Newsgate

Plug-in Guide

2.4.2.167332

escenic
A CCI COMPANY

Table of Contents

1 Introduction

The Newsgate plug-in enables two-way integration between Escenic Content Studio and CCI Newsgate.

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

A story created in CCI Newsgate can be opened in Escenic Content Studio. If the story does not exist in Escenic, the story will be created automatically. When the article is stored in Escenic Content Studio, 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.

Newsgate story folders can be previewed in Escenic without opening Escenic Content Studio.

The plug-in also offers a set of services for exchanging content and meta data between Escenic Content Engine 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 Escenic.

Export of content types
A service for exporting Escenic 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 an Escenic 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 Escenic Content Studio.

If the article is modified in Escenic Content Studio the changes are updated automatically in CCI Newsgate.

Newsgate makes it possible for a content creator to preview a story folder from CCI Newsgate in Escenic without first opening Escenic Content Studio.

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

Newsgate also offers a set of services for easy exchange of meta data between Escenic Content Engine and CCI Newsgate.

2.3 How does the Newsgate plug-in work?

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

- An Escenic Content Studio plug-in
- A web application containing web services for importing, exporting, previewing stories packes as zip files. Escenic Content Studio 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 Content Studio integration

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

- An XML file in Escenic 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 Escenic Content Studio communicate using regular socket connections. CCI Newsgate is responsible for finding a port where CCI Newsgate and Escenic Content Studio can communicate.

2.3.1.1 Opening a story from CCI Newsgate in Escenic Content Studio

CCI Newsgate creates a zip file with all the content of a story. Escenic Content Studio is launched using Java Webstart. Information about the port, filename, web service URL and relaunch port is sent as parameters in the URL. The format of this parameter is

```
newsgate.start=port;filename webservice-url;relaunch-port
```

webservice-url and *relaunch-port* are optional parameters.

If the story does not exist in Escenic, the story will be imported as an Escenic article.

Escenic Content Studio will be launched with the new article opened. This article can be modified just like any other article in Escenic.

When the article is saved, the new content of the article is stored in the zip file given. CCI Newsgate is notified each time the zip file is modified and the story is updated in CCI Newsgate.

When the article is closed, CCI Newsgate will be notified and the socket will be closed.

While opened in Escenic Content Studio the story will be locked in CCI Newsgate.

2.3.1.2 Previewing a story in Escenic

CCI Newsgate creates a zip file with all the content of the story and posts the file to a preview web service provided by Escenic.

The web service creates a new preview article based on the content of the zip file and the request is redirected to a preview URL for the article.

2.3.2 URLResolver

The URL resolver is an Escenic web service adding support for CCI URL resolver functionality. It accepts Escenic specific URLs and is responsible for retrieving the actual Escenic 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 Escenic 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 Escenic (database) 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 Escenic 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 Escenic 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 Escenic. CCI is responsible for pushing the changed nodes to Escenic.

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 Escenic category import XML and imported into Escenic.

3 Installation

The following preconditions must be met before you can install Newsgate 2.4.2.167332:

- The Content Engine and Escenic assembly tool have been installed as described in the **Escenic Content Engine Installation Guide** and are in working order.
- The Content Engine's **Forum** plug-in has been installed. The Newsgate plug-in depends on the Forum plug-in.
- You have the required plug-in distribution file **newsgate-dist-2.4.2.167332.zip**.

3.1 Conventions

The instructions in the following section assume that you have a standard Content Engine installation, as described in the **Escenic Content Engine 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 Engine 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 **Escenic Content Engine 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

Installing Newsgate involves the following steps:

1. Log in as **escenic** on your **assembly-host**.
2. Download the Newsgate distribution from the Escenic Technet web site (http://technet.escenic.com). If you have a multi-host installation with shared folders as described in the **Escenic Content Engine Installation Guide**, then it is a good idea to download the distribution to your shared **/mnt/download** folder:

```
$ cd /mnt/download
$ wget http://user:password@technet.escenic.com/downloads/PATH??/newsgate-
dist-2.4.2.167332.zip
```

Otherwise, download it to some temporary location of your choice.

3. If the folder **/opt/escenic/plugins** does not already exist, create it:

   ```
   $ mkdir /opt/escenic/plugins
   ```

4. Unpack the Newsgate distribution file:

   ```
   $ cd /opt/escenic/plugins
   $ unzip /mnt/download/newsgate-dist-2.4.2.167332.zip
   ```

   This will result in the creation of an **/opt/escenic/plugins/newsgate** folder.

5. Copy **newsgate-webservice.war** to the assembly tool's **publications** folder:

   ```
   $ cp /opt/escenic/plugins/newsgate/wars/newsgate-webservice.war /opt/escenic/
   assemblytool/publications/
   ```

6. Create a text file called **newsgate-webservice.properties** in the **/opt/escenic/assemblytool/publications/** folder, with the following contents:

   ```
   name:           newsgate-webservice
   source-war:     newsgate-webservice.war
   context-root:   /newsgate-webservice
   ```

   > If you have a multi-host installation, then repeat this two steps on each **engine-host** .

7. Log in as **escenic** on your **assembly-host**.

8. Run the **ece** script to re-assemble your Content Engine applications

   ```
   $ ece assemble
   ```

   This generates an EAR file (**/var/cache/escenic/engine.ear**) that you can deploy on all your **engine-host**s.

9. > If you have a single-host installation, then skip this step.

   On each **engine-host**, copy **/var/cache/escenic/engine.ear** from the **assembly-host**. If you have installed an SSH server on the **assembly-host** and SSH clients on your **engine-host**s, then you can do this as follows:

   ```
   $ scp -r escenic@assembly-host-ip-address:/var/cache/escenic/engine.ear /var/
   cache/escenic/
   ```

   where *assembly-host-ip-address* is the host name or IP address of your **assembly-host**.

10. On each **engine-host**, deploy the EAR file and restart the Content Engine by entering:

    ```
    $ ece deploy
      $ ece restart
    ```

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 Configuring extra information

It is possible to configure the plug-in to display more information from CCI in Escenic Content Studio. This section describes how this can be done. You will need to complete three steps.

1. CCI Newsgate must be configured to make the extra information available as a web page accessible via HTTP.
2. Escenic Content Engine must be configured with a **`DefaultLinkHeaderPlugin`** component that accesses the above service and makes the information available to Content Studio for display.
3. Escenic Content Studio must be configured to display this information.

3.6.1 Configuring CCI Newsgate

How this is done will depend on your specific installation and configuration of CCI Newsgate, but the basic information you need will be outlined here. You will need a web server that can produce a HTML page of some sort given a few request parameters. Make a note of the server's URI and the parameters that must be supplied.

3.6.2 Configuring the Content Engine

You now need to configure a Content Engine **link header plug-in** component that can access the CCI Newsgate web server you configured in the previous step. For general information about link header plug-ins and how to configure them, see the **Escenic Content Engine Advanced Developer Guide**.

You can base your component on the default link header plug-in supplied with the Content Engine. To do this you must create a **`.properties`** file in one of your configuration layers. You can call it anything you like, and put it anywhere you like in the configuration layer. You might, for example, call it *configuration-root***`/com/escenic/newsgate/publication/NewsgateInfo.properties`**, and give it the following contents:

```
$class=com.escenic.webservice.spi.DefaultLinkHeaderPlugin
vendor=myCompany.com
version=1.0-1
description=Retrieves CCI Newsgate info for display in Content Studio

relation=CCI-Newsgate
mimeType=text/html
title=CCI Newsgate sample
uriTemplate=http://newsgate.myCompany.com/{content-type}/{source}/{sourceid}
objectLoader=/io/api/ObjectLoader
```

All the properties shown in the above listing must be specified, but the values are just examples. For a description of the properties and advice on how to set them, see the **Escenic Content Engine Advanced Developer Guide**.

When you have done this, you must also add *a configuration-root***`/com/escenic/service/LinkHeaderManager.properties`** file to the same configuration layer, and define a **`service`** property for your new component. For example:

```
$class=com.escenic.webservice.manager.LinkHeaderPluginManager

service.myProvider-1=/com/escenic/newsgate/publication/NewsgateInfo
```

For additional information about this, see the **Escenic Content Engine Advanced Developer Guide**.

3.6.2.1 Configuring Authentication

You can configure the Content Engine to use CCI Newsgate as an authentication server, so that CCI Newsgate users don't need to re-authenticate in order to access Escenic functionality and content. To do this, add a *configuration-root***`/com/escenic/auth/ng/NGLoginModuleFactory.properties`** file to the same configuration layer as the other configuration files you have created, and define an **`endpoint`** property. For example:

```
$class=com.escenic.auth.ng.NGLoginModuleFactory

endpoint=cci-authenticator-url
```

where *cci-authenticator-url* is the URL of the CCI Newsgate authentication web service.

3.6.3 Configuring Content Studio

For the information provided to be displayed in Content Studio you need to add a property to **`com/escenic/webstart/StudioConfig.properties`**. For example:

```
property.com.escenic.studio.browserbox.linkheaders=CCI-Newsgate
```

Note that the value specified for this property (**`CCI-Newsgate`**) must match the **`relation`** property specified in your component configuration file. This name will also be used to identify the **research panel** in which the information is displayed in Content Studio. For further information about this, see the **Escenic Content Engine Advanced Developer Guide**.

Finally, you need to log into Content Studio and set it up to display the correct **research panel**. For details, see the **Escenic Content Studio User Guide**.

3.7 Configure Escenic Content Studio

Escenic Content Studio needs to know the URL of the import web service. This is configured in **`/com/escenic/webstart/StudioConfig.properties`** in one of your configuration layers.

The URL of the web service is normally **`http://myserver/newsgate/import`** where **`myserver`** is the public address of the server where **`newsgate-webservice.war`** is deployed. If the name of the war-file is changed, the URL should reflect this.

Example from StudioConfig.properties:

```
property.newsgate.webservice.url=http://localhost:8080/newsgate/import
```

3.8 Publication mapping

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

3.8.1 Create a publication mapping

A sample file for mapping an Escenic 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 Escenic

`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.

3.8.2 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
```

3.9 Enabling export of content types

3.9.1 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.

3.9.2 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.

3.10 Enabling export of section structure

3.10.1 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.

3.10.2 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.

3.11 Configure the URLResolver

The URLResolver has proper default values and should work properly without any additional configuration. However, it is possible to tweak the behaviour of this component by editing the properties in **`/com/escenic/newsgate/PostingResolver.properties`**. Available configuration options are:

`docType`
If defined, this doctype will be added to the XML.

xsl
If defined, the XML will be transformed using this stylesheet.

3.12 Configure export

This step is only required if you have enabled continues export of content from an Escenic 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 **Escenic Content Engine Syndication Reference**

3.13 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.

3.14 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.

3.14.1 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.

3.14.2 Verify Newsgate services

Open a browser and go to **http://***host***/escenic-admin/do/services/display**.

Verify that the following services is running normally:

1. /com/escenic/newsgate/PostingResolver
2. /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.

3.14.3 Verify servlets

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

3.14.3.1 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.

3.14.3.2 Verify import servlet - POST

This step requires that you already have a zip file which can be imported in Escenic and that you have a tool for performing a **HTTP POST**.

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

3.14.3.3 Verify preview servlet

This step requires that you already have a zip file which can be imported in Escenic and that you have a tool for performing a **HTTP POST**.

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

3.14.4 Verify Content Studio parameters

To be able to import and open an article from Newsgate in Content Studio some system properties must be sent to Content Studio.

To be able to verify the system properties the Java console must be enabled in Java Web Start. If not enabled, launch **`javaws`** from a command prompt and enable Java console.

When launching Escenic Content Studio, the Java console will open as a new window. Pressing **s** when the Java console is active will print a list of all system properties defined. Please verify that these properties are set:

- **`newsgate.start`**
- **`newsgate.webservice.url`**

If one or more is not set the plug-in will not work.

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 multi-platform 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 Escenic 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 Escenic (database) 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.