**ASP.NET Themes and Skins Overview**

This page is specific to

**Microsoft Visual Studio 2008/.NET Framework 3.5**

Other versions are also available for the following:

[Microsoft Visual Studio 2005/.NET Framework 2.0](http://msdn.microsoft.com/en-us/library/ykzx33wh(VS.80).aspx)

[.NET Framework 3.0](http://msdn.microsoft.com/en-us/library/ykzx33wh(VS.85).aspx)

[Microsoft Visual Studio 2010/.NET Framework 4.0](http://msdn.microsoft.com/en-us/library/ykzx33wh(VS.100).aspx)

A theme is a collection of property settings that allow you to define the look of pages and controls, and then apply the look consistently across pages in a Web application, across an entire Web application, or across all Web applications on a server.

A set of example ASP.NET themes is also available: [Download](http://go.microsoft.com/fwlink/?LinkId=157239).

http://i.msdn.microsoft.com/Global/Images/clear.gif Themes and Control Skins

Themes are made up of a set of elements: skins, cascading style sheets (CSS), images, and other resources. At a minimum, a theme will contain skins. Themes are defined in special directories in your Web site or on your Web server.

**Skins**

A skin file has the file name extension .skin and contains property settings for individual controls such as [Button](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.button.aspx), [Label](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.label.aspx), [TextBox](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.textbox.aspx), or [Calendar](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.calendar.aspx) controls. Control skin settings are like the control markup itself, but contain only the properties you want to set as part of the theme. For example, the following is a control skin for a [Button](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.button.aspx) control:

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl46_ctl00_ctl05');" \o "Copy Code)

<asp:button runat="server" BackColor="lightblue" ForeColor="black" />

You create .skin files in the Theme folder. A .skin file can contain one or more control skins for one or more control types. You can define skins in a separate file for each control or define all the skins for a theme in a single file.

There are two types of control skins, def*ault skins* and *named skins*:

* A default skin automatically applies to all controls of the same type when a theme is applied to a page. A control skin is a default skin if it does not have a **SkinID** attribute. For example, if you create a default skin for a [Calendar](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.calendar.aspx) control, the control skin applies to all [Calendar](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.calendar.aspx) controls on pages that use the theme. (Default skins are matched exactly by control type, so that a [Button](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.button.aspx) control skin applies to all [Button](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.button.aspx) controls, but not to [LinkButton](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.linkbutton.aspx) controls or to controls that derive from the [Button](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.button.aspx) object.)
* A named skin is a control skin with a [SkinID](http://msdn.microsoft.com/en-us/library/system.web.ui.control.skinid.aspx) property set. Named skins do not automatically apply to controls by type. Instead, you explicitly apply a named skin to a control by setting the control's [SkinID](http://msdn.microsoft.com/en-us/library/system.web.ui.control.skinid.aspx) property. Creating named skins allows you to set different skins for different instances of the same control in an application.

**Cascading Style Sheets**

A theme can also include a cascading style sheet (.css file). When you put a .css file in the theme folder, the style sheet is applied automatically as part of the theme. You define a style sheet using the file name extension .css in the theme folder.

**Theme Graphics and Other Resources**

Themes can also include graphics and other resources, such as script files or sound files. For example, part of your page theme might include a skin for a [TreeView](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.treeview.aspx) control. As part of the theme, you can include the graphics used to represent the expand button and the collapse button.

Typically, the resource files for the theme are in the same folder as the skin files for that theme, but they can be elsewhere in the Web application, in a subfolder of the theme folder for example. To refer to a resource file in a subfolder of the theme folder, use a path like the one shown in this [Image](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.image.aspx) control skin:

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl46_ctl00_ctl16');" \o "Copy Code)

<asp:Image runat="server" ImageUrl="ThemeSubfolder/filename.ext" />

You can also store your resource files outside the theme folder. If you use the tilde (~) syntax to refer to the resource files, the Web application will automatically find the images. For example, if you place the resources for a theme in a subfolder of your application, you can use paths of the form ~/*SubFolder*/*filename.ext* to refer to resource files, as in the following example.

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl46_ctl00_ctl17');" \o "Copy Code)

<asp:Image runat="server" ImageUrl="~/AppSubfolder/filename.ext" />

http://i.msdn.microsoft.com/Global/Images/clear.gif Scoping Themes

You can define themes for a single Web application, or as global themes that can be used by all applications on a Web server. After a theme is defined, it can be placed on individual pages using the **Theme** or **StyleSheetTheme** attribute of the [@ Page](http://msdn.microsoft.com/en-us/library/ydy4x04a.aspx) directive, or it can be applied to all pages in an application by setting the [<pages>](http://msdn.microsoft.com/en-us/library/950xf363.aspx) element in the application configuration file. If the [<pages>](http://msdn.microsoft.com/en-us/library/950xf363.aspx) element is defined in the Machine.config file, the theme will apply to all pages in Web applications on the server.

**Page Themes**

A page theme is a theme folder with control skins, style sheets, graphics files and other resources created as a subfolder of the \App\_Themes folder in your Web site. Each theme is a different subfolder of the \App\_Themes folder. The following example shows a typical page theme, defining two themes named BlueTheme and PinkTheme.

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl47_ctl00_ctl03');" \o "Copy Code)

MyWebSite

App\_Themes

BlueTheme

Controls.skin

BlueTheme.css

PinkTheme

Controls.skin

PinkTheme.css

**Global Themes**

A global theme is a theme that you can apply to all the Web sites on a server. Global themes allow you to define an overall look for your domain when you maintain multiple Web sites on the same server.

Global themes are like page themes in that they include property settings, style sheet settings, and graphics. However, global themes are stored in a folder named Themes that is global to the Web server. Any Web site on the server, and any page in any Web site, can reference a global theme.

http://i.msdn.microsoft.com/Global/Images/clear.gif Theme Settings Precedence

You can specify the precedence that theme settings take over local control settings by specifying how the theme is applied.

If you set a page's [Theme](http://msdn.microsoft.com/en-us/library/system.web.ui.page.theme.aspx) property, control settings in the theme and the page are merged to form the final settings for the control. If a control setting is defined in both the control and the theme, the control settings from the theme override any page settings on the control. This strategy enables the theme to create a consistent look across pages, even if controls on the pages already have individual property settings. For example, it allows you to apply a theme to a page you created in an earlier version of ASP.NET.

Alternatively, you can apply a theme as a style sheet theme by setting the page's [StyleSheetTheme](http://msdn.microsoft.com/en-us/library/system.web.ui.page.stylesheettheme.aspx) property. In this case, local page settings take precedence over those defined in the theme when the setting is defined in both places. This is the model used by cascading style sheets. You might apply a theme as a style sheet theme if you want to be able to set the properties of individual controls on the page while still applying a theme for an overall look.

Global theme elements cannot be partially replaced by elements of application-level themes. If you create an application-level theme with the same name as a global theme, theme elements in the application-level theme will not override the global theme elements.

http://i.msdn.microsoft.com/Global/Images/clear.gif Properties You Can Define Using Themes

As a rule, you can use themes to define properties that concern a page or control's appearance or static content. You can set only those properties that have a [ThemeableAttribute](http://msdn.microsoft.com/en-us/library/system.web.ui.themeableattribute.aspx) attribute set to **true** in the control class.

Properties that explicitly specify control behavior rather than appearance do not accept theme values. For example, you cannot set a [Button](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.button.aspx) control's [CommandName](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.button.commandname.aspx) property by using a theme. Similarly, you cannot use a theme to set a [GridView](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.gridview.aspx) control's [AllowPaging](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.gridview.allowpaging.aspx) property or [DataSource](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.basedatalist.datasource.aspx) property.

Note that you cannot use expression builders, which generate code expressions for assignment in a page at compile time, in themes or skins.

http://i.msdn.microsoft.com/Global/Images/clear.gif Themes vs. Cascading Style Sheets

Themes are similar to cascading style sheets in that both themes and style sheets define a set of common attributes that can be applied to any page. However, themes differ from style sheets in the following ways:

* Themes can define many properties of a control or page, not just style properties. For example, using themes, you can specify the graphics for a [TreeView](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.treeview.aspx) control, the template layout of a [GridView](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.gridview.aspx) control, and so on.
* Themes can include graphics.
* Themes do not cascade the way style sheets do. By default, any property values defined in a theme referenced by a page's [Theme](http://msdn.microsoft.com/en-us/library/system.web.ui.page.theme.aspx) property override the property values declaratively set on a control, unless you explicitly apply the theme using the [StyleSheetTheme](http://msdn.microsoft.com/en-us/library/system.web.ui.page.stylesheettheme.aspx) property. For more information, see the Theme Settings Precedence section above.
* Only one theme can be applied to each page. You cannot apply multiple themes to a page, unlike style sheets where multiple style sheets can be applied.

http://i.msdn.microsoft.com/Global/Images/clear.gif Security Considerations

Themes can cause security issues when they are used on your Web site. Malicious themes can be used to:

* Alter a control's behavior so that it does not behave as expected.
* Inject client-side script, therefore posing a cross-site scripting risk.
* Alter validation.
* Expose sensitive information.
* The mitigations for these common threats are:
* Protect the global and application theme directories with proper access control settings. Only trusted users should be allowed to write files to the theme directories.
* Do not use themes from an untrusted source. Always examine any themes from outside your organization for malicious code before using them on you Web site.
* Do not expose the theme name in query data. Malicious users could use this information to use themes that are unknown to the developer and thereby expose sensitive information.

**How to: Define ASP.NET Page Themes**

This page is specific to

**Microsoft Visual Studio 2008/.NET Framework 3.5**

Other versions are also available for the following:

[Microsoft Visual Studio 2005/.NET Framework 2.0](http://msdn.microsoft.com/en-us/library/ms247256(VS.80).aspx)

[Microsoft Visual Studio 2010/.NET Framework 4.0](http://msdn.microsoft.com/en-us/library/ms247256(VS.100).aspx)

In Visual Web Developer you can define page themes which you can then apply to one or more pages in your application. You can also create themes at the machine level that can be used in multiple applications on the server.

Themes consist of several supporting files, including style sheets for page appearance, control skins to define the appearance of server controls, and any other supporting images or files that make up the theme. The content of a theme is the same whether the theme is defined as a page theme or as a global theme.

Themes can be applied by using either the **Theme** or **StyleSheetTheme** attribute of the [@ Page](http://msdn.microsoft.com/en-us/library/ydy4x04a.aspx) directive, or by setting the [pages Element (ASP.NET Settings Schema)](http://msdn.microsoft.com/en-us/library/950xf363.aspx) element in the application configuration file. Visual Web Developer will only visually represent themes applied by using the **StyleSheetTheme** attribute.

**To create a page theme**

1. In Solution Explorer, right-click the name of the Web site for which you want to create a page theme, and then click **Add ASP.NET Folder**.
2. Click **Theme**.

If the **App\_Themes** folder does not already exist, Visual Web Developer creates it. Visual Web Developer then creates a new folder for the theme as a child folder of the **App\_Themes** folder.

1. Type a name for the new folder.

The name of this folder is also the name of the page theme. For example, if you create a folder named \**App\_Themes**\FirstTheme, the name of your theme is FirstTheme.

1. Add files to your new folder for control skins, style sheets, and images that make up the theme.

**To add a skin file and a skin to a page theme**

1. In Solution Explorer, right-click the name of your theme and then click **Add New Item**.
2. In the **Add New Item** dialog box, click **Skin File.**
3. In the **Name** box, type a name for the .skin file, and then click **Add**.

The typical convention is to create one .skin file per control, such as Button.skin or Calendar.skin. However, you can create as many or as few .skin files as you need.

1. In the .skin file, add a normal control definition by using declarative syntax, but include only the properties that you want to set for the theme. The control definition must include the runat="server" attribute, and it must not include the ID="" attribute.

The following code example shows a default control skin for a [Button](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.button.aspx) control, defining the color and font for all of the **Button** controls in the theme.

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl04');" \o "Copy Code)

<asp:Button runat="server"

BackColor="Red"

ForeColor="White"

Font-Name="Arial"

Font-Size="9px" />

This [Button](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.button.aspx) control skin does not contain a **skinID** attribute. It will be applied to all of the [Button](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.button.aspx) controls in the themed application that do not specify the **skinID** attribute.

|  |
| --- |
| **NoteNote:** |
| An easy way to create a control skin is to add the control to a page and configure it so that it has the look you want. For example, you might add a [Calendar](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.calendar.aspx) control to a page and set its day header, selected date, and other properties. Then, you can copy the control definition from the page to a skin file, but you must remove the **ID** attribute. |

1. Repeat steps 2 and 3 for each control skin file that you want to create.

|  |
| --- |
| **NoteNote:** |
| You can define only one default skin per control. Use the **SkinID** attribute in the skin's control declaration to create named skins for the same type of control. |

**To add a cascading style sheet file to your page theme**

1. In Solution Explorer, right-click the name of your theme and then click **Add New Item**.
2. In the **Add New Item** dialog box, click **Style Sheet**.
3. In the **Name** box, type a name for the .css file, and then click **Add**.

When the theme is applied to a page, ASP.NET adds a reference to the style sheet to the head element of the page. For more information, see [How to: Apply ASP.NET Themes](http://msdn.microsoft.com/en-us/library/0yy5hxdk.aspx)

 Creating Global Themes

A global theme applies to all of the Web sites on a server. The location in which you create a folder for global themes depends on whether you are running your Web site using Internet Information Services (IIS) or testing it using the ASP.NET Development Server.

|  |
| --- |
| **NoteNote:** |
| If you are working with a file system Web site, by default Visual Web Developer runs your Web site for testing by launching the ASP.NET Development Server. For other types of Web sites, Visual Web Developer runs pages in IIS for testing. |

**To create a global theme**

1. Create a **Themes** folder using the following path.

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl27_ctl00_ctl01');" \o "Copy Code)

%windows%\Microsoft.NET\Framework\version\ASP.NETClientFiles\Themes

|  |
| --- |
| **NoteNote:** |
| The folder name for global themes is **Themes**, not **App\_Themes**, as it is for page themes. |

Create a subfolder of the **Themes** folder to hold your global theme files.

The name of the subfolder is the name of the theme. For example, if you create a folder named \**Themes**\FirstTheme, the name of your theme is FirstTheme.

1. Add files to your new folder for control skins, style sheets, and images that make up the global theme.

|  |
| --- |
| **NoteNote:** |
| When you define a global theme, you cannot directly add skin and style sheet files to it using Visual Web Developer. You might find it easier to define and test the theme as a page theme and then copy it to the folder used for global themes on your Web server. |

1. If your Web site is a file-system Web site that you are testing with the ASP.NET Development Server, your theme is ready to test.
2. If you are testing your Web site using a local IIS Web site, open a command window and run aspnet\_regiis -c to install the theme on the server running IIS.
3. If you are testing your theme on a remote Web site or an FTP Web site, you must manually create a Themes folder using the following path.

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl27_ctl00_ctl04');" \o "Copy Code)

IISRootWeb\aspnet\_client\system\_web\version\Themes

**How to: Apply ASP.NET Themes**

This page is specific to

**Microsoft Visual Studio 2008/.NET Framework 3.5**

Other versions are also available for the following:

[Microsoft Visual Studio 2005/.NET Framework 2.0](http://msdn.microsoft.com/en-us/library/0yy5hxdk(VS.80).aspx)

[.NET Framework 3.0](http://msdn.microsoft.com/en-us/library/0yy5hxdk(VS.85).aspx)

[Microsoft Visual Studio 2010/.NET Framework 4.0](http://msdn.microsoft.com/en-us/library/0yy5hxdk(VS.100).aspx)

You can apply themes to a page, a Web site, or globally. Setting a theme at the Web site level applies styles and skins to all the pages and controls in the site unless you override a theme for an individual page. Setting a theme at the page level applies styles and skins to that page and all its controls.

By default, themes override local control settings. Alternatively, you can set a theme as a style sheet theme, so that the theme applies only to control settings that are not explicitly set on the control.

**To apply a theme to a Web site**

1. In the application's Web.config file, set the [<pages>](http://msdn.microsoft.com/en-us/library/950xf363.aspx) element to the name of the theme, either a global theme or a page theme, as shown in the following example:

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl02');" \o "Copy Code)

<configuration>

<system.web>

<pages theme="ThemeName" />

</system.web>

</configuration>

|  |
| --- |
| **NoteNote:** |
| If an application theme has the same name as a global application theme, the page theme takes precedence. |

1. To set a theme as a style sheet theme and be subordinated to local control settings), set the **styleSheetTheme** attribute instead:

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl04');" \o "Copy Code)

<configuration>

<system.web>

<pages styleSheetTheme="Themename" />

</system.web>

</configuration>

A theme setting in the Web.config file applies to all ASP.NET Web pages in that application. Theme settings in the Web.config file follow normal configuration hierarchy conventions. For example, to apply a theme to only a subset of pages, you can put the pages in a folder with their own Web.config file or create a [<location>](http://msdn.microsoft.com/en-us/library/b6x6shw7.aspx) element in the root Web.config file to specify a folder. For details, see [Configuring Specific Files and Subdirectories](http://msdn.microsoft.com/en-us/library/6hbkh9s7.aspx).

**To apply a theme to an individual page**

* Set the **Theme** or **StyleSheetTheme** attribute of the [@ Page](http://msdn.microsoft.com/en-us/library/ydy4x04a.aspx) directive to the name of the theme to use, as shown in the following example:

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl08');" \o "Copy Code)

<%@ Page Theme="ThemeName" %>

<%@ Page StyleSheetTheme="ThemeName" %>

The theme and its corresponding styles and skins now applies only to the page declaring it.

 Applying Skins to Controls

Skins defined in your theme apply to all control instances in the application or pages to which the theme is applied. In some cases, you might want to apply a specific set of properties to an individual control. You can do that by creating a named skin (an entry in a .skin file that has a **SkinID** property set) and then applying it by ID to individual controls.

**To apply a named skin to a control**

* Set the control's [SkinID](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.webcontrol.skinid.aspx) property, as shown in the following example:

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl24_ctl00_ctl01');" \o "Copy Code)

<asp:Calendar runat="server" ID="DatePicker" SkinID="SmallCalendar" />

If the page theme does not include a control skin that matches the **SkinID** property, the control uses the default skin for that control type.