ASP.NET Master Pages in Visual Studio .Net

ASP.NET master pages allow you to create a consistent layout and consistent behavior for many pages in your web application. A master page (Site.master.aspx) defines the common HTML areas and it’s code behind file (Site.master.aspx.cs) can define common behavior. This document deals only with the common HTML areas in the aspx files. A master page is a fully formed html page (starting with <html>, ending with </html>, containing all the code that is common for many pages (e.g., title, navigation bar, footer), and it has “placeholders” for areas that will be different page to page (like the content). There is an example master page on the next page.

Once you have a master page, you can create many “regular” pages such that each page
(1) references the master page and
(2) provides page specific data, like page content.

When the user requests a page, Dot Net finds the requested page and the master page it references, then delivers these two pages, properly merged, to the browser.

When you create a New Web Site in Visual Studio, you get a Solution with a Single Project like this. A default master page is stored in Site.master. They set it up so that About.aspx references that master page.

![Solution Explorer](image)

The default Site.master file that they provide is a little too complex for educational purposes, so I simplified it in the example on the next page. You also should edit your Site.master to be the way you want your pages to look.

Note: you may also find information about using Master Pages in Microsoft Visual Studio from this link (I have not followed this tutorial, but it looks about right):

Example:

<table>
<thead>
<tr>
<th>Site.master</th>
<th>About.aspx</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;!DOCTYPE html&gt;</code></td>
<td><code>&lt;asp:Content runat=&quot;server&quot; ContentPlaceHolderID=&quot;HeadContent&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;html&gt;</code></td>
<td><code>&lt;style type=&quot;text/css&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;/html&gt;</code></td>
<td><code>body {</code></td>
</tr>
<tr>
<td><code>&lt;head runat=&quot;server&quot;&gt;</code></td>
<td><code>background-color: Teal;</code></td>
</tr>
<tr>
<td><code>&lt;/head&gt;</code></td>
<td><code>.myContent {</code></td>
</tr>
<tr>
<td><code>&lt;form runat=&quot;server&quot;&gt;</code></td>
<td><code>border: 2px solid gray;</code></td>
</tr>
<tr>
<td><code>&lt;/form&gt;</code></td>
<td><code>padding: 10px;</code></td>
</tr>
<tr>
<td><code>&lt;/body&gt;</code></td>
<td><code>}&lt;/style&gt;</code></td>
</tr>
<tr>
<td><code>&lt;/html&gt;</code></td>
<td><code>&lt;/asp:Content&gt;</code></td>
</tr>
<tr>
<td><code>&lt;/html&gt;</code></td>
<td><code>&lt;asp:Content runat=&quot;server&quot; ContentPlaceHolderID=&quot;MainContent&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;h1&gt;</code></td>
<td><code>&lt;p&gt;</code></td>
</tr>
<tr>
<td><code>My ASP.NET Application</code></td>
<td><code>Here's the content for my About Page.</code></td>
</tr>
<tr>
<td><code>&lt;/h1&gt;</code></td>
<td><code>&lt;/p&gt;</code></td>
</tr>
<tr>
<td><code>&lt;div class=&quot;myContent&quot;&gt;</code></td>
<td><code>&lt;/div&gt;</code></td>
</tr>
<tr>
<td><code>&lt;/asp:ContentPlaceHolderID=&quot;MainContent&quot; runat=&quot;server&quot;&gt;</code></td>
<td><code>&lt;/h4&gt;</code></td>
</tr>
<tr>
<td><code>&lt;/div&gt;</code></td>
<td><code>&lt;/h4&gt;</code></td>
</tr>
<tr>
<td><code>&lt;/asp:ContentPlaceHolderID=&quot;HeadContent&quot; runat=&quot;server&quot;&gt;</code></td>
<td><code>&lt;/form&gt;</code></td>
</tr>
<tr>
<td><code>&lt;/asp:ContentPlaceHolderID=&quot;HeadContent&quot;&gt;</code></td>
<td><code>&lt;/body&gt;</code></td>
</tr>
<tr>
<td><code>&lt;/form&gt;</code></td>
<td><code>&lt;/html&gt;</code></td>
</tr>
</tbody>
</table>

Merged Result in Browser

```
<!DOCTYPE html>
<html>
<head><title>About Us</title>
  <style type="text/css">
    body {
      background-color: Teal;
      .myContent {
        border: 2px solid gray;
        padding: 10px;
      }
    }
  </style>
</head>
<body>
  <h1>My ASP.NET Application</h1>
  <div class="myContent">
    Here's the content for my About Page.
  </div>
  <h4>My Little Footer</h4>
</body>
</html>
```

“View Source” of Browser Page - Shows Merged Code

```html
<!DOCTYPE html>
<html>
<head><title>About Us</title>
  <style type="text/css">
    body {
      color: navy;
    }
  </style>
</head>
<body method="post" action="About.aspx" id="ctl01">
  <h1>My ASP.NET Application</h1>
  <div class="myContent">
    Here's the content for my About Page.
  </div>
  <h4>My Little Footer</h4>
</body>
</html>```
Let's say you want to add a new page to your web application and you want the new page to reference the Site.master, because, of course, you want a consistent look and feel to your whole site. You would do this:

- Right Click your Project Name (in the Solution Explorer Pane) and
- select “Add New Item” – “Web Form”.
- Before clicking the “Add” button, remember to rename your page (e.g., labs.aspx)

The code for the new page (**labs.aspx**) would be a plain complete page with no reference to the master page, like this:

```html
%@ Page Language="C#" AutoEventWireup="true" CodeFile="labs.aspx.cs" Inherits="labs" %
<!DOCTYPE html>
<html>
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>
</div>
</form>
</body>
</html>
```

1. Add the “MasterPageFile” reference (as shown highlighted **yellow**). You can also add a **page title** if you like.
2. Delete everything else, from `<html>` to `</html>`.
3. Insert the references to the placeholders (as shown highlighted **gray**) and put your page specific items inside (e.g., anything special in the header, the page specific content).

```html
%@ Page Language="C#" AutoEventWireup="true" CodeFile="labs.aspx.cs" Inherits="labs"
MasterPageFile="/Site.master" Page Title="My Labs" %
<asp:Content runat="server" ContentPlaceHolderID="HeadContent">
<style type="text/css"> /* page specific style */
body { color: red; }
</style>
</asp:Content>
<asp:Content runat="server" ContentPlaceHolderID="MainContent">
<p>Here's the content for my Labs Page.</p>
</asp:Content>
```

When you “View in Browser” labs.aspx, you will see this in your browser: elements common to all pages are from the master page, but page specific elements (e.g., the content area, possibly in the header area) are pulled from labs.aspx.

---

**My ASP.NET Application**

- Here’s the content for my Labs Page.

**My Little Footer**
Master pages and Relative Paths

One quirk is the way Master Pages handle relative paths. For instance, say, that the master page is in a subfolder named MasterPages, and this image tag is in the master page

```
<img src="banner.jpg">
```

If the image lies in the MasterPages subfolder, this works fine for the master page. But say it won’t work fine for the webform (at the root) that references the Master Page.

One quick fix is to make the image tag into a server side control

```
<img src="banner.jpg" runat="server">
```

Or, we can use the root path syntax

```
<img src="/MasterPages/banner.jpg" runat="server">
```

Applying Master Pages through a Configuration File

If we do not want to have to specify the master page reference on each of our web pages, we can apply the master page for all the pages in the website at once using the web.config file. All we need to do is

```
<configuration>
  <system.web>
    <pages masterPageFile ="Site1.master"/>
  </system.web>
</configuration>
```

![Solution Explorer](image)

Good references:

