

Custom Templates

Custom templates enable a user to define a default layout, functionality, and default content for a newly created site. A parallel can be drawn between a Microsoft Word stationery template and a SharePoint template. For instance, you might have a stationary template for Microsoft Word with your company logo, company colors, office address, and a myriad of other customizations. You can construct a SharePoint template with the same company logo, company colors, and many other customizations. The parallels continue in the application of SharePoint templates as well. Just as you can apply your Microsoft Word template to a new document, you can apply your SharePoint template to a new site.

Two types of custom templates exist: custom *list* templates and custom *site* templates. Custom list templates contain schema and content data for all types of lists. Thus, custom list templates apply to SharePoint lists. They therefore apply to document libraries, picture libraries, links, announcements, custom lists, and many others. Custom site templates apply to SharePoint sites, which could be top-level sites or subsites. Custom site templates contain schema and content data for everything that can be defined on a site, including lists, themes, web part content and layout, as well as many other customizations.

One of the most attractive features of custom templates is the existence of a graphical interface to construct them. This interface exists in the form of the SharePoint web interface as well as Microsoft Front-Page. The simplicity of this interface empowers you to quickly create a default look for your sites and lists that can be applied even more quickly to new sites. This feature avoids the tedious chore of manually performing each customization upon creating a new site. Instead, you click and drag your way to all these customizations.

Understanding Custom Templates

Custom site templates and site definition templates (see Chapter 2, “Site Definitions”) appear to the user at site creation as a unified list of choices as shown in Figure 1.1. Custom list and site templates represent a delta or additional customizations to site definition templates. This delta acts as an addition to the base layer to form a two-layer structure. Consequently, custom site and list templates require more server processing than site definition templates.

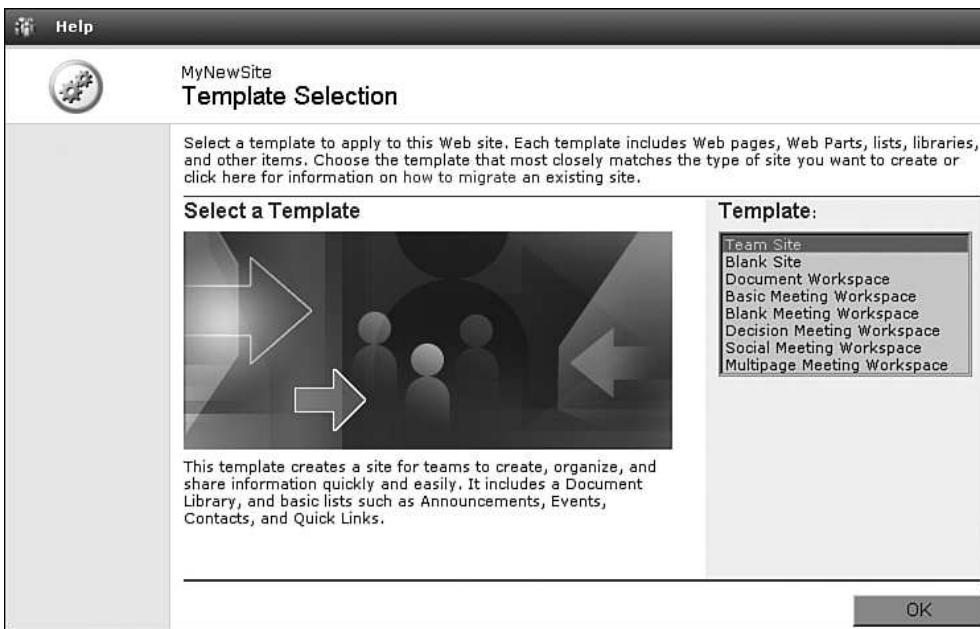


Figure 1.1 Custom template and site definition template selection.

When a user designs a SharePoint list or site, those settings are captured in the database. For instance, a user’s customization of a particular site can include applying a site theme, placing web parts, and creating additional list views. All these customizations are stored in the database. The same parallel can be drawn for lists. List customizations—including additional views and stock content—are stored in the database.

SharePoint provides a mechanism to harvest all these customizations to a list or site from the database. That mechanism is a custom template. A custom template is a compilation of customization settings in a single, reusable package. This package defines the delta between the site definition and our site or list customizations.

Custom site and list templates can be created on one server farm and copied to another. A key caveat to successfully rendering the copied template is the existence of the site definition on which the custom template is based. In fact, web parts or any other resources that are referenced in the template must also exist on the new server. If the site definition or other resources that the custom template expects do not exist, the custom template (delta) will have nothing on which to base its changes and will not render. If the site definition is different between server farms, the custom template can render in an unexpected fashion.

Using custom list or site templates is essentially a three-step process. The first step is to customize a list or site as desired. The second step is to save the template to the template gallery. The third and final step is to create a site or list with that template.

It should be noted that there are a few caveats to using custom templates:

- A 10-megabyte limit exists on the total size of a site template.
- Custom templates do not retain the security settings of the site or list from which they were copied.
- Child sites cannot be retained in site templates. You must create any desired child site anew after site creation.

Exploring Custom Site Templates

For the purposes of our conversation, we need a customized site. This site will be customized to serve as the basis for our custom template. We therefore have created a new site based on the *Blank Site* template and customized it through the SharePoint web user interface and Front-Page. The result is shown in Figure 1.2.

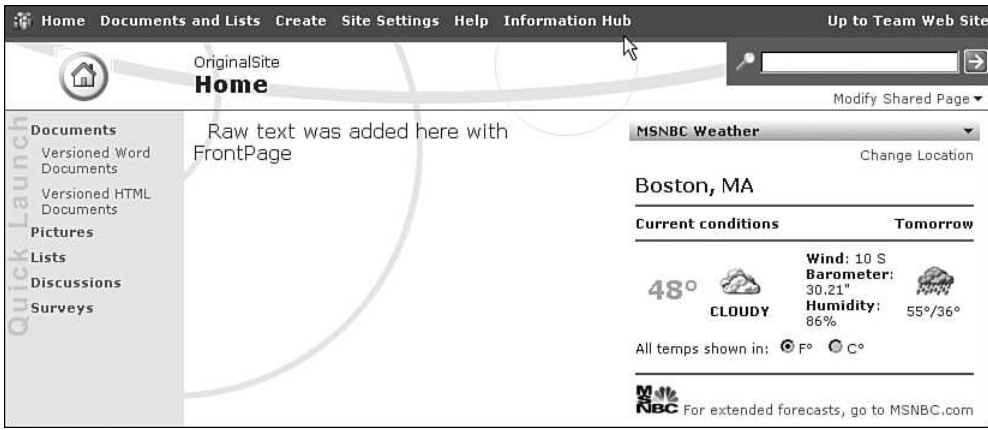


Figure 1.2 Customized site on which we want to base our custom site template.

If you are new to SharePoint and are unfamiliar with how to customize a site, either through the SharePoint web interface or through FrontPage, we suggest you refer to one of the many books or articles on the subject because the emphasis in this chapter and this book is on some of the more advanced concepts of custom templates.

Saving a Custom Site Template

The site template is saved through the SharePoint web user interface, which empowers users and avoids the need for administrative intervention. Now that we have a customized site, we are ready to save it as a custom site template for others to reuse. The steps to save the custom site template are enumerated next.

1. Navigate to Site Settings in the newly customized site.
2. Select Site Administration under the Administration section.
3. Select Save Site As Template, which is in the Management and Statistics section (see Figure 1.3).

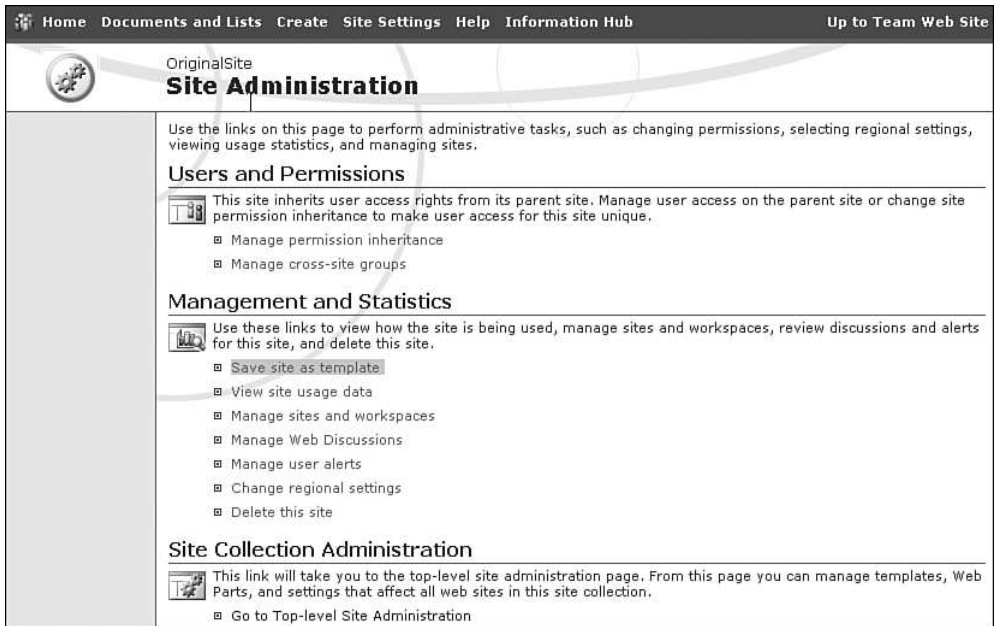


Figure 1.3 Site Administration page with Save Site As Template highlighted.

4. Fill out the Save Site as Template page (see Figure 1.4) as appropriate and then click OK. You can always change the filename, template title, and template description later in the template library. At this time you must decide whether to include the list and library content.

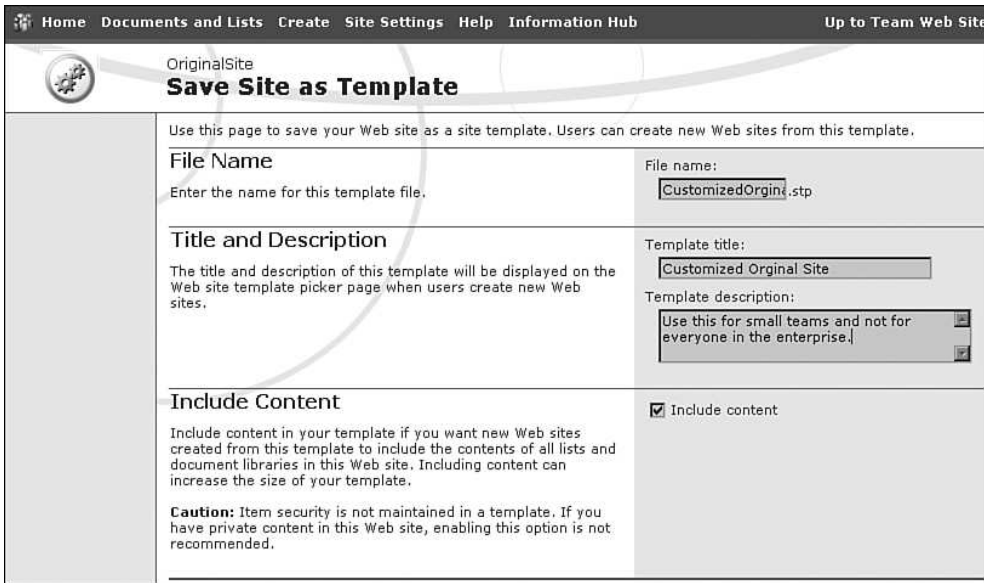


Figure 1.4 Saving a site template.

After completing these steps, the template is now saved to the site template gallery. SharePoint confirms that the operation completed successfully as shown in Figure 1.5, which displays immediately after clicking OK in the Save Site as Template page (refer to Figure 1.4).

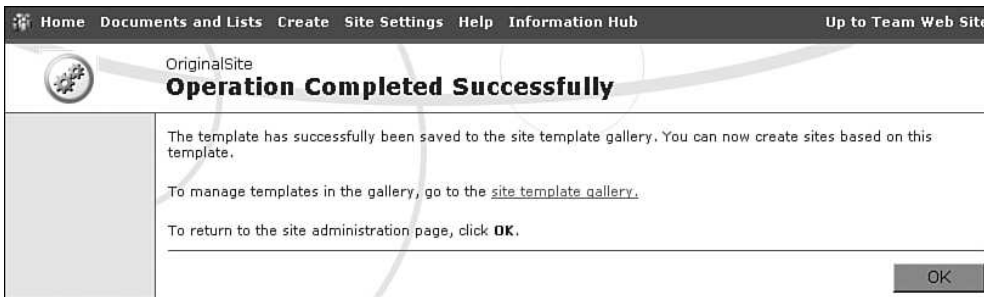


Figure 1.5 Successfully saving a site template.

You manage the Site Template Gallery through the *Top-level Site Administration* (see Figure 1.6) page. Because the Site Template Gallery exists at the top-level site of a site collection, the newly created

custom template is accessible only from that site collection. This important caveat enables you to isolate custom site templates from other site collections. If you want a site template to be accessible within another site collection, you simply need to upload the desired custom site template to the desired site collection's site template gallery.

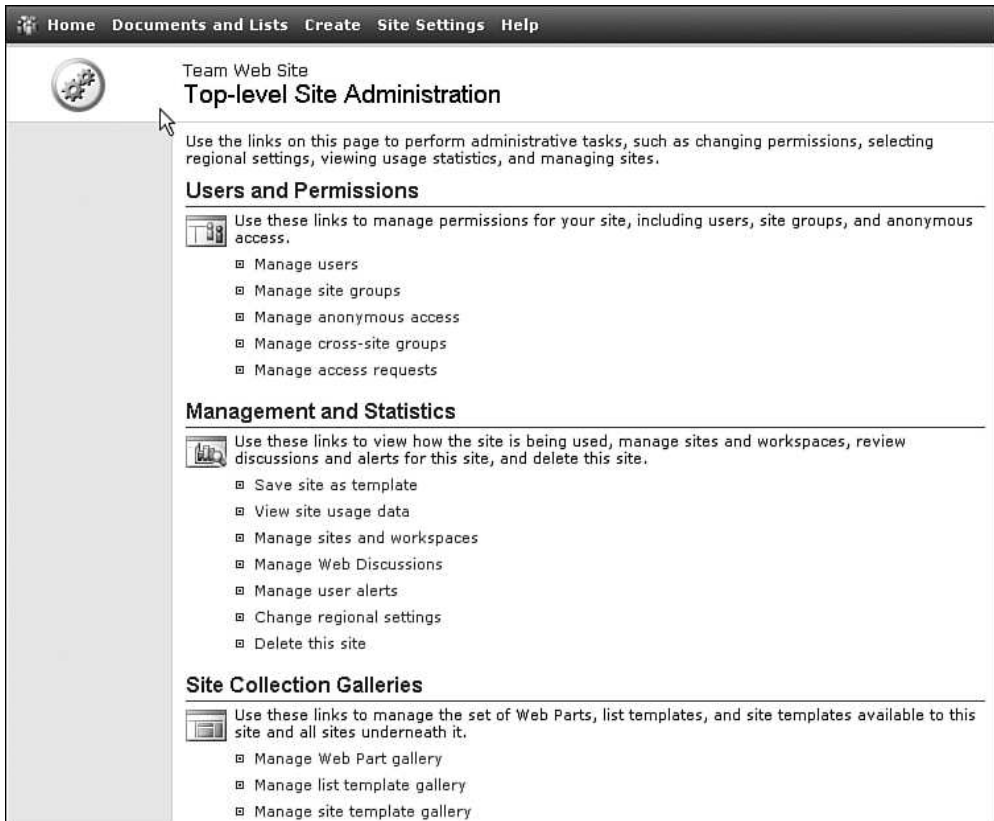


Figure 1.6 Top-level Site Administration.

The *Site Template Gallery* (see Figure 1.7) is simply a customized document library for an entire site collection, which explains why our newly saved template (file) can only be used in the current site collection and not other site collections. It also implies that the template is wholly contained within a single file.

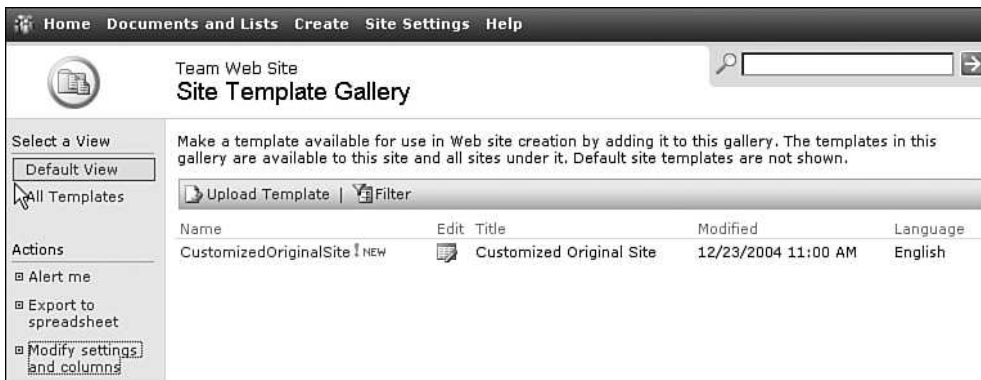


Figure 1.7 Site Template Gallery.

Through the Site Template Gallery, the custom site template filename, title, and description are all editable. Thus, if you are unhappy with the values you specified for them in Figure 1.4, you can change them.

Applying the Custom Site Template

The next time you create a site, your custom site template appears in the Template Selection web page (see Figure 1.1 and Figure 1.8). In fact, all custom site templates for the current site collection appear in the Template Selection dialog. Notice that the title and the description that we specified while saving the custom template are reflected in the Template Selection web page.

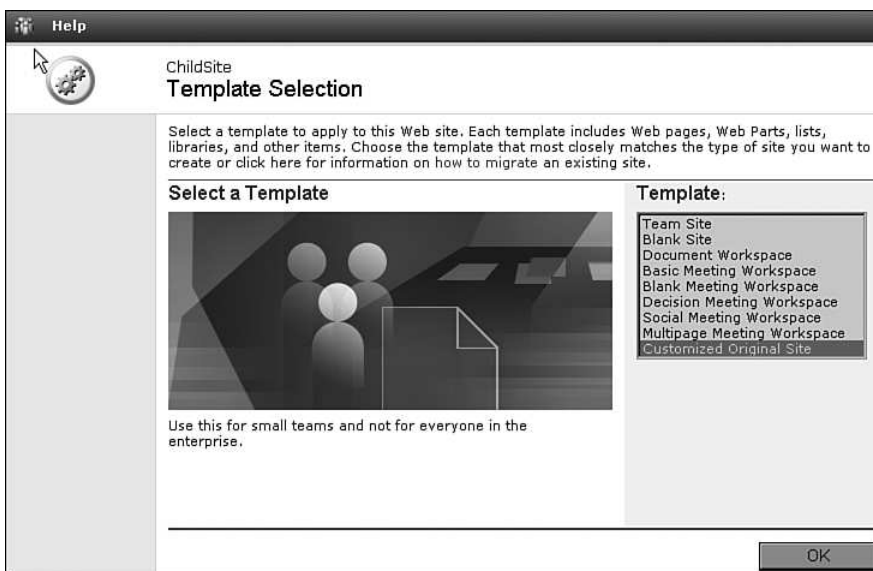


Figure 1.8 Template Selection that includes our newly created Customized Original Site.

After we apply the template to a new site, it looks exactly like the original site from which the template was saved (see Figure 1.9).

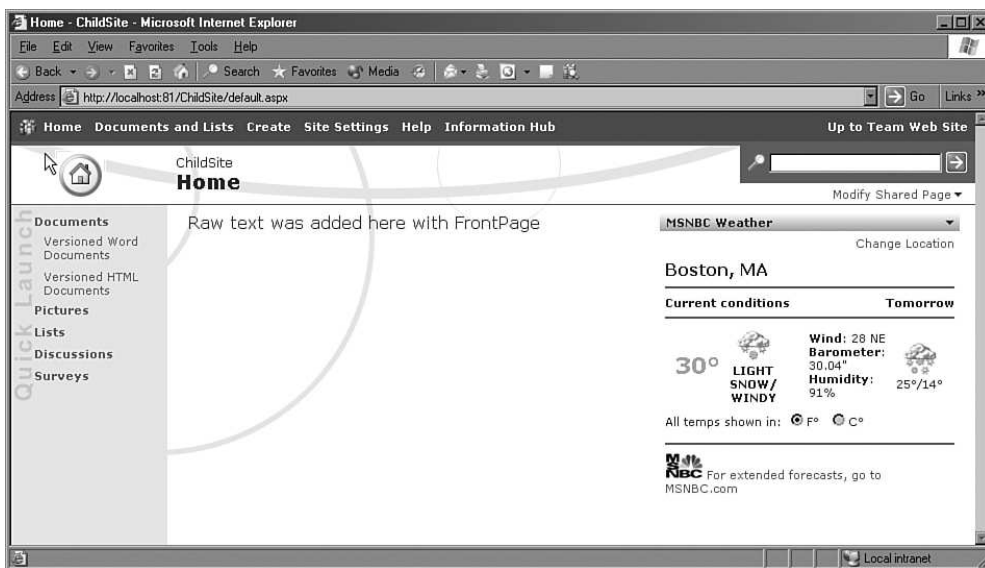


Figure 1.9 ChildSite with Customized Original Site template applied.

The Site Template File

As previously demonstrated, the site template was saved to the Site Template Gallery (document library) as a file. SharePoint saves it to the Site Template Gallery as a file with an STP file extension. This STP file is actually a CAB (Cabinet) formatted file. A CAB file is a compressed collection of files and therefore is similar to a ZIP file. In the case of the custom site template, this collection of files represents the difference between the customization made to the site and the base site definition.

The site template can be copied to a user's local disk through the normal methods afforded by a SharePoint document library. This capability facilitates uploading the site template to another Site Template Gallery and therefore reusing it in another site collection. It also facilitates the extraction and reuse of the custom template's code.

Both the site definition and custom template make extensive use of Collaborative Application Markup Language (CAML). CAML is an XML-based language that SharePoint uses. As such, coding site definitions requires the manual manipulation of several XML files, which leverage CAML. It is sometimes easier to design our site through the graphical user interfaces of SharePoint and FrontPage—as we just did—and then extract those CAML changes. These changes could then be injected into the site definition, which would save a significant amount of manual coding time.

Some of the files contained within the compressed CAB (STP) file are shown in Figure 1.10. Perhaps the most significant file and the only consistently named file within the custom site template is MANIFEST.XML.

11000000.000	15 KB	000 File	5/1/2004 6:00 AM	A
k0000000.000	29 KB	000 File	5/1/2004 6:00 AM	A
k1000000.000	1 KB	000 File	5/1/2004 6:00 AM	A
l0000000.000	8 KB	000 File	5/1/2004 6:00 AM	A
l1000000.000	1 KB	000 File	5/1/2004 6:00 AM	A
m0000000.000	1 KB	000 File	5/1/2004 6:00 AM	A
manifest.xml	263 KB	XML Document	5/1/2004 6:00 AM	A
r0000000.000	1 KB	000 File	5/1/2004 6:00 AM	A
s0000000.000	1 KB	000 File	5/1/2004 6:00 AM	A
p0000000.000	1 KB	000 File	5/1/2004 6:00 AM	A
q0000000.000	32 KB	000 File	5/1/2004 6:00 AM	A
r0000000.000	1 KB	000 File	5/1/2004 6:00 AM	A
s0000000.000	1 KB	000 File	5/1/2004 6:00 AM	A
t0000000.000	1 KB	000 File	5/1/2004 6:00 AM	A
u0000000.000	1 KB	000 File	5/1/2004 6:00 AM	A
v0000000.000	1 KB	000 File	5/1/2004 6:00 AM	A

Figure 1.10 STP contents.

The manifest (MANIFEST.XML) is composed of six sections as shown in Listing 1.1. The `MetaInfo` section contains meta data for the site, the `Details` section describes high-level site parameters, the `Structure` section specifies the top navigation and quick launch bar, the `Files` section maps the other files in the CAB as well as virtual files and folders to the site, the `UserLists` section defines the various document or picture libraries and lists, and finally the `WebParts` section defines the placement and settings of web parts throughout the site's web pages.

Listing 1.1 MANIFEST.XML Structure

```
<Web>
  <MetaInfo />
  <Details />
  <Structure />
  <Files />
  <UserLists />
  <WebParts />
</Web>
```

The `Details` section is where the custom template specifies the Site Definition Template on which it is based. This can be seen in Listing 1.2 through the `TemplateID` and `Configuration` elements. We talk more about the Template ID and the Configuration ID in the `WEBTEMP.XML` section in a later chapter.

Listing 1.2 MANIFEST.XML Details Example

```
<Details>
  <TemplateDescription>Use this for small teams and not for
    everyone in the enterprise.</TemplateDescription>
  <TemplateTitle>Customized Original Site</TemplateTitle>
  <Language>1033</Language>
  <TemplateID>1</TemplateID>
  <Configuration>1</Configuration>
  <Title>OriginalSite</Title>
  <Description />
```

```
<CalendarType>1</CalendarType>
<AlternateCSS />
<CustomJSUrl />
<AlternateHeader />
<Subweb>1</Subweb>
<Locale>1033</Locale>
<Collation>25</Collation>
<TimeZone>10</TimeZone>
</Details>
```

Central Template Gallery: Custom Site Templates beyond the Site Collection

The Site Template Gallery is site collection-specific. However, you can make a custom site template universally available to all site collections. SharePoint has a central template gallery that is shared for your entire server farm—which could consist of only one server. Unfortunately, SharePoint only exposes access to it through the `stsadm.exe` utility and does not provide a set of web pages to manage it as it does with the site template gallery.

However, using the StsAdm Windows GUI (www.microsoft.com/sharepoint/downloads/components/detail.asp?a1=443) does make this a bit more palatable. If you were previously unfamiliar with it, it is simply a wrapper for the `stsadm.exe` command line program.

Table 1.1 highlights the central template gallery operations. Most of the operations are straightforward. However, `CreateSite` needs some explanation.

Table 1.1 StsAdm.exe Central Template Gallery Operations

Operation	Description
<code>AddTemplate</code>	Adds a template.
<code>CreateSite</code>	Creates a site. You can optionally apply a template from the central template gallery.
<code>DeleteTemplate</code>	Deletes a template.
<code>EnumTemplates</code>	Lists the templates.

The `SiteTemplate` suboption of `CreateSite` requires that you look up the name of the site template through the `EnumTemplates` operation first. The way to list the templates in the custom template gallery is shown in Listing 1.3.

Listing 1.3 STSADM.EXE Syntax to Enumerate the Global Templates

```
C:>stsadm.exe -o EnumTemplates
```

```
Customized Original Site - Language: 1033 - Site Template:  
_GLOBAL_#1 - Template Id: 1 Use this for small teams and  
not for everyone in the enterprise.
```

Enumerating the templates is the only way to determine the dynamically generated name of custom templates added to the central template gallery. In Listing 1.3, we can see that the Central Template Gallery contains only one template, which has been named `GLOBAL_#1`. This name can be used with the optional parameter `SiteTemplate` in the `CreateSite` operation. The syntax for the `CreateSite` operation is shown in Listing 1.4.

Listing 1.4 STSADM.EXE Syntax to Create a Site with a Global Template

```
stsadm.exe -o createsite  
-url http://moose/sites/sitecollection2  
-ownerlogin InformationHub\JNadrowski  
-owneremail Jason@InformationHub.com  
-sitetemplate _GLOBAL_#1
```

Exploring Custom List Templates

Custom list templates are very similar to custom site templates except that they are narrower in scope. With list templates, you can create your own custom list types to complement the existing list types such as document libraries, announcements, and links.

NOTE: Because a significant functionality overlap exists between custom site templates and custom list templates, we will try to minimize redundant explanations. As such, we will only cover the saving of a custom list template and the subsequent creation of a list based on that newly created custom list template. We will not detail the internals of the list template file. Instead, we refer you to the earlier section, “The Site Template File.”

As we did in our explanation of custom site templates, we will create a customized list. In the following scenario we customize a document library.

Saving a Custom List Template

The list template is saved through the SharePoint web user interface, which empowers users and avoids the need for administrative intervention. Now that we have a customized list, we are ready to save it as a custom list template for others to reuse. The following steps demonstrate how to save the custom list template.

1. Select Save Document Library As Template, as shown in Figure 1.11.

Home Documents and Lists Create Site Settings Help

Team Web Site
Customize Shared Documents

Use this page to change the design of the document library, such as its name, security settings, and columns. You can also create or change views of the document library.

Go Back to "Shared Documents"

General Settings

General settings of this document library include its name, description, and security. Current general settings of this document library:

Title:	Shared Documents
Web Address:	http://localhost:81/Shared Documents/Forms/AllItems.aspx
Document Template:	http://localhost:81/Shared Documents/Forms/template.doc (Edit Template)
Description:	Share a document with the team by adding it to this document library.
On Quick Launch Bar:	Yes
Generate Versions:	Yes
Content Approval Required:	No

- Change general settings
- Save document library as template
- Change permissions for this document library
- Delete this document library

Columns

A column stores information about each item in the document library. Columns currently in this document library:

Column (click to edit)	Type	Required
Title	Single line of text	
Document Type	Choice	

- Add a new column
- Change the order of the fields

Figure 1.11 List administration.

- Fill out the Save As Template page (see Figure 1.12) as appropriate and then click OK. You can always change the filename, template title, and template description later in the template library. At this time you must decide whether to include the list content.



The screenshot shows a SharePoint web site interface. At the top, there is a navigation bar with links: Home, Documents and Lists, Create, Site Settings, and Help. Below the navigation bar, the page title is "Team Web Site" and the main heading is "Save as Template: Shared Documents".

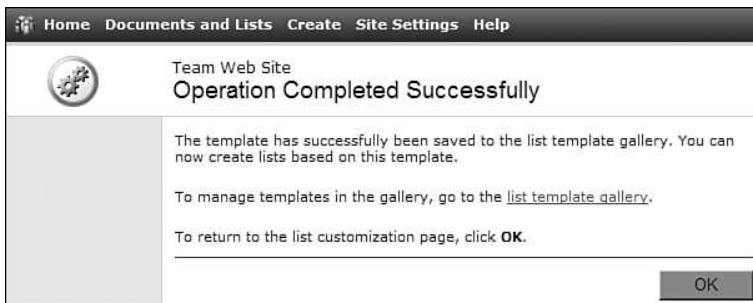
The page content is organized into three main sections:

- File Name:** A text box labeled "File name:" contains the text "CustomDocLib .stp".
- Title and Description:** A text box labeled "Template title:" contains the text "Customized Document Library". Below it, a text box labeled "Template description:" contains the text "Document library plus Versioning and additional column.".
- Include Content:** A checkbox labeled "Include content" is currently unchecked.

Below the "Include Content" section, there is a **Caution:** "Item security is not maintained in a template. If you have private content in this document library, enabling this option is not recommended."

Figure 1.12 Saving a custom list template.

After completing the previous steps, the template is now saved. SharePoint confirms that the operation completed successfully as shown in Figure 1.13, which immediately displays after clicking OK in the Save As Template page.



The screenshot shows a SharePoint web site interface. At the top, there is a navigation bar with links: Home, Documents and Lists, Create, Site Settings, and Help. Below the navigation bar, the page title is "Team Web Site" and the main heading is "Operation Completed Successfully".

The page content consists of a single message box:

The template has successfully been saved to the list template gallery. You can now create lists based on this template.

To manage templates in the gallery, go to the [list template gallery](#).

To return to the list customization page, click **OK**.

At the bottom right of the message box, there is an "OK" button.

Figure 1.13 Successfully saving a list template.

You manage the List Template Gallery through the *Top-level Site Administration* (refer to Figure 1.6) page just as you manage the other site collection galleries. Because the List Template Gallery exists at the top-level site, the newly created custom template is only accessible from that site collection. This important caveat enables you to isolate custom list templates from other site collections. If you want to make a list template accessible within another site collection, you simply need to upload the desired custom list template to the desired site collection's list template gallery.

The List Template Gallery (see Figure 1.14) is also a customized document library for an entire site collection, which explains why our newly saved template (file) can only be used in the current site collection and not other site collections. It also implies that the template is wholly contained within a single file.

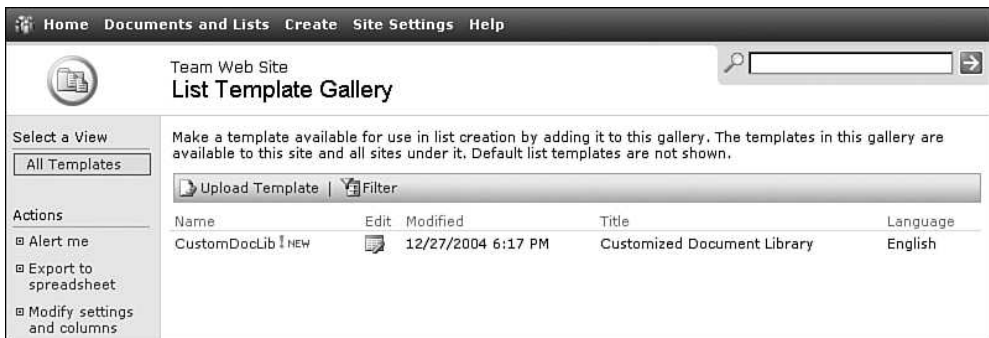


Figure 1.14 List Template Gallery.

Through the List Template Gallery, the custom list template filename, title, and description are all editable. Thus, if you are unhappy with the values you specified for them in Figure 1.12, you can change them.

Applying the Custom List Template

The next time you create a list or library, your new list template will appear on the CREATE.ASPX page. In our example, Customized Document Library appears as a choice (see Figure 1.15). In fact, all custom

list templates for the current site collection will also appear on that page. Notice that the title and the description that we specified while saving the custom template are reflected on the CREATE.ASPX web page.

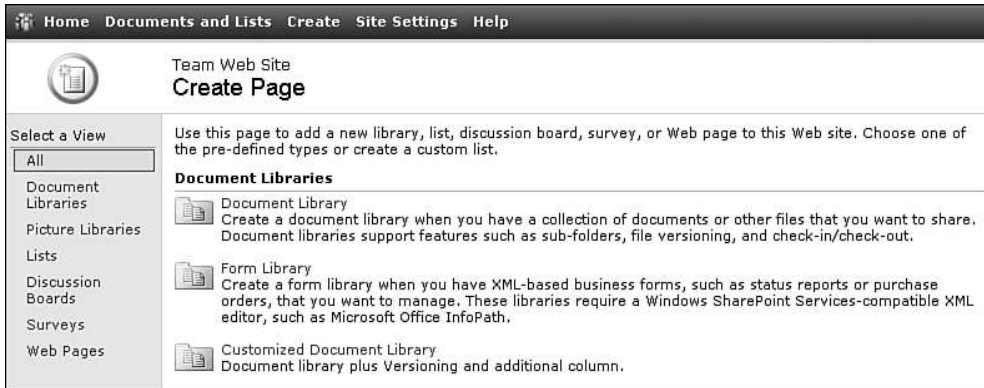


Figure 1.15 Create Page (CREATE.ASPX).

Ghosted and Unghosted Pages

If you explore a SharePoint web site (virtual server) through the IIS MMC, you notice that neither the team sites nor portal areas are visible. However, when you navigate your browser to a team site web page or portal web page, it is rendered. That page is rendered because SharePoint answers the request before IIS is given a chance to look for a physical file mapped to the requested URL.

When SharePoint answers the request for a URL, it looks up the page (URL) in the database. It first examines the Sites table in the configuration database to determine the site collection and consequently the content database to which the page belongs. Next, SharePoint queries the Docs table in the specified content database.

If the returned row from the Docs table has a null Content column, the file is rendered by the file specified in the SetupPath column. This path points to a physical file on the SharePoint front end web server's hard disk. This page is therefore considered a ghosted page because its entry in the Docs table is a pointer or a ghost of a physical file. Because

this file exists on the server's hard disk, the ASP.NET parser is used to render the page.

In contrast, if the returned row from the `Docs` table has a non-null `Content` column and therefore a null `SetupPath`, the page is rendered from the data in the `Content` column. This page is considered unghosted because it is the reverse of a ghosted page. It does not contain a pointer to a physical file. Because there is only a virtual file to parse and no physical file, the ASP.NET parser cannot be used. SharePoint uses its own `SafeMode` parser.

The benefits and drawbacks to ghosted and unghosted pages can be boiled down to the differences between the ASP.NET and SharePoint `SafeMode` parsers. The ASP.NET parser compiles a page into an assembly the first time it is rendered. On subsequent page executions, the compilation step is skipped because an assembly exists. This results in faster page execution. In contrast, the `SafeMode` parser does not compile a page into an assembly. It always parses the page anew to form an object tree structure and then executes against that structure. In this regard it is similar to how ASP (the predecessor to ASP.NET) works. The `SafeMode` parser not only works differently from the ASP.NET parser but also operates under different rules. These rules effectively make unghosted pages execute slower than comparable ghosted pages. The `SafeMode` parser works differently from the ASP.NET parser in the following ways:

1. The page will not be compiled. Therefore, all compile directives will be ignored.
2. Inline code cannot be run. It will cause an error, and the page will not execute. The only dynamic code allowed is server-side controls that are marked safe (`SafeControls`, trusted controls, etc.).
3. The page cannot be executed on a single threaded apartment (STA) thread. Therefore, the `Page` directive's `AspCompat` attribute cannot be used.
4. Session state is either on for all pages or off for all pages. The `Page` directive's `EnableSessionState` cannot be used to selectively turn on session state for some pages, while disabling it for others.

Now that we understand the functional differences between the parsers, the question becomes what actions make a page ghosted or unghosted? In answering that question, we must talk a little about site definitions (see Chapter 2 for more details). Site definitions consist of physical files that reside on the front end web server's hard disk. When creating a new site, the user can choose a template from one of the front end web server's site definitions. Upon choosing one of these site definition templates, the ASPX pages of the selected template are ghosted and therefore mapped to the physical files on the server's hard disk. Pages become unghosted if the file has been updated using FrontPage 2003 or web folders or if the document library fields have been modified. In addition, all files uploaded to SharePoint are by definition unghosted because there is no physical file on the server's hard disk to reference.

When you understand what types of pages are unghosted (uploaded pages, pages modified with FrontPage, etc.), it should make more sense why the SafeMode parser is more restrictive and therefore slower than the ASP.NET parser. Among other things, the SafeMode parser restricts the user's ability to run malicious server-side code on the front end web server.

It is worth noting that custom site templates are based on site definitions. Therefore, a custom site template duplicates the same ghosting and unghosting found in the site from which it was copied. Thus, all new sites created with this custom site template have the same ghosted and unghosted pages as the original—at the time the template was created. Of course changes to the original site do not cascade to sites using this template.

However, ghosted pages reflect updates made to the physical file to which they are mapped. You could therefore make small formatting changes to these web pages and cascade the changes to sites using them. You would need to be careful not to remove certain resources—such as web part zones—from the page. Modifying or removing resources can result in a loss of functionality. Cascading changes to unghosted pages is generally not practical and almost always a nightmare.

Summary

Custom site and list templates are an easy way for non-technical users to create a reusable site or list definition. They are created by making a copy of a customized list or site. After being created, they seamlessly appear within the SharePoint web user interface and appear as built-in list types or site types. In addition, custom templates have the following properties:

- The templates represent a delta between a site definition (see Chapter 2) and a customized list or site. As such, they put an additional burden on the server.
- Custom site templates can be migrated from one site collection to another or globally installed for all site collections in a farm.
- Custom templates contain no security settings from the list or site from which they were copied.
- A 10-megabyte limit exists on the total size of a template.
- The site template exists as a single file stored in a cabinet (CAB) format within a template library.
- The code within a site template can be extracted and reused in site definitions.
- Ghosted web pages are interpreted with the ASP.NET parser. They exist in the SharePoint database as a pointer to a physical file on the front end web server. They are compiled into an assembly the first time they are executed and run from that compiled assembly thereafter.
- Unghosted web pages are interpreted by the SharePoint SafeMode parser. Their executable code exists entirely within the database and not on the physical disk. They are not compiled. They run slower than a comparable ghosted page.