Ting Huang

© 2011, All Rights Reserved.  
<http://www.codeplex.com/MvcScriptManager>

[About MvcScriptManager 2](#_Toc244357240)

[Features 2](#_Toc244357241)

[What MvcScriptManager Won't Offer 2](#_Toc244357242)

[How to Use MvcScriptManager 3](#_Toc244357243)

[Download MvcScriptManager 3](#_Toc244357244)

[Reference MvcScriptManager in Your Project 3](#_Toc244357245)

[Update web.config and Global.asax 3](#_Toc244357246)

[Add MvcScriptManager to the Views 4](#_Toc244357247)

[Localized Resource for Script 4](#_Toc244357248)

[Documentation 5](#_Toc244357249)

Last Update: 02/03/2011

About MvcScriptManager

MvcScriptManager is aimed to port certain key features available in AjaxControlToolkit's ToolkitScriptManager into the current ASP.NET MVC Framework. You will be able to use it as a control in your ASP.NET MVC application.

# Features

* Script combining. Scripts declared with MvcScriptManager will be dynamically combined into a single script file request when the page is rendered. This behavior can be disabled for debugging purposes.
* Render localized resources for stand-alone script files. Localized strings will be appended to the script if specified.
* Support script crunching in release mode. Crunching can be enabled/disabled for each script.
* Support configurable HTTP compression and expiration setting when outputing scripts.
* Script content caching with file dependency. Script file content is cached so that rendering combined script file will be much more performant. Cache dependency is linked to the physical file therefore any script update in the file system will be reflected in the cache instantly.
* Support rendering scripts in debug/release mode based on the running environment.
* Resolving different paths for stand-alone script files.
* Support multiple MvcScriptManager's on a single page (or master page). Support both Master and Slave rendering modes:
  + Master rendering mode – will render <Script/> tag, and include all scripts from itself and those from its slave(s).
  + Slave rendering mode – will not render anything. But appends current instance’s scripts to preceding Master instance. This property is useful, especially when you have one or more ScriptManagers declared in the master page(s), and you are declaring another ScriptManager in the specific (non-master) page to add more scripts, but you want to hand over the rendering to the ScriptManager inside the master page only. In this case, you can set the ScriptManager's RenderMode in master page to Master, and the ScriptManager's RenderMode in the specific page to Slave.
* Support web farm scenario.

# What MvcScriptManager Won't Offer

Below are the things MvcScriptManager **doesn't (and probably won't) offer** in case you are used to using the ScriptManager and assume these features are included:

* Support for ASP.NET Ajax extensibility model. In my opinion, with ASP.NET MVC Framework and jQuery, writing ajax script/extender controls is no longer necessary.
* Support for embedded script files. We should avoid embedding script files in assembly anyway because updates of script will require a rebuild of the project which is not ideal.
* Provide access to web methods from script by registering web services. This is overhead and you can totally avoid it by either using jQuery’s AJAX utilities or Sys.Net.WebServiceProxy class available in ASP.NET AJAX client library.
* Support UpdatePanel and partial-page Rendering.

How to Use MvcScriptManager

# Download MvcScriptManager

Download MvcScriptManager binary or source from:  
<http://www.codeplex.com/MvcScriptManager>

# Reference MvcScriptManager in Your Project

You can either reference the MvcScriptManager.dll or the source project in your web project.

# Update web.config and Global.asax

Add the mvc tag prefix in the web.config:

<system.web>

<pages>

<controls>

<add tagPrefix="mvc" namespace="MvcScriptManager" assembly="MvcScriptManager, Version=1.0.0.0, Culture=neutral, PublicKeyToken=6eb4f344e8972dc6"/>

</controls>

</pages>  
</system.web>

If using IIS 7.0 or later, add the CombineScriptHandler entry in the web.config:

<system.webServer>

<handlers>

<add name="CombineScriptHandler" verb="\*" path="CombineScriptHandler.aspx" preCondition="integratedMode" type="MvcScriptManager.CombineScriptHandler, MvcScriptManager, Version=1.0.0.0, Culture=neutral, PublicKeyToken=6eb4f344e8972dc6"/>

</handlers>

</system.webServer>

For compatibility with IIS 6.0, add the following to the web.config:

<system.web>

<httpHandlers>

<add verb="\*" path="CombineScriptHandler.aspx" type="MvcScriptManager.CombineScriptHandler, MvcScriptManager, Version=1.0.0.0, Culture=neutral, PublicKeyToken=6eb4f344e8972dc6"/>

</httpHandlers>

</system.web>

Add the following line of code in global.asax in your web project (please see the global.asax in the sample web project for reference) in the RegisterRoutes method:

routes.IgnoreRoute("CombineScriptHandler.aspx/{\*pathInfo}");

The web.config in the sample web project in MvcScriptManager source is an example of how these are set up.

# Add MvcScriptManager to the Views

Insert MvcScriptManager onto the view pages:

<mvc:ScriptManager runat="server">

<Scripts>

<mvc:ScriptReference Path="~/Scripts/jquery-1.2.6.js" />

<mvc:ScriptReference Path="~/Scripts/Sample.js" EnableCrunching="true" ResourceAssembly="SampleWeb" ResourceName="SampleWeb.Sample" ResourceTypeName="Sample" />

</Scripts>

</mvc:ScriptManager>

The /Views/Home/Index.aspx in the sample web project shows how to add the MvcScriptManager.

# Localized Resource for Script

When ScriptReference is specified with a ResourceAssembly value, the target assembly that the ResourceAssembly value refer to should contain the localized resource for the script. You will need to specify ResourceName and ResourceTypeName in order for the resource values to be appended to the script when it is rendered. See the documentation below for detail explanations of each property.

Documentation

|  |  |
| --- | --- |
| **ScriptManager** | |
| **Property** | **Description** |
| Scripts | A collection of **ScriptReference**s. |
| CombineScriptsHandlerUrl | The URL of the HTTP handler which generates the combined script file. Default value is “~/CombineScriptHandler.aspx”. |
| DisableCombiningWhileDebugging | Controls whether scripts are combined when debug compilation is enabled in web.config. This is useful when debugging JavaScript, as large combined JavaScript files can be hard to navigate.  Default is true.  Set false to combine the scripts in debug compilations. |
| EnableHttpCompression | A value indicating whether HTTP compression should be enabled when outputting scripts. Default is true.  Set to false if you can enable HTTP compression on the web server or the load balancer. |
| EnableHttpExpiration | A value indicating whether HTTP expiration header should be enabled when outputting scripts. Default is true.  Set to false if you can enable HTTP expiration header on the web server or the load balancer. |
| HttpExpirationInDays | Sets the HTTP expiration in days.  Default is 365.  This setting will only be in effect when EnableHttpExpiration is set to true. |
| ScriptToken | A value to be appended to query string when generating combined script URL.  When set, it will force the previous browser cache on the script to expire under release/production mode. Note that under debug mode, no need to add this setting because a built-in debug param is added automatically to each combined script request. |
| RenderMode | Sets the rendering mode for the current ScriptManager instance.  Master - the current instance will render <Script /> tag.  Slave - the current instance will not render anything. Instead, it will serve as a container for appending new scripts to the preceding Master. The default value is Master. |

(Continue to next page)

|  |  |
| --- | --- |
| **ScriptReference** | |
| **Property** | **Description** |
| Path | The virtual path of the script. For example, “~/Scripts/Sample.js”. |
| ResourceAssembly | The assembly string representing an assembly where the script’s localized resource resides in. For example, “CompanyA.SampleWeb”. |
| ResourceName | The name of the resource file where the script localization resource resides in. |
| ResourceTypeName | The name of the type to create for the values in the resource file. |
| EnableCrunching | A value indicating whether the script content should be crunched when it is rendered. Default is false. |
| UniqueKeyEncodingScheme | The encoding scheme used for the file reference keys. Default value is TwoWayEncoding, which will support web farm deployment. If your MVC application is hosted on one single web server, set the value to OneWayEncoding. |