In addition to the default performance counters, you can also implement custom performance counters to monitor other values.

The following example demonstrates how to implement a custom performance counter that monitors the number of accesses to the **Home** page of any website running in the system. The counter will be available in both the **General** and the **Sites** counter categories, enabling you to monitor access to the **Home** page of each individual website or globally for all websites in the system.

1. Navigate to the ~*App_Data**CMSModules**HealthMonitoring* folder (or any other folder under ~*App_Data**CMSModules*\).
2. Create a new XML file with a **.xpc** extension, for example, *MyCounters.xpc*.
3. Copy the following code into the file:

```
<?xml version="1.0" encoding="utf-8"?>
<Counters>

 <Counter Key="requestshomepage" Name="Home page requests"
Description="The number of Home page requests."
Type="NumberOfItems32" Enabled="True" OnlyGlobal="False" />
</Counters>
```

4. Open your Kentico web project in Visual Studio (using the **WebSite.sln** or **WebApp.sln** file).
5. Create a custom module class. For example, name the class **CustomCounterModule.cs**.
   - 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).

```
using CMS;
using CMS.Core;
using CMS.DataEngine;
using CMS.HealthMonitoring;

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

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

    // Counter of total Home page requests
    private static IPerformanceCounter mTotalHomePageRequests = null;

    // Counter of total Home page requests
    public static IPerformanceCounter TotalHomePageRequests
    {
        get
        {
            if (mTotalHomePageRequests == null)
            {
                mTotalHomePageRequests = Service.Resolve<IPerformanceCounter>();
            }

            return mTotalHomePageRequests;
        }
    }

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

        // Assigns a handler to the OnLogCustomCounter event
        HealthMonitoringLogHelper.OnLogCustomCounter +=
HealthMonitoringLogHelper_OnLogCustomCounter;
    }

    private static IPerformanceCounter
HealthMonitoringLogHelper_OnLogCustomCounter(Counter counter)
    {
        if (counter.Key.ToLower() == "requestshomepage")
        {
            return TotalHomePageRequests;
        }

        return null;
    }
}
```

> The private variable *mTotalHomePageRequests* holds the value logged in the counter. The value is made accessible by the *TotalHomePageRequests* public property.
>
> The code in the override of the *OnInit()* method assigns the *HealthMonitoringLogHelper_OnLogCustomCounter* method as a handler for the *OnLogCustomCounter* event.

6. Add the *OnLoad* method to ~*CMSPages**PortalTemplate.aspx.cs*. The code ensures that the counter value is incremented each time a page with the */home* alias path is accessed.

```
protected override void OnLoad(EventArgs e)
{
        // Increments the Home page requests counter
        if (CMS.DocumentEngine.DocumentContext.CurrentPageInfo.NodeAliasPath.
Equals("/home", StringComparison.InvariantCultureIgnoreCase))
        {
                CustomCounterModule.TotalHomePageRequests.Increment(CMS.
SiteProvider.SiteContext.CurrentSiteName);
        }

        base.OnLoad(e);
}
```

7. Register counters manually by executing the Health Monitoring Windows service with the appropriate parameters, as explained in Registering performance counters.
    - Open the Windows command line, navigate to the **Bin** folder inside the Kentico installation folder (typically *C:\Program Files\Kentico\<version>\Bin*) and execute the **HealthMonitoringService.exe** file with the following parameters:

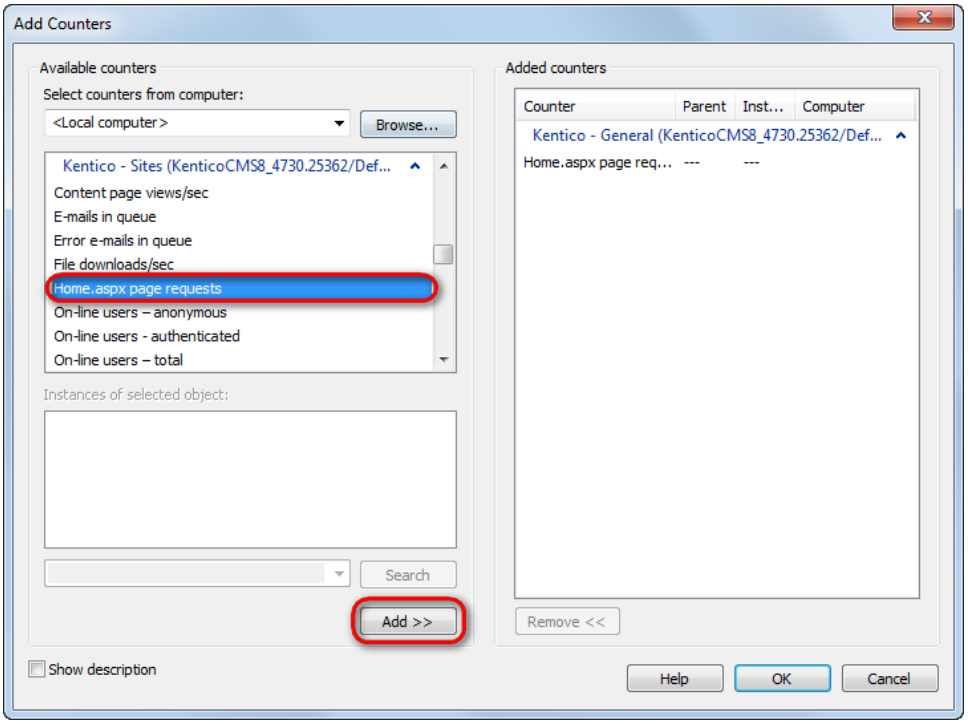
```
HealthMonitoringService.exe /webpath=<disk path to web project root>
/createcounters
```

This action reloads all counters in both categories, including the newly added one.

> **Custom counters and Health Monitoring Windows service**
>
> Values of all custom counters can be logged only by the application itself, i.e. it is not possible for values of custom counters to be logged by the Windows service. The only purpose of using the service in this step is to register the added counter, as described in Registering performance counters.

To display the new counter values in the Performance monitor:

1. Make sure that the **Enable health monitoring** setting is enabled in Kentico (**Settings -> System -> Health monitoring**).
2. Launch the Performance monitor (type *perfmon* in Windows Start menu search box and press *Enter*).
3. Click **Add.**
    - The new counter **Home page requests** should be present in both counter categories of the current Kentico instance.
4. Add the counters from both categories (**General** and **Sites**) using the **Add >>** button.



5. Click **OK**.
6. In the Kentico administration interface, open the **System** application and click **Restart application**.
7. Now try accessing the **Home** page multiple times and see how the value gets incremented after each access.