

[Register](#) [New password](#) [Login](#)[Documentation](#) / [EPiServer CMS](#) / [7.5](#) /[Hide menu](#)

Last updated: Jul 09 2014

**+ Add-ons**[Architecture](#)[BLOB providers](#)**+ Caching**[Client resources](#)**+ Configuration****+ Content****+ Deployment****+ Dynamic content**[Dynamic data store](#)**+ Editing****+ EPiServer CMO****+ Event management****+ Globalization****+ Initialization****+ Localization****+ Logging****+ Personalization****+ Rendering**

---

**+ Routing**

---

**Scheduled jobs**

---

---

**+ Search**

---

---

**+ Security**

---

---

**+ User interface**

---

**Virtual path providers**

---

---

**+ Workflows**

---

**XForms**

---

## Extending the navigation

### Table of contents

- [Introduction](#)
- [Using menu providers](#)

### Introduction

OnlineCenter is equipped with extendable navigation. This allows EPiServer CMS and other products to be accessed by editors from the top menu. The preferred way of adding an interface to the menu is using the [MenuItem] attribute. Other options include configuration and implementing a custom menu provider. Both attributes and menu providers need to reside in an assembly configured as a shell module.

#### [MenuItem] Attribute

MenuItem is an attribute used on ASP.NET MVC controllers and WebForm pages. This attribute adds links to the top menu (requires the module to be registered, see Shell Modules). The required parameter menuPath is a logical path of the menu element in the menu. The URL to the actual interface to link to is added using the Url parameter.

Example (WebForm page):

C#

```
1 [MenuItem("/global/cms/edit", Text = "Edit", Url =  
2   "Path/To/Default.aspx")]  
3   public partial class DefaultPage : SystemPageBase  
4   {  
5   }
```

This adds a menu item to the CMS menu section with the text Edit and the URL `/[PublicOrProtected]/[ModuleName]/Path/To/Default.aspx`.

Example (MVC action):

C#

```
1 public class DashboardController : Controller
2     {
3         [MenuItem("/global/dashboard", Text = "Start")]
4         public ActionResult Index()
5         {
6         }
7     }
```

This adds a menu item to the top menu bar with the text Start and the URL is inferred from the ASP.NET MVC route table.

### Localization

Localization of menu items defined with attributes is done via a static property of a resource class. ResourceType references [MenuItem("/global/dashboard", TextResourceKey = "Start", Resource

a class with a static property. TextResourceKey is the name of the static property that returns the localized text.

Example:

### Menu path

To organize menu items in a tree structure the menu path is used. All menu items in the top menu are in the “/global” bucket. The next segment is the name of menu section, for example, “/global/cms”. The last segment represents the actual user interface, for example, “/global/cms/edit”.

### URL

Some menu items have an URL. This creates a link in the menu. The page at the end of this URL should render a menu where the corresponding menu path is selected.

### Permissions

You can restrict who sees a certain menu item by using the [Authorize] attribute.

Example:

C#

```
1 public class DashboardController : Controller
2     {
3         [MenuItem("/global/dashboard")]
4         [Authorize(Roles = "NoOne")]
5         public ActionResult Index()
6         {
7         }
8     }
```

### Using menu providers

The building blocks of the navigation are menu providers and menu items. A menu provider provides an enumeration of menu items which are organized in a tree according to their menu path. EPiServer CMS contains a menu provider that looks at [MenuItem] attributes and provides them as menu items.

### IMenuProvider

A slightly more technical alternative to attributes to extend the standard menu is implementing a menu provider. The menu provider is responsible for returning menu items that are correctly localized. For a menu provider to be used it needs to implement the `IMenuProvider` interface, be decorated with the `[MenuProvider]` attribute and be part of a registered shell module.

Note for future compatibility: The `[MenuProvider]` attribute extends a library that will be changed in a version after EPiServer CMS 6. This means that any code using this attribute will need to be recompiled.

### Menu items

It is possible to add menu sections and sub-menu items. The types supported are menu sections, drop-downs, URLs and pop-up menu items. Each menu item defines a path which decides its location in the menu hierarchy. For example, a URL menu item with path `/global/cms/myMenu` is placed in the CMS section of the menu (which has the path `/global/cms`).

Types:

- **URL Menu Item** are links, a click navigates to specified URL
- **Popup Menu Item** are links opened in a popup window (javascript window.open)
- **Section Menu Item** are top menu sections which open an underlying menu
- **Drop-Down Menu** Item are drop-down style menu items designed for the utility menu area (right)

Example:

C#

```

1  /// <summary>
2      /// Provides menu items for the CMS product.
3      /// </summary>
4      [MenuProvider]
5      public class CmsMenuProvider : IMenuProvider
6      {
7          /// <summary>
8          /// Provides the CMS menu section and the CMS settings section.
9          /// </summary>
10         /// <returns>
11         /// A list of <see cref="MenuItem"/>s that the provider exposes.
12         /// </returns>
13         public IEnumerable<MenuItem> GetMenuItems()
14         {
15             // Create the top menu section
16             var section = new MenuSection("CMS", // Title
17
18                 "/global/cms"); // Logical path
19             // Visible to CMS editors
20             section.IsAvailable = (request) =>
21                 PrincipalInfo.HasEditAccess;
22
23             // Create the edit menu item below the top
24             section
25             var cmsEdit = new UrlMenuItem("Edit", // Title
26
27                 "/global/cms/edit", // Logical path
28
29                 "/path/to/edit/default.aspx"); // URL
30             // Visible to CMS editors
31             cmsEdit.IsAvailable = (request) =>
32                 PrincipalInfo.HasEditAccess;
33             return new MenuItem[] { section, cmsEdit };
34         }
35     }

```

```
var cmsEdit = new UrlMenuItem("Edit", "/global/cms/edit", "/path/to/edit/default.aspx");
```

## Localization

A menu provider is responsible for returning localized menu items.

## Permissions

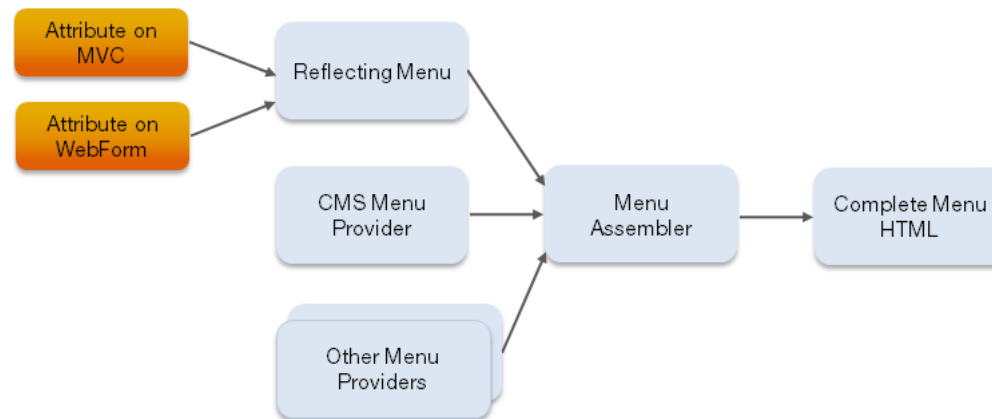
The menu provider can defer permission filtering to the menu item by setting the “IsAvailable” delegate to a method that checks access for the user provided with the RequestContext parameter.

Example:

```
// Make menu item visible to CMS editors
cmsEdit.IsAvailable = (request) => PrincipalInfo.HasEditAccess;
```

## Flow of menu Items

Menu items flow from the providers to into a hierarchical model which is rendered into HTML.



## Configuring web.config

To extend the menu you can, for example, configure web.config as follows:

XML

```

1  <episerver.shell>
2  <navigation>
3    <add text="Intranet" menuPath="/global/intra"
4      url="http://intra" sortIndex="100" />
5    <add text="My section" menuPath="/global/my"
6      menuItemType="Section" sortIndex="300" />
7    <add text="Edit" menuPath="/global/my/edit"
8      url="/my/edit.aspx" sortIndex="100" />
9    <add text="Admin" menuPath="/global/my/admin"
10     url="/my/admin.aspx" sortIndex="200" />
11  </navigation>
12 </episerver.shell>
```

## Comments



(By [Søren Raarup](#), 4/9/2015 12:14:07 PM)

So how do I remove an existing MenuItem like /global/user/logout ?

I tried [remove menuItem="/global/user/logout" /]  
It did not do the trick.



Post by [Søren Raarup](#) deleted, 4/9/2015 12:14:44 PM

---

© Copyright 2015 [EPiServer AB](#) | [About EPiServer World](#)