275 Grove Street, Suite 2-400
Newton, MA 02466, USA
P: 888.365.2779 F: 617.249.2116
E: Info@Telerik.com

Adding the Telerik ASP.NET 2.0 RadMenu Control to MOSS 2007 Publishing Sites

Forward

With the release of Windows SharePoint Services (WSS) v3 and Microsoft Office SharePoint Server (MOSS) 2007, Microsoft has implemented significant architectural changes to the SharePoint platform which simplifies the task of customizing the user interface. Most of these improvements are because WSS v3, unlike its predecessor, is built on top of the .NET 3.0 Framework bringing the native capabilities of ASP.NET 2.0 directly to the SharePoint platform. One such ASP.NET 2.0 concept that is leveraged by WSS v3 is the navigation provider model. The navigation provider model separates the navigation hierarchy (the data portion, otherwise known as site-map data sources) from the rendering (the presentation portion, otherwise known as navigation controls). The role of the site-map data source is to abstract the navigation hierarchy from the underlying system (such as SharePoint) to make it transparent to the navigation control. This allows developers to easily snap ASP.NET 2.0 navigation provider model compatible controls into a SharePoint site by simply configuring them to receive the navigation hierarchy from the provided SharePoint site-map data sources.

This white paper demonstrates how to incorporate the Telerik RadMenu navigation control in a MOSS 2007 Publishing site. In order to leverage the Telerik RadMenu control, some required files must be deployed to each SharePoint Web Front End (WFE) server. The deployment of the files is conducted in one of two ways: (1) manual deployment of all necessary files using simple file-copy operations or (2) leveraging the WSS solution framework for deployment. Once the required files are deployed, the Telerik RadMenu is then added to the Publishing site by editing the site’s master page(s).

While only the Telerik RadMenu navigation control is documented in this white paper, the same steps can be used to incorporate other RadControls suite such as the RadTreeView and RadTabStrip into WSS v2 or MOSS 2007 sites since they have existing default counterparts. Other components in the RadControls suite such as RadRotator, RadSplitter and RadComboBox do not have a direct comparable control in WSS v3 or MOSS 2007 sites can be used and deployed using the steps outlined in this white paper, but additional custom code may be required. In addition, the fully functional WYSIWYG RadEditor will be available within Publishing sites via a Telerik supplied custom wrapper.

In addition, this white paper is written to work with the 2007 Q1 release of RadMenu, version 4.3.0.0. These same steps apply to prior and future versions of RadMenu, just make sure the correct version is specified where appropriate (<SafeControl> entries, <%@Register %> directives, etc.).

Deploying Telerik's RadMenu Control Overview

The Telerik RadMenu navigation control, just like all the controls in the Telerik RadControls suite, requires a few files to operate correctly:

- **Assembly**: All logic associated with the RadMenu is compiled into a single assembly: **RadMenu.Net2.dll**. This assembly must be copied to the desired SharePoint site's **bin** directory located in the webroot of the SharePoint site. *The assembly could also be deployed to the GAC if the RadMenu is to be used across multiple SharePoint sites, however in this white paper it's assumed that it is to be used in a single SharePoint site.*
- **Skin files:** The visual appearance of RadMenu is controlled using skins. Skins include images and a single stylesheet (styles.css). All files included in the skin are contained in a single folder. Just as other controls in the RadControls suite, multiple skins are supported by the RadMenu. All skins are located in the **skins** directory. This directory must be copied to the desired SharePoint site's **wpresources** directory located in the webroot of the SharePoint site. *If the assembly is deployed to the GAC, the skin files can be deployed to the C:\Program Files\Common Files\Microsoft Shared\web server extensions\wpresources\ directory to be available to all Web Applications on the server.*

Finally, the last step is to tell SharePoint that the RadMenu assembly is safe to invoke. SharePoint's collaborative nature allows users to create and upload ASPX pages into SharePoint sites. However, in order to keep these untrusted users from adding malicious code in the ASPX pages, Microsoft added a safe mode parser to SharePoint that requires code to be explicitly designated as safe to invoke. The RadMenu assembly needs to be registered as a safe assembly, or control, so it can execute correctly within SharePoint. This is done by registering the RadMenu assembly as a safe control within the SharePoint site's web.config file.

All these steps, from copying the files to the appropriate locations to registering the RadMenu assembly as a safe control, can all be carried out in one of two ways. The first option is to manually perform all the steps for each SharePoint site that will utilize the RadMenu. A section option involves automating the deployment process by packaging all the RadMenu files up into a WSS solution package and utilize the WSS solution infrastructure to deploy all the necessary changes automatically.

The following two sections outline each deployment process (manual vs. automated). It is recommended to review both processes, as even if the automated deployment process is favored, developers should be very familiar with the manual process in the chance issues arise down the road that requiring troubleshooting.

Both sections assume that a MOSS 2007 Publishing site has already been created in a new web application within Internet Information Server (IIS) with the default settings, except for the following exceptions:

- *Site name of **Telerik***
- *Port **80***
- *Host header specified as **telerik***

- *Site template: **Publishing Portal***

*Provided the default settings were used when the Publishing site was created, except where otherwise noted above, SharePoint would have created the site with a webroot on the file system in the following location: **c:\inetpub\wwwroot\wss\VirtualDirectories\telerik80**. It is not important if different options were used… the instructions white paper simply assume the above options as a point of reference.*

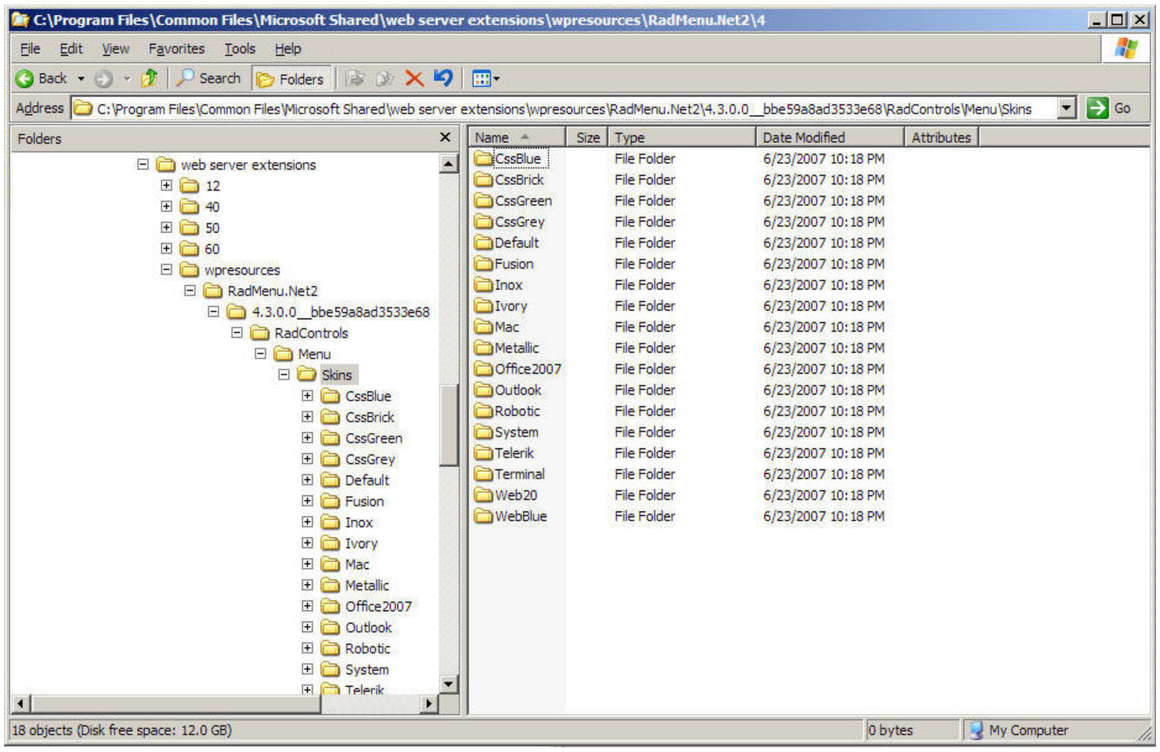
Manual Deployment Process

The manual process of deploying the required files and making the necessary changes to existing files is just that: a process that involves many steps that are performed one after another by a developer or server administrator. This section outlines the required steps that need to be taken in order to deploy the Telerik RadMenu prior to implementing it within an existing MOSS 2007 Publishing site.

1. Install the RadMenu assembly that contains all the logic for the control, **RadMenu.Net2.dll** to the Global Assembly Cache (GAC).

2. Copy the RadMenu skins directory that contains the presentation templates, **RadControls\Menu\skins**, to the following location: **c:\Program Files\Common Files\Microsoft Shared\web server extensions\wpresources\RadMenu.Net2\4.3.0.0__bbe59a8ad3533e68\RadControls\Menu\Skins**.

   Refer to Figure 1 for an example of the deployed files within the Publishing site's web root.



C:\Program Files\Common Files\Microsoft Shared\web server extensions\wpresources\RadMenu.Net2\4.3.0.0__bbe59a8ad3533e68\RadControls\Menu\Skins

Name	Size	Type	Date Modified	Attributes
CssBlue		File Folder	6/23/2007 10:18 PM	
CssBrick		File Folder	6/23/2007 10:18 PM	
CssGreen		File Folder	6/23/2007 10:18 PM	
CssGrey		File Folder	6/23/2007 10:18 PM	
Default		File Folder	6/23/2007 10:18 PM	
Fusion		File Folder	6/23/2007 10:18 PM	
Inox		File Folder	6/23/2007 10:18 PM	
Ivory		File Folder	6/23/2007 10:18 PM	
Mac		File Folder	6/23/2007 10:18 PM	
Metallic		File Folder	6/23/2007 10:18 PM	
Office2007		File Folder	6/23/2007 10:18 PM	
Outlook		File Folder	6/23/2007 10:18 PM	
Robotic		File Folder	6/23/2007 10:18 PM	
System		File Folder	6/23/2007 10:18 PM	
Telerik		File Folder	6/23/2007 10:18 PM	
Terminal		File Folder	6/23/2007 10:18 PM	
Web20		File Folder	6/23/2007 10:18 PM	
WebBlue		File Folder	6/23/2007 10:18 PM	

18 objects (Disk free space: 12.0 GB) 0 bytes My Computer

Figure 1 – Telerik skin files after copying the RadMenu skin files into the directory structure

3. Next, tell SharePoint that the RadMenu is safe to execute. Register two namespaces and all the classes within them as safe controls with the SharePoint Publishing site by opening the **web.config** within the SharePoint Publishing site's web root and add the following two entries at the end of the **<SafeControls>** section, just before the closing </SafeControls> tag:

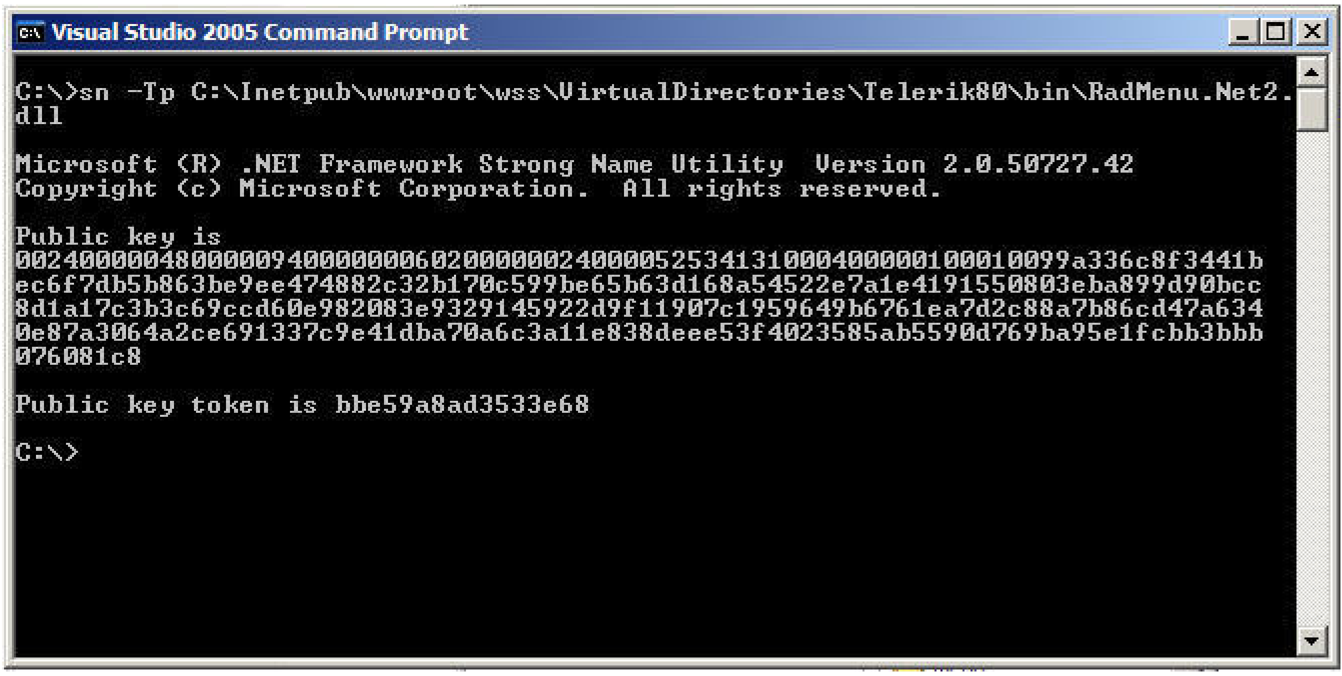
```
<SafeControl Assembly="RadMenu.Net2, Version=4.3.0.0, Culture=neutral,
PublicKeyToken=bbe59a8ad3533e68" Namespace="Telerik.RadMenuUtils"
TypeName="*" Safe="True" />

<SafeControl Assembly="RadMenu.Net2, Version=4.3.0.0, Culture=neutral,
PublicKeyToken=bbe59a8ad3533e68" Namespace="Telerik.WebControls"
TypeName="*" Safe="True" />
```

Note that the version of the assembly is specified in each <SafeControl> tag. If deploying a different version of the RadMenu, ensure that the version number is correct here.

To obtain the public key token for the RadMenu.Net2.dll, open a Visual Studio 2005 Command Prompt and enter the following (FYI: the sn.exe utility is case sensitive) command which will return the public key blob and token, as shown in Figure 2:

```
sn.exe –Tp [full path to RadMenu.Net2.dll]\RadMenu.Net2.dll
```



```
C:\>sn -Tp C:\Inetpub\wwwroot\wss\VirtualDirectories\Telerik80\bin\RadMenu.Net2.
dll

Microsoft (R) .NET Framework Strong Name Utility  Version 2.0.50727.42
Copyright (c) Microsoft Corporation.  All rights reserved.

Public key is
002400000480000094000000060200000024000052534131000400000100010099a336c8f3441b
ec6f7db5b863be9ee474882c32b170c599be65b63d168a54522e7a1e4191550803eba899d90bcc
8d1a17c3b3c69ccd60e982083e9329145922d9f11907c1959649b6761ea7d2c88a7b86cd47a634
0e87a3064a2ce691337c9e41dba70a6c3a11e838deee53f4023585ab5590d769ba95e1fcbb3bbb
076081c8

Public key token is bbe59a8ad3533e68

C:\>
```

Figure 2 – Using sn.exe to obtain the public key blob and public key token from the RadMenu.Net2.dll assembly.

Save and close the file **web.config**.

If the control is going to be used within a SharePoint Publishing site that is duplicated on multiple SharePoint WFE's to implement load balancing then the manual deployment process outlined above must happen on each WFE. *This is just one of the many advantages that the automated deployment process has over the manual process.*

Automated Deployment Process

The automated process of deploying the required files and making the necessary changes to existing files is intended to greatly reduce the number of steps when developers want to leverage the RadMenu in a new or existing MOSS 2007 Publishing site. All the steps previously outlined in the manual process can be automated using WSS solution packages, a new concept and capability in the latest release of SharePoint, WSS v3.

WSS solution packages, or SharePoint packages for short, are Microsoft cabinet files with a *.WSP filename extension that are used to deploy custom code, files, and configuration file changes to a SharePoint environment. These SharePoint packages can deploy assemblies to sites or the GAC, deploy files anywhere in the SharePoint [..]\12* directory structure and edit web.config files among other things. Once a SharePoint package has been created, a farm administrator adds it to the SharePoint farm's solution store using STSADM.EXE where it can then be deployed either using STSADM.EXE or through the SharePoint farm's Central Administration site. Conversely, the solution can also be retracted which will undo all the changes that were implemented when it was deployed (such as remove safe control entries from the web.config, remove the assembly from the bin (or GAC), and remove any resource files). *For more information on WSS solution packages, refer to the WSS v3 SDK: http://msdn2.microsoft.com/en-us/library/aa543214.aspx.*

Using a SharePoint package, the RadMenu files and necessary changes can be packaged into a single *.WSP file that can be used to automate all necessary deployment steps of the control. But first, developers need to create the SharePoint package that will be used for deployment. This section outlines the steps necessary to create a WSS SharePoint package WSP file that can be used to deploy the RadMenu control quickly to any MOSS 2007 Publishing site.

The following steps outline how to create a WSS solution package using Visual Studio 2005 and MSBuild in an automated and easily repeatable process. Visual Studio 2005 and MSBuild are not required to create WSS solution packages. For instructions on creating a WSS solution package, refer to the WSS v3 SDK: http://msdn2.microsoft.com/en-us/library/aa543741.aspx.

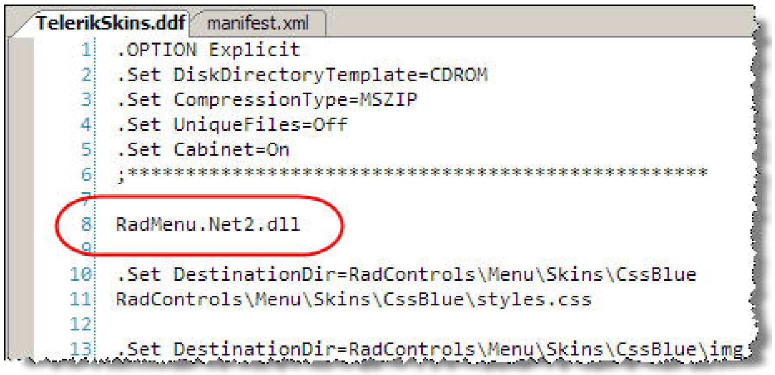
1. Open Visual Studio 2005 and create a new Visual C# or Visual Basic project using the **Empty Project** template and give it the name of **Telerik.RadMenu**. This will be used as a container for the WSS solution package source files.

2. Copy the RadMenu assembly, **RadMenu.Net2.dll**, to the root of the project. *Refer to Figure 4 for an example of the project.*

3. Copy the RadMenu provided skin files to the root of the project under the directory structure **[root]\RadControls\Menu\Skins**. *Refer to Figure 4 for an example of the project.*

4. Create a diamond directive file (*.ddf) that contains a manifest of all the files into include in the WSS solution package. Due to the large number of files that make up the default RadMenu skins (200+ files) this can be a tedious and significant task. To simplify this task, copy the utility **TelerikSkinDDFCreator.exe** (available via download at the end of this white paper) into the **RadControls** directory added to the Visual Studio project in the previous step and execute it from Windows Explorer or the command line. It will generate the file **TelerikSkins.ddf** that lists all files, preserving the folder structure, within the **RadControls** directory. Copy this generated file to the root of the Visual Studio 2005 project.

5. Once the TelerikSkins.ddf file has been created using the provided utility and added to the project, the RadMenu assembly needs to be added to the diamond directive file. Just after the comment in the **TelerikSkins.ddf** file, add an entry for the **RadMenu.Net2.dll**, as shown in Figure 3:



TelerikSkins.ddf | manifest.xml

```
.OPTION Explicit
.Set DiskDirectoryTemplate=CDROM
.Set CompressionType=MSZIP
.Set UniqueFiles=Off
.Set Cabinet=On
;**************************************************

RadMenu.Net2.dll

.Set DestinationDir=RadControls\Menu\Skins\CssBlue
RadControls\Menu\Skins\CssBlue\styles.css

.Set DestinationDir=RadControls\Menu\Skins\CssBlue\img
```

Figure 3 – TelerikSkins.ddf

At this point all necessary deployment files are in the Visual Studio project. Now the WSS solution package manifest needs to be created and added to the project. This file, manifest.xml, is used by SharePoint when adding the WSS solution package to the SharePoint farm's solution store as well as when the solution is deployed. The following detail the creation of manifest.xml:

6. Create a new XML file in the root of the Visual Studio project with the name **manifest.xml**. This file is used by SharePoint when the WSS solution package is added to the SharePoint farm's solution store and deployed; think of it as the instructions telling SharePoint what to do with the WSS solution package and all files within the package.

   Copy the code in Listing 3 into the manifest.xml file.

Listing 3

```
<?xml version="1.0" encoding="utf-8" ?>
<Solution xmlns="http://schemas.microsoft.com/sharepoint/"
          SolutionId="DC821995-2B38-4c5c-813D-6FB092C5731F"
          DeploymentServerType="WebFrontEnd"
          ResetWebServer="TRUE">

  <Assemblies>
    <!-- deploy the assembly to the server’s bin directory -->
    <Assembly DeploymentTarget="GlobalAssemblyCache"
              Location="RadMenu.Net2.dll">

      <!-- add two safe control entries to the web.config -->
      <SafeControls>
        <SafeControl Namespace="Telerik.RadMenuUtils"
                     TypeName="*" Safe="True" />
        <SafeControl Namespace="Telerik.WebControls"
                     TypeName="*" Safe="True" />
      </SafeControls>

      <!-- deploy resources used by RadMenu to wpresources -->
      <ClassResources>
        <!-- INSERT skin files -->
      </ClassResources>
    </Assembly>
  </Assemblies>
</Solution>
```

Note: To obtain the public key for the RadMenu.Net2.dll assembly, refer to the end of step 3 in the Manual Deployment Process section of this white paper.

7. Next, all the files making up the RadMenu skins need to be added to the manifest. Just like in the case of the diamond directive file, this is daunting task with 200+ files to add to add. Again, to simplify this task, copy the utility **TelerikSkinManifextCreator.exe** (available via download at the end of this white paper) into the **RadControls** directory added to the Visual Studio project in the previous step and execute it from Windows Explorer or the command line. It will generate the file **TelerikSkins.xml** that contains a long list of <ClassResource> nodes, each containing a reference to one of the files in the skins directory. Copy all the **<ClassResource>** nodes and insert them in the **<ClassResources>** node (replacing the comment) in the **manifest.xml** file created in the last step.

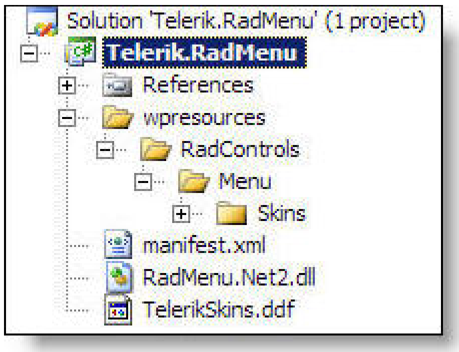
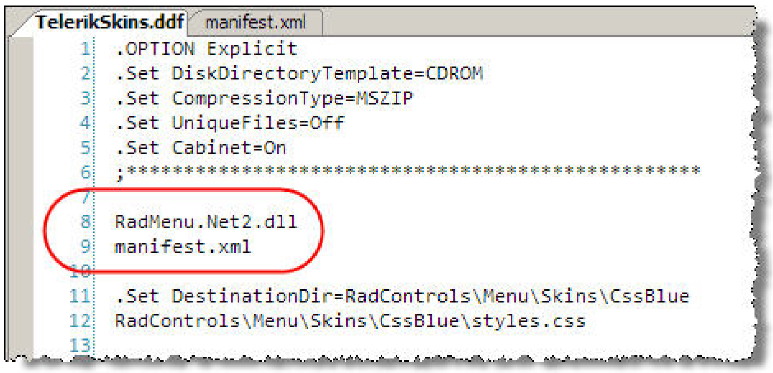


Figure 4 – Visual Studio 2005 Project Created in the Automated Deployment Process

8. Once the manifest.xml file has been created using the provided utility and added to the project, the manifest.xml needs to be added to the diamond directive file. Just after the assembly entry previously added in the **TelerikSkins.ddf** file, add an entry for the **manifest.xml**, as shown in Figure 5:



```
.OPTION Explicit
.Set DiskDirectoryTemplate=CDROM
.Set CompressionType=MSZIP
.Set UniqueFiles=Off
.Set Cabinet=On
;***************************************************

RadMenu.Net2.dll
manifest.xml

.Set DestinationDir=RadControls\Menu\Skins\CssBlue
RadControls\Menu\Skins\CssBlue\styles.css
```

Figure 5 – TelerikSkins.ddf

9. At this point our WSS solution package is prepared, all necessary files have been created, and it is ready to be packaged up into a *.WSP. The files will be packaged up using makecab.exe, a utility included in the Microsoft Cabinet SDK (*available from the Microsoft Knowledge Base: http://support.microsoft.com/default.aspx/kb/310618*). Download the Microsoft Cabinet SDK and extract the contents. *The recommended location to extract the MSFT Cabinet SDK is c:\Program Files\Microsoft Cabinet SDK.*

10. Now the Visual Studio 2005 project file needs to be configured so that when the project is built, MSBuild will not compile anything, but rather call makecab.exe to create the *.WSP file. One option is to edit the project file directly using Notepad or similar text editor. The other option is to edit it directly in Visual Studio. To do this, right-click the project name in the **Solution Explorer** tool window and select **Unload Project**. Next, right-click the project name again in **Solution Explorer** and select **Edit [project name].csproj**. Scroll down to the line where the C# MSBuild targets file is imported:

    ```
    <Import Project="$(MSBuildBinPath)\Microsoft.CSharp.targets" />
    ```

    Delete that line down through the commented lines (including `<Target Name=”AfterBuild”></Target>`). Add the following code to define a variable **MakeCabPath** that references the full path to makecab.exe:

    ```
    <PropertyGroup>
      <MakeCabPath>"C:\Program Files\Microsoft Cabinet 
    SDK\BIN\MAKECAB.EXE"</MakeCabPath>
    </PropertyGroup>
    ```

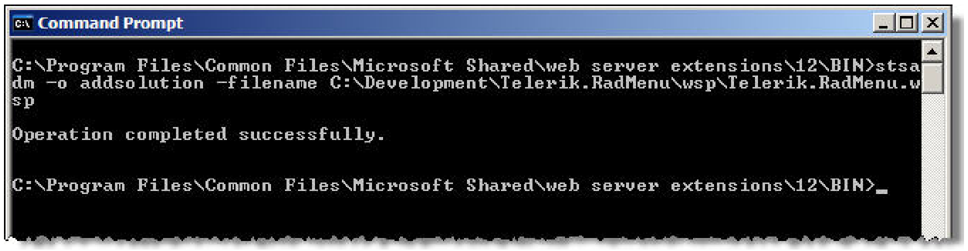
    Next, replace the **Build** target that was removed when the C# MSBuild targets file was deleted. This MSBuild target executes makecab.exe from the command line passing in the DDF file and creates a *.WSP with the Visual Studio 2005 project name as the filename. This file is created in the **wsp** subdirectory within the project (parallel to where **bin** and **obj** are typically found):

    ```
    <Target Name="Build">
      <Exec Command="$(MakeCabPath) /F TelerikSkins.ddf /D 
    CabinetNameTemplate=$(MSBuildProjectName).wsp /D DiskDirectory1=wsp" />
    </Target>
    ```

    Save the changes and right-click the project name in the **Solution Explorer** tool window and select **Reload Project** (if a warning pops up, simply **OK** though it… that is Visual Studio warning that the project file does not look like one of the stock project files).

11. That’s it! Now build the project. Assuming a successful build, it is now time to add the WSS solution package to the SharePoint farm’s solution store. Open a command project and enter the following command (replacing file paths where necessary):

```
c:\>[path to stsadm.exe]\stsadm.exe -o addsolution -filename [path to
WSP]\telerik.radmenu.wsp
```



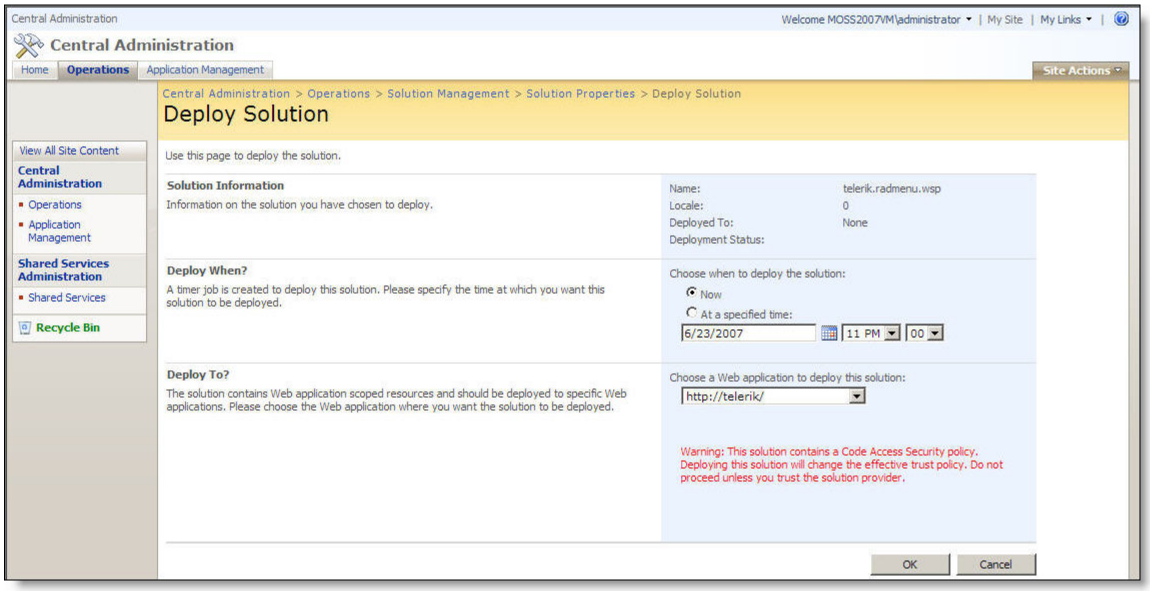
```
C:\Program Files\Common Files\Microsoft Shared\web server extensions\12\BIN>stsa
dm -o addsolution -filename C:\Development\Telerik.RadMenu\wsp\Telerik.RadMenu.w
sp

Operation completed successfully.

C:\Program Files\Common Files\Microsoft Shared\web server extensions\12\BIN>_
```

Figure 6 – Adding the telerik.radmenu.wsp WSS solution package to the SharePoint farm's solution store

12. With the solution added to the SharePoint farm's solution store, it can now be deployed. Before deploying, notice how the assembly is not in the target site's **bin** directory, the two safe control entries are not present in the web.config, and the skin files are not in **wpresources**. Open Internet Explorer and navigate to the farm's Central Administration site. Select the **Operations** tab and then **Solution Management** under the **Global Configuration** section. Select **telerik.radmenu.wsp** and click **Deploy Solution**. When prompted where to deploy the solution, select the desired SharePoint Publishing site.



Deploy Solution

Use this page to deploy the solution.

Solution Information
Information on the solution you have chosen to deploy.

Name: telerik.radmenu.wsp
Locale: 0
Deployed To: None
Deployment Status:

Deploy When?
A timer job is created to deploy this solution. Please specify the time at which you want this solution to be deployed.

Choose when to deploy the solution:
[✓ Now]
At a specified time:
6/23/2007 11 PM 00

Deploy To?
The solution contains Web application scoped resources and should be deployed to specific Web applications. Please choose the Web application where you want the solution to be deployed.

Choose a Web application to deploy this solution:
http://telerik/

Warning: This solution contains a Code Access Security policy. Deploying this solution will change the effective trust policy. Do not proceed unless you trust the solution provider.

Figure 7 – Deploying the telerik.radmenu.wsp WSS solution package to the target SharePoint site

That's it! The best part is the **telerik.radmenu.wsp** file created by following these steps can be used on other SharePoint servers and farms to deploy the Telerik RadMenu control! The only reason that the package would need to be updated is if either the RadMenu assembly was updated to a new version or a skin was added/updated.

In addition, if a SharePoint farm has multiple WFE's load balanced and thus, running multiple copies of the same site across multiple servers, the SharePoint will automatically deploy the WSS solution package to each site on each server. If the manual deployment process was elected, this would either have to be scripted, or each server would need to be manually updated one-by-one.

Incorporating the RadMenu Control in Office SharePoint Server 2007 Publishing Sites

Regardless of the deployment process selected (manual or automated), all necessary files are now in place and all required changes have been implemented. Now it the control can be added to Publishing site master pages (and page layouts if necessary). This is a very easy and quick two-step process. The following steps outline how to swap out the default SharePoint ASP.NET Menu navigation control used within the default master page for a Publishing Portal site: BlueBand.master.

1. Launch **Office SharePoint Designer 2007** and open the desired Publishing site: **http://telerik**.

2. Within the **Folder List** tool window, expand the tree to show the site's **Master Page Gallery** in the following location: **http://telerik/_catalogs/masterpage**.

3. Open **BlueBand.master** within the **masterpage** list. If prompted to check out the file, select **Yes**.

4. Add the following register directive just before the <HTML> tag to register the RadMenu assembly on the page:

   ```
   <%@ Register TagPrefix="telerik" Namespace="Telerik.WebControls"
   Assembly="RadMenu.Net2, Version=4.3.0.0, Culture=neutral,
   PublicKeyToken=bbe59a8ad3533e68" %>
   ```

   Note that the version of the assembly is specified in the <%@Register %> directive. If deploying a different version of the RadMenu, ensure that the version number is correct here.

5. Scroll down to around line #91, where the <**SharePoint:AspMenu ID="GlobalNav" …**> node is. Delete the entire **<SharePoint:AspMenu>** node and its contents (including the closing node).

6. Add the following code just before the opening **<PublishingNavigation:PortalSiteMapDataSource>** node:

   ```
   <!-- fix for RadMenu CSS conflict when used in SharePoint -->
   <style type="text/css">.radmenu ul li {margin:0px;}</style>
   <telerik:RadMenu id="radMenu1" runat="server" Skin="Default"
   ```

```
         DataSourceId="siteMapDataSource1" CollapseDelay="0"
RadControlsDir="/_wpresources/RadMenu.Net2/4.3.0.0__bbe59a8ad3533e68/Ra
dControls">
  <ExpandAnimation Type="None" Duration="0"></ExpandAnimation>
  <CollapseAnimation Type="None" Duration="1000"></CollapseAnimation>
</telerik:RadMenu>
```

7. Verify the **DataSourceId** attribute in the **<telerik.RadMenu>** node contains the same value specified in the **ID** attribute within the **<PublishingNavigation:PortalSiteMapDataSource>** node immediately following the control.

8. Save all changes and check in the file, publishing a new major version, followed by approving the updated master page.

With the RadMenu deployed and incorporated into a MOSS Publishing site, open Internet Explorer and browse to the Publishing site's homepage. The top, horizontal navigation control just below the site logo and title should display the rendered menu. However, instead of using the default menu control, it is using the Telerik RadMenu as shown in Figure 8:

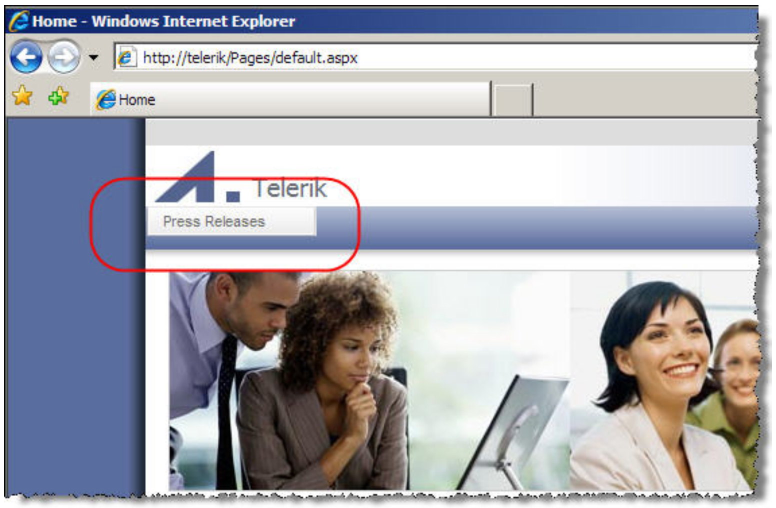

Home - Windows Internet Explorer
http://telerik/Pages/default.aspx
Home
Telerik
Press Releases

Figure 8 – RadMenu used to render the navigation in the default SharePoint Publishing Portal site template

Conclusion

This white paper outlined the two options available for deploying the files required by the Telerik RadMenu as well as making the appropriate changes to the necessary files. While the manual deployment process may feel more familiar to traditional ASP.NET 2.0 developers, the automated deployment process is much preferred as it utilizes the embedded installation framework provided by SharePoint (specifically WSS v3): WSS solution packages.

Finally, two points made at the beginning of this white paper need to be reiterated:

- Other controls in the Telerik RadControls suite follow the same deployment model and requirements. This white paper simply addressed one of the controls (RadMenu) within the RadControls suite.

- While the demonstration site used in this white paper is a MOSS 2007 Publishing site, these same steps apply to all WSS v3 based sites, including sites created using the **Team Site** or **Blank Site** site template.

Downloads

- **Microsoft Cabinet SDK**: Microsoft Knowledge Base #Q310618
- **DDF Builder Utility**: TelerikSkinDDFcreator.exe
- **Manifest ClassResource Utility**: TelerikSkinManifextCreator.exe

Credits

Created by **Andrew Connell**, MVP Microsoft Office SharePoint Server (http://www.andrewconnell.com) on behalf of **Telerik** (http://www.telerik.com).