

Geocode  
**Plug-in Guide**  
2.8.0.148641

# Table of Contents

<a href="#">1 Introduction</a> .....	3
<a href="#">1.1 Using The Geocode Plug-in</a> .....	3
<a href="#">2 Installation</a> .....	5
<a href="#">2.1 Install Geocode</a> .....	5
<a href="#">2.1.1 Configuration</a> .....	6
<a href="#">2.2 Verify The Installation</a> .....	9
<a href="#">3 Using the Geocode Field in Templates</a> .....	10
<a href="#">3.1 Form Properties</a> .....	10
<a href="#">4 Writing your own map service</a> .....	12

# 1 Introduction

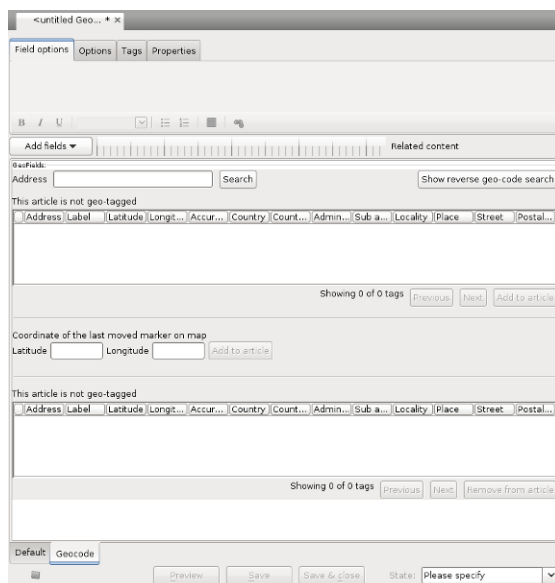
The Geocode plug-in adds easy-to-use geocoding functionality to Escenic Content Engine installations. It allows geographical location records (called **geotags**) to be associated with content items.

With the Geocode plug-in, journalists and editors working in Content Studio are able to easily add geotags to a content item, which are then stored along with the content item in the Content Engine database. Once these tags are stored with content items, they can subsequently be used in a variety of ways:

- To search for items in Content Studio by location: either address (city, state, country) or coordinates
- To plot locations on maps
- To create user services that search for news by location

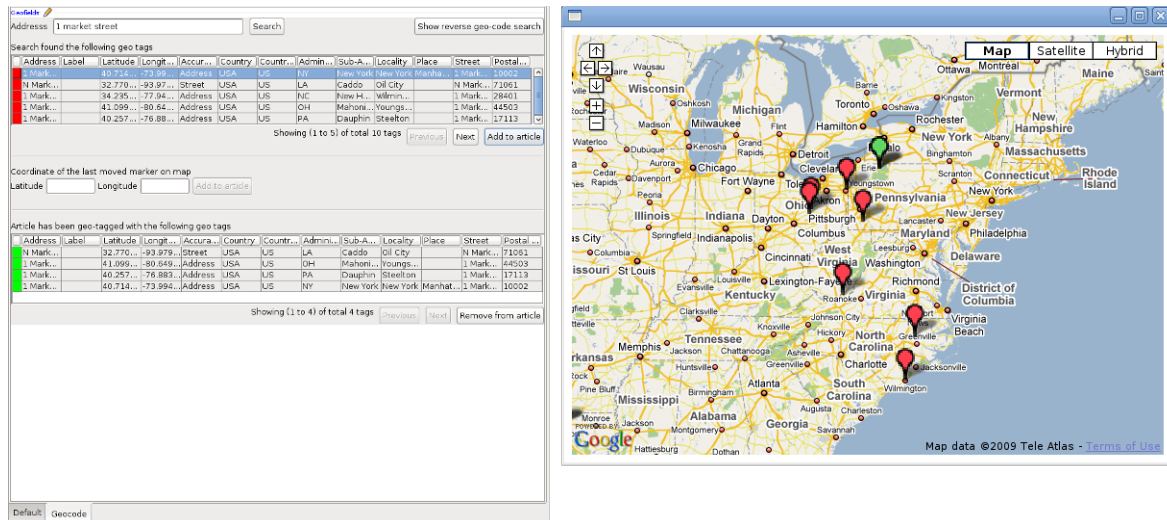
## 1.1 Using The Geocode Plug-in

When a user creates a "geocodable" content item in Content Studio, the Geocode plug-in is activated, enabling the display of geotags. The geocode field editor of a content item that does not yet have any geotags looks like this:



To add a geotag to a content item you can enter an address in the **Address** field and click on **Search**, in much the same way as if you were looking for an address in an online mapping service. In fact your query is simply passed on to an underlying mapping service such as Google Maps, and the results

of the search are displayed in the table immediately below the **Address** field. The results are also displayed in a map browser window as green or red pins.



The screenshot shows the Geocode Plug-in interface. On the left, there is a search bar with the text "1 market street" and a "Search" button. Below the search bar, there are two tables displaying search results. The top table shows results for "1 Market Street" in New York, NY, with a green pin. The bottom table shows results for "1 Market Street" in New York, NY, with a red pin. On the right, there is a map browser window showing a map of the United States with several pins. A green pin is located in New York, NY, and several red pins are located in other parts of the country, including New York, NY, Pennsylvania, and Virginia.

Address	Label	Latitude	Longitude	Accuracy	Country	County	Admin...	Sub-A...	Locality	Place	Street	Postal...
1 Market...		40.714...	-73.99...		USA	US	NY	New York	New York	Manhat...	1 Market...	10002
1 Market...		32.770...	-99.97...		USA	US	LA	Cadodo	Oil City		1 Market...	71061
1 Market...		34.295...	-77.98...		USA	US	DC	New H...	Wilmin...		1 Market...	20001
1 Market...		41.099...	-80.64...		USA	US	OH	Mahoni...	Youngs...		1 Market...	44503
1 Market...		40.257...	-76.88...		USA	US	PA	Dauphin	Steelton		1 Market...	17113

Address	Label	Latitude	Longitude	Accuracy	Country	County	Admin...	Sub-A...	Locality	Place	Street	Postal...
1 Market...		32.770...	-99.979...		USA	US	LA	Cadodo	Oil City		1 Market...	71061
1 Market...		41.099...	-80.649...		USA	US	OH	Mahoni...	Youngs...		1 Market...	44503
1 Market...		40.257...	-76.883...		USA	US	PA	Dauphin	Steelton		1 Market...	17113
1 Market...		40.714...	-73.996...		USA	US	NY	New York	New York	Manhat...	1 Market...	10002

A green pin on the map represents a location that matches the search but is not associated with the current content item. A red pin represents a location that matches the search and is associated with the current content item. These locations are also listed in the table at the bottom of the geocode field editor.

To add a geotag to the current content item, simply select one of the locations in the top table and click on the **Add to article** button. To remove a geotag, select one of the locations in the bottom table and click on the **Remove from article** button. Note that selecting a location in either of the tables will cause the map browser to zoom in on the corresponding pin.

If you have several geocoded content items open in Content Studio, then then the geocode field editor displays the tags associated with the currently selected content item. Once you select the geocode field editor, the map browser is also updated to display the same set of tags.

You can also search for geotags by coordinate using the **Show reverse geo-code search** function. You enter a latitude, a longitude and a radius, and all the geotags in the circle defined by these numbers are displayed in the results table.

## 2 Installation

The following preconditions must be met before you can install Geocode 2.8.0.148641:

- The Content Engine is installed and in working order.
- The Escenic assembly tool has been extracted and successfully used to set up a test EAR file as described in the **Escenic Content Engine Installation Guide**.
- The Escenic Lucy plug-in is installed and in working order.
- You have the correct distribution file (**geocode-dist-2.8.0.148641.zip**).
- If you intend to use the default service provider implementation included with the Geocode plug-in then you need a Google Map API key to be able to access Google Maps. To get a key you need to register as a user of the API with Google.

Plug-ins must be installed using the Escenic assembly tool.

### 2.1 Install Geocode

In the following description, *escenic-home* refers to the server folder in which the Content Engine is installed.

Installing Lucy on the server involves the following steps:

1. **Make sure there is a plug-in folder:** If the folder *escenic-home/plugins* does not already exist on your server, create it. If for some reason you need to create the plug-in folder in some other location, edit the *escenic-home/assemblytool/assemble.properties* file and set the **plugins** property accordingly. For example:

```
plugins=escenic-home/my/plugin/folder
```

This folder will be referred to as *plugin-home* in the rest of this manual.

2. **Unpack the Geocode distribution:** Unpack the Geocode distribution file to *plugin-home*. This will result in the creation of a *plugin-home/geocode* folder.
3. **Configure Geocode:** See [section 2.1.1](#) for details.
4. **Rebuild and deploy the Content Engine:** Build the Escenic enterprise archive by entering the following commands:

```
cd escenic-home/assemblytool
ant ear
```

The assembly tool will then add the Geocode plug-in to the Content Engine's classpath, including default configuration files and any required web application components. Deploy the new EAR file. (For general instructions on how to deploy the EAR file on different application servers, see the **Escenic Content Engine Installation Guide**.)

5. **Verify the plug-in installation:** See [section 2.2](#) for details of how to verify plug-in installations.

If the application server does not support EAR-based deployment, then all the JAR files located in the `plugin-home/geocode/lib` folder must be added to the application server's classpath. All the WAR files that have been rebuilt with the assembly tool should be redeployed.

## 2.1.1 Configuration

The following sections describe how to configure the Geocode plug-in. In these instructions, the placeholder `escenic-config` is used to represent the path of your Escenic configuration, as defined with the `com.escenic.config` property in `escenic-home/assemblytool/assemble.properties`. If `com.escenic.config` is not defined, then `escenic-config` has a default definition of `escenic-home/localconfig`.

### 2.1.1.1 Configure the Solr search engine

Geocode requires some geocode-specific fields to be indexed by the Solr search engine. In order to ensure that this is done, you must:

1. Open the Solr `schema.xml` file for editing. This file is usually located in the `${solr.solr.home}/conf` folder.
2. Add the following element as a child of the `fields` element:

```
<dynamicField name="geocode.*" type="string" indexed="true" stored="true"
  multiValued="true"/>
```

3. Save the file and exit.

### 2.1.1.2 Set the Lucy SearchEngine Properties

By default, Lucy (which is used by the Geocode plug-in) looks for its search engine at `http://localhost:8080/solr`, which is the location of the default Lucy search engine, called `LucySearchEngine`. If this is not the Solr engine you want Lucy to use, then you will need to:

1. Copy `SearchEngine.properties` from `escenic-home/engine/plugins/geocode/misc/siteconfig/com/escenic/geocode/search/lucy/SearchEngine.properties` to `escenic-config/com/escenic/geocode/search/lucy/SearchEngine.properties`.
2. Open the copied file for editing.
3. Set the `solrURI` property to point to the correct Solr engine location. For example:

```
solrURI=http://mysolrhost:8080/solr
```

4. Save the file and exit.

### 2.1.1.3 Configure the plug-in to use a Proxy Server

Content Studio uses whatever proxy is defined for the browser used to start it, and Geocode uses the same proxy (if any).

### 2.1.1.4 Add geocode fields to content types

In order to make content items "geocodable" you must add a special geocode field to their content type definitions in the `content-type` resource. The following examples shows what such field definitions should look like. The exact content of the field definitions depends on which map provider you use. For Google Maps, your field definition should look something like this:

```
<content-types xmlns="http://xmlns.escenic.com/2008/content-type"
```

```

        xmlns:geocode="http://xmlns.escenic.com/2009/studio/plugin/geocode"
        xmlns:ui="http://xmlns.escenic.com/2008/interface-hints"
        version="4">
<!-- Please note the geocode namespace declaration above -->
...
<content-type name="restaurant-review">
...
<panel name="geocode">
...
<field type="basic" name="com.escenic.geocode" mime-type="text/plain">
  <geocode:geocode-editor enabled="true">
    <geocode:google-api>
      <geocode:client-id><!-- Google Maps API for Business Client ID --></
geocode:client-id>
      <geocode:api-key><!-- Google Maps API Key --></geocode:api-key>
      </geocode:google-api>
    </geocode:geocode-editor>
  </field>
</panel>
...
</content-type>
...
</content-types>

```

For OpenStreetmap on the other hand, your field definition should look something like this:

```

<content-types xmlns="http://xmlns.escenic.com/2008/content-type"
        xmlns:geocode="http://xmlns.escenic.com/2009/studio/plugin/geocode"
        xmlns:ui="http://xmlns.escenic.com/2008/interface-hints"
        version="4">
<!-- Please note the geocode namespace declaration above -->
...
<content-type name="restaurant-review">
...
<panel name="geocode">
...
<field type="basic" name="com.escenic.geocode" mime-type="text/plain">
  <geocode:geocode-editor enabled="true">
    <geocode:osm>
      <geocode:nominatim-base-uri><!-- Nominatim Base URI --></
geocode:nominatim-base-uri>
      <geocode:reverse-nominatim-base-uri><!-- Reverse Nominatim Base URI --></
geocode:reverse-nominatim-base-uri>
      <geocode:tiles-uri><!-- Tiles URI --></geocode:tiles-uri>
      <geocode:tiles-attribution><!-- Tiles Attribution --></geocode:tiles-
attribution>
      <geocode:tiles-subdomains><!-- Tiles Sub Domains --></geocode:tiles-
subdomains>
    </geocode:osm>
  </geocode:geocode-editor>
</field>
</panel>
...
</content-type>
...
</content-types>

```

You must add one such field definition to each content type that you want to be able to geocode.

Note the following:

- A geocode field must have the name **com.escenic.geocode** (with all characters in lower case).
- Geocode fields contain elements belonging to a special **geocode** XML namespace. You must therefore add a declaration for this namespace to your **content-type** resource (see the bold text in the examples above).

The **geocode** namespace elements to be included in the field definition are:

**geocode:geocode-editor**

This element specifies that the field is a geocode editor field, and contains all the other elements. The **enabled** attribute must be set to **true**.

**Either geocode:google-api or geocode:osm**

- If you want to use Google Maps as your map provider, then you must add a **geocode:google-api** element. This element must contain either a **geocode:client-id** or a **geocode:api-key** element.
- If you want to use OpenStreetMap as your map provider, then you must add a **geocode:osm** element (see <http://wiki.openstreetmap.org>). This element must contain the following elements: **geocode:nominatim-base-uri**, **geocode:reverse-nominatim-base-uri**, **geocode:tiles-uri** and **geocode:tiles-attribution**. It may also contain a **geocode:tiles-subdomains** element.

**geocode:client-id**

The value you enter here must be a Google Maps API for Business Client ID, obtained by registering with Google as a user of their mapping API. You **must** specify either **client-id** or **api-key**.

**geocode:api-key**

The value you enter here must be a Google Maps API Key, obtained by registering with Google as a user of their mapping API. You **must** specify either **client-id** or **api-key**.

**geocode:nominatim-base-uri**

The value you enter here must be the URI of a Nominatim search service, see <http://wiki.openstreetmap.org/wiki/Nominatim>. To use MapQuest Open, for example, specify the following:

```
| http://open.mapquestapi.com/nominatim/v1/search
```

**geocode:reverse-nominatim-base-uri**

The value you enter here must be the URI of a Nominatim reverse search service, see <http://wiki.openstreetmap.org/wiki/Nominatim>. To use MapQuest Open, for example, specify the following:

```
| http://open.mapquestapi.com/nominatim/v1/reverse
```

**geocode:tiles-uri**

The value you enter here must be the URI of a tiles provider (see <http://wiki.openstreetmap.org/wiki/Tiles>). To use MapQuest Open, for example, specify the following:

```
| http://{s}.mqcdn.com/tiles/1.0.0/osm/{z}/{x}/{y}.png
```

**geocode:tiles-attribution**

The value you enter here must be an attribution to OpenStreetMap and the tiles provider you are using (see [http://wiki.openstreetmap.org/wiki/Legal\\_FAQ](http://wiki.openstreetmap.org/wiki/Legal_FAQ)). To use MapQuest Open, for example, specify the following:



```
<![CDATA[Data, imagery and map information provided by <a href="http://
open.mapquest.co.uk" target="_blank">MapQuest</a>, <a href="http://
www.openstreetmap.org/" target="_blank">OpenStreetMap</a> and contributors, <a
 href="http://creativecommons.org/licenses/by-sa/2.0/" target="_blank">CC-BY-
SA</a>.] ]>
```

**geocode:tiles-subdomains**

You may use this element to specify a comma-separated list of subdomains of the tiles provider you use (see <http://wiki.openstreetmap.org/wiki/Tiles>). To use MapQuest Open, for example, specify the following:

```
otile1,otile2,otile3,otile4
```

If you have implemented your own service provider class that makes use of a different mapping service, then you can replace the **google-api** or **osm** element with other elements holding whatever identification parameters are required by your chosen service, and program your service provider class to retrieve and use the parameters as required.

For general information about the **content-type** resource and how to edit it, see the **Escenic Content Engine Resource Reference**.

## 2.2 Verify The Installation

To verify the status of the Lucy installation, open the Escenic Admin web application (usually located at <http://server/escenic-admin>) and click on **View installed plugins**. The status of the plug-ins is indicated as follows.



The plug-in is correctly installed.



The plug-in is not correctly installed.

## 3 Using the Geocode Field in Templates

To help the template developers retrieve geocoded articles, the plug-in ships with some struts Action and ActionForm.

**com.escenic.geocode.publication.struts.GeoArticleSearchAction:** the Action class that can be used to search for geocoded Articles

**com.escenic.geocode.publication.struts.GeoArticleSearchForm:** the ActionForm used by the action class mentioned above.

The form and the action class can be configured in your struts-config.xml as follows:

```
<struts-config>
  <form-beans>
    <form-bean name="com.escenic.geocode.presentation.struts.GeoArticleSearchForm"
      type="com.escenic.geocode.presentation.struts.GeoArticleSearchForm"/>
  </form-beans>
  <action-mappings>
    <action path="/geoArticleSearch"
      type="com.escenic.geocode.presentation.struts.GeoArticleSearchAction"
      name="com.escenic.geocode.presentation.struts.GeoArticleSearchForm"
      scope="request"
      validate="false"
      parameter="actionCommand">
      <forward name="success" path="/template/common.jsp"/>
      <forward name="error" path="/template/common.jsp"/>
    </action>
  </action-mappings>
  <message-resources parameter="com.escenic.geocode.Resources"/>
</struts-config>
```

### 3.1 Form Properties

A brief explanation of the form properties is as follows:

**publicationIds:** the publications in which to search for articles

**searchExpression:** the text to look for in article fields

**fromLatitude:** the start latitude against which article location is to be tested

**fromLongitude:** the start longitude against which article location is to be tested

**toLatitude:** the end latitude against which article location is to be tested

**toLongitude:** the end longitude against which article location is to be tested

**startDay:** the start day against which article date is to be tested

**startMonth:** the start month against which article date is to be tested

**startYear:** the start year against which article date is to be tested

**toDay:** the end day against which article date is to be tested

**toMonth:** the end month against which article date is to be tested

**toYear:** the end year against which article date is to be tested

**sectionIds:** the ids of the sections in which to search for articles

**searchSubSectionAllowed:** indicates whether or not the subsections of sections specified with **sectionIds** should be searched as well

**articleTypes:** restricts the search to specific article types

Please see javadoc for further information about the form properties.

An example war file is shipped with the plug-in that shows the basic functionality of the plug-in in publication templates.

## 4 Writing your own map service

Geocode plug-in allows you to write your own map services and plug them into the plug-in. The plug-in provides some java interfaces. Any class implementing them can be used with the plug-in to provide the functionality. Below is a set of interfaces and abstract classes that you can implement to provide custom map services.

### **com.escenic.geocode.studio.spi.GeocodeServiceSpi**

This abstract class declares the methods required to create **GeocodeService** and **MapView**.

### **com.escenic.geocode.studio.service.GeocodeService**

This interface declares the methods required by the plug-in to allow a user to search for geotags based on address. The interface also declares the methods required by the plug-in to enable reverse geo searching. This allows users to search for geotagged locations close to a specified latitude and longitude.

### **com.escenic.geocode.studio.mapviewer.MapViewer**

This interface defines the methods needed to display a custom map in the Geocode plug-in.

Please see the Javadoc for more information about these the interfaces and classes.