**Accessing the controls and methods of a MasterPage**

<%@ Master Language="C#" AutoEventWireup="true" CodeFile="DefaultLayout.master.cs" Inherits="DefaultLayout" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >

<head runat="[server](http://www.geekpedia.com/tutorial184_Accessing-the-controls-and-methods-of-a-MasterPage.html)">

    <title>Sample Page</title>

</head>

<body>

    <form id="form1" runat="server">

    <div>

        <h1><asp:Label runat="server" ID="lblHeading"></asp:Label></h1>

        <asp:contentplaceholder id="MainPageContent" runat="server">

        </asp:contentplaceholder>

    </div>

    </form>

</body>

</html>

As you can see, we placed a label *lblHeading* and a placeholder entitled *MainPageContent*. The label will be the one that will be assigned a value to from inside the Default.aspx, through a method inside the MasterPage. The placeholder is there just to give a reason for the MasterPage; we will set it with a some content inside Default.aspx.  
  
Now let us create a (very simple) method inside DefaultLayout.master:

public void SetPageHeading(string PageHeading)

{

   lblHeading.Text = PageHeading;

}

This is the method we're going to access from inside Default.aspx's code-behind file. The parameter passed (*PageHeading*) is the one to which the label gets assigned.  
  
Now switch to the Default.aspx page and inside the code view replace all its current content with the following:

<%@ Page Language="C#" AutoEventWireup="true" MasterPageFile="DefaultLayout.master" CodeFile="Default.aspx.cs" Inherits="\_Default" %>

<asp:content id="HomePageContent" contentplaceholderid="MainPageContent" runat="server">

This phrase is placed in a content placeholder inside the Default.aspx page which uses the DefaultLayout.master MasterPage.

</asp:content>

With this markup we are setting the MasterPage to be used with this Web Form, and the content of the placeholder.

**Accessing the methods of a MasterPage from inside the Web Form**

What's left is setting the label of the MasterPage through code, thus switch to code view and inside the *Page\_Load* event use the following piece of code:

((DefaultLayout)this.Master).SetPageHeading("This heading is set from inside Default.aspx");

This is the most important line of the whole tutorial, it casts the *this.Master* object to *DefaultLayout* which is the type of the MasterPage (remember we named the MasterPage *DefaultLayout*). After we do the casting, the method is accessed normally, as any other method, with the parameters passed inside parenthesis.  
  
We used the label changing example method just to prove how you can access a MasterPage's method from inside a web form, however if the only thing you are looking to do is to change the value of a label or access the properties of a similar control, you should have a look at the following:

**Accessing the controls of a MasterPage from inside a Web Form**

If you wish to access a control that's inside the MasterPage, you need to use the **FindControl()** method as shown below:

Label lblHeading;

lblHeading = (Label)Master.FindControl("lblHeading");

lblHeading.Text = "This heading is set from inside Default.aspx";

And here is the same version of the code, this time in VB.NET:

Dim lblHeading As Label

lblHeading = CType(Master.FindControl("lblHeading"), Label)

lblHeading.Text = "This heading is set from inside Default.aspx"