

Customizing SharePoint Site Templates

Ron Charity

(Reprinted WindowsITPro Magazine)

Microsoft SharePoint sites are groups of smart web pages in Windows SharePoint Services (WSS) 3.0 and Microsoft Office SharePoint Server (MOSS) 2007, that let users store, manage, and share data via lists and document libraries. Although SharePoint ships with several site templates that simplify site creation, often administrators want to create a custom template so that users can provision sites that meet their specific needs and become productive faster. There are two ways to customize a template: site templates and site definitions. This overview of site templates and site definitions explains the differences between the two methods and walks you through designing, creating, and customizing SharePoint site templates. This is an extensive subject; I've tried to squeeze a lot of information into this article but have barely scratched the surface. I've included references to examples and further reading that will help you dive deeper into the subject.

Site Templates vs. Definitions

A site template is a package that contains a set of differences from a base site definition. Site templates let you use a browser to easily create a template site from an existing SharePoint site. The site, including pages, lists, libraries, and contents, is archived into a single .stp file.

A site definition is a set of files that define the look and feel of a SharePoint site and are stored on the web front-end (WFE) file system in the 12\TEMPLATE\SiteTemplates folder. Creating or modifying site definitions requires programming skills and is far more complicated than using site templates because when creating a site definition, you need to create XML files that describe the site layout and content and package it in a deployment solution.

At some point, most SharePoint administrators will want to change or add certain functionality to the standard site templates. Let's use the example of a proposal in a document management site that requires certain features. These features aren't included in the standard template. So instead of the user creating a site using the standard document management template and manually customizing the site, he or she can simply use a custom template that has already been created for proposal management.

Site Templates and Site Definitions: Which Should You Use?

So how do you choose between site templates and site definitions when customizing a template? Let's look at some of the pros and cons of each method. The more you understand about the two techniques, the better you'll be able to determine the best method for customizing your sites.

Site Template Pros. Site templates enable administrators to create predefined sites to simplify site provisioning from a user perspective. Sites that are created with them are consistent, which improves usability and maintenance. The pages are stored in just one location—the content database.

Site Template Cons. Because the pages are stored in the content database, performance degradation could occur. Once you create and save a template, you can't update it, which might lead to future product compatibility issues.

Site Definition Pros. Administrators must programmatically create site definitions, giving them more control over site creation. Sites can be provisioned easily; users don't have to customize them. Site definitions provide greater control over a site's look and feel. Because sites are more consistent, there are fewer user support calls for help.

Customizing SharePoint Site Templates

Ron Charity

(Reprinted WindowsITPro Magazine)

Site Definition Cons. Creating site definitions requires programming skills (i.e., Visual Studio). Site definitions are costly to maintain. The definition files are stored on each WFE server; therefore, updates must follow change control management processes when multiple WFE servers are involved.

Designing Sites

Many SharePoint administrators think that to create a site, you simply open a template, create some web parts and any needed columns, set up security, and you're done. But you and your users will likely be disappointed if you don't take the time to design a site that provides the functionality to meet the needs of the users. For example, the first step in the design process is to meet with the users to:

- Understand the tasks or job the site will facilitate and support. This will give you insight regarding functionality (their wish list), and its value as mapped to their jobs (business need).
- Identify the web parts required and whether custom web parts or features are required.
- Understand the lingo they use in the job activities and tasks related to the site. This will help you understand how to label the site, web parts, columns, and views, so that they make sense to the user.

Creating Site Templates

The first step in creating a site template is to create the site that will be a base for your template. Note that site templates are created using the SharePoint standard Site Actions pages. Visual Studio isn't required. This section assumes that you're familiar with adding sites and web parts and adding or modifying columns.

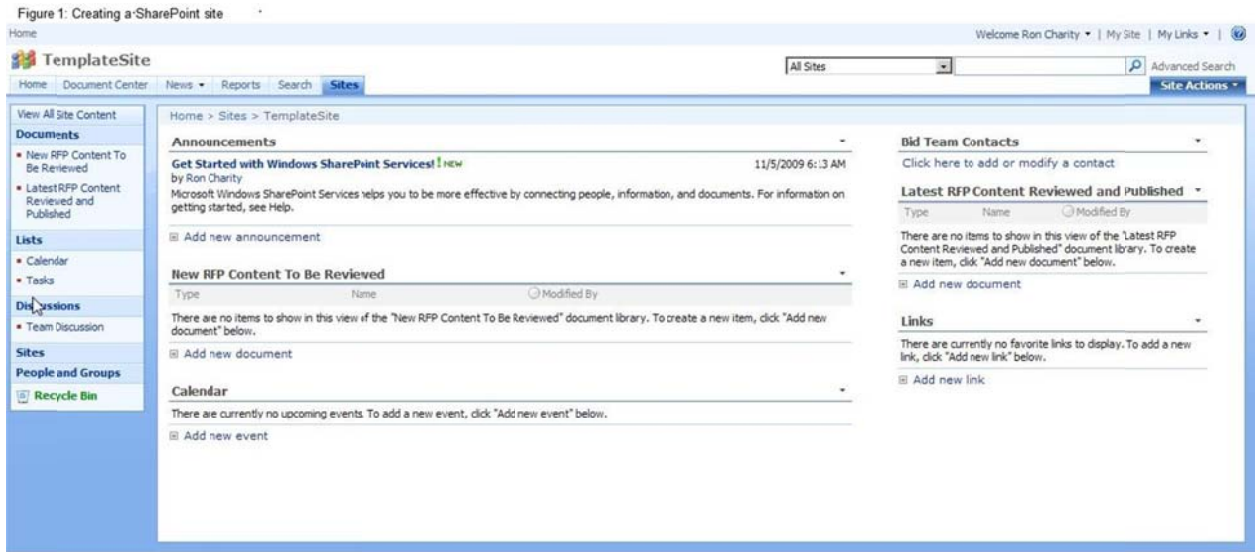
To create your site, open a browser, enter the URL to your SharePoint server, and follow these steps:

1. Select Site Actions, Manage Hierarchy. Right-click the area in the site hierarchy where you want to create a site, and select Add Site.
2. When prompted for a site name, enter TemplateSite, and for the URL, enter TemplateSite, as Figure 1 shows. Select a site template (in our example, Team Site) and click OK. Make note of the URL because it's required for the next step.

Customizing SharePoint Site Templates

Ron Charity

(Reprinted WindowsITPro Magazine)



3. To customize the site, enter the site's URL, press enter, and wait for the site to load. Select Site Action and chose Edit Page. Add the web parts listed in Table 1 to the page. Figure 2 shows the site in its completed form.



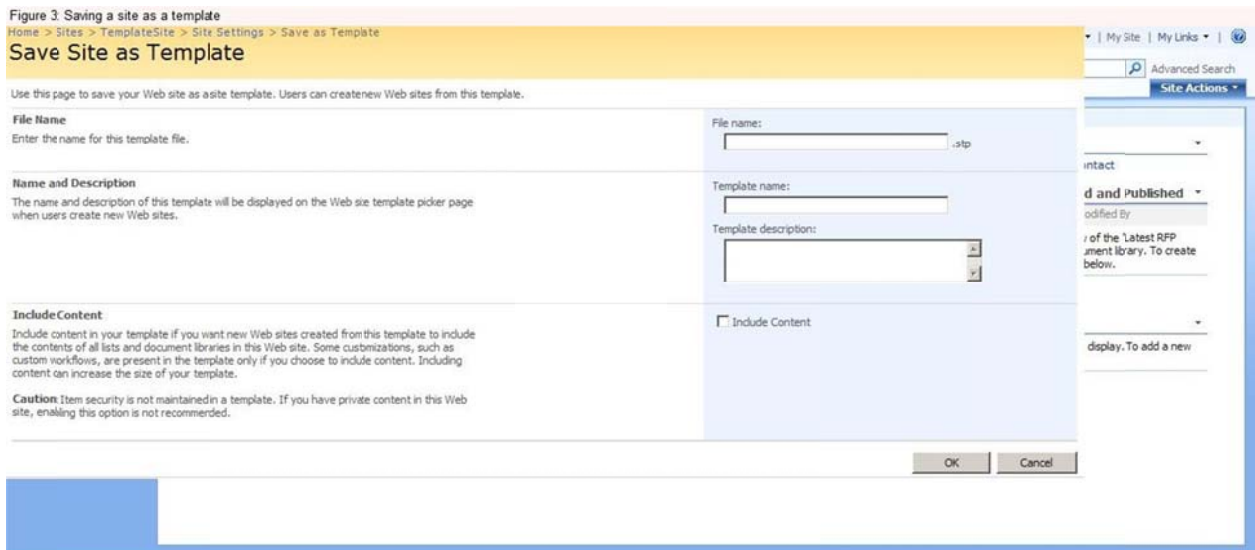
Customizing SharePoint Site Templates

Ron Charity

(Reprinted WindowsITPro Magazine)

Web Part Name	Web Part Type	Location	Custom Columns
New RFP Content To Be Reviewed	Document library	Middle column	RFP Section
Announcements	Announcements	Middle column	
Latest RFP Content Reviewed and Published	Document library	Right column	RFP Section
Bid Team Contacts	Contact list	Right column	Bid Team Role
Links to materials	Links	Right column	
Bid Team Discussions	Discussion library	Middle column	Section of RFP
RFP Questions	Custom list	Middle column	Question Section Author Answered
Calendar	Calendar	Not shown on main page	

- To save the site as a template, enter the URL of your site, press enter, and wait for the site to load. Select Site Action, and chose Site Settings. Under Look and Feel, click Save Site as Template. On the Save Site as Template page, which Figure 3 shows, in the File Name section, enter TemplateSite. Then select the Include Content check box, and click OK. Note the site template name.



Customizing SharePoint Site Templates

Ron Charity

(Reprinted WindowsITPro Magazine)

5. To use the template, enter the URL to the provision site (web page). Select Create a New Site. You're then prompted to enter a Site Name, URL, and Description.
6. At the bottom of the page, you're prompted for the site template. Here you have the option to select one of the Template SharePoint sites. Select the custom site template you created and continue answering the remaining prompts.

Site Definitions

A site definition, as mentioned earlier, is a group of configuration files that provide a framework for creating SharePoint sites with a certain structure and functionality. Each site definition is contained in its own folder in Local_Drive:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\12\TEMPLATE\SiteTemplates. The file that contains the site definition is onet.xml, which is located in a subfolder called \TEMPLATE\SiteTemplates\SMS\XML\ONET.XML. The following sections describe the key configuration files for site definitions and provide examples of how to create a definition and make simple edits to customize it.

Create a Definition

The first step is to create a definition you can work with by copying an existing site definition. To do so, open Windows Explorer and browse to Local_Drive:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\12\TEMPLATE\SiteTemplates. Create a folder called MYCUSTOMSITE. Copy the STS folder contents to the MYTEMPLATESITE.

Webtemp.xml

The Webtemp.xml file determines how site definitions are listed in the different controls that SharePoint uses to create new site collections and subsites (I call it a master list of site templates). This file is located in a language-specific folder in the 12 hive: 12\TEMPLATE\1033\XML\ for English. SharePoint will merge any file that starts with Webtemp into one list of available site definitions.

Each WFE server in a SharePoint deployment (both WSS 3.0 and MOSS 2007) has at least the originally installed Webtemp.xml file located in the Local_Drive:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\12\TEMPLATE\LCID\XML folder, where LCID is the numeric ID of the language/culture (e.g., 1033 for English). There may also be one or more custom Webtemp*.xml files. The Webtemp*.xml files contain the site definition configurations that are available on the Template Selection section of the New SharePoint Site page. Listing 1 shows a sample Webtemp.xml file.

Each site definition is identified by a name and a unique ID. If your ID isn't unique across the farm, you'll receive invalid template errors. I usually work in a range far away from the standard Microsoft IDs (e.g., 2002, 2003). You also need to maintain a central registry list (e.g., in your project SharePoint site) of all the IDs that you have used in your customer project.

I could add a custom site by inserting the following lines of XML within the Template element:

```
<Template  
  Name="MYTEMPLATESITE"  
  ID="2002">
```


Customizing SharePoint Site Templates

Ron Charity

(Reprinted WindowsITPro Magazine)

I could add a link to Google by inserting the following lines of XML code within the NavBar element:

```
<NavBarLink Name="Google"
  Url="http://www.google.com">
</NavBarLink>
```

The ListTemplates section defines all the lists and libraries you can make available within your site definition. Note the ListTemplates section isn't included in the onet.xml file of the standard SharePoint site definition, which instead uses list definition Features and element .xml files to define its list templates. The example that Listing 2 shows is taken from the onet.xml file.

DocumentTemplates

The DocumentTemplates section defines the document templates that are listed on the new page. It also contains the collection of document library templates that are available for creating document libraries. This collection is listed in the Template Type drop-down list box on the New Document Library page.

```
<DocumentTemplates>
  <DocumentTemplate
    Path="STS"
    DisplayName="$Resources:core,doctemp_Word;"
    Type="121"
    Default="TRUE"
    Description="$Resources:core,doctemp_Word_Desc;">
    <DocumentTemplateFiles>
      <DocumentTemplateFile
        Name="doctemp\word\wdtmpl.dotx"
        TargetName="Forms/template.dotx"
        Default="TRUE" />
    </DocumentTemplateFiles>
  </DocumentTemplate>
  ...
</DocumentTemplates>
```

You could edit this <DocumentTemplateFile /> tag to include a custom template such as CustomerOrder.dotx:

```
<DocumentTemplateFile
  Name="doctemp\word\CustomerOrder.dotx"
  TargetName="Forms/CustomerOrder.dotx"
  Default="TRUE" />
```

Note that you must replace the template.dotx in the forms directory with your new template.

BaseTypes

The BaseType element of the global onet.xml file is used during site or list creation to define the five list types on which all list definitions in SharePoint are based. Each list template that is specified in the ListTemplate section is identified with one of the following base types: Generic

Customizing SharePoint Site Templates

Ron Charity

(Reprinted WindowsITPro Magazine)

List, Document Library, Discussion Forum, Vote or Survey, or Issues List. Listing 3 shows the code for the BaseTypes element.

Configurations

Each Configuration element in the Configurations section, which Listing 4 shows, specifies the lists and modules that are created by default when the site definition configuration is instantiated. For instance, one configuration can determine the layout of a site when created as a root of a site collection, and another configuration in the same onet.xml file can provide the contents and page layout of a new site when it's created as a subsite in a site collection. For more information, see the article "Configuration Element (Site)".

Features Overview and Example

With WSS 3.0 and MOSS 2007, Microsoft introduced Features. Essentially, **Features** enable you to add new functionality to SharePoint or make site customizations in an easy and consistent way. The Feature framework enables you to scope features at the farm, web application, site, and web level, depending on the purpose of your feature. A feature is implemented using a feature.xml file. The following is an example of a feature file that I created in Visual Studio that adds a link to the tool bar:

```
<?xml version="1.0 encoding="utf-8" ?>
<Feature xmlns=http://schemas.microsoft.com/sharepoint/"
  Id="Simple Link"
  Title="MySite Link"
  Description="Add link to toolbar menu for MySite"
  Scope="Web"
  Hidden="FALSE"
  AlwaysForceInstall="TRUE">
  <ElementManifests>
    <ElementManifest location="toolbar.xml" />
  </ElementManifests>
</Feature>
```

Open Visual Studio and paste the above code to a file. Save the file to C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\12\TEMPLATE\FEATURES\MySiteToolBar\Feature.xml.

The following code is the toolbar.xml file that adds a link to MySite:

```
<Elements xmlns="http://schemas.microsoft.com/sharepoint/">
  <CustomAction Title="My Site"
    Id="MySite Toolbar "
    Sequence="10"
    RegistrationType="List"
    RegistrationType="101"
    Description="Add a MySite link to the tool bar. " >
    <UrlAction Url="/_layouts/myssite.aspx"/>
  </CustomAction>
</Elements>
```

Customizing SharePoint Site Templates

Ron Charity

(Reprinted WindowsITPro Magazine)

Save the file to C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\12\TEMPLATE\FEATURES\MySiteToolBar\ToolBar.xml. Once you've created the feature, you install it using Stsadm. Here is the syntax.

```
stsadm.exe -o installfeature - filename c:\features\ToolBar\Feature.xml
```

After you complete the installation, you activate the feature by opening your browser and entering the URL for your portal. Click Site Actions, Site Settings, then Under Site Administration, click the Site Features link. Look for your feature and click the Activate button. An alternative approach would be to use Sysadm, with the following syntax:

```
C:\Program Files\Common Files\Microsoft Shared\Web Server  
Extensions\12\bin\stsadm -o activatefeature -filename  
MySiteToolBar\Feature.xml -url http://ServerName/Sitecollection
```

MSDN provides a lab, "How to: Add Actions to the User Interface," that shows how to create a Feature that steps you through adding functionality to the menus of the UI in WSS. The example shows how to add actions to various menus by using a Feature and how to activate the actions within the deployment. I suggest you do this lab and create your Site Template to get a sense of the Feature framework and its power. View the lab here. Two other articles, "Features for SharePoint," and "Working with Features," provide further information about using features.

It's All about Usability

Site templates and site definitions can help your organization boost SharePoint usability by creating sites with consistent look and feel, which helps reduce training and support costs. They also provide an approach for designing site templates that support business requirements and will help reduce the time required to deploy sites. Using the Feature framework, your organization can deploy features to sites globally or to specific sites to add functionality and improve user experience.

Listing 1: Sample Webtemp.xml file

```
<Templates  
  xmlns:ows="Microsoft SharePoint">  
  <Template  
    Name="STS"  
    ID="1">  
    <Configuration  
      ID="0"  
      Title="Team Site"  
      Hidden="FALSE"  
      ImageUrl="/_layouts/images/stsprev.png"  
      Description="A site for teams to quickly organize, author, and share  
information. It provides a document library, and lists for managing  
announcements, calendar items, tasks, and discussions."  
      DisplayCategory="Collaboration" />  
    <Configuration  
      ID="2"  
      Title="Document Workspace"
```

Customizing SharePoint Site Templates

Ron Charity

(Reprinted WindowsITPro Magazine)

```
Hidden="FALSE"
ImageUrl="/_layouts/images/dwsprev.png"
Description="A site for colleagues to work together on a document. It
provides a document library for storing the primary document and supporting
files, a tasks list for assigning to-do items, and a links list for resources
related to the document."
DisplayCategory="Collaboration" />
</Template>
<Template
Name="MPS"
ID="2">
<Configuration
ID="0"
Title="Basic Meeting Workspace"
Hidden="FALSE"
ImageUrl="/_layouts/images/mwsprev.png"
Description="All the basics to plan, organize and track your
meeting. This Meeting Workspace contains the following lists:
Objectives, Attendees, Agenda, and Document Library."
DisplayCategory="Meetings" />
</Template>
</Templates>
```

Listing 2: The ListTemplates section from the onet.xml file

```
<ListTemplates>
<ListTemplate
Name="meetings"
DisplayName="$Resources:xml_onet_mwsidmeetingDisp;"
Type="200"
BaseType="0"
Unique="TRUE"
Hidden="TRUE"
HiddenList="TRUE"
DontSaveInTemplate="TRUE"
SecurityBits="11"
Description="$Resources:xml_onet_mwsidmeetingDesc;"
Image="/_layouts/images/itevent.gif">
</ListTemplate>
<ListTemplate
Name="agenda"
DisplayName="$Resources:xml_onet_mwsidagendaDisp;"
Type="201"
BaseType="0"
FolderCreation="FALSE"
DisallowContentTypes="TRUE"
SecurityBits="11"
Description="$Resources:xml_onet_mwsidagendaDesc"
Image="/_layouts/images/itagnda.gif">
</ListTemplate>
...
</ListTemplates>
```

Customizing SharePoint Site Templates

Ron Charity

(Reprinted WindowsITPro Magazine)

Listing 3: The BaseType section of onet.xml

```
<BaseTypes>
  <BaseType
    Title="Generic List"
    Image="/_layouts/images/itgen.gif"
    Type="0">
    <MetaData>
      <Fields>
        <Field
          ID="{1d22ea11-1e32-424e-89ab-9fedbadb6ce1}"
          ColName="tp_ID"
          RowOrdinal="0"
          ReadOnly="TRUE"
          Type="Counter"
          Name="ID"
          PrimaryKey="TRUE"
          DisplayName="$Resources:core,ID"
          SourceID="http://schemas.microsoft.com/sharepoint/v3"
          StaticName="ID">
        </Field>
        <Field
          ID="{03e45e84-1992-4d42-9116-26f756012634}"
          RowOrdinal="0"
          Type="ContentTypeId"
          Sealed="TRUE"
          ReadOnly="TRUE"
          Hidden="TRUE"
          DisplayName="$Resources:core,Content_Type_ID;"
          Name="ContentTypeId"
          DisplaceOnUpgrade="TRUE"
          SourceID="http://schemas.microsoft.com/sharepoint/v3"
          StaticName="ContentTypeId"
          ColName="tp_ContentTypeId">
        </Field>
        ...
      </Fields>
    </MetaData>
  </BaseType>
  ...
</BaseTypes>
```

Listing 4: The Configurations section of onet.xml

```
<Configurations>
  ...
  <Configuration
    ID="0"
    Name="Default">
    <Lists>
      <List
        FeatureId="00BFEA71-E717-4E80-AA17-D0C71B360101"
```

Customizing SharePoint Site Templates

Ron Charity

(Reprinted WindowsITPro Magazine)

```
Type="101"
Title="$Resources:core,shareddocuments_Title;"
Url="$Resources:core,shareddocuments_Folder;"
QuickLaunchUrl="$Resources:core,shareddocuments_Folder;/Forms/AllItems.aspx" />
...
</Lists>
<Modules>
  <Module
    Name="Default" />
</Modules>
<SiteFeatures>
  <Feature
    ID="00BFEA71-1C5E-4A24-B310-BA51C3EB7A57" />
  <Feature
    ID="FDE5D850-671E-4143-950A-87B473922DC7" />
</SiteFeatures>
<WebFeatures>
  <Feature
    ID="00BFEA71-4EA5-48D4-A4AD-7EA5C011ABE5" />
  <Feature
    ID="F41CC668-37E5-4743-B4A8-74D1DB3FD8A4" />
</WebFeatures>
</Configuration>
...
</Configurations>
```