



# Themes

Anand.P.K

# Themes

- Theme enables you to apply a consistent style to the pages in your website.

# Introduction

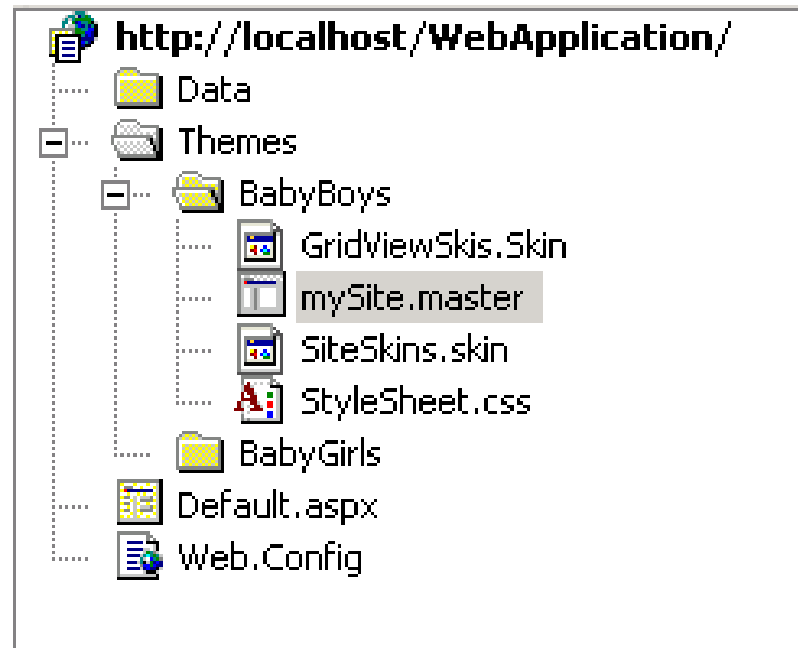
- Themes are similar to CSS in that they enable you to define visual styles for your web pages.
- Themes go further than CSS, however, in that they allow you to apply styles, graphics, and even CSS files themselves to the page of your application.
- You can apply ASP.NET themes at the application, page or server control level.

# Themes

- Application Themes
  - \themes directory of application.
  - Theme name is defined by the directory name.
- Global Themes
  - Install by default
  - Located in the Framework director
    - <WINDIR>\ASPNET\_Client\system\_web\2\_0\_\*\themes
  - Built in themes
    - Basic Blue – simple no CSS
    - Smoke and glass – more stylized with CSS

# How to create a Theme

1. Create a Themes directory
2. Create a Directory in the theme directory
3. Add the skins and other files



# How to create a Skin

1. Create text file with .skin extension
2. Add control-level definitions:
  - Properties (if property exposed as theme-able)
    - Including Databinding code
  - Templates
  - Collections
3. To create an optional skin for a pre-define control
  - Define the SkinId property of the control

# Using Theme and Skin

1. On the page
  - Define a reference to the *theme*.
  - Note: Links are page relative.
2. On a control
  - Define nothing. If a default control skin exists it will be picked up.
  - Define the *Skinid* on the control; will utilize the named skin.
3. Prevent themeing by:
  - *EnableThemeing=false* on page or control.
  - All child controls inherit the parent setting.

## Creating .Skin File

- `<asp:Label runat="server" ForeColor="Aqua" Font-Names="Verdana" Font-Size="X-Small" />`
- `<asp:TextBox runat="server" ForeColor="Aqua" Font-Names="Verdana" Font-Size="X-Small" BorderWidth="1px" BorderColor="Blue" Font-Bold="true" />`
- `<asp:Button runat="server" ForeColor="Aqua" Font-Names="Verdana" Font-Size="X-Small" BorderWidth="1px" BorderColor="Blue" Font-Bold="true" BackColor="AliceBlue"/>`
-



# Applying a theme to a single asp.net page

- `<%@ Page Language="C#" Theme="Stars" %>`
- Applying the theme attribute to the page directive changes the appearance of everything on the page that is defined in "Stars" theme file

## Registering Themes in the Web Configuration File

- web.config file is used for application level theme
- you don't need to defined the theme again in the page directive of your asp.net pages.

```
<configuration>  
  <system.web>  
    <page theme="Stars"/>  
  </system.web>  
</configuration>
```

# Removing theme from server controls

```
<asp:textbox id="TextBox1" RunAt="server"  
BackColor="#000000" EnableTheming="false"/>
```

# Removing Theme from Web Pages

- `<%@ Page Language="c#" EnableTheming="False" %>`
- Even if EnableTheming attribute is set to false at the page level, you enable theming for specific control level.

Eg: `<asp:textbox id="TextBox1" RunAt="server" BackColor="#000000" EnableTheming="true" Theme="stars"/>`

# The stylesheetTheme attribute

- The attributes are overridden by the theme if theme attribute is used.
- Whereas attributes are not overridden by stylesheetTheme attribute

- `<%@ Page Language="c#" StyleSheetTheme="stars" %/>`

**Attribute are  
not overridden**

`<asp:textbox id="TextBox1" RunAt="server" BackColor="#000000"/>`

- `<asp:TextBox runat="server" ForeColor="Aqua" Font-Names="Verdana" Font-Size="X-Small" BorderWidth="1px" BorderColor="Blue" Font-Bold="true" SkinID="sk1" />`
- `<asp:TextBox runat="server" ForeColor="Blue" Font-Names="Arial" Font-Size="X-Large" BorderWidth="1px" BorderColor="Red" Font-Bold="false" SkinID="sk2"/>`

# Creating a skin

- A *skin* is a definition of styles applied to the server controls in your asp.net page.