ALLAN THRAEN | 🕐 17 years ago | 📄 PDF | 🖬

# CUSTOM PROPERTY: CATEGORY DROP DOWN

Here's a request I've heard a couple of times: A DropDownList Property Control, with content that editors / administrators can control.
There's a built-in property control (AppConfigSettings), that can show a DDL, but it's based on settings in web.config, which makes it a bit more cumbersome for editors and administrators to modify the selection. Luckily we have the EPiServer Categories - an amazing, yet simple feature that has far from lived up to it's potential. Viewed correctly this can be the taxonomy you use to describe just about anything. For instance for defining the domain that you want your drop down list to be related to.

Imagine you're designing a system where each page needs a property identifying which world-region it is intended for - and the requirement specs for the web site in their eternal wisdom demands that this list is customizable if a continent should disappear due to global warming :-) In that case it's easy to define a taxonomy branch with-in EPiServer Categories called "Region", with sub-categories for each region: "Europe", "North America", "Asia", ... however you know that the editors never look under the "Categories" tab when editing a page, and you feel that this is so important it should be displayed in a simple list on the "Information" tab. In comes the CategoryDropDownList to save the day. Add the property to your page-type and make sure to **give it the same name as the category branch which children it should list**.

The Custom Property is fairly simple and build on top of the TextBoxBase Property Control. The core of the logic is here:

```
protected System.Web.UI.WebControls.DropDownList ddl;

public override void CreateEditControls()
{
    ddl = new System.Web.UI.WebControls.DropDownList();
    Category main_c=EPiServer.DataAbstraction.Category.Find(CategoryDropDown.Name);
    ddl.Items.Add(new ListItem("", ""));
    foreach (Category c in main_c.Categories)
    {
        ListItem li = new ListItem(c.LocalizedDescription, c.Name);
        li.Selected = (((string)CategoryDropDown.Value) == c.Name);
        ddl.Items.Add(li);
    }

    this.ApplyControlAttributes(ddl);
    Controls.Add(ddl);
}

public override void ApplyEditChanges()
{
    if (ddl.SelectedValue != null)  SetValue(ddl.SelectedValue);
    else SetValue(null);
}
```

To use this, simply include the 2 source-files for it in your solution. Enjoy!