

Newsgate
Plug-in Guide
3.1.1-2

CUE

Table of Contents

| | |
|--|----|
| 1 Introduction | 4 |
| 2 Overview | 5 |
| 2.1 What is Newsgate plug-in | 5 |
| 2.2 What purpose does Newsgate serve? | 5 |
| 2.3 How does the Newsgate plug-in work? | 5 |
| 2.3.1 CUE integration | 5 |
| 2.3.2 URLResolver | 6 |
| 2.3.3 Article details | 6 |
| 2.3.4 Exporting content types | 7 |
| 2.3.5 Exporting section structure | 7 |
| 2.3.6 Importing taxonomy definitions | 7 |
| 3 Installation | 8 |
| 3.1 Conventions | 8 |
| 3.2 Install Newsgate | 8 |
| 3.3 Verify The Installation | 9 |
| 3.4 Configure XSL style sheet | 9 |
| 3.5 Configure doctype | 9 |
| 3.6 Publication mapping | 10 |
| 3.6.1 Create a publication mapping | 10 |
| 3.6.2 Register the mapping | 10 |
| 3.7 Enabling export of content types | 10 |
| 3.7.1 Enable service | 11 |
| 3.7.2 Configure XSL style sheet | 11 |
| 3.8 Enabling export of section structure | 11 |
| 3.8.1 Enable service | 11 |
| 3.8.2 Configure XSL style sheet | 12 |
| 3.9 Configure export | 12 |
| 3.10 Configuration in CCI | 12 |
| 3.11 Troubleshooting | 12 |
| 3.11.1 Verify plug-in installation | 12 |
| 3.11.2 Verify Newsgate services | 13 |
| 3.11.3 Verify servlets | 13 |
| 4 Servlet Reference | 14 |
| 4.1 ZipImportServlet | 14 |

| | |
|--|----|
| 4.2 ZipPreviewServlet | 14 |
| 4.3 DevicePreviewServlet | 14 |
| 4.4 TaxonomyServlet | 15 |
| 4.5 DetailsServlet | 15 |

1 Introduction

The Newsgate plug-in enables two-way integration between CUE and CCI Newsgate.

CUE can be launched from CCI Newsgate. The user needs to perform a log in when CUE is launched. CUE only needs to be launched once per session.

A story created in CCI Newsgate can be opened in CUE. If the story does not exist in the CUE Content Store, it will be created automatically. When the article is stored in CUE, the story will be updated in CCI Newsgate. Related content such as related images, related media and related articles will also be updated in CCI Newsgate.

The plug-in also offers a set of services for exchanging content and meta data between the CUE Content Store and CCI

URL resolver

A URL resolver implemented to support CCI URL resolver functionality

Article details

A web service for retrieving additional information for a given article in CUE.

Export of content types

A service for exporting CUE content types to CCI.

Preview

A service for providing an online preview of content sent from CCI.

Device Preview

A service for providing a multi-platform online preview of content sent from CCI.

Export publication structure

A service for exporting the publication structure to CCI.

Importing taxonomy definitions

A web service for importing taxonomy definitions from CCI.

2 Overview

2.1 What is Newsgate plug-in

Newsgate is a CUE Content Store plug-in which adds the ability for two-way integration with CCI Newsgate.

2.2 What purpose does Newsgate serve?

Newsgate makes it easy for a content creator to create stories in CCI Newsgate and to export these stories as articles to CUE.

If the article is modified in CUE the changes are updated automatically in CCI Newsgate.

Newsgate makes it possible for a content creator to preview a story folder from CCI Newsgate without first opening CUE.

All data exchange between the CUE Content Store and CCI Newsgate is performed automatically. This integration has no impact on how articles and other content objects are created or edited in CUE.

Newsgate also offers a set of services for easy exchange of meta data between the CUE Content Store and CCI Newsgate.

2.3 How does the Newsgate plug-in work?

The Newsgate plug-in is divided into two separate components:

- A Content Store plug-in
- A web application containing web services for importing, exporting, previewing stories packaged as zip files. CUE uses the web service when exchanging content with CCI Newsgate.

The web application also contains web services for retrieving article details, importing taxonomies and URL resolving.

For the rest of this document this web application is named **newsgate-webservice.war**

2.3.1 CUE integration

Content in CCI Newsgate is exported to CUE as regular zip files. One zip file contains information about one story. The content of the zip file should be

- An XML file in CUE XML format. The XML file must contain all information about the article and its related content.

The name of the XML file must correspond to the name of the zip file. For instance, if the name of the zip file is **myZipFile**, the name of the XML file must be **myZipFile.xml**.

- If the article has related images and media objects the zip file must contain the actual media objects.

CCI Newsgate is responsible for creating a unique name of the zip file.

CCI Newsgate and CUE communicate using regular socket connections. CCI Newsgate is responsible for finding a port where CCI Newsgate and CUE can communicate.

2.3.2 URLResolver

The URL resolver is a web service adding support for CCI URL resolver functionality. It accepts CUE-specific URLs and is responsible for retrieving the actual CUE object and transforming it to XML.

This web service is included in **newsletter.war** and is available using the following URL: *url-to-newsgate.war/resolve*.

Required request parameter is

url

The Content Store specific URL. The URL schema is *baseURL/objecttype/objectid*.

A sample URL could be **http://resolve.escenic.com/comment/123**.

The web service will resolve the URL and return a XML representation of the requested object.

Currently this service only supports exporting article comments.

2.3.3 Article details

Article details is a web service for retrieving additional information for an article. Such information could for instance be the number of comments or the hit count.

The web service is included in **newsletter.war** and can be invoked by invoking *url-to-newsgate.war/details*.

Required request attributes are:

dbid

The internal Content Store identifier of the object.

source

Identifies the external source of the object.

sourceid

The external identifier of this object. This id must be unique for every article from a given source.

Either **dbid** or a combination of **source** and **sourceid** must be specified.

When invoked the web service resolves the requested article and returns an HTML page with additional article information.

If the article is not published, the web service will return an **HTTP 404** error message.

2.3.4 Exporting content types

All CUE publications have a publication resource named `/escenic/content-type`. This resource contains information regarding all content types available in a publication.

When the resource is changed, the service will automatically post the changed resource to a CCI soap web service.

This service is by default disabled. See chapter for information on how to enable this service.

2.3.5 Exporting section structure

When the section structure of a publication changes, for instance if a section is created, modified or deleted, the service will automatically post the changed section structure to a CCI soap web service.

This service is by default disabled. See chapter for information on how to enable this service.

2.3.6 Importing taxonomy definitions

In CUE, taxonomy definitions are imported and stored as categories. The master definition is maintained by CCI. When one or more taxonomy nodes are changed in CCI, the changes are imported into the CUE Content Store. CCI is responsible for pushing the changed nodes to the CUE Content Store.

The web service is included in `newsgate-webservice.war`. It can be invoked by invoking `url-to-newsgate.war/taxonomy`. The request should be of type `text/xml` and should contain a CCI taxonomy definition. The XML will be transformed to CUE category import XML and imported into the CUE Content Store.

3 Installation

The following preconditions must be met before you can install Newsgate 3.1.1-2:

- The Content Store and assembly tool have been installed as described in the **CUE Content Store Installation Guide** and are in working order.
- You have the required plug-in distribution file `newsgate-dist-3.1.1-2.zip`.

3.1 Conventions

The instructions in the following section assume that you have a standard Content Store installation, as described in the **CUE Content Store Installation Guide** for releases 5.1-1 and above. *escenic-home* is used to refer to the `/opt/escenic` folder under which both the Content Store itself and all plug-ins are installed).

The Content Engine and the software it depends on may be installed on one or several host machines depending on the type of installation required. In order to unambiguously identify the machines on which various installation actions must be carried out, the **CUE Content Store Installation Guide** defines a set of special host names that are used throughout the manual.

Some of these names are also used here:

assembly-host

The machine used to assemble the various Content Engine components into a enterprise archive or .EAR file.

engine-host

The machine(s) used to host application servers and Content Engine instances.

The host names always appear in a bold typeface. If you are installing everything on one host you can, of course, ignore them: you can just do everything on the same machine. If you are creating a larger multi-host installation, then they should help ensure that you do things in the right places.

3.2 Install Newsgate

To install Newsgate on an Ubuntu or other Debian-based system, do the following on your **assembly-host** and on each of your **engine-hosts**:

1. Log in as **root**.
2. If necessary, add the CUE **apt** repository to your list of sources:

```
# echo "deb http://user:password@apt.escenic.com stable main non-free" >> /etc/
apt/sources.list.d/escenic.list
```

where *user* and *password* are your CUE download credentials (the same ones you use to access the CUE Maven repository). If you do not have any download credentials, please contact [CUE support](#).

3. Enter the following commands:


```
# apt-get update
# apt-get install escenic-newsgate
```

On RedHat / CentOS systems, enter the following command as **root** on your **assembly-host** and each of your **engine-hosts**:

```
# rpm -Uvh https://user:password:yum.escenic.com/rpm/escenic-
newsgate-3.1.1-2.x86_64.rpm
```

3.3 Verify The Installation

To verify the status of the Newsgate plug-in, open the **escenic-admin** web application (usually located at <http://server/escenic-admin>) and click on **View installed plugins**. The status of all currently installed plug-ins is shown here.

You now know that the Newsgate plug-in is installed. Next step is to do some required configuration changes

3.4 Configure XSL style sheet

The import web service supports both **HTTP GET** and **HTTP POST**. The **GET** operation has support for transforming the XML using an XSL style sheet.

The location of the style sheet is configured by adding the **XSL** property to **/com/escenic/newsgate/Zipper**. The value of this property must be the absolute path of the style sheet.

Example:

```
xsl=/opt/escenic/xsl/myXsl.xsl
```

The configured style sheet is used before the zip file is created.

3.5 Configure doctype

To be able to deliver valid XML documents, it might be necessary adding a DTD to the XML file. This is only supported on **HTTP GET**.

The location of the style sheet is configured by adding the **doctype** property to **/com/escenic/newsgate/Zipper**. The value of this property will be added to the **doctype** statement in the XML file.

Example:

```
docType=file://localhost/usr/local/engine/dtd/system.dtd
docType=<!DOCTYPE io [
<!ENTITY % xhtml-lat1 PUBLIC "-//W3C//ENTITIES Latin 1 for XHTML//EN"
"http://www.w3.org/TR/xhtml-modularization/DTD/xhtml-lat1.ent">
<!ENTITY % xhtml-special PUBLIC "-//W3C//ENTITIES Special for XHTML//EN"
"http://www.w3.org/TR/xhtml-modularization/DTD/xhtml-special.ent">
%xhtml-lat1;
```

```
%xhtml-special;
]>
```

3.6 Publication mapping

To be able to export content types and section structure to CCI, a mapping between CUE Content Store publications and CCI newsrooms must be configured.

3.6.1 Create a publication mapping

A sample file for mapping a CUE Content Store publication and a CCI newsroom can be found in **/com/escenic/newsgate/publication/SamplePublication**.

```
$class=com.escenic.newsgate.config.PublicationConfiguration
publicationName=My news site
newsroom=My newsroom
productName=Online publication
importConfiguration=/import/MyNewsSiteImportConfiguration
```

publicationName

The name of the publication in CUE

newsroom

The name of the CCI newsroom

productName

The name of the product in CCI

importConfiguration

Optional. Defines an import configuration to use when importing to this publication. If not defined, a default configuration will automatically be created.

A similar mapping file must be created for all publications.

3.6.2 Register the mapping

To register a publication mapping, the property named **publications** in **/com/escenic/newsgate/publication/PublicationConfigurationLoader** must be edited.

Let's say that in the previous chapter you created two publication mappings:

- **MyFirstPublication**
- **MySecondPublication**

both located in **/com/escenic/newsgate/publication/**. To register these to mappings the property must be set to .

```
publications=./MyFirstPublication,./MySecondPublication
```

3.7 Enabling export of content types

3.7.1 Enable service

This service is default disabled. Enabling the service is done by adding two new properties to `/com/escenic/newsgate/service/ContentTypeExporter`

serviceEnabled

Enables the service

endpoint

The URL of a SOAP web service

The service can be enabled on all servers, but it will only run on one server at a time. If the server running the service dies, another server will take over the responsibility

When `/escenic/content-type` is changed, the new resource will be posted to the configured URL.

This service requires a properly configured publication mapping.

3.7.2 Configure XSL style sheet

The service supports transforming the XML using an XSL stylesheet.

The location of the style sheet is configured by setting the `xsl` property in `/com/escenic/newsgate/service/ContentTypeExporter`. The value of this property must be the absolute path of the style sheet.

Example:

```
xsl=/opt/escenic/xsl/myXsl.xsl
```

The configured style sheet is used before the resource is posted.

3.8 Enabling export of section structure

3.8.1 Enable service

This service is default disabled. Enabling the service is done by adding two new properties to `/com/escenic/newsgate/service/PublicationExporter`

serviceEnabled

Enables the service

endpoint

The URL of a SOAP web service

The service can be enabled on all servers, but it will only run on one server at a time. If the server running the service dies, another server will take over the responsibility

When enabled, the section structure will be posted to the `endpoint` each time a section is created, modified or deleted.

This service requires a properly configured publication mapping.

3.8.2 Configure XSL style sheet

The service supports transforming the XML using an XSL stylesheet.

The location of the style sheet is configured by setting the `xsl` property in `/com/escenic/newsgate/service/PublicationExporter`. The value of this property must be the absolute path of the style sheet.

Example:

```
| xsl=/opt/escenic/xsl/myXsl.xsl
```

The configured style sheet is used before the resource is posted.

3.9 Configure export

This step is only required if you have enabled continues export of content from a CUE publication and this content is fed back into CCI Newsgate.

Open the export configuration file in your favorite editor and change

```
| exportWriter=./StandardExportWriter
```

to

```
| exportWriter=/com/escenic/newsgate/syndication/NewsgateFileExportWriter
```

For a detail description of how to configure continues export of content, please see the **CUE Content Store Syndication Reference**.

3.10 Configuration in CCI

For the plug-in to work properly, some additional configuration needs to be done in CCI Newsgate. Please refer to CCI Newsgate documentation for information about how to configure CCI Newsgate to work with this plug-in.

3.11 Troubleshooting

This chapter contains tips when troubleshooting a non-working Newsgate plug-in.

If these tips don't solve the problem, please contact support@escenic.com.

3.11.1 Verify plug-in installation

Open a browser and go to `http://host/escenic-admin/pages/plugin/list.jsp`.

This page lists all installed plug-ins. If `newsgate` is not listed here, then the plug-in is not installed.

3.11.2 Verify Newsgate services

Open Component browser from `http://host/escenic-admin`.

Verify that the following services is running normally:

1. `/com/escenic/newsgate/Zipper`

If you have enabled export of content types and section structure you should also verify that

1. `/com/escenic/newsgate/service/ContentTypeExporter`
2. `/com/escenic/newsgate/service/PublicationExporter`

is running normally.

3.11.3 Verify servlets

This plug-in contains two web services. Both should be verified working.

3.11.3.1 Verify import servlet - GET

Open a browser and go to `http://host/newsgate/import?articleId=some-articleid`. If working, this servlet should return a zip file containing the content of the given article. Please verify the content of this zip file.

3.11.3.2 Verify import servlet - POST

This step requires that you already have a zip file which can be imported into the CUE Content Store and that you have a tool for performing a **HTTP POST**.

Post a zip file containing CUE content to `http://host/newsgate/import`. If working, the response should contain the id of the imported article.

3.11.3.3 Verify preview servlet

This step requires that you already have a zip file which can be imported into the CUE Content Store and that you have a tool for performing a **HTTP POST**.

Post a zip file containing CUE content to `http://host/newsgate/preview`. If working, the response should be redirected to the preview URL of the article.

4 Servlet Reference

This chapter gives an overview of the web services in this plug-in

4.1 ZipImportServlet

A servlet for importing zip files. Supports both **POST** and **GET**:

POST: `http://host/import/publication-name`

This **POST** operation requires that the incoming request is a ZIP input stream. The content of the stream is extracted and stored on a temporary location on disk. When all content is extracted, the files are imported either using the import configuration configured for the given publication or the web service will create a default import configuration.

GET

This **GET** operation supports two request parameters: **articleId** and **filename**. **articleId** is required and is the id of the article to be exported. **filename** is not required, but if not defined, the name of the XML file will be the articleId. The response of a get operation will be a zip stream containing the content of the given article. This includes the XML and all media files attached to the article.

4.2 ZipPreviewServlet

A servlet for previewing content of zip files. Supports only **POST**:

POST: `http://host/preview/publication-name`

This **POST** operation requires that the incoming request is a ZIP input stream. The content of the stream is extracted and stored on a temporary location on disk. When all content is extracted, the files are imported either using the import configuration configured for the given publication or the web service will create a default import configuration. When finished the request is redirected to a preview URL for the article.

4.3 DevicePreviewServlet

A servlet for providing multiplatform previews of zip file content. Supports only **POST**:

POST: `http://host/device-preview/publication-name`

This **POST** operation requires that the incoming request is a ZIP input stream. The content of the stream is extracted and stored on a temporary location on disk. When all content is extracted, the files are imported either using the import configuration configured for the given publication or the web service will create a default import configuration. When finished the request is redirected to a multiplatform preview URL for the article, allowing the end user to see what the ZIP file content will look like on a range of different devices.

4.4 TaxonomyServlet

A servlet for importing taxonomy definition from CCI. Supports only **POST**:

POST: `http://host/taxonomy`

This **POST** operation requires that the incoming request is of type `text/xml` and conforms to `CCITaxonomyObjectXML.xsd`. The incoming XML will be transformed to CUE Category format and imported as categories using the configured `ImportHandler`.

4.5 DetailsServlet

A servlet for retrieving additional information for an article. Supports only **GET**:

GET: `http://host/details?dbid=database-id&source=object-source&sourceid=object-source-id`

dbid

The internal Content Store identifier of the object.

source

Identifies the external source of the object.

sourceid

The external identifier of this object. This id must be unique for every article from a given source.

Either **dbid** or a combination of **source** and **sourceid** must be specified.

When invoked the servlet resolves the requested article and returns an HTML page with additional article information.

If the article is not published, the servlet will return a **HTTP 404** error message.