

Analysis Engine
Plug-in Guide
3.0.1-1

Table of Contents

1 Overview	3
1.1 Overview	3
2 Introduction	4
2.1 What is EAE Plugin	4
2.2 What purpose does EAE Plugin serve?	4
2.3 How does EAE Plugin work?	4
3 Installation	6
3.1 Pre-installation requirements	6
3.2 Installing EAE Plug-in	6
3.3 Configuring EAE Plug-in	6
4 Getting Started	7
4.1 Escenic Content Studio	7
5 Plug-in installation	9
5.1 Installation	9
5.1.1 Server Install	9
5.1.2 Installing Plug-in Web Applications	10
5.1.3 Installing Demo Templates	10
5.2 Verifying the installation	11
5.3 Troubleshooting	11

1 Overview

1.1 Overview

EAE Plugin connects ECE to the web statistics tool Escenic Analysis Engine. Using the ECE Plugin service API, ECE components can query the EAE and retrieve statistical information.

This document describes all aspects of EAE Plugin. Firstly, it explains how it works. The subsequent chapter describes the installation procedures. The following chapter is a tutorial on how to get started using this product. A general guide on how to install ECE plugins is provided as a chapter **Plug-in installation**.

2 Introduction

2.1 What is EAE Plugin

EAE Plugin connects Escenic Content Engine Content Engine to Escenic Analysis Engine, a web statistics tool. ECE components can then use the ECE Plugin service API to retrieve statistical information from EAE.

2.2 What purpose does EAE Plugin serve?

The purpose of the EAE Plugin is to provide ECE components with statistical information from EAE. Below is a list of ECE components using EAE Plugin and for what purpose:

Escenic Content Studio

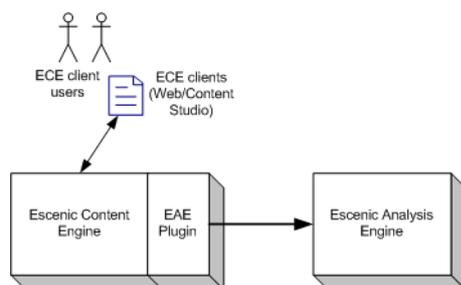
Escenic Content Studio will automatically use the EAE Plugin to connect to EAE if EAE Plugin is found. Studio will use the plugin to retrieve page view statistics for articles and sections. This information will be presented in the section editor, where Studio users can see the pageview statistics for the section and its desked articles.

EAE Query Service Client

The plug-in installs the EAE Query Service Client in the global class path of the Content Engine and adds the EAE Query Service Taglib to all publications. Please refer to the Escenic Analysis Engine documentation for a discussion of these components.

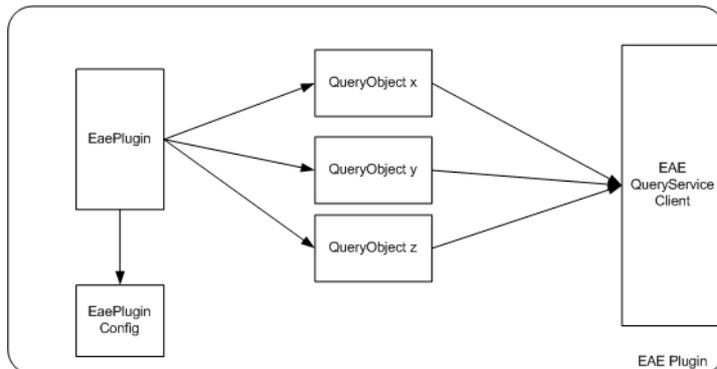
2.3 How does EAE Plugin work?

EAE Plugin is an ECE plugin and runs within an ECE instance. A figure below provides an architectural overview of an ECE deployed with EAE Plugin.



The EAE Plugin consists of 2 core classes, **EaePlugin** and **EaePluginConfig**. **EaePlugin** implements **neo.xreditsys.plugin.PluginService**, providing a service method that can be called by ECE components. This method takes two parameters, **ServiceRequest** and **ServiceResponse**. The service method will resolve the **ServiceRequest**, extracting a query object which is forwarded to a query component. The query component will execute the query using EAE QueryService Client

and then convert the query result to a pre-defined format that will be stored in the **ServiceResponse** object sent back to the ECE component. This is shown in the next figure.



Escenic Content Studio uses a query component called **StudioQuery1**.

When this plugin is installed, EAE Query Service Client will be available for the ECE components as well. Please refer to EAE Query Service Client JavaDoc, distributed with the EAE release, for documentation.

3 Installation

3.1 Pre-installation requirements

EAE Plug-in release

You need an EAE Plug-in release, **analysis-engine.zip**. If you do not have a release, please contact your Escenic representative.

ECE

Escenic Content Engine must be set up and running in your infrastructure.

EAE

Escenic Analysis Engine must be set up and running in your infrastructure.

3.2 Installing EAE Plug-in

Follow the standard ECE plug-in installation found as **Plug-in installation chapter** later in this document.

When the plug-in is installed, please finish section 3.3 before restarting.

3.3 Configuring EAE Plug-in

The EAE Plug-in needs to know the URL to the EAE Query Service and the Escenic Analysis Reports. Please copy the file **misc/siteconfig/com/escenic/analysis/EaePluginConfig.properties** to one of the configuration layers. For example, you can copy the file to **{ECE_HOME}/common/com/escenic/analysis/** and edit it as appropriate for your site.

Update **eaeQsUrl** with the correct URL to the EAE Query Service, and the update **earUrl** with the correct URL to the Escenic Analysis Reports application.

If EAE page view Logger is storing client user IDs in the meta field, for the purpose of viewing unique number of sessions, users or similar, and you want i.e. Content Studio to show this distinct meta count as well, then **distinctMetaCountEnabled** should be set to **true**. Distinct meta count, or unique client IDs, will then be showed inside parenthesis beside the page views.

Please copy the file **misc/siteconfig/com/escenic/analysis/webservice/AnalysisLinkHeaderPlugin.properties** to one of the configuration layers. For example, you can copy the file to **{ECE_HOME}/common/com/escenic/analysis/webservice/** and edit **queryServiceURL** with correct URL as appropriate for your site. The LinkHeader plugin needs to know the URL otherwise content Studio will not be able to show Statistics

Restart ECE application server when finished editing the property file.

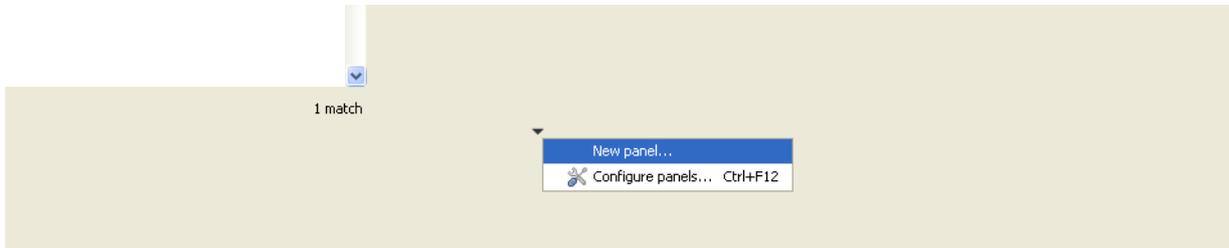
4 Getting Started

This chapter will help you get started accessing web statistics for the ECE components that use the EAE Plugin.

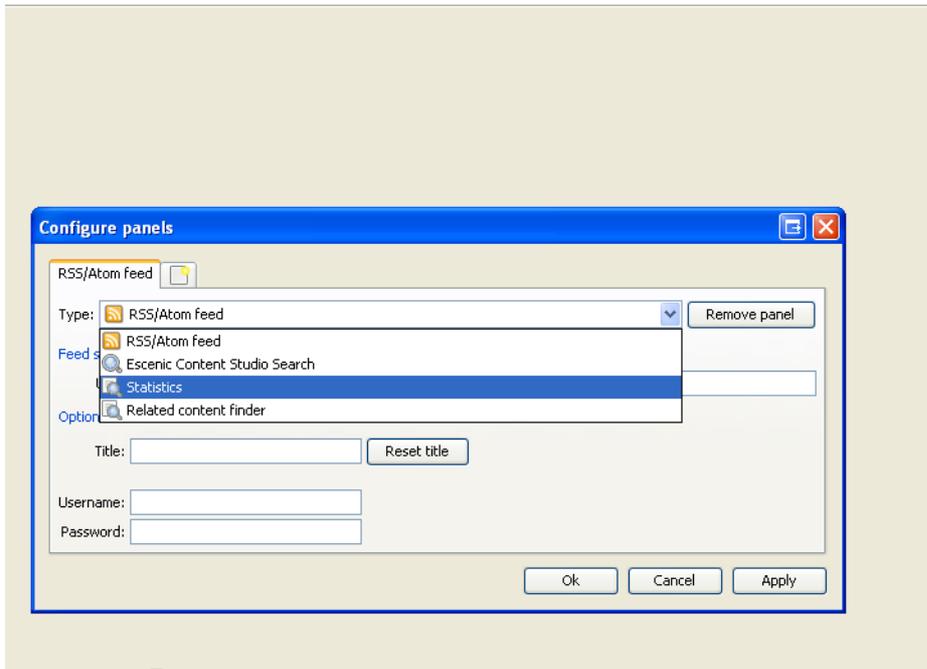
4.1 Escenic Content Studio

Escenic Content Studio will provide web statistics for section frontpages and its desked articles when the EAE Plugin is installed.

In order to access this information, the user needs to create a 'Statistics' panel in the Content Studio. To add a new panel, log into Content Studio, and click on 'View' on the top menu bar. In the 'View' menu, click on 'Configure panels' to configure the Statistics panel to be added. The snapshot below demonstrates this.



When the 'Configure panels' window is open, create a new panel. 'Statistics' should be selected from the drop down menu for the type of the new panel.



Open a section, i.e. the frontpage section. Doing this will populate the statistics panel with the most popular articles viewed under this section. The panel shows the title of the articles viewed the most, the last time these were viewed, the total number of page views for these articles, and the trend information for these articles.

The trend information is presented to give a visual indicator of increase or decrease in page views for the particular article. Trend calculation is based on a five minutes slot comparison. The trend is calculated for the articles by comparing their statistics between these five minutes slots. If the page view of an article is greater in the current slot than that of the previous slot, the trend icon will show an upward green arrow indicating the trend of the article has increased. If the page view for the current slot is less than that of the previous slot, the trend will be a downward red icon. If there is no change in the page views of two consecutive slots, the trend will be a sideways yellow icon to indicate no change in popularity of the article.

The Statistics panel has a refresh button to repopulate the panel with the most recent statistics along with the trend information for the section in question. A sample statistics is shown below.



Title	Last Viewed	Pageviews	Trend
Capello: I can take the pressure	11 minutes ago	10	↑
Nkufo: Spain still the favourites	12 minutes ago	9	↑
Brazil buoyant, Nazionale ner...	14 minutes ago	1	↓

When refreshing, the statistical information available in the local cache (EAE Plugin query cache) is returned immediately. The first time a query is executed, it will return **n/a** since no data exists in the cache. The next time (if not too soon) the refresh button is pressed, numbers will appear. The query results are valid for 2 minutes (this can be changed in `EaePluginConfig.properties`). On each query, the cached data will be returned and then if the cached result is older than 2 minutes, it will be updated in the background. This way, the refresh button must sometimes be pressed 2 times (how long between each click depends on the load on the server), to update the statistical information.

5 Plug-in installation

This chapter is an installation guide for Escenic plug-ins.

- Escenic is installed and in working order.
- The EAR assembly tool has been extracted and successfully used to set up a test ear file as per the instructions in the ECE Installation Guide.
- The plug-in distribution file: `<plugin>-<version>.jar` (or `.zip`).

5.1 Installation

Escenic Plug-ins should be installed using the assembly tool available from the ECE download page.

5.1.1 Server Install

Create Plug-in Directory

If `ECE_HOME/plugins` do not exist, create it. If you want to change the location of the plug-in folder, edit the `<assembly-tool-home>/assemble.properties` file.

Unpack Plug-in Distribution

Unpack the plug-in distribution file to `ECE_HOME/plugins`, such that there is a directory `plugins/<plugin-name>` afterwards.

Run DB Scripts

Run database scripts from `plugins/<plugin-name>/misc/database/eae-<db-vendor>.sql`. If there is no `misc/database` directory there are no db scripts. The scripts must be run in this order:

Rebuild and Deploy ECE

Build the ECE Enterprise Archive using the command `ant ear`. The assembly tool will add the plug-in to ECE's classpath, including the default plug-in configuration. It will add any required components to the web applications. Deploy the new EAR-file including updated webapps.

Customize the Plug-in Configuration

Refer to chapter [chapter 3](#) for any configuration options available to this plug-in.

Some plug-ins contain a `<plugin-name>/misc/siteconfig` folder. Prior to ECE 4.3 the content of this folder had to be copied to the local configuration layer and changes to `Initial.properties` applied. This is no longer necessary. The plug-ins default configuration is read from the classpath. The `siteconfig` folder, if it exists, contains skeleton configuration files that may be used to customize the plug-in.

Verify Installation

Restart the application server and verify the installation status in **View installed plugins** in the Escenic Admin webapp.

If the application server does not support EAR based deployment, the JAR files located in the `<plugin-name>/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.

5.1.2 Installing Plug-in Web Applications

Some Escenic plug-ins come with their own administrator web applications. If available, these are located in the `<plugin-name>/webapps/` directory of the plug-in distribution. These will be included in the EAR by the assembly tool and will be available after deployment. The web applications may also be deployed manually by using the war file located in `ECE_HOME/dist/war/`. Please refer to your application server documentation on how to deploy a web application.

Excluding the Plug-in Web Wpplcation From the EAR

The plug-in web application can be excluded from the EAR by using the property `global-exclude-webapps` in `assemble.properties` or by using the property `exclude-webapps` in the server class configuration.

The Statistics Plug-in

The web applications for the statistics plug-in are not included in the EAR by default. They can be included by removing them from the property `global-exclude-webapps` in `assemble.properties`.

5.1.3 Installing Demo Templates

Some Escenic plug-ins come with example templates. If available, these are located in the `<plugin-name>/wars/` directory of the plug-in distribution. To install the demo templates, do the following:

Create Publication

Upload the WAR file to the Escenic Admin webapp as a publication definition. Create a publication. If any content is bundled with the templates, an import will be started automatically.

Build Publication

Add a publication definition to your `/publications` folder, defining the publication name and the WAR file location. An example definition for the Calendar demo templates may look like this:

```
source-war: ../../plugins/calendar/wars/calendardemo.war
context-root: /calendar
```

Save the file and build using the `ant -q ear` command. Note that the relative path in the above example depends on your setting for the `publication-wars` property of the assembly tool configuration.

Deploy the Publication

Deploy the complete EAR-file, which includes the new publication, or deploy the single WAR file located in the `ECE_HOME/dist/war` directory.

Development Setup

In a development environment, the publications should be deployed as exploded webapps. After building the demo templates the first time, extract the resulting WAR file located in the `<assemblytool>/dist/war` directory to an appropriate location and deploy this as an exploded webapp.

Redeploying a Webapp

OC4J, Tomcat, Resin

Overwrite the existing 'war' file with the new version and restart the appserver.

WebLogic

Use the 'console' web application (default: `http://<server>:7001/console`):

1. undeploy the running webapp (uncheck the 'deploy' box)
2. deploy the new 'war' file

5.2 Verifying the installation

Installation status can be checked in the Escenic **admin** webapp (default: `http://<server>/admin`). The link **View installed plugins** lists all installed plug-ins and their status. 'Green check' means OK, 'red cross' means that there is an installation problem.

The 'installation status' page does not check for all possible problems. When the installation status shows green, the installation should be checked further by logging in to Escenic, accessing the plug-in, and creating and deleting a few elements.

Menu				
Home List pubs New pubs	analysis-engine		trunk-SNAPSHOT	The Escenic Analysis Engine plugin
				The EAE plugin allows esce statistical information from

The 'category' and 'menuEditor' plug-ins are successfully installed. The 'forum' plug-in shows a 'ClassNotFoundException', meaning that there is a problem with the appserver classpath.

5.3 Troubleshooting

This section lists the most common problems and describes how to fix them.

ClassNotFoundException

On the plug-in status page or in the appserver log: The plug-in is not correctly installed or not installed at all. Check the contents of the `engine.ear` file and verify that all the libraries located in `<plugin-name>/lib` are present in the EAR file's lib folder.

404 Not Found

In the browser, when accessing a plug-in-related link: This may be caused by the webapp not being updated with the plug-in files. Please check that the corresponding WAR file has been updated by the assembly tool and redeploy the webapp if necessary. Note that the correct WAR file to deploy is located in the assembly tool's `dist/war` folder, not in `ECE_HOME/webapps`.

ClassCastException, NoSuchMethodError

In the appserver log or in the browser: A code version issue - there is a mismatch between the code version of Escenic and the code version of the plug-in. Contact Escenic for an update.

For other problems contact Escenic support.