

How to define properties in EPiServer 7 - A quick reference

Monday, 16 July 2012



By: [Alexander Haneng](#)

COO & Head of Technology at Geta,
EPiServer Expert (EMVP)

Number of votes: 6

Views: 19640

Average rating: 

Syndication and Sharing



This is a quick reference on how to define strongly typed properties in code in EPiServer CMS 7.

Updated to EPiServer 7 CMS on 14.01.2013

This blog post was originally written for a preview version of EPiServer 7 CMS, but is now updated to the final release version.

This is the way you define a property in a [page type](#) or in a [block type](#). You can define if a property is searchable, required etc. using attributes.

```
[BackingType(typeof(PropertyLongString))]
```

```
[CultureSpecific]
```

```
[Searchable]
```

```
[Required]
```

```
[Editable(true)]
```

```
[Display(
```

```
    Name = "Heading",
```

```
    Description = "",
```

```
    GroupName = SystemTabNames.Content,
```

```
    Order = 10)]
```

```
public virtual string Heading { get; set; }
```

The attribute defaults are very good, so in most cases you can safely omit them. This code will accomplish the same as the code above:

```
[Display(
```

```
    Name = "Heading",
```

```
    Description = "",
```

```
    GroupName = SystemTabNames.Content,
```

```
    Order = 10)]
```

```
public virtual string Heading { get; set; }
```

The following attributes are available:

<i>CultureSpecific</i> (bool)	Defines if this property should have a different
-------------------------------	--

	value for each language. If not set: true for strings, other types false.
<i>Searchable</i>	Defines if the property value should be searchable. If not set: true for strings, other types false.
<i>Required</i>	Defines if a value for this property must be set before being able to save a page of the parent type. If not set: false.
<i>ScaffoldColumnAttribute(bool)</i>	Defines if this property is visible in Edit mode.If not set: true.
<i>Display(Name=..., Description=..., GroupName=..., Order=...)</i>	The Name, Description, GroupName and Order properties are used to set the EditCaption, HelpText , Tab and FieldOrder respectively.
UIHint	Used to select either editor/renderer or both by defining a hint string. You can use EPiServer.Web.UIHint to use hints for known types in the system, for instance UIHint.Image.
Ignore	By default all properties are included in a content type, you can use this attribute to tell the system to ignore this property.
<i>BackingType(Type)</i>	Defines the PropertyData type used to store values for this property. Must be a type inheriting from PropertyData. The backing type will be auto determined.

Validation attributes

If you want to restrict what values the editor can enter into a property you can define this in code and it will be enforced by EPiServer in edit mode. Here are some examples:

Restrict the length of a PropertyLongString:

```
[StringLength(20)]
```

```
[Display(
```

```
    Name = "Heading",
```

```
    Description = "",
```

```
    GroupName = SystemTabNames.Content,
```

```
    Order = 10)]
```

```
public virtual string Heading { get; set; }
```

Heading

The editor will only be able to enter 20 or less characters (HTML maxlength).

Restrict the value of a PropertyNumber to a range.

```
[Range(1,100)]
```

```
[Display(
    Name = "Max number of results",
    Description = "",
    GroupName = SystemTabNames.Content,
    Order = 10)]

public virtual Int32 MaxResults { get; set; }
```

Max number of
results

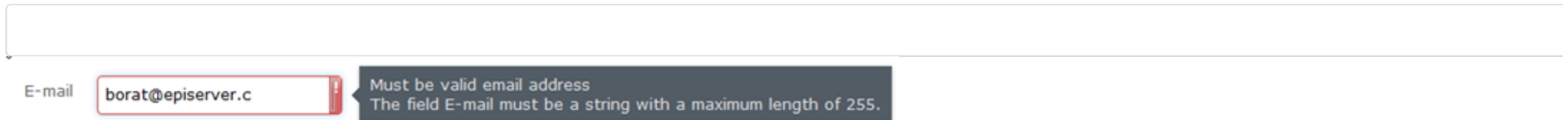
This value is out of range.

Restrict the value of a PropertyString with RegExg

```
[RegularExpression(@"^([A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,9})$")
, ErrorMessage = "Must be valid email address")]

[Display(
    Name = "Email",
    Description = "",
    GroupName = SystemTabNames.Content,
    Order = 10)]

public virtual string Email { get; set; }
```



The screenshot shows a web form with a label 'E-mail' and a text input field containing 'borat@episerver.c'. A red border highlights the input field. To the right of the field, a dark grey tooltip box contains the text: 'Must be valid email address' and 'The field E-mail must be a string with a maximum length of 255.'

Setting default property values

You can set default values for your properties as well as the built in properties by [overriding the method SetDefaultValues on the PageData object](#).

Example page type with all the property types

The code below defines a new page type called *PageWithAllPropertyTypes* which contains (almost) all the property types EPiServer supports. For a few properties you also need to define a *UIHint* to tell EPiServer exactly what you want. For example if you want an image property you need to define the property as a *Url* and add the attribute *[UIHint(UIHint.Image)]*.

```
using System;
using System.ComponentModel.DataAnnotations;
using EPiServer.Core;
using EPiServer.DataAbstraction;
using EPiServer.DataAnnotations;
using EPiServer.SpecializedProperties;
using EPiServer.Web;
using EPiServer.XForms;

namespace EPiServer.Templates.Alloy.Models.Pages
{
    [ContentType(DisplayName = "Page with all property types",
        Description = "A page with all the property types defined")]
    public class PageWithAllPropertyTypes : PageData
    {

```

```
// -----
// PropertyLongString (PropertyString)
// -----

[Display(
    Name = "Heading",
    Description = "",
    GroupName = SystemTabNames.Content,
    Order = 10)]
public virtual string Heading { get; set; }

// -----
// PropertyXhtmlString
// -----

[Display(
    Name = "Main body",
    Description = "",
    GroupName = SystemTabNames.Content,
    Order = 20)]
public virtual XhtmlString MainBody { get; set; }

// -----
// PropertyUrl
// -----

[Backintype(typeof(PropertyUrl))]
[Display(
    Name = "Link",
    Description = "",
    GroupName = SystemTabNames.Content,
    Order = 30)]
public virtual Url Link { get; set; }

// -----
// PropertyUrl (Image)
// -----

[UIHint(UIHint.Image)]
[Display(
    Name = "Image",
    Description = "",
    GroupName = SystemTabNames.Content,
    Order = 40)]
public virtual Url Image { get; set; }

// -----
```

```
// PropertyUrl (Document)
// -----
[UIHint(UIHint.Document)]
[Display(
    Name = "File",
    Description = "",
    GroupName = SystemTabNames.Content,
    Order = 50)]
public virtual Url File { get; set; }

// -----
// PropertyPageReference
// -----
[Display(
    Name = "Sitemap page",
    Description = "",
    GroupName = SystemTabNames.Content,
    Order = 55)]
public virtual PageReference SitemapPageLink { get; set; }

// -----
// PropertyLinkCollection
// -----
[Display(
    Name = "Links",
    Description = "",
    GroupName = SystemTabNames.Content,
    Order = 60)]
public virtual LinkItemCollection Links { get; set; }

// -----
// PropertyContentArea
// -----
[Display(
    Name = "Right block area",
    Description = "",
    GroupName = SystemTabNames.Content,
    Order = 70)]
public virtual ContentArea RightBlockArea { get; set; }

// -----
// PropertyBoolean
// -----
```



```
[Display(
    Name = "Show teaser",
    Description = "",
    GroupName = SystemTabNames.Content,
    Order = 80)]
public virtual Boolean ShowTeaser { get; set; }

// -----
// PropertyCategory
// -----
[Display(
    Name = "Selected categories",
    Description = "",
    GroupName = SystemTabNames.Content,
    Order = 90)]
public virtual CategoryList SelectedCategories { get; set; }

// -----
// PropertyDate
// -----
[Display(
    Name = "Event start time",
    Description = "",
    GroupName = SystemTabNames.Content,
    Order = 100)]
public virtual DateTime EventStartTime { get; set; }

// -----
// PropertyFloatNumber
// -----
[Display(
    Name = "Price",
    Description = "",
    GroupName = SystemTabNames.Content,
    Order = 110)]
public virtual Double Price { get; set; }

// -----
// PropertyNumber
// -----
[Display(
    Name = "Votes",
    Description = "",
```

```

        GroupName = SystemTabNames.Content,
        Order = 120)]
    public virtual Int32 Votes { get; set; }

// -----
// PropertyPageType
// -----
[Display(
    Name = "Filter for page type",
    Description = "",
    GroupName = SystemTabNames.Content,
    Order = 130)]
    public virtual PageType FilterPageType { get; set; }

// -----
// PropertyXForm
// -----
[Display(
    Name = "Form",
    Description = "",
    GroupName = SystemTabNames.Content,
    Order = 140)]
    public virtual XForm Form { get; set; }

// -----
// PropertyString
// Only use if you really have to.
// Use PropertyLongString instead with [StringLength(255)]
// -----
[BackingType(typeof(PropertyString))]
[Display(
    Name = "Short intro",
    Description = "",
    GroupName = SystemTabNames.Content,
    Order = 150)]
    public virtual string ShortIntro { get; set; }

}

```

The resulting page type in admin mode:

Page with all property types



A page with all the property types defined

Information

From code Yes
 Name PageWithAllPropertyTypes
 Display name Page with all property types

Settings

Add Property

		Name	Field name	Type	Required	Localized	Searchable	Tab	From code
		Heading	Heading	Long string (>255)			Yes	Content	Yes
		MainBody	Main body	XHTML string (>255)			Yes	Content	Yes
		Link	Link	URL to page/external address				Content	Yes
		Image	Image	URL to page/external address				Content	Yes
		File	File	URL to page/external address				Content	Yes
		SitemapPageLink	Sitemap page	Page				Content	Yes
		Links	Links	Link collection				Content	Yes
		RightBlockArea	Right block area	ContentArea			Yes	Content	Yes
		ShowTeaser	Show teaser	Selected/not selected				Content	Yes
		SelectedCategories	Selected categories	Category selection				Content	Yes
		EventStartTime	Event start time	Date/Time				Content	Yes
		Price	Price	Floating point number				Content	Yes
		Votes	Votes	Integer				Content	Yes
		FilterPageType	Filter for page type	Page Type				Content	Yes
		Form	Form	XForms form				Content	Yes
		ShortIntro	Short intro	String (<= 255)			Yes	Content	Yes

Other posts in this blog series:

How to create a simple Page Type in code for EPiServer CMS 7

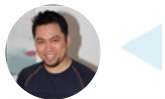
How to create a simple Block Type in EPiServer CMS 7

Comments



(By [Andrew Markham-Davies](#) , 25 July 2012 17:01, [Permanent link](#))

certainly an improvment over episerver < 6



(By [Dzulkarnain Nasir](#) , 30 August 2012 14:38, [Permanent link](#))

Very useful. Thanks for the info.

I'm trying to implement a PropertyDropDownList but I'm not sure how to set default option values in code.



(By [Alexander Haneng](#) , 04 September 2012 13:00, [Permanent link](#))

To set default values use:

```
public override void SetDefaultValues(ContentTypes contentType)
{
    base.SetDefaultValues(contentType);
    //Set up your defaults here
```

```
}
```



(By [Dzulqarnain Nasir](#) , 05 September 2012 12:23, [Permanent link](#))

What I meant was, is it possible to programatically set a list of options for the select box? If so, how does one go about this?



(By [Ted](#) , 19 October 2012 11:10, [Permanent link](#))

About list of options for the select box, check out Linus Ekström's posts on editor descriptors: <http://world.episerver.com/Blogs/Linus-Ekstrom/Dates/2012/9/EPiServer-7-Configuring-editors-for-your-properties/>



(By [Dzulqarnain Nasir](#) , 22 November 2012 09:51, [Permanent link](#))

Ted: Thanks for the heads up!



(By [Matt Cole-Baker](#) , 08 October 2014 21:58, [Permanent link](#))

Trying to define the property "EPISUBSCRIBE-EXCLUDE "
in code behind but the hyphen is an illegal character. Any way to do this?

```
[Display(  
    Name = "Deactivate subscription",
```

```
Description = "",  
GroupName = SystemTabNames.Content,  
Order = 5]]  
public virtual Boolean EPISUBSCRIBE-EXCLUDE { get; set; }
```

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