

Enumerator API

Content Router provides a public HTTP 1.1 server interface component as part of the Content Router Publisher service. The purpose of the HTTP interface is to enable a simple, standards-based approach for building "plug-in like" Netmail Store internal and third-party applications that require some form of object enumeration. The Content Router Replicator service utilizes this interface extensively to provide reliable object replication across Wide Area Networks. Some other applications might include third party object replication and/or backup applications, object index/search engines, metadata query engines and virus scan applications. The base URL for any enumerator is:
`http://<publisherHost>:<publicationServerPort>/`

Enumerator Types

Enumerator types are defined based on the type of data that should be returned with the response. The default type is a Metadata enumerator. The list of supported types is as follows:

- **Metadata Enumerator:** Metadata enumerators return the UUIDs and all associated metadata for objects that match the specified filters. Metadata enumeration includes all creates and updates regardless of subsequent deletes. Every event associated with a mutable object (POST, COPY, APPEND, PUT, DELETE, etc) is included as a separate enumeration item. Metadata enumerators include metadata for both named and unnamed Netmail Store objects.
- **UUID Enumerator:** UUID enumerators return the UUIDs for all objects that match the specified filters. UUID enumeration includes the UUIDs for create events of immutable objects unless a subsequent delete event has been registered by the Publisher for the same object. For mutable objects, the alias UUIDs for creates and updates are included regardless of subsequent deletes in order to ensure proper processing of all object revisions across different nodes and clusters. This will result in the same alias UUID being enumerated multiple times if there are update or delete events. Only unnamed Netmail Store objects are returned by the UUID enumerator.
- **Event Enumerator:** Event enumerators return the UUIDs for objects that match the specified filters as well as the last event associated with the UUID (create, update or delete). Only unnamed Netmail Store objects are returned by the Event enumerator.

Note: Delete events are considered to match all channels and are sent to all subscribers. This ensures delete events transmitted via a previous rule version are known to be deleted even if they no longer match the current rules.

Enumerator Start

The Enumeration Start command instantiates an object enumerator for a given channel in Publisher, and returns a unique identifier for this enumerator. The format of the request is as follows:

```
POST /<channel>?type=<enumerator type>&start=<date-time1>&end=<date-time2>
HTTP/1.1
Host: <publisherhost>
```

Here channel is the "subscription name" as specified when configuring a Content Router replicator service. It corresponds to one of the sets of filter rules identified by a select tag in the publisher *rules.xml* file. The start and end parameters delimit the create dates of objects to be enumerated for metadata and UUID enumerator types. Event type enumerators do not support start and end dates. Both dates are ISO 8601 date-time values; RFC 1123 formatted date-time values are not yet supported. The time-of-day specification may be omitted, in which case the time 00:00:00 is assumed. The start date-time is inclusive, the end date-time non-inclusive.

Enumerator Start Query Arguments

The following query arguments are supported for the Start command. The Publisher saves all provided query arguments so arguments need not be supplied with every call after the initial one unless a change is desired.

Argument Name	Description
type	The case-insensitive value for enumerator type: Metadata, UUID, or Event.
start	An ISO 8601 formatted date-time string that corresponds to the start of the time range that should be enumerated. All time stamp comparisons are based on file creation timestamps.
end	An ISO 8601 formatted date-time string that corresponds to the end of the time range that should be enumerated. All time stamp comparisons are based on file creation timestamps.

Enumerator Start Response

The normal response to an Enumerator Start command is:

```
HTTP/1.1 201
Date: ...
Server: Content Router Publisher
Content-UUID: 41a140b5271dc8d22ff8d027176a0821
Content-Sync-Token: <token value>
Content-Type: text/plain
Content-Length: <bytes in response body>
Object Enumerator created - channel: '<channel name>', type:
  '<enumerator type>'[,start: '<date-time1>'][, end: '<date-time2>']
```

The Content-UUID header contains a generated UUID identifying the enumerator instance created. The Content-Sync-Token is used by the Next command to indicate the previous request was successfully received. The (optional) start/end date-times in the response body are displayed as unix epoch time seconds. If for any reason the request is not successful, a 404 response code will be returned with a descriptive message in the response body as to the encountered problem.

Enumerator Next

The Enumerator Next command is a request for the next objects in an enumeration which was previously initiated with an Enumerator Start command. For performance reasons Publisher does not attempt to retrieve elements in a specific order. The format of the request is as follows:

```
GET /<Enumerator UUID>?maxItems=<max-objects-to-retrieve> HTTP/1.1
Host: <publisherhost>
```

Note that the maxItems query argument is only supported for the UUID and Event enumerator types. A maxItems argument supplied for a Metadata enumerator will be ignored.

Enumerator Next Query Arguments

The following optional query arguments are defined for the Next command only and used to convey status information about the Enumerator to the Publisher. The Publisher will use some but not all of this information to update Enumerator status on the Publisher console. See also the Configuration and Status Query Arguments section below for additional optional arguments that are shared with the Start method:

Argument Name	Description
upTime	Number of seconds since the subscriber was started.
backLog	Number of items the subscriber is holding that it has not begun processing.
inProgress	Number of items the subscriber is actively processing.
dropped	Number of items that the subscriber has failed to process. The dropped query argument provides the subscriber a way to report possible trouble conditions to the Publisher. Subscribers should take care to only "drop" events after sufficient retries. Note: Dropped events will not be resent by the Publisher without manual intervention using the Republish or Republish All commands from the Publisher Console.
syncToken	The value of the Content-Sync-Token header from the last successfully received Start or Next response. The Content-Sync-Token is used to confirm with the Publisher that a request was successfully received and ensure UUIDs are not missed due to a network or connectivity issue. If the syncToken matches the one expected by the Publisher from the previous request, the next set of requested UUIDs will be sent. If the syncToken does not match, the Publisher will resend the previous set of UUIDs. If the syncToken is not present on the request, the Publisher will automatically send the next set of requested UUIDs.

Enumerator Next Response

A typical normal response to an Enumerator Next command of type UUID or Event is:

```
HTTP/1.1 200
Date: ...
Server: Content Router Publisher
Content-Type: text/plain
Content-Length: <bytes in response body>
Content-Sync-Token: <token value>
```

For a Metadata Enumerator, the response has an additional "Content-UUID" header containing the object's UUID.

```
HTTP/1.1 200
Date: ...
Server: Content Router Publisher
Content-UUID: <UUID of object>
Content-Type: text/plain
Content-Length: <bytes in response body>
Content-Sync-Token: <token value>
```

The content and format of each line of the response body varies by enumerator type as follows:

- UUID enumerators: "<object uuid>"
- Event enumerators: "UUID,<evType>" , where evType equals event types of 1 (deleted), 2 (created), or 4 (created/updated for anchor streams) (3 is not currently utilized)
- Metadata enumerators: "<header name>: <value>" (for each header present on the object)

Metadata enumerators will only contain metadata for a single object, but UUID and Event enumerators support inclusion of data for multiple objects. If for any reason the request is not successful, a 404 response code will be returned with a descriptive message in the response body as to the encountered problem.

Enumerator End

The Enumerator End command is called to end an enumeration which was previously initiated with an Enumerator Start command. The format of the request is as follows:

```
DELETE /<Enumerator UUID> HTTP/1.1
Host: <publisherhost>
```

End Response

The normal response to an Enumerator End command is:

```
HTTP/1.1 200
Date: ...
Server: Content Router Publisher
Content-Type: text/plain
Content-Length: ...
Object Enumerator deleted
```

If, for any reason, the request is not successful, a 404 response code will be returned with a descriptive message in the response body as to the encountered problem.

Enumerator Timeout

The Content Router Publisher will repurpose the existing configuration parameter called *subscriberTimeout*. If an enumerator is not accessed using GET for a time period equal to *subscriberTimeout* seconds, it will be automatically terminated and removed. The timeout period applies to enumerators created for any channel of a Publisher service. It is configurable in the *publisher.cfg* file, with a default value of 90000 seconds (25 hrs). An individual Enumerator may override the publisher timeout value. This is done by including a timeout query arg on the POST request, as follows:

```
POST /<channel>?start=<date-time1>&end=<date-time2>&type=UUID&timeout=86400 HTTP/1.1
```

Configuration and Status Query Arguments

The following can be supplied on either an enumerator Start or Next command:

Argument Name	Description
version	Enumerator client's software version string. When provided, this value will be displayed on the Publisher console in association with the channel/subscription rule set name.
context	The descriptive name for an enumerator instance of a particular channel. When provided, this value will be displayed on the Publisher console in the expandable section with enumerator details.
maxItems	Maximum number of items to send in this request response. Sending a value of 0 is a logical pause that changes the enumerator's state to "Paused" on the Publisher console. The enumerator will remain paused until a nonzero maxItems value is received. If neither the enumerator start or a next request specifies a value for the maxItems parameter, Publisher sets the maxItems value to 5000. A negative or non-integer value is interpreted as a 0.
offlineAfter	Number of seconds the Publisher should wait after the last enumerator query before declaring the enumerator as offline. When used, this parameter should be some multiple of the expected interval between enumerator queries. When not specified, a default value for offlineAfter will be derived from the Publisher's "subscriberOfflineAfter" configuration value.
errOfflineAfter	Number of seconds the Publisher should wait after the last enumerator query before showing an offline error. When used, this parameter should be some multiple of the expected interval between enumerator queries. When not specified, a default value for errOfflineAfter will be derived from the Publisher's "subscriberErrOfflineAfter" configuration value.
timeout	Number of seconds before the Publisher will terminate an enumerator if it has not been heard from. The minimum supported value is 600 seconds.
