10
Building Reusable Components

Getting Started

At this point in the book, you not only know how to design form templates that look nice but you can also take advantage of most of the basic (and some not so basic) features available when using InfoPath in design mode. Now you can create professional-looking forms in no time. However, over time, as you create more and more form templates, you may find that many of them are very similar or share a lot of the same constructs.

For example, several of your form templates may contain sections used to collect address information from the user. It would be pretty tedious to have to add the same address block to each of your form templates, especially if there is no difference in this data from one form template to another. You could simply copy the controls and layout from the view and paste them into another form template. This would work fine for the view aspects of the form. However, what about the data structure, rules, calculated default values, data validation, and data connection information that you may have added to your form template? How do you copy these settings from one form template to another? Using copy and paste, this just isn't possible.

There is another way to share different parts of a form template in multiple form templates. With InfoPath 2007, you can build custom components that can be reused in one or more of your form templates. This means that you can create your own components or reuse ones created by third-party software providers.

You can create reusable components in two ways. The first is to write ActiveX controls in C++, Visual Basic, C#, or any COM-aware programming language. Creating controls this way requires you to put on your developer hat and write quite a bit of code. Since this is an advanced topic, we'll talk about this in the advanced section of this book in Chapter 19.

Fortunately, there is a second way to create custom components in InfoPath, one that doesn't require you to write even one line of code. These custom components are called **template parts**. As their name implies, template parts are parts of a form template. More specifically, template parts group pieces of a form template into a component that can be reused in multiple form templates. These custom components, which include controls, data structure, and many of the design features you've learned about so far (such as rules and data validation), enable you to quickly and easily reuse common design features across as many form templates as you choose. Let's jump right in and see how easy it is to create one of these components.

Designing a New Template Part

Designing a template part is just as easy as designing a new blank form. In fact, you design a template part the same way you would design a form template. Let's look at an example. Let's say that you work in the IT department at the MOI Consulting Corporation. MOI has a mandate that all forms that collect address information should follow a standardized format and data structure for that part of the form. Figures 10.1 and 10.2 show the format and data structure, respectively. (These figures should look very familiar to you since this is the address block from the MOI employment application form we used in earlier chapters.) Until now,

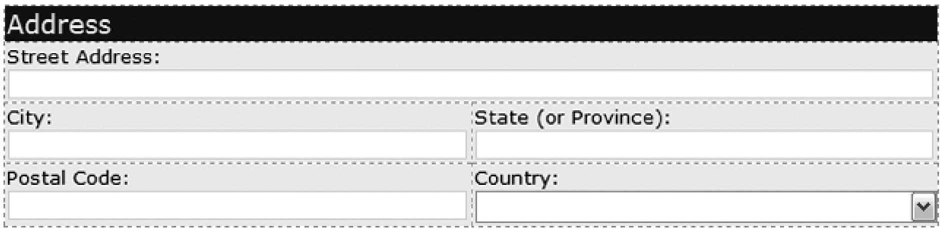

Address
Street Address:
City:
State (or Province):
Postal Code:
Country:

FIGURE 10.1: Address block section of the MOI employment application form

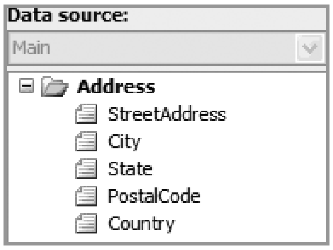
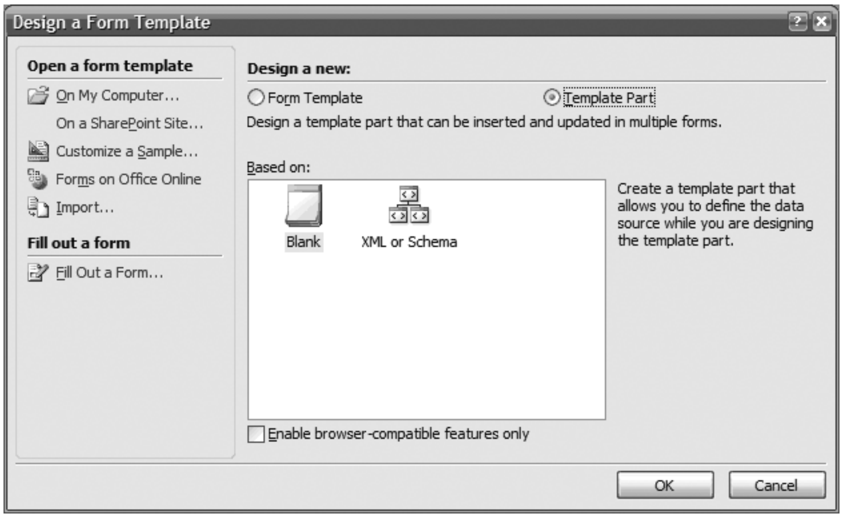


FIGURE 10.2: Data source for the address block template part

everybody at MOI who has created a form has either had to recreate this part of the form template from scratch or copy an existing form template and remove all the nonrelevant parts. Being the innovative form designer that you are, you figure out a way to make this process easier for everybody in the company—you create an address block template part. Let's look at the process for creating this component.

The first step (after starting InfoPath, obviously) is to open the *Design a Form Template* dialog as you normally would when designing a new form template. One thing you'll do differently, though, is to select the *Template Part* option at the top of the dialog, at which point the dialog changes to look like the one shown in Figure 10.3. Notice that the *Based on* section of the dialog looks a bit different. Only the *Blank* and *XML or Schema* options appear. This is because when designing a template part, you can either create a new



Form Template [✓ Template Part]

Design a template part that can be inserted and updated in multiple forms.

Create a template part that allows you to define the data source while you are designing the template part.

☐ Enable browser-compatible features only

FIGURE 10.3: Design a Form Template dialog with Template Part option selected

blank template part or base it on an existing XML file or schema. It's not possible to base your template part on a Web service, database, or connection library as you can when designing a form template.

Since MOI Consulting has a required data structure for address blocks in forms, it seems natural that you would want to create your template part based on an existing schema. However, if you don't have an existing schema or you want to create one for your address block, you can create a new blank template part. So, let's select the *Blank* option from the *Based on* section of the dialog and then click the *OK* button. When you do so, InfoPath opens in design mode with a completely new template part that contains nothing in the view or data source. We'll add controls to the view next, which will also create the data structure we require for this address block template part.

Template Part Design Mode

When you open InfoPath in design mode when designing a template part, at first everything looks the same as it does when designing a normal form template. However, as you look around you will notice subtle differences. For example, if you look at the *Design Tasks* task pane (Figure 10.4), you'll notice that it doesn't contain all the design tasks that it normally does when you are designing a form template—the *Views* and *Publish Form Template* tasks are missing.

To understand the reason behind this difference, it's important to understand the concept of design targets. Whenever you open InfoPath in design

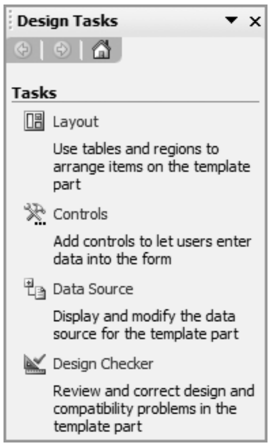

Design Tasks
Tasks
Layout
Use tables and regions to arrange items on the template part
Controls
Add controls to let users enter data into the form
Data Source
Display and modify the data source for the template part
Design Checker
Review and correct design and compatibility problems in the template part

FIGURE 10.4: Design Tasks pane when designing a template part

mode, it can have one of two targets—form template or template part—which correspond to the option buttons with the same names in the *Design a Form Template* dialog. (Of course, in each case, you can specify that the form template or template part is browser-enabled. We will discuss this more in Chapter 14 when we talk about browser forms.) When you are targeting a form template, all the InfoPath design-mode features are available. When you are targeting a template part, only a subset of the design-mode features (those supported by template parts) are available.

For example, a template part can have only one view. So, the *Views* task pane is not available in the *Design Tasks* pane (or anywhere else, for that matter). As with the tasks in the *Design Tasks* pane, the entry points (e.g., menus, dialogs, and so on) for any features that are not available in template parts will be hidden. Table 10.1 lists all the features that are not supported by

TABLE 10.1: List of Features Not Supported by Template Parts

Unsupported Feature	Description
ActiveX controls	ActiveX controls, which we will talk about in Chapter 19, are not supported in template parts.
User roles	User roles available from the *Tools* menu are not supported by template parts.
Information Rights Management	IRM is not supported.
Digital signatures	Digital signatures cannot be added to the entire form or to the Section or Optional Section controls.
Multiple views	Only one view is supported in template parts.
Word print views	Since you cannot create multiple views, you also cannot create a print view for use in Word.
Print settings	Print settings are specific to a form template and cannot be included in template parts.
Page settings	Page settings are not supported for the same reason that print settings are not supported.
Changing the view name	The view that a template part is inserted in is defined by the form template and not the template part. So, it doesn't make sense to change the view name for a template part.

continued

Table 10.1: List of Features Not Supported by Template Parts *(continued)*

Unsupported Feature	Description
Background pictures for the view	As with the view name, the background picture is set by the form template. However, setting the background color from the *View Properties* dialog is supported, and the color chosen there will be used as the background color for the template part itself when inserted.
Form code	Code (script or managed) is not supported in template parts.
Publishing	You do not publish template parts as you do form templates. Saving a template part has the same effect as publishing in this case.
Printing multiple views	Since multiple views aren't supported, printing multiple views is also not supported.
Color schemes	Color schemes are defined by the form template in which a template part is inserted. Template parts will honor the color scheme of the form template.
Form Options dialog	Many options on the *Form Options* dialog pertain to form templates only. Therefore, the menu item for this dialog is hidden.
Designing a template part from a data connection	You can design only a blank template part or one from an XML file or schema. You can add a secondary data connection after creating the template part, however.
Form submit	It is not possible to define the submit behavior for a form template in a template part. This is controlled by the form template itself.
Form merging	It is not possible to define the merge actions for nodes in a template part. The merge actions must be specified in the form template. (Merging forms is discussed in Chapter 12.)
Form export	It is not possible to export a template part as you can a form template.
Mixed namespace editing	If you create a template part from a fixed schema, you cannot add new nodes to the schema. This is because template parts don't support mixed namespaces in the schema.

template parts and, therefore, won't be available in design mode when designing a template part.

Now that you've opened InfoPath in design mode to design a template part, the experience should be very familiar since you've designed a form template before. So, let's create our address block component. As we mentioned, the address block we want to create is the same one used in the MOI employment application form. So, to make things a little easier, let's just copy all the controls and layout from the view in the employment application form and paste them into the view of the template part we are creating. To do this, open the employment application form we created back in Chapter 2 (which is available with the samples for Chapter 2 on the book's Web site), select the *Address* Section control by clicking the design-time visual tab for the Section control and then press Ctrl+C to copy it. Then go to the template part and paste the controls and layout into the view for the template part.

After you paste, you'll notice that all the controls have error design-time visuals, as shown in Figure 10.5. This is because when you copy controls

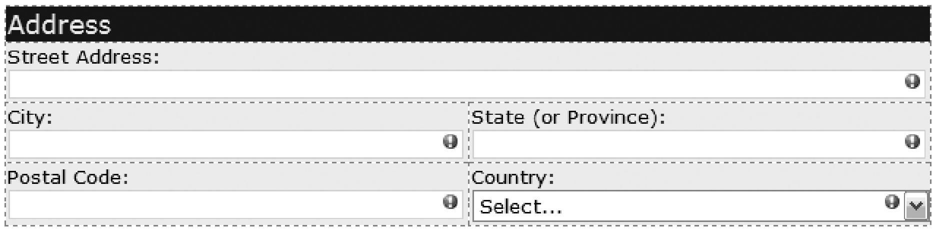


FIGURE 10.5: Address block after pasting it into the view

from one form template and paste them into another, only the view features are copied. None of the data bindings are preserved. (In fact, none of the data-centric features, such as data validation, calculated default values, and so on, are copied.) However, this is pretty easy to fix. First, you have to create the data structure shown in Figure 10.2. (If you need a refresher about how to create the data structure manually, see Chapter 3.) Once the data source is created, all you have to do is right-click on each control and choose the *Change Binding* option. Then from the binding dialog, choose the node to which you want to bind each control. (Of course, as you learned earlier, you can also create the data structure right from the binding dialog.)

Once you've changed the bindings for each of the controls, you have a fully functional template part that you can use in any form template. But before you can reuse this component in other form templates, you have to save it. In terms of what you do, saving a template part is no different than saving a form template. However, when you save, InfoPath will give the template part the .xtp extension instead of the .xsn extension used for form templates. This will be important later when you try to locate template parts to reuse. When you save the template part, the name you give the file will be the name you'll use when you install the template part into your *Controls* task pane, which we'll show you how to do next.

You've created your first template part, so let's put it to use. (In the last chapter, you learned about publishing form templates. You don't actually publish template parts as you do form templates. Saving a template part is analogous to publishing it in this case.)

Using a Template Part

Once you have a template part—whether you created it yourself as we just did or you are reusing one created by somebody else—you can add it to your *Controls* task pane so that you can use it just like any of the built-in InfoPath controls. Let's look at how you can install template parts so they can be inserted into your form templates.

Adding a Template Part to the Controls Task Pane

Reusing the address block template part in your own form template is easy. First you must create a new form template or open an existing one in design mode. Then open the *Controls* task pane. At the bottom of the task pane is a link titled *Add or Remove Custom Controls.* Clicking on this link opens the *Add or Remove Custom Controls* dialog (Figure 10.6). From this dialog, you can add or manage all your custom controls, whether they are template parts or ActiveX controls. If you've never added a custom control before, this dialog will be empty. However, once you add a control or two, each control name will appear in the *Custom controls* list box. (Figure 10.6 shows the *Add or Remove Custom Controls* dialog after adding the Address-Block template part.)

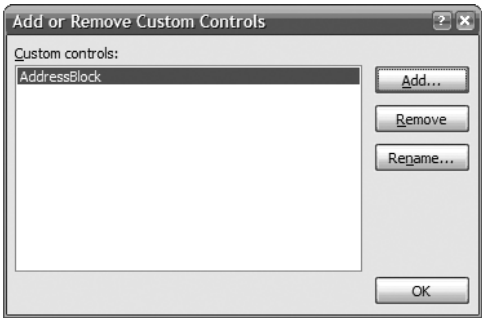
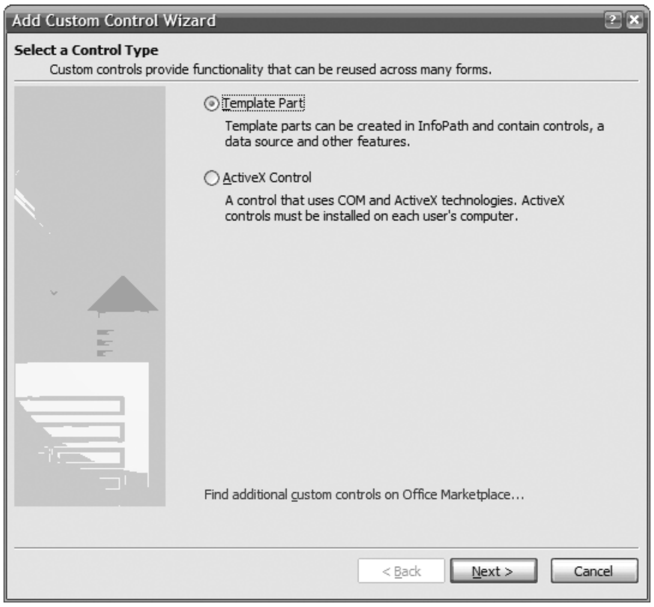


FIGURE 10.6: Add or Remove Custom Controls dialog



Custom controls provide functionality that can be reused across many forms.

[✓ Template Part]
Template parts can be created in InfoPath and contain controls, a data source and other features.

ActiveX Control
A control that uses COM and ActiveX technologies. ActiveX controls must be installed on each user's computer.

Find additional custom controls on Office Marketplace...

FIGURE 10.7: Add Custom Control Wizard's Select a Control Type page

In order to add a custom control, just click on the *Add* button, which will start the Add Custom Control Wizard. When you first start the wizard, the *Select a Control Type* page appears (Figure 10.7), from which you choose the type of control you want to insert into the *Controls* task pane.

As we've mentioned, InfoPath supports two types of custom controls—template parts and ActiveX controls. Since we are going to add a template part (and this option is the default), click the *Next* button, which will open

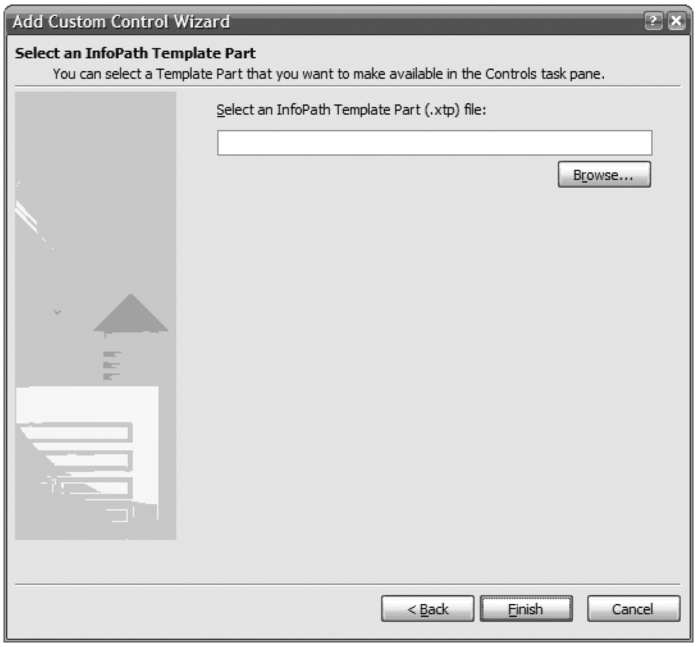
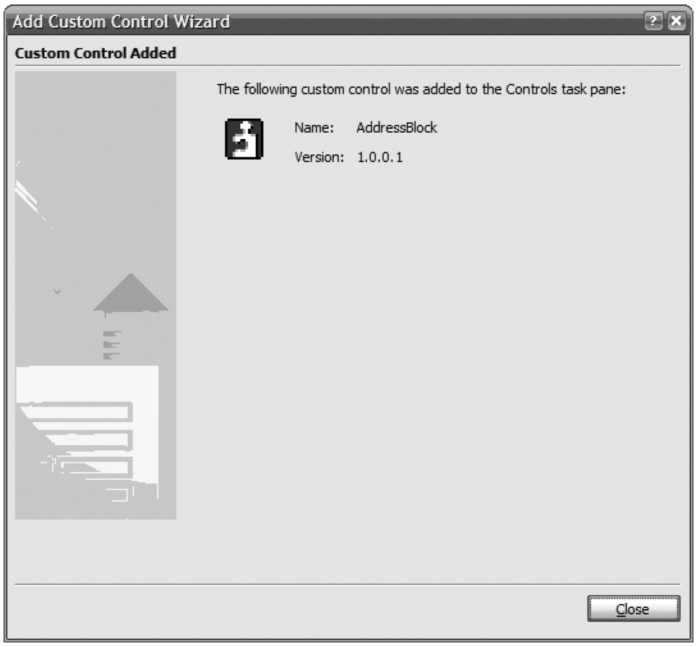

Add Custom Control Wizard
Select an InfoPath Template Part
You can select a Template Part that you want to make available in the Controls task pane.
Select an InfoPath Template Part (.xtp) file:
Browse...
< Back
Finish
Cancel

FIGURE 10.8: Add Custom Control Wizard's Select an InfoPath Template Part page

the *Select an InfoPath Template Part* page (Figure 10.8). From this page you can select which template part you want to install. Clicking on the *Browse* button will open the typical *Browse* dialog that we all know and love. Select a template part from your local machine, a shared location, a SharePoint site, or any Web site you choose. As we mentioned, template part files have an extension of .xtp, so this dialog will show only those types of files by default.

Once you choose the template part you want, click the *Open* button in the *Browse* dialog, and then click the *Finish* button after the dialog is closed. The *Custom Control Added* wizard page appears (Figure 10.9). As you can see, template parts have an icon and a version number associated with them. InfoPath automatically sets the version number and increments it each time you save the template part. We'll show you a little later in this chapter how to change the icon for the template part.



The following custom control was added to the Controls task pane:

Name: AddressBlock
Version: 1.0.0.1

FIGURE 10.9: Add Custom Control Wizard's Custom Control Added page

Once you click the *Close* button on the *Custom Control Added* page of the wizard, the name of the template part is added to the list of custom controls in the *Add or Remove Custom Controls* dialog, as we shown earlier in Figure 10.6. Once you close this dialog, the template part is added to the *Custom* category of the *Controls* task pane (Figure 10.10). Once the template part appears in the *Controls* task pane, you can insert it into your form template just like any of the built-in controls supplied by InfoPath.

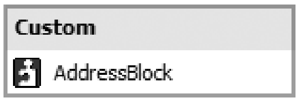


FIGURE 10.10: Custom category of the Controls task pane after adding the AddressBlock template part

NOTE Specifying an Alternate Location for Template Parts

When you add a template part to your *Controls* task pane, the component is installed on your computer in the following directory on Windows XP: C:\Documents and Settings\<*username*>\Local Settings\Application Data\Microsoft\InfoPath\Controls (where C:\ is your system drive and <*username*> is your login name). When you start InfoPath in design mode, it loads all the template parts from this location into the *Controls* task pane.

However, what if you want InfoPath to automatically retrieve template parts from an alternate location? Those who are designated as the administrator for your computer (which might be you) can create a registry key that points to an alternate location for template parts. This registry key is HKEY_CURRENT_USER\Software\Microsoft\Office\12.0\InfoPath\Designer. Create a string value called `IPCustomControlsFolder` under that key that points to the alternate location from where InfoPath should load the template parts.

There are a couple of things to note about this, though. First, this path can be only to a folder on your local machine or on a network share. Internet URLs won't work. However, if you want to place your template parts on a Web site, you can use folder mirroring to create a share that points to an Internet address. (You can do this by creating a new Network Place in Windows that points to the URL.) Second, if any of the template parts in the alternate location have the same template part ID as one already installed on you computer, each time you open InfoPath in design mode you will receive an error stating that the template part is already installed. In that case, the template part(s) from the alternate location will not be installed in the *Controls* task pane.

Inserting a Template Part into a Form Template

Once you install a template part by using the Add Custom Control Wizard, you can use it in any form template you create. Let's take a look at what happens when you insert a template part into a form template. As we mentioned, once the template part is in your *Controls* task pane, you can use it just like any of the other built-in controls—you can click on it in the task pane to insert it into the view, drag it into the view, or open the *Data Source* task pane and bind it to any field or group just as you can with the built-in InfoPath controls. If you insert your template part into a form template that has the *Automatically create data source* option set in the *Controls* task pane and you've created a blank template part (not one based on an XML file or schema), the data structure specified by the template part will be inserted into the form template. (We'll talk more about this shortly.)

If, instead, you open the *Data Source* pane to bind your template part to existing data in a form template, the data structure you can bind the template part to will depend on how the template part is defined. For example, look at the data structure for the AddressBlock template part in Figure 10.2 again. This template part is associated with a group node that has five fields below it. Therefore, you can bind the AddressBlock template part only to a group node that has at least five field nodes as children. If there are more than five child nodes, the first five nodes are used and the remaining nodes are ignored.

However, those first five nodes must match the data structure defined by the template part. Also, the data types of the nodes must be compatible with the data types to which you are trying to bind the template part (e.g., string and whole number data types are compatible but whole number and XHTML are not). Also, all template parts include at least one top-level group node. Therefore, in order to bind a template part to data in a form template, the data structure you are binding to must start with a group node. (Note, however, that it's not necessary for the field names in the form template to match the names of the fields in the template part itself.)

If you right-click on a group node in the *Data Source* task pane that fits the requirements just mentioned, the AddressBlock template part will be available. If you click on a group node that has only four children, for example, then the AddressBlock template part will not be available. (Note that template parts have a lower priority than the built-in control types. Therefore, to bind a template part to a node in the *Data Source* task pane, you may have to click on the *More* menu item on the context menu when you right-click on a node in the *Data Source* task pane. You will find template parts in the *Select a Control* dialog, which is opened from this menu item.)

Now, let's insert our AddressBlock template part into a new form template and see what's happening behind the scenes. Figure 10.11 shows this template part after inserting it into a blank form. As you can see, it looks a little different than the address block shown in Figure 10.1. In this case, the entire address block is inside a Section control. Since template parts are custom components, they are always contained within a Section control.

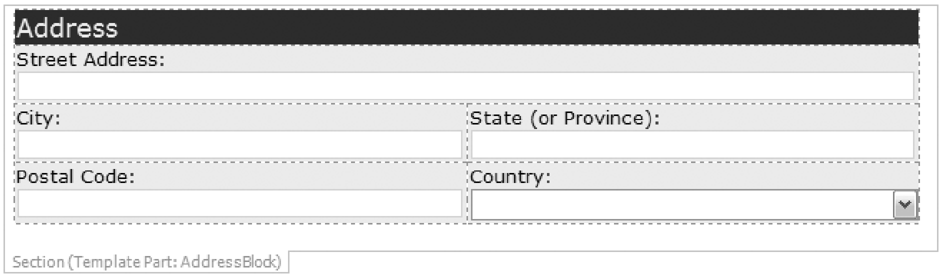


FIGURE 10.11: AddressBlock template part inserted into a blank form

This helps InfoPath treat the template part as a component, which is necessary for such things as updating existing parts. (You'll see why this is needed later in this chapter.)

Identifying a Template Part in the View

Template parts are very easy to identify because they are always contained within Section controls. The design-time visual for the Section control for a template part contains text that identifies it as a template part. The name of the part is also included in the design-time visual, as you can see in Figure 10.11. (Take a look at the tab at the bottom of the Section control.)

Once the template part is in the view, the contents of the template part (e.g., controls, calculated default values, data validation, and so on) all become part of the form template you are designing. In other words, once inserted into the form template, the template part is no longer a component treated as one atomic unit. You are free to move controls around in the view (or delete them); change rules, data validation, and calculated default values; change the data structure; and so on.

However, since the template part is no longer an atomic unit, you cannot delete everything added to the form template when you inserted the template part with one simple delete operation. (*Undo* won't help in this regard, either.) If you want to remove everything you added, you have to delete every item individually—the controls (which can all be deleted by deleting the template part Section control as long as you haven't moved any controls outside of it), the data structure, and so on.

As you can see, template parts provide you with an easy way to share and reuse aspects of your form, but they are not atomic controls such as a Text Box or Date Picker. In fact, template parts are more like the compound built-in controls such as the Master/Detail or Horizontal Repeating Table.

When you insert a template part into the view from the *Controls* task pane, how the data binding for the template part is handled depends on how it was originally created. If it was created as a blank template part, when you insert it into a form template, InfoPath treats it like one of the built-in controls. If your form template into which you have inserted a template part was created based on a blank form template or your form template is based on an open schema, not only are the controls inserted into the view, but the underlying data structure is created as well. If you designed your template part based on an existing XML file or schema, when you insert the template part into a form template, you will be asked to choose the group node to which the template part should be bound. (Of course, in either case, if you insert a template part into a form template that is based on a fixed schema, you will always be asked to choose the group node to which the template part should be bound. You must bind the template part to a group node since the component is always wrapped in a Section control, as mentioned earlier.)

Since our AddressBlock template part was created from scratch, when you insert it into a form template that allows editing of the data source, a group node containing five field nodes will be inserted into the main data source. As with other controls, where in the data source this group node is inserted depends on the current data source context in the view when you insert the control. For example, if the insertion point is inside a Repeating Section control in the view when you insert the AddressBlock, the group node for this template part will be inserted as a child of the repeating group node to which the Repeating Section is bound.

In addition to creating the data structure, when a template part is inserted, any calculated default values, rules, data validation, or data connections associated with the component are incorporated into the form template as well. When you insert a template part into a blank form template, this just means that these items are added to the new form template. However, if you are inserting a template part into an existing form template, you should be careful. Be particularly careful if you are binding it to a node that has calculated default values or data validation associated with it already or if your template part has rules and/or data connections associated with it and you have previously inserted it into the same form template.

NOTE **Issue with Data Connections When Inserting Multiple Instances of the Same Template Part**

If you have a template part that contains a data connection, when you insert multiple instances of that template part into a form template, all prior instances lose their connection to the secondary data source. For example, let's say that you created a template part that contains a Drop-Down List Box control that gets its list items from a secondary data source. When you insert two instances of that template part into a form template, the second instance retains its connection to the data source, but the first instance loses the connection. That means that, when a user fills out the form, the first instance of the Drop-Down List Box will have no list items, but the second one will. In this case, you have to reset the list items by opening the properties dialog for the Drop-Down List Box and specifying that the list items come from the same secondary data source as the second instance of the template part you inserted into the form template.

When you insert a template part, if you are binding it to an existing node that has calculated default values or data validation, the existing values and data validation will be replaced. If the values and data validation were created when you previously inserted this template part and you haven't made any changes, this shouldn't cause a problem. If you have made changes, though, you will lose those changes when you insert the template part. (This is a good reason to make regular backups of your form templates.) You will be warned about the fact that you might lose some of your changes ahead of time and will be given the option to cancel. However, the warning helps only if you know ahead of time that you have made changes to these items in the form template, which you may not know if you're maintaining an existing form template, for example. You can use the Logic Inspector as explained in Chapter 5 to determine whether there are calculated default values and data validation associated with a node.

If your template part has rules and data connections associated with it and you have previously inserted this template part into the form template, when you insert it again, those rules and data connections will also be replaced. However, in this case, only those rules and data connections associated with this particular template part will be replaced. Those from other template parts will not be affected. The main point here is to be careful and to know what calculated default values, rules, data validation, and data connections exist in your form template. Also, if you are modifying an existing form template, it's always a good idea to make a backup copy before you make any changes.

Template Parts with No Controls

It's possible to create a template part that contains no controls whatsoever. You may want to do this, for example, in order to reuse a particular data structure, complicated rules, or calculated default values. However, even though the template part contains no controls in the view, when you insert it into a form template, it will still contain a single Section control. You can delete that Section control from the view if you like. However, if you do so, there will be no way for you to update existing form templates with new versions of the template part.

Updating Template Parts

Let's say that your IT department has finally completed the AddressBlock template part and distributed it to the entire company. It has been six months, and now hundreds of form templates use this address block. Form template designers love this component because it saves them a lot of time and energy. Instead of having to add the address block to their form templates manually, they can now just insert the AddressBlock template part and their work is done.

However, users have been complaining that it's cumbersome to have to type the city and state every time they use a form that collects address information. They would like to choose this data from a list of cities and

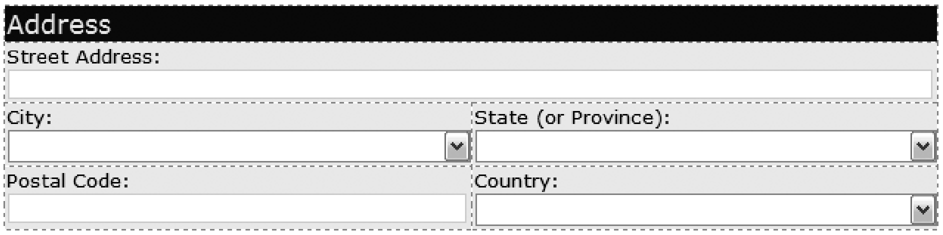
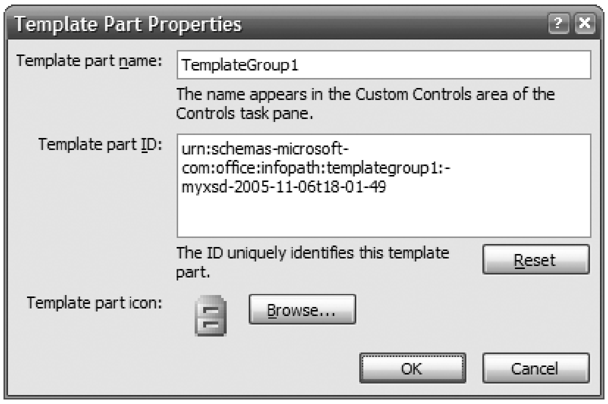


FIGURE 10.12: AddressBlock template part with cascading Combo Box controls

states. The designers in the IT department have graciously decided to make that change to improve the usability of forms throughout the company. They have updated the AddressBlock template part to replace the two Text Box controls for city and state with two cascading Combo Box controls—one that contains the state and another that shows the cities in the chosen state. (Also, the list of states will depend on which country is chosen from the country Combo Box. See Chapter 7 if you need a refresher about how to implement cascading List Box controls in a form template.) Figure 10.12 shows the new version of the AddressBlock template part after making these changes.

The other piece of feedback the IT department has received, this time from the form template designers, is that the AddressBlock component uses the default icon for all template parts. This makes it difficult to quickly differentiate between multiple template parts in the *Controls* task pane. So, the designers in the IT department decide to change the icon as well. Let's walk through how to do this.

To change the icon, first open the *Template Part Properties* dialog for the template part (Figure 10.13). You can open this dialog by clicking on the *Properties* menu item on the *File* menu while designing the template part. From this dialog, you can change the name of the template part, the ID, and the icon associated with the template part. The name and ID are generated automatically the first time you save your template part. (We'll change these later in this chapter.) The icon is set to a default icon for all template parts. The dialog in Figure 10.13 shows the *Template Parts Properties* dialog after changing the icon.



TemplateGroup1

The name appears in the Custom Controls area of the Controls task pane.

urn:schemas-microsoft-com:office:infopath:templategroup1:-myxsd-2005-11-06t18-01-49

The ID uniquely identifies this template part.

FIGURE 10.13: Template Part Properties dialog

Updating a Form Template That Contains the AddressBlock Template Part

Once the designers in the IT department have completed all the necessary changes to the AddressBlock template part, they tell all the form template designers in the company that a new version of the AddressBlock is available. As a form template designer, you want to take advantage of these changes. You also want an easy way to update the form templates that use existing AddressBlock template parts. It could be quite tiring to open each form template and search for all the template parts in every view before you can update them. Fortunately, there is an easier way to do this. You do have to open each of the form templates, but, once you do, you can update all instances of the AddressBlock throughout the form template (even across multiple views).

The first thing you must do is install the new AddressBlock. You install the new template part the same way you did the first time you installed the component. (If you have previously set up a repository of template parts using the registry key we talked about earlier, this step isn't necessary. In that case, InfoPath will load the new template part automatically.) Once you install the new version of the AddressBlock template part, all the existing AddressBlock components in the form template will now have a warning design-time visual, as shown in Figure 10.14. This visual simply alerts you to the fact that there is a new version of this component. When you see this design-time visual, if you then right-click on the tab at the bottom of

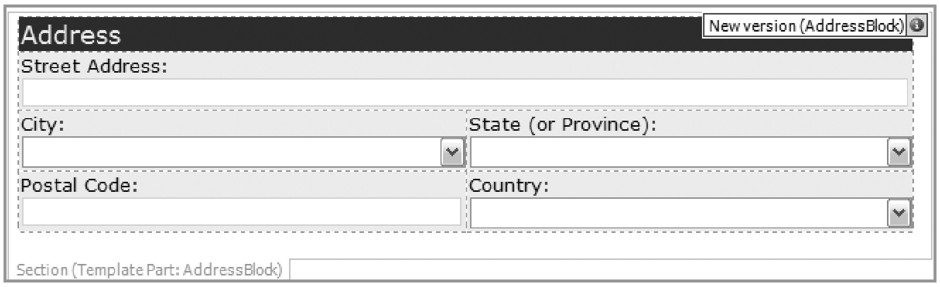
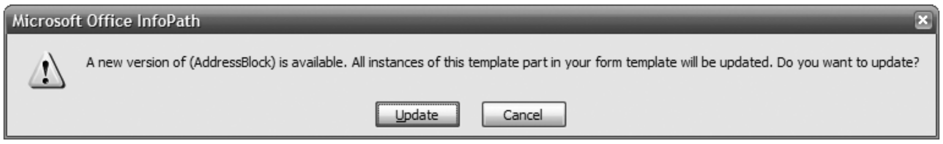


FIGURE 10.14: Design-time visual shown on the new AddressBlock template part

the Section for the template part, you can choose *Update* from the context menu. This will update all the template parts of that type throughout your form template, even in different views. This is quite useful if you have a form template that uses many instances of the same template part.

In addition to the *Update* context menu item, you will see a *More Details* menu item on the context menu. Clicking on this item opens the dialog shown in Figure 10.15, which will give you more information about why there is a warning design-time visual on this Section control. You can then choose to update all AddressBlock template parts in the form template by clicking the *Update* button.



A new version of (AddressBlock) is available. All instances of this template part in your form template will be updated. Do you want to update?

FIGURE 10.15: More details for the new template part

As we mentioned earlier, when you insert a template part into a form template, it's no longer an atomic component. The controls contained in the template part become part of the current form template. You are free to move the controls around or delete them entirely. However, when you update a template part, the component will be reinserted. That means that view changes are lost. For example, let's say that you deleted the city Combo Box from the template part. When you update the template part, the city Combo Box will be reinserted. (If you have previously moved pieces of the template part around in the view, those items will remain in the view.)

Also, when you update template parts, any view changes will be incorporated into the current form template, as will any calculated default values, rules, data validation, or data connections associated with the template part. When you update a template part that contains these features, as is the case when inserting a template part into an existing form template, any existing values, rules, data validation, or data connections will be replaced, as we talked about earlier, and not simply appended to the form template. In addition, in the case of updating an existing template part, if the component has any conditional formatting associated with it, those conditions and actions will also be replaced. Of course, when you update your template part, you will be warned that you may lose this data, but that won't help you if you've spent hours making those changes to your form templates in the first place. So, as we mentioned earlier, it's always a good idea to back up your form template before you update a template part.

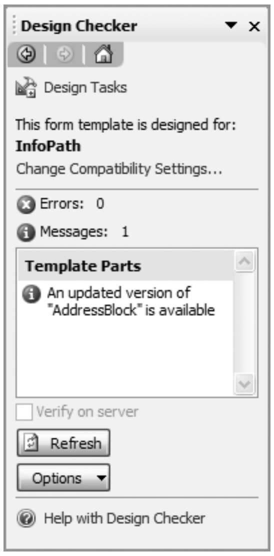
Locating All Template Parts That Need to Be Updated

As you can tell, updating all template parts is relatively easy. However, what if you want to identify all the template parts that need to be updated not only in the current view but also in every view of your form template? You may, for example, want to inspect all instances of a template part before you choose to update them all so that you can determine whether there are any customizations (e.g., conditional formatting, rules, and so on) that you want to save. Locating all template parts that need to be updated is relatively easy as well.

First, go to the *Design Tasks* pane by clicking on the menu item of the same name on the *View* menu. Then, click on the *Design Checker* link in the task pane. This opens the *Design Checker* task pane (Figure 10.16). The *Design Checker* task pane provides you with a way to check your form template for various types of issues. This includes updates needed for template parts as well as many other items (which we'll talk about in more detail in Chapter 14).

Notice that a warning appears in the *Design Checker* task pane shown in Figure 10.16. (There will be one warning for each template part in the form template.) This warning tells you that there is an AddressBlock template part in the form template that needs to be updated. If you click on the warning text in the task pane, the template part that needs to be updated will be

34629 10 431-454 r1 bm 1/11/07 5:25 PM Page 452


Design Checker

Design Tasks

This form template is designed for:
InfoPath
Change Compatibility Settings...

Errors: 0

Messages: 1

Template Parts

An updated version of "AddressBlock" is available

FIGURE 10.16: Design Checker task pane showing template parts that need to be updated

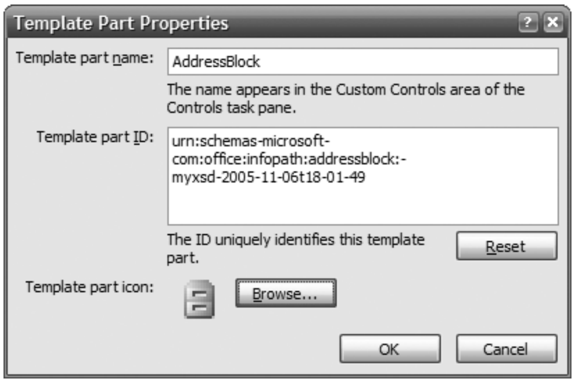
located. If the component is in the current view, it will simply be selected. If the component is outside of the viewable area of the current view, the view will be scrolled to show it. If the template part is in another view, the view will be switched to that view and the template part will be selected. Once you've located all the instances of the template part that need to be updated, and you've determined that it's okay to update them, you can then right-click on one of the components and update the current template part and any other template parts of this type by clicking the *Update* menu item as you normally would.

Customizing Existing Template Parts

Now that you know how to use template parts in your form templates, the next thing you'll probably want to do is go out to the Web and find some interesting template parts that you can reuse. There is no doubt that, when you do, you'll want to modify an existing template part in some way. For example, you may want to add your company logo, change or add some data validation, change the name of the template part, or simply change the icon.

Customizing an existing template part is very easy. When you find the template part you want to change, the first thing you'll probably want to do is download it to your computer (and pay for it if it isn't free). Then, you just open it in design mode in InfoPath as you would for any other template part. Once you make the changes you want, the next thing you'll obviously want to do is save. However, when you save, at this point you haven't created a new template part. All you've done is update an existing one. That means that if you then distribute this component to your users who happen to be using the original one, when they install your component and open a form template that uses the original one, they will be warned by InfoPath that an update is needed.

Maybe this is what you want. It is more than likely, though, that this isn't what you intended. Instead, you probably wanted to create a completely new template part based on an existing one. The one thing to remember is that a template part has an ID that uniquely identifies the component. (Refer back to Figure 10.13.) When you create a new template part, this ID is generated automatically for you. When you customize an existing part and save it, the ID does not change. Therefore, in order to create a new template part, you must change the ID. (It's also a good idea to change the name as well. In fact, when you change the name of the template part the ID is automatically changed. This is one easy way to create a new template part based on an existing one.) Figure 10.17 shows the *Template Parts Properties* dialog again, but this time after we changed the name and ID of the template part.



Template part name: AddressBlock
The name appears in the Custom Controls area of the Controls task pane.
Template part ID: urn:schemas-microsoft-com:office:infopath:addressblock:-myxsd-2005-11-06t18-01-49
The ID uniquely identifies this template part.
Template part icon:

FIGURE 10.17: Template Part Properties dialog after changing name and ID

The name is simply a string and can be any name you choose. The ID is a Uniform Resource Name (URN) that uniquely identifies the template part. As you can see from Figure 10.17, the URN contains the name of the template part, the date, and the time, which helps to uniquely identify the template part.

If you want to create a new template part, you must change this URN. For example, you could change the name part of the URN and the date and time by hand. (You'll notice that when you change the name of the template part, the ID changes as well, but not vice versa.) If you decide that you want to regenerate the template part ID based on the new name you entered, just click the *Reset* button. Once you change the URN and save, you have a completely new template part. (Of course, since it's possible to manually edit the template part ID in the *Template Part Properties* dialog, it is also possible to enter the ID of an existing template part. This is only useful if you want to overwrite an existing part and not create a new one.)

What's Next?

In this chapter, you learned how to create your own components. Now you can not only design professional form templates but also create components that can be reused in other form templates. Template parts make designing form templates much easier and also give you a way to ensure consistency across the form templates in your organization.

In the next chapter, we'll talk about two of the final steps involved in designing a form template—security and deployment. We'll discuss how you can add digital signatures to your form templates in design mode and how you can sign the forms you fill out.