

Issue

7

UNDERSTANDING SHAREPOINT JOURNAL

Bjørn Furuknap

Introducing SharePoint Visual Studio Workflows

UNDERSTANDING SHAREPOINT JOURNAL

Introducing SharePoint Visual Studio Workflows

This book is dedicated to my wife.

© Understanding SharePoint
Rubina Ranasgt. 10
N-0190 Oslo, Norway
Phone +47 91 39 85 86 • Web <http://www.understandingsharepoint.com/>

Credits

About the Author



Bjørn Christoffer Thorsmæhlum Furu knap is a senior solutions architect, published author of *Building the SharePoint User Experience*, speaker, and passionate SharePointaholic. He has been doing software development professionally for 16 years for small companies as well as multinational corporations. He has also been a teacher at a college-level school, teaching programming and development to aspiring students, a job that inspired him to begin teaching what he has learned and learns every day.

About *Understanding SharePoint Journal*

Understanding SharePoint Journal is a periodical published by UnderstandingSharePoint.com. The journal covers few topics in each issue, focusing to teach a deeper understanding of each topic while showing how to use SharePoint in real-life scenarios.

You can read more about *USP Journal*, as well as get other issues and sign up for regular updates, discounts, and previews of upcoming issues, at <http://www.understandingsharepoint.com/journal>.

Other Credits
















A great big thanks to Kim Wimpsett for doing the copyedit. The quality of work in this issue is greatly attributed to her skill.

Table of Contents

Credits.....	i
About the Author	i
About <i>Understanding SharePoint Journal</i>	i
Other Credits.....	i
Table of Contents	i
Exercises	2
Introduction	4
Introduction and Setup	1
Workflow in a Nutshell.....	1
So, What Is a Workflow Anyway?	2
Sequential and State Machine Workflows	3
Workflow as a Business Tool.....	4
Installation and Setup.....	5
Introducing Visual Studio Workflows	9
Welcome to Visual Studio	9
The Visual Studio Stage	14
Removing MOSS Dependencies	19
Testing Your Workflow	22
Adding the First Action	25
Controlling the Flow	33
Composite and Basic Activities	33
Conditions	34
Creating a Conditional Activity	35
Beautifying the Workflow	40
Testing the Condition	42
Exploring Conditional Activities.....	43
Workflow Loops	44
Testing the Loop	48
Multiple Activities in a While Loop?	51
Reusing Workflows	55
Deploying the Workflow	55
Is There a Solution?.....	56
Reusing Workflow Components.....	61
Waiting for Business Hours Workflow	62
Re-using a Workflow	67
Re-using a Custom Activity	72

State Machine Workflows	77
State Machine Is a What?	77
Creating a Visual Studio State Machine Workflow	80
Working with States	82
DataEntryState.....	85
ApprovalState	87
Finalizing the Workflow	93
Testing the State Machine	96
Final Thoughts and Additional Resources	99
Previous <i>USP Journal</i> Issues	100
Issue 1: SPCurrentUsers Explained.....	100
Issue 2: Developing SharePoint Content Types.....	100
Issue 3: SPTags Explained	100
Issue 4: SharePoint Designer Workflow.....	101
Issue 5: Beginning SharePoint Development.....	101
Issue 6: SPThemes and SPTags Explained	101
Special Issue 1: Using Nintex Workflow 2007.....	101
<i>USP Journal</i> Affiliate Program.....	103

Exercises

 Creating a Visual Studio Workflow	9
 Fixing MOSS Dependencies in Workflow	19
 Testing a Workflow.....	22
 Adding Actions to Sequential Workflow	25
 Using an If/Else Activity.....	35
 Using a While Activity.....	45
 Utilizing the Sequence Activity	51
 Setting Up WSPBuilder	57
 Attaching a Workflow to a List.....	60
 Creating a Delay Workflow	62
 Creating a Custom Workflow Activity	68
 Adding a Custom Activity	72
 Building a State Machine Workflow	80
 Configuring the DataEntryState.....	85
 Configuring the ApprovalState	87

Introduction

Asking the right question at the right time to the right people can trigger a landslide.

Welcome to the third issue of the Business Process Management series of *Understanding SharePoint Journal*. Previously, we explored SharePoint Designer workflows in issue 4 and Nintex Workflow 2007 in the first free, special issue.

This time, we will tackle a much more complex workflow authoring tool, Visual Studio (VS). VS is often considered a programmer's tool, but in this issue, I will introduce you to authoring workflows without having to use code. Or at least, you won't need much code.

Why use a code-generating tool like Visual Studio to create low-code workflows? Well, I want to introduce you to the tool and the method without forcing you to adopt or learn .NET programming. I realize this will limit the options we have slightly, and you won't see the full potential of Visual Studio, but on the other hand, you won't drown in class declarations and delegate methods either.

Although this issue focuses on low-code Visual Studio workflow authoring, I am in no way recommending that you should avoid code in your VS workflows. The main goal here is to introduce you to the power of Visual Studio, even if you have no programming experience.

The examples we will develop in this issue may seem basic, but I have chosen these examples because you may already be familiar with the logic of the workflow. That way, you can focus on learning how you can use Visual Studio to implement known logic rather than having to learn completely new workflows from scratch.

With that said, let's get rolling.

Introduction and Setup

Ready, steady, flow!

Me: Hello!

You: Hello!

Me: How are you?

You: Fine, how are you?

Me: Fine.

You may not realize this, but you are already using workflows. Every day you perform some action or actions that follow a more or less predefined pattern or flow. Even a simple greeting follows a certain flow—things we say not because we are so darn interested in how the other person is but rather because that's just how it flows.





In this chapter, I will introduce you to workflows as they are used in the IT world and attempt to explain to you how powerful workflows are as tools for optimizing business processes. Then, I'll show you how to set up your environment for developing workflows in Visual Studio.

And the best part is, it won't cost you a dime in software.

Workflow in a Nutshell

Workflows saw a rise in popularity after Microsoft started marketing Windows Workflow Foundation as part of the .NET Framework. Released as part of .NET 3.0 in 2006, workflows quickly became an important part of business applications even for smaller businesses and projects. These days, it seems you can't walk down the street without stepping into one or more workflows.

ICON KEY

	Valuable information
	Test your knowledge
	Exercise
	Caution

Note

Although initially marketed as WWF, the marketing guys and gals in Redmond got a visit from some really huge and scary men wearing funny clothes and masks and sporting names like Hulk Hogan and The Undertaker and decided that they might as well use the acronym WF.

Either that, or they realized they didn't need the connection to pandas and other threatened species. In any case, I will be using the official acronym WF throughout this journal.

So, What Is a Workflow Anyway?

Workflows are just what their name implies, flows of work. Think of any process where some set of actions is performed, such as buying an item in a store, taking a shower, filing a letter, or even dating someone special. All of these tasks are composed of tasks or states where some action or event is expected.

Let's take a look at one of them, buying an item in a store.

When you buy an item, you need to do the following:

- Find the correct item.
- Take the item to the counter.
- Receive the price from the attendant.
- Decide between using cash or using a credit card.
 - If using cash:
 - Give appropriate amount to the attendant.
 - Receive any change.
 - Verify change.
 - Pocket change.
 - If using a credit card:
 - Give credit card to the attendant.
 - Receive receipt for signature.

- Verify amount.
 - Sign receipt.
 - Return receipt to the attendant.
 - Put pen in pocket.
- Retrieve item, and run before the attendant notices the missing pen.

Don't be fooled. This task may seem simple, but that's only because we do it so often that it has become routine. Several of these steps require a number of substeps, and any failure will need to be handled according to some set of rules.

A workflow structures such processes into manageable chunks called *activities*. Each activity performs one or more small operations, such as verifying an amount or stealing a pen, and the workflow makes sure that everything is done in order and handles exceptions to the regular flow.

Sequential and State Machine Workflows

Workflows come in two varieties, as either a sequential workflow or a state machine workflow. Although they're two different terms, the two forms are closely connected.

The simplest form is the *sequential workflow* in which several actions occur in a predefined sequence. You can use conditions to branch the workflow, but in the end it is just a matter of going from one end to the other in order.

We will explore sequential workflow in Chapters 2 through 4.

A *state machine workflow* is a bit more complex, but if you think of state machine workflows as collections of sequential workflows, they may be easier to understand. Each state in a state machine is really just a separate sequential workflow. Either during or at the end of each such subworkflow, you can enter another state and thus trigger a second sequential workflow.

SharePoint Designer workflows are sequential workflows, while you can design whichever type you like when using Visual Studio.

Another difference is that although a sequential workflow always ends, a state machine workflow may keep going indefinitely, at least in theory. Think of the workflow to have someone exchange the calendars in an office at the end of the year—unless time itself stops or we stop using calendars, that workflow will keep going forever.

State machine workflows are particularly well suited for workflows that require human interaction. The reason is that humans rarely behave in a strictly sequential manner and often jump in another direction than the designer of the workflow anticipated.

State machine workflows can cope with this situation much better than sequential workflows because state machines by design encourage jumping between different states depending on the outcome of other states.

I'll show you how to create a state machine workflow in Chapter 5.

Workflow as a Business Tool

Most organizations have formal processes that define how they want to perform business tasks. Examples of such tasks are document and records management, customer follow-up, employee training, knowledge management, website content management, and so on.

I'm not going to market using workflows to you, since, reading this issue, you obviously want to learn how to create workflows. However, I would like to give you a couple of examples of how you can save your organization money by applying your new skills.

Let's say that you have a simple process, something like adding new customers to a customer database and following up with that customer after a few weeks.

Let's further assume that someone in your organization currently spends two minutes following up with one new customer by writing a standard email, asking the customer if they are happy with your organization's service.

That's not a lot of time, right? But then again, copying a standard email and entering a customer's email address in Outlook doesn't take that long.

Now, let's further assume you get 10,000 new customers each year. Even if there were absolutely no overhead on the two minutes required to follow up a customer, that's still 20,000 minutes, or 333 hours and 20 minutes. At a salary rate of \$50 per hour, that's \$16,000 per year.

Writing a simple workflow that automates this process may take you a day; add a few days for testing, and you have saved your organization more than \$15,000 every year.

Let's look at something else—content approval. Employees want to travel, but sadly, not every trip to Hawaii is beneficial to the company, and a manager may need to check and approve new travel requests.

Not just that, but once the request is approved, a lot of events need to happen. The manager needs to notify the employee about the approval, the HR department needs to be notified as well, and perhaps the employee even needs reimbursements of expenses.

With a workflow, you can automate many of these tasks. The employee may fill out a form for a travel request, including the expected cost and any expected expenses to be reimbursed. Once the employee hits Send, the workflow notifies the manager and requests an approval.

When the manager sees the request, the workflow may already have performed sanity checks on the data. For example, asking for a too-expensive plane ticket may automatically decline the request.

Regardless of automatic or manual denial of the request, the workflow can ask the employee to revise the request, perhaps booking a less expensive ticket or one for a different week.

After the manager decides whether to approve the request, the workflow can automatically notify the employee and let the HR department know that you will be away on ~~vaca.~~ a business trip that week. Sending a notification to Exchange can make sure your status is set to Busy while you're away.

Depending on how far you want to take this, you can add all sorts of automation as part of a workflow. Your expenses can be automatically transferred to your account; your plane tickets can be booked; your travel arrangements can be arranged; you can send an email to your spouse, telling him or her to remember the sunscreen; and, heck, you can even program your Roomba to clean the place before you get home.

All of this happens just by filling out a simple form and clicking one button.

Enough dreaming—let's get started and see whether we can make some of these dreams come true.

Installation and Setup

If you have never developed anything on SharePoint before, first, let me congratulate you on being very brave. Starting your first development project on workflow is a daunting task.

I will assume, however, that you are well versed in using SharePoint, since I won't be explaining anything about lists, list items, sites, and such.

Second, I won't show you how to set up your lab server or SharePoint. Or, more accurately, I'll show you, but not in this issue.

Setting Up a Lab Machine

If you need help getting your lab machine set up, check out the first chapter of the “Beginning SharePoint Development” issue, available as a free download on <http://www.beginningsharepointdevelopment.com/>.

Also, refer to the videos on the *USP Journal* YouTube channel at <http://youtube.com/uspjournal>.

Third, you will benefit greatly if you have some experience with workflow, either from authoring workflows in SharePoint Designer or even from just using workflow. Throughout this issue, I will re-create in Visual Studio some workflows that are common examples in SharePoint Designer tutorials.

Yes, there will be a notify administrator workflow.

Learning SharePoint Designer Workflow

If you want to learn more about SharePoint Designer workflow, check out issue 4 of *Understanding SharePoint Journal*, available at <http://www.sharepointdesignerworkflow.com/>.

Technical Requirements

The technical requirements for this issue are the same as for the “Beginning SharePoint Development” issue. In short, this means you must have the following:

- Windows Server 2003 or 2008, preferably running in a virtual machine. I’m using 2003.
- SQL Server 2005 or 2008.
- SharePoint, either WSS or MOSS, although we won’t need MOSS at all.
- Visual Studio 2008 Professional. You need the professional edition.

I also highly recommend downloading and installing WSPBuilder. Although we won’t be using WSPBuilder most of the time, install it anyway. You’ll still save tons of time, and you’ll thank me somewhere in Chapter 4.

All the software that you need for the exercises in this issue is available free of charge or as trial versions. Below, I have added links to the various sites providing downloads.

Resource Links

Virtual PC

<http://www.microsoft.com/downloadS/details.aspx?FamilyID=04d26402-3199-48a3-afa2-2dc0b40a73b6&displaylang=en>

VMware Workstation evaluation

<http://vmware.com/products/ws/>

Visual Studio 2008, Professional Edition evaluation

<http://www.microsoft.com/visualstudio/en-us/try/default.aspx>

Windows SharePoint Services with Service Pack 2, Infrastructure Update

<http://technet.microsoft.com/en-us/windowsserver/sharepoint/bb400747.aspx>

WSPBuilder

<http://www.codeplex.com/wspbuilder>

SQL Server 2005/2008 Trial

<http://www.microsoft.com/sqlserver/2005/en/us/trial-software.aspx>

<http://www.microsoft.com/sqlserver/2008/en/us/trial-software.aspx>

Windows SharePoint Services SDK

<http://msdn.microsoft.com/en-us/library/ms441339.aspx>

Installing Visual Studio

When installing Visual Studio, follow the normal spouse-mode installation (Yes, Yes, Yes, Yes), but make sure that you include the Visual Studio Tools for Office. You will find these under both the Visual Basic node and the Visual C# node.

These tools provide two very useful templates for creating workflows in SharePoint: the SharePoint 2007 Sequential Workflow and the SharePoint 2007 State Machine Workflow templates. We will look at both of these throughout the issue.

Creating a New Team Site

After you have set up your SharePoint installation, including Visual Studio 2008, make sure you create a Team Site that we will use as the lab site for our workflows. Actually, you can create pretty much any site you like, but the default Team Site gives you a few lists and libraries we want to use.

Now that you have met the requirements, meaning you have installed Windows 2003 or 2008 Server, SQL Server, SharePoint, and Visual Studio 2008 and you have set up your lab Team Site, I think it's time to get started.

Oh, you haven't done all that yet? Well, now would be a great time. Go ahead, get the first chapter of "Beginning SharePoint Development" and build your lab environment.

I'll wait right here.

Introducing Visual Studio Workflows

Get your feet moist.

Let's start with the simple stuff, shall we? In this chapter, I'll walk you through creating a very simple workflow that will notify the administrator of a site whenever someone uploads a new document.

Yeah, I know, I'm always using this as an example, but it is one of the simplest workflows I know. The point of this chapter is to introduce you to the Visual Studio stage and show you the basics of workflow authoring.

And we'll stay away from writing code. You'll have to delete a couple of lines of code if you don't have MOSS installed on your lab computer, but there's no code writing.

Welcome to Visual Studio

Your first task, should you choose to accept it, is to start Visual Studio. If you do not choose to accept the task, feel free to have someone else do it for you, but you need to have Visual Studio started before you proceed.

 Creating a Visual Studio Workflow

Note

When starting Visual Studio for the first time, you may be asked for your preferred development setup. Just select any option if you've never worked with Visual Studio before, or select your preference if you have one.

I'll walk you through all the various areas of Visual Studio throughout this chapter, so for now, let's just get our first project up and running.

Click the link to create a new project, shown in Figure 1.

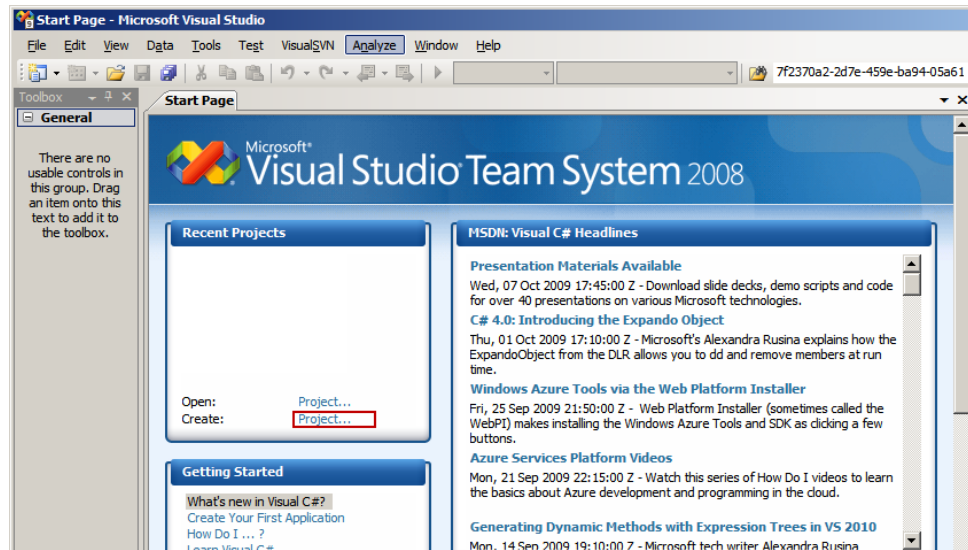


FIGURE 1. VISUAL STUDIO START PAGE

Visual Studio allows you to create a wide range of projects, but in our case, we are mostly interested in the workflow project types, so select the Workflow node in the left tree.

On the right side of the New Project dialog box, you can select multiple workflow projects, but the project type that's easiest for now is the SharePoint 2007 Sequential Workflow project, shown in Figure 2.

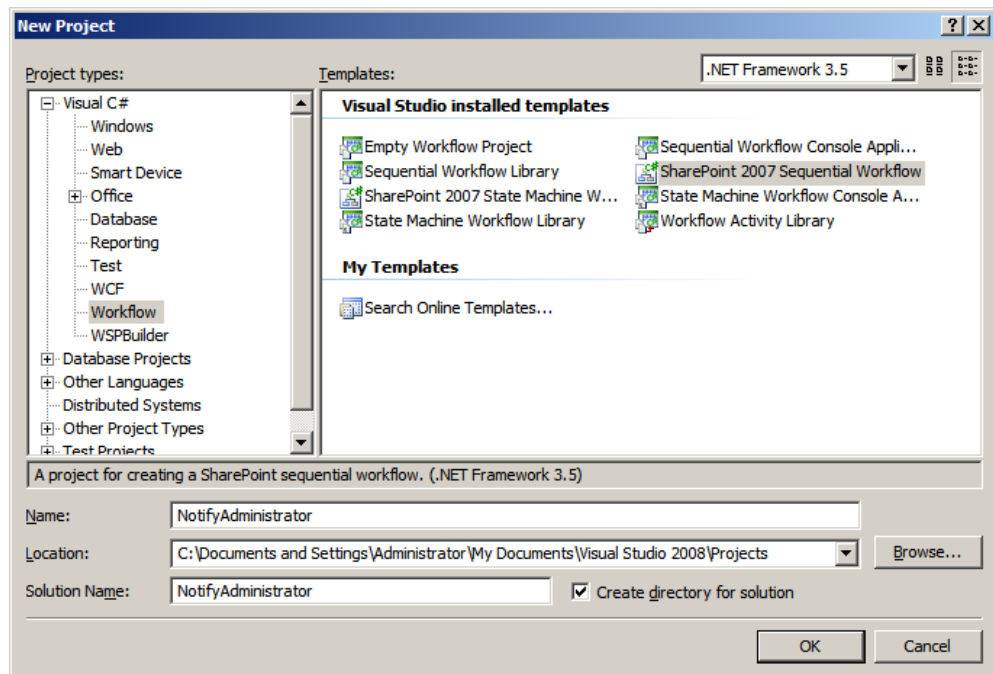


FIGURE 2. NEW PROJECT DIALOG BOX

Enter a name, such as **NotifyAdministrator**, and hit OK to create the workflow project and start the New Office SharePoint Workflow Wizard, shown in Figure 3.

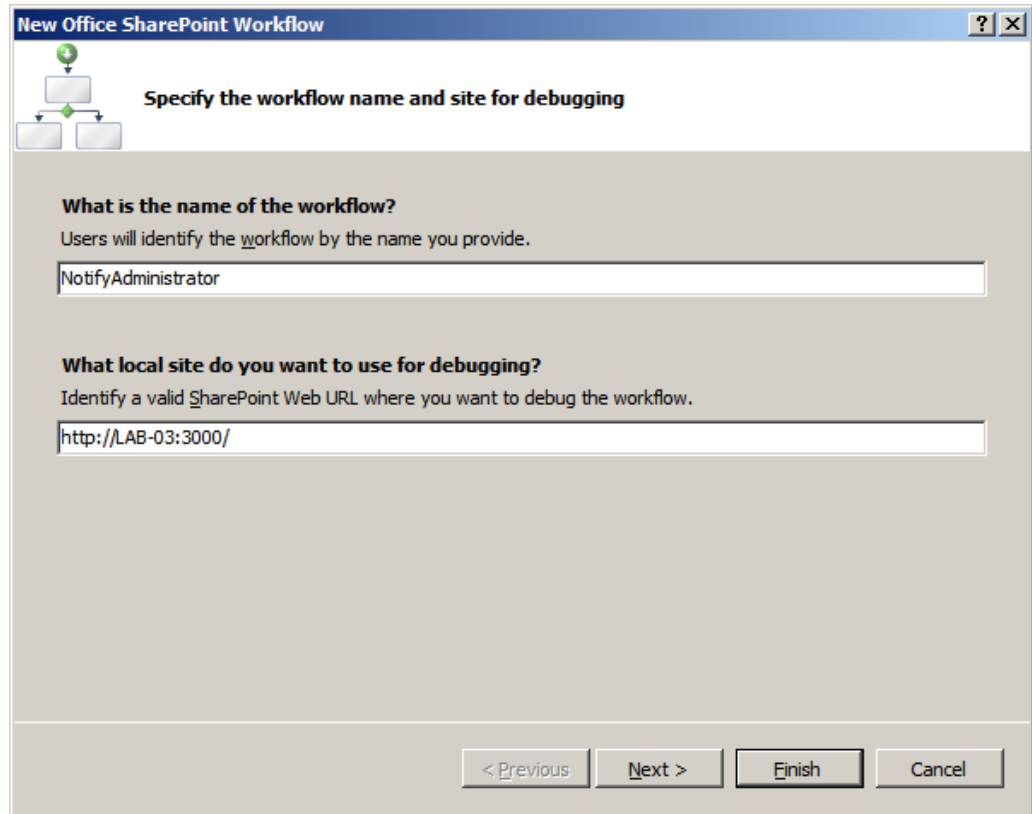


FIGURE 3. NEW OFFICE SHAREPOINT WORKFLOW WIZARD, STEP 1

Yes, I know I said we will avoid the MOSS features, but that’s the name of the wizard. Don’t worry. Even though the Visual Studio project will try to trick us into creating a MOSS workflow, we have our own tricks up our sleeves.

The first step of the wizard is where you set the name of the workflow and select which site you want to use for testing and debugging. Enter any name you like and enter the URL of the site you created in Chapter 1 before hitting Next to move to step 2.

In step 2, you set up some parameters for how you want to debug your workflow. For our purpose, leave the “Automatically associate workflow?” checkbox on, and select Shared Documents in the Library or List drop-down, as shown in Figure 4.

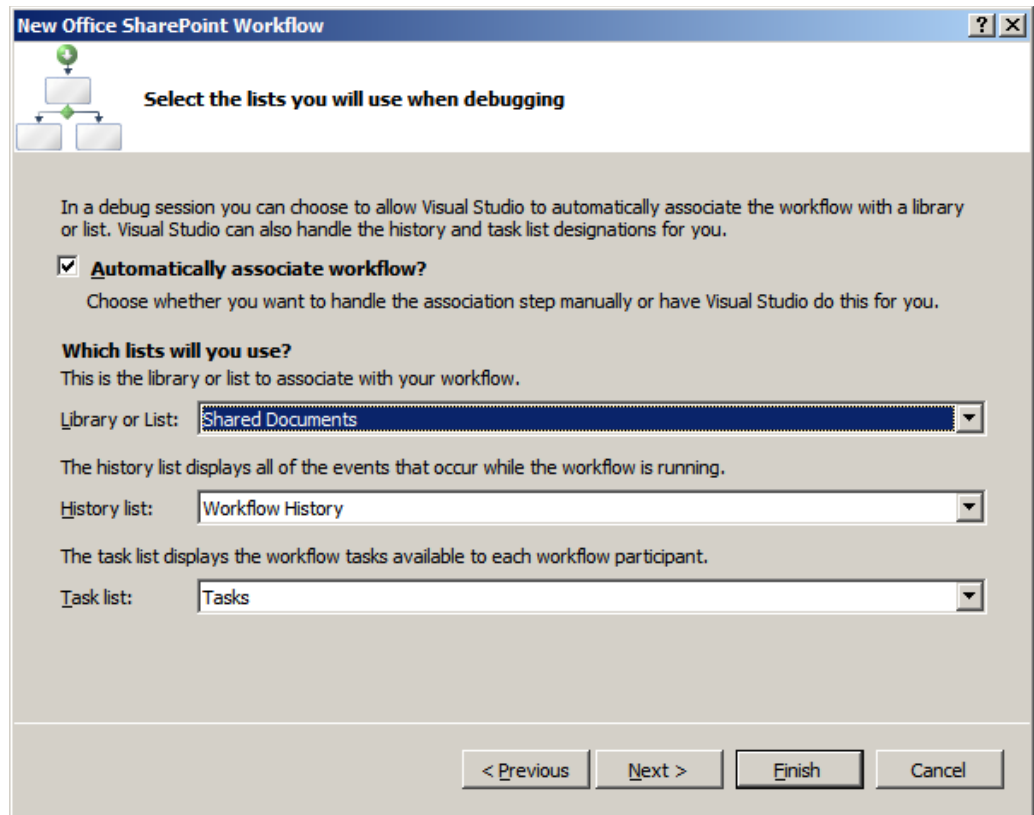


FIGURE 4. NEW OFFICE SHAREPOINT WORKFLOW WIZARD, STEP 2

Note that, since we are developing a Visual Studio workflow, these parameters are for testing only. When you deploy your full workflow later, you can attach the workflow to any list or library you want, and you can also set both the History list and the Task list when you attach the workflow to that list or library.

Note

One of the major benefits of Visual Studio workflows over SharePoint Designer 2007 workflows is the ability to reuse workflows. This feature becomes available in SharePoint Designer 2010.

For now, leave the default values for the History list and Task list and hit Next.

Finally, you need to set the start options for the workflow. The options available are to start the workflow manually or automatically when someone or something creates or changes an item.

Select the “Manually by users” and “When an item is changed” options, as shown in Figure 5 and hit Finish.

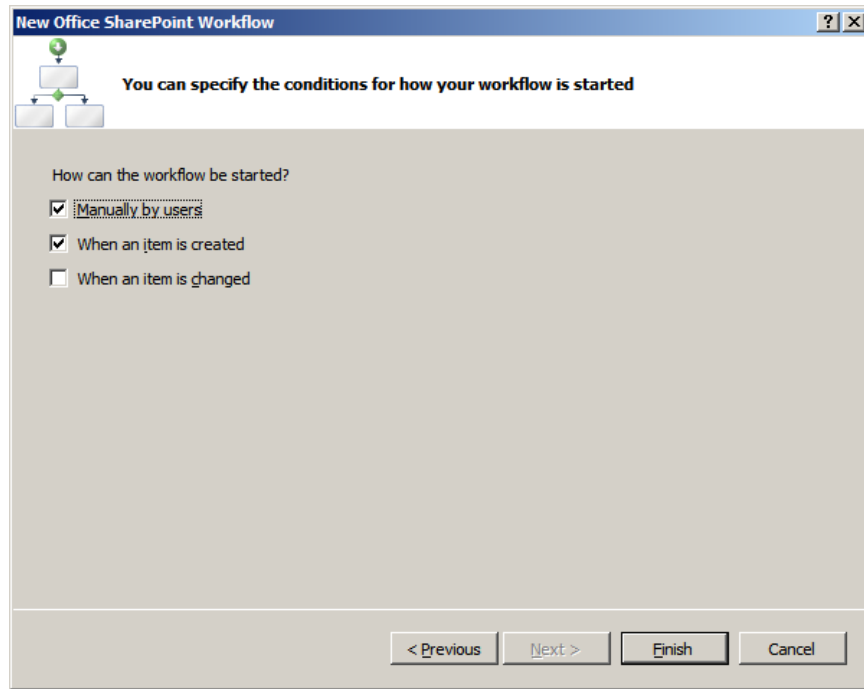


FIGURE 5. NEW OFFICE SHAREPOINT WORKFLOW WIZARD, STEP 3

After a few seconds, Visual Studio has created the project and set it up for debugging as you instructed. Congratulations, you have created your first Visual Studio workflow, as shown in Figure 6.

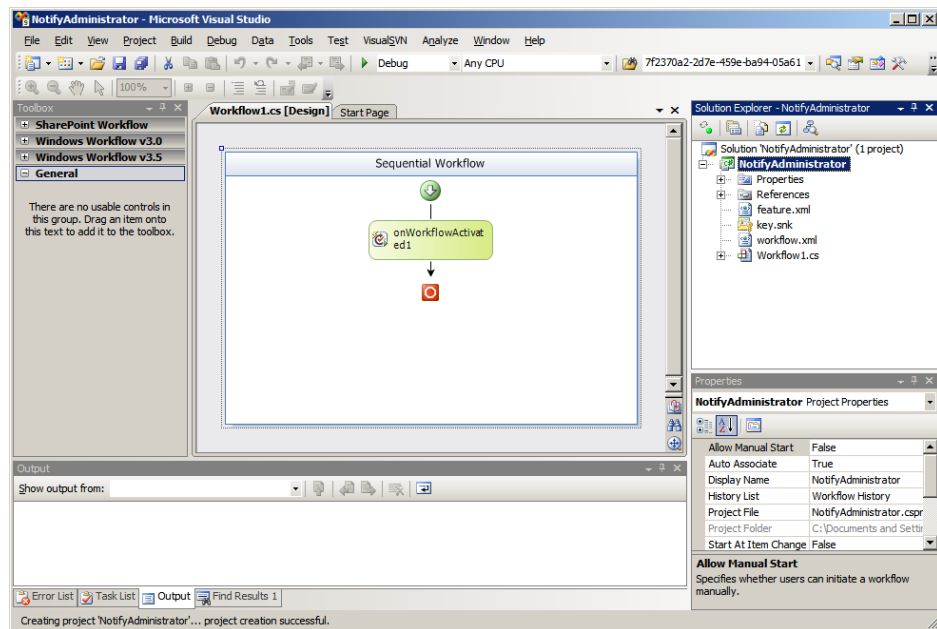


FIGURE 6. WORKFLOW PROJECT CREATED

Oh, OK, perhaps the workflow isn't doing anything at the moment, but you do, in fact, have a fully working Visual Studio workflow.

Let's walk through the Visual Studio interface and learn what the various panes and areas do.

Before we do, note that you can restart the New Office SharePoint Workflow Wizard to change your selections at any time. To do so, right-click the project in the Solution Explorer, and select SharePoint Debug Settings, as shown in Figure 7.

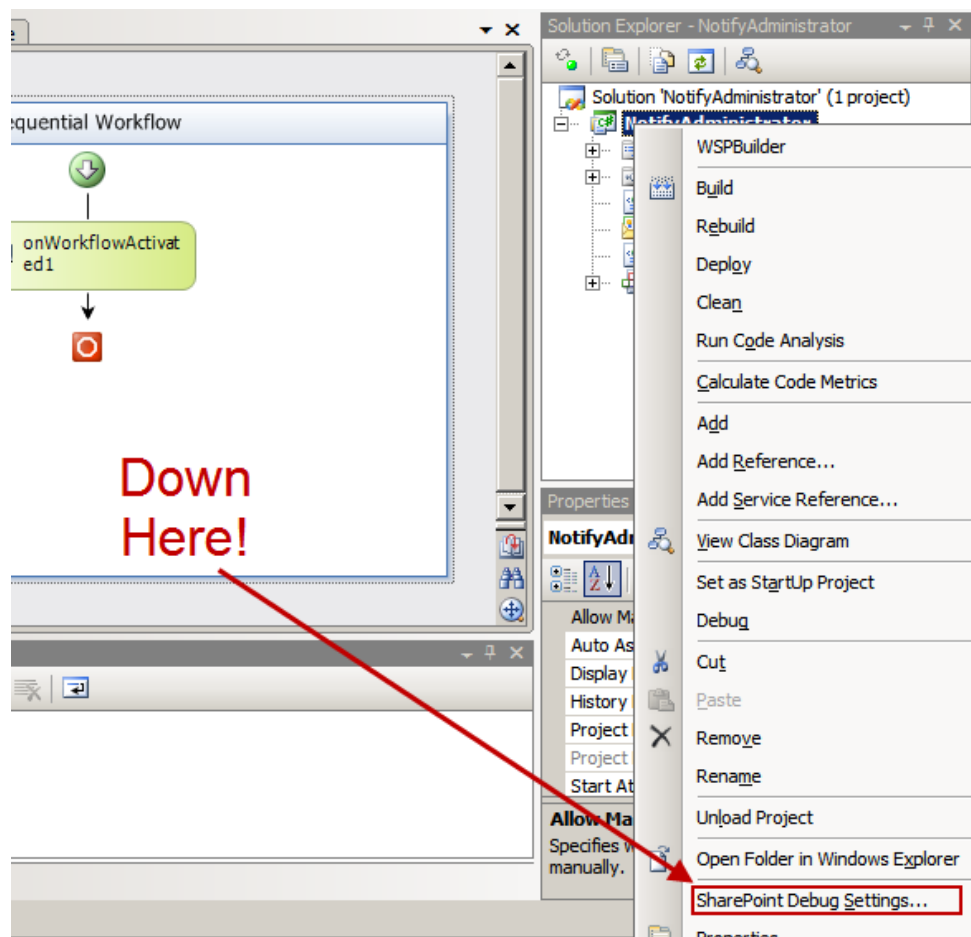


FIGURE 7. SHAREPOINT DEBUG SETTINGS

The Visual Studio Stage

This was actually a nice segue into the next topic, introducing Solution Explorer like that. I need to make a note to use that in another setting as well.

In any case, there are five areas, or *panes*, that you want to remember for the duration of this issue. I'll introduce you to all of them now.

Note

If you ever close any of the panes, you can always reopen them from the View menu.

Solution Explorer

Solution Explorer, shown in Figure 8, is usually located on the right side of your screen; it is where you get an overview of the content of your solution and projects. You'll find a lot of files there that you will likely never touch, but you'll also find all the artifacts of your projects there, such as the workflows you will develop throughout this issue.

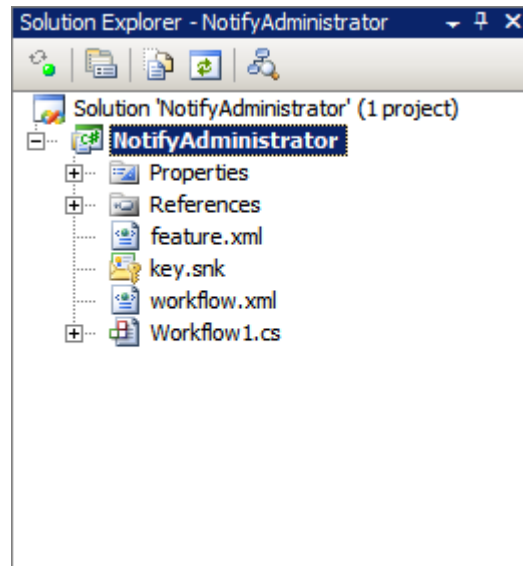


FIGURE 8. SOLUTION EXPLORER

Note

In Visual Studio, a *solution* contains one or more *projects*. Each project is independent of the other projects and can be built and deployed as isolated units.

You can add multiple projects to a solution and thus create, for example, multiple workflow projects in the same solution. However, you do not need to add multiple projects to add more workflows.

Properties

The Properties pane is where you set or view properties for various artifacts. The Properties pane is highly contextual, meaning you get different properties depending on which object you select.

For example, if you select the workflow project, you get a list of the settings for the entire project, as shown in Figure 9. If you select an action, which we will do in a bit, you get the properties of only that single action.

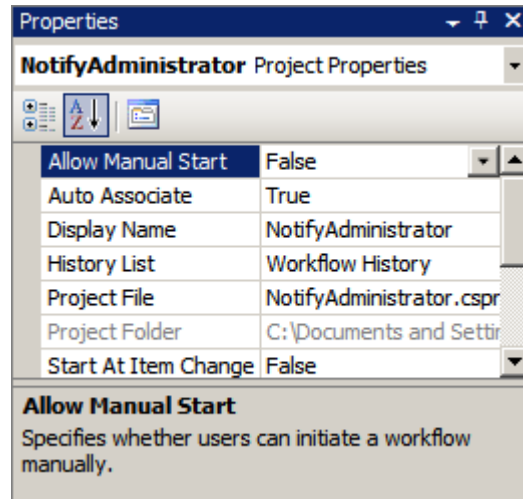


FIGURE 9. PROPERTIES PANE

The properties have different edit modes, depending on the possible values for the property. For example, for the Allow Manual Start property for the workflow, shown in Figure 9, the drop-down will display True or False only since these are the only legal values.

Workflow Designer Surface

When you open a workflow in Visual Studio, the center editor pane changes to the workflow designer surface, as shown in Figure 10.

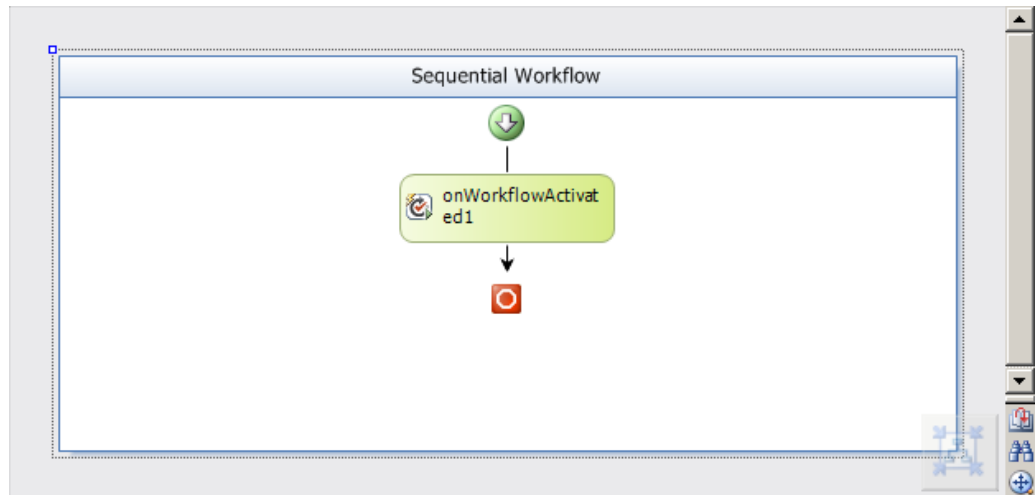


FIGURE 10. WORKFLOW DESIGNER SURFACE

The workflow design surface is where you will design your new workflow by adding actions, branches, conditions, and other activities that represent the business process you want to automate.

For our workflow, we have a sequential workflow designer. The workflow designer surface looks slightly different when working with state machine workflows, but the principles are the same.

Note

I'll cover state machine workflows in Chapter 5.

The editor pane also holds many other types of editors, depending on what type of file you open.

By default, as you can see in Figure 10, Visual Studio adds a single activity to our SharePoint workflow, the `onWorkflowActivated1` activity. All Visual Studio SharePoint workflows begin with this activity, which handles activities related to starting the workflow.

You will learn more about this and other event activities in later chapters.

Toolbox

At the very left, usually, you will find the Toolbox. This is where all your draggable tools reside, as shown in Figure 11.

End of Preview

But not the end of the issue

I hope you have enjoyed this small preview of *Introducing SharePoint Visual Studio Workflows*. To get the full issue for just \$14.95, visit the web page at <http://www.sharepointvisualstudioworkflows.com/> or buy directly from this link:

<http://www.understandingsharepoint.com/url/uspj/0107>

Previous *USP Journal* Issues

What you have missed so far but can still get from our website.

Issue 1: SPCurrentUsers Explained

This issue deals with the solution SPCurrentUsers available from CodePlex and covers the following main topics:

- DelegateControls
- CustomAction, both CAML-based and Assembly-based
- Custom application pages in SharePoint, using code-behind files
- List deployment and column manipulation
- Solution setup using SharePoint event receivers

URL: <http://www.understandingsharepoint.com/journal/volume-1/issue-1>

Buy now!

Issue 2: Developing SharePoint Content Types

If you want to learn everything about developing SharePoint content types, you should get your hands on issue 2, covering the following:

- Content type introduction
- Visual interface using FormTemplates
- Inheritance
- Behavior using event receivers
- Content types as folders
- Custom XmlDocument solutions

URL: <http://www.understandingsharepoint.com/journal/volume-1/issue-2>

Buy now!

Issue 3: SPTags Explained

If you have ever tried to build custom field types in SharePoint, you definitely want to pick up issue 3, covering the following:

- Custom field type development
- SharePoint web part development
- Custom page development
- Custom web part properties
- Field type custom property storage

URL: <http://www.understandingsharepoint.com/journal/volume-1/issue-3>

Buy now!

Issue 4: SharePoint Designer Workflow

In issue 4 of *Understanding SharePoint Journal*, you will learn how to create workflows in SharePoint Designer. The issue covers the following:

- Workflow in a nutshell
- SharePoint Designer workflow activities
- Workflow branching
- Working with variables
- Loops in SharePoint Designer workflows

URL: <http://www.understandingsharepoint.com/journal/volume-1/issue-4>

Buy now!

Issue 5: Beginning SharePoint Development

Issue 5 introduces you to development for SharePoint. You will learn the basics of various SharePoint development techniques, including the following:

- Setting up your development lab
- Using Collaborative Application Markup Language
- Using the SharePoint object model
- Creating features and solutions from scratch
- Using event receivers and workflow
- Working with web parts

URL: <http://www.understandingsharepoint.com/journal/volume-1/issue-5>

Buy now!

Issue 6: SPThemes and SPTags Explained

Understanding SharePoint Journal presents two new solutions, SPThemes and SPSampleData, designed to teach you new aspects of SharePoint development. In this issue, spanning 90 pages, you will learn the following aspects of SharePoint development:

- DelegateControl development
- CustomAction
- User information list management
- Custom property storage using extension methods
- Programmatic list creation
- Custom application page development

URL: <http://www.understandingsharepoint.com/journal/volume-1/issue-6>

Buy now!

Special Issue 1: Using Nintex Workflow 2007

The first *free* issue of *Understanding SharePoint Journal* covers Nintex Workflow 2007. In this 128-page special issue of *Understanding SharePoint Journal*, I will introduce you to Nintex Workflow 2007 and to the following:

- How to install, configure, and test Nintex Workflow 2007
- How to use the Nintex workflow designer
- How to write completely custom workflows
- How to use and modify workflow templates and snippets
- How to use both sequential and state machine workflows, including combining the two
- How to develop a complete workflow solution to send “Get well soon!” greetings to sick employees
- Some of the gotchas, do’s and don’t’s, and problems you may encounter
- Some best practices and advice on monitoring and managing workflows

URL: <http://www.learnnintex.com/>

USP Journal Affiliate Program

Want to promote USP Journal— and make a buck?

The *USP Journal* affiliate program is your chance to earn money by promoting *USP Journal* to your friends, readers, colleagues, or others who are interested in learning more about SharePoint.

Here are four great reasons to join the *USP Journal* affiliate program:

- You'll earn a generous share of sales just by recommending to others what you like yourself.
- You'll get free tools to help you refer more readers, including free advance previews of upcoming issues and excerpt content to use on your own site or blog.
- High-quality products mean high conversion rates and few returns. And, most importantly, they mean happy readers.
- You'll get the safety and ease of mind from regular payments, handled by E-junkie and PayPal.

To learn more about the program, visit
<http://www.understandingsharepoint.com/affiliates>.