# Branding Guide

Microsoft | Services

## Contents

# Introduction

Welcome to the *Branding Guide*. This guide covers branding (the process of designing, building, and deploying the visual artifacts that create the complete end-user experience), and branding components (in particular master pages and page layouts). Upgrade considerations, deployment considerations, and new branding capabilities are also discussed. Additional references are provided at the end of the document.

Information in this guide applies to Microsoft® SharePoint® Server 2010 and Microsoft® SharePoint® Foundation 2010and may not necessarily apply to earlier or later versions. The guide is organized as follows:

- Branding Overview. An overview of how branding operates with SharePoint Server, upgrade and deployment considerations, and new branding capabilities.
- Master Pages. Master pages defined, upgrade, development, and deployment considerations, and usage of new and existing master pages.
- Page Layouts. Page layouts defined and upgrade considerations.
- References. Further references for branding.

# Branding Overview

Branding is the process of designing, building, and deploying the visual artifacts that create the complete end-user experience. Branding can be applied to particular site collections or subsites within the Web application. Branding in Web Content Management (WCM) solutions is especially important because of the need to have a customized and differentiating user experience to meet business objectives. For example, if you look at the Ferrari Web site (http://www.ferrari.com/Pages/Country_Selector.aspx), it is quite customized and looks nothing like out-of-the-box SharePoint Server 2010 site templates.

For more information about branding and WCM, see the MSDN article Web Content Management and Branding (http://msdn.microsoft.com/en-us/library/ms569214(office.14).aspx).

## Branding Customization

There are two general approaches to take when customizing site branding:

- Create or customize a SharePoint Server 2010 theme.
- Create custom branding (for example, cascading style sheets (CSS) and master pages).

SharePoint Server 2010 themes provide a quick and easy way to apply colors and fonts to SharePoint Server 2010 sites. When a theme is applied to a site, the color of most page elements changes (like background images, text, and hyperlinks). The fonts used for some page elements (like titles) also change. Themes can be used with the standard SharePoint Server 2010 site templates or with custom master pages. Themes can also be created by site owners and applied to any sites the site owner controls.

An advantage of using SharePoint Server 2010 theme functionality is that developer resources are not needed for site owners and users with designer rights to make basic changes to a site. Themes are a simple method for branding as they do not affect the overall layout of a site.

By default, a theme is only applied to a site for which a specific theme is selected. If the site is a publishing site, or if the publishing feature was enabled for a site, you can choose to either inherit the theme from the parent site or specify a theme that will be used by the site and all sites that inherit from it.

For more information about themes, see the Microsoft TechNet article Themes overview (SharePoint Server 2010) (http://technet.microsoft.com/en-us/library/ee424397(office.14).aspx).

Creating custom branding typically involves two teams, a design team and a development team. The design team creates wire frames, color schemes, images, and sample pages, and is critical to designing the overall user experience and flow based on business requirements. The development team incorporates these designs, uses them to build SharePoint Server 2010 components, packages them into Windows® SharePoint® Services Solution Package files (.wsp files), and creates deployment scripts as necessary.

The following list includes the most common SharePoint Server 2010 artifacts for a custom-branded site:

- Master pages
- Page layouts
- Content pages
- CSS
- Script files
- Images
- Navigation
- Custom controls
- Web Parts

Web 2.0 styles are also available in WCM solutions given the popularity of that design metaphor (for example, it's typical to see AJAX communications, DHTML pop-ups, and transparencies on many Internet sites today). While many of the capabilities are not new, SharePoint Server 2010 provides improved support to simplify usage of Web 2.0 styling.

For more information about client programming, see the MSDN article What's New: Client Object Model (http://msdn.microsoft.com/en-us/library/ee535231(office.14).aspx).

For more information about new client capabilities, see the MSDN article What's New: Silverlight Integration and the Fluid Application Model (http://msdn.microsoft.com/en-us/library/ee539062(office.14).aspx).

## Upgrade Considerations

Upgrading to the SharePoint Server 2010 user interface (UI) presents unique considerations. The Visual Upgrade feature allows gradual conversions of individual subsites from the Microsoft® Office SharePoint® Server 2007 UI to that of SharePoint Server 2010 at the convenience of site owners. This is supported by upgrading the binaries using either the in-place or database attach upgrade processes and then running the Visual Upgrade feature on site collections as desired.

The Office SharePoint Server 2007 UI can be kept and utilized while using SharePoint Server 2010 in backward compatibility mode with the Visual Upgrade feature. The Visual Upgrade feature allows the server administrator or site owner to determine when (and if) the new look for SharePoint Server 2010 is used for a particular site collection. Server administrators can also choose to let site owners make the choice after upgrade to keep the old look and feel for all sites or use the new look.

**Note**: It is strongly recommended that you validate site functionality by using Visual Upgrade for each site prior to switching to SharePoint Server 2010 UI.

SharePoint sites should work in most cases when run in backward compatibility mode. However, when you upgrade the sites to the SharePoint Server 2010 UI there are some standard steps to follow. The actual steps vary depending on customizations. For custom-branded solutions (like WCM solutions) you will need to follow these steps.

**Note**: The following steps are specific to branding and the visual upgrade. They do not describe the larger upgrade process (including issues like database backup, building out the SharePoint Server 2010 farm, and configuring authentication). For more information on upgrading, see the *Web Content Management Upgrade Guide* in the Web Content Management for SharePoint Server 2010 offering documentation.

To perform a visual upgrade using the Visual Upgrade feature:

1. On the Site Settings page, in the Visual Upgrade section, click **Update All Sites.**
2. After running the Visual Upgrade feature, resolve any UI-related issues. This includes:
    a. For master pages, use Microsoft® SharePoint® Designer 2010 to:
        i. Add placeholders for the new Microsoft® Office Ribbon.
        ii. Add the **VersionedPlaceHolder** control to the page to support the look and feel of Office SharePoint Server 2007 or SharePoint Server 2010 (depending on the specified UI preference).
        iii. Fix any custom CSS issues that may interfere with the SharePoint Server 2010 UI (like Office Ribbon conflicts).
        iv. Fix any JavaScript issues that may interfere with the SharePoint Server 2010 UI.
        v. Change **12** object model references to **14**.
    b. For page layouts, use SharePoint Designer 2010 to:
        i. Change **12** object model references to **14**.
        ii. Fix any Web Parts that no longer function correctly as a result of the upgrade (described below).
        iii. Remove the CSS references from any page layouts and add them to the master page CSS (if possible).
    c. For Web Parts, use Microsoft® Visual Studio® 2010 to:
        i. Fix any calls that reference deprecated application programming interfaces (APIs).
        ii. Fix any **12** object model references.
        iii. Add feature upgrade support for .wsp files. For more information on upgrade support options, see the MSDN article Upgrading Features (http://msdn.microsoft.com/en-us/library/aa544511(office.14).aspx).
        iv. Migrate the Web Parts to a sandbox solution (if appropriate).
    d. Replace custom controls with SharePoint Server 2010 controls where appropriate (for example, the SharePoint Server 2010 ratings controls).
    e. Package all changes made through performing these step into .wsp files using Visual Studio 2010 or a third-party SharePoint package generation tool and redeploy the .wsp files.
3. Perform the database upgrade again, install the updated .wsp files, and then run the Visual Upgrade feature to verify the upgrade was performed correctly.

It is suggested that the architecture design team configure any as-needed services (such as Search services, Microsoft Business Connectivity Services (BCSs), and Shared services). For more information, see the Microsoft TechNet article Manage service applications (SharePoint Server 2010) (http://technet.microsoft.com/en-us/library/ee704544(office.14).aspx). Table 1 lists upgrade considerations specific to branding deployment.

Table 1: Branding Deployment Upgrade Considerations

| **SharePoint Server 2010 Capability** | **Upgrade Considerations** |
|---|---|
| Master pages | Typical changes required for master pages include:<br>• Assembly references.<br>• CSS references.<br>• Adding the Office Ribbon.<br>• Repackaging in .wsp files for SharePoint Server 2010 deployment. |
| CSS | Typical changes required for CSS include:<br>• Any customizations to core.css files may be lost.<br>• CSS is now split into multiple files. Previous methods that were required for core.css file overrides may need to be reviewed and changed accordingly. |
| Web Parts | Typical changes for Web Parts include:<br>• References to deprecated Office SharePoint Server 2007 object model APIs upgraded to SharePoint Server 2010 object model references.<br>• Leveraging content query Web Part improvements. |

| | |
|---|---|
| Feature deployment | Typical changes for feature deployment include:<br>• Customizing features validated as part of the dry-run upgrade process.<br>• Resolution of deprecated Office SharePoint Server 2007 object model calls and repackaging Office SharePoint Server 2007 custom features in Visual Studio 2010.<br>• Using .wsp files for content types, list definitions, list instances, and page layouts.<br>• Using custom workflows (to support approval requirements) and custom document converters.<br>• Using Windows PowerShell™ for deployment scripts.<br>• Using sandbox-based solutions where appropriate.<br>• Using the SharePoint Server 2010 feature upgrade process. |

## New Branding Capabilities

There are many new capabilities for branding in SharePoint Server 2010. Some of the new capabilities below are platform improvements that are necessary to be aware of (like CSS and accessibility improvements). These capabilities may not necessitate changes to corresponding WCM artifacts used in Office SharePoint Server 2007, but are important to understand from a platform perspective. Other capabilities, such as the client object model, will require changes to artifacts. Table 2 is a list of new branding capabilities for SharePoint Server 2010.

Table 2: New Branding Capabilities

| Capability | New for SharePoint Server 2010 |
|---|---|
| Client object model | • JavaScript wrappers for SharePoint Server 2010 object models.<br>• Microsoft® Silverlight wrappers for SharePoint Server 2010 object models. |
| CSS and script improvements | • Core.css files are now split into multiple files. SharePoint Server 2010 only downloads what is necessary.<br>• Script on demand allows the delaying of script downloads until they are needed.<br>• Ability to use debug versions of JavaScript files for development and debugging. Optimized and obfuscated versions of JavaScript are used for production.<br>• Support for **AJAX**, **JSON**, and **RESTful** APIs. |

| | |
|---|---|
| Accessibility and Compliance | • Web Content Accessibility Guidelines (WCAG) 2.0 AA standards compliant markup is generated.<br>• Reduced use of tables. This is generally accepted as the preferred way to apply styling. Reducing the number of tables also makes the page more readable as well as easier to modify and update.<br>• Cross-browser support for Windows® Internet Explorer®, Firefox, and Safari. For more information on browser support, see the Microsoft TechNet article Plan browser support (SharePoint Server 2010) (http://technet.microsoft.com/en-us/library/cc288142(office.14).aspx). |
| Silverlight Web Parts | • Users can now quickly launch Silverlight applications on SharePoint Server 2010 pages. |
| Fluid application model | • Can be used to host applications outside of SharePoint Server 2010.<br>• Access to SharePoint Server 2010 data if permission is given. |
| Content query Web Part | • Uses dynamic filters to filter based on other page fields or from a value in the **QueryString**.<br>• Simplifies working with item styles by introducing slots. A slot is a marker in the item style template which is filled with content at run time. Use the property pane in the editor to map slots to fields.<br>• Working with the **CommonViewFields** property is much simpler. Instead of having to export and update the value and then import the changes, you can now manage all of this within the property editor. |

The new capabilities in SharePoint Server 2010 enable additional scenarios and enhance productivity. The capabilities have not yet been validated in real-world, large-scale WCM deployments. Most WCM solutions demand fast, reliable responses. Leveraging caching techniques where appropriate is suggested. The impact of these new capabilities on reliability, performance, and caching will be discovered as more solutions are developed on SharePoint Server 2010. Through these implementations, best practices for these new features will be derived and propagated accordingly.

## Master Pages

A master page defines the outer frame of the Web page. It contains the elements that all pages in the site share, and it provides a single place to control common elements of the page. Typically, a site uses a single master page, although large Internet sites might use more. For example, a corporate Web site that is used to publicize more than one product line, or multiple brands, could use separate master pages so that the content for each product is properly branded.

There are three parts to any page rendering. The master page represents one part. Page layout and page instance are the other two parts. The master page provides the outer frame (like the header, footer, navigation, CSS, and script references). The body of the page is structured and rendered through the page layout along with the page instance or content page.

## Upgrade Considerations

Custom master pages built with Office SharePoint Server 2007 will continue to work using the Office SharePoint Server 2007 UI. However, they will not work with SharePoint Server 2010 UI unless you upgrade the master page.

Typical master page upgrades include:

- Adding new content place holders that are required for SharePoint Foundation 2010.
- Adding the Office Ribbon.
- Adding controls that are required for SharePoint Foundation 2010 to function properly.
- Updating the assembly references for SharePoint Server 2010.

For more information about upgrading master pages, see the MSDN article Upgrading an Existing Master Page to the SharePoint Foundation Master Page (http://msdn.microsoft.com/en-us/library/ee539981(office.14).aspx).

## SharePoint Server 2010 Master Pages

SharePoint Server 2010 provides several master pages out of the box. SharePoint Server 2010 uses the v4.master page as its primary master page. Other master pages that are available for use are located in the **%RootDirectory%\Common Files\Microsoft Shared\Web Server Extensions\14\Template\Global** folder. As long as this master page is not customized, its page definition is cached on the front-end Web server and shared across sites. If the master page is edited for a particular SharePoint Foundation 2010 site, the edited copy of the master page file is stored in the site's content database.

SharePoint Server 2010 master pages can be edited to create a customized master page. Customized master pages can be deployed by using either SharePoint Designer 2010 or .wsp files. When a customized master page is deployed using SharePoint Designer 2010, it is persisted in the content database and, as a result, has performance implications when the page is rendered. However, if you use a .wsp file, the customized master page can be deployed directly to the Web front-end servers. This avoids any performance issues as the customized master page is local to those Web front-end servers.

Table 3 lists some new and existing master pages and how they are used in terms of branding in SharePoint Server 2010 and SharePoint Foundation 2010.

Table 3: Master Pages and Branding Purposes

| Master Page | Branding Purposes |
|---|---|
| V4.master | • Team site master page.<br>• User content pages.<br>• _layouts pages. |
| _starter.master<br>_foundation_starter.master | • Used for application UI, (like the Search feature in SharePoint Server 2010 and Microsoft® Office 2010 Web applications.<br>• Good to use if you do not need site navigation.<br>• Good to use if you do not have the Office Ribbon. |
| Simple.master | • Used for error pages and logon pages.<br>• Not customizable, but pages can be replaced. |
| Default.master (in SharePoint Server 2010 for backward compatibility with Office SharePoint Server 2007) | • UI has same look and feel as Office SharePoint Server 2007.<br>• No Office Ribbon. |

There have been significant changes to the UI for SharePoint Foundation 2010, including the addition of the Office Ribbon. By default, the v4.master page includes the Office Ribbon. Many of the commands previously found in menus and toolbars now exist in the Office Ribbon. As a result, if the existing master page does not contain the Office Ribbon, many commands will be unavailable.

There are some controls that were previously included with the default master page that have been moved into the new Office Ribbon UI. If you plan to update an existing master page with the Office Ribbon, you need to remove the following controls from the master page.

- **Publishing Console** - **<PublishingConsole:Console>**
- **Site Actions** menu - **<PublishingSiteAction:SiteActionMenu>**
- **Sign-in** and **Logon**

These controls are added when you incorporate the Office Ribbon into the master page.

**Note**: Before implementing the branding of the master page, it is strongly suggested that you or the design team performing the branding understand the personas of the primary users to better provide and remove functionality as necessary.

## Development and Deployment

SharePoint Designer 2010 is ideally suited for creation of new master pages or modification to existing master pages. It provides an editing environment with designer and code views for page editing. In addition, SharePoint Designer 2010 provides file check-in and check-out, versioning control, publishing, and links to the site for published pages approval.

However, for deployment it is typically recommended that the master page be copied from the SharePoint Designer 2010 environment and added to a Visual Studio 2010 product to be deployed as part of a .wsp file. This approach simplifies deployment to multiple environments (like Development, Quality Assurance, Stage, and Production). You can also use Visual Studio 2010 to take advantage of greater source control through Microsoft® Visual Studio® Team System 2010 Team Foundation Server and schedule nightly builds and deployment processes.

Customized master pages can be deployed by using either SharePoint Designer 2010 or .wsp files. In terms of performance benefits, if you use a .wsp file the customized master page can be deployed directly to the Web front-end servers. This avoids any performance issues as the customized master page is local to those Web front-end servers.

## Page Layouts

A page layout is a template that is used in conjunction with a master page to control the look, feel, and content of a page. Page layouts (like master pages) dictate the overall look and feel of the SharePoint Server 2010 site. Unlike master pages (which contain controls that are shared across multiple page layouts like navigation, Search, and language preferences for multilingual sites), page layouts contain field controls and Web Parts.

Each page layout has an associated content type that determines the kind of content that can be stored on pages based on that page layout. SharePoint Server 2010 provides three default publishing content types: page, article page, and welcome page. Each content type contains columns that define content that can appear on a page and metadata associated with the page. Content in a page is stored as SharePoint Server 2010 list items in the Pages document library. When users view or edit the page, content is pulled from the SharePoint Server 2010 list and displayed in field controls.

The top-level SharePoint Server 2010 site for a site collection has a special document library called the Master Page and Page Layout Gallery. All page layouts and master pages are stored in this library. By default, SharePoint Server 2010 creates a master page gallery for every site. However, you can only create new pages using the page layouts stored in the Master Page and Page Layout Gallery.

### Upgrade Considerations

Custom page layouts built with Office SharePoint Server 2007 will continue to work using Office SharePoint Server 2007 UI. However, they will not work with SharePoint Server 2010 UI if you do not upgrade the page. For more information on upgrading, see the *Web Content Management Upgrade Guide* in the Web Content Management for SharePoint Server 2010 offering documentation.

# References

## Branding Overview

Upgrading an Existing Master Page to the SharePoint Foundation Master Page: http://msdn.microsoft.com/en-us/library/ee539981(office.14).aspx

Master Pages: http://msdn.microsoft.com/en-us/library/ms443795(office.14).aspx

SharePoint Page Types: http://msdn.microsoft.com/en-us/library/aa979592(office.14).aspx

The Ribbon in SharePoint Foundation: http://msdn.microsoft.com/en-us/library/ee540027(office.14).aspx

Web Parts in SharePoint Foundation: http://msdn.microsoft.com/en-us/library/ms476318(office.14).aspx

Using the Client APIs: http://msdn.microsoft.com/en-us/library/ee537564(office.14).aspx

Cascading Style Sheets Class Usage in SharePoint Foundation: http://msdn.microsoft.com/en-us/library/ms438349(office.14).aspx

What's New: UI Improvements: http://msdn.microsoft.com/en-us/library/ee538245(office.14).aspx

Using the Client Object Model: http://msdn.microsoft.com/en-us/library/ee535231(office.14).aspx

New Client capabilities: http://msdn.microsoft.com/en-us/library/ee539062(office.14).aspx

## Master Pages

Upgrading an Existing Master Page to the SharePoint Foundation Master Page: http://msdn.microsoft.com/en-us/library/ee539981(office.14).aspx

Master Pages: http://msdn.microsoft.com/en-us/library/ms443795(office.14).aspx

SharePoint Page Types: http://msdn.microsoft.com/en-us/library/aa979592(office.14).aspx

The Ribbon in SharePoint Foundation: http://msdn.microsoft.com/en-us/library/ee540027(office.14).aspx

## Page Layouts

Page Layouts and Master Pages: http://msdn.microsoft.com/en-us/library/ms543497(office.14).aspx

Page Layout Model: http://msdn.microsoft.com/en-us/library/ms544928(office.14).aspx

Page Processing Model: http://msdn.microsoft.com/en-us/library/ms498550(office.14).aspx

Building Block: Pages and User Interface:
http://msdn.microsoft.com/en-us/library/ee539040(office.14).aspx

SharePoint Page Types:
http://msdn.microsoft.com/en-us/library/aa979592(office.14).aspx

Common Page and Site Customization Tasks:
http://msdn.microsoft.com/en-us/library/ms563713(office.14).aspx

Custom Caching Overview:
http://msdn.microsoft.com/en-us/library/aa589700(office.14).aspx