResource>/12345</serviceResource>
    <time>2010-12-15 22:37:01 +0000</time>
    <mbps>9.360001</mbps>
    <requestsPerSecond>4.299999</requestsPerSecond>
    <missMbps>0.01</missMbps>
    <missPerSecond>1.2199999</missPerSecond>
    <status404PerSec>0.0</status404PerSec>
    <status503PerSec>0.0</status503PerSec>
    <status504PerSec>0.0</status504PerSec>
    <status5xxPerSec>0.0</status5xxPerSec>
    <ptPerSec>0.16</ptPerSec>
    <hitRatePercentage>NaN</hitRatePercentage>
    <authPercentage>NaN</authPercentage>
    <signedPercentage>NaN</signedPercentage>

```



```

<regions>
  <region name="North America">
    <mbps>6.6800003</mbps>
    <requestsPerSecond>3.4099994</requestsPerSecond>
    <missMbps>0.01</missMbps>
    <missPerSecond>0.73999995</missPerSecond>
    <status404PerSec>0.0</status404PerSec>
    <status503PerSec>0.0</status503PerSec>
    <status504PerSec>0.0</status504PerSec>
    <status5xxPerSec>0.0</status5xxPerSec>
    <ptPerSec>0.16</ptPerSec>
    <hitRatePercentage>91.08448</hitRatePercentage>
    <authPercentage>0.0</authPercentage>
    <signedPercentage>0.0</signedPercentage>
  </region>
  <region name="..."
</region>
</regions>
</accessGroup>

```

Sample 3 Request - Access Group, by Property

<https://ws.level3.com/rtm/cdn/v1.0/12345/?serviceType=c&property=true>
 Returns caching RTM data for AG 12345 by property within the access group.

Sample 3 Response - Access Group, by Property

```

<accessGroup id="12345" name="My Access Group">
  <apiCorrelationId>CDNPortal-1300295389283-1600</apiCorrelationId>
  <serviceResource>/12345</serviceResource>
  <time>2012-03-15 22:44:01 +0000</time>
  <services>
    <service id="12345">
      <serviceResource>/12345</serviceResource>
      <networkIdentifiers>
        <ni id="cdn.level3.com">
          <serviceResource>/12345/BBBL1111/cdn.level3.com</serviceResource
          >
          <mbps>17482.7</mbps>
          <requestsPerSecond>26238.7</requestsPerSecond>
          <missMbps>921.05</missMbps>
          <missPerSecond>986.08</missPerSecond>
          <status404PerSec>3.49</status404PerSec>
          <status503PerSec>0</status503PerSec>
          <status504PerSec>0</status504PerSec>
          <status5xxPerSec>0</status5xxPerSec>
          <ptPerSec>233.2</ptPerSec>
          <hitRatePercentage>99.84</hitRatePercentage>
          <authPercentage>0.00</authPercentage>
          <signedPercentage>0.00</signedPercentage>
        </ni>
        <ni id="...">
          </ni>
      </networkIdentifiers>
    </service>
  </services>
</accessGroup>

```

Sample 4 Request - NI

<https://ws.level3.com/rtm/cdn/v1.0/12345/BL1111/cdn.level3.com?serviceType=c>

Returns caching RTM data for AG 12345 and metros within the property.

**Sample 4 Response -
 NI and Caching RTM
 Data**

```

<accessGroup id="12345" name="Win Update - MSFT End Cust"
  xsi:noNamespaceSchemaLocation=
    "https://ws.level3.com/schema/cachingRTM/v1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <apiCorrelationId>CDNPortal-1331852553017-5039</apiCorrelationId>
  <serviceResource>/12345</serviceResource>
  <services>
    <service id="BL1111">
      <serviceResource>/12345/BL1111</serviceResource>
      <product>CACHING</product>
      <networkIdentifiers>
        <ni id="cdn.level3.com">
          <serviceResource>/12345/BL1111/cdn.level3.com</serviceResource>
          <product>CACHING</product>
          <time>2012-03-15 23:02:33 +0000</time>
          <metros>
            <metro name="Atlanta, GA" region="North America"
              latitude="33.44" longitude="-84.23">
              <mbps>8864.86</mbps>
              <requestsPerSecond>9178.78</requestsPerSecond>
              <missMbps>39.59</missMbps>
              <missPerSecond>28.38</missPerSecond>
              <status404PerSec>0</status404PerSec>
              <status503PerSec>0</status503PerSec>
              <status504PerSec>0</status504PerSec>
              <status5xxPerSec>0</status5xxPerSec>
              <ptPerSec>78.35</ptPerSec>
              <hitRatePercentage>99.99</hitRatePercentage>
              <authPercentage>0.00</authPercentage>
              <signedPercentage>0.00</signedPercentage>
            </metro>
            <metro name="Boston, MA" region="North America"
              latitude="42.22" longitude="-71.2">
              <mbps>223.69</mbps>
              <requestsPerSecond>194.78</requestsPerSecond>
              <missMbps>3.17</missMbps>
              <missPerSecond>1.52</missPerSecond>
              <status404PerSec>0</status404PerSec>
              <status503PerSec>0</status503PerSec>
              <status504PerSec>0</status504PerSec>
              <status5xxPerSec>0</status5xxPerSec>
              <ptPerSec>9.08</ptPerSec>
              <hitRatePercentage>99.98</hitRatePercentage>
              <authPercentage>0.00</authPercentage>
              <signedPercentage>0.00</signedPercentage>
            </metro>
          </metros>
        </ni>
      </networkIdentifiers>
    </service>
  </services>
</accessGroup>

```

**Possible Status and
 Error Messages
 Returned to Client**

See "Appendix: Error Responses" on page 122.

Cost Per Call

See Appendix: Cost per Call.

Services Hierarchy (Partially Deprecated)

Base URI	https://ws.level3.com																				
Method	GET																				
Description	Returns service details for supplied identifier(s) - service hierarchy.																				
	Note: This service functions normally, but will not be enhanced or updated																				
Typical Use	To determine the SCID and NI portions of the scope that are used to build any of the API calls. Also used to determine whether the services have Content Analytics collections and their collections IDs.																				
Schema Location	https://ws.level3.com/schema/services/cdn/v1.0																				
URI Syntax	<p>/services/cdn/(version)/(scope)</p> <table border="1"> <tr> <td>cdn</td> <td>Required. Names CDN API as the correct engine.</td> </tr> <tr> <td>version</td> <td>Required version.</td> </tr> <tr> <td>scope</td> <td>Scope must retain sequence that reflects hierarchy. Scope cannot have an optional middle value. AG is Access Group¹⁷ ID. NI¹⁸ is network identifier name.</td> </tr> <tr> <td>/(AG/</td> <td></td> </tr> <tr> <td>or</td> <td></td> </tr> <tr> <td>/(AG)/(SCID¹⁶/</td> <td></td> </tr> <tr> <td>or</td> <td></td> </tr> <tr> <td>/(AG)/(SCID)/(NI/</td> <td></td> </tr> <tr> <td>?contentAnalytics=1</td> <td>Optional. Returns contentAnalytics hierarchy and collection IDs.</td> </tr> <tr> <td>?showConfiguration=true</td> <td>Optional. Returns current service configuration settings. Scope must be specified to the NI level.</td> </tr> </table>	cdn	Required. Names CDN API as the correct engine.	version	Required version.	scope	Scope must retain sequence that reflects hierarchy. Scope cannot have an optional middle value. AG is Access Group ¹⁷ ID. NI ¹⁸ is network identifier name.	/(AG/		or		/(AG)/(SCID ¹⁶ /		or		/(AG)/(SCID)/(NI/		?contentAnalytics=1	Optional. Returns contentAnalytics hierarchy and collection IDs.	?showConfiguration=true	Optional. Returns current service configuration settings. Scope must be specified to the NI level.
cdn	Required. Names CDN API as the correct engine.																				
version	Required version.																				
scope	Scope must retain sequence that reflects hierarchy. Scope cannot have an optional middle value. AG is Access Group ¹⁷ ID. NI ¹⁸ is network identifier name.																				
/(AG/																					
or																					
/(AG)/(SCID ¹⁶ /																					
or																					
/(AG)/(SCID)/(NI/																					
?contentAnalytics=1	Optional. Returns contentAnalytics hierarchy and collection IDs.																				
?showConfiguration=true	Optional. Returns current service configuration settings. Scope must be specified to the NI level.																				
Sample 1 Request	<p>https://ws.level3.com/services/cdn/v1.0/12345</p> <p>Returns services (and individual NIs) in AG 12345</p>																				
Sample 1 Response: Services under AG	<pre><?xml version="1.0" encoding="UTF-8"?> <accessGroup id="12345" name="My Access Group" xsi:noNamespaceSchemaLocation= "https://ws.level3.com/schema/services/cdn/v1.0 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <apiCorrelationId>CDNPortal-1270071184562-8650 </apiCorrelationId> <serviceResource>/12345</serviceResource> <services> <service id="BBBN56789"></pre>																				

¹⁶Service Component Identification. Unique ID number in the order entry system associated to a billable component of a service. See Service.

¹⁷An administration method used to create and manage groups or sub-accounts that are then used to grant and limit access by service. Administration of those sub-accounts can be delegated to business units or customers.

¹⁸A common reference to the name used in the CDN. In Caching, the primary alias or "property". In Streaming, the Streaming ID, while in FMS 3.5, the primary supename plus Streaming ID. In Origin Storage, the VHost name.

```

    <serviceResource>/12345/BBBN56789</serviceResource>
    <product>CACHING</product>
    <networkIdentifiers>
      <ni id="cdn.exempleni.com">
        <serviceResource>/12345/BBBN56789/cdn.exempleni.com
        </serviceResource>
        <product>CACHING</product>
      </ni>
    </networkIdentifiers>
  </service>
</services>
<networkIdentifiers>
  <ni id="sample-live">
    <serviceResource>/12345/BBBN10111/sample-live
    </serviceResource>
    <product>STREAMING</product>
    <type>Live</type>
  </ni>
</networkIdentifiers>
</accessGroup>

```

**Sample 2
Request**

<https://ws.level3.com/services/cdn/v1.0/12345/BBBN56789>
Returns NIs in AG 12345 under SCID BBN56789.

**Sample 2
Response: NIs
under a SCID**

```

<?xml version="1.0" encoding="UTF-8"?>
<accessGroup id="12345" name="My Access Group"
  xsi:noNamespaceSchemaLocation=
    "https://ws.level3.com/schema/services/cdn/v1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <apiCorrelationId>CDNPortal-1270071184562-8650
  </apiCorrelationId>
  <serviceResource>/12345</serviceResource>
  <services>
    <service id="BBBN56789">
      <serviceResource>/12345/BBBN56789</serviceResource>
      <product>CACHING</product>
      <networkIdentifiers>
        <ni id="cdn.exempleni.com">
          <serviceResource>/12345/BBBN56789/cdn.exempleni.com
          </serviceResource>
          <product>CACHING</product>
        </ni>
      </networkIdentifiers>
    </service>
  </services>
</accessGroup>

```

**Sample 3
Request**

<https://ws.level3.com/services/cdn/v1.0/12345/BBBN56789/cdn.exempleni.com>
Returns details on NI cdn.exempleni.com in AG 12345 under SCID BBN56789.

**Sample 3
 Response: NI
 Details**

```
<?xml version="1.0" encoding="UTF-8"?>
<accessGroup id="12345" name="My Access Group"
  xsi:noNamespaceSchemaLocation=
    "https://ws.level3.com/schema/services/cdn/v1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <apiCorrelationId>CDNPortal-1270071184562-8650
  </apiCorrelationId>
  <serviceResource>/12345</serviceResource>
  <services>
    <service id="BBBN56789">
      <serviceResource>/12345/BBBN56789</serviceResource>
      <product>CACHING</product>
      <networkIdentifiers>
        <ni id="cdn.exempleni.com">
          <serviceResource>/12345/BBBN56789/cdn.exempleni.com
          </serviceResource>
          <product>CACHING</product>
        </ni>
      </networkIdentifiers>
    </service>
  </services>
</accessGroup>
```

**Sample 4
 Request**

<https://ws.level3.com/services/cdn/v1.0/12345?contentAnalytics=1>

Returns Content Analytics-enabled NIs for AG 12345 and their collection detail.

**Sample 4
 Response:
 Content
 Analytics
 Hierarchy for
 AG 12345**

```
<accessGroup id="12345" name="My Access Group"
  xsi:noNamespaceSchemaLocation
    ="https://ws.level3.com/schema/services/cdn/v1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <apiCorrelationId>CDNPortal-1283545533522-5690</apiCorrelationId>
  <serviceResource>/12345</serviceResource>
  <services>
    <service id="BBBL12345">
      <serviceResource>/12345/BBBL12345</serviceResource>
      <product>CACHING</product>
      <networkIdentifiers>
        <ni id="smooth.level3.com">
          <serviceResource>/202/BBBZ12345/my.sampleni.com
          </serviceResource>
          <product>CACHING</product>
          <contentAnalytics>
            <active>>true</active>
            <serviceLevel>premium</serviceLevel>
            <samplingRate>premium</samplingRate>
            <collections>
              <collection id="3">
                <serviceResource>/202/BBBZ12345/my.sampleni.com/1
                </serviceResource>
                <pattern>*</pattern>
                <token>-</token>
                <type>pattern</type>
                <active>true</active>
                <asn>>false</asn>
```

```

        <detailed>true</detailed>
        <lastModified>2009-10-08 15:12:49 +0000
        </lastModified>
    </collection>
</collections>
</contentAnalytics>
</ni>
</networkIdentifiers>
</service>
</services>
</accessGroup>

```

**Sample 5
Request**

[https://ws.level3.com/services/cdn/v1.0/\[accessGroupID\]/\[SCID\]/www.level3.com?showConfiguration=true](https://ws.level3.com/services/cdn/v1.0/[accessGroupID]/[SCID]/www.level3.com?showConfiguration=true)
Returns configuration settings for www.level3.com.

**Sample 5
Response:
Show
Configuration
of
www.level3.com**

```

<accessGroup id="[accessGroupID]" name="Level 3 Communications,
LLC." xsi:noNamespaceSchemaLocation="https://ws.level3.com//
schema/servicesv1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <apiCorrelationId>API-b3dac651-06b7-4b47-8257-eec836b76f50</apiCorrelationId>
  <serviceResource>/[accessGroupID]</serviceResource>
  <services>
    <service id="[SCID]">
      <serviceResource>/[accessGroupID]/[SCID]</serviceResource>
      <product>CACHING</product>
      <networkIdentifiers>
        <ni id="www.level3.com">
          <serviceResource>/[accessGroupID]/[SCID]/www.level3.com</serviceResource>
          <product>CACHING</product>
          <active>Y</active>
          <serviceDetails>
            <name>www.level3.com</name>
            <installDate>06/11/2013</installDate>
            <aliases>
              <primaryAlias>www.level3.com</primaryAlias>
              <secondaryAliases/>
              <altIDExtendedAliases/>
            </aliases>
            <originHostname>v1.level3.com</originHostname>
            <cacheFillProtocol>http</cacheFillProtocol>
            <cacheFillPort>80</cacheFillPort>
            <altWebRoot/>
            <fillHostHeader>v1.level3.com</fillHostHeader>
            <level3Origin>>false</level3Origin>
            <adminFlags>
              <flag name="url_stats">standard</flag>
            </adminFlags>
            <coserverWideFlags>
              <flag name="qshmode">*</flag>
              <flag name="mp4_scrubbable">ext=.mp4,start=start,end=end</flag>
            </coserverWideFlags>
            <ruleBases>
              <ruleBase name="Requests to Origin Server"/>
              <ruleBase name="Responses from Origin Server"/>
            </ruleBases>
          </serviceDetails>
        </ni>
      </networkIdentifiers>
    </service>
  </services>
</accessGroup>

```

Possible Status and Error Messages Returned to Client See "Appendix: Error Responses" on page 122.

Cost Per Call See Appendix: Cost per Call.

Token Definitions

The following services allow managing Token Authentication definitions at both the service component and property level (service component if no alias is provided, or a property when an alias has been specified).

Token Authentication allows customers to protect content from URL tampering or unauthorized re-use or re-publication via email forwarding or deep linking to content. Using shared secrets defined in the tokens, a URL signature appended to the query string of the resource URL can be validated by the CDN before serving content without contacting the customer environment for authentication.

Up to 10 token definitions can be defined at the service component and each property level. A token definition is defined by a unique ID, represented by an integer between 0 and 9, a string representing the secret, and an optional start and end time stamp. Time stamps are given in UNIX epoch format, for example 145251839. The time stamps define a period for which the token will be in effect.

Note: Since resources are cached by URL and the use of Token Authentication requires insertion of query string parameters into the request URL, Query String Handling Mode should also be implemented as appropriate. Typically, the token parameters (stime, etime, encoded) should be excluded from inclusion in the cached resource URL.

For more information about Token Authentication, see Media Portal Help at <https://mediaportal.level3.com/webhelp/help/Content/ServicesDocs-Caching/CDNAssetSecurity/Section3URLTokenAuthentication.htm>

Create Token Definition

Base URI	https://ws.level3.com
Method	POST
Description	Creates a new Token Definition. Requires a token id and a secret. A start and end time stamp in UNIX epoch are optional parameters that can be included. The response echoes the supplied token definition on success.
URI Syntax	/serviceConfiguration/(version)/(scope)/Tokens version [CDATA[]] Values: "v1.0" (required) scope [CDATA[]] (AG)/(SCID)/(ALIAS)] AG = Access Group SCID = Service Component Identifier ALIAS = Optional Alias identifies Property
Body Syntax	<pre>{ "tokenId": "<token id, and integer between 0 and 9>", # required "secret": "<shared secret string>", # required "start": "<the secret start time in UNIX epoch>", # optional "end": "<the secret end time in UNIX epoch>", # optional }</pre>
Sample Request	URI: https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/test1.caching.cdn.level3.net/Tokens https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/Tokens Body: <pre>{ "tokenId": "1", "secret": "secret1", "start": "1415252687", "end": "1415339087" }</pre>
Sample Response	<pre>{ "tokenId": "1", "secret": "secret1", "start": "1415252687", "end": "1415339087" }</pre>
Possible Status and Error Messages Returned to Client	201: Created. See Appendix A – Status Codes & Error Messages for additional return codes and messages.

Retrieve all Token Definitions

Base URI	https://ws.level3.com
Method	GET
Description	Retrieves the array of token definitions for service component or property.
URI Syntax	/serviceConfiguration/(version)/(scope)/Tokens version [CDATA[]] Values: "v1.0" (required) scope [CDATA[]] (AG)/(SCID)/[(ALIAS)] AG = Access Group SCID = Service Component Identifier /serviceConfiguration/(version)/(scope)/Tokens
Sample Request	URI: https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/ test1.caching.cdn.level3.net/Tokens https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/Tokens
Sample Response	<pre> { "tokenId": "0", "secret": "secret0", "start": "1415251839", "end": "1415338239" }, { "tokenId": "1", "secret": "secret1", "start": "1415252687", "end": "1415339087" }] </pre>
Possible Status and Error Messages Returned to Client	200: Request Successful. See Appendix A – Status Codes & Error Messages for additional return codes and messages.

Retrieve an Individual Token Definition

Base URI	https://ws.level3.com
Method	GET
Description	Retrieves a single token definition for a service component or property.
URI Syntax	/serviceConfiguration/(version)/(scope)/Tokens/(tokenid) version [CDATA[]] Values: "v1.0" (required) scope [CDATA[]] (AG)/(SCID)[/(ALIAS)] AG = Access Group SCID = Service Component Identifier ALIAS = Optional Alias identifies Property
Sample Request	URI: https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/ test1.caching.cdn.level3.net/Tokens/1 https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/Tokens/1
Sample Response	<pre>{ "tokenId": "1", "secret": "secret1", "start": "1415252687", "end": "1415339087" }</pre>
Possible Status and Error Messages Returned to Client	200: Request Successful. See Appendix A – Status Codes & Error Messages for additional return codes and messages.

Delete a Token Definition

Base URI	https://ws.level3.com
Method	DELETE
Description	Removes a token definition for a service component or property.
URI Syntax	/serviceConfiguration/(version)/(scope)/Tokens/(tokenid) version [CDATA[]] Values: "v1.0" (required) scope [CDATA[]] (AG)/(SCID)/[(ALIAS)] AG = Access Group SCID = Service Component Identifier ALIAS = Optional Alias identifies Property
Sample Request	URI: https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/ test1.caching.cdn.level3.net/Tokens/1 https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/Tokens/1
Sample Response	Empty
Possible Status and Error Messages Returned to Client	200: Request Successful. See Appendix A – Status Codes & Error Messages for additional return codes and messages.

Usage Reporting

Base URI	https://ws.level3.com																
Method	GET																
Description	Returns Caching property, Streaming ID or Origin Storage account usage details for an access group, SCID or network identifier. Returns the service hierarchy with usage data.																
Schema Locations	https://ws.level3.com/schema/cachingUsage/v1.0 https://ws.level3.com/schema/streamingUsage/v1.0 https://ws.level3.com/schema/originUsage/v1.0																
URI Syntax	<code>/usage/cdn/(version)/(scope)?serviceType=(serviceType)[&geo=(geo)][&dateFrom=(dateFrom)&dateTo=(dateTo)][&dateMonth=(dateMonth)]</code> <table border="1"> <tr> <td>cdn</td> <td>Required. Names CDN API as the correct engine.</td> </tr> <tr> <td>version</td> <td>Required version.</td> </tr> <tr> <td>scope /(AG/ or /(AG)/(SCID¹⁹)/ or /(AG)/(SCID)/(NI)/</td> <td>Required. Scope must retain sequence that reflects hierarchy. AG is Access Group²⁰ ID. NI²¹ is network identifier name.</td> </tr> <tr> <td>serviceType "caching" "c" "streaming" "s" "origin" "o"</td> <td>Required. Service type: required for scope to AG and SCID. Optional for NI.</td> </tr> <tr> <td>geo "region" "metro"</td> <td>Optional. Return results by server region or by server metropolitan area.</td> </tr> <tr> <td>dateFrom yyyyMMddHHnn</td> <td>Optional. Date range – starting date/time</td> </tr> <tr> <td>dateTo yyyyMMddHHnn</td> <td>Optional. Date range – end date/time</td> </tr> <tr> <td>dateMonth yyyyMM</td> <td>Optional. One month. Used to report caching summary data that includes the 95%.</td> </tr> </table>	cdn	Required. Names CDN API as the correct engine.	version	Required version.	scope /(AG/ or /(AG)/(SCID ¹⁹)/ or /(AG)/(SCID)/(NI)/	Required. Scope must retain sequence that reflects hierarchy. AG is Access Group ²⁰ ID. NI ²¹ is network identifier name.	serviceType "caching" "c" "streaming" "s" "origin" "o"	Required. Service type: required for scope to AG and SCID. Optional for NI.	geo "region" "metro"	Optional. Return results by server region or by server metropolitan area.	dateFrom yyyyMMddHHnn	Optional. Date range – starting date/time	dateTo yyyyMMddHHnn	Optional. Date range – end date/time	dateMonth yyyyMM	Optional. One month. Used to report caching summary data that includes the 95%.
cdn	Required. Names CDN API as the correct engine.																
version	Required version.																
scope /(AG/ or /(AG)/(SCID ¹⁹)/ or /(AG)/(SCID)/(NI)/	Required. Scope must retain sequence that reflects hierarchy. AG is Access Group ²⁰ ID. NI ²¹ is network identifier name.																
serviceType "caching" "c" "streaming" "s" "origin" "o"	Required. Service type: required for scope to AG and SCID. Optional for NI.																
geo "region" "metro"	Optional. Return results by server region or by server metropolitan area.																
dateFrom yyyyMMddHHnn	Optional. Date range – starting date/time																
dateTo yyyyMMddHHnn	Optional. Date range – end date/time																
dateMonth yyyyMM	Optional. One month. Used to report caching summary data that includes the 95%.																
Sample 1 Request	<code>https://ws.level3.com/usage/cdn/v1.0/12345?serviceType=caching&dateFrom=201002010000&dateTo=201002040000</code>																
Sample 1 Response	<code><?xml version="1.0" encoding="UTF-8" ?></code>																

¹⁹Service Component Identification. Unique ID number in the order entry system associated to a billable component of a service. See Service.

²⁰An administration method used to create and manage groups or sub-accounts that are then used to grant and limit access by service. Administration of those sub-accounts can be delegated to business units or customers.

²¹A common reference to the name used in the CDN. In Caching, the primary alias or "property". In Streaming, the Streaming ID, while in FMS 3.5, the primary supename plus Streaming ID. In Origin Storage, the VHost name.

**- Caching Usage for
 AG 12345 Hierarchy
 (Summary/Table
 Data) for Feb 1 –
 Feb 4 2010**

```

<accessGroup id="12345" name="My Access Group"
xsi:noNamespaceSchemaLocation=
  "https://ws.level3.com/schema/cachingUsage/v1.0"
xmlns:xsi="http://www.w3.org/2001X/MLSchema-instance">
  <apiCorrelationId>CDNPortal-1270071184562-8650

  </apiCorrelationId>
  <serviceResource>/12345</serviceResource>
  <parentId/>
  <summaryData>
    <volume>155092.02829</volume>
    <averageThroughput>4786.79100</averageThroughput>
    <peakThroughput>6415.05689</peakThroughput>
    <requests>1660617484</requests>
    <averageRequestsPerSecond>6406.70326

    </averageRequestsPerSecond>
    <peakRequestsPerSecond>8855.37000</peakRequestsPerSecond>
    <originVolume>47705.25872</originVolume>
    <averageOriginThroughput>1472.38453
    </averageOriginThroughput>
    <peakOriginThroughput>2050.03164</peakOriginThroughput>
    <cacheEfficiency>76.47638</cacheEfficiency>
  </summaryData>
  <services>
    <service id="BBBN56789">
      <product>CACHING</product>
      <serviceResource>/12345/BBBN56789</serviceResource>
      <parentId>12345</parentId>
      <summaryData.../>
      <networkIdentifiers>
        <ni id="cdn.example.level3.com">
          <serviceResource>/12345/BBBN56789/cdn.example.level3.com
          </serviceResource>
          <product>CACHING</product>
          <parentId>BBBN56789</parentId>
          <summaryData.../>
        </ni>
      </ni.../>
      </networkIdentifiers>
    </service>
  </services>
</accessGroup>

```

Sample 2 Request <https://ws.level3.com/usage/cdn/v1.0/12345?serviceType=caching&dateMonth=201012>

Sample 2 Response - Caching usage summary for 12345/BBBN56179/cdn.level3.com for December 2012

```

<?xml version="1.0" encoding="UTF-8" ?>
<accessGroup id="12345" name="BBBN56179"
xsi:noNamespaceSchemaLocation=
  "https://ws.level3.com/schema/cachingUsage/v1.0"
xmlns:xsi="http://www.w3.org/2001X/MLSchema-instance">
<accessGroup id="12345" name="Level 3 - Internal Provisioning CDN">

```

```

<apiCorrelationId>CDNPortal-1292450115423-3484
</apiCorrelationId>
<dataInterval>monthly</dataInterval>
<serviceResource>/12345</serviceResource>
<services>
  <service id="BBBN56179">
    <serviceResource>/12345/BBBN56179</serviceResource>
    <product>CACHING</product>
    <networkIdentifiers>
      <ni id="cdn.level3.com">
        <serviceResource>/12345/BBBN56179/cdn.level3.com

        </serviceResource>
      <summaryData>
        <volume>302.79262</volume>
        <averageThroughput>0.50065</averageThroughput>
        <peakThroughput>3.12019</peakThroughput>
        <mbps95>0.97615</mbps95>
        <requests>11699692</requests>
        <averageRequestsPerSecond>2.41809
          </averageRequestsPerSecond>
        <peakRequestsPerSecond>8.04667</peakRequestsPerSecond>
        <originVolume>6.07578</originVolume>
        <averageOriginThroughput>0.56257
          </averageOriginThroughput>
        <peakOriginThroughput>5.75245</peakOriginThroughput>
        <cacheEfficiency>98.03289</cacheEfficiency>
      </summaryData>
    </ni>
  </networkIdentifiers>
</service>
</services>
</accessGroup>

```

Sample 3 Request <https://ws.level3.com/usage/cdn/v1.0/12345?serviceType=s&dateFrom=201101010000&dateTo=201102010000>

Sample 3 Response - Streaming Usage Summary

```

<?xml version="1.0" encoding="UTF-8" ?>
<accessGroup id="12345" name="My Access Group"
xsi:noNamespaceSchemaLocation=
  "https://ws.level3.com/schema/streamingUsage/v1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <apiCorrelationId>CDNPortal-1270071184562-8650

  </apiCorrelationId>
  <serviceResource>/12345</serviceResource>

  <dataInterval>hourly</dataInterval>

  <summaryData>
    <requests>3706056</requests>
    <volume>141104.97930</volume>

```

```

    <averageConnectedPlayers>44485.80000
  </averageConnectedPlayers>

  <averageMbps>19411.64214</averageMbps>
  <averageDuration>16630.97593</averageDuration>
  <peakConnectedPlayers>16976.00000</peakConnectedPlayers>

  <peakMbps>6614.40501</peakMbps>
</summaryData>
<services>
  <service id="BBBP11111">
    <serviceResource>/12345/BBBP11111</serviceResource>
    <product>STREAMING</product>

    <streamType>On-Demand
  </streamType>

  <summaryData>
    <requests>3017979</requests>
    <volume>94.65392</volume>
    <averageConnectedPlayers>2968.17391
  </averageConnectedPlayers>
    <averageMbps>5595.56396</averageMbps>
    <averageDuration>2.66299</averageDuration>
    <peakConnectedPlayers>67.00000</peakConnectedPlayers>

    <peakMbps>1683.82148</peakMbps>
  </summaryData>
  <networkIdentifiers>
    <ni id="samplestream">
      <serviceResource>/12345/BBBP11111/samplestream

      </serviceResource>
      <summaryData>
        <requests>1</requests>
        <volume>0.00013</volume>

        <averageConnectedPlayers/>
        <averageMbps/>
        <averageDuration>0.00000</averageDuration>
        <peakConnectedPlayers/>
        <peakMbps/>

```

See next page.

Sample 3, cont.

```

    </summaryData>
  </ni>
  <ni.../>
</networkIdentifiers>
</service>
<service.../>
</services>
<networkIdentifiers.../>
</accessGroup>

```

Sample 4 Request <https://ws.level3.com/usage/cdn/v1.0/12345?serviceType=o&dateFrom=201012010000&dateTo=201012152359>

Sample 4 Response - Origin Usage from December 1, 2010 through December 15, 2010

```
<?xml version="1.0" encoding="UTF-8" ?>
<accessGroup id="12345" name="My Access Group"
xsi:noNamespaceSchemaLocation=
"https://ws.level3.com/schema/originUsage/v1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <apiCorrelationId>CDNPortal-1270071184562-8650
</apiCorrelationId>
  <serviceResource>/12345</serviceResource>
  <dataInterval>daily</dataInterval>
  <summaryData>
    <peakUsage>1014.45465</peakUsage>
  </summaryData>
  <services>
    <service id="BBBP11111">
      <serviceResource>/12345/BBBP11111</serviceResource>
      <product>STORAGE</product>
      <summaryData>
        <peakUsage>0.0002345</peakUsage>
      </summaryData>
      <networkIdentifiers>
        <ni id=" origin.cdn.test.net ">
          <serviceResource>/12345/BBBP11111/origin.cdn.test.net
          </serviceResource>
          <summaryData>
            <peakUsage>0.00002345</peakUsage>
          </summaryData>
        </ni>
        <ni .../>
      </networkIdentifiers>
    </service>
    <service .../>
  </services>
</networkIdentifiers .../>
</accessGroup>
```

Possible Status and Error Messages Returned to Client See "Appendix: Error Responses" on page 122.

Cost Per Call See Appendix: Cost per Call.

Usage Reporting – Data Interval

Base URI	https://ws.level3.com
Method	GET
Description	Returns usage details for an access group, scid or network identifier, divided by the specified data interval. The data intervals available depend on the service. For example, caching usage can be selected in data intervals of 5 minutes, 1 hour or 1 day.
Schema Locations	<p>https://ws.level3.com/schema/cachingUsageDataInterval/v1.0</p> <p>https://ws.level3.com/schema/streamingUsageDataInterval/v1.0</p> <p>https://ws.level3.com/schema/originUsageDataInterval/v1.0</p>
URI Syntax	<p>/usage/cdn/(version)/(scope)?[serviceType=(serviceType)][&geo=(region metro)]&dateFrom=(dateFrom)&dateTo=(dateTo)[&dataInterval=(dataInterval)]</p> <p>cdn Required. Names CDN API as the correct engine.</p> <p>version Required version.</p> <p>scope Scope must retain sequence that reflects hierarchy. AG is Access Group²³ ID. NI²⁴ is network identifier name.</p> <p>/(AG)/</p> <p>or</p> <p>/(AG)/(SCID²²)/</p> <p>or</p> <p>/(AG)/(SCID)/(NI)/</p> <p>serviceType Service type: required for scope to AG and SCID. Optional for NI.</p> <p>“caching” “c” “streaming” “s” “origin” “o”</p> <p>geo Optional. Returns results by server region or by server area for each time interval requested.</p> <p>“region” “metro”</p> <div style="background-color: #e0ffe0; padding: 5px; border: 1px solid black; margin-top: 10px;"> <p>Note: Specifying both dataInterval and geo may result in very large result sets and very long API service times. For best performance, limit the time and property scope of each service call.</p> </div>

²²Service Component Identification. Unique ID number in the order entry system associated to a billable component of a service. See Service.

²³An administration method used to create and manage groups or sub-accounts that are then used to grant and limit access by service. Administration of those sub-accounts can be delegated to business units or customers.

²⁴A common reference to the name used in the CDN. In Caching, the primary alias or "property". In Streaming, the Streaming ID, while in FMS 3.5, the primary surname plus Streaming ID. In Origin Storage, the VHost name.

dateFrom yyyyMMddHHnn	Date range—starting date/time
dateTo yyyyMMddHHnn	Date range—end date/time
dataInterval	Optional. If not specified, return service hierarchy with usage data. If specified, returns data only at the requested scope level.
"5min" "hourly" "daily" "monthly"	Caching—5min—maximum 1 day Hourly—maximum 20 days Daily—maximum 90 days
	Note: 95th% caching data is only returned with "monthly".
	Streaming
"hourly" "daily" "monthly"	Origin Storage
"daily" "monthly"	

Sample 1 Request <https://ws.level3.com/usage/cdn/v1.0/12345/BBBZ11111?serviceType=caching&dateFrom=201002010000&dateTo=201002040000&dataInterval=hourly>

Sample 1 Response - Caching Usage Hourly dataInterval for AG 12345 from Feb 1 – Feb 4 2010

```
<?xml version="1.0" encoding="UTF-8"?>
<data xsi:noNamespaceSchemaLocation=
"https://ws.level3.com/schema/cachingUsageDataInterval/v1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <apiCorrelationId>CDNPortal-1270071184562-8650</apiCorrelationId>
  <point id="02/01/2010 00:00:00">
    <item id="/12345/BBBZ11111" name="BBBZ11111">
      <volume>0.00978</volume>
      <averageThroughput>0.02174</averageThroughput>
      <peakThroughput>0.05068</peakThroughput>
      <requests>210</requests>
      <averageRequestsPerSecond>0.05833</averageRequestsPerSecond>
      <peakRequestsPerSecond>0.12000</peakRequestsPerSecond>
      <originVolume>0.00962</originVolume>
      <averageOriginThroughput>0.02138</averageOriginThroughput>
      <peakOriginThroughput>0.05087</peakOriginThroughput>
      <cacheEfficiency>47.95000</cacheEfficiency>
    </item>
    <item .../>
  </point>
  <point .../>
</data>>
```

Sample 2 Request <https://ws.level3.com/usage/cdn/v1.0/12345/BBBN56179/cdn.level3.com?serviceType=caching&dateFrom=201011010000&dateTo=201012312359&dataInterval=monthly>

Sample 2 Response
- Caching Usage
Monthly
dataInterval for AG
12345 from
November &
December, 2010

```
<?xml version="1.0" encoding="UTF-8"?>
<data xsi:noNamespaceSchemaLocation=
  "https://ws.level3.com/schema/cachingUsageDataInterval/v1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<data>
  <apiCorrelationId>CDNPortal-1292451181264-8498</apiCorrelationId>
  <point id="11/01/2010 00:00:00">
    <item serviceResource="/12345/BBBN56179/cdn.level3.com">
      <volume>933.35894</volume>
      <averageThroughput>0.55756</averageThroughput>
      <peakThroughput>3.26684</peakThroughput>
      <mbps95>0.23985</mbps95>
      <requests>35718620</requests>
      <averageRequestsPerSecond>2.66716</averageRequestsPerSecond>
      <peakRequestsPerSecond>17.32333</peakRequestsPerSecond>
      <originVolume>11.45859</originVolume>
      <averageOriginThroughput>1.06098</averageOriginThroughput>
      <peakOriginThroughput>2.32732</peakOriginThroughput>
      <cacheEfficiency>98.79000</cacheEfficiency>
    </item>
  </point>
  <point id="12/01/2010 00:00:00">
    <item serviceResource="/12345/BBBN56179/cdn.level3.com">
      <volume>302.79262</volume>
      <averageThroughput>0.50065</averageThroughput>
      <peakThroughput>3.12019</peakThroughput>
      <mbps95>0.97615</mbps95>
      <requests>11699692</requests>
      <averageRequestsPerSecond>2.41809</averageRequestsPerSecond>
      <peakRequestsPerSecond>8.04667</peakRequestsPerSecond>
      <originVolume>6.07578</originVolume>
      <averageOriginThroughput>0.56257</averageOriginThroughput>
      <peakOriginThroughput>5.75245</peakOriginThroughput>
      <cacheEfficiency>98.03000</cacheEfficiency>
    </item>
  </point>
</data>
```

Sample 3 Request `https://ws.level3.com/usage/cdn/v1.0/12345/BBBP11111?
serviceType=s&dateFrom=201002010000&dateTo=
201001040000&dataInterval=hourly`

Sample 3 Response - Streaming Usage with Hourly dataInterval on February 1, 2010, from 1 AM to 4 AM

```
<?xml version="1.0" encoding="UTF-8"?>
<data xsi:noNamespaceSchemaLocation=
  "https://ws.level3.com/schema/cachingUsageDataInterval/v1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <apiCorrelationId>CDNPortal-1270071184562-8650
  </apiCorrelationId>
  <point id="02/01/2010 00:00:00">
    <item serviceResource="/12345/BBBP11111">
      <requests>5926</requests>
      <volume>0.19233</volume>
      <averageConnectedPlayers>147.00000</averageConnectedPlayers>
      <averageMbps>96.76320</averageMbps>
      <averageDuration>1.62319</averageDuration>
      <peakConnectedPlayers>37.00000</peakConnectedPlayers>
      <peakMbps>35.47849</peakMbps>
    </item>
    <item ... />
  </point>
  <point id="02/01/2010 01:00:00" .../>
  <point id="02/01/2010 02:00:00" .../>
</data>
```

Sample 4 Request `https://ws.level3.com/usage/cdn/v1.0/12345/BBBR222222?
serviceType=o&dateFrom=201002010000&dateTo=
201002235959&dataInterval=daily`

Sample 4 Response - Origin Usage with Daily dataInterval on February 1 & 2, 2010

```
<?xml version="1.0" encoding="UTF-8"?>
<data xsi:noNamespaceSchemaLocation=
  "https://ws.level3.com/schema/originUsageDataInterval/v1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <apiCorrelationId>CDNPortal-1270071184562-8650
  </apiCorrelationId>
  <point id="02/01/2010 00:00:00">
    <item serviceResource="/12345/BBBR222222">
      <volume>0.12496</volume>
    </item>
    <item serviceResource="/12345/BBC11111/origin.cdn.test.net">
      <volume>0.22514</volume>
    </item>
  </point>
  <point id="02/02/2010 00:00:00" .../>
</data>
```

Possible Status and Error Messages Returned to Client See "Appendix: Error Responses" on page 122.

Cost Per Call See Appendix: Cost per Call.

Usage by Access Group

Base URI	https://ws.level3.com
Method	GET
Description	Returns a hierchical representation of your Access Group data for all three services. Includes monthly, or for the current month, month-to-date data through yesterday's total. Data is updated once a day.
Typical Use	This report provides a picture of the data at each level of the access group hierarchy for all three services. Available data includes: Volume, Requests, Cache Efficiency, 95th Percentile, Peak Mbps, and Peak Requests per second.
Schema Location	https://ws.level3.com/schema/summaryUsage/v1.0
URI Syntax	<p>/usage/cdn/(version)/(scope-AGID only)?summary=true&dateMonth=<yyyymm></p> <p>cdn Required. Names CDN API as the correct engine.</p> <p>version Required version.</p> <p>scope AG is Access Group²⁵ ID.</p> <p>/(AG/</p> <p>dateMonth Date - requested month</p> <p>yyyymm</p>
Sample Request	<p>https://ws.level3.com/usage/cdn/v1.0/12345?summary=true&dateMonth=201007</p> <p>Returns caching usage By Access Group for AG 12345 in July 2010.</p>
Sample Response - By Access Group Usage for AG 12345 Hierarchy (Summary/Table Data) for July 2010	<pre><?xml version="1.0" encoding="UTF-8" ?> <accessGroup id="12345" name="My Access Group" xsi:noNamespaceSchemaLocation= "https://ws.level3.com/schema/summaryUsage/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <accessGroup id="12345" name="Level 3 - Internal Provisioning CDN"> <apiCorrelationId>CDNPortal-1282780385870-4284</apiCorrelationId> <serviceResource>/12345</serviceResource> <caching> <volume>3332.36564</volume> <requests>21219582</requests> <mbps95>11.49775</mbps95> </caching> <streaming> <volume>169536.88131</volume> <views>2714843</views></pre>

²⁵An administration method used to create and manage groups or sub-accounts that are then used to grant and limit access by service. Administration of those sub-accounts can be delegated to business units or customers.

```

</streaming>
<origin>
  <peakUsage>1733.15556</peakUsage>
</origin>
<accessGroups>
  <accessGroup id="1235" name="L3EUInternal">
    <serviceResource>/1235</serviceResource>
    <caching>
      <volume>2529.27034</volume>
      <requests>18873353</requests>
      <mbps95>7.96394</mbps95>
    </caching>
    <streaming>
      <volume>167923.79626</volume>
      <views>1213187</views>
    </streaming>
    <origin>
      <peakUsage>450.48279</peakUsage>
    </origin>
  </accessGroups>
</accessGroup>

```

Possible Status and Error Messages Returned to Client	See "Appendix: Error Responses" on page 122.
<hr/>	
Cost Per Call	See Appendix: Cost per Call.

Vyvx Reservations

Vyvx Reservation – Get Active Status

Base URI	https://ws.level3.com/vyvx/1.0
Method	GET
Description	<p>Get the status of all active services (reservations).</p> <div style="border: 1px solid black; background-color: #e0ffe0; padding: 5px; margin: 5px 0;"> <p>Note: This service does not require or accept a content body. Please ensure that the Content-Length HTTP header is set to 0.</p> </div> <p>The call returns a JSON collection with one keyed entry per reservation. The “state” field describes the overall state of the reservation, and the “ingress_state” field describes the informed estimation of the state of the signal at the ingress to the Level 3 network.</p> <p>Possible state values:</p> <ul style="list-style-type: none"> • online—service is operating normally • impaired—service is operating normally, with some non-service-impacting anomaly • degraded—service is still operational, but is experiencing some service-impacting anomaly • offline—service is not operational • unknown—the state cannot be determined
URI Syntax	/customer/service/_active/status
Sample Request	https://ws.level3.com/vyvx/1.0/customer/service/_active/status
Sample Response	<pre>{ "data": { "services": { "1234567": { "state": "offline", "ingress_state": "online" }, "9999999": { "state": "offline", "ingress_state": "online" } } }, "time": "2017-01-02T15:04:05.000Z" }</pre>
Possible Status and Error Messages Returned to Client	<p>200: Request Successful.</p> <p>400: Bad Request – Missing required request information</p> <p>401: Not Authorized – Access to unauthorized resource</p> <p>500: Internal Server Error – Internal errors (DB, etc)</p>
Possible State and Ingress State Responses Returned to Client	See Appendix: Vyvx State and Ingress State Response Definitions.

Vyvx Reservation – Get Service Status

Base URI	https://ws.level3.com/vyvx/1.0
Method	GET
Description	<p>Get the status of a single active service (reservation).</p> <div style="background-color: #e0ffe0; padding: 5px; border: 1px solid black;"> <p>Note: This service does not require or accept a content body. Please ensure that the Content-Length HTTP header is set to 0.</p> </div> <p>The call returns a JSON collection with one keyed entry per reservation. The “state” field describes the overall state of the reservation, and the “ingress_state” field describes the informed estimation of the state of the signal at the ingress to the Level 3 network.</p> <p>Possible state values:</p> <ul style="list-style-type: none"> • online—service is operating normally • impaired—service is operating normally, with some non-service-impacting anomaly • degraded—service is still operational, but is experiencing some service-impacting anomaly • offline—service is not operational • unknown—the state cannot be determined
URI Syntax	<p>/customer/service/(id)/status</p> <p>id : Reservation ID (required)</p>
Sample Request	https://ws.level3.com/vyvx/1.0/customer/service/1234567/status
Sample Response	<pre>{ "data": { "services": { "1234567": { "state": "offline", "ingress_state": "online" } }, "time": "2006-01-02T15:04:05.000Z" } }</pre>
Possible Status and Error Messages Returned to Client	<p>200: Request Successful.</p> <p>400: Bad Request – Missing required request information</p> <p>401: Not Authorized – Access to unauthorized resource</p> <p>500: Internal Server Error – Internal errors (DB, etc)</p>
Possible State and Ingress State Responses Returned to Client	See Appendix: Vyvx State and Ingress State Response Definitions.

Vyvx Reservation – Get Service History

Base URI	https://ws.level3.com/vyvx/1.0
Method	GET
Description	Get the alarm history of a single active service (reservation).
	Note: This service does not require or accept a content body. Please ensure that the Content-Length HTTP header is set to 0.
URI Syntax	/customer/service/(id)/history?start=(start)&end=(end)
Sample Request	https://ws.level3.com/vyvx/1.0/customer/service/1234567/history?2016-01-02T15:04:05.000Z&end=2016-01-03T15:04:05.000Z
Sample Response	<pre>{ "data": { "services": { "1234567": { "history": { "events": [{ "interval": { "start": "2016-01-02T16:00:00.000Z", "end": "2016-01-02T17:00:00.000Z", "state": "offline", "message": "TS Loss" }] } } } }, "time": "2006-01-02T15:04:05.000Z" }</pre>
Possible Status and Error Messages Returned to Client	200: Request Successful. 400: Bad Request – Missing required request information 401: Not Authorized – Access to unauthorized resource 500: Internal Server Error – Internal errors (DB, etc)
Possible State and Ingress State Responses Returned to Client	See Appendix: Vyvx State and Ingress State Response Definitions.

Appendix: Error Responses

Whenever the service returns a 4xx or 5xx response code to the caller, the entity body will contain an XML representation of more detailed error information to allow diagnosis and appropriate handling:

```
<?xml version="1.0" encoding="UTF-8" ?>
<error>
  <errorCode>21708</errorCode>
  <message>Unknown service type.</message>
  <httpStatus>400</httpStatus>
  <apiCorrelationId>CDNPortal-1270071184562-8650</apiCorrelationId>
</error>
```

Description	Error Code	Associated HTTP Status
URI cannot be parsed	21700	400
Unknown Access Group	21701	404
Unknown SCID	21702	404
Unknown network identifier	21703	404
SCID is not a child of specified Access Group	21704	404
Network Identifier is not child of specified SCID	21705	404
Network identifier is not child of specified Access Group	21706	404
Not authorized for requested scope (Access Group, SCID or Network Identifier)	21707	401
Unknown service type	21708	400
Unknown group options	21709	400
Unknown report dimensions	21710	400
Specified report dimensions unavailable for service type	21711	400
Date range too wide	21712	403
Date range too wide for requested granularity	21713	403
Too many services included in requested scope	21714	403
Requested time granularity unavailable	21715	403
Invalidation ID does not exist	21716	404
Unsupported/Unknown API version	21717	404
Internal Server Error (general)	21718	500
Request timestamp too old	21719	403
Credit check failure (Deprecated)	21720	403
API Key is disabled	21721	403
Access Group API privileges suspended	21722	403
API Key request rate too high	21723	503
Could not parse the request header date.	21724	500
Unauthorized access. Invalid id/secret combination.	21725	403

Description	Error Code	Associated HTTP Status
Authorization header not recognized.	21726	403
This operation is not permitted for this key.	21727	403
Invalidation status not available	21728	404
Parameter value is invalid	21729	400
Date range too narrow for requested granularity.	21730	403
Multiple wildcard paths not allowed.	21731	403
Maximum number of paths exceeded.	21732	403
dataInterval specified is not valid.	21733	403
Invalidation Path XML is not valid.	21734	400
Invalidation Path must start with a slash.	21735	400
Access Group ID must be numeric.	21736	400
serviceType must be specified.	21737	400
dateFrom and dateTo must be specified.	21738	400
Requested schema name invalid.	21739	400
Date format invalid.	21740	400
Accept header value not supported.	21741	400
dateFrom comes after dateTo.	21742	400
No data found for specified mediaType. Errors only when there is no data for all measures. If data for at least one measure is returned, then the remaining measures should have empty blocks.	21743	404
dateMonth must be specified (yyyymm)	21744	400
Access Group not set up for Media Portal invalidation.	21745	500
groupBy must be specified.	21746	400
Unknown groupBy value.	21747	400
Collection is not a child of specified NI. Please verify the requested resource.	21748	404
id must be specified.	21749	400
The requested groupBy is not configured for the requested Content Analytics resource.	21750	403
Live streaming content is not supported for invalidation.	21751	400
Network Identifier(s) are not valid for requested Access Group: [NIs]	21752	404
Invalidation path specified is not valid.	21753	400
Mobile login failed.	21754	500
Mobile login failed, invalid username/password.	21755	403
API request rate triggered denial of service warning, request will not be fulfilled.	21756	403
Invalidation failed - network error.	21757	400
Invalid use of property parameter.	21758	400
API Key ID must be numeric.	21759	400

Description	Error Code	Associated HTTP Status
Unsupported query string combination. metricLens and metric query string parameters were sent on the same request.	21760	403
mediaType query string parameter is required. If the /clientSideStats service is called, the mediaType query string parameter must be provided.	21761	404
Invalid metric value for purchased services. The customer's Client Side Stats contract does not include the requested metric.	21762	403
Date value is required. Data for a specific metric has been requested (query string, metric) but the user did not provide the date query string. Only valid for historical data.	21763	400
Access Group does not have Client Side Stats permission.	21764	403

Appendix: Usage Granularity Rules

Caching

Date Range	Time Interval
1 day	5 min
2 - 20 days	1 hour
21 - 90 days	1 day
91+ days	1 month

Streaming

Date Range	Time Interval
1 - 20 days	1 hour
21 - 90 days	1 day
91+ days	1 month

Origin Storage

Date Range	Time Interval
1 - 90 days	1 day
91+ days	1 month

The tables above list the allowed values when requesting usage data with a specified data interval. For example:

Caching date range: 1/1/2018 – 2/4/2018, time intervals:

- **5 min**—not allowed, scope too large (because date range is more than 1 day)
- **Hourly**—not allowed, scope too large (because date range is more than 20 days)
- **Daily**—allowed.
- **Monthly**—allowed. Service will return all of January and all of February because the request spans both those months.

Note: The month to date is considered a full month.

Likewise, ranges that are too narrow for their interval will be considered as well. For example:

Caching date range: 1/1/2018 to 1/2/2018, time intervals:

- **5 min**—allowed.
- **Hourly**—allowed.
- **Daily**—allowed.
- **Monthly**—not allowed, date range too narrow for time interval (fewer than 28 days).

Appendix: Vyvx State and Ingress State Response Definitions

Level 3 provides live reservation state (health) information for video reservations through the Vyvx API system. Level 3 defines two types of reservation state:

- (overall) state
- ingress state

The 'state' refers to the overall health of the service from end-to-end, while the 'ingress state' represents an informed estimation of the health of the signal being presented to the ingress demarcation point of the network (i.e. the quality of the signal being provided to Level 3 for transport).

Possible State Values

The state and ingress state fields use a defined terminology and are always one of the following values:

- online—service is operating normally
- impaired—service is operating normally, with some non-service-impacting anomaly
- degraded—service is still operational, but is experiencing some service-impacting anomaly
- offline—service is not operational
- unknown—the state cannot be determined

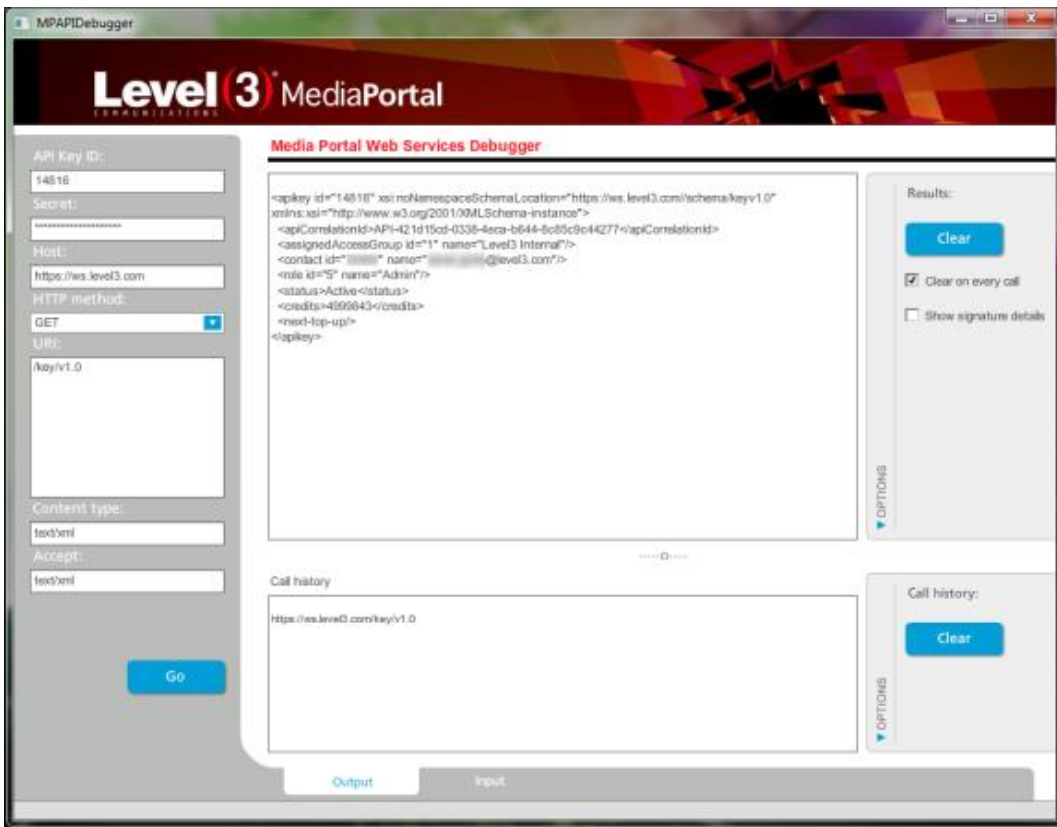
Ingress State Calculation

Level 3 provides the ingress state as an informed estimation of the health of the signal being provided by the source. The ingress state is calculated by examining the health of the signal being presented to the first piece of equipment in the signal flow chain that can be actively monitored. Because Level 3 does not maintain active monitoring equipment at all customer demarcations, the first monitored equipment may be at a location other than the demarcation point. This means that the ingress state presented by Level 3 systems is an informed estimation of the health of the signal being provided, not an authoritative declaration.

Should there be no active monitoring equipment in the signal flow chain prior to long-haul transport, Level 3 will determine the ingress state to be “unknown” should a service impairment be detected.

Using Media Portal's API Debugger

This tool helps you test whether a request works before building an application around the request. The API Debugger tool lets you test the security key, whether the host is active, and the request's URI formatting and input parameters, among other things. The tool sends a single request to the servers the same as an automated system might, so you can test your requests and see the results in a single interface.



Installing the Application

The application's installation file is found in the "API Sample Code" on page 129. If you agree to the terms, download the Sample Code file, extract the Debugger application, and install it on your own computer.

Building a Request

With the application open, complete these fields:

API Key ID:	<input type="text" value="14816"/>
Secret:	<input type="text" value="*****"/>
Host:	<input type="text" value="https://ws.level3.com"/>
HTTP method:	<input type="text" value="GET"/>
URI:	<input type="text" value="/key/v1.0"/>
Content type:	<input type="text" value="text/xml"/>
Accept:	<input type="text" value="text/xml"/>

1. Enter the API Key ID and Secret.
Generate the information for these two fields in Media Portal using the steps in "API Security Keys" on page 12.
2. Enter the Host name.
This is where the request is sent. In general, it is: https://ws.level3.com
3. Select the HTTP method.
Most of the Media Web Services requests use the GET method, but can also use POST.
4. Enter the URI that forms your request.
The Host and URI field contents are joined to form the complete request. Depending on the request, the format of the URI will vary greatly. For more information on building your request, see the topics under "API Specification" on page 21
Use the Input tab at the bottom of the application window to enter additional data for a POST or PUT request. Some illustrations of these are in Invalidations: "Example 1 Body of POST" on page 59
5. In the Content type and Accept fields, enter text/xml.
6. If this request uses a HTTP method of POST, click the *Input* tab and enter the request body data.
7. Click **Go**.

Using Output or Input Tabs

The Output tab at the bottom of the application window is used to display results of the requests. The Input tab displays the body of the request.

API Sample Code

This page provides access to sample code for use in building your own API requests.

The sample code package includes these files:

- ActionScript 3.0 class allowing connection to Media Portal APIs - this is not a full application, only a sample class implementation.
- C# sample class allowing connection to Media Portal APIs.
- Java class allowing connection to Media Portal APIs - this is not a full application, only a sample class implementation.
- PHP file that allows the APIs to be queried via a simple web-page.
- Python sample class allowing connection to Media Portal APIs.
- Adobe Air application that you can install to build and test your API requests.

Note: If you are reading this document in PDF format or as a printed document, ask Level 3 Support or MediaPortalFeedback@Level3.com for the sample code attachments.

Terms of Use

These terms of use must be accepted before downloading the sample code file.

By accepting the attached files (“THE SOFTWARE”) you and your company (collectively, “you”) agree to the following terms as a condition of gaining access to the Level 3 Portal. If you are a customer of Level 3 and you have a separate Portal Access Agreement or Service Schedule executed with Level 3, that separate agreement will control to the extent it may be inconsistent with the terms below, and the term “Customer” shall be defined in accordance with such separate Portal Access Agreement or Service Schedule.

1. **Authorized User and Binding Effect.** Only authorized users may access the Portal. By accessing the Portal, you represent you are authorized by Customer to view the information available and/or take those actions you submit via this Portal, all of which actions are binding upon Customer. Level 3 may deny Portal access, in its sole and absolute discretion, at any time for any or no reason.
2. **Portal Features and Functionality.** The Portal is a web-based application that provides visibility into and control over certain aspects of you or Customer’s Service relationship with Level 3 via a graphical user interface. Level 3 provides access to certain features or functionality of the Portal free of charge (unless otherwise set forth in a Customer Order), which may include placing Customer Orders for select Services; opening and monitoring trouble tickets; obtaining billing and usage information; and/or paying invoices. If and to the extent that Level 3 provides service performance or other network monitoring information, such information is indicative only and shall not necessarily be used to determine the applicability of service credits or other contract remedies. Level 3 may change the features, functionality of and/or the information available through the Portal in its sole and absolute discretion, including discontinuing any functionality or the Portal completely.
3. **Information Obtained Via the Portal.** You agree that any and all information accessed via use of the Portal is and will remain confidential. Such information may be used only in regard to your Level 3 provided Services and you may not disclose such information to third parties except as set forth in a mutually executed Non-Disclosure Agreement.
4. **Portal Use and Security.** You shall not input any data into the Portal that is false, misleading, threatening, indecent, libelous, defamatory, unlawful or that violates or infringes any trademark, copyright or similar rights of others. You shall indemnify, defend and hold Level 3 harmless from any claims arising from your use of or access to the Portal. You are responsible for ensuring the security of all passwords, user names, and other specific access information. You are responsible for all access and use of the Portal, whether authorized or not, to the Portal using your security information. You are responsible for updating Level 3 to ensure only approved users have appropriate levels of access. You will notify Level 3 immediately of any suspected or actual breach of security on the Portal or compromise of access related or other Portal information.
5. **Special Terms for Delegated Administrators.** If you have been designated as the Delegated Administrator, you are responsible for setting up and keeping current all security and administration of Customer’s use of the Portal, including but not limited to: (i) assigning each user a separate ID for entry into the Portal; (ii) assigning levels of permission for each user to assure that users have access only to those aspects of the Portal if such user has authority to act for Customer; (iii) ensuring users who should no longer have access are denied access to the Portal; and (iv) ensuring that any end user access you provide to the Portal is done only with Level 3’s consent, on a read only basis, and subject to confidentiality and other terms no less stringent than those set forth herein.
6. **Limitation of Liability/No Warranty.** LEVEL 3 SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, SPECIAL, ACTUAL, INCIDENTAL, PUNITIVE OR ANY OTHER DAMAGES, OR FOR ANY LOST PROFITS OF ANY KIND OR NATURE WHATSOEVER, EVEN IF LEVEL 3 HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE OR LOSS, ARISING OR RESULTING OR FROM

OR IN ANY WAY RELATING TO CUSTOMER'S USE OF OR ATTEMPT TO USE ANY LEVEL 3 PORTAL OR THE USE, REPRODUCTION, MODIFICATION AND/OR DISTRIBUTION OF THE SOFTWARE OR THE ATTEMPT OF ANY OF THE FOREGOING, HOWEVER CAUSED AND WHETHER UNDER THEORY OF CONTRACT, TORT (INCLUDING NEGLIGENCE), STRICT LIABILITY OR OTHERWISE. THE SOFTWARE AND PORTALS ARE PROVIDED "AS-IS" AND LEVEL 3 MAKES NO REPRESENTATIONS OR WARRANTIES WHATSOEVER REGARDING THEIR USE AND MAKES NO WARRANTY THAT THE SOFTWARE AND PORTALS OR USE THEREOF WILL BE UNINTERRUPTED, ERROR-FREE OR VIRUS-FREE. LEVEL 3 HEREBY DISCLAIMS ALL WARRANTIES INCLUDING ANY IMPLIED REPRESENTATIONS OR WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR WARRANTIES OF NON-INFRINGEMENT.