WebDesktop.NET 2.0 White paper

This white paper discusses changes, improvements and new features introduced in WebDesktop.NET 2.0.

Table of Contents

I. Breaking Changes 3

II. Upgrading from previous version 3

Overview 3

Assembly References 4

III. What's New in 2.0 4

WebMenu Runtime Engine 4

MenuWindowType 4

Fading Animation 4

DropShadow Effect 5

WebFlyPostBackManager Component 6

AJAX over Intersoft's components 6

WebNotification Component 7

Unbound Notify 7

SmartWebResources 7

Overview 7

Benefits 8

Styles 10

WebDesktopManager Office 2007 Style 10

WebPaneManager Office 2007 Style 11

WebContextMenu/WebMenu Office 2007 Style 11

WebToolBar Office 2007 Style 12

WebToolBar Vista Style 12

WebMenuBar Office 2007 Style 12

WebMenuBar Vista Style 13

PlaceHolderManager Office 2007 Style 13

WebNavPane Office 2007 Style 13
WebExplorerPane Vista Style 14
WebNotification Vista Style 14
WebTab Vista Style 15
WebTab Office 2007 Style 15
The complete WebUI Studio.NET® 2007 components in the fashion of Office 2007 Style 16
Master Page Support 17
Component Designer 2.0 17

I. Breaking Changes

There is no breaking changes in WebDesktop.NET 2.0. All existing object model, features and behaviors remain the same.

II. Upgrading from previous version

Overview

WebDesktop.NET® 2.0 includes major enhancements in user interface, styles, AJAX improvements, wider scenario support for *WebFlyPostBackManager* and *WebNotification*, and rich designer experience.

If you are planning to upgrade your existing web application to use WebDesktop.NET 2.0, please take the following points into consideration:

- Enable SmartWebResources™ for WebDesktop and take advantage of the hassles-free deployment approach. With this new feature, you can now develop web application using *File System Project* (FSP) and *Web Application Project* (WAP) which introduced in Visual Studio 2005 SP1. The SmartWebResources™ enables all client resources including images and background images in the theme to display properly in design-time environment.

  If you are a new web developer and would like to develop ASP.NET 2.0 web application using WebDesktop.NET 2.0, you may want to check out Microsoft's Visual Web Developer Express. Visual Web Developer Express is a lightweight version of Visual Studio 2005, and is provided for free by Microsoft. Since that WebDesktop.NET 2.0 is built on the top of Visual Studio 2005 architecture, Visual Web Developer Express is fully supported as well. To download Visual Web Developer Express, visit http://msdn.microsoft.com/vstudio/express/vwd/download/

- If you are using a lot of *WebMenuBar*, *WebToolBar* and *WebContextMenu* component, consider to use the new *Normal* option for *MenuWindowType*. This new *WindowType* solves numerous display and positioning issues that occurred in Internet Explorer 7.0. Note that these issues are *behavior by design* in Internet Explorer 7.0 which limits the usage of "popup" object to prevent security issues.
- We have added new styles to all of the User Interface components included in WebDesktop.NET. The new styles are mainly focused on Office 2007 and Vista style. Check them out and easily apply the new style to your existing web application.

Assembly References

Our development team is working hard to ensure that you can upgrade your web application as smooth as possible in less efforts. You can easily migrate your existing web application to use version 2.0 by simply changing the assembly references to 2.0 assemblies.

To upgrade an existing ASP.NET 2.0 web application to use WebDesktop.NET 2.0, please do the following:

- Open an existing web application in Visual Studio 2005/Express
- Expand Bin folder
- Remove ISNet.dll, ISNet.WebUI.dll and ISNet.WebUI.WebGrid.dll
- Right click on the Project node then click Add Reference
- Add ISNet.dll, ISNet.WebUI.dll from [Installation Folder]\WebUI.NET Framework 3.0\Bin [Note that the file version should be 3.0.5000.400+]
- Add ISNet.WebUI.WebDesktop.dll from [Installation Folder]\WebDesktop.NET 2.0\Bin
- Rebuild your web application and run in browser

If you are planning to use new SmartWebResources™ feature available in WebDesktop.NET 2.0 for simpler deployment, please refer to Hassle-free deployment through SmartWebResources™ technology [TODO: Link to Framework]

III. What's New in 2.0

WebMenu Runtime Engine

MenuWindowType

In previous version, components that used WebMenu Engine such as WebContextMenu, WebMenuBar, WebToolBar, WebDesktopManager and WebNotification will have its window displayed in a special "popup" type when running in Internet Explorer browser. This behavior is by default and unchangeable in previous version.

The main advantage of the "popup" window type is its ability to display its content out of the browser's window boundary. However, Internet Explorer 7.0 has removed this function from "popup" object and introduced limited positioning by design.

To overcome this limitation and to introduce other possible new features, the *MenuWindowType* is now added WebMenu 2.0 Component.

The default value is set to *Popup* to preserve the behavior in previous version.

Fading Animation

In previous version, animation effect is not possible because WebMenu used Internet Explorer's special popup object.

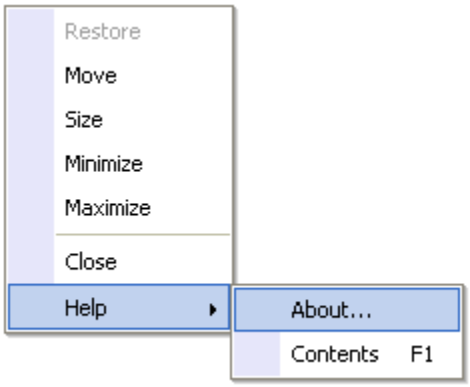
In this new version, you can now add smooth fading animation effect to all components that based on WebMenu. For instance, WebContextMenu, WebToolBar, WebMenuBar and other Intersoft's components that integrates WebMenu Runtime System.

Before you can enable animation, you will need to set *MenuWindowType* to Normal. The default value of MenuWindowType is set to *Popup* to preserve compatibility with previous version.

To enable animation, simply set *MenuStyleSettings.MenuAnimation* to True.

DropShadow Effect

In addition to new animation capability in version 2.0, WebMenu also adds the ability to show *dropshadow* effect when displayed. All sub menu will be automatically applied as well.



Several notes regarding this feature:

- Best used with animation effect to create smooth, visually appealing Menu.
- This feature is not supported in Mozilla browsers since Mozilla has not provide such API.
- As in animation feature, you also need to set the *MenuWindowType* to Normal in order to enable dropshadow effect.

WebFlyPostBackManager Component

AJAX over Intersoft's components

In previous version, *WebFlyPostBackManager* enables you to instantly convert standard .NET controls that perform page postback become AJAX callback, without changes to the codes. However, this function is not supported on complex component or other third party component.

In version 2.0, *WebFlyPostBackManager* takes you a step further by adding support to perform partial rendering over Intersoft's own components such as WebGrid, WebCombo, WebInput or any UI components in the WebDesktop. This means you can now write coding logic in the professional approach by using the server-side programming.

For instance, take a very basic scenario such as adding a *GroupColumn* to WebGrid and then present the new changes when the user clicks on a standard ASP.NET Button. See the following codes:

```
protected void Button1_Click(object sender, EventArgs e)
    {
        WebGrid1.RootTable.TableHeaderVisible = true;
        WebGrid1.RootTable.GroupedColumns.Add(new
            WebGridGroup("ContactTitle"));
        WebGrid1.RequiresUIRefresh = true;
        WebGrid1.DataBind();
    }
```

A new property called *RequiresUIRefresh* has been added to all Intersoft's components. As you noticed in above codes, the *RequiresUIRefresh* property of WebGrid1 instance is set to true. The property is required to be set to true to indicate its user interface needs to be updated by the FlyPostbackManager control.

While standard .NET controls can be automatically updated when they have changed, Intersoft controls do not use similar technique as our controls provide broad range of scenarios. For instance, there are scenarios where some of the data properties need to be changed, but the user interface does not need to be updated. With property control, developers will have more flexibility to control the behavior of the partial update.

To learn more about this new capability, launch *Live Samples* and see the sample at *WebDesktop.NET > What's new in version 2.0 > WebFlyPostBackManager AJAX over Intersoft Control*.

How-to: Enable automatic callback in FlyPostBackManager.

[TODO]

WebNotification Component

Unbound Notify

In its initial concept, WebNotification is a control that acts as a timer which periodically interact with a server side resource, either it is WebForm or WebService. When the server side returns WebNotificationEvent object, the WebNotification control will display an Outlook-like or Messenger-like notification window. This means the WebNotification control should be assigned to a WebForm or WebService resource through its *ServiceUrl* property. This is called as Bound Mode Notification.

New in WebNotification 2.0 is the ability to send notification from the WebForm page's code behind without the needs to create additional WebService or other proxy resource.

Called as *Unbound mode*, this new feature enables basic scenarios that do not require periodic timer to be implemented easily and programmatically in code behind.

For instance, you want to display a one-time notification during first load of a web page to inform visitors that the site is currently under maintenance. You can easily add a WebNotification control to the web page and use Notify method in the web page's code behind. See sample codes in the following.

```
protected void Page_Load(object sender, EventArgs e)
      {
        WebNotification1.Notify(
            "Some of our website's features are disabled due to
            maintenance. We apologize for the inconvenience.",
            "./Images/Warning-32.gif",
            "Maintenance");
      }
```

The *Notify* method used in above sample takes 3 parameters, the description text, the image and the caption text. There are 4 overloads for the *Notify* method with various parameters for your convenience in coding.

SmartWebResources

Overview

Oftentimes, developers found it too complex in order to deploy their web application that used third party components. There are too much things that need to be recalled before a web application can be successfully deployed. For an instance, the javascript files, images and other runtime client files need to be copied in order for the component to run properly.

One of the goal in Intersoft's 2007 product platform is to simplify the deployment process required in Intersoft's controls. Our approach is by developing a new

infrastructure that handles the storage and retrieval system of these client resources so that physical files can be totally eliminated. This new infrastructure is called SmartWebResources™.

WebDesktop.NET 2.0 fully supports SmartWebResources infrastructure. This enables WebGrid control to function perfectly right out-of-the-box after the simple XCOPY deployment.

Both developers and site administrators can now deploy Intersoft's components by simply copying the assemblies in the Bin folder of the project – and totally eliminates the need to copy javascript files, component-specific images and runtime files.

As best practice, it is highly recommended to enable SmartWebResources feature for all Intersoft's components especially WebGrid. Furthermore, WebGrid.NET Enterprise 5.0 exclusively requires *WebDesktop's SmartWebResources Assembly* in order to use some advanced *User Interface* features in WebGrid such as the new ContextMenu engine and runtime engines for dialog boxes and windowing system.

To learn how SmartWebResources works, please read How does SmartWebResources™ work

Benefits

The SmartWebResources feature offers more benefits to WebDesktop.NET components compared to other Intersoft's components since WebDesktop.NET components are mainly User Interface components that contain lot of images, styles and themes.

With SmartWebResources in WebDesktop.NET 2.0, you get the benefit such as:

- *Increasing Productivity* – Quickly apply a sophisticated style from Layout Manager without has to copy the images to root directory of the web application. This is mostly applicable to non-HTTP projects such as *File System Project* (FSP) and *Web Application Project* (WAP).
- *Time Saving* – Instantly deploy your web application without the need to configure the images that needed by the UI component.

To learn about the benefits introduced by SmartWebResources, read Features and benefits introduced by SmartWebResources™

FAQ: Does SmartWebResources feature automatically enabled in WebDesktop.NET 2.0?

By default, Intersoft 2007 product platform have the *SmartWebResources* set to *Automatic* mode. This automatic mode ensures compatibility with existing web application when developers migrated to 2007 platform, while at the same time attempting to utilize SmartWebResources in new web application whenever possible.

When you created a new web application and started to use WebDesktop.NET 2.0, the SmartWebResources is not configured initially. With the automatic mode as default behavior, WebDesktop.NET 2.0will use physical resources since the SmartWebResources is not configured initially.

FAQ: Briefly, what are the files included in WebDesktop SmartWebResources Assembly?

The single assembly of WebDesktop's SmartWebResources contains all required scripts and images that used by the control at runtime. It includes:

- Core Scripts, such as WebDesktopManager.js, WebMenu.js, etc.
- Core Images Set, such as tree lines, background images, command icons etc.
- Theme Images Set, such as the Outlook 2007 or Vista style images contained in *Predefined Style*.
- Stylesheets and other document files that will be automatically loaded when needed by the module in client side.

How-to: Configure SmartWebResources in a new web application

As the best practice in using Intersoft 2007's product lines, you are recommended to perform the following steps after you have created a new web application or migrated an existing web application.

1) Right click on one of the Intersoft's control instance. For instance, WebDesktopManager1.
2) Click Register SmartWebResources™ command in the context menu.

3) In short time after you clicked the command, you should receive a message prompt similar to the following:
"SmartWebResources has been successfully configured for this web application."

If you are receiving an error message, it is likely that you have not installed the product properly. Please note that you can't enable this feature in older products (non 2007-platform products).

To learn more about SmartWebResources, please read Introducing SmartWebResources™

Styles

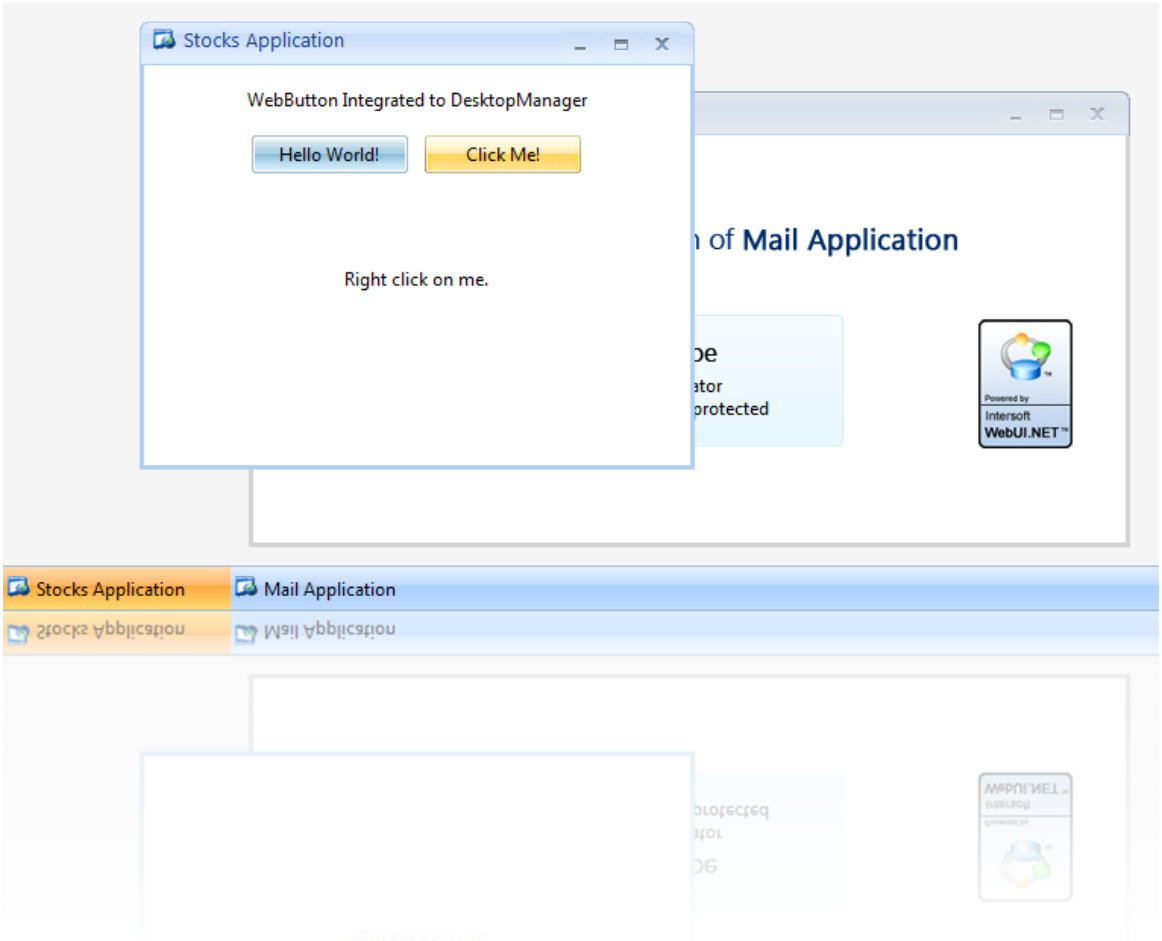
WebDesktop.NET 2.0 is strongly focused on new visual styles and appearance that based on Microsoft's latest user interface breakthrough, such as in Windows® Vista and Office 2007® – which have quickly become the new standards in user interface components industry.

To satisfy your end users with these breakthrough visual styles in your web application, WebDesktop.NET 2.0 has included Office 2007® visual style to most applicable components such as toolbar, menubar, navigation pane and menu system. These new visual styles can be easily applied with one button click from *Layout Manager*.

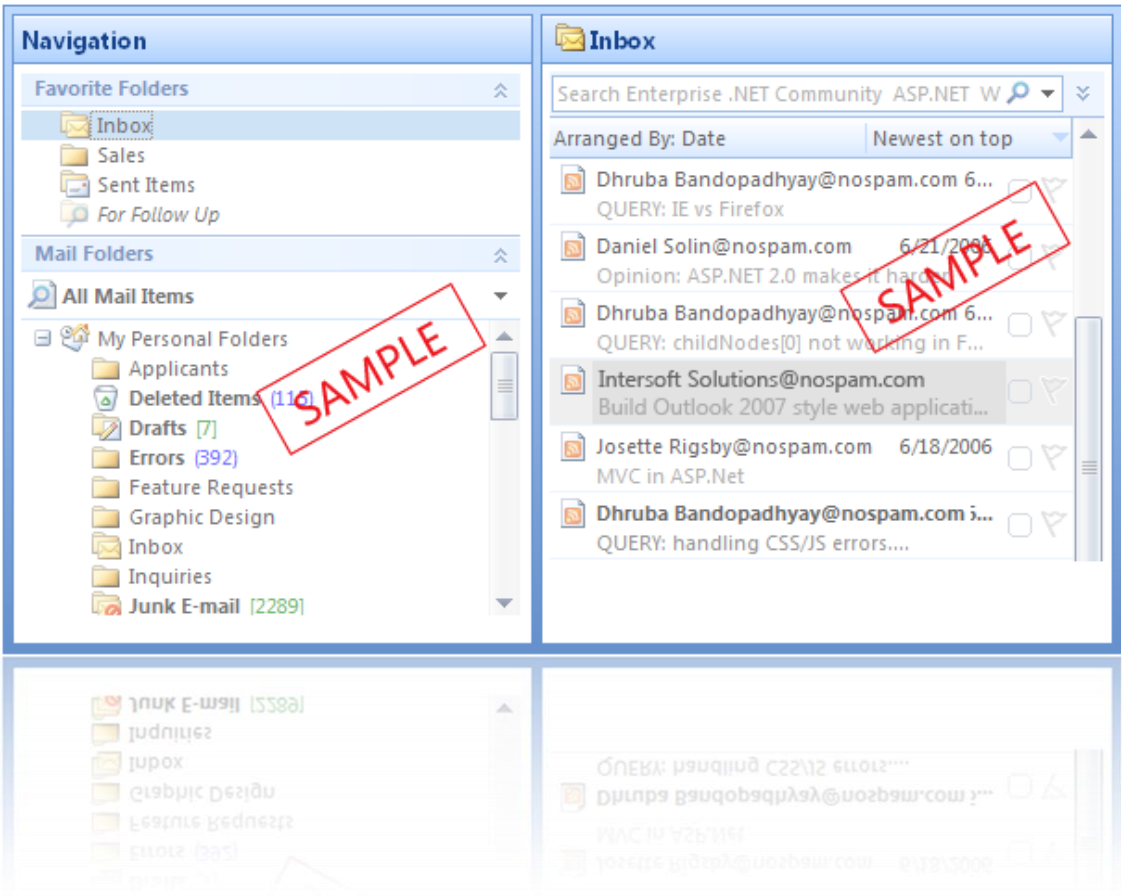
The following images are provided to give you quick reference to the new visual styles included in WebDesktop.NET 2.0.

You can also experience these new visual styles in *Live Samples* > *WebDesktop.NET* > *What's new in version 2.0*.

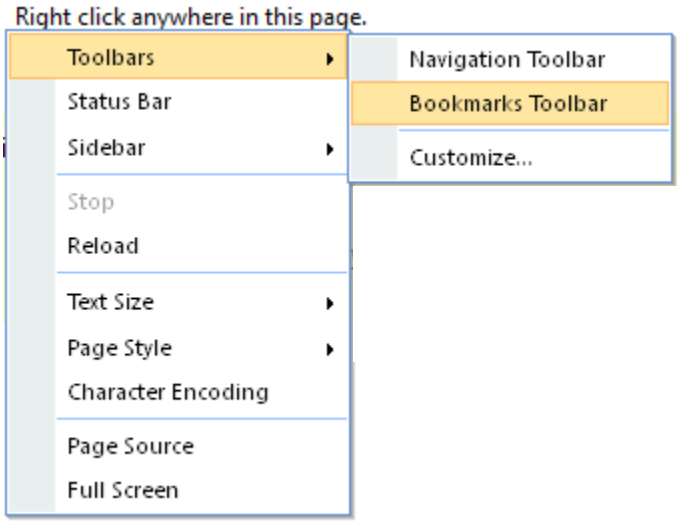
WebDesktopManager Office 2007 Style



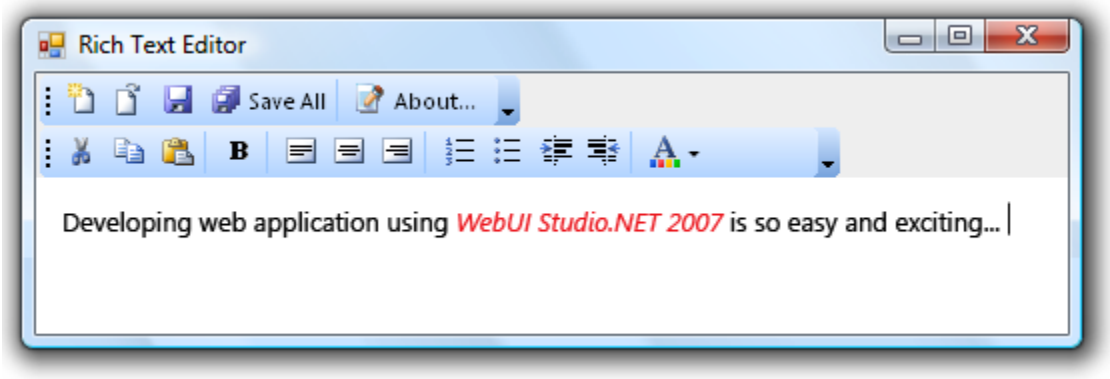
WebPaneManager Office 2007 Style



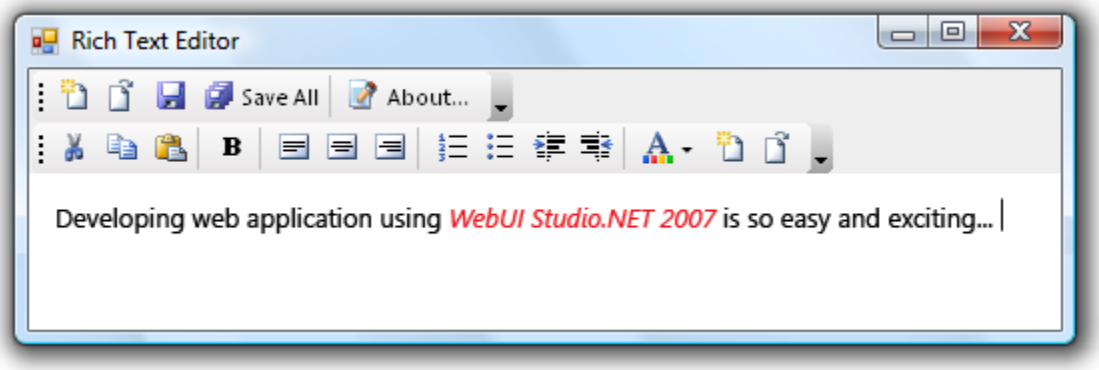
WebContextMenu/WebMenu Office 2007 Style



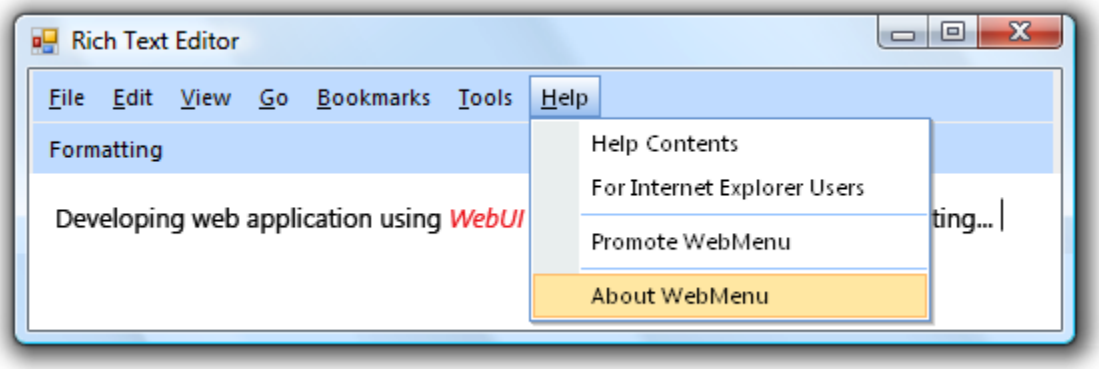
WebToolBar Office 2007 Style



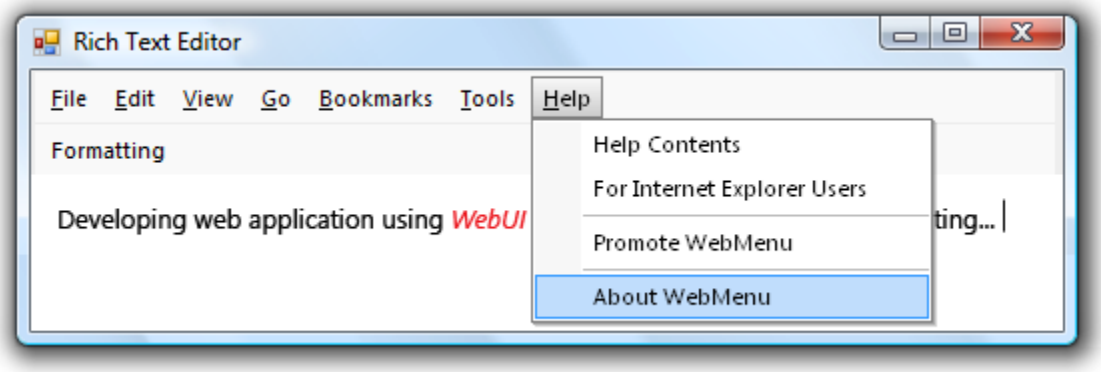
WebToolBar Vista Style



WebMenuBar Office 2007 Style

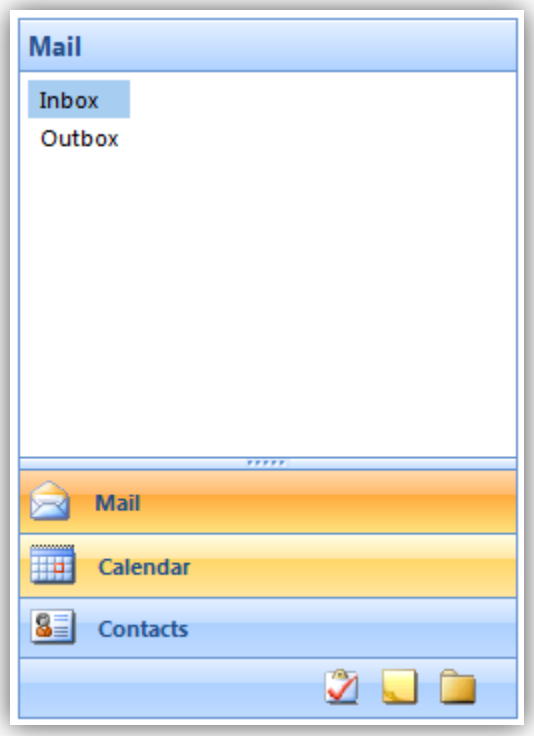


WebMenuBar Vista Style

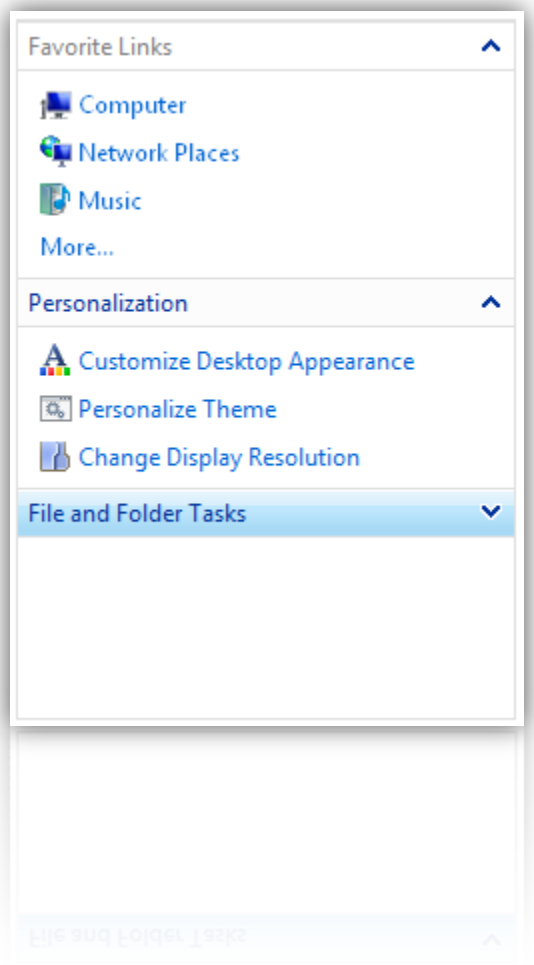


PlaceHolderManager Office 2007 Style

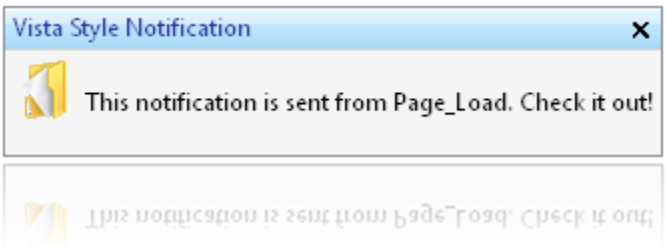
WebNavPane Office 2007 Style



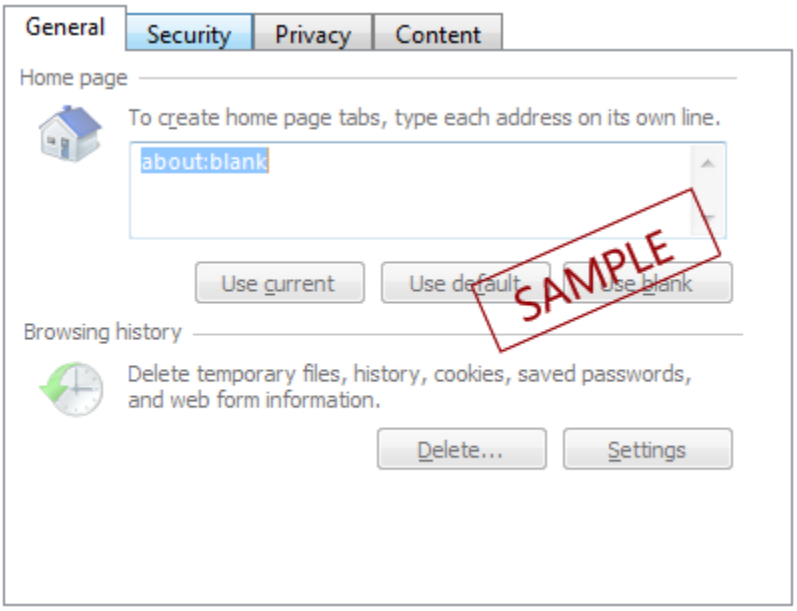
WebExplorerPane Vista Style



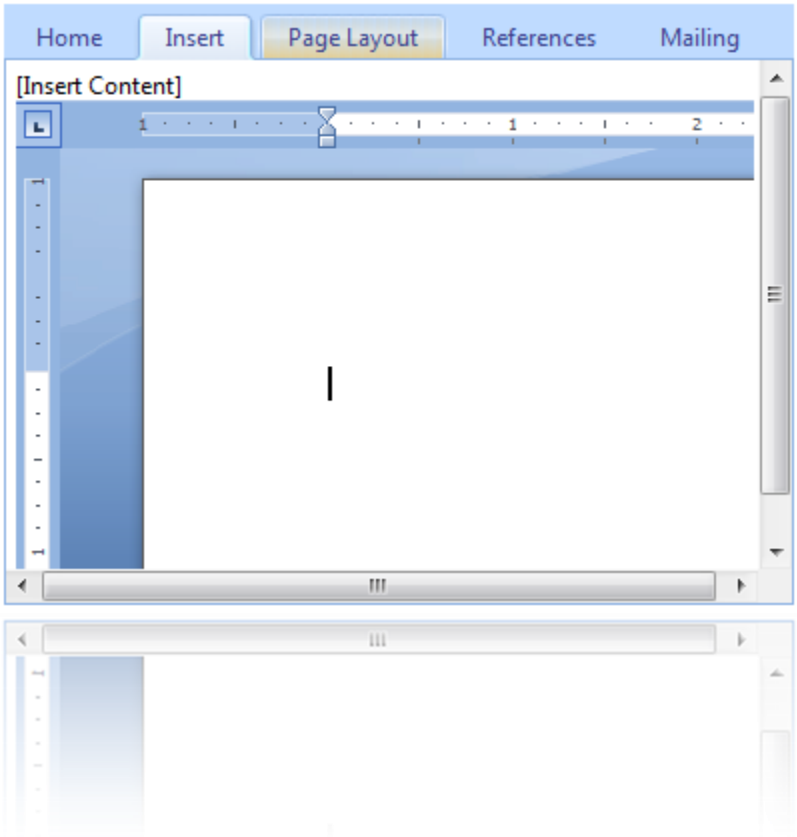
WebNotification Vista Style



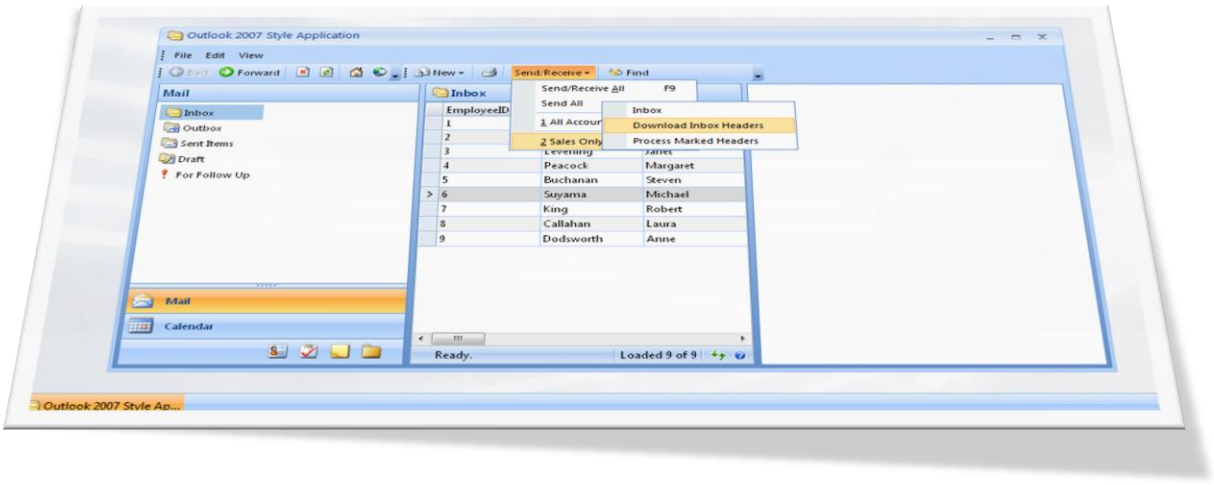
WebTab Vista Style

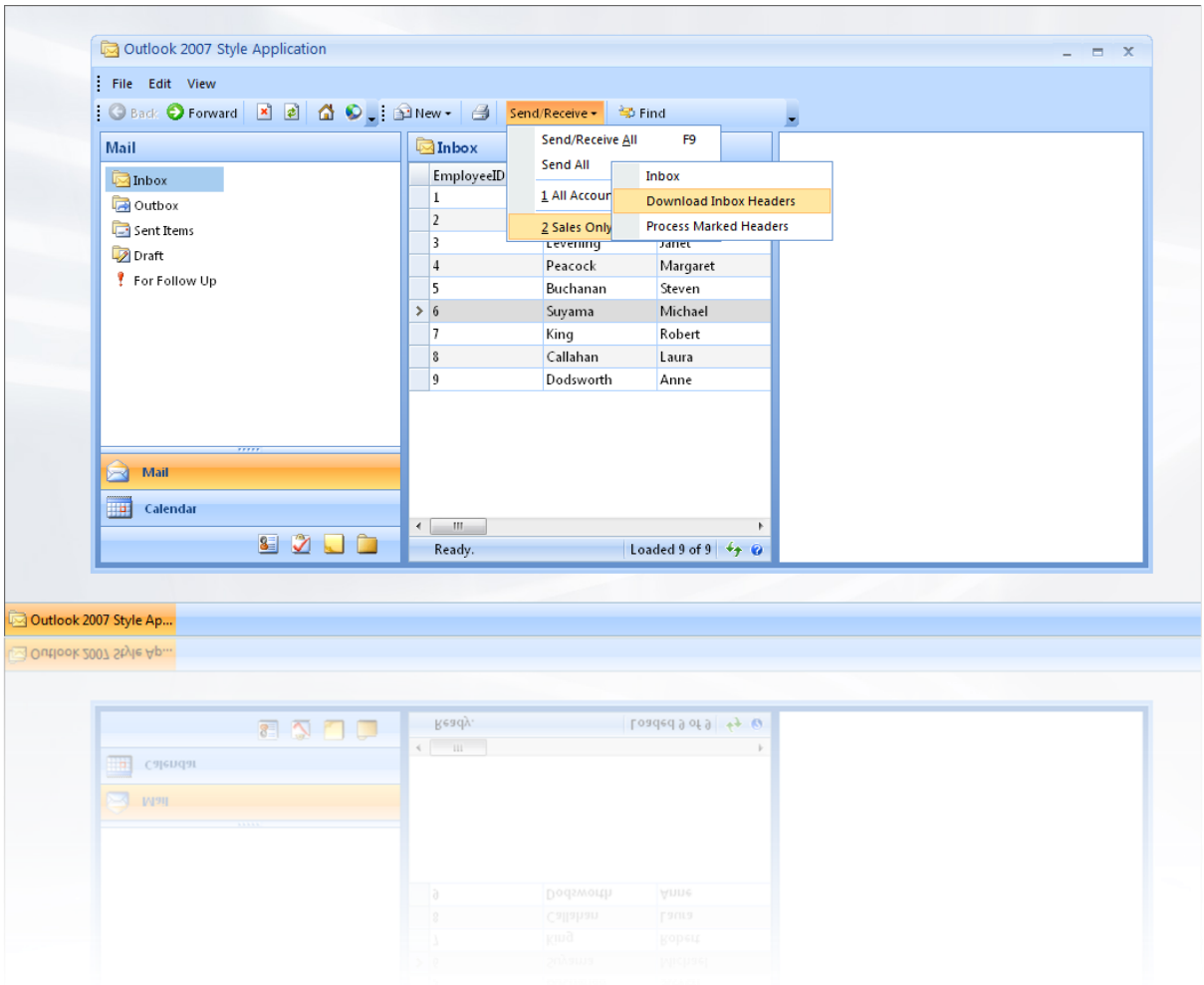


WebTab Office 2007 Style



The complete WebUI Studio.NET® 2007 components in the fashion of Office 2007 Style



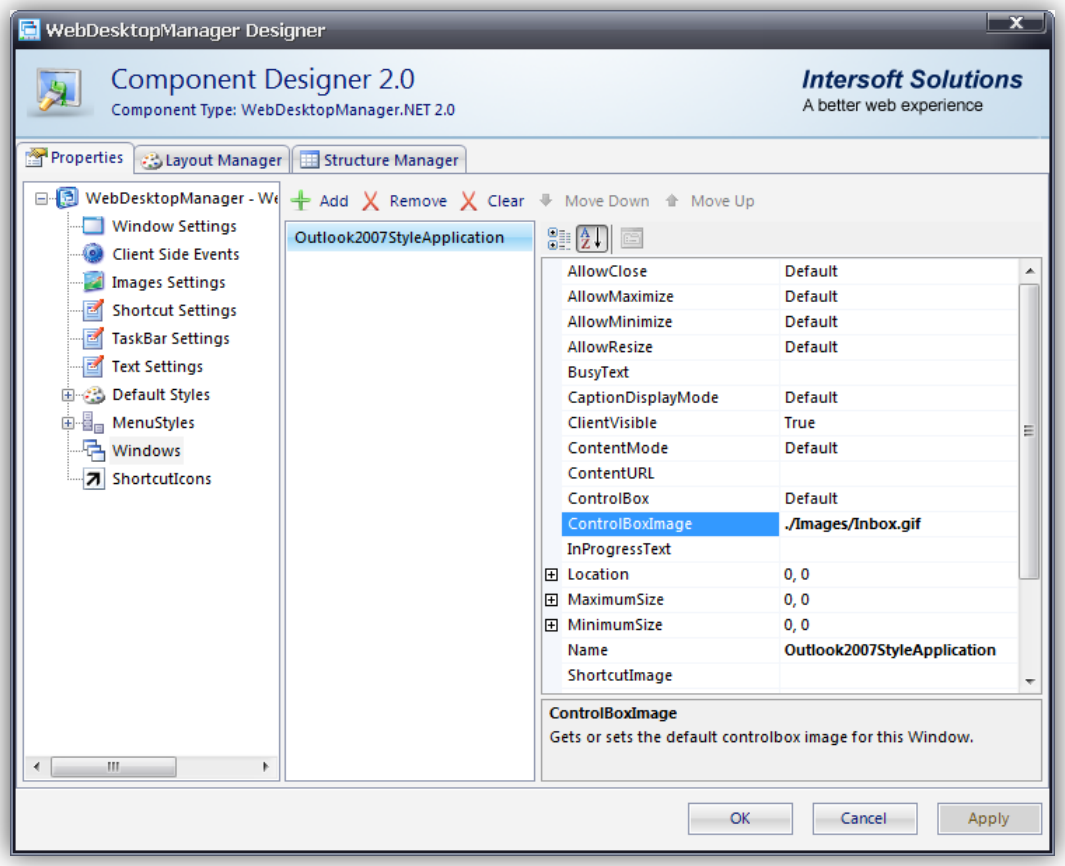


Master Page Support

All components in WebDesktop.NET 2.0 have been refined to fully support Master Page in ASP.NET 2.0. You can use WebDesktop.NET components in either single or nested Master Page.

Component Designer 2.0

WebDesktop.NET 2.0 incorporates new *Component Designer 2.0* as the main editor for all visual components in the collection. The new Component Designer sports consistent Vista style dialog and appearance to improve usability and developer's productivity.



The new Component Designer featuring:

- Redesigned user experience for better layout management.
- Improved Layout Manager for easier user interaction.
- Improved Structure Manager, including the new ability to Preview Xml.