



By default, Kentico uses page type IDs to categorize products in the Strands integration – both in the Strands catalog feed and the [Strands recommendations web part](#). To fully customize the categorization in both places, you need to create a custom implementation of the **IStrandsCatalogCategoryMapper** interface. By customizing the categorization, you can have it better represent the structure of your site.

This page describes how to create a custom class in which you perform such a customization. The page also provides two sample customizations for you to build on.

Implementing custom Strands product categorization

1. Open your Kentico web project in Visual Studio (using the **WebSite.sln** or **WebApp.sln** file).
2. Create a new class, for example named **CustomStrandsCatalogCategoryMapper.cs**.
 - Either add the class into a custom code library 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).



When using a custom project (assembly), you need to perform the following steps to ensure that the system can detect your custom interface implementations:

- a. Edit your custom assembly's **AssemblyInfo.cs** file (in the *Properties* folder).
- b. Add the **AssemblyDiscoverable** assembly attribute (requires a reference to *CMS.Core.dll*):

```
using CMS;

[assembly:AssemblyDiscoverable]
```

- c. Save the file and **Build** the custom project.

3. Make the class implement the **IStrandsCatalogCategoryMapper** interface:

```
using System;
using System.Globalization;

using CMS;
using CMS.Core;
using CMS.StrandsRecommender;
using CMS.DocumentEngine;
using CMS.Ecommerce;

// Registers the custom IStrandsCatalogCategoryMapper implementation
[assembly: RegisterImplementation(typeof(IStrandsCatalogCategoryMapper), typeof(
CustomStrandsCatalogCategoryMapper))]

public class CustomStrandsCatalogCategoryMapper : IStrandsCatalogCategoryMapper
{
    public string GetItemCategory(TreeNode catalogItem)
    {
        // Body of your Strands catalog categorization customization
    }
}
```

4. Add a custom implementation for the **GetItemCategory** method, as described in one of the following examples:
 - a. **Using an SKU department ID as the category field in the Strands catalog feed**



```
/// <summary>
/// Retrieves SKUDepartmentID which will be used for categorizing the page
/// in the Strands integration.
/// </summary>
/// <param name="catalogItem">Document</param>
/// <returns>Category field</returns>
public string GetItemCategory(TreeNode catalogItem)
{
    SKUInfo skuInfo = SKUInfoProvider.GetSKUInfo(catalogItem.NodeSKUID);
    if (skuInfo != null)
    {
        return skuInfo.SKUDepartmentID.ToString(CultureInfo.
InvariantCulture);
    }
    throw new Exception("[CustomStrandsCategoryCatalogMapper.
GetItemCategory]: SKUInfo was not found for specified TreeNode.");
}
```

b. Using the path to the page as the category field in the Strands catalog feed

```
/// <summary>
/// Retrieves the path to the page, excluding the page's name. The path
/// will be used for categorizing the page in the Strands integration.
/// </summary>
/// <param name="catalogItem">Document</param>
/// <returns>Category field</returns>
public string GetItemCategory(TreeNode catalogItem)
{
    string docName = catalogItem.DocumentName;
    string docNamePath = catalogItem.DocumentNamePath;

    return docNamePath.Substring(0, docNamePath.Length - docName.Length);
}
```

5. **Save** the class file and build your solution.

Now that you have customized the *GetItemCategory* method, view your Strands catalog feed. You should see the system categorizing products in the feed according to the new customization.

- You can view your Strands catalog feed at: `~/CMSModules/StrandsRecommender/CMSPages/StrandsCatalogFeed.ashx`
- The same categorization is now used in the [Strands recommendations web part](#) in the **Category** recommendation type.