


The [UniGrid](#) provides built-in transformations for easy formatting of the content of columns. To learn about the UniGrid transformations that are available by default, see the description of the **externalsourcename** column attribute in [Reference - UniGrid definition](#).

If you need to display column values in a custom format, you can register your own UniGrid transformations:

1. Open your Kentico project in Visual Studio.
2. Create a [custom module class](#).
 - Either add the class into a custom project within the Kentico solution (recommended) or directly into the Kentico web project (into a custom folder under the **CMSApp** project for *web application* installations, into the **App_Code** folder for *web site* installations).
3. Override the module's **OnInit** method and register your UniGrid transformations.
 - Call the **UniGridTransformations.Global.RegisterTransformation** method for each transformation.
4. Implement the methods that define the functionality of your custom UniGrid transformations.

 For basic execution of initialization code, you only need to register a "code-only" module through the API. You do NOT need to create a new module within the **Modules** application in the Kentico administration interface.

Example

```
using CMS;
using CMS.Base;
using CMS.DataEngine;
using CMS.UIControls;
using CMS.Helpers;
using CMS.PortalEngine;

// Registers the custom module into the system
[assembly: RegisterModule(typeof(CustomUniGridTransformationModule))]

public class CustomUniGridTransformationModule : Module
{
    // Module class constructor, the system registers the module under the name
    "CustomUniGridTransformations"
    public CustomUniGridTransformationModule()
        : base("CustomUniGridTransformations")
    {
    }

    // Contains initialization code that is executed when the application starts
    protected override void OnInit()
    {
        base.OnInit();

        // Registers the #customBoldText UniGrid transformation
        UniGridTransformations.Global.RegisterTransformation
        ("#customBoldText", CustomBoldText);

        // Registers the #isStyleSheetCheckedOut UniGrid transformation
        UniGridTransformations.Global.RegisterTransformation
        ("#isStyleSheetCheckedOut", IsStyleSheetCheckedOut);
    }

    // Method that defines the #customBoldText UniGrid transformation
    // Wraps the UniGrid column text value into a <strong> element
    private static object CustomBoldText(object parameter)
```



```
        {
            return "<strong>" + ValidationHelper.GetString(parameter, "") + "<
/strong>";
        }

        // Method that defines the #isCheckedOut UniGrid transformation
        // Indicates whether the specified stylesheet object is currently checked out
        for editing
        private static object IsStyleSheetCheckedOut(object parameter)
        {
            // Gets the stylesheet object based on the ID in the parameter
            CssStyleSheetInfo stylesheet = CssStyleSheetInfoProvider.
GetCssStyleSheetInfo(ValidationHelper.GetInteger(parameter, 0));

            // Evaluates whether the stylesheet is checked out
            if ((stylesheet != null) && stylesheet.Generalized.IsCheckedOut)
            {
                return "<span class=\"StatusEnabled\">" + ResHelper.GetString
("general.yes") + "</span>";
            }

            return "<span class=\"StatusDisabled\">" + ResHelper.GetString
("general.no") + "</span>";
        }
    }
```

Once a UniGrid transformation is registered, you can use it in the **externalsourcename** attribute of UniGrid columns.

Examples:

```
...

<GridColumn>
    <ug:Column Source="UserName" Caption="$general.username$" Width="100%"
ExternalSourceName="#customBoldText" />
</GridColumn>

...
```

```
...

<columns>
    <column source="StyleSheetID" caption="Checked out" wrap="false"
externalsourcename="#isStyleSheetCheckedOut" />
</columns>

...
```