

The SharePoint Designer: Friend or Foe

By Ted Pattison
Ted Pattison Group
May 14th, 2008

As SharePoint sites continue to spread like wildfire across the corporate landscape, many companies are contemplating when and where to use the SharePoint Designer (SPD) to apply their corporate branding and to utilize SPD productivity features such as creating and formatting DataView Web Parts. While the SPD makes performing customization work quick and easy, there are some things you should always keep in mind. First, you should acknowledge that SPD customizations are typically impractical or impossible to reuse across site collections or to migrate from a staging environment to production.

When you are working with a site collection of WSS team sites, you can use the SharePoint Designer to customize a top-level site's master page and to add a custom CSS file. However, the SharePoint Designer cannot synchronize child sites within a site collection to use the same version of your customized master page. This can lead to scenarios where you are forced to copy and paste the edits to your master page edits across multiple master page files.

Things get better when you use SPD together with MOSS and Publishing portals. A Publishing portal can automatically synchronize child sites to use the same master page and the same CSS file as the top-level site. This means the SharePoint Designer can be used as an effective tool to brand all the sites within an entire Publishing portal just by customizing a single master page file and a single CSS file.

Remember that changes made with the SharePoint Designer are always made as one-off customizations within the context of one particular site. This is why customization made with the SharePoint Designer can be tricky or impossible to migrate between site collections or between farms. Also, it's impractical to integrate site customizations made with the SharePoint Designer into a source code management system such as Team System.

It's important to understand from an architectural viewpoint that it's possible to develop a custom SharePoint branding solution that leverages the capabilities of WSS to integrate custom master pages and custom CSS files at the level of the site collection. That means you can have all the sites within a site collection link to a single master page file without having to buy MOSS. Furthermore, you can create your development solution as a Visual Studio project and configure it for deployment using a WSS solution package. This is the best approach by far if you are serious about reusability across site collections and if you are concerned about flexibility and ease of deployment of the same solution in different environments.

Using Visual Studio to generate a solution package makes it possible to deploy your branding solution to any farm running WSS 3.0 or MOSS. It also makes it easy to create a single source file for each master page and CSS file which can be reused across multiple site collections or across farms. Furthermore, Visual Studio approach can be far more appealing than using the SharePoint Designer because it allows you to check all your source files into a source code management system and to move your development efforts from a development farm to a staging farm for quality assurance testing and then into production.

Understanding the Difference between Customization versus Development

As you begin to design software for the WSS platform, it is critical that you differentiate between customization and development. WSS is very flexible for users because it was designed to support high levels of customization. In one sense that is very valuable because you don't need to be a developer to build a complex and highly functional site.

As you probably know, a sophisticated SharePoint user can create and customize lists and document libraries. Users can also customize and personalize the appearance of their pages through the use of Web Parts.

A Ted Pattison Group Whitepaper

Premiere SharePoint Training - <http://www.TedPattison.net>

Advanced users can even provide an effective branding solution by using the SharePoint Designer to customize the master pages and Cascading Styles Sheets associated with sites.

You should remember that WSS records every site customization by modifying data within the content database. This is true whenever you create a new list or whenever you customize an existing list by adding new columns and views. It's also true for all types of site customization you can perform using The SharePoint Designer.

The fact that all site customization is recorded as modifications to the content database is both a strength and a weakness for WSS. It is a strength because it provides so much flexibility to users and site administrators with respect to performing ad-hoc customization. It is a weakness for an architectural perspective because customization changes are hard to version and to make repeatable across multiple sites.

Think about a typical .NET developer in an enterprise environment who is working on a standard ASP.NET development project. All the source files for an ASP.NET application exist within a single directory on a development workstation. When the developer has completed the site's design and implementation, you can add all the site's source files to a source control management system such as Microsoft Visual SourceSafe. This makes it possible to have a disciplined approach to deploying and updating the site after it has gone into production. You can also elect to push changes out to a staging environment where the site's pages and code can be thoroughly tested before they are finally pushed out to the production environment.

Ask yourself the following questions: How do you conduct source control management of customization changes made with the browser or the SharePoint Designer? How do you make a customization change to a list definition or a page instance and then move this change from a development environment to a staging environment and finally to a production environment? How do you make a customization change within a site and then reuse it across a hundred different sites? Unfortunately, these questions have very tough answers, and usually you will find that a possible solution isn't worth the trouble.

Fortunately, SharePoint Development allows you to work at a level underneath the WSS customization infrastructure. To be more specific, you can work with the low-level source files to create underlying definitions for things like lists and page templates. These low-level source files do not live inside the content database, but instead they live within the file system of the front-end Web server. Working at this level is more complex and has a significantly steeper learning curve. However, it lets you centralize source code management and have a more disciplined approach involving code sign-off when moving pages and code from development to staging to production. This approach also makes versioning and reuse of code far more manageable across multiple sites, Web applications, and farms.

I encourage you to differentiate between customization and development according to these criteria. WSS customizations are updates to a site accomplished by making changes to the content database, generally through the Web browser or the SharePoint Designer. A site customization never requires touching the front-end Web server. WSS development, on the other hand, involves working with files that must be deployed to the file system of the front-end Web server. WSS development includes creating page templates and list definitions, as well as creating components deployed in compiled assemblies such as custom Web Parts, event handlers, and workflow templates. WSS development at this level is also referred to as developing provisioning components.

Keep in mind that development requires a degree of control over the front-end Web server because of the need to push out files such as page templates and assemblies. If you work in a WSS environment where you are not permitted to make any changes to front-end Web servers, you will not be able to deploy any of your development efforts. In such an environment, you will be restricted to using only customization techniques that can be accomplished by making changes to the content database.

A Ted Pattison Group Whitepaper

Premiere SharePoint Training - <http://www.TedPattison.net>

Although customization provides users with a great deal of flexibility, development is far more powerful. It's a much better approach for creating reusable WSS solutions, as well as the best way to optimize the response times and throughput for high-traffic sites.

Summary

I have discussed why going with a WSS or MOSS development solution has benefits over performing customizations with the SharePoint Designer from the perspective of reusability and deployment flexibility. However, in no way am I trying to dissuade you from using the SharePoint Designer. I just want to encourage you to use it in the appropriate scenarios. Here's a checklist of items that must all be true if you want to use the SharePoint Designer on a live production site.

- Reusing SPD customizations across site collections is not important
- Your company does not require code sign off with your customization/development efforts
- Your company does not require you to move your customization/development efforts between multiple environment such as development, staging and production

However, if you answered "no" to any of these questions, you will be better off developing a solution that is deployed via a WSS solution package. However, the SharePoint Designer can still be your friend. It's just that you should use the SharePoint Designer to customize a "dummy site" in a development environment and then after the fact you can copy-and-paste the HTML, CSS styles and JavaScript from the SharePoint Designer editors into Visual Studio source files. Then you really get the best of both worlds. You get to leverage the productivity features and the WYSIWYG environment of the SharePoint Designer while not being constrained by the limitations of SharePoint customization work.

Continuing Education on these topics from Ted Pattison Group

Learning more about working with the SharePoint Designer

- **Hands-on Training: SAB301: Building Enterprise Solutions with SharePoint Server 2007**
<http://www.tedpattison.net/Courses/SAB301.aspx>
- **Virtual Online Training: WC-SAB301: Building Enterprise Solutions with SharePoint Server 2007**
http://www.tedpattison.net/Courses/WC_sab301.aspx

Learn more about SharePoint Development

- **Hands-on Training: GSA401: The Great SharePoint Adventure**
<http://www.tedpattison.net/Courses/GSA401.aspx>
- **Hands-on Training: WSS401: Inside Windows SharePoint Services 3.0**
<http://www.tedpattison.net/Courses/WSS401.aspx>
- **Virtual Online Training: WC-WSS401: Inside Windows SharePoint Services 3.0**
http://www.tedpattison.net/Courses/WC_WSS401.aspx