



If your project uses an **Azure Blob Storage** file system (Azure projects deployed to Cloud Services by default or other projects with the `<add key="CMSExternalStorageName" value="azure"/>` web.config key), the system stores [web analytics](#) logs on the blob storage. The web analytics functionality writes and processes logs every minute, which can cause latency problems when using blob storage. Additionally, web analytics may create a large number of files, which further degrades the performance of the blob storage.

We highly recommend that you map the folder storing web analytics data to the local file system of the server:

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 use the following code to map the Web analytics storage to the local file system (in `~/App_Data/CMSModules/WebAnalytics`). See [Configuring file system providers](#) for more information.



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.

```
using CMS;
using CMS.Base;
using CMS.DataEngine;
using CMS.IO;

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

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

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

        // Creates a new StorageProvider instance for the Windows file
        system
        var webAnalyticsProvider = StorageProvider.
        CreateFileSystemStorageProvider();

        // Maps the web analytics directory to the provider
        StorageHelper.MapStoragePath("~/App_Data/CMSModules/WebAnalytics",
        webAnalyticsProvider);
    }
}
```

4. Save the file. If your project is installed in the web application format, rebuild the solution.
5. If your project is deployed to Azure, re-deploy your project.

The system now stores web analytics log files on the server's file system instead of the blob storage.