Themes

Introduction

- One option for specifying the look & feel of our web site is to use CSS files.
- The ASP.NET includes a complementary technology for specifying the look & feel known as themes.
- The theme is applied on the server side and it can be applied either programmatically or declaratively.

The Syntax

- The syntax we use when creating a theme is the same syntax we use when creating ____`.aspx` files.

The `App_Themes` Folder

- We place each one of the themes in a separated sub directory within App_Themes, a directory within our web application.

The `*.skin` Files

- Each theme can include multiple files and each theme must include (at the minimum) a `*.skin` file.
- Files of the *.skip type define the look and feel for web controls they refer to. The syntax is the same syntax we use when declaring the controls we use.
- The Visual Studio doesn't provide a designer tool for creating *.skin files.

The `*.skin` Files

- The simplest way to create a `*.skin` file would be to copy paste the code from a simple `*.aspx` file. Taking out the `ID` will apply the setting for all controls (instead for the very specific one with the ID we took out).

Sample

```
<asp:Button runat="server" Text="Button" BackColor="Yellow"
            BorderColor="#006600" />

<asp:TextBox runat="server" BackColor="#CCCCCC"></asp:TextBox>
```

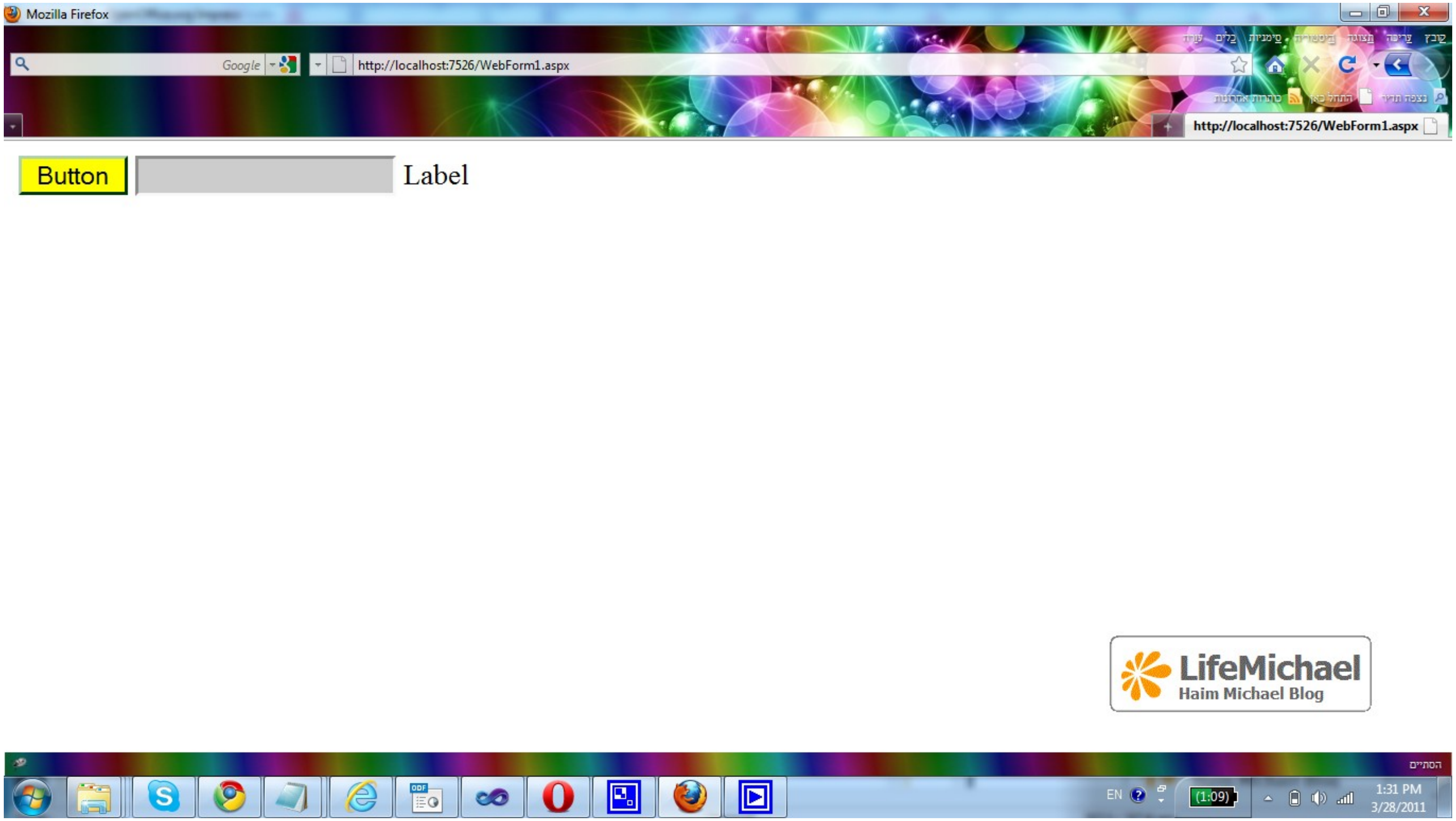
simplesetting.skin

Sample

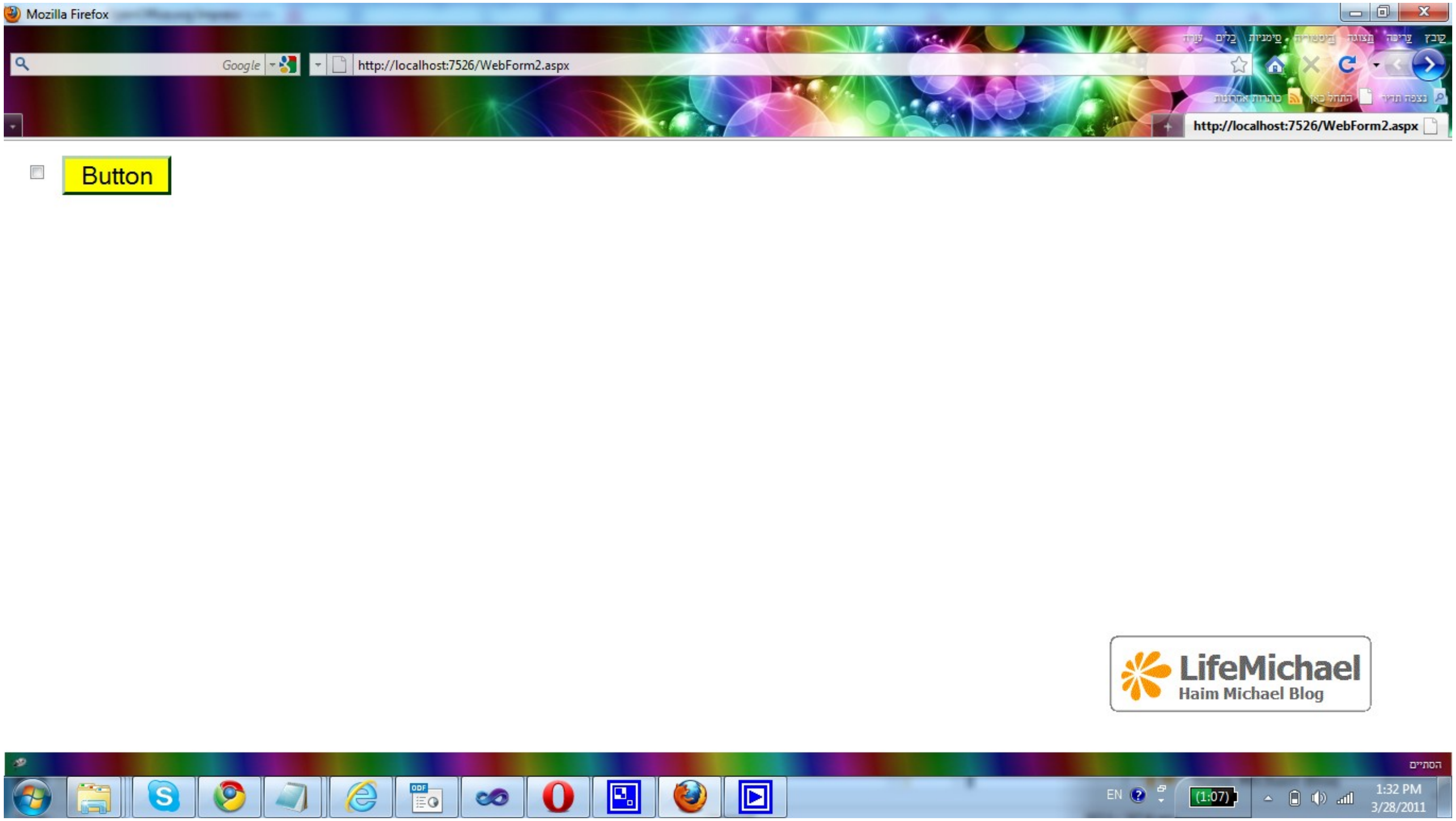
```
<?xml version="1.0"?>

<!--
  For more information on how to configure your ASP.NET application,
please visit
  http://go.microsoft.com/fwlink/?LinkId=169433
  -->

<configuration>
    <system.web>
        <compilation debug="true" targetFramework="4.0" />
        <pages theme="simpletheme" />
    </system.web>
</configuration>
```

Web.Config

Sample


Mozilla Firefox
http://localhost:7526/WebForm1.aspx
Button
Label
LifeMichael
Haim Michael Blog
1:31 PM
3/28/2011

Sample



Mozilla Firefox
http://localhost:7526/WebForm2.aspx
□ Button
LifeMichael
Haim Michael Blog
1:32 PM
3/28/2011

Applying Themes at Page Level

- We can assign a theme to specific page. We can actually assign different themes to different pages.
- When applying a specific theme for a specific page the new theme setting will be instead of any other theme.

```
<%@ Page Language="C#" Theme="SimpleTheme".. %>
```

The `SkinID` Property

- We can include in our `*.skin` file the same control more than once, and assign each one of its occurrences with a different `SkinID` value.

```
<asp:Button runat="server" SkinID="sadbutton" BackColor="332C2C">
<asp:Button runat="server" SkinID="happybutton" BackColor="F62C2C">
```

- When coding our `*.aspx` file we can specify in each one of our control occurrences the very specific `SkinID` we want to apply.

```
<asp:Button runat="server" SkinID="happybutton" ... />
<asp:Button runat="server" SkinID="sadbutton" ... />
<asp:Button runat="server" SkinID="sadbutton" ... />
```

Assigning Themes Programmatically

- One of the properties inherited from `Page` is `Theme`. We can programmatically assign it with the name of a theme.

```
public virtual string Theme { get; set; }
```

The `StyleSheetTheme` Attirbute

- When having a conflict between the control specific setting and the theme, the theme wins.
- We can change that behavior when assigning the theme name to the `StyleSheetTheme` property instead of `Theme`.

```
<%@ Page Language="C#" StyleSheetTheme="SomeTheme"... %>
```

YouTube

The `EnableTheming` Attirbute

- We can add the `EnableTheming` attribute within any control a theme might apply. Assigning `EnableTheming` with false will opt out the control.

```
<asp:Button ID=”cancelButton” runat=”server”... EnableTheming=”false” />
```

Images

- We can easily include images as part of our theme. We will usually place them within a sub directory within our theme directory.

CSS

- We can place CSS files as part of the theme. We just need to ensure the HEAD html element in our `*.aspx` file includes the `runat="server"` attribute.

```
<head runat="server">
    <title>blabla</title>
</head>
```