

Concepts for EPiServer (1.0)

Installation

Concepts for EPiServer (CfE) is shipped as an EPiServer package and can be installed by using the standard deployment centre. In addition to this installation there are a few web.config changes required. This guide walks through the complete setup process.

Pre-Requisites

1. Web3 Platform (<http://www.networkedplanet.com/products/web3>) version 1.4.
2. Ensure that the Web3 service can be accessed by the EPiServer environment. Either by installing on the same machine as EPiServer or installing Web3 on a machine accessible over HTTP by EPiServer.
3. The EPiServer site is set to run under .NET 4.0.

Installing Concepts for EPiServer

1. Open the EPiServer deployment centre
2. Select "Install a Module from a Compressed File"
3. Browse to and select the ConceptsForEPiServer.epimodule file.
4. The deployment centre will present the install options to 'Overwrite existing files' and 'Apply configuration changes'. Click Next.
5. Select the EPiServer site where CfE will be installed.
6. Then click "Install" to deploy the files.

The installer will add make most of the necessary changes required to web.config.

IMPORTANT

The location of the Web3 server and topic map to be used should be specified by editing Web3Service and TopicMapAddress attributes in configuration/Web3Settings path. See the Concepts for EPiServer Configuration Reference for details.


Web3 Configuration Changes

In the Web3 Hive folder (typically C:\Program Files\Common Files\Web3\CONFIG\) open the serverConfig.xml file and add the following commit queue information.

```
<commitQueue>.\Private$\Web3EPiServerCommits</commitQueue>
```

Activate Web3 Event Handler Scheduled Job

To ensure that the data in the Web3 Platform stays in sync with that in EPiServer, CfE includes an event handler scheduled job that must be manually activated. From the EPiServer site administration page, the 'Web3 Event Handler' job can be found under the 'Scheduled jobs' section.

Web3 Event Handler 

A scheduled job that syncs web3 changes into EPiServer.

Settings | History

Active

This job should be run every

Next scheduled date

1 Web3 Event Handler Scheduled Job

The frequency of how often the event handler job is run depends on how often the site is updated and how urgent changes in data should be reflected in the Web3 Platform. As a general guide, scheduling the event handler to run at 5 minute intervals is the recommended setting.

Concepts for EPiServer Configuration Reference

The following section lists the configuration changes that are made to the EPiServer site configuration automatically by the installer.

Web.config Change Reference

The following changes are made to the site web.config by the installer.

Configuration sections and handlers

Configuration sections for Web3Settings and LinkedData are added to configuration/configSections

```
<configuration>
  <configSections>
    <section name="Web3Settings"
      type="NetworkedPlanet.Web3.EPiServer.Web3Configuration,
      NetworkedPlanet.Web3.EPiServer" />

    <section name="LinkedData"
      type="NetworkedPlanet.Web3.EPiServer.LinkedData.Configuration,
      NetworkedPlanet.Web3.EPiServer"/>
  </configSections>
```

The Web3Settings element with the following default attributes are added to the configuration section.

```
<configuration>
  <Web3Settings Web3Service="http://127.0.0.1/web3"
    Web3QueueName=". \Private$\Web3EPiServerCommits"
    TopicMapAddress="http://127.0.0.1/web3/topicmaps/{topic-guid}"
    SchemaPrefix=http://www.networkedplanet.com/epiweb3/types/
    TopicMapEditorUserGroup="Administrators">
  </Web3Settings>
```

The attributes in the Web3Settings element are as follows:

Web3Service	The base URL where the Web3 service is installed. <i>e.g. http://localhost/web3</i>
Web3QueueName	The message queue name. <i>e.g. .\Private\$\Web3EpiServerCommits</i>
TopicMapAddress	The full URL of the Topic Map being used for this EPiServer instance. <i>e.g. http://localhost/web3/topicmaps/cc05ce31-efa7-4dec-99c6-0ee443e32b35</i>
SchemaPrefix	An optional URL that defines the base URI for Web3 type identifiers. <i>e.g. http://www.networkedplanet.com/epiweb3/types/</i>
TopicMapEditorUserGroup	Name of an EPiServer user group. Members of this group will be shown links to edit Web3 Concepts. <i>e.g. Administrators</i>

HTTP Module

Within the configuration/system.webServer section, a new HTTP module Web3HttpModule is added.

```
<configuration>
  <system.webServer>
    <modules runAllManagedModulesForAllRequests="true">
      <add name="Web3HttpModule"
type="NetworkedPlanet.Web3.EPiServer.Web3HttpModule, NetworkedPlanet.Web3.EPiServer"
/>
    />
  />
```

Episerver.Shell

The Web3 module is added to the publicModules element in the episerver.Shell element.

```
<configuration>
  <episerver.shell>
    <publicModules rootPath="~/modules/" autoDiscovery="Minimal">
      <add name="Web3">
        <assemblies>
          <add assembly="NetworkedPlanet.Web3.EPiServer"/>
        </assemblies>
      </add>
    </publicModules>
  />
</configuration>
```

EPiServer.Config Change Reference

Within the EPiServer.Config file the Web3 URL rewrite provider is added as the default handler.

URLRewrite Handler

```
<urlRewrite defaultProvider="EPiServerWeb3TopicUrlRewriteProvider">
  <providers>
    <add name="EPiServerWeb3TopicUrlRewriteProvider"
description="Provides concept based url re-writing"
type="NetworkedPlanet.Web3.EPiServer.TopicUrlRewriter,NetworkedPlanet.Web3.EPiServer" />
  />
</urlRewrite>
```

Web3 Platform serverConfig.xml

Concepts for EPiServer uses MSMQ to receive change notifications from the Web3 Platform. To enable this, the following change needs to be made to serverConfig.xml. This file can typically be found on the machine where the Web3 Platform is installed at "C:\Program Files\Common Files\Web3\CONFIG".

```
<commitQueue>.\Private$\Web3EPiServerCommits</commitQueue>
```

Configuration Guide

Configuring Linked Data support

Linked Data support is configured within the configuration/LinkedData section in web.config.

```
<configuration>
  <LinkedData>
    <Namespace prefix="rdfs">http://www.w3.org/2000/01/rdf-schema#</Namespace>
    <Namespace
prefix="demo">http://www.networkedplanet.com/psi/demo/types/</Namespace>

    <TypeMapping PageType="Article" RdfType="demo:article" />
    <PropertyMapping PropertyName="PageName" RdfType="rdfs:label" />
    <PropertyMapping PropertyName="RelatedPeople" RdfType="demo:people" />
  </LinkedData>
```

- Namespace defines namespace prefixes that are expanded as part of the property mapping.
- TypeMapping defines the PageType attribute (the name of an EPiServer page type) and the corresponding RDF type.
- PropertyMapping defines how an EPiServer page property should map to an RDF type.

Topic Type Page Mapping

CfE supports Concept pages. These pages are rendered using templates in EPiServer. The configuration defines which topic types are rendered by which templates.

```
<configuration>
  <Web3Settings ... >
    <TopicTypeMappings>
      <add PageTemplate="Person"
TopicTypeIdentifier="http://www.networkedplanet.com/epiweb3/types/person" />
    </TopicTypeMappings>
  </Web3Settings>
```

- PageTemplate is the name of .aspx page in the 'templates' folder inside the EPiServer solution. (Note: the location of the templates will be made configurable in final release.)

- TopicTypeIdentifier attribute is the subject identifier of the type in Web3.

Editor Guide

Configuring Page Types

CfE uses two page properties to map pages and topics to and from the Web3 Platform.

PageTopicMappingProperty

This property is used to denote a page created in EPiServer is an instance of a topic in the Web3 Platform. That is the page should be considered to be the topic. This allows for additional information to be added to the EPiServer page instance without requiring the topic to include the same information. Only a single page in EPiServer can represent a topic in the Web3 Platform.

RelatedTopicProperty

This property is used to create relations from a page in EPiServer to a topic in the Web3 Platform. A page can be related to many topics.

Configuring RelatedTopicProperty and PageTopicMappingProperty

When adding either property to a Page Type, both properties share the same required settings to be configured in order for the property to properly link with the topic type in the Web3 Platform. This is achieved by selecting a subject identifier for a required topic type.

The screenshot shows the 'Edit Property' dialog box with the following details:

- Title:** Edit Property
- Tabs:** Common Settings (selected), Custom Settings
- General Section:**
 - Type:** PageTopicMappingProperty (selected in dropdown)
 - Name:** (empty)
 - Presentation control:** (empty)
- User Interface Section:**
 - Display in Edit Mode
 - Field name:** (empty)
 - Help Text:** (empty)
 - Tab:** Content
- Buttons:** Save, Delete, Cancel

2 Adding PageTopicMappingProperty to Page Type

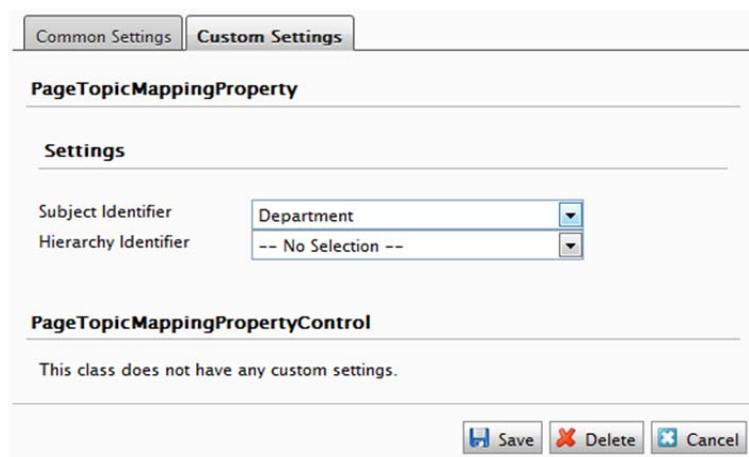
The topic identifiers are configured by editing the topic type mappings shown in the Topic Type Page Mapping section.

Subject Identifier

The subject identifier the property maps to in the Web3 Platform.

Hierarchy Identifier (Optional)

An optional subject identifier for a hierarchy type can be specified to map the property to a hierarchy type in the Web3 Platform.



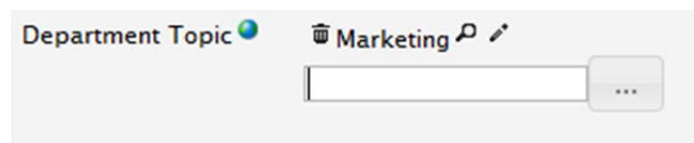
3 Selecting subject identifier 'Department'

Editing topic associations

Since the CfE controls are EPiServer properties, they are accessed by Editors from the usual page Edit interface. Both the PageTopicMappingProperty and RelatedTopicProperty implement the same edit interface. For simplicity only the PageTopicMappingProperty control is shown.




Editing Interface

The control shows a list of selected topics by label.

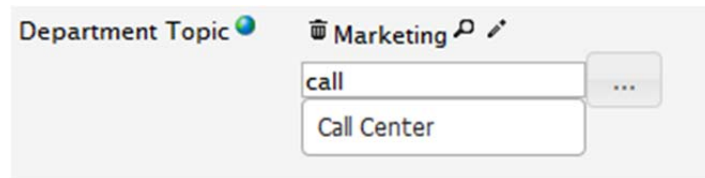


4 PageTopicMappingProperty with selected topic 'Marketing'

Besides the topic label, the property control has the following buttons.

-  Remove topic link – Removes the selected topic from the list.
-  View topic in Web3 Platform – Displays the topic as stored in the Web3 Platform. (Only available in PageTopicMappingProperty)
-  Edit topic in Web3 Platform – Allows the editing of the topic within the Web3 Platform. (Only available in PageTopicMappingProperty)

To choose a topic from the Web3 Platform, the name of the topic, if known can be entered and a list of matching topics will appear underneath the text input area. Selecting a topic from the drop down list will add it to the list of topics.

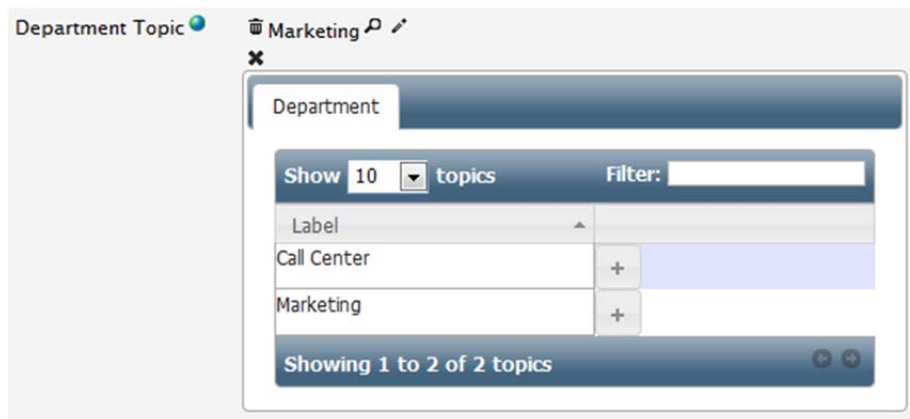


5 Type ahead topic picker for selecting topic 'Call Center'

Alternatively, a list of all available topics can be obtained by clicking on the button next to the text input area to open the Topic Picker.



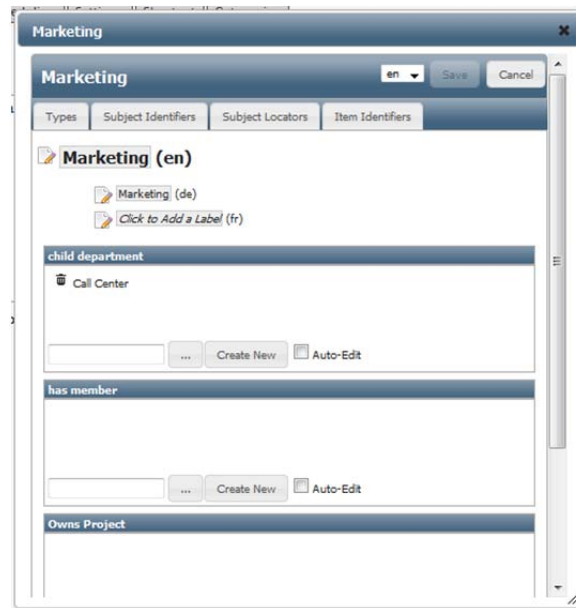
Open Topic Picker – Opens the Topic Picker control showing a list of topics to choose from.



6 Topic Picker show list of available topics

To add a topic using the Topic Picker control, click on the '+' button next to a desired topic's label to add it.

When editing a topic within EPiServer using the Edit topic icon, a modal dialog is displaying featuring the Web3 Platform topic editor. This allows changes to be made to the topic within the Web3 Platform from the EPiServer environment.



7 Editing the topic 'Marketing' using the inline Web3 Platform topic editor

After making any changes to a topic using the editor, these must be saved by clicking on the Save button which will save the changes to the Web3 Platform.

Developer Guide

The following guide will show you how you can fetch related topics or pages using the CfE API from within EPiServer pages or Concept template pages.

Web3DataHelper Class

The Web3DataHelper class within the NetworkedPlanet.Web3.EPiServer namespace contains a number of useful static methods as well as extension methods to other classes. This guide will demonstrate the most commonly used methods for achieving the following tasks;

- Fetching related topics from an EPiServer page topic or from a Concept page.
- Fetching related EPiServer pages from an EPiServer page topic or from a Concept page.
- Executing a SPARQL query within the Web3 Platform.

Retrieving a Topic for Concept Page

Before making use of any of the Web3DataHelper methods to find related topics, the starting topic must first be retrieved. For Concept pages, an extension method GetTopic() on the Page class is provided. This can be used in the Page_Load() method of the page to retrieve the Topic for the page. The Topic class can be found in the namespace NetworkedPlanet.Web3.Platform.Service.Client.Model.TopicMapService.

```
using NetworkedPlanet.Web3.Platform.Service.Client.Model.TopicMapService;

...

private Topic _topic;

protected void Page_Load(object sender, EventArgs e)
```

```
{
    _topic = this.GetTopic();
}
```

Retrieving a Topic for a Concept or EPiServer Page

A Concept template page that can have a page in EPiServer the topic should first be requested using the extension method `GetTopic()` to determine if the page has been requested as a Concept template page. If `GetTopic()` returns null, then `GetTopicForPage()` will determine the topic from the Page ID. This ensures that a Topic is returned both for when the page is requested as a Concept template page or as an instance of an EPiServer page.

```
protected void Page_Load(object sender, EventArgs e)
{
    // try and get the topic as though it was a normal concept page
    _topic = this.GetTopic();

    // if no topic see if there is a mapped topic
    if (_topic == null)
    {
        _topic = this.GetTopicForPage();
    }
}
```

Fetching Related Links to Topics

With the topic retrieved for the page, the Topic extension helper method `RelatedTopicLinks()` can be used to return an `IEnumerable<Link>`. The Link class contains the topic URL and topic Label. The Label is obtained from the topic using `GetDisplayLabel()` on the topic which returns a label in current page language.

The `RelatedTopicLinks()` method for Topic accepts the following parameters;

- string **identifier** – The subject identifier for association role type. This can be a partial or full URI for the subject identifier. A partial identifier is combined with the base URI. To specify a full URI it must begin with 'http://'.
- string **templateName** (optional) – If specified, then this will be used as for template page rendering.

The `RelatedTopicLinks()` method for PageBase accepts the following parameters;

- string **property** – The name of the RelatedTopicProperty or PageTopicMappingProperty that is contained on the Page Type.
- string **templateName** (optional) – If specified, then this will be used as for template page rendering.

```
public IEnumerable<Link> AreasOfExpertise
{
    get
    {
        return CurrentPage.RelatedTopicLinks("relatedExpertise",
"AreasOfExpertise");
    }
}
```

```
}  
}
```

As an alternative to returning `Link` objects, `TopicReference` objects can be retrieved using the similar method `RelatedTopics()` which returns `IEnumerable<TopicReference>`. `RelatedTopics()` accepts the same parameters as `RelatedTopicLinks()`.

Fetching Related Pages

Related EPiServer pages that have been mapped to a topic in the Web3 Platform can be returned by using the `Topic` extension method `RelatedPages()`. The `RelatedPages()` method accepts the following parameter;

- `string property` – The name of the `RelatedTopicProperty` or `PageTopicMappingProperty` that is contained on the Page Type.

```
public PageDataCollection Authors  
{  
    get { return _topic.RelatedPages("author"); }  
}
```

Only EPiServer pages that are mapped to topics will be returned when using this method.

SPARQL Query

SPARQL queries can be executed by C# against the configured topic map in the Web3 Platform. The results are returned as an `XDocument` using the W3 SPARQL namespace, which is available as a static property from `Web3DataHelper.SparqlNamespace` for convenience.

Executing a SPARQL Query

To execute a SPARQL query, `Web3DataHelper` provides an `ExecuteSparql()` method. This method accepts a single query parameter to run against the configured topic map. For detailed instructions on creating SPARQL queries to execute against the Web3 Platform, see the included documentation for the Web3 Platform.

To make use of the configured scheming C# as a PREFIX in the query, this can be accessed from the `SchemaPrefix` property from the `Web3Configuration` instance.

```
string query = @"PREFIX tms: <http://www.networkedplanet.com/tmsparql/> "  
query += "PREFIX ont: <"+ Web3Configuration.Instance.SchemaPrefix + "> "  
query += "SELECT ?expertise ?label WHERE { <" + _topic.WebAddress +  
> ont:ownsProject ?project . ?project ont:relatedExpertise ?expertise . ?expertise  
tms:label ?label } "  
;
```

```
XDocument result = Web3DataHelper.ExecuteSparql(query);
```

Processing SPARQL Results

SPARQL results are returned as an `XDocument`. The descents of the SPARQL, can be iterated through using the namespace under `Web3DataHelper.SparqlNamespace + "result"`. This will return each result from the query. As a convenience, `Web3DataHelper` adds an extension method `GetRowColumnValue()` to `XElement`, that will return the value from a given column name.

```

private static IEnumerable<Link> GetSparqlQueryLinks(XDocument queryResult)
{
    var result = new List<Link>();
    foreach (var row in queryResult.Descendants(Web3DataHelper.SparqlNamespace +
"result"))
    {
        Uri topicLink =
Web3DataHelper.GetTopicLink(row.GetRowColumnValue("expertise"), "AreaOfExpertise");
        string topicLabel = row.GetRowColumnValue("label");
        result.Add(new Link(topicLink, topicLabel));
    }
    return result;
}

```

Links to topics can be obtained by passing the returned topic value to the Web3DataHelper method GetTopicLink() as seen in the code above.

Importing and Exporting Topic Map and EPiServer Page Data

When exporting page data from within EPiServer, it is important to ensure that as well as the pages that the topic map data is also exported at the same time if it is to be imported to a different location. The following sections show how to ensure that classified pages remain in sync with the topic map.

Exporting EPiServer Page Data

Concepts for EPiServer stores related topic information within page properties. When exporting pages from within EPiServer, it is important that the property settings for these pages are exported at the same time as the pages and page types by checking the 'Export property settings' box is checked.

Export Data

Export data from one EPiServer CMS to another.

Export pages
 Select part of structure
 International Industries [5]

Export files that the pages link to
 Export globalized pages
 Automatically export dependent page types

Export page types
 Article Normal Page SysRecycleBin
 Department SysRoot

Export frames
 Export dynamic property definitions
 Export tabs
 Export categories
 Export files
 Export property settings

8 Ensure that Export property settings is checked during export

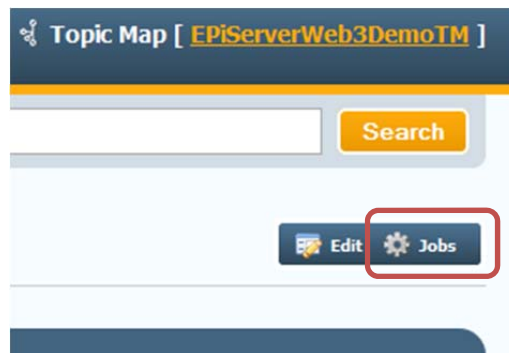
IMPORTANT

If a new topic map is to be created in a new site installation, the topic map that was used to create the classifications for the exported page data must be exported at the same time. Importing page data without using the same topic map data will not function correctly.

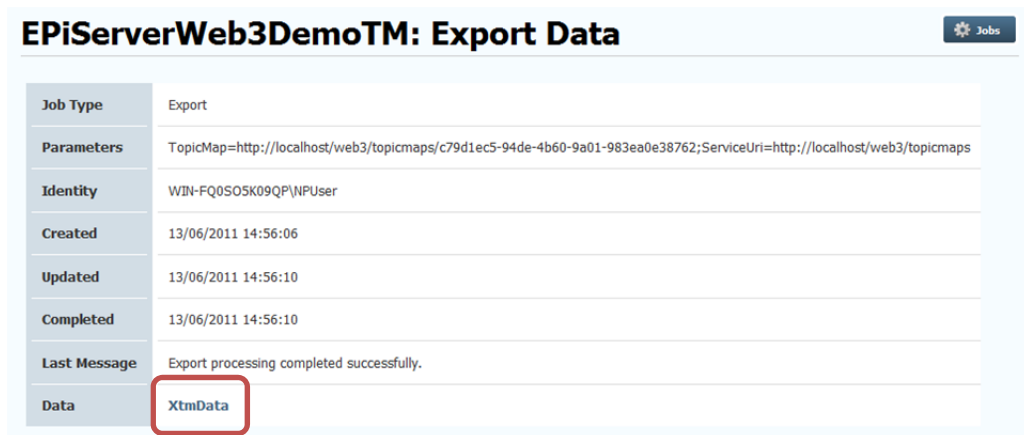
Exporting Topic Map Data

Exporting topic map data requires the Web3Admin PowerShell snap-in is installed.

1. Open a new PowerShell prompt and load the Web3Admin snap-in
> Add-pssnapin Web3Admin
2. Create the export topic map job
> export-topicmap
> TopicMapAddress: <address of the topic map to export data from>
3. This will create an export topic map data job. To download the XtmData file, browse to the topic map in the Web3 Platform and click on the 'Jobs' button.



4. Click on the 'Export Data' job, if the job is not yet completed click the 'Refresh' button until it is.



5. Click on the 'XtmData' link within the job to download the topic map data.

Importing EPiServer Page Data

After importing page data into EPiServer, it is essential that the Web3 Consistency Checker job is executed. This can be found from within the Admin section of the EPiServer site under Scheduled Jobs -> Web3 Data Consistency Checker.