## Contents

**Media Web Services (API)** ......................................................... 1

**Getting Started with Media Portal APIs** ............................................................. 3

- Using APIs ........................................................................................................ ......................................................... 3
  - Determining the Access Group ID ........................................................................ 3
  - API Interfaces ........................................................................................................ 4
- Host ................................................................................................................................. 4
- Interface Examples ........................................................................................................ 4
- API Call Details ........................................................................................................ 5
  - API Security ............................................................................................................ 5
- Signing your API Requests ......................................................................................... 5

**API Security Keys** ........................................................................................................ 6

**Understanding the API Security Framework** ............................................................... 9

**API Specification** ........................................................................................................... 13

- HTTP Delivery ........................................................................................................... 14
- Retrieve All Properties ............................................................................................... 16
- Create New Property .................................................................................................. 17
- Retrieve A Specified Property ................................................................................... 19
- Deactivate (Delete) A Specified Property .................................................................. 20
- Reactivate A Specified Property ................................................................................. 21
- Retrieve Origin Definition ......................................................................................... 22
  - Update Origin Definition ......................................................................................... 23
- Retrieve Aliases ........................................................................................................... 24
- Add Alias ...................................................................................................................... 25
- Remove Aliases ........................................................................................................... 26
- Resource Groups ......................................................................................................... 27
- Create Resource Group ............................................................................................. 28
- Retrieve Individual Resource Group ........................................................................... 29
- Retrieve Resource Groups ......................................................................................... 30
- Update Resource Group ............................................................................................. 31
- Update Resource Group Order .................................................................................... 32
- Delete Resource Group ............................................................................................... 33
- Configuration Groups ................................................................................................. 34
- Create Geo Definition ................................................................................................. 35
- Retrieve Geo Definition ............................................................................................... 36
- Retrieve Specified Geo Definition .............................................................................. 37
- Update Geo Definition ................................................................................................. 38
Delete Geo Definition ........................................................................................................... 39
Token Definition .................................................................................................................. 40
Create Token Definition ...................................................................................................... 41
Retrieve Token Definitions ................................................................................................... 42
Retrieve Individual Token Definition ................................................................................... 43
Delete Token Definition ...................................................................................................... 44
Access Control Overview ..................................................................................................... 45
Retrieve Access Control Configuration Group .................................................................... 46
Create Access Control Configuration Group ....................................................................... 47
Update Access Control Configuration .................................................................................. 48
Retrieve Cache Control (CCHO) Configuration Groups ....................................................... 49
Cache Control Overview ...................................................................................................... 50
Create Cache Control (CCHO) Configuration Group .......................................................... 51
Update Cache Control (CCHO) Configuration Group ......................................................... 52
Content Manipulation .......................................................................................................... 53
Retrieve Content Manipulation (QSHMode) Configuration Group .................................... 54
Create Content Manipulation (QSHMode) Configuration Group ....................................... 55
Update Content Manipulation (QSHMode) Configuration Group ...................................... 56
Retrieve All Configuration Groups ....................................................................................... 57
Key (Deprecated) ................................................................................................................... 58
Mobile Login ........................................................................................................................ 59
Access Group Hierarchy ........................................................................................................ 60
Services Hierarchy (Partially Deprecated) ........................................................................... 61
Invalidations: Request Invalidation (POST) ......................................................................... 65
Invalidations: Request Origin Invalidation (POST) .............................................................. 69
Invalidations: Check status (GET) ....................................................................................... 71
RTM (Caching) ....................................................................................................................... 75
Usage Reporting ..................................................................................................................... 79
Usage Reporting - Data Interval ............................................................................................ 84
Usage By Access Group ......................................................................................................... 88
Content Analytics Summary & Trend .................................................................................... 90
Origin Storage Overview ....................................................................................................... 96
Create New Virtual Host ....................................................................................................... 97
Create New User Account .................................................................................................... 98
Update User Account ........................................................................................................... 99
Delete User Account ............................................................................................................ 100
Services Hierarchy (Partially Deprecated) .......................................................................... 101
# Usage Reporting

- Data Interval ................................................................. 105

# Usage By Access Group

- ................................................................................................................. 109

# Content Analytics Summary & Trend

- ................................................................................................................. 111

# Get Configuration Information

- ................................................................................................................. 117

# Order Status

- ................................................................................................................. 119

# Appendix: Error Responses

- ................................................................................................................. 120

# Appendix: Usage Granularity Rules

- ................................................................................................................. 122

# Access Group Hierarchy

- ................................................................................................................. 123

# Delete Configuration Group

- ................................................................................................................. 124

# Delete Existing Virtual Host

- ................................................................................................................. 125

# Vyvx Reservation - Get Active Status

- ................................................................................................................. 126

# Vyvx Reservation - Get Service Status

- ................................................................................................................. 127

# Vyvx Reservation - Get Service History

- ................................................................................................................. 128

# Appendix: Vyvx State and Ingress State Response Definitions

- ................................................................................................................. 129
  - Possible State Values ......................................................................................... 129
  - Ingress State Calculation ..................................................................................... 129

# Using the Media Portal API Debugger

- ................................................................................................................. 130

# API Sample Code

- ................................................................................................................. 132

# Terms of Use

- ................................................................................................................. 132

# Index

- ................................................................................................................. 134
Media Web Services (API)

Overview
The 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 the Media Web Services made available to customers through the 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?" on page 9

- **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 "Adding a New Key" on page 6

- **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" on page 11.

- **How do I construct an API request?**
  Each API request must conform to an interface specification. See "API Specification" on page 13 and find the interface you want to use. This page also has an appendix with several other useful API aspects.

- **Where do I send the API request?**
  Each API interface has a specific URL. See "API Specification" on page 13 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 13 and find the interface you want to use.

The authentication, authorization and API interface style is common across the full product set. For example, if you understand how to use Caching APIs, the Streaming APIs use a similar style, without requiring a different set of credentials to access them.

The Media Portal uses Access Groups to control the scope of services and network identifiers (properties, streaming IDs and vhosts) accessible to any portal user. Access groups security rules govern what a user can do to services/network identifiers if they have access. The same model applies to API keys.

Conceptual Reference
If you would like a more detailed description of the concepts behind the API architecture, see *RESTful Web Services* by Leonard Richardson and Sam Ruby. The API Key pages contain references to this book.

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**

<table>
<thead>
<tr>
<th>API Interface Style</th>
<th>The API architecture is stateless.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Every HTTPS API request must include all of the information required for the server to fulfill that request.</td>
</tr>
<tr>
<td></td>
<td>• Each request is constructed to be complete, unless information is provided back to the server from the client within the new request.</td>
</tr>
<tr>
<td></td>
<td>• 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.</td>
</tr>
</tbody>
</table>

**Definitions**

These terms are used in the API documentation pages:

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>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.</td>
</tr>
<tr>
<td>REST</td>
<td>Representational state transfer (REST) is a style of software architecture for distributed systems such as the World Wide Web.</td>
</tr>
<tr>
<td>RESTful Web Service</td>
<td>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 &quot;RESTful Web Services&quot; 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.</td>
</tr>
<tr>
<td>API Key</td>
<td>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.</td>
</tr>
<tr>
<td>URI</td>
<td>A RESTful, resource-oriented service exposes a URI for every piece of data the client might want to operate on.</td>
</tr>
</tbody>
</table>
Getting Started with Media Portal APIs

Media Portal (http://mediaportal.level3.com) is Level 3’s customer portal for CDN services. Through the 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 the 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 requiring use of the browser-based graphical user interface. Our aim is, over time, to provide an API equivalent for each functional area that is available in Media Portal itself.

New Service Configuration API Services are now available! See related sections through this help system.

Using APIs

The basic steps required to use the 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. For more information, see: Getting Started with Media Portal APIs.

2. Determine the Access Group ID. Since the access group name is editable, the 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: Determining the Access Group ID below.

3. If you are developing an API that includes a specific SCID or Network Identifier as part of the scope, you will also need to determine those values to include in your URI syntax by using the Services Hierarchy API. For more information, see: Services Hierarchy (Partially Deprecated) on page 101.

4. Determine which API interfaces you need to call, and 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 on page 4 or API Specification on page 13.

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” on page 5.

Determining 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, follow this procedure:

1. Send a request using the API Key or Access Group interface. See the API: "Key (Deprecated)" on page 58 or the API: “Access Group Hierarchy” on page 123.

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

API Interfaces

The 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.

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/key</td>
<td>Basic information about the API Key used to make the call</td>
</tr>
<tr>
<td>/accessGroups</td>
<td>The access group hierarchy addressable by the calling API Key</td>
</tr>
<tr>
<td>/services</td>
<td>The service hierarchy addressable by the calling API Key</td>
</tr>
<tr>
<td>/rtm</td>
<td>Real-time monitor data for both caching and streaming services</td>
</tr>
<tr>
<td>/usage</td>
<td>Historical usage data for caching, streaming and storage services</td>
</tr>
<tr>
<td>/invalidations</td>
<td>Support for creating and querying status of caching content invalidation requests</td>
</tr>
</tbody>
</table>
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 13.

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.

Signing your API Requests

**Note:** 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.

In testing your connectivity to the Media Portal APIs, an HTTP debugger may be useful, such as Charles (http://www.charlesproxy.com/). 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

**Audience**
This topic is intended for authorized customers who create API keys for use in an API request.

**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" on page 9
- How a signature is built, see "Request Signature Form" on page 11
- 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:

**Adding a New Key**
An API key is needed to construct an API request.

To add a new key, follow this procedure:
1. Select the **API Security Key** menu item.
2. Select the Access Group that will use the key.
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**.

**Viewing or Editing 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. Be sure to click the "Save" button when finished.
3. If needed, you can click on "Generate New Secret". Once confirmed this will succeed and does not require you to "Save".

**Security Hierarchy**

Depending on a user's role and permissions, the 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.
Creating 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 the 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, follow this procedure:

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.

Disabling or Enabling 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.
Understanding the API Security Framework

Audience
This page is intended for developers who use API keys to create the API signature. This document provides the details to help you understand a key’s components. Use the guidelines from most relevant "API Specification" on page 13 and examples from the API Sample Code to build your request.

What is the API Key?
In the 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 6.

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 the Media Portal. This key is public and does not need to be kept secret.
- The secret is assigned by the 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" on page 8.

- 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.

API Keys are Limited
The number of API keys is limited. Authorized users can generate up to 5 API keys per access group.

- If necessary, a user with the correct role and permissions can disable or re-enable an API key.
- If one user disables a key, another user can re-enable it if he has been assigned the API Key permission at the same level access group or higher.

Key Statuses
An API key can have one of four statuses:

<table>
<thead>
<tr>
<th>Status</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active</td>
<td>Functioning.</td>
</tr>
<tr>
<td>Disabled</td>
<td>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.</td>
</tr>
</tbody>
</table>
If the key has been disabled by an admin in a parent access group, the enable key is not active in child access groups.

<table>
<thead>
<tr>
<th>Suspending</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Suspended</td>
<td>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.</td>
</tr>
</tbody>
</table>

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 keys can be disabled**

If necessary, a single API key can be disabled. All requests using a key with disabled status are rejected. See “Disabling or Enabling an API key” on page 8.

- Disabled API keys can be enabled by the same administrator, by a peer administrator, or by one associated with a parent access group.

**API Keys can be Suspended**

If necessary, all API privileges can be suspended for an access group and all its child access groups. No changes can be made to any API key in the access group during the suspension. See “API Security Keys”.

- Suspended access groups can be resumed by the same administrator, by a peer administrator, or by an administrator associated with a parent access group. Similarly, suspended access groups can be resumed by an administrator with the same or higher-level permissions.

**API Request Acceptance**

Each time the 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.

The Media Portal returns a HTTP status code if the request is rejected for one of these reasons. This table lists the codes:
<table>
<thead>
<tr>
<th>Description</th>
<th>Response Code</th>
<th>Entity Body Returned to Client</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication failure</td>
<td>403</td>
<td>None. No entity body is returned to the caller to limit exposing data to a potentially malicious request.</td>
</tr>
<tr>
<td>Request timestamp is too old</td>
<td>403</td>
<td>mpeRequestTooOld</td>
</tr>
<tr>
<td>API Key is disabled</td>
<td>403</td>
<td>mpeAPIKeyDisabled</td>
</tr>
<tr>
<td>Access Group API privileges suspended</td>
<td>403</td>
<td>mpeAPIPrivilegesSuspended</td>
</tr>
<tr>
<td>API Key request rate too high</td>
<td>503</td>
<td>mpeRequestRateTooHigh</td>
</tr>
</tbody>
</table>

**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:

```text
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:
### Unauthenticated Requests

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

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

The 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 the 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, the 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

The 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 Specification

### Audience
These pages are intended for developers who build applications that need to make API requests.

### What is the 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, the Media Portal provides a response. Follow the steps listed in "Getting Started with Media Portal APIs" on page 3 to create a key.

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

- "Vyvx Reservation - Get Service Status" on page 127
- "Vyvx Reservation - Get Service History" on page 128
HTTP Delivery

Limited Availability

Release Scope

The presentation of this API is broken into different categories corresponding to the major component of a service configuration.

<table>
<thead>
<tr>
<th>Category</th>
<th>Function</th>
<th>Scope</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property Origin Aliases</td>
<td>Retrieve, add, inactivate/reactivate a property</td>
<td>Service or Property</td>
</tr>
<tr>
<td>Origin</td>
<td>Retrieve and update the origin server section of a property</td>
<td>Property</td>
</tr>
<tr>
<td>Aliases</td>
<td>Retrieve, add, and remove aliases for a property</td>
<td>Property</td>
</tr>
<tr>
<td>Geo Definition</td>
<td>Retrieve, create, update, and delete a Geo definition for Geo Blocking</td>
<td>Service or Property</td>
</tr>
<tr>
<td>Token</td>
<td>Retrieve, add, update, and remove Token definitions for Token Authentication</td>
<td>Service or Property</td>
</tr>
<tr>
<td>Resource Groups</td>
<td>Retrieve, add, update, and remove Resource Groups (typically path based URI filters)</td>
<td>Service or Property</td>
</tr>
<tr>
<td>Configuration Groups</td>
<td>Retrieve, add, update, and remove Configuration Groups for Access Control, Cache Control, and Content Manipulation rules associated with a Resource Group filter.</td>
<td>Service or Property</td>
</tr>
</tbody>
</table>

Propagation of Configuration Changes

The Service Configuration services for HTTP Delivery potentially affect the operation of hundreds or thousands of CDN devices globally. To efficiently manage 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:

- **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.

Property

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.

Adding and Editing Properties

Viewing and Updating Origin Definition

Adding and Updating Aliases

Creating and updating Match Pattern / URL Path (Resource Groups)

- "Retrieve All Properties" on page 16
- "Retrieve Aliases" on page 24
- "Add Alias" on page 25
- "Remove Aliases" on page 26
- "Create Resource Group" on page 28
- "Retrieve Resource Groups" on page 30
- "Retrieve Individual Resource Group" on page 29
- "Retrieve Individual Resource Group" on page 29
<table>
<thead>
<tr>
<th>Creating and Updating Configuration Rules (Resource Groups)</th>
</tr>
</thead>
<tbody>
<tr>
<td>• &quot;Update Resource Group Order&quot; on page 32</td>
</tr>
<tr>
<td>• &quot;Update Resource Group&quot; on page 31</td>
</tr>
<tr>
<td>• &quot;Delete Resource Group&quot; on page 33</td>
</tr>
<tr>
<td>• &quot;Retrieve Individual Resource Group&quot; on page 29</td>
</tr>
<tr>
<td>• &quot;Delete Configuration Group&quot; on page 124</td>
</tr>
<tr>
<td>Creating and Updating Geo Definitions</td>
</tr>
<tr>
<td>• &quot;Create Geo Definition&quot; on page 35</td>
</tr>
<tr>
<td>• &quot;Retrieve Geo Definition&quot; on page 36</td>
</tr>
<tr>
<td>• &quot;Retrieve Specified Geo Definition&quot; on page 37</td>
</tr>
<tr>
<td>• &quot;Update Geo Definition&quot; on page 38</td>
</tr>
<tr>
<td>• &quot;Delete Geo Definition&quot; on page 39</td>
</tr>
<tr>
<td>Creating and Updating Token Definitions</td>
</tr>
<tr>
<td>• &quot;Create Token Definition&quot; on page 41</td>
</tr>
<tr>
<td>• &quot;Retrieve Token Definitions&quot; on page 42</td>
</tr>
<tr>
<td>• &quot;Retrieve Individual Token Definition&quot; on page 43</td>
</tr>
<tr>
<td>• &quot;Delete Token Definition&quot; on page 44</td>
</tr>
<tr>
<td>Manage Access for Geo and Token Definitions</td>
</tr>
<tr>
<td>• &quot;Access Control Overview&quot; on page 45</td>
</tr>
<tr>
<td>• &quot;Retrieve Access Control Configuration Group&quot; on page 46</td>
</tr>
<tr>
<td>• &quot;Create Access Control Configuration Group&quot; on page 47</td>
</tr>
<tr>
<td>• &quot;Update Access Control Configuration&quot; on page 48</td>
</tr>
<tr>
<td>Creating and Updating Cache Control Management</td>
</tr>
<tr>
<td>• &quot;Retrieve Cache Control (CCHO) Configuration Groups&quot; on page 49</td>
</tr>
<tr>
<td>• &quot;Create Cache Control (CCHO) Configuration Group&quot; on page 51</td>
</tr>
<tr>
<td>• &quot;Update Cache Control (CCHO) Configuration Group&quot; on page 52</td>
</tr>
<tr>
<td>Creating and Updating Query String Handling Mode (Content Manipulation)</td>
</tr>
<tr>
<td>• &quot;Content Manipulation&quot; on page 53</td>
</tr>
<tr>
<td>• &quot;Retrieve Content Manipulation (QSHMode) Configuration Group&quot; on page 54</td>
</tr>
<tr>
<td>• &quot;Create Content Manipulation (QSHMode) Configuration Group&quot; on page 55</td>
</tr>
<tr>
<td>• &quot;Update Content Manipulation (QSHMode) Configuration Group&quot; on page 56</td>
</tr>
</tbody>
</table>
Retrieve All 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
{
  "accessGroup": {
    "id": 1234,
    "name": "Customer Corporation",
    "apiCorrelationId": "API-XXXXXXXXXXXXXXXXX",
    "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"
          }
        ]
      }
    ]
  }
}

Possible Status and Error Messages

- 200: Request Successful.
- See Appendix A – Status Codes & Error Messages for additional return codes and messages.
Create New Property

Base URI

https://ws.level3.com

Method

POST

Description

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.

**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 the Media Portal for usage reporting, invalidation, and trouble ticketing. This primary alias cannot be deleted or changed.

**Note:** There is a delay of approximately 15 minutes after a new property is created until the property is available for further configuration. The property is created immediately. However, no additional property configuration may be retrieved or set during that initial 15 minutes. After this time, all other operations will take effect immediately.

**URI Syntax**

/serviceConfiguration/(version)/(scope)

version [CDATA[ ]]

Values: "v1.0" (required)

scope [CDATA[ ]]

(AG)/(SCID)

AG = Access Group

SCID = Service Component IDentifier

**Body Syntax**

```
{
  "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
    ...
  ]
}
```

**Sample Request**

```
URI: https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678

Body:
{
  "originserver" : {
    "host" : "test.origin.com",
    "port" : "8001",
    "protocol" : "http"
  },
  "aliases" : [
    "test1.caching.cdn.level3.net",
    "test2.caching.cdn.level3.net"
  ]
}
```
Sample Response

```json
{
  "originserver": {
    "host": "test.origin.com",
    "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 Specified Property

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

### Base URI

https://ws.level3.com

### Method

DELETE

### Description

Removing a property is done via 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)

### URI Sample

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 Specified Property

<table>
<thead>
<tr>
<th><strong>Base URI</strong></th>
<th><a href="https://ws.level3.com">https://ws.level3.com</a></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Method</strong></td>
<td>PUT</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>Reinstate a deactivated property. Status changes to &quot;active&quot;.</td>
</tr>
<tr>
<td><strong>URI Syntax</strong></td>
<td>/serviceConfiguration/(version)/(scope)</td>
</tr>
</tbody>
</table>

- **version** [CDATA[ ]]  
  Values: "v1.0" (required)
- **scope** [CDATA[ ]]  
  (AG)/(SCID)/(ALIAS)

| **URI Sample** | 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.
## Retrieve Origin Definition

<table>
<thead>
<tr>
<th>Base URI</th>
<th><a href="https://ws.level3.com">https://ws.level3.com</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>GET</td>
</tr>
<tr>
<td>Description</td>
<td>Retrieves the origin server definition for the property.</td>
</tr>
<tr>
<td>Schema Location</td>
<td><a href="https://ws.level3.com/schema/serviceConfiguration/v1.0">https://ws.level3.com/schema/serviceConfiguration/v1.0</a></td>
</tr>
<tr>
<td>URI Syntax</td>
<td><code>/serviceConfiguration/(version)/(scope)/Origin</code></td>
</tr>
<tr>
<td>version [CDATA[ ]]</td>
<td>Values: &quot;v1.0&quot; (required)</td>
</tr>
<tr>
<td>scope [CDATA[ ]]</td>
<td>(AG)/(SCID)</td>
</tr>
<tr>
<td>AG = Access Group</td>
<td>SCID = Service Component IDentifier</td>
</tr>
<tr>
<td>Sample Request</td>
<td>URI: <a href="https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/test1.caching.cdn.level3.net/Origin">https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/test1.caching.cdn.level3.net/Origin</a></td>
</tr>
</tbody>
</table>
| Sample Response   | `{  
|                   | "webroot": "/",  
|                   | "port": "8001",  
|                   | "protocol": "http",  
|                   | "host": "test.origin.com"  
|                   | }` |
| 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

<table>
<thead>
<tr>
<th>Base URI</th>
<th><a href="https://ws.level3.com">https://ws.level3.com</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>PUT</td>
</tr>
<tr>
<td>Description</td>
<td>Update the origin server definition for the property. Only the property that requires changing needs to be sent.</td>
</tr>
<tr>
<td>Schema Location</td>
<td><a href="https://ws.level3.com/schema/serviceConfiguration/v1.0">https://ws.level3.com/schema/serviceConfiguration/v1.0</a></td>
</tr>
<tr>
<td>URI Syntax</td>
<td>/serviceConfiguration/(version)/(scope)/Origin</td>
</tr>
<tr>
<td></td>
<td>version [CDATA[ ]] Values: &quot;v1.0&quot; (required)</td>
</tr>
<tr>
<td></td>
<td>scope [CDATA[ ]]      (AG)/(SCID)</td>
</tr>
<tr>
<td></td>
<td>AG = Access Group</td>
</tr>
<tr>
<td></td>
<td>SCID = Service Component IDentifier</td>
</tr>
<tr>
<td>Body Syntax</td>
<td>{</td>
</tr>
<tr>
<td></td>
<td>&quot;originserver&quot; : {</td>
</tr>
<tr>
<td></td>
<td>&quot;host&quot; : &quot;&lt;FQDN of the origin server&gt;&quot;, # required</td>
</tr>
<tr>
<td></td>
<td>&quot;port&quot; : &quot;&lt;port number of origin server&gt;&quot; , # required</td>
</tr>
<tr>
<td></td>
<td>&quot;protocol&quot; : (&quot;http&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;webroot&quot; : &quot;&lt;root path&gt;&quot; # optional, default: &quot;/&quot;</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
</tbody>
</table>

### Sample Request

URI: https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/test1.caching.cdn.level3.net/Origin |

|                         | { |
|                         | "port" : "8007" |
|                         | } |

### 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.
## 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
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>version</td>
<td>[CDATA[]]</td>
<td>Values: &quot;v1.0&quot; (required)</td>
</tr>
<tr>
<td>scope</td>
<td>[CDATA[]]</td>
<td>(AG)/(SCID)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AG = Access Group</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SCID = Service Component IDentifier</td>
</tr>
</tbody>
</table>

**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
Adds a new alias to the array of aliases for the property.

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
/serviceConfiguration/(version)/(scope)/Aliases/(ALIAS)

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/mynew.alias.caching.cdn.level3.net

Sample Response
[
  "mynew.alias.caching.cdn.level3.net"
]

Possible Status and Error Messages
201: Created

See Appendix A – Status Codes & Error Messages for additional return codes and messages.
Remove Aliases

Base URI
https://ws.level3.com

Method
DELETE

Description
Removes an alias from the array of aliases for the property.

*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
/serviceConfiguration/(version)/(scope)/Aliases/(ALIAS)

- **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/mynew.alias.caching.cdn.level3.net

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.
Resource Groups

Overview

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

```
{  
  "rgid": "<string with characters [A-Za-z0-9_]>", # required  
  "rgtype": "path" # required  
  "rgdef": [ "<identifier>", … ], # required  
}
```

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:
```
{
  "rgdef": [ ".flv", ".mp4", ".f4p" ],
  "rgtype": "path",
  "rgid": "videos"
}
```

Sample Response  
```
{
  "rgdef": [ ".flv", ".mp4", ".f4p" ],
  "rgtype": "path",
  "rgid": "videos"
}
```

Possible Status and Error Messages Returned to Client

201: Created.

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)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>version [CDATA[ ]]</td>
<td>Values: “v1.0” (required)</td>
</tr>
<tr>
<td>scope [CDATA[ ]]</td>
<td>(AG)/(SCID)[/(ALIAS)]</td>
</tr>
<tr>
<td>AG = Access Group</td>
<td></td>
</tr>
<tr>
<td>SCID = Service Component Identifier</td>
<td></td>
</tr>
<tr>
<td>ALIAS = Optional Alias identifies Property</td>
<td></td>
</tr>
</tbody>
</table>

**Sample Request**  
URL: 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**  
```json
{
  "rgdef": [ "*.flv", "*.mp4", "*.f4p" ],
  "order": 0,
  "rgtype": "path",
  "rgid": "videos"
}
```

**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 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
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Values</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>version</td>
<td>[CDATA[]]</td>
<td>v1.0</td>
<td>yes</td>
</tr>
<tr>
<td>scope</td>
<td>[CDATA[]]</td>
<td>(AG)/(SCID)/[/(ALIAS)]</td>
<td>AG = Access Group, SCID = Service Component Identifier, ALIAS = Optional Alias identifies Property</td>
</tr>
</tbody>
</table>

**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**

```
[
  {
    "rgdef": [ ".flv", ".mp4", ".f4p" ],
    "order": 0,
    "rgtype": "path",
    "rgid": "videos"
  },
  {
    "rgdef": [ ".jpg", ".png" ],
    "order": 1,
    "rgtype": "path",
    "rgid": "images"
  }
]
```

**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 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  
{
  "rgid": "<string with characters [A-Za-z0-9_.]>", # required
  "rgtype": "path" # optional
  "rgdef": [ "<identifier>“, … ], # required
}

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.
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

[
  {
    "rgid": "<string with characters [A-Za-z0-9_.]>", # required
    "order": <integer> # required
  }, ...
]

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:
[  
  { "rgid": "videos", "order": 0 },
  { "rgid": "images", "order": 1 }
]

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.
Configuration Groups

Overview

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.
## 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.

**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.

**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**
```
{
    "geoid" : "<String containing [A-Za-z0-9_]>", # required
    "cc" : [ "<country code>", ... ] # required
}
```

**Sample Response**
```
{
    "geoid": "NorthAmerica",
    "cc": [ "CA", "US" ]
}
```

**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 Geo Definition

Base URI

https://ws.level3.com

Method

GET

Description

Retrieves the array of geo definitions for service component or property.

**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.

**URI Syntax**

/serviceConfiguration/(version)/(scope)/GeoDefs

version [CDATA[ ]]

Values: "v1.0" (required)

scope [CDATA[ ]]

(AG)/(SCID)

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/GeoDefs

https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/GeoDefs

Sample Response

```json
[
  {
    "cc": "GE, FR",
    "geoid": "Europe"
  },
  {
    "cc": "CA, US",
    "geoid": "NorthAmerica"
  }
]
```

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 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.

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.

URI Syntax
/serviceConfiguration/(version)/(scope)/GeoDefs/(geoid)

- `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/GeoDefs/NorthAmerica

Sample Response
{
  "cc": "CA, US",
  "geoid": "NorthAmerica",
  "order": 0
}

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 Geo Definition

**Base URI**

https://ws.level3.com

**Method**

PUT

**Description**

Update the country codes for an existing Geo Definition.

**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.

**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

**Body Syntax**

```json
{
  "geoid": "<String containing [A-Za-z0-9_]>", # required
  "cc": [ "<country code>", ... ] # required
}
```

**Sample Request**

URI: https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/test1.caching.cdn.level3.net/GeoDefs/NorthAmerica

Body:

```json
{
  "geoid": "NorthAmerica",
  "cc": [ "CA", "US", "MX" ]
}
```

**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 Geo Definition

<table>
<thead>
<tr>
<th><strong>Base URI</strong></th>
<th><a href="https://ws.level3.com">https://ws.level3.com</a></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Method</strong></td>
<td>DELETE</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>Removes an existing Geo Definition identified by geoid.</td>
</tr>
</tbody>
</table>

**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.

**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**

URI: https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/test1.caching.cdn.level3.net/GeoDefs/NorthAmerica

https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/GeoDefs/NorthAmerica

**Sample Response**

EMPTYY

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

- 200: Request successful.

See Appendix A – Status Codes & Error Messages for additional return codes and messages.
Token Definition

Overview

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 the 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**

```
{
  "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
}
```

**Sample Request**

URI: https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/test1.caching.cdn.level3.net/Tokens

Body:

```
{
  "tokenId": "1",
  "secret": "secret1",
  "start": "1415252687",
  "end": "1415339087"
}
```

**Sample Response**

```
{
  "tokenId": "1",
  "secret": "secret1",
  "start": "1415252687",
  "end": "1415339087"
}
```

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

201: Created.

See Appendix A – Status Codes & Error Messages for additional return codes and messages.
# Retrieve 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

**Sample Request**

Sample Request:

- URI: https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/test1.caching.cdn.level3.net/Tokens
- URI: https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/Tokens

**Sample Response**

```
{
   "tokenid": "0",
   "secret": "secret0",
   "start": "1415251839",
   "end": "1415338239"
},
{
   "tokenid": "1",
   "secret": "secret1",
   "start": "1415252687",
   "end": "1415339087"
}
```

**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 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**  
```
{
  "tokenid": "1",
  "secret": "secret1",
  "start": "1415252687",
  "end": "1415339087"
}
```

**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 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.
Access Control Overview

Overview

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**
/serviceConfiguration/(version)/(scope)/ResourceGroups/(rgid)/ConfGroups/AccessControl

- **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**


**Sample Response**

```json
{
  "AccessControl": {
    "tokenauth": {
      "action": "fail"
    },
    "geoblocking": {
      "geoid": ["NorthAmerica", "Asia"],
      "type": "deny"
    }
  }
}
```

**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 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**

```
{
  "AccessControl":
  (
    "geoblocking":
    {# optional "geoid": [" <geoid>", … ] # required
    "action": ("deny" | "allow") # required,
    "tokenauth":
    { # optional "action": "fail" # required
    }
  }
}
```

**URI Syntax**

```
/serviceConfiguration/(version)/(scope)/ResourceGroups/(rgid)/ConfGroups/AccessControl
```

- **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/AccessControl

Body:

```
{
  "AccessControl": {  
    "geoblocking": {  
    "geoid": [ "NorthAmerica" ],  
    "type": "allow"
    }
  }
}
```

**Sample Response**

```
{
  "AccessControl": {  
    "geoblocking": {  
    "geoid": [ "NorthAmerica" ],  
    "type": "allow"
    }
  }
}
```

**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**

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.

**URI Syntax**

/serviceConfiguration/(version)/(scope)/ResourceGroups/(rgid)/ConfGroups/AccessControl

- **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**

```json
{
  "AccessControl": {
    "geoblocking": { # optional
      "geoid": [ "<geoid>", ... ] # required
      "action": ("deny" | "allow") # required
    },
    "tokenauth": { # optional
      "type": "fail" # required
    }
  }
}
```

**Sample Request**


Body:

```json
{
  "AccessControl": {
    "geoblocking": {
      "geoid": [ "NorthAmerica", "Asia" ],
      "type": "allow"
    },
    "tokenauth": {
      "action": "fail"
    }
  }
}
```

**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.
# 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.

<table>
<thead>
<tr>
<th>URI Syntax</th>
<th>/serviceConfiguration/(version)/(scope)/ResourceGroups/(rgid)/ConfGroups/CacheControl</th>
</tr>
</thead>
</table>

- 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:

**Sample Response**

```json
{
  "CacheControl": {
    "cchomode": {
      "ext": "no-store",
      "force": "yes",
      "int": "1234"
    }
  }
}
```

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

- 200: Request Successful.

See Appendix A – Status Codes & Error Messages for additional return codes and messages.
Cache Control Overview

Overview

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 with a 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.
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**

```json
{ "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**

```json
{ "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.
```

**Sample Response**

```json
{ "CacheControl": { "cchomode": { "ext": "no-store", "force": "yes", "int": "120m" } } }
```

**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**
```
{
  "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)

**Sample Request**
```

Body:
```
{
  "CacheControl": {
    "cchomode": {
      "ext": "no-cache",
      "force": "yes",
      "int": "120m"
    }
  }
}
```

**Sample Response**
```
{
  "CacheControl": {
    "cchomode": {
      "ext": "no-store",
      "force": "yes",
      "int": "120m"
    }
  }
}
```

**Possible Status and Error Messages**

- **204**: Success no Content.

  See Appendix A – Status Codes & Error Messages for additional return codes and messages.
Content Manipulation

Overview

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


Sample Response
{
  "ContentManipulation": {
    "qshmode": {
      "action": "ignore",
      "type": "string"
    }
  }
}

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
{
    "ContentManipulation": {
        "qshmode": {
            "type": "string", # required
            "action": ("honor" | "ignore") # required
        }
    }
}

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
Body:
{
    "ContentManipulation": {
        "qshmode": {
            "action": "ignore",
            "type": "string"
        }
    }
}

Sample Response
{
    "ContentManipulation": {
        "qshmode": {
            "action": "ignore",
            "type": "string"
        }
    }
}

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

```
/serviceConfiguration/(version)/(scope)/ResourceGroups/(rgid)/ConfGroups/ContentManipulation
```

<table>
<thead>
<tr>
<th>Version</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>[CDATA[ ]]</td>
<td>&quot;v1.0&quot; (required)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scope</th>
<th>AG/(SCID)[(ALIAS)]</th>
</tr>
</thead>
<tbody>
<tr>
<td>AG = Access Group</td>
<td></td>
</tr>
<tr>
<td>SCID = Service Component Identifier</td>
<td></td>
</tr>
<tr>
<td>ALIAS = Optional Alias identifies Property</td>
<td></td>
</tr>
</tbody>
</table>

**Body Syntax**

```
{  
  "ContentManipulation": {  
    "qshmode": {  
      "type": "string", # required  
      "action": ("honor" | "ignore") # required  
    }  
  }  
}
```

**Sample Request**

URI: https://ws.level3.com/serviceConfiguration/v1.0/1234/BBBN5678/test1.caching.cdn.level3.net/ResourceGroups/videos/ConfGroups/ContentManipulation

Body:

```
{  
  "ContentManipulation": {  
    "qshmode": {  
      "action": "honor",  
      "type": "string"  
    }  
  }  
}
```

**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.
### Retrieve All Configuration Groups

<table>
<thead>
<tr>
<th>Base URI</th>
<th><a href="https://ws.level3.com">https://ws.level3.com</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>GET</td>
</tr>
<tr>
<td>Description</td>
<td>Retrieves the array of Configuration Groups for a given Resource Group of a service component or property.</td>
</tr>
</tbody>
</table>

#### 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

Sample Request

```
```

```
```

#### Sample Response

```
{  
  "ContentManipulation": {  
    "qshmode": {  
      "action": "ignore",  
      "type": "string"  
    },  
  },  
  "CacheControl": {  
    "cchomode": {  
      "ext": "no-store",  
      "force": "yes",  
      "int": "1234"  
    },  
  },  
  "AccessControl": {  
    "geoblocking": {  
      "geoid": [ "NorthAmerica",  
      "type": "allow"  
    }  
  }  
}
```

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

- **200: Request Successful.**

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

```xml
<?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>
```

**Possible Status and Error Messages Returned to Client**
See "Appendix: Error Responses" on page 120.

**Cost Per Call**
See Appendix: Cost per Call.
Mobile Login

Base URI  

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

```xml
<user>
  <username>{media portal username - email address}</username>
  <password>{media portal password}</password>
</user>
```

Example 1 Response

```xml
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <apiCorrelationId>CDNPortal-1300303970131-6830</apiCorrelationId>
  <userApiKey>54321</userApiKey>
  <userApiSecret>QYsM3YqMeYxXbMTYDNr</userApiSecret>
  <contact id="2121" name=""/>
  <status>Active</status>
</apikey>
```

Possible Status and Error Messages

See "Appendix: Error Responses" on page 120.

Cost Per Call

See Appendix: Cost per Call.
## 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**  
```xml
<?xml version="1.0" encoding="UTF-8"?>
<accessGroup id="12345" name="My Access Group"
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>
```

**Possible Status and Error Messages Returned to Client**  
See “Appendix: Error Responses” on page 120.

**Cost Per Call**  
See Appendix: Cost per Call.
<table>
<thead>
<tr>
<th><strong>Services Hierarchy (Partially Deprecated)</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Base URI</strong></td>
</tr>
<tr>
<td><strong>Method</strong></td>
</tr>
<tr>
<td><strong>Description</strong></td>
</tr>
<tr>
<td><strong>Note:</strong> This service functions normally, but will not be enhanced or updated</td>
</tr>
<tr>
<td><strong>Typical Use</strong></td>
</tr>
<tr>
<td><strong>Schema Location</strong></td>
</tr>
</tbody>
</table>
| **URI Syntax** | /services/cdn/(version)/(scope)
| **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**
| or | ID. **NI** is network identifier name. |
| or | 
| or | 
| ?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. |

<table>
<thead>
<tr>
<th><strong>Sample 1 Request</strong></th>
<th><a href="https://ws.level3.com/services/cdn/v1.0/12345">https://ws.level3.com/services/cdn/v1.0/12345</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns services (and individual NIs) in AG 12345</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Sample 1 Response: Services under AG</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;xml version=&quot;1.0&quot; encoding=&quot;UTF-8&quot;?&gt;</td>
</tr>
<tr>
<td>&lt;accessGroup id=&quot;12345&quot; name=&quot;My Access Group&quot;</td>
</tr>
<tr>
<td>xmlns:xsi=&quot;<a href="http://www.w3.org/2001/XMLSchema-instance">http://www.w3.org/2001/XMLSchema-instance</a>&quot; &gt;</td>
</tr>
<tr>
<td>&lt;apiCorrelationId&gt;CDNPortal-1270071184562-8650</td>
</tr>
<tr>
<td>&lt;/apiCorrelationId&gt;</td>
</tr>
<tr>
<td>&lt;serviceResource&gt;/12345&lt;/serviceResource&gt;</td>
</tr>
<tr>
<td>&lt;services&gt;</td>
</tr>
<tr>
<td>&lt;service id=&quot;BBBN56789&quot;&gt;</td>
</tr>
<tr>
<td>&lt;serviceResource&gt;/12345/BBBN56789&lt;/serviceResource&gt;</td>
</tr>
<tr>
<td>&lt;product&gt;CACHING&lt;/product&gt;</td>
</tr>
<tr>
<td>&lt;networkIdentifiers&gt;</td>
</tr>
<tr>
<td>&lt;ni id=&quot;cdn.exampleni.com&quot;&gt;</td>
</tr>
<tr>
<td>&lt;serviceResource&gt;/12345/BBBN56789/cdn.exampleni.com</td>
</tr>
<tr>
<td>&lt;/serviceResource&gt;</td>
</tr>
<tr>
<td>&lt;product&gt;CACHING&lt;/product&gt;</td>
</tr>
</tbody>
</table>

---

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

2 An 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.

3 A 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 supername plus Streaming ID. In Origin Storage, the VHost name.
```xml

</ni>

<ni .../>

</networkIdentifiers>

</service>

<service>...</service>

</services>

<networkIdentifiers>

<ni id="sample-live">

<serviceResource>/12345/BBBN10111/sample-live

</serviceResource>

<product>STREAMING</product>

<type>Live</type>

</ni>

<ni .../>

</networkIdentifiers>

</accessGroup>

Sample 2 Request | https://ws.level3.com/services/cdn/v1.0/12345/BBBN56789
Returns NIs in AG 12345 under SCID BBBN56789.

Sample 2 Response: NIs under a SCID

```xml

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

<accessGroup id="12345" name="My Access Group"

 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.exampleni.com">

<serviceResource>/12345/BBBN56789/cdn.exampleni.com</serviceResource>

<product>CACHING</product>

</ni>

<ni .../>

</networkIdentifiers>

</service>

</services>

</accessGroup>

Sample 3 Request | https://ws.level3.com/services/cdn/v1.0/12345/BBBN56789/cdn.exampleni.com
Returns details on NI cdn.exampleni.com in AG 12345 under SCID BBBN56789.

Sample 3 Response: NI Details

```xml

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

<accessGroup id="12345" name="My Access Group"

 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.exampleni.com">
      <serviceResource>/12345/BBBN56789/cdn.exampleni.com</serviceResource>
      <product>CACHING</product>
    </ni>
  </networkIdentifiers>
</service>
</services>

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/BBBL2345/my.sampleni.com</serviceResource>
          <product>CACHING</product>
          <contentAnalytics>
            <active>true</active>
            <serviceLevel>premium</serviceLevel>
            <samplingRate>premium</samplingRate>
            <collections>
              <collection id="3">
                <serviceResource>/202/BBBL2345/my.sampleni.com/1</serviceResource>
              </collection>
              <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>
            </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


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.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <apiCorrelationId>API-b3dac651-06b7-4b47-8257-ee536b76f500</apiCorrelationId>
  <serviceResource>/[accessGroupID]</serviceResource>
  <services>
    <service id="[SCID]">
      <serviceResource>/[accessGroupID]/[SCID]</serviceResource>
      <product>CACHING</product>
      <networkIdentifiers>
        <ni id="www.level3.com">
          <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="gshmode">*</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 120.

Cost Per Call

See Appendix: Cost per Call.
Invalidate 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**  
```
/invalidations/(version)/(scope)?[force=true][&ignoreCase=true]
(notification=name@exampledomain.com)
```

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>version</strong></td>
<td>Required version.</td>
</tr>
<tr>
<td><strong>scope</strong></td>
<td>(AG)/(SCID)/NI</td>
</tr>
<tr>
<td><strong>force</strong></td>
<td>Optional. Defines Invalidation Type. &quot;true&quot; matches Forced; &quot;false&quot; matches Normal radio buttons in the Media Portal interface. See Submitting invalidation requests.</td>
</tr>
<tr>
<td><strong>ignoreCase</strong></td>
<td>Optional. Defines how URLs are matched to those in cache. See Submitting invalidation requests.</td>
</tr>
<tr>
<td><strong>notification</strong></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. For more information about wildcards and valid path characters, see Path Attributes.</td>
</tr>
</tbody>
</table>

**Example 1 Request**  
https://ws.level3.com/invalidations/v1.0/12345/BBBN56789/cdn.exampleni.com?notification=name@domain.com,name2@domain.com

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.

**Example 1 Body of POST**  
```xml
<paths>
  <path>/images/*.jpg</path>
</paths>
```

Invalidate ALL jpg objects (*.jpg) in the /images directory, including any subdirectories.

**Example 1 Response**  
```xml
<accessGroup id="12345" name="Level 3 - Internal Provisioning CDN"
xsi:namespaceSchemaLocation="https://ws.level3.com/schema/invalidations/v1.0"
xsi:schemaLocation="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"/>
      </networkIdentifiers>
    </service>
  </services>
</accessGroup>
```

---

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 supername plus Streaming ID. In Origin Storage, the VHost name.
Example 2 Request


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):

```xml
<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
<?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.example.com">
     <serviceResource>/12345/BBBN56789/cdn.example.com
    </serviceResource>
    <product>CACHING</product>
    <invalidations>
     <invalidation id="12345/6789@LEVEL3@12345-126419 2808754" path="/directory/structure"/>
     <invalidation id="12345/6789@LEVEL3@12345-126419 2808754" path="/directory/structure02"/>
    </invalidations>
    </ni>
   </networkIdentifiers>
  </service>
  </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):

```xml
<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**

```xml
<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="exampleproperty1">
      <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>/example1.jpg</path>
      <path>/example2.jpg</path>
    </paths>
  </property>
  <property>
    <name>exampleproperty2</name>
    <paths>
      <path>/example3.jpg</path>
    </paths>
  </property>
</properties>
```
<property>
  <name>exampleproperty1</name>
  <paths>
    <path>/example*.jpg</path>
  </paths>
</property>

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

Example 4
Response

Example 4
Response

Possible Status and Error Messages Returned to Client

Cost Per Call
## Invalidations: Request Origin Invalidation (POST)

<table>
<thead>
<tr>
<th><strong>Base URI</strong></th>
<th><code>https://ws.level3.com</code></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Method</strong></td>
<td>POST</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>Request to invalidate content at the origin storage server</td>
</tr>
<tr>
<td><strong>Typical Use</strong></td>
<td>Send invalidation requests for caching properties or streaming IDs on the origin server.</td>
</tr>
<tr>
<td><strong>Schema Location</strong></td>
<td><code>https://ws.level3.com/schema/originInvalidations/v1.0</code></td>
</tr>
</tbody>
</table>

### URI Syntax
- `/originInvalidations/(version)/(scope)?[force=true][&ignoreCase=true][notification=name@exampledomain.com]`
  - **version**: Required version.
  - **scope**: Required. AG is Access Group ID. ORIGIN is the origin storage server name.
  - **force**: Optional. Defines Invalidation Type. "true" matches Forced; "false" matches Normal radio buttons in the Media Portal interface. See Submitting invalidation requests.
  - **ignoreCase**: Optional. Defines how URLs are matched to those in cache. See Submitting invalidation requests.
  - **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.

For more information about wildcards and valid path characters, see Path Attributes.

### Example Request
- `https://ws.level3.com/originInvalidations/v1.0/12345/BBBN56789/cdn.exampleorigin.com?notification=name@domain.com,name2@domain.com`
  - Invalidate path(s) specified in the body of the POST request. Send email notification to two addresses when complete.

### Example Body of POST
- Example (does not use wildcards):
  ```xml
  <paths>
    <path>/path1/example1.jpg</path>
    <path>/path1/example2.jpg</path>
  </paths>
  ```
  - Invalidates two properties with one API request.

### Example Response
- `<accessGroup id="12345" name="Level 3 - Internal Provisioning CDN">
  `<apiCorrelationId>CDNPortal-1289931176093-7961</apiCorrelationId>
  `<serviceResource>/202</serviceResource>
  `<services>
    `<service id="BBBN56789">
      `<serviceResource>/12345/BBBN56789</serviceResource>
      `<product>CACHING</product>
      `<networkIdentifiers`
        `<ni id="cdn.level3.com">

---

7 Service Component Identification. Unique ID number in the order entry system associated to a billable component of a service. See Service.
8 An 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>/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>
</networkIdentifiers>

</service>
</services>

Possible Status and Error Messages Returned to Client

See "Appendix: Error Responses" on page 120.

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**
/invalidations/(version)/(scope)?[id=(invalidation tracking ID)]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>version</td>
<td>Required version.</td>
</tr>
<tr>
<td>scope</td>
<td>Required. Scope must retain sequence that reflects hierarchy.</td>
</tr>
<tr>
<td>or</td>
<td>Scope cannot have an optional middle value. AG is Access Group ID.</td>
</tr>
<tr>
<td>or</td>
<td>NI is network identifier name.</td>
</tr>
<tr>
<td>id (resource)</td>
<td>Optional. Invalidation id – status for that id. Left blank – status for all invalidations submitted within the scope</td>
</tr>
</tbody>
</table>

**Query string variables**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>verbose</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>
<tr>
<td>true</td>
<td>1</td>
</tr>
</tbody>
</table>

**URI Samples**

- **Sample 1**
  - **Request**
    - https://ws.level3.com/invalidations/v1.0/12345/BBBN56789/cdn.exampleni.com
  - **Response**
    - `<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">
            <serviceResource>/12345/BBBN56789</serviceResource>
            <product>CACHING</product>
            <networkIdentifiers>
              <ni id="cdn.exampleni.com">
                <serviceResource>/12345/BBBN56789/cdn.exampleni.com</serviceResource>
              </ni>
            </networkIdentifiers>
          </service>
        </services>`

---

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 supername plus Streaming ID. In Origin Storage, the VHost name.
<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>
  </invalidations>
</accessGroup>

Sample 2 Request
https://ws.level3.com/invalidations/v1.0/12345/BBBN56789
Returns status of invalidations within Access Group 12345 and SCID BBBN56789.

Sample 2 Response

<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>
        </networkIdentifiers>
        <invalidations>
          <invalidation id="DAG_12345/9774@12345-13776-1295308738596"
              percentComplete="100">
            <paths>
              <path>/images/*.cpp</path>
            </paths>
          </invalidation>
          <invalidation id="DAG_12345/9774@DAG_12345-13776-129530828333"
              percentComplete="100">
            <paths>
              <path>/images/*.cpp</path>
            </paths>
          </invalidation>
        </invalidations>
      </networkIdentifiers>
    </service>
  </services>
Example 3 Request

https://ws.level3.com/invalidations/v1.0/12345/BBBN56789/cdn.exampleni.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"
 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.exampleni.com">
          <serviceResource>/12345/BBBN56789/cdn.exampleni.com
          </serviceResource>
        </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="examplen11">
          <serviceResource>/12345/BBBM54321/examplen11</serviceResource>
          <product>STREAMING</product>
          <type>On Demand</type>
        </ni>
      </networkIdentifiers>
    </service>
  </services>
</accessGroup>
Example 4
Response, cont.

Possible Status and Error Messages Returned to Client

Cost Per Call
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=(AG)/(SCID)/(NI)/
```  

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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>Required. Scope for caching is by Access Group ID. AG is Access Group ID.</td>
</tr>
<tr>
<td>geo</td>
<td>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.</td>
</tr>
<tr>
<td>property</td>
<td>Optional. Provides RTM data at the NI level.</td>
</tr>
<tr>
<td>serviceType</td>
<td>Required service type when using caching (as caching RTM is provided on the Access Group level).</td>
</tr>
</tbody>
</table>

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>
  <metro name="Atlanta, GA" region="North America" latitude="33.44"
longiture="-84.23">
    <Mbps>184</Mbps>
    <requestsPerSecond>295.83</requestsPerSecond>
  </metro>
</metros>
</accessGroup>
```

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 supename plus Streaming ID. In Origin Storage, the VHost name.
<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"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<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>1.219999</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>0.7399995</missPerSecond>
**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**

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>accessGroup id</td>
<td>12345</td>
</tr>
<tr>
<td>name</td>
<td>My Access Group</td>
</tr>
<tr>
<td>serviceResource</td>
<td>/12345</td>
</tr>
<tr>
<td>time</td>
<td>2012-03-15 22:44:01 +0000</td>
</tr>
<tr>
<td>services</td>
<td></td>
</tr>
<tr>
<td>service id</td>
<td>12345</td>
</tr>
<tr>
<td>serviceResource</td>
<td>/12345</td>
</tr>
<tr>
<td>networkIdentifiers</td>
<td></td>
</tr>
<tr>
<td>ni id</td>
<td>cdn.level3.com</td>
</tr>
<tr>
<td>mbps</td>
<td>17482.7</td>
</tr>
<tr>
<td>requestsPerSecond</td>
<td>26238.7</td>
</tr>
<tr>
<td>missMbups</td>
<td>921.05</td>
</tr>
<tr>
<td>missPerSecond</td>
<td>986.08</td>
</tr>
<tr>
<td>status404PerSec</td>
<td>3.49</td>
</tr>
<tr>
<td>status503PerSec</td>
<td>0.0</td>
</tr>
<tr>
<td>status504PerSec</td>
<td>0.0</td>
</tr>
<tr>
<td>status5xxPerSec</td>
<td>0.0</td>
</tr>
<tr>
<td>status5tttPerSec</td>
<td>0.0</td>
</tr>
<tr>
<td>status504PerSec</td>
<td>0.0</td>
</tr>
<tr>
<td>status5xxPerSec</td>
<td>0.0</td>
</tr>
<tr>
<td>ptPerSec</td>
<td>233.2</td>
</tr>
<tr>
<td>hitRatePercentage</td>
<td>99.84</td>
</tr>
<tr>
<td>authPercentage</td>
<td>0.0</td>
</tr>
<tr>
<td>signedPercentage</td>
<td>0.0</td>
</tr>
</tbody>
</table>

**Sample 4 Request - NI**

https://ws.level3.com/rtm/cdn/v1.0/12345/BL1111

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>
```

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>accessGroup id</td>
<td>12345</td>
</tr>
<tr>
<td>name</td>
<td>Win Update - MSFT End Cust</td>
</tr>
<tr>
<td>xsi:noNamespaceSchemaLocation</td>
<td><a href="https://ws.level3.com/schema/cachingRTM/v1.0">https://ws.level3.com/schema/cachingRTM/v1.0</a></td>
</tr>
<tr>
<td>xmlns:xsi</td>
<td><a href="http://www.w3.org/2001/XMLSchema-instance">http://www.w3.org/2001/XMLSchema-instance</a></td>
</tr>
<tr>
<td>apiCorrelationId</td>
<td>CDNPortal-1331852553017-5039</td>
</tr>
<tr>
<td>serviceResource</td>
<td>/12345</td>
</tr>
</tbody>
</table>
<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>

Possible Status and Error Messages Returned to Client

See "Appendix: Error Responses" on page 120.

Cost Per Call

See Appendix: Cost per Call.
## Usage Reporting

<table>
<thead>
<tr>
<th><strong>Base URI</strong></th>
<th><a href="https://ws.level3.com">https://ws.level3.com</a></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Method</strong></td>
<td>GET</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Schema Locations</strong></td>
<td><a href="https://ws.level3.com/schema/cachingUsage/v1.0">https://ws.level3.com/schema/cachingUsage/v1.0</a></td>
</tr>
<tr>
<td></td>
<td><a href="https://ws.level3.com/schema/streamingUsage/v1.0">https://ws.level3.com/schema/streamingUsage/v1.0</a></td>
</tr>
<tr>
<td></td>
<td><a href="https://ws.level3.com/schema/originUsage/v1.0">https://ws.level3.com/schema/originUsage/v1.0</a></td>
</tr>
</tbody>
</table>

### URI Syntax

```
/usage/cdn/(version)/(scope)?serviceType=(serviceType)[&geo=(geo)][(&dateFrom=(dateFrom)&dateTo=(dateTo))[&dateMonth=(dateMonth)]
```

- **cdn**  
  Required. Names CDN API as the correct engine.
- **version**  
  Required version.
- **scope**  
  Required. Scope must retain sequence that reflects hierarchy. AG is Access Group\(^{17}\) ID. NI\(^{18}\) is network identifier name.
- **serviceType**  
  Required. Service type: required for scope to AG and SCID. Optional for NI.
- **geo**  
  Optional. Return results by server region or by server metropolitan area.
- **dateFrom**  
  Optional. Date range – starting date/time
- **dateTo**  
  Optional. Date range – end date/time
- **dateMonth**  
  Optional. One month. Used to report caching summary data that includes the 95%.

### Sample Request

```
https://ws.level3.com/usage/cdn/v1.0/12345?
serviceType=caching&dateFrom=201002010000&dateTo=
201002040000
```

### Sample Response - Caching Usage for AG 12345

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

<accessGroup id="12345" name="My Access Group"
xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance>
```

---

\(^{16}\)Service Component Identification. Unique ID number in the order entry system associated to a billable component of a service. See Service.

\(^{17}\)An 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.

\(^{18}\)A 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 supername plus Streaming ID. In Origin Storage, the VHost name.
<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>
    </networkIdentifiers>
  </service>
</services>

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/2001/XMLSchema-instance">
  <apiCorrelationId>CDNPortal-1292450115423-3484
</apiCorrelationId>
  <dataInterval>monthly</dataInterval>
  <serviceResource>/12345</serviceResource>
  <services>
    <service id="BBBN56179">
      <product>CACHING</product>
      <serviceResource>/12345/BBBN56179</serviceResource>
      <parentId>BBBN56179</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>
    </service>
  </services>
</accessGroup>
<serviceResource>/12345/BBBN56179</serviceResource>
<product>CACHING</product>
<networkIdentifiers>
    <ni id="cdn.level3.com">
        <serviceResource>/12345/BBBN56179/cdn.level3.com</serviceResource>
    </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>
</accessGroup>

<dataInterval>hourly</dataInterval>

<summaryData>
    <requests>3706056</requests>
    <volume>141104.97930</volume>
    <averageConnectedPlayers>44485.80000</averageConnectedPlayers>
</summaryData>

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

See next page.

Sample 3, cont.

<?xml version="1.0" encoding="UTF-8" ?>
<accessGroup id="12345" name="My Access Group"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <apiCorrelationId>CDNPortal-1270071184562-8650</apiCorrelationId>
    <serviceResource>/12345</serviceResource>
</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
<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>
    </networkIdentifiers>
  </service>
  <service .../>
</services>

Possible Status and Error Messages Returned to Client
See "Appendix: Error Responses" on page 120.

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:
- https://ws.level3.com/schema/cachingUsageDataInterval/v1.0
- https://ws.level3.com/schema/streamingUsageDataInterval/v1.0
- https://ws.level3.com/schema/originUsageDataInterval/v1.0

URI Syntax:
/usage/cdn/(version)/(scope)?[serviceType=(serviceType)][&geo=(region|metro)&dateFrom=(dateFrom)&dateTo=(dateTo)[&dataInterval=(dataInterval)]

- **cdn** Required. Names CDN API as the correct engine.
- **version** Required version.
- **scope** Scope must retain sequence that reflects hierarchy. AG is Access Group\(^{20}\) ID. NI\(^{21}\) is network identifier name.
- **serviceType** Service type: required for scope to AG and SCID. Optional for NI.
  - "caching" | "c"
  - "streaming" | "s"
  - "origin" | "o"
- **geo** Optional. Returns results by server region or by server area for each time interval requested.
  - "region" | "metro"

**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.

---

\(^{19}\)Service Component Identification. Unique ID number in the order entry system associated to a billable component of a service. See Service.

\(^{20}\)An 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.

\(^{21}\)A 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 supername plus Streaming ID. In Origin Storage, the VHost name.
**Date From**

`yyyyMMddHHnn`

**Date range—starting date/time**

**Date To**

`yyyyMMddHHnn`

**Date range—end date/time**

**Data Interval**

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**

**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>
</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 datInterval for AG 12345 from November & December, 2010

```xml
<?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-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"?>

  <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"?>

  <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/BBCK11111/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 120.

Cost Per Call

See Appendix: Cost per Call.
Usage By Access Group

Base URI  https://ws.level3.com

Method  GET

Description  Returns a hierarchical 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  
/usage/cdn/(version)/(scope-AGID only)?summary=true&dateMonth=<yyyymm>

cdn  Required. Names CDN API as the correct engine.

version  Required version.

scope  AG is Access Group22 ID.

/(AG)/

dateMonth  Date - requested month

yyyymm

Sample Request  https://ws.level3.com/usage/cdn/v1.0/12345?

summary=true&dateMonth=201007

Returns caching usage By Access Group for AG 12345 in July 2010.

Sample Response - By Access Group Usage for AG 12345 Hierarchy (Summary/Table Data) for July 2010

<?xml version="1.0" encoding="UTF-8" ?>
<accessGroup id="12345" name="My Access Group"
xsi:noNamespaceSchemaLocation="https://ws.level3.com/schema/summaryUsage/v1.0"
xmns:xsi="http://www.w3.org/2001/X/MLSchema-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>

---

22An 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>
  <volume>169536.88131</volume>
  <views>2714843</views>
</streaming>

<origin>
  <peakUsage>1733.15556</peakUsage>
</origin>

<accessGroups>
  <accessGroup id="1235" name="L3EULocal">
    <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>
  </accessGroup>
</accessGroups>

Possible Status and Error Messages
See "Appendix: Error Responses" on page 120.

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

**Note:** The first step requires using the "Services Hierarchy (Partially Deprecated)" on page 101 to determine the Content Analytics hierarchy and collections.

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.

**Schema Location**
https://ws.level3.com/schema/contentAnalytics/v1.0
https://ws.level3.com/schema/contentAnalyticsDataInterval/v1.0

**URI Syntax**
/contentAnalytics/(version)/(AG)/(SCID)/(NI)/(collectionID)
?groupBy=(reporttype)&dateFrom=<yyyymmddhhmm>&
dateTo=<yyyymmddhhmm>&id=(value)&dataInterval=(dataInterval)

- **version** Required version.
- **scope** Required. Scope must retain sequence that reflects hierarchy. AG is Access Group. NI is network identifier name.
- **collectionID** Required. Returned with "Services Hierarchy (Partially Deprecated)" on page 101. Same as "Slot" on Media Portal Content Analytics Collections screen.
- **groupBy** Required. Report data type.
  - "collection" | "requestor" |
  - "referer" | "server" | "asn" |
  - "urlDetails" | "statusCodes"
- **dateFrom** Required. Date range – starting date/time

---

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

24. An 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.

25. A 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 supername plus Streaming ID. In Origin Storage, the VHost name.
yyyymmddhhnn

**dateTo**  | **Required.** Date range – end date/time

yyyymmddhhnn

**id**  | Provides collection data for *groupBy* entity. Returned with Summary call.

**dataInterval**  | **Required** for *groupBy* argument. Used with *id* for collection data.

"day" | "hour"

### Sample 1 Request

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

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

```xml
  <apiCorrelationId>CDNPortal-12838977706255-9806</apiCorrelationId>
  <point id="08/02/10">
    <item serviceResource="/12345/BBBN12345/cdn.exampleni.com/1">
      <time>08/02/10</time>
    </item>
  </point>
  <point id="08/03/10">
    <item serviceResource="/12345/BBBN12345/cdn.exampleni.com/1">
      <time>08/03/10</time>
    </item>
  </point>
</data>
```

### Sample 2 Request

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

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

```xml
<accessGroup id="12345" name="12345/BBBN12345"
  <apiCorrelationId>CDNPortal-1283984445602-0592</apiCorrelationId>
  <dataInterval/>
  <serviceResource>/170</serviceResource>
  <services>
    <service id="BBBN12345">
      <serviceResource>/12345/BBBN12345</serviceResource>
      <product>CACHING</product>
    </services>
    <networkIdentifiers>
      <ni id="cdn.exampleni.com">
        <serviceResource>/12345/BBBN12345/cdn.exampleni.com
      </ni>
    </collections>
  </services>
```

---

June 9, 2017

91 of 139
<collection id="1">
    <serviceResource>/12345/BBBN12345/cdn.exampleni.com/1</serviceResource>
    <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)
    </collection>
    </collections>
</networkIdentifiers>
</service>
</services>
</accessGroup>

Sample 3 Request
https://ws.level3.com/contentAnalytics/v1.0/12345/BBBN12345/cdn.exampleni.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

<apiCorrelationId>CDNPortal-1283986736514-8796</apiCorrelationId>

<point id="04/01/10">
    <item serviceResource="/12345/BBBN12345/cdn.exampleni.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.exampleni.com/1">
        <bytes>1036800.00</bytes>
        <name>CHINA</name>
        <requests>1036800</requests>
        <time>04/02/10</time>
        <value>1036800.00</value>
    </item>
</point>
<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.exampleni.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.exampleni.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.exampleni.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/>
</accessGroup>
<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>
      </ni>
    </networkIdentifiers>
  </service>
</services>

Sample 5 Request - Status Codes
https://ws.level3.com/contentAnalytics/v1.0/12345/BBBN56179/cdn.example.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>
    </statusCodes>
    <statusCode code="200">
      <bytes>37601.89</bytes>
      <requests>37400</requests>
      <urlList>
        <summaryData>
          <id>40971</id>
        </summaryData>
        <url>/downloads/Rural_Broadband_Stimulus.pdf</url>
          <bytes>36961.58</bytes>
          <requests>36200</requests>
        </summaryData>
        ...
        <urlList>
          ...
        </urlList>
        ...
        </statusCodes>
      </collection>
    </collections>
  </networkIdentifiers>
</service>
</services>

<table>
<thead>
<tr>
<th>Possible Status and Error Messages Returned to Client</th>
<th>See &quot;Appendix: Error Responses&quot; on page 120.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost Per Call</td>
<td>See Appendix: Cost per Call.</td>
</tr>
</tbody>
</table>
# Origin Storage Overview

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:

<table>
<thead>
<tr>
<th>Service</th>
<th>Method</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>serviceConfiguration</td>
<td>GET</td>
<td>Retrieve configuration details</td>
</tr>
<tr>
<td>serviceConfiguration</td>
<td>POST</td>
<td>Create new network identifier</td>
</tr>
<tr>
<td>serviceConfiguration</td>
<td>DELETE</td>
<td>Delete network identifier</td>
</tr>
<tr>
<td>storageUser</td>
<td>POST</td>
<td>Create new storage user for FTP/SFTP access to storage content</td>
</tr>
<tr>
<td>storageUser</td>
<td>PUT</td>
<td>Update storage user password</td>
</tr>
<tr>
<td>storageUser</td>
<td>DELETE</td>
<td>Delete storage user</td>
</tr>
<tr>
<td>orderStatus</td>
<td>GET</td>
<td>Check order status for service changes and new service orders</td>
</tr>
</tbody>
</table>
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
/serviceConfiguration/(version)/(scope)?requestor=(requestor)
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
requestor [CDATA[ ]] Dataset Values:
{
  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"
}
}

URI Sample
https://ws.level3.com/serviceConfiguration/v1.0/12345/BBBN5678/testaccount.origin.cdn.level3.net

Body:
{
  "uploadLocation": "US East",
  "vhostPrefix": "testaccount",
  "uploadProtocol": "Default",
  "usernames":
  {
    "userId": "bobsmith",
    "password": "letmein"
  }
}

Sample Request
https://ws.level3.com/serviceConfiguration/v1.0/12345/BBBN5678/testaccount.origin.cdn.level3.net?requestor=bob.smith@customer.com

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".

Sample Response
Order ID

Possible Status and Error Messages Returned to Client
201: Request Complete.—the Origin Storage virtual host account has been successfully provisioned and is available for use.
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.
See Appendix A – Status Codes & Error Messages for additional return codes and messages.
## Create New User Account

**Base URI**
https://ws.level3.com

**Method**
POST

**Description**
Create a new User 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

**URI Sample**

**Sample Request**
Body:
```
{
  "userId": "janesmith",
  "password": "letmein"
}
```
Creates a new user, giving "janesmith" access to the "testaccount.origin.cdn.level3.net" account under SCID BBN5678, and assigns the password "letmein"

**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.
# 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**


**Sample Request**


Body:

```json
{
  "userId": "janesmith",
  "password": "letmelogin"
}
```

Updates password to existing "janesmith" user in the "testaccount.origin.cdn.level3.net" account under SCID BBBN5678, 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.

**Note:** This service does not require or accept a content body. Please ensure that the Content-Length HTTP header is set to 0.

**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**


**Sample Request**


Body:

```json
{
  "userId": "bobsmith",
}
```

Deletes "janesmith" User from the "testaccount.origin.cdn.level3.net" account under SCID BBBN5678

**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.
**Services Hierarchy (Partially Deprecated)**

<table>
<thead>
<tr>
<th>Base URI</th>
<th>GET</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Method</strong></td>
<td>GET</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>Returns service details for supplied identifier(s) - service hierarchy.</td>
</tr>
<tr>
<td><strong>Base URI</strong></td>
<td><code>https://ws.level3.com</code></td>
</tr>
<tr>
<td><strong>Method</strong></td>
<td>GET</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>Returns service details for supplied identifier(s) - service hierarchy.</td>
</tr>
<tr>
<td><strong>Typical Use</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Schema Location</strong></td>
<td><code>https://ws.level3.com/schema/services/cdn/v1.0</code></td>
</tr>
<tr>
<td><strong>URI Syntax</strong></td>
<td><code>/services/cdn/(version)/(scope)</code></td>
</tr>
<tr>
<td><strong>cdn</strong></td>
<td>Required. Names CDN API as the correct engine.</td>
</tr>
<tr>
<td><strong>version</strong></td>
<td>Required version.</td>
</tr>
<tr>
<td><strong>scope</strong></td>
<td>Scope must retain sequence that reflects hierarchy. Scope cannot have an optional middle value. AG is <strong>Access Group</strong>(^{27}) ID. NI(^{28}) is network identifier name.</td>
</tr>
<tr>
<td><strong>Sample 1 Request</strong></td>
<td><code>https://ws.level3.com/services/cdn/v1.0/12345</code></td>
</tr>
<tr>
<td><strong>Sample 1 Response: Services under AG</strong></td>
<td>Returns services (and individual NIs) in AG 12345</td>
</tr>
</tbody>
</table>
| **Sample 1 Response:** | `<?xml version="1.0" encoding="UTF-8"?>
<accessGroup id="12345" name="My Access Group"
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.exampleeni.com">
          <serviceResource>/12345/BBBN56789/cdn.exampleeni.com</serviceResource>
        </ni>
      </networkIdentifiers>
      <product>CACHING</product>
    </service>
  </services>` |

---

\(^{26}\)Service Component Identification. Unique ID number in the order entry system associated to a billable component of a service. See Service.

\(^{27}\)An 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.

\(^{28}\)A 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 supername plus Streaming ID. In Origin Storage, the VHost name.
<networkIdentifiers>
</service>
<service>…</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.exampleni.com">
          <serviceResource>/12345/BBBN56789/cdn.exampleni.com</serviceResource>
          <product>CACHING</product>
        </ni>
        <ni ...
      </networkIdentifiers>
    </service>
  </services>
</accessGroup>

Sample 3 Request https://ws.level3.com/services/cdn/v1.0/12345/BBBN56789/cdn.exampleni.com
Returns details on NI cdn.exampleni.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.exampleni.com">
          <serviceResource>/12345/BBBN56789/cdn.exampleni.com</serviceResource>
          <product>CACHING</product>
        </ni>
        <ni ...
      </networkIdentifiers>
    </service>
  </services>
</accessGroup>
<product>CACHING</product>

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"
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
          </collection>
        </collections>
      </ni>
    </networkIdentifiers>
  </service>
</services>
</accessGroup>

Sample 5 Request
https://ws.level3.com/services/cdn/v1.0/[accessGroupID]/[SCID]/www.level3.com?showConfiguration=true

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

<accessGroup id="[accessGroupID]" name="Level 3 Communications, LLC." 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">
        <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>

Possible Status and Error Messages Returned to Client
See "Appendix: Error Responses" on page 120.

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**  
https://ws.level3.com/schema/cachingUsageDataInterval/v1.0
https://ws.level3.com/schema/streamingUsageDataInterval/v1.0
https://ws.level3.com/schema/originUsageDataInterval/v1.0

**URI Syntax**  
/usage/cdn/(version)/(scope)?[serviceType=(serviceType)][&geo=(region|metro)]&dateFrom=(dateFrom)&dateTo=(dateTo)[&dataInterval=(dataInterval)]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>cdn</strong></td>
<td>Required. Names CDN API as the correct engine.</td>
</tr>
<tr>
<td><strong>version</strong></td>
<td>Required version.</td>
</tr>
<tr>
<td><strong>scope</strong></td>
<td>Scope must retain sequence that reflects hierarchy. AG is Access Group(^{30}) ID. NI(^{31}) is network identifier name.</td>
</tr>
<tr>
<td><strong>serviceType</strong></td>
<td>Service type: required for scope to AG and SCID. Optional for NI.</td>
</tr>
<tr>
<td><strong>geo</strong></td>
<td>Optional. Returns results by server region or by server area for each time interval requested.</td>
</tr>
</tbody>
</table>

---

\(^{29}\)Service Component Identification. Unique ID number in the order entry system associated to a billable component of a service. See Service.

\(^{30}\)An 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.

\(^{31}\)A 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 supername 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**

**Origin Storage**

- **"hourly" | "daily" | "monthly"**

**Sample 1**

**Request**

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

**Response**

```xml
<?xml version="1.0" encoding="UTF-8"?>
  <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>
</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>
  <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>
  <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>
    </item>
  </point>
</data>
<peakConnectedPlayers>37.00000</peakConnectedPlayers>
<peakMbps>35.47849</peakMbps>
</item>
</point>
</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"?>
<apiCorrelationId>CDNPortal-1270071184562-8650</apiCorrelationId>
</data>

Possible Status and Error Messages Returned to Client
See "Appendix: Error Responses" on page 120.

Cost Per Call
See Appendix: Cost per Call.
Usage By Access Group

Base URI
https://ws.level3.com

Method
GET

Description
Returns a hierarchical 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
/usage/cdn/(version)/(scope-AGID only)?summary=true&dateMonth=<yyyymm>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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>AG is Access Group ID.</td>
</tr>
<tr>
<td>/&lt;AG&gt;/</td>
<td></td>
</tr>
<tr>
<td>dateMonth</td>
<td>Date - requested month</td>
</tr>
<tr>
<td>yyyymm</td>
<td></td>
</tr>
</tbody>
</table>

Sample Request
https://ws.level3.com/usage/cdn/v1.0/12345?summary=true&dateMonth=201007
Returns caching usage By Access Group for AG 12345 in July 2010.

Sample Response - By Access Group Usage for AG 12345 Hierarchy (Summary/Table Data) for July 2010
<?xml version="1.0" encoding="UTF-8" ?>
<accessGroup id="12345" name="My Access Group"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<apiCorrelationId>CDNPortal-1282780385870-4284</apiCorrelationId>
<serviceResource>/12345</serviceResource>
<cache>
  <volume>3332.36564</volume>
  <requests>21219582</requests>
  <mbps95>11.49775</mbps95>
</cache>
<streaming>
  <volume>169536.88131</volume>
  <views>2714843</views>
</streaming>

32An 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.
<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>
</accessGroup>
</accessGroups>

Possible Status and Error Messages Returned to Client
See "Appendix: Error Responses" on page 120.

Cost Per Call
See Appendix: Cost per Call.
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.

Note: The first step requires using the "Services Hierarchy (Partially Deprecated)" on page 101 to determine the Content Analytics hierarchy and collections.

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.

Schema Location
https://ws.level3.com/schema/contentAnalytics/v1.0
https://ws.level3.com/schema/contentAnalyticsDataInterval/v1.0

URI Syntax
/contentAnalytics/(version)/(AG)/(SCID33)/(NI)/(collectionID)?groupBy=(reporttype)&dateFrom=<yyyymmdhhmm>&dateTo=<yyyymmdhhmm>&id=(value)&dataInterval=(dataInterval)

version Required version.

scope
/(AG)/(SCID33)/(NI)/

collectionID Required. Returned with "Services Hierarchy (Partially Deprecated)" on page 101. Same as "Slot" on Media Portal Content Analytics Collections screen.

groupBy Required. Report data type.
"collection" | "requestor" | "referer" | "server" | "asn" | "urlDetails" | "statusCodes"

dateFrom Required. Date range – starting date/time

yyyymmdhhmm

33Service Component Identification. Unique ID number in the order entry system associated to a billable component of a service. See Service.
34An 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.
35A 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 supername plus Streaming ID. In Origin Storage, the VHost name.
**dateTo** | **Required.** Date range – end date/time  
--- | ---  
`yyyymmddhhnn`  

**id** | Provides collection data for `groupBy` entity. Returned with Summary call.  
--- | ---  

**dataInterval** | **Required** for `groupBy` argument. Used with `id` for collection data.  
--- | ---  
"day" | "hour"  

| Sample 1 Request | Sample 1 Response - Collection Trend for AG 12345 for August 1 to 4, 2010 |
| --- | ---  
| [Sample 1 Request](https://ws.level3.com/contentAnalytics/v1.0/12345/BBBN12345/cdn.exampleni.com/1?groupBy=collection&dateFrom=201008010000&dateTo=201008300000&dataInterval=day) | `<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.exampleni.com/1">`  
|  | `<time>08/02/10</time>`  
|  | `</item>`  
|  | `</point>`  
|  | `<point id="08/03/10">`  
|  | `<item serviceResource="/12345/BBBN12345/cdn.exampleni.com/1">`  
|  | `<time>08/03/10</time>`  
|  | `</item>`  
|  | `</point>`  
|  | `</data>`  

| Sample 2 Request | Sample 2 Response - Requestor Summary for AG 12345 for April 1 to April 5, 2010 |
| --- | ---  
| [Sample 2 Request](https://ws.level3.com/contentAnalytics/v1.0/12345/BBBN12345/cdn.exampleni.com/1?groupBy=requestor&dateFrom=201008010000&dateTo=201008300000) | `<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.exampleni.com">`  
|  | `<serviceResource>/12345/BBBN12345/cdn.exampleni.com`  
|  | `</serviceResource>`  
|  | `</collections>`  
|  | `<collection id="1">`  
|  | `<serviceResource>/12345/BBBN12345/cdn.exampleni.com/1`  
|  | `</serviceResource>`  

---
<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>
</nir>
</networkIdentifiers>
</service>
</services>
</accessGroup>

Sample 3 Request
https://ws.level3.com/contentAnalytics/v1.0/12345
/BBBN12345/cdn.exampleni.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

<apiCorrelationId>CDNPortal-1283986736514-8796</apiCorrelationId>

<point id="04/01/10">
  <item serviceResource="/12345/BBBN12345/cdn.exampleni.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.exampleni.com/1">
    <bytes>1027296.00</bytes>
  </item>
</point>

<item serviceResource="/12345/BBBN12345/cdn.exampleni.com/1">
  <bytes>1036800.00</bytes>
</item>

<item serviceResource="/12345/BBBN12345/cdn.exampleni.com/1">
  <bytes>1027296.00</bytes>
</item>
<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.exampleni.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.exampleni.com/1">
  <bytes>1036800.00</bytes>
  <name>CHINA</name>
  <requests>1036800</requests>
  <time>04/04/10</time>
  <value>1036800.00</value>
 </item>
</point>

Sample 4 Request - URL Details

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

<accessGroup id="12345" name="Level 3 - Internal Provisioning CDN"
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>
        </ni>
      </networkIdentifiers>
      <collections>
        <collection id="51">
          <serviceResource>/12345/BBBN56179/cdn.level3.com/51</serviceResource>
          <summaryData>
            <id>905243</id>
            <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>
        </collection>
        ...
      </collections>
    </service>
  </services>
</accessGroup>

Sample 5 Request - Status Codes

https://ws.level3.com/contentAnalytics/v1.0/12345/BBBN56179/cdn.examplenl.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"
xmns: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>
                </ni>
            </networkIdentifiers>
        </service>
        <collections>
            <collection id="51">
                <serviceResource>/12345/BBBN56179/cdn.level3.com/51</serviceResource>
                <statusCodes code="2xx">
                    <bytes>376.952</bytes>
                    <requests>37900</requests>
                    <statusCode code="200">
                        <bytes>376.0189</bytes>
                        <requests>37400</requests>
                        <urlList>
                            <summaryData>
                                <id>40971</id>
                            </summaryData>
                            <url>/downloads/Rural_Broadband_Stimulus.pdf</url>
                            <bytes>36961.58</bytes>
                            <requests>36200</requests>
                        </urlList>
                        ...
                    </statusCode>
                    ...
                </statusCodes>
            </collection>
        </collections>
    </services>
</accessGroup>

Possible Status and Error Messages Returned to Client

See "Appendix: Error Responses" on page 120.

Cost Per Call

See Appendix: Cost per Call.
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.

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.

Schema Location
https://ws.level3.com/schema/serviceConfiguration/v1.0

URI Syntax
/serviceConfiguration/(version)/(scope)?serviceType=(serviceType)

/serviceConfiguration/(version)/(scope)?serviceType=(serviceType)

version [CDATA[ ]]
Values: "v1.0" (required)

scope [CDATA[ ]]
Must retain sequence that reflects hierarchy and cannot have an optional middle value.

/(AG)/
AG = Access Group

or

/(AG)/(SCID)/
SCID = Service Component IDentity

or

/(AG)/(SCID)/(NI)/serviceType
NI = Network Identifier serviceType

Values:
"storage" | "o"
"download" | "d"
"streaming" | "s"

Required for Scope to AG and SCID. Optional for NI.

Sample URI
https://ws.level3.com/serviceConfiguration/v1.0/12345
https://ws.level3.com/serviceConfiguration/v1.0/12345/BBBN5678
https://ws.level3.com/serviceConfiguration/v1.0/12345/BBBN5678/testaccount.origin.cdn.level3.net
Sample Request

https://ws.level3.com/serviceConfiguration/v1.0/12345/BBBN5678/testaccount.origin.cdn.level3.net

Returns all streams in Access Group 12345 including those with no data.

Sample Response

```xml
<?xml version="1.0" encoding="UTF-8"?>
<accessGroup id="12345" name="My Access Group"
xsi:noNamespaceSchemaLocation="https://ws.level3.com/schema/serviceConfiguration/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>
<services>
    <service id="BBBN56789">
        <serviceResource>/12345/BBBN56789</serviceResource>
        <product>STORAGE</product>
        <networkIdentifiers>
            <ni id="testaccount">
                <serviceResource>/12345/BBBN56789/testaccount.origin.cdn.level3.net</serviceResource>
            </ni id="testaccount">
        </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.
### 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"
Appendix: Error Responses

In every case where the service returns a 4xx or 5xx response code to the caller, the entity body shall contain an XML representation of more detailed error information to allow diagnosis and appropriate handling.

XML format:

```xml
<?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>
```

<table>
<thead>
<tr>
<th>Description</th>
<th>ErrorCode</th>
<th>Associated HTTP Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>URI cannot be parsed</td>
<td>21700</td>
<td>400</td>
</tr>
<tr>
<td>Unknown Access Group</td>
<td>21701</td>
<td>404</td>
</tr>
<tr>
<td>Unknown SCID</td>
<td>21702</td>
<td>404</td>
</tr>
<tr>
<td>Unknown network identifier</td>
<td>21703</td>
<td>404</td>
</tr>
<tr>
<td>SCID is not a child of specified Access Group</td>
<td>21704</td>
<td>404</td>
</tr>
<tr>
<td>Network Identifier is not child of specified SCID</td>
<td>21705</td>
<td>404</td>
</tr>
<tr>
<td>Network identifier is not child of specified Access Group</td>
<td>21706</td>
<td>404</td>
</tr>
<tr>
<td>Not authorized for requested scope (Access Group, SCID or Network Identifier)</td>
<td>21707</td>
<td>401</td>
</tr>
<tr>
<td>Unknown service type</td>
<td>21708</td>
<td>400</td>
</tr>
<tr>
<td>Unknown group options</td>
<td>21709</td>
<td>400</td>
</tr>
<tr>
<td>Unknown report dimensions</td>
<td>21710</td>
<td>400</td>
</tr>
<tr>
<td>Specified report dimensions unavailable for service type</td>
<td>21711</td>
<td>400</td>
</tr>
<tr>
<td>Date range too wide</td>
<td>21712</td>
<td>403</td>
</tr>
<tr>
<td>Date range too wide for requested granularity</td>
<td>21713</td>
<td>403</td>
</tr>
<tr>
<td>Too many services included in requested scope</td>
<td>21714</td>
<td>403</td>
</tr>
<tr>
<td>Requested time granularity unavailable</td>
<td>21715</td>
<td>403</td>
</tr>
<tr>
<td>Invalidation ID does not exist</td>
<td>21716</td>
<td>404</td>
</tr>
<tr>
<td>Unsupported/Unknown API version</td>
<td>21717</td>
<td>404</td>
</tr>
<tr>
<td>Internal Server Error (general)</td>
<td>21718</td>
<td>500</td>
</tr>
<tr>
<td>Request timestamp too old</td>
<td>21719</td>
<td>403</td>
</tr>
<tr>
<td>Credit check failure (Deprecated)</td>
<td>21720</td>
<td>403</td>
</tr>
<tr>
<td>API Key is disabled</td>
<td>21721</td>
<td>403</td>
</tr>
<tr>
<td>Access Group API privileges suspended</td>
<td>21722</td>
<td>403</td>
</tr>
<tr>
<td>API Key request rate too high</td>
<td>21723</td>
<td>503</td>
</tr>
<tr>
<td>Could not parse the request header date.</td>
<td>21724</td>
<td>500</td>
</tr>
<tr>
<td>Unauthorized access. Invalid id/secret combination.</td>
<td>21725</td>
<td>403</td>
</tr>
<tr>
<td>Authorization header not recognized.</td>
<td>21726</td>
<td>403</td>
</tr>
<tr>
<td>This operation is not permitted for this key.</td>
<td>21727</td>
<td>403</td>
</tr>
<tr>
<td>Invalidation status not available</td>
<td>21728</td>
<td>404</td>
</tr>
<tr>
<td>Parameter value is invalid</td>
<td>21729</td>
<td>400</td>
</tr>
<tr>
<td>Error Description</td>
<td>Status Code</td>
<td>Reason Code</td>
</tr>
<tr>
<td>----------------------------------------------------------------------------------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Date range too narrow for requested granularity.</td>
<td>21730</td>
<td>403</td>
</tr>
<tr>
<td>Multiple wildcard paths not allowed.</td>
<td>21731</td>
<td>403</td>
</tr>
<tr>
<td>Maximum number of paths exceeded.</td>
<td>21732</td>
<td>403</td>
</tr>
<tr>
<td>dataInterval specified is not valid.</td>
<td>21733</td>
<td>403</td>
</tr>
<tr>
<td>Invalidation Path XML is not valid.</td>
<td>21734</td>
<td>400</td>
</tr>
<tr>
<td>Invalidation Path must start with a slash.</td>
<td>21735</td>
<td>400</td>
</tr>
<tr>
<td>Access Group ID must be numeric.</td>
<td>21736</td>
<td>400</td>
</tr>
<tr>
<td>serviceType must be specified.</td>
<td>21737</td>
<td>400</td>
</tr>
<tr>
<td>dateFrom and dateTo must be specified.</td>
<td>21738</td>
<td>400</td>
</tr>
<tr>
<td>Requested schema name invalid.</td>
<td>21739</td>
<td>400</td>
</tr>
<tr>
<td>Date format invalid.</td>
<td>21740</td>
<td>400</td>
</tr>
<tr>
<td>Accept header value not supported.</td>
<td>21741</td>
<td>400</td>
</tr>
<tr>
<td>dateFrom comes after dateTo.</td>
<td>21742</td>
<td>400</td>
</tr>
<tr>
<td>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.</td>
<td>21743</td>
<td>404</td>
</tr>
<tr>
<td>dateMonth must be specified (yyyyymm)</td>
<td>21744</td>
<td>400</td>
</tr>
<tr>
<td>Access Group not set up for Media Portal invalidation.</td>
<td>21745</td>
<td>500</td>
</tr>
<tr>
<td>groupBy must be specified.</td>
<td>21746</td>
<td>400</td>
</tr>
<tr>
<td>Unknown groupBy value.</td>
<td>21747</td>
<td>400</td>
</tr>
<tr>
<td>Collection is not a child of specified NI. Please verify the requested resource.</td>
<td>21748</td>
<td>404</td>
</tr>
<tr>
<td>id must be specified.</td>
<td>21749</td>
<td>400</td>
</tr>
<tr>
<td>The requested groupBy is not configured for the requested Content Analytics resource.</td>
<td>21750</td>
<td>403</td>
</tr>
<tr>
<td>Live streaming content is not supported for invalidation.</td>
<td>21751</td>
<td>400</td>
</tr>
<tr>
<td>Network Identifier(s) are not valid for requested Access Group: [NIs]</td>
<td>21752</td>
<td>404</td>
</tr>
<tr>
<td>Invalidation path specified is not valid.</td>
<td>21753</td>
<td>400</td>
</tr>
<tr>
<td>Mobile login failed.</td>
<td>21754</td>
<td>500</td>
</tr>
<tr>
<td>Mobile login failed, invalid username/password.</td>
<td>21755</td>
<td>403</td>
</tr>
<tr>
<td>API request rate triggered denial of service warning, request will not be fulfilled.</td>
<td>21756</td>
<td>403</td>
</tr>
<tr>
<td>Invalidation failed - network error.</td>
<td>21757</td>
<td>400</td>
</tr>
<tr>
<td>Invalid use of property parameter.</td>
<td>21758</td>
<td>400</td>
</tr>
<tr>
<td>API Key ID must be numeric.</td>
<td>21759</td>
<td>400</td>
</tr>
<tr>
<td>Unsupported query string combination.metricLens and metric query string parameters were sent on the same request.</td>
<td>21760</td>
<td>403</td>
</tr>
<tr>
<td>mediaType query string parameter is required. If the /clientSideStats service is called, the mediaType query string parameter must be provided.</td>
<td>21761</td>
<td>404</td>
</tr>
<tr>
<td>Invalid metric value for purchased services. The customer's Client Side Stats contract does not include the requested metric.</td>
<td>21762</td>
<td>403</td>
</tr>
<tr>
<td>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.</td>
<td>21763</td>
<td>400</td>
</tr>
<tr>
<td>Access Group does not have Client Side Stats permission.</td>
<td>21764</td>
<td>403</td>
</tr>
</tbody>
</table>
Appendix: Usage Granularity Rules

Caching

<table>
<thead>
<tr>
<th>Date Range</th>
<th>Time Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 day</td>
<td>5 min</td>
</tr>
<tr>
<td>2 - 20 days</td>
<td>1 hour</td>
</tr>
<tr>
<td>21 - 90 days</td>
<td>1 day</td>
</tr>
<tr>
<td>91+ days</td>
<td>1 month</td>
</tr>
</tbody>
</table>

Streaming

<table>
<thead>
<tr>
<th>Date Range</th>
<th>Time Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 20 days</td>
<td>1 hour</td>
</tr>
<tr>
<td>21 - 90 days</td>
<td>1 day</td>
</tr>
<tr>
<td>91+ days</td>
<td>1 month</td>
</tr>
</tbody>
</table>

Origin Storage

<table>
<thead>
<tr>
<th>Date Range</th>
<th>Time Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 90 days</td>
<td>1 day</td>
</tr>
<tr>
<td>91+ days</td>
<td>1 month</td>
</tr>
</tbody>
</table>

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

Caching date range: 1/1/2010 – 2/4/2010, 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: 1/1/2010 to 1/2/2010, time intervals:

- **5 min**—allowed.
- **Hourly**—allowed.
- **Daily**—allowed.
- **Monthly**—not allowed, date range too narrow for time interval (fewer than 28 days).
# 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**
```xml
<?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>
  </accessGroups>
</accessGroup>
```

**Possible Status and Error Messages Returned to Client**
See "Appendix: Error Responses" on page 120.

**Cost Per Call**
See Appendix: Cost per Call.
## 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.
## Delete Existing Virtual Host

**Base URI**  
https://ws.level3.com

**Method**  
DELETE

**Description**  
Delete an existing Virtual Host from the Level 3 Origin Storage service.

**Note:** This service does not require or accept a content body. Please ensure that the Content-Length HTTP header is set to 0.

**Schema Location**  
https://ws.level3.com/schema/serviceConfiguration/v1.0

**URI Syntax**  
/serviceConfiguration/(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/serviceConfiguration/v1.0/12345/BBBN5678/testaccount.origin.cdn.level3.net

**Sample Request**  
https://ws.level3.com/serviceConfiguration/v1.0/12345/BBBN5678/testaccount.origin.cdn.level3.net

Deletes the Origin Storage virtual host account "testaccount.origin.cdn.level3.net" from the child service assigned to SCID BBN5678.

**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.
## Vyvx Reservation - Get Active Status

**Base URI**

https://ws.level3.com/vyvx/1.0

**Method**

GET

**Description**

Get the status of all active services (reservations).

**Note:** This service does not require or accept a content body. Please ensure that the Content-Length HTTP header is set to 0.

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.

Possible state 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

**URI Syntax**

`/customer/service/_active/status`

**Sample Request**

https://ws.level3.com/vyvx/1.0/customer/service/_active/status

**Sample Response**

```json
{
  "data": {
    "services": {
      "1234567": { "state": "offline","ingress_state": "online" },
      "9999999": { "state": "offline","ingress_state": "online" }
    },
    "time": "2017-01-02T15:04:05.000Z"
  }
}
```

**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.
Vyvx Reservation - Get Service Status

**Base URI**
https://ws.level3.com/vyvx/1.0

**Method**
GET

**Description**
Get the status 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.

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.

Possible state 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

**URI Syntax**
/customer/service/(id)/status

id : Reservation ID (required)

**Sample Request**
https://ws.level3.com/vyvx/1.0/customer/service/1234567/status

**Sample Response**
```json
{
  "data": {
    "services": {
      "1234567": {
        "state": "offline",
        "ingress_state": "online"
      }
    }
  },
  "time": "2006-01-02T15:04:05.000Z"
}
```

**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.
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?start=2016-02T15:04:05.000Z&end=2016-01-03T15:04:05.000Z

**Sample Response**
```json
{
  "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"
}
```

**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: 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 the Media Portal API Debugger

Audience
This topic is intended for anyone who builds Media Web Services (API) requests and wants to use the MPAPIDebugger.air application as a testing tool.

What is the 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 132. 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:
1. Enter the **API Key ID** and **Secret**.

   Generate the information for these two fields in the Media Portal using the steps in "API Security Keys" on page 6.

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. For more information on HTTP methods, see http://en.wikipedia.org/wiki/HTTP#Request_methods.

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 13

   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 65

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 the Media Portal APIs - this is not a full application, only a sample class implementation.
- C Sharp sample class allowing connection to the Media Portal APIs.
- Java class allowing connection to the 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 the 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.
Index

A
Access Group through API
   API by AG, 119, 121, 145, 147
   API data, 108
AP Interfaces
   Access Control, 57
   Access Group Hierarchy, 80, 163
   Error Responses, 157
   Invalidations-Check Status (GET), 96
   Invalidations-Request Invalidation (POST), 88
   Invalidations-Request Origin Invalidation (POST), 94
   Key, 18, 20, 22, 25, 26, 27, 28, 29, 30,
      31, 32, 33, 34, 36, 37, 39, 41, 43, 45, 46,
      47, 48, 50, 51, 52, 54, 55, 56, 58, 59, 61,
      63, 64, 65, 67, 69, 70, 71, 73, 75, 77, 78,
      128, 129, 131, 132, 133, 154, 156, 165,
      166, 167, 169, 171
   RTM (Caching), 102
   Services Hierarchy, 82, 134
   Usage by Access Group, 119, 121, 145,
      147
   Usage by Data Interval, 114, 140
   Usage Granularity Rules, 161
   Usage Reporting, 108
API
   Architecture, 2
   Definition, 2
   Getting Started, 3
   Interfaces overview, 4
   Overview, 1
   Rate limit, 16
   Request processing, 14
   Roles, 16
   Sample code, 178
   Security, 8
   User interface, 10
API Key
   Defined, 13
   Disable, 12
   editing, 11
   Hierarchy, 11
   New key, 10
   New secret, 12
   Status, 13
   API request, signature form, 15
   API requests, signing mechanism, 8
C
   Code, API samples, 178
   Configuration Groups, 44
D
   Data Interval, API, 114, 140
   Date & Time Usage Rules, API, 161
   Date Range reports, API, 114, 140
   Debugging API requests, 9
   Definition
      API, 2
   Depleted, API key status, 13
   Disable, API Key, 12
   Disabled, API key status, 13
E
   Errors
      AP Interfaces, 157
   Existing API key, 11
H
   Hash, HMAC-SHA1 for API signatures, 13
   Hierarchy
      API keys, 11
      Hierarchy, AG Usage API, 119, 121, 145,
         147
      HMAC-SHA1 hash for API signatures, 13
   Host, for API calls, 6
   HTTP debugger, for API, 9
   HTTP errors, API, 157
I
   Interfaces, API, 4
   Invalidations
      Origin, 94
K
   Key
      Adding new for API, 10
      API overview, 10
      New secret, 12
M
MPA, 15

N
NI, API data, 108

O
Origin Invalidations, 94

P
Permission
   in Access Group, 16

R
Rate limits, API, 16
Request processing, API, 14
RESTful web services, 2
Roles
   API, 16

S
Sample Code, API, 178

SCID, API data, 108
Secret, for API Key, 10, 13
Security
   API Overview, 8
   Using API Keys, 10
Signature
   API, 8, 15
Start one API, 12
Status
   API key, 13
Stop one API, 12
Suspended, API key status, 13

T
Time & Date Usage Rules, API, 161

U
Usage Rules, API, 161