


By assigning custom handlers to the **OnBeforeFiltering** and **OnAfterFiltering** events, you can make changes to the output HTML code. The events must be connected to the processing of the requests on which you want to make the modifications.

 **Note:** Output customizations can significantly affect site performance.

[Assign a handler](#) to the **OutputFilterContext.CurrentFilter.OnBeforeFiltering** or **OnAfterFiltering** event.

- The example uses a [custom module class](#) to register the event handlers.
- The example makes use of the *PostMapRequestHandler.Execute* event to create an output filter instance on every request.

```
using System;

using CMS;
using CMS.DataEngine;
using CMS.Base;
using CMS.OutputFilter;

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

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

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

        // Ensures that the an output filter instance is created on every request
        RequestEvents.PostMapRequestHandler.Execute +=
        PostMapRequestHandler_Execute;
    }

    private void PostMapRequestHandler_Execute(object sender, EventArgs e)
    {
        // Creates an output filter instance
        ResponseOutputFilter.EnsureOutputFilter();

        // Assigns a handler to the OutputFilterContext.CurrentFilter.
        OnAfterFiltering event
        OutputFilterContext.CurrentFilter.OnAfterFiltering +=
        CurrentFilter_OnAfterFiltering;
    }

    private void CurrentFilter_OnAfterFiltering(ResponseOutputFilter filter, ref
    string finalHtml)
    {
        // Process the resulting HTML
    }
}
```

Each request will now be processed and the output HTML will be modified by the **CurrentFilter\_OnAfterFiltering** method.