

# Solve Common Business Problems with InfoPath and SharePoint

Ryan Thomas

(Reprinted from WindowsITPro Magazine)

Microsoft Office InfoPath 2007 is one of the lesser-known tools in the Microsoft Office suite of applications. Compared with Word and Excel, it has a much smaller user base and an even smaller number of people who actually know what to do with it. In this article I explain what InfoPath can offer, focusing on how to use it with Microsoft Office SharePoint Server (MOSS) 2007. I use a common real-world expense report example to illustrate InfoPath's benefits. SharePoint's recent popularity makes InfoPath a useful tool that your organization should investigate and evaluate.

## InfoPath Basics

InfoPath is essentially a tool for designing and creating forms. The application allows nontechnical users to build and deliver methods to collect and manage data. Although a common perception is that you can accomplish the same tasks with Word or Excel, InfoPath provides greater functionality. In addition, you can easily convert Word and Excel files to more robust InfoPath data-collection forms.

InfoPath is really just a package of associated files. At its heart is an XML file that represents the data source for your collected data inside the forms. This flexible format is extremely useful for additional applications to read and process the form data. The designer or front-end view is simply XSL, with some additional files to manage rules, data connections, and so forth. If you rename your InfoPath template with the .XSN extension to a .CAB file, you can extract and view the individual components as text files, and you can easily see how they are connected.

InfoPath has built-in capabilities to connect with Microsoft SQL Server, Access, SharePoint, and Web Services to read and write data to a significant number of additional applications and data stores. These features make InfoPath an excellent option for building small applications that connect to multiple systems at once to select and update data. In addition, InfoPath can then collect and send data in human-readable formats via e-mail or to SharePoint. Most of these tasks can be accomplished with no compiled coding.

Two significant features of standard InfoPath development are the rich rules and validation components that users can build without code. The application lets the form designer view and manage common interface controls. The underlying data source can be viewed and manipulated with intuitive XPath functions abstracted away from the designer. For example, you can have a number of rules on a control; these rules check the contents or any other controls, process calculations, or immediately let you know which rules passed or failed. Rules can be strung together to cover some fairly sophisticated data validation and specific display control management.

You can save sections of forms as templates for reuse across multiple forms. This approach eliminates cutting and pasting and gives organizations the option to build components with specific functionality or required schema items to share with form designers.

## SharePoint Integration

InfoPath connects natively to SharePoint in multiple ways. It can read data from SharePoint lists quickly and easily, query live SharePoint data, and return results to the form to process a variety of options for the designer or end user. Connection options include binding data to drop-down lists for selection, obtaining user profile information, and querying sources of configuration data for rules,

# Solve Common Business Problems with InfoPath and SharePoint

Ryan Thomas

(Reprinted from WindowsITPro Magazine)

validation, and much more. InfoPath forms can be stored locally in SharePoint document libraries in the same way as any other type of document. They can be made the default template for a given content type, allowing the New command on a list to automatically create an instance of these custom forms to open, fill out, and save locally in the library for business processing.

## Forms Services

Some of the real power when using InfoPath with SharePoint lies in the use of InfoPath Forms Services, which is an enterprise feature of SharePoint that dynamically translates an InfoPath form to a web page via specific server technologies. Consider my previous example, in which an InfoPath form is used as the template for a content type. Web-enabling the form lets you build and publish forms directly to SharePoint and use them to start collecting XML data immediately, without requiring any additional client software beyond a web browser.

Forms Services is context-aware from a SharePoint perspective. It knows who is logged on, giving you additional flexibility in managing permissions and security for data access. When querying and using SharePoint data, you get the built-in security trimming to ensure that only appropriate access is given for each form instance.

Significant options are available for designing forms and collecting data. InfoPath is designed to send chosen fields to SharePoint fields as metadata, using out-of-the-box functionality with very little user effort. This data can then be searched with SharePoint's robust indexing and searching components or used to drive workflow, business logic, or custom applications that already exist within your environment.

InfoPath is a significant upgrade over standard SharePoint data collection with built-in lists. Typically with SharePoint lists, the designer has limited ability to make changes to the out-of-the-box *new forms* or *edit forms* that are generated on all SharePoint lists. These standard forms lack certain flexibility, such as the ability to limit access to specific fields when editing a SharePoint list item, or provide dynamic access to controls or additional data sources outside of traditional SharePoint lookup columns. SharePoint's native storage mechanism of list items limits the potential for exporting and accessing the list data in the robust way an InfoPath XML form can. All of these problems are quick and easy to address if you choose InfoPath as the form to collect data. InfoPath is also extremely easy to set up.

## Using InfoPath

Now that I've given you a basic overview of InfoPath, let's look at a common problem that organizations are using the application to solve. Perhaps this example will spark some ideas for improving efficiency or data management within your own company.

A common use for InfoPath is converting Excel or paper-based expense report forms to consolidate them into a digital environment. InfoPath's design surface is well-suited to manage the repetitive nature of this data and upload it to a SharePoint list where it can be calculated, categorized, and sent to managers and accounting for approval using either an out-of-the-box or a custom-built workflow.

I'll walk through the process at a high level to explain what pieces need to be built and how they are assembled. Our expense projects seem to break down roughly into the following steps:

# Solve Common Business Problems with InfoPath and SharePoint

Ryan Thomas

(Reprinted from WindowsITPro Magazine)

- Converting existing forms to InfoPath
- Approval
- Connecting to external applications

I'll include some extra components that aren't required, to give you an idea of how to easily extend the project with additional functionality.

**Convert existing forms to InfoPath.** Existing paper-based forms, along with Word and Excel files, can be cataloged and converted to equivalent InfoPath forms or aggregated into a single flexible form. For example, an expense form converted to an InfoPath form will collect expense data from individual users, to be submitted to a form library.

Figure 1 shows an InfoPath expense form that can look up current exchange rates, mileage allowances, and so forth, then calculate the amounts and automatically assign them as line items. The form can also look up existing user information, require explicit sign-off, and more.

Figure 1: InfoPath form for gathering expense account information

| Date     | Description | Expense Type | Mileage | Receipt Currency | Currency Amount | Exchange Rate | US Dollar Amount | Mileage Receipt                     | Receipt                             | Exchange Rate receipt               |
|----------|-------------|--------------|---------|------------------|-----------------|---------------|------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| 8/6/2009 | Lunch       | MEALS        |         | USD              |                 |               | \$125.34         | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input type="checkbox"/>            |
| 8/7/2009 | Hotel       | TRAVEL       |         | USD              |                 |               | \$179.89         | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input type="checkbox"/>            |
| 8/7/2009 | Drive       | GAS / AUTO   | 78      |                  |                 |               | \$45.24          | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/>            |
| 8/9/2009 | Lunch       | MEALS        |         |                  | 42.50           | 1.05          | \$44.63          | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |

As expense reports are added to a form library, code can be attached to the library to look at the form data and adjust permissions as necessary. The code can extract specific data elements and apply that data to existing local or external applications based on accounting rules. Although this element isn't required, it adds significant options to the overall application and is a good place to start writing some lightweight custom code as part of the application.

**Approval.** An out-of-the-box or custom workflow is created to assign forms to a user's manager for approval, then finally to accounting for approval. A custom workflow can override existing task edit forms with custom InfoPath forms. Custom forms are valuable in their ability to collect and process custom data as users complete their approval tasks for expense reports. Data can be as simple as a check box asking for additional verification, as Figure 2 shows, or as complicated as querying additional systems to look up and apply that data to the local expense report.

# Solve Common Business Problems with InfoPath and SharePoint

Ryan Thomas

(Reprinted from WindowsITPro Magazine)

**Figure 2: Task to obtain manager approval for an expense report**

The screenshot shows a SharePoint workflow task interface. At the top, a breadcrumb trail reads 'CSL > Expenses > Tasks > Expense Report - Manager Approval'. Below this is a yellow header bar with the title 'Tasks: Expense Report - Manager Approval'. A blue bar contains a 'Delete Item' button. A yellow bar below that states 'This workflow task applies to spadmin-2009-06-30-2.' The main content area has a blue header 'Manager Expense Report Approval'. It contains three numbered instructions: 1. 'If you approve this request, please select Approved, add appropriate comments, and click Submit.' 2. 'If you wish the initiator to make changes, do not select either checkbox, simply add comments and click Submit.' 3. 'If you wish to completely reject the request, select Reject, add appropriate comments, and click Submit.' Below the instructions are two checkboxes: 'Approved: ' and 'Rejected '. A 'Comments:' label is followed by a large white text input area. At the bottom left is a 'Submit' button.

For additional functionality, you can easily build a custom web part that lets users attach digital receipts and proof of expenses to their expense reports. These files can be attached to specific line items on an expense report, as Figure 3 shows. Accountants play a large role in managing data and can benefit from a custom dashboard that shows all expense data as individual line items, with appropriate attachments categorized into custom accounting codes for easier management and exporting.

# Solve Common Business Problems with InfoPath and SharePoint

Ryan Thomas

(Reprinted from WindowsITPro Magazine)

Figure 3: Custom web part with proof of expenses attached

The screenshot displays a web interface for managing expense reports. It features three main sections: a summary table for expense reports, a detailed table for accounting management, and a table for associated attachments.

| Expense Reports     |                   |                |                 |                     |                |              |    |                         |  |  |  |  |
|---------------------|-------------------|----------------|-----------------|---------------------|----------------|--------------|----|-------------------------|--|--|--|--|
| Name                | Tit Employee Name | AccountingLink | AttachmentsLink | Accounting Currency | Dt Submit Date | ExpenseTotal | 30 | Title                   |  |  |  |  |
| spadmn-2009-10-06-6 |                   | Accounting     | Attachments     | US Dollar           | 10/6/2009      | 350.47       | 24 | spadmn-2009-10-06-6.xml |  |  |  |  |

||| Add new document

| Accounting Expense Report Management |             |              |                 |         |            |              |          |        |           |      |                 |                    |
|--------------------------------------|-------------|--------------|-----------------|---------|------------|--------------|----------|--------|-----------|------|-----------------|--------------------|
| Date                                 | Description | Expense Cat. | Expense Cat. ID | Mileage | Rcpt Curr. | Curr. Amount | Ex. Rate | Amount | File Rcpt | Rcpt | Exch. Rate Rcpt | Attachments        |
| 8/6/2009                             | Lunch       | MEALS        | 215-800-80140   | 0       | USD        | 0            | 0        | 125.34 | No        | Yes  | No              | ADWPivotSheet.xlsx |
| 8/7/2009                             | Hotel       | TRAVEL       | 215-800-80130   | 0       | USD        | 0            | 0        | 179.89 | No        | Yes  | No              | ADWPivotSheet.xlsx |
| 8/7/2009                             | Drive       | GAS / AUTO   | 215-800-80115   | 78      |            | 0            | 0        | 45.24  | Yes       | No   | No              | ADWPivot.xlsx      |

| Expense Report Associated Attachments |                     |              |
|---------------------------------------|---------------------|--------------|
| Attachment                            | Attachment Category | Expense Item |
| ADWPivot.xlsx                         | Mileage Receipt     | Drive        |
| ADWPivotSheet.xlsx                    | Receipt             | Lunch, Hotel |

**Connecting to external applications.** As an added bonus, custom code can be written from various components in the application to connect to external line-of-business systems. One of the more common requests is to export approved expense reports directly into an organization's accounting system via web services or additional connectivity options.

## Well Worth Trying

Combining InfoPath and SharePoint gives non-software developers many options for collecting and managing data, beyond simply using SharePoint lists and metadata to create standard list collection forms—and all without writing a single line of code. Developers might also want to check out InfoPath's robust features and capabilities before they resort to using Visual Studio to build an ASP.NET web form or an entire new application. You can still add code to InfoPath behind partial classes to scale the application to fit your needs.

InfoPath's learning curve isn't very steep, and you can jump right into using it. The basic functionality is as easy to learn as for Word and Excel. Microsoft's movement toward collaboration and server-based offerings in its Office suite, combined with SharePoint's features, make InfoPath a useful tool for achieving your business goals. SharePoint 2010 and InfoPath 2010 will combine even more integration and allow for more granular flexibility in terms of converting existing list forms to custom InfoPath documents quickly and easily.

## InfoPath 2010 with SharePoint 2010: A Walkthrough

InfoPath is a versatile data-collection tool designed to make the creation of forms within SharePoint extremely easy, efficient, and connected to out-of-the-box or custom workflows. And yet, for IT pros and even SharePoint developers, InfoPath remains a mysterious application. Because SharePoint adoption is steadily increasing, I want to pull back the veil to show you how you can integrate InfoPath forms in your SharePoint environment without the need for coding.

In "Solve Common Business Problems with InfoPath and SharePoint" (InstantDoc ID 103462), I discussed the benefits of combining InfoPath with SharePoint. In this article, I'll dig a little deeper, showing you how to build a form from the ground up, using InfoPath 2010 Beta 2 and SharePoint 2010 Beta 2. As I walk through some examples, I'll also point out ways to enhance certain components or functionality *with* custom code.

# Solve Common Business Problems with InfoPath and SharePoint

Ryan Thomas

(Reprinted from WindowsITPro Magazine)

## Getting Your Feet Wet

First, you'll want to use InfoPath 2010 to create your first form. I want to focus mostly on InfoPath, so I won't spend too much time on the SharePoint infrastructure; however, I'll cover some of the SharePoint 2010 requirements where necessary. To get started, open InfoPath 2010 and click File, New from the top ribbon. You should see an image similar to Figure 1.

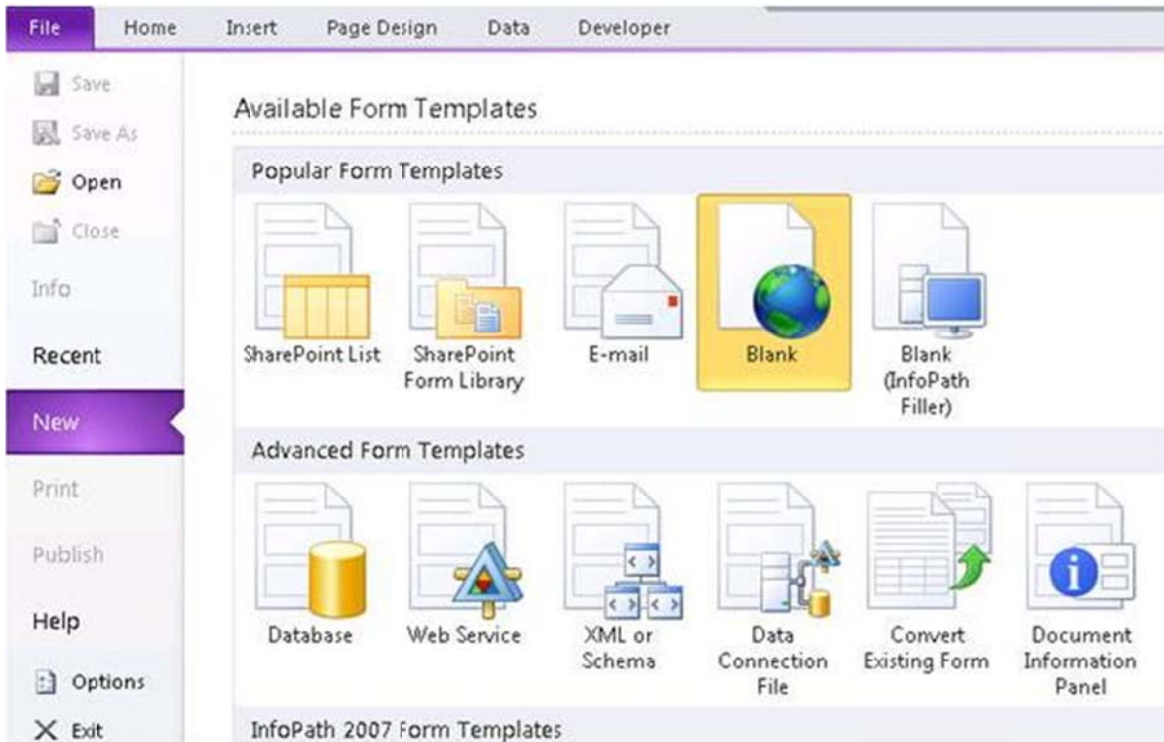


Figure 1: Creating a new form

Select the SharePoint Form Library template, then choose *Design this Form* in the navigation pane. You'll see a new form in the design window with a basic-looking table from which to add labels and controls. To start, we'll create a lightweight expense report form to illustrate how quickly you can begin to gather data into SharePoint.

Within the designer, you can select a color scheme under the Page Design tab. Remove the unwanted rows and add some quick labels to get the form to look bland and ready to have controls added to it, as Figure 2 shows.

# Solve Common Business Problems with InfoPath and SharePoint

Ryan Thomas

(Reprinted from WindowsITPro Magazine)

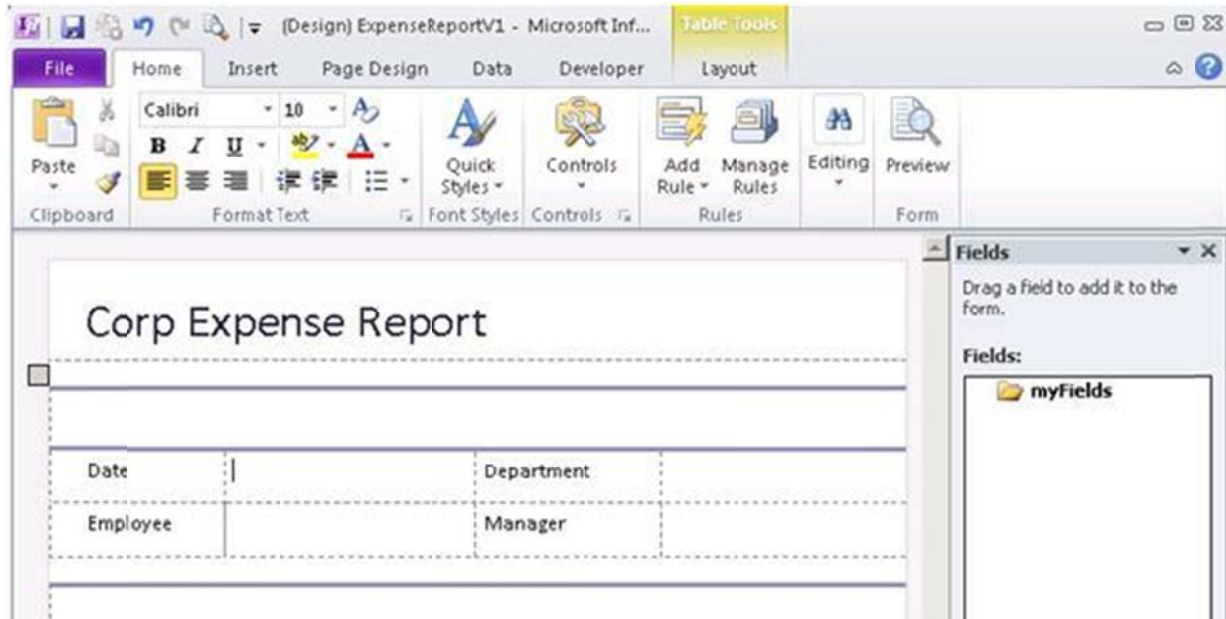


Figure 2: Ready for controls

Place your cursor in the table cell to the right of the Date cell, and from within the Home tab select the Controls menu item. Within this menu, select the DatePicker control. You should see a new DatePicker control added to your design surface where your cursor was placed, and a new XML element added to your Fields box on the right, as you see in Figure 3. These fields represent the data you'll capture and their associated type. If you right-click the DatePicker control and select Properties, you can set formatting rules and supply a more readable name for the underlying data it represents. As you add more controls, you'll get a growing XML container file ready to capture your form data in a standard format.

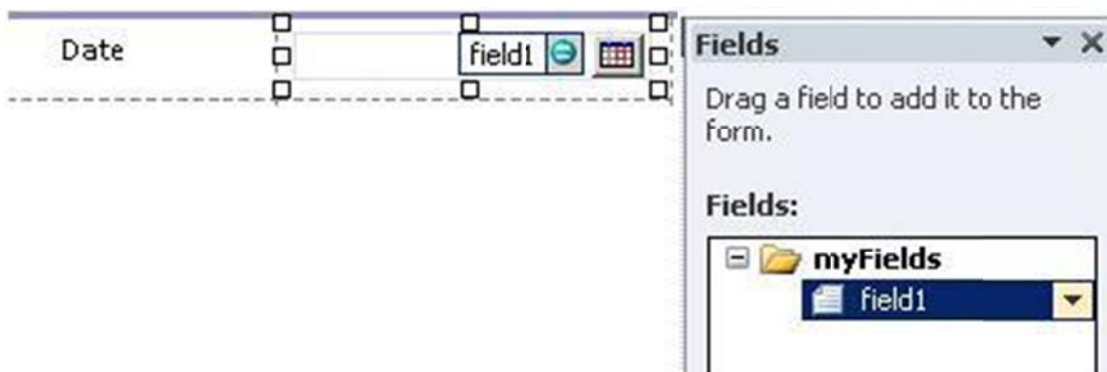


Figure 3: DatePicker control

When adding data elements and associated controls, you should strongly consider giving your data the correct type (e.g., numeric, text, date, Boolean) and applying specific formatting and rules for these controls—anything from enforcing currency formatting to complex functions such as using

# Solve Common Business Problems with InfoPath and SharePoint

Ryan Thomas

(Reprinted from WindowsITPro Magazine)

XPath to evaluate other data elements to enable or disable controls based on the value of one or more other controls within your form. Doing so can add significant options.

I've filled out the design surface with some additional controls to show you what a basic expense report might look like. InfoPath provides much more extensive options for additional rules, formatting, and connectivity to additional data sources for querying and for creating business logic within your form. In the case of expense reports, you might want to query your SharePoint lists or SQL Server databases to obtain employee department and manager information automatically, mileage rates, per-diem amounts, and so on. This can easily be included to make more dynamic and flexible forms. Figure 4 shows a rudimentary expense report form. (I added a repeating table with some additional data collection to account for a varying number of expense items that can be filled out by end-users.)

| Corp Expense Report  |                      |                      |                      |
|----------------------|----------------------|----------------------|----------------------|
| Employee Information |                      |                      |                      |
| Date                 | <input type="text"/> | Department           | Select...            |
| Employee             | <input type="text"/> | Manager              | <input type="text"/> |
| Itemized Expenses    |                      |                      |                      |
| Date                 | Description          | Category             | Cost                 |
| <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> |
| Repeating Table      |                      |                      |                      |
| Submit               |                      |                      |                      |

Figure 4: Rudimentary expense report form

Repeating tables are powerful within InfoPath. They represent something that competing options don't: easily capturing any number of user-defined records at run time while still maintaining all your rules, formatting, and functions. In other applications, many designers use to collect data (e.g., Microsoft Excel, Word, ASP.NET) creating a "add another item" for the user can be very frustrating within a single form. Excel has difficulty with using functions in named ranges when it doesn't know how many rows it must account for; ASP.NET forms must create the new row in code and then be specifically built to manage the code to iterate over all the submitted record. InfoPath manages this data very easily because it knows in advance that there might be any number of

# Solve Common Business Problems with InfoPath and SharePoint

Ryan Thomas

(Reprinted from WindowsITPro Magazine)

rows within a group (XML Nodes). Most of the rules and formatting are automatically applied to each new row as it's created and filled out.

## Publish Your Expense Report

If you would like to examine your XML container file, you can use the File menu in InfoPath and choose Publish, Export Source Files. Doing so will export the InfoPath package from an XSN file into native components. If you open these files, you can begin to make some sense of how InfoPath is packaged and designed. Under the covers, it's basically an XML schema definition, XML template, an XSL stylesheet (used to create your display elements), and an associated XSF file, which is a proprietary InfoPath file used to apply the rules, datasource details, and other native functionality.

If you feel particularly daring, you can make modifications to these files manually and repackage them into your XSN file. If you ever want to venture into building InfoPath forms in Visual Studio to add compiled code, this becomes very easy. Some of the potential benefits include copying and pasting rules for faster and less risky reuse, assigning property promotion to SharePoint via specific Site Columns in advance, and understanding how rules and data sources are applied. Although this is an advanced topic, it's important to know how the files are put together in case you ever need to crack them open.

With a basic Expense Report form built, we're now ready to publish and use it inside SharePoint 2010. The first necessary action is to perform a design check to ensure that the form we've built can properly function within InfoPath Form Services. These services, native to SharePoint, can convert your InfoPath form into a fully functional web page by simply publishing the form to SharePoint. Select File, Info, Design Checker. In the Design Checker window, click the *Change settings* link at the bottom. Choose Web Browser Form in the dropdown list, then type in the URL to the SharePoint site where you wish to publish the form. Then, click OK. Select the *Verify on server* check box, and click Refresh. A dialog box will inform you that the form is being checked against your server.

The Design Checker will be your friend as you design larger and more complicated forms, up until your form gets too big—at which point, it will abruptly cease to be your friend. It will time out when trying to convert and evaluate your form to make sure it can be properly converted to a web page. This was a major problem in InfoPath 2007, and I'm hopeful that a fix is in sight. Even if the checker times out, your form will still work in SharePoint. The best option is to disable the Design Checker when it begins to time out. As long as your forms aren't very large, you'll be told if you're attempting to use any InfoPath features that aren't supported with web forms.

The next step is to select File, Submit Options, and insert the path to your document library where you want to store the filled-out forms in SharePoint. In the File Name text box, you need to set something that will enforce unique values to ensure that you don't have duplicate file names. There are many ways to do this, but for now I just used the date of the report with some text: `concat("Expense Report - ", ReportDate)`. Select Next, and choose a name for this data connection. This represents one way to quickly submit data to SharePoint, but you might find that you need to add submit buttons with more flexibility. For example, a button can run some rules and analysis on the data before allowing a submission.

# Solve Common Business Problems with InfoPath and SharePoint

Ryan Thomas

(Reprinted from WindowsITPro Magazine)

The next step is to select File, Publish, *Publish form to a SharePoint Library*. Insert the URL to your SharePoint list and select Next. On the next page, ensure that you select the *Enable this form to be filled out by using a browser* check box. Choose the Site Content Type radio button, and select Next. By selecting this radio button, you're telling InfoPath that you want this form to be added as the template for a new SharePoint Content Type. On the following screen, select *Create a new content type* and select Next. Now, add a name and description for your new content type and select Next.

On the following page, you need to decide where to place this template so that it can be used by the new Expense Report content type. I usually like to put the template in a place where it can't be easily accessed. In this example, you can hide it in the list's Forms folder so that it will reside in your new form library but will be tough to readily find. (Here's where I placed mine: <http://2k8-sp2010-dev/Expense Reports/Forms/ExpenseReport.xsn>.)

The following screen is where you'll promote local fields in your InfoPath form to SharePoint lists as columns. Click the Add button on the upper list. In the *Select a Field or Group* dialog box, select any fields in your form that you want to promote to SharePoint. In the *Site column group* dropdown list, select *(None: Create a new site column)*. Below that, create a name for your new SharePoint column. This screen will enable SharePoint to automatically extract data from your submitted forms to your SharePoint site columns with no additional effort. You will see this in action when you submit your first form shortly. Continue to add as many columns as you want. Figure 5 shows my test form.

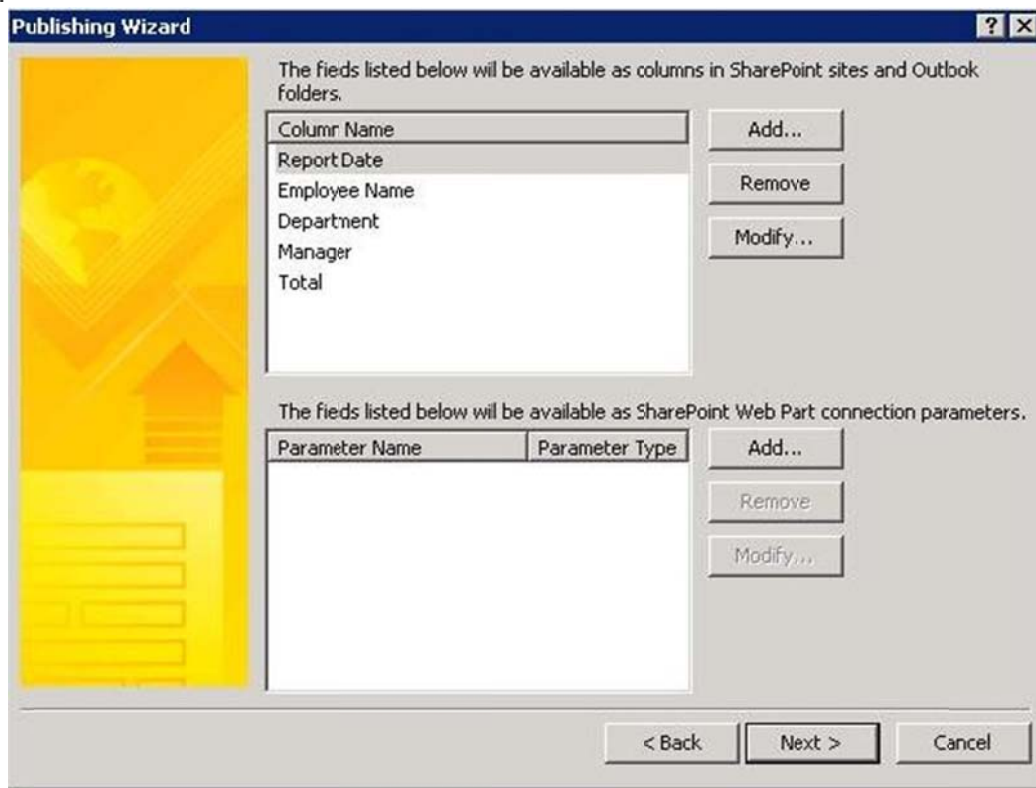


Figure 5: Sample test form

# Solve Common Business Problems with InfoPath and SharePoint

Ryan Thomas

(Reprinted from WindowsITPro Magazine)

For a more robust and reusable solution, you'll want to create your required site columns in advance within SharePoint. In that case, you can tell InfoPath to use them instead of creating new ones when publishing your form from within InfoPath. This becomes a powerful tool for collecting essential managed metadata from external sources and using custom search options in SharePoint to create a legitimate application around expense reporting or any other business process you might need to automate.

Now, you can continue through the remaining dialog boxes and publish your form. And you can browse to SharePoint to test your form. When you navigate to your form library, you must first disable the default form content type and add your new Expense Report Form content type. To do so, navigate to Library Tools, Library, Settings, Library Settings from within your Form Library. Click Advanced Settings, and ensure that the *Allow management of content types?* checkbox is selected. Then, navigate back to your Form Library Settings, and in the Content Types section, select *Add from existing site content types*, locate your new content type in the list box, and add it to your library. Return to the previous page and in the Content Types section, click *Change new button order and default content type* and clear the Form box so that it doesn't appear in the list of available content types for this library when adding items.

When you're done with that, you can see your form in action by returning to the library and selecting Library Tools, Documents, New Document, Expense Report Form from the available document types. You should see your form in your browser window, as you see in Figure 6. (I've filled out some data to show what it can look like.) At the top of your screen, you should be able to submit the form and return to your SharePoint list.

| Corp Expense Report                        |             |            |            |
|--|-------------|------------|------------|
| <b>Employee Information</b>                |             |            |            |
| Date                                       | 2/8/2010    | Department | Accounting |
| Employee                                   | Ryan Thomas | Manager    | John Doe   |
| <b>Itemized Expenses</b>                   |             |            |            |
| Date                                       | Description | Category   | Cost       |
| 2/1/2010                                   | Fuel        | Travel     | \$45       |
| 2/2/2010                                   | Lunch       | Meals      | \$23       |
| 2/3/2010                                   | Cab Fare    | Travel     | \$37       |
| <input type="button" value="Insert item"/> |             |            |            |
| <b>Total:</b>                              |             | \$105      |            |

Figure 6: Viewing the form in your browser

# Solve Common Business Problems with InfoPath and SharePoint

Ryan Thomas

(Reprinted from WindowsITPro Magazine)

## Workflow

Now that you have a working expense report form, I'll walk you through the process of creating a workflow to allow submissions of expense reports to be approved by the accounting department. You'll use SharePoint Designer 2010 Beta 2 for this task.

Open SharePoint Designer and choose to open your site via the URL. After seeing your site information page, select Workflows in the left navigation. You'll see a list of workflows that already exist in your site. SharePoint 2010 now supports modifying the out-of-the-box workflows using SharePoint Designer 2010. In this case, you'll create a brand-new workflow for your expense reports. To see some new features in SharePoint Designer 2010, select Reusable Workflow from the ribbon menu. SharePoint Designer 2010 can now create workflows that can be saved as templates and reused, or in this case attached to your expense report content type. Figure 7 shows the Create Reusable Workflow dialog box.

The screenshot shows a dialog box titled "Create Reusable Workflow". The main content area is titled "Add a new reusable workflow to your site" and includes a calendar icon. Below this, there are two sections for entering workflow details. The first section, "Enter a name and description for your new workflow", contains a "Name:" field with the text "Expense Report Approval Workflow" and a "Description:" field with the text "Approve users expense reports for accounting and reimbursement". The second section, "Pick a base content type to limit this workflow to", contains a "Content Type:" dropdown menu currently set to "Expense Report Form". At the bottom right of the dialog box are "OK" and "Cancel" buttons.

Figure 7: The Create Reusable Workflow dialog box

Select OK after you fill out the form. SharePoint Designer will connect to your server and create the shell for your new workflow. In the ribbon, select Insert, Action, Start Approval Process. You'll see your new action appear in Step 1. Click Step 1 and rename it, for example, Expense Approval—something to help denote the intention of the step. Now, click the underlined *these*

# Solve Common Business Problems with InfoPath and SharePoint

Ryan Thomas

(Reprinted from WindowsITPro Magazine)

users. In the Participants field, you can put the user or group that will be assigned the approval task. In the Title field, you can click the ellipses to build a string with some dynamic values. In this example, I allowed for a duration of two days for the task. When you're done, click OK.

Now, you can save and click Publish in the ribbon. Navigate back to your SharePoint site and to your Expense Report list. In the ribbon, click Library Tools, Library, Library Settings, Expense Report Form (*content type*), Workflow Settings, Add a Workflow. Select *Expense Report Form for the Content Type*, then select Expense Report Approval Workflow for your workflow template, and give your workflow a unique name. The rest of the settings can probably stay as defaults. I ran into what I believe to be a beta problem that required me to click the Back button after selecting my content type (in order to select my custom workflow). Remember that some of these names might be different, depending on how you name your workflow and what content type is selected. When you're finished, click OK. Your workflow is now ready to run. To test the workflow, create a new expense report and select Workflows in the item's context menu.

On the Start a New Workflow page, you can select your custom workflow and click Start. Doing so should create an approval task for your accounting user. This user will have the option to approve, reject, reassign, and so on. Assuming he or she approves your report, you'll see an Approved link in the custom workflow column in your expense report library. If you click on the link, you should see a screen similar to Figure 8.

The screenshot shows a workflow task window titled "Assign item for approval". It displays the start and completion times: "Started: 2/15/2010 3:51 PM" and "Completed: 2/15/2010 3:51 PM". Below this, there is a user icon and the name "Jill Accounting".

Below the task window is a "Tasks" section with a table of assigned tasks:

| Assigned To     | Title                                | Due Date  | Status    | Related Content             | Outcome  |
|-----------------|--------------------------------------|-----------|-----------|-----------------------------|----------|
| Jill Accounting | Expense Approval for: System Account | 2/17/2010 | Completed | Expense Report - 2010-02-08 | Approved |

Below the tasks section is a "Workflow History" section with a table of events:

| Date Occurred     | Event Type         | User ID         | Description  | Outcome  |
|-------------------|--------------------|-----------------|--|--|
| 2/15/2010 3:51 PM | Error              | System Account  | The e-mail message cannot be sent. Make sure the e-mail has a valid recipient. |  |
| 2/15/2010 3:51 PM | Error              | System Account  | The e-mail message cannot be sent. Make sure the e-mail has a valid recipient. |  |
| 2/15/2010 3:51 PM | Workflow Initiated | System Account  | Expense Report WF Test was started. Participants: Jill Accounting              |  |
| 2/15/2010 3:51 PM | Task Created       | System Account  | Task created for Jill Accounting. Due by: 2/17/2010 3:51:37 PM                 |  |
| 2/15/2010 3:52 PM | Error              | System Account  | The e-mail message cannot be sent. Make sure the e-mail has a valid recipient. |  |
| 2/15/2010 3:52 PM | Task Completed     | Jill Accounting | Task assigned to Jill Accounting was approved by System Account. Comments:     | Approved by System Account   |
| 2/15/2010 3:52 PM | Workflow Completed | System Account  | Expense Report WF Test was completed.  | Expense Report WF Test on Expense Report - 2010-02-08 has successfully completed. All participants have completed their tasks. |

Figure 8: Approving the report

# Solve Common Business Problems with InfoPath and SharePoint

Ryan Thomas

(Reprinted from WindowsITPro Magazine)

This screen contains a lot of information, including any tasks created during this workflow, plus all the history data logged by the workflow process. There are some errors logged in my history because I didn't get email correctly configured in my test environment. You'll also notice another new feature at the top of the image: Visio Services. SharePoint Designer 2010 workflows can be exported and imported via Visio to provide a more robust and familiar design surface for workflow creation. These services automatically created a flowchart to depict my custom workflow. You'll see the benefits of this feature as your workflows become more complex with additional steps and business logic.

This is an example of the most basic workflow. I don't have enough space to go into more depth, but if you look through the available actions for workflow steps, you'll get an idea of what's available. You can also custom-build these actions in Visual Studio and use them in SharePoint Designer. Workflows can make decisions based on user input, look up data from the current item or SharePoint lists, and span multiple users. They are capable of significant complexity.

## Extremely Powerful

Although this is a quick and relatively simple display of the technology, you can see how easy it is to build custom forms with workflow, and you can get a sense of the additional power and flexibility at your fingertips. Combining workflow, some custom code behind the forms, robust rules, datasource connectivity, and the ability to package and deploy forms for re-use across multiple locations, InfoPath is an extremely powerful tool in conjunction with SharePoint. I should also point out that all the form-development functionality is similar to that of InfoPath 2007 and Microsoft Office SharePoint Server 2007. Although some of the screen layouts are different, everything we walked through can also be built in the current version of these products. Only the workflow components at the end have changed significantly in the 2010 release.