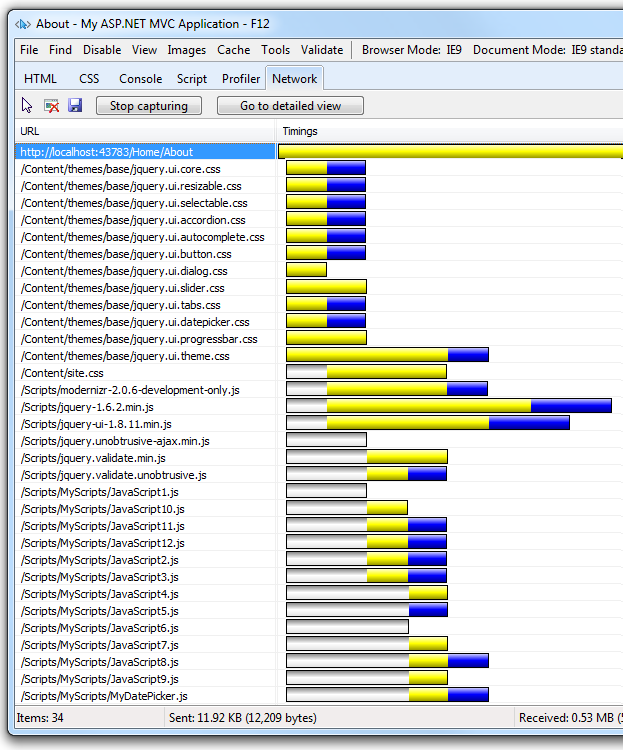
**Bundling and Minification**

Bundling and minification are two techniques you can use in ASP.NET and Asp.net MVCvmmb to improve request load time. Bundling and minification improves load time by reducing the number of requests to the server and reducing the size of requested assets (such as CSS and JavaScript.)

Most of the current major browsers limit the number of [simultaneous connections](http://www.browserscope.org/?category=network) per each hostname to six. That means that while six requests are being processed, additional requests for assets on a host will be queued by the browser. In the image below, the IE F12 developer tools network tabs shows the timing for assets required by the About view of a sample application.

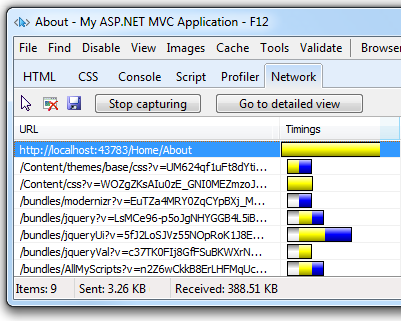


The gray bars show the time the request is queued by the browser waiting on the six connection limit. The yellow bar is the request time to first byte, that is, the time taken to send the request and receive the first response from the server. The blue bars show the time taken to receive the response data from the server.

## Bundling

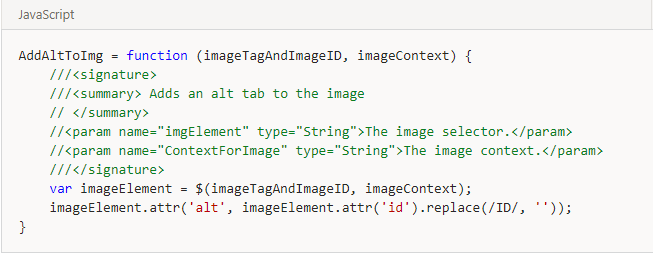
Bundling is a new feature in ASP.NET 4.5 that makes it easy to combine or bundle multiple files into a single file. You can create CSS, JavaScript and other bundles. Fewer files means fewer HTTP requests and that can improve first page load performance.

The following image shows the same timing view of the About view shown previously, but this time with bundling and minification enabled.



## Minification

Minification performs a variety of different code optimizations to scripts or css, such as removing unnecessary white space and comments and shortening variable names to one character. Consider the following JavaScript function.



After minification, the function is reduced to the following:

AddAltToImg = function (n, t) { var i = $(n, t); i.attr("alt", i.attr("id").replace(/ID/, "")) }

In addition to removing the comments and unnecessary whitespace, the following parameters and variable names were renamed (shortened) as follows:

| **Original** | **Renamed** |
| --- | --- |
| imageTagAndImageID | n |
| imageContext | t |
| imageElement | i |

## Impact of Bundling and Minification

The following table shows several important differences between listing all the assets individually and using bundling and minification (B/M) in the sample program.

|  | **Using B/M** | **Without B/M** | **Change** |
| --- | --- | --- | --- |
| **File Requests** | 9 | 34 | 256% |
| **KB Sent** | 3.26 | 11.92 | 266% |
| **KB Received** | 388.51 | 530 | 36% |
| **Load Time** | 510 MS | 780 MS | 53% |

## Controlling Bundling and Minification

Bundling and minification is enabled or disabled by setting the value of the debug attribute in the [compilation Element](https://msdn.microsoft.com/en-us/library/s10awwz0.aspx) in the Web.config file. In the following XML, debug is set to true so bundling and minification is disabled.

<system.web>

<compilation debug="true" />

<!-- Lines removed for clarity. -->

</system.web>

To enable bundling and minification, set the debug value to "false". You can override the *Web.config* setting with the EnableOptimizations property on the BundleTable class.

The following code enables bundling and minification and overrides any setting in the *Web.config* file.

public static void RegisterBundles(BundleCollection bundles)

{

bundles.Add(new ScriptBundle("~/bundles/jquery").Include(

"~/Scripts/jquery-{version}.js"));

// Code removed for clarity.

BundleTable.EnableOptimizations = true;

}

## Bundling and Minification with ASP.NET MVC

Open the *App\_Start\BundleConfig.cs* file and examine the RegisterBundles method which is used to create, register and configure bundles. The following code shows a portion of the RegisterBundles method.

public static void RegisterBundles(BundleCollection bundles)

{

bundles.Add(new ScriptBundle("~/bundles/jquery").Include(

"~/Scripts/jquery-{version}.js"));

// Code removed for clarity.

}

The preceding code creates a new JavaScript bundle named ~/bundles/jquery that includes all the appropriate (that is debug or minified but not .vsdoc) files in the Scripts folder that match the wild card string "~/Scripts/jquery-{version}.js".

The {version} wild card matching shown above is used to automatically create a jQuery bundle with the appropriate version of jQuery in your *Scripts* folder. In this example, using a wild card provides the following benefits:

* Allows you to use NuGet to update to a newer jQuery version without changing the preceding bundling code or jQuery references in your view pages.
* Automatically selects the full version for debug configurations and the ".min" version for release builds.

## Creating a Bundle

The [Bundle](https://msdn.microsoft.com/en-us/library/system.web.optimization.bundle(v=VS.110).aspx) class Include method takes an array of strings, where each string is a virtual path to resource. The following code from the RegisterBundles method in the App\_Start\BundleConfig.csfile shows how multiple files are added to a bundle

bundles.Add(new StyleBundle("~/Content/themes/base/css").Include(

"~/Content/themes/base/jquery.ui.core.css",

"~/Content/themes/base/jquery.ui.resizable.css",

"~/Content/themes/base/jquery.ui.selectable.css",

"~/Content/themes/base/jquery.ui.accordion.css",

"~/Content/themes/base/jquery.ui.autocomplete.css",

"~/Content/themes/base/jquery.ui.button.css",

"~/Content/themes/base/jquery.ui.dialog.css",

"~/Content/themes/base/jquery.ui.slider.css",

"~/Content/themes/base/jquery.ui.tabs.css",

"~/Content/themes/base/jquery.ui.datepicker.css",

"~/Content/themes/base/jquery.ui.progressbar.css",

"~/Content/themes/base/jquery.ui.theme.css"));

The [Bundle](https://msdn.microsoft.com/en-us/library/system.web.optimization.bundle(v=VS.110).aspx) class IncludeDirectory method is provided to add all the files in a directory (and optionally all subdirectories) which match a search pattern.

Bundles are referenced in views using the Render method, ( Styles.Render for CSS and Scripts.Render for JavaScript). The following markup from the *Views\Shared\\_Layout.cshtml* file shows how the default ASP.NET internet project views reference CSS and JavaScript bundles.

<!DOCTYPE html>

<html lang="en">

<head>

@\* Markup removed for clarity.\*@

@Styles.Render("~/Content/themes/base/css", "~/Content/css")

@Scripts.Render("~/bundles/modernizr")

</head>

<body>

@\* Markup removed for clarity.\*@

@Scripts.Render("~/bundles/jquery")

@RenderSection("scripts", required: false)

</body>

</html>

Notice the Render methods takes an array of strings, so you can add multiple bundles in one line of code. You will generally want to use the Render methods which create the necessary HTML to reference the asset.

Using the "\*" Wildcard Character to Select Files

The virtual path specified in the Include method and the search pattern in the IncludeDirectory method can accept one "\*" wildcard character as a prefix or suffix to in the last path segment. The search string is case insensitive. The IncludeDirectory method has the option of searching subdirectories.

Consider a project with the following JavaScript files:

* *Scripts\Common\AddAltToImg.js*
* *Scripts\Common\ToggleDiv.js*
* *Scripts\Common\ToggleImg.js*
* *Scripts\Common\Sub1\ToggleLinks.js*

The following table shows the files added to a bundle using the wildcard as shown:

| **Call** | **Files Added or Exception Raised** |
| --- | --- |
| Include("~/Scripts/Common/\*.js") | *AddAltToImg.js, ToggleDiv.js, ToggleImg.js* |
| Include("~/Scripts/Common/T\*.js") | Invalid pattern exception. The wildcard character is only allowed on the prefix or suffix. |
| Include("~/Scripts/Common/\*og.\*") | Invalid pattern exception. Only one wildcard character is allowed. |
| "Include("~/Scripts/Common/T\*") | *ToggleDiv.js, ToggleImg.js* |
| "Include("~/Scripts/Common/\*") | Invalid pattern exception. A pure wildcard segment is not valid. |
| IncludeDirectory("~/Scripts/Common", "T\*") | *ToggleDiv.js, ToggleImg.js* |
| IncludeDirectory("~/Scripts/Common", "T\*",true) | *ToggleDiv.js, ToggleImg.js, ToggleLinks.js* |