Issue

4

UNDERSTANDING SHAREPOINT JOURNAL

Bjørn Furuknap

# SharePoint Designer Workflows

UNDERSTANDING SHAREPOINT JOURNAL

# BPM in SharePoint: SharePoint Designer Workflows

*This book is dedicated to my wife.*


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 Furuknap 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

# Exercises



# Legal Disclaimer

# Introduction

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

(Not completed)

Chapter 1

# Workflow in a Nutshell

*Understanding the path you walk.*

Welcome to one of the most powerful features of SharePoint; workflow. Inside that small word lies a world of opportunity, and perhaps the single most valuable piece of technology currently available. Workflow bears the promise of business process optimization beyond the wildest dreams of anyone.

Ok, I'll stop the marketing talk, we're not here to learn how to market, we are here to understand SharePoint and increase our knowledge of workflow in SharePoint Designer. By that virtue alone, we will become all the marketing required.

Before we can commence any investigation into SharePoint Designer, we need to understand some key concepts in workflow.

| ICON KEY | |
|---|---|
| ⌘ | Valuable information |
| | Test your knowledge |
| | Exercise |
| | Caution |

**Workflow** saw its rise in popularity after Microsoft started marketing Windows Workflow Foundation as a part of the .NET framework. Released as part of the .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.

**Note**

While initially marketed as WWF, the marketing guys and gals in Redmond got a visit from some really huge and scary men wearing funny clothes, masks and sporting names like "Hulk" 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, flow of work. Think of any process where some set of actions are performed, for example buying an item in a store, taking a shower, filing a letter, or even dating. 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:

- find correct item
- take item to counter
- receive the price from attendant
- decide between credit card or cash
    - if cash:
        - give appropriate amount to attendant
        - receive any change
        - verify change
        - pocket change
    - if credit card:
        - give credit card to attendant
        - receive receipt for signature
        - verify amount
        - sign receipt
        - return receipt to 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 it has become a routine. Several of these steps require an amount of sub-steps, 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 perform 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. While two terms, the two forms are closely connected.

The simplest form is the sequential workflow in which a series of actions occur in a pre-defined 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. SharePoint Designer workflows are sequential workflows.

A state machine workflow is a bit more complex, but if you think of state machine workflows as a collection of sequential workflows, it 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 sub-workflow you can enter another state and thus trigger a second sequential workflow.

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

This issue will deal primarily with sequential workflows, but in Chapter 4, I will show you a method to simulate a state machine workflow using just SharePoint Designer and a repeating workflow.

If you do not understand the concepts of state machines yet, read on and look at the flow chart in the mission statement below where I will show you an example.

**Actions or Activities?**

The terms Actions or Activities refer to the same thing; the individual tasks performed in a workflow. Throughout this issue, I am using the term action, but you may encounter the term activity in other situations.

## Mission Statement

Throughout this journal, we will learn about SharePoint Designer workflows. We will do so by creating a simple CRM system where an order is received through email and then followed up until the order has been processed and the customer is happy. We will make sure of the latter simply by automatically emailing the customer to ask.

Figure 1 shows a flow chart of the solution.

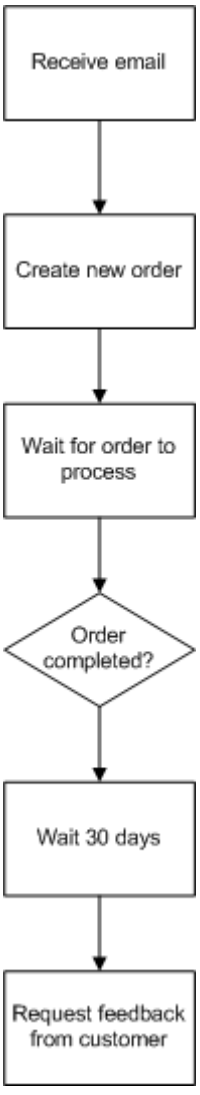



FIGURE 1. SOLUTION FLOWCHART

…

## What You Will Need

The first thing you will need is an up-to-date SharePoint installation. You should make sure you get the most recent updates as well, as fixes and updates may alter your experience. You should make sure you have the latest service pack, which at the time of this writing is Service Pack 2.

**XXX: Add link to SPD DL and SP2**

Besides SharePoint, you will of course need SharePoint Designer. SharePoint Designer was released free of charge on April 2, 2009, so all you need to do now is download and install.

Third, you should set up a basic Team Site if you want to get the exact results in the exercises. If you do not know how to set up a Team Site, talk to your SharePoint administrator and ask to get a site where you can play around. The workflows here only rely on a few lists, however, so anywhere you can create new lists should suffice.

## Rules of Engagement

In order to secure that we work in a manner that is supportable and that we learn using SharePoint Designer in an efficient manner, we need to define a set of rules for what we will do.

- **We will use only supportable methods and not harm a single Microsoft-provided file.**
  If you modify any file that ships with SharePoint, you will not be able to get support for your solution from Microsoft.

- **We will use the least amount of effort possible.**
  As per the introduction, we are going to learn how to utilize SharePoint, not necessarily using the "best practice" for any given scenario. Expect to add or improve error handling and modify texts, for instance.

- **We will utilize SharePoint native techniques whenever possible.**
  We might end up taking a step or two back to solve a problem. That is OK; we want to learn to use SharePoint.

Chapter 2

# SharePoint Designer Workflows

*Starring none other than…*

Now that we have a good understanding of what workflow is we can move on to explore how workflow is implemented in SharePoint Designer.

Perhaps a good starting point is to explain what SPD workflows are not.

- SPD workflows are not reusable. Once you have defined a workflow you cannot move that workflow to another list or reuse the workflow somewhere else. Well, you can, but you need to do some configuring magic. I'll show you how later in the chapter, but I don't recommend doing so on a regular basis.

- SPD workflows cannot do loops, at least not by default. By loops, I mean an iterative task performed repeatedly, such as modifying all items in a list, or sending an email to remind people about a task every day.

- SPD workflows have a limited set of activities. It is quite possible to utilize these activities to achieve impressive results but you are still limited more in what you can do than you are in other workflow scenarios, such as Visual Studio workflows. For example, you cannot do state machine workflows without getting very creative.

- While you can create new activities to use in SharePoint Designer, this requires programming skills and the upgradability of these custom activities is not certain for the next version of SharePoint.

If none of this makes any sense to you now, I'll explain more throughout this chapter, and you can then return here to see if everything is a bit clearer then.

So, let's take a look at what SPD workflows are all about.

## What are SharePoint Designer Workflows?

An SPD workflow is an easy, cheap, and somewhat limited entry point to workflow development. Easy, because most end users and administrators can become workflow developers without too much training. Cheap, because both SPD and SharePoint are now free. Somewhat limited, because of the reasons explained in the introduction to this chapter.

The main advantage to SPD workflows, however, is the speed with which you can create simple and custom business processes. Creating a workflow in SPD should not take too long, because if it does, you are likely making a more complex workflow than SharePoint Designer is designed to accommodate.

Let's get down to business and learn by exploring.

## Get to Know the SharePoint Designer Workflow Designer

Before we get into the exercises of the following chapters, I would like to show off some of the Workflow Designer options that are available for you. We will explore these in greater detail in the following chapters, so consider this just a tour of what is to come.

The Workflow Designer is your main tool for creating workflows in SPD.

You can create a new workflow from several menus in SharePoint Designer. From the File→New menu you can create a workflow directly, as shown in Figure 2, or you can right-click on any of the items in the Folder List and select New->SharePoint Content as shown in Figure 3.

**Note**

If you go the New SharePoint Content route, it does not matter which item in the Folder List you right-click. The Workflow Designer will always ask you on which list or library you want to create the workflow.

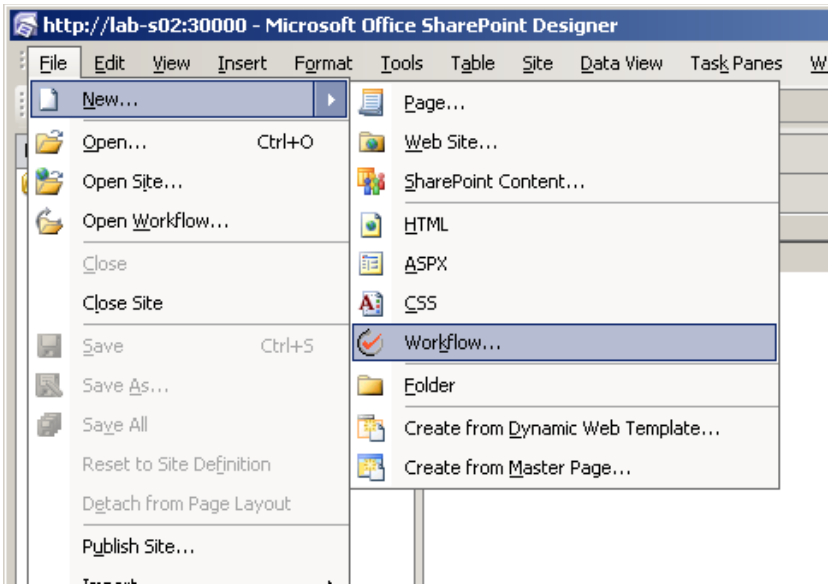



FIGURE 2. FILE→NEW→WORKFLOW

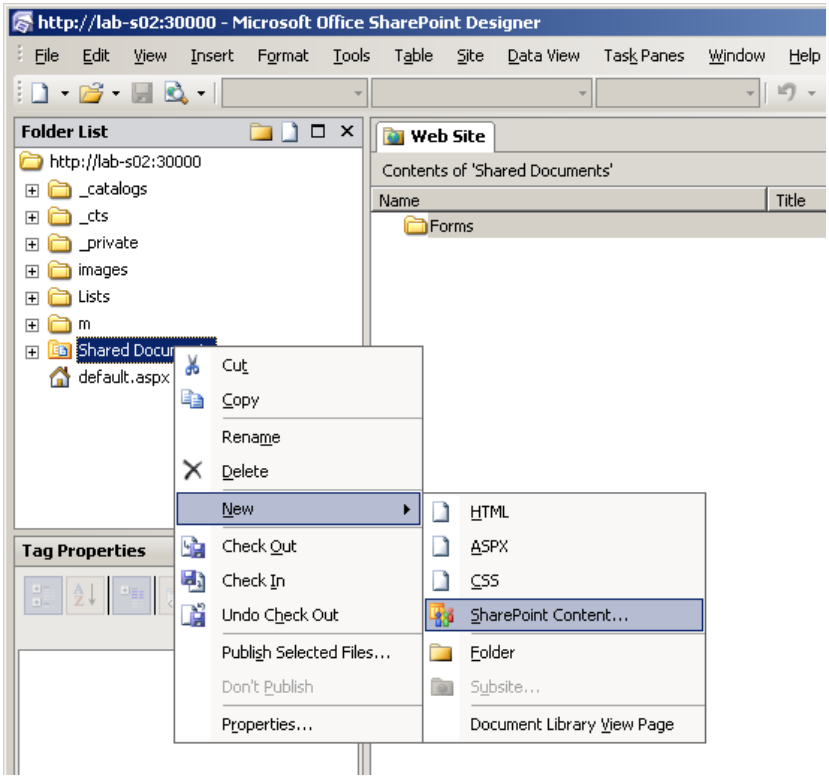



FIGURE 3. NEW→SHAREPOINT CONTENT

If you choose the New→SharePoint Content method you will get a New item dialog box as shown in Figure 4. Select the Workflow tab and the Blank Workflow item to start the Workflow Designer.

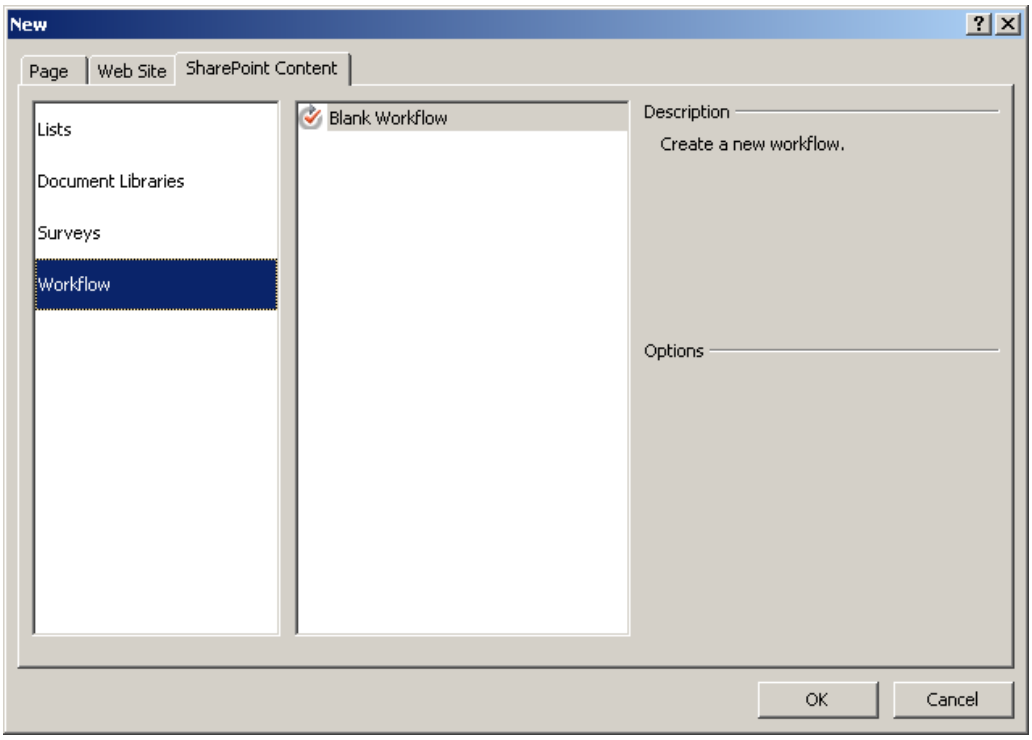



FIGURE 4. NEW ITEM DIALOG BOX

If you choose the File→New→Workflow the Workflow Designer launches directly.

The first page of the Workflow Designer, shown in Figure 5, requests information about where the workflow should be created, the options for how the workflow should launch, as well as giving you access to workflow variables management.

The location here is very important, because once you have selected the location and saved the workflow once, you cannot easily change the location. As I mentioned in the introduction to this chapter, SPD workflows are not reusable, so once you have set the location, the workflow locks to that list or library.

**Tip**

The other options on the Workflow Designer front page are changeable later. For example, for testing an automatically starting workflow, you might want to initially allow manual start to enable testing without having to create new items for each test. We will do this later in the exercises in Chapter 3.

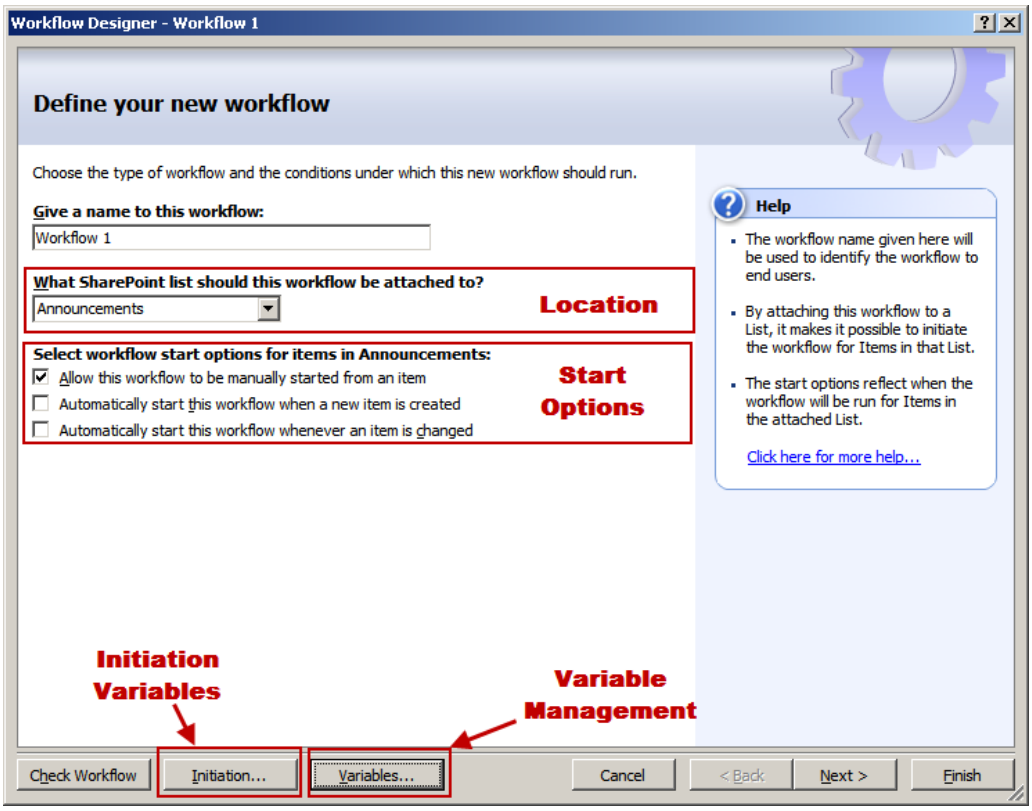



FIGURE 5. WORKFLOW DESIGNER FRONT PAGE

The Start Options determine how your workflow should trigger. Your options, and you can select them all, is to start workflows manually or to automatically start workflows either when an item is created or when an item is changed.

A very common setup during development of automatic workflows, as mentioned earlier, is to enable the manual option while you are developing. Then, after you have completed you can reopen your workflow and disable the manual start.

You may also want to set up initiation variables on this page, although you can create these on the next page as well. Initiation variables will be covered later in this chapter, but for now, just know that these are values the user can fill in when starting the workflow.

Once you have completed the first page, you should click Next to go to the main page of the Workflow Designer as shown in Figure 6.

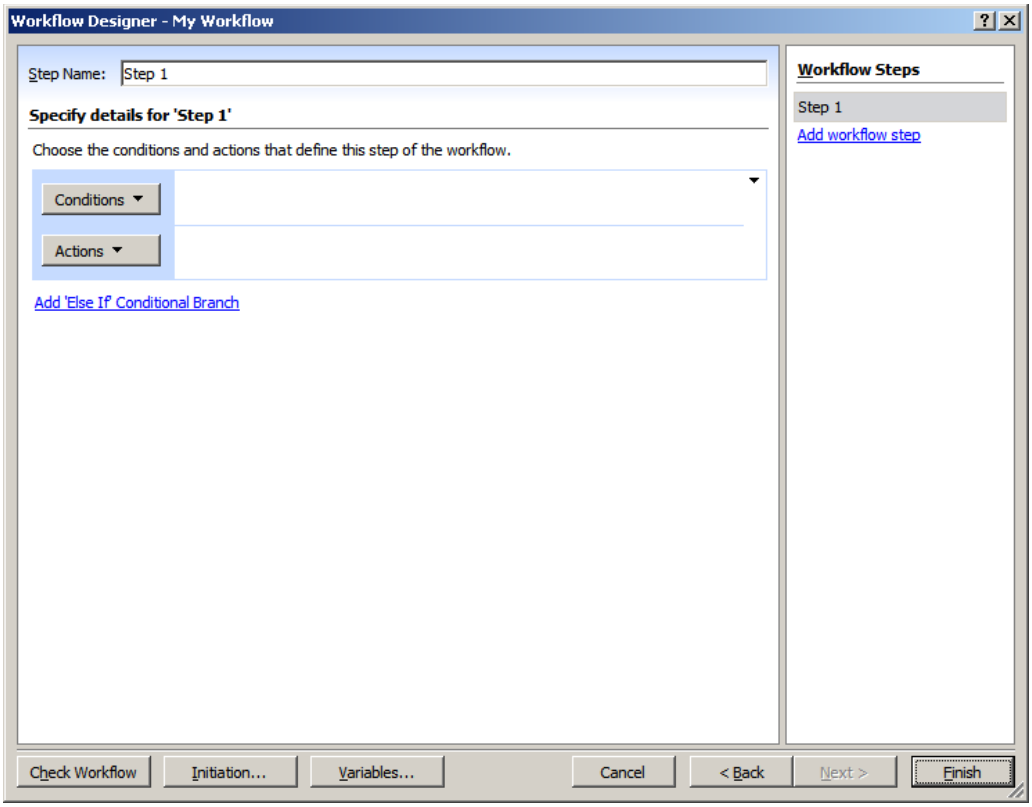



FIGURE 6. MAIN WORKFLOW DESIGNER PAGE

Note that the buttons for initiation variables and workflow variables are also present on the main page.

The main page of the Workflow Designer is where you add the actual logic to your workflow. On the right side, you see the various steps of the workflow, while the left side is where you specify the details of each step.

Each step can also perform multiple actions. For example, if you want to send an email you may want to construct the text of the email as one action and the subject as a second action, while sending the email is the third action. These three actions still constitute a single step.

To make matters even more complex, the actions of each step can also be grouped inside one or more branches, with each branch performing a set of actions depending on a condition. We'll be adding conditional branches later.

If, at this point, your head is spinning a bit, don't worry, I'll explain everything and show you how this all works throughout this chapter.

## Branching

The left side, as mentioned, is where you add the logic to your steps. The entire workflow step design process is done inside the Conditions and Actions row, called a branch. Initially, only a single branch is added, as shown in Figure 7.

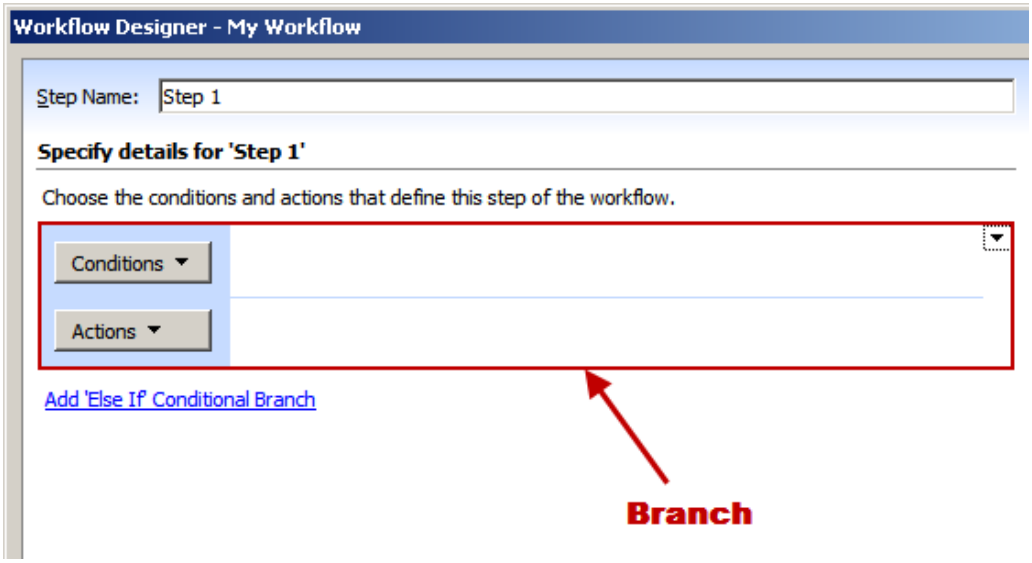



FIGURE 7. WORKFLOW STEP BRANCH

Branching is useful if you want to perform a certain set of actions only if a certain condition is met. Consider the flowchart in Figure 8:

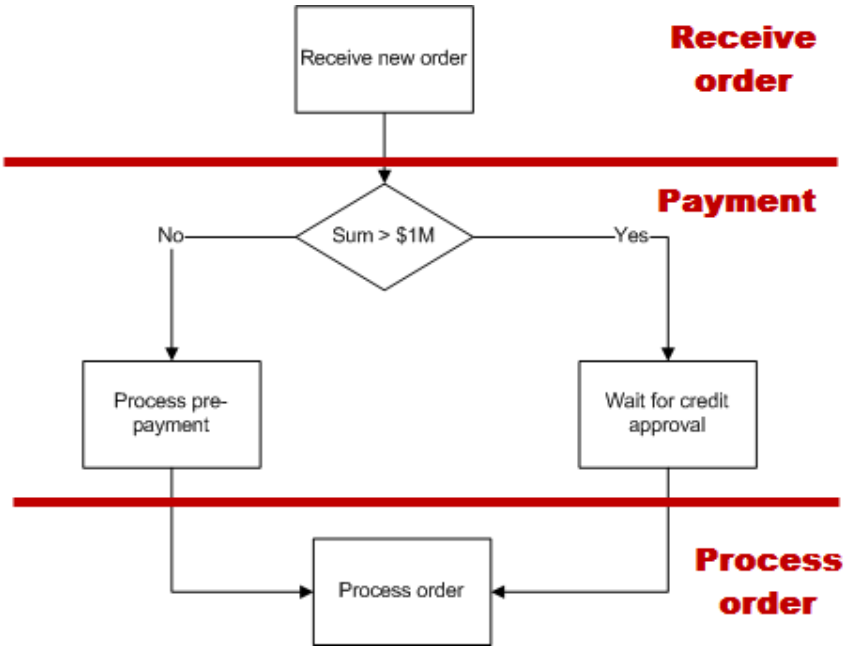



FIGURE 8. CONDITIONAL FLOWCHART

In this workflow, the method for payment of an order differs depending on whether the order sum is above $1 million. The Payment step is still a single step, but with two sets of actions. Which 'path' the step will take depends on the sum, but both paths are still considered the same step.

SharePoint Designer does not support nested branching. For example, in the above flowchart, if you need to perform a separate set of tasks if the credit is denied, you would need to get creative. Doing so is not difficult, but you need to think a bit outside of the normal way SharePoint Designer works. I'll show you how in Chapter 6.

If you want to create a conditional branch, you would first click the Conditions button. Doing so will open a menu that allows you to select what kind of comparison you want to make, as shown in Figure 9.

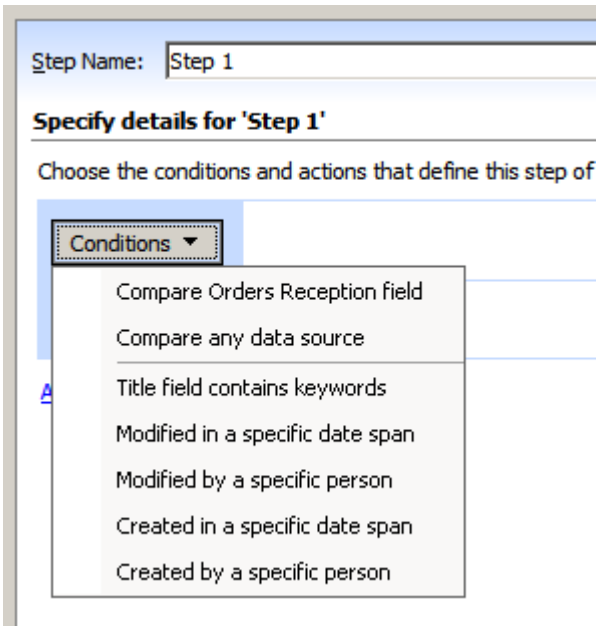



FIGURE 9. TYPES OF CONDITIONS

In Figure 9, the text Orders Reception refers to the list name to which I am adding this workflow.

Once you select the type of condition, the *condition editor* appears next to the conditions button, as shown in Figure 10.

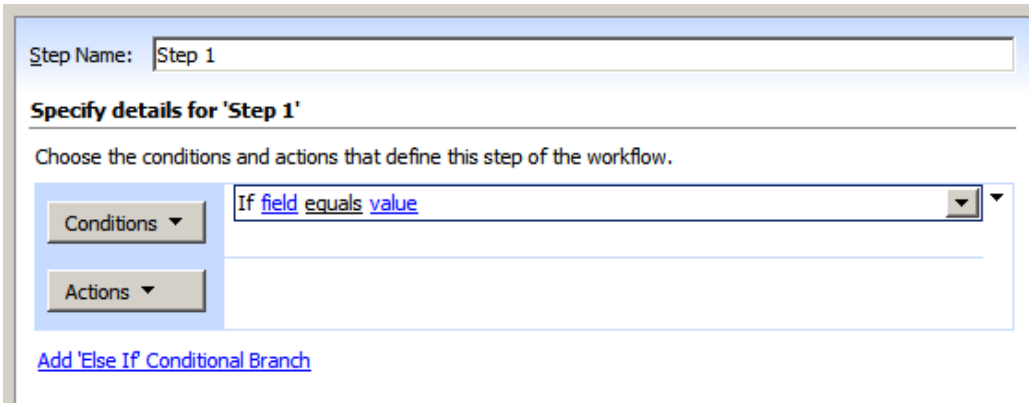



FIGURE 10. CONDITION CONFIGURATION

We will spend a lot of time with these editors, as they are also used to configure actions, but I want to just introduce you to how these editors work.

In general, any text that is underlined is something you can configure. For example, clicking the *field* text in Figure 10 will display a drop-down control that lets you select which field to test. Clicking the text *equals* will allow you to change the operator, and clicking *value* allows you to set the value to which the field should be compared. In Figure 11, I have set up a condition to test whether the Title field contains the text 'New order'.

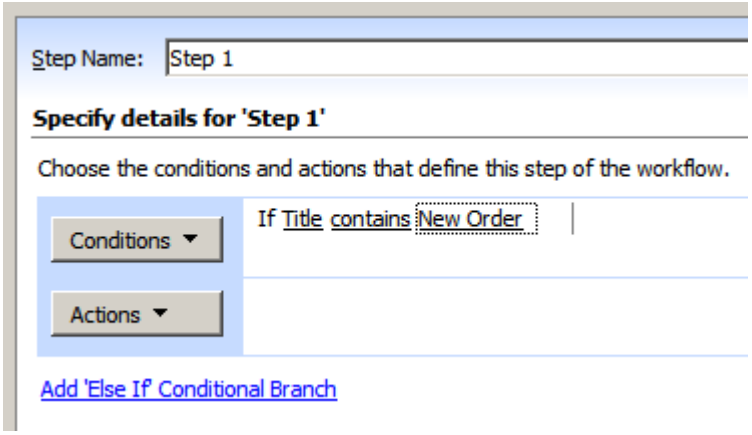



FIGURE 11. SAMPLE CONDITION

The behavior and appearance of the editors vary greatly. Some editors are quite simple, while others are very complex, with several levels of nested lookups to configure a condition or action. We'll see more of these editors later.

You can also set up more than one condition in a single branch. For example, you may check if the sum of an order is more than $1 million *and* that the order came from a new customer that has not previously been approved for credit.

To add a second condition to the same branch, simply click the Conditions button again, and select the second type of condition. Doing so will add the second condition under the first condition, and also give you a 'and' and 'or' text where you can select whether both conditions or just one must be true in order to perform the actions. If you click on the text *and* you change the text to *or,* meaning either of the conditions must be true for the branch to perform.

Figure 12 shows an example of multiple conditions. In this branch, I test whether the Title contains the text 'New order' AND that the item was NOT created by the Administrator user.

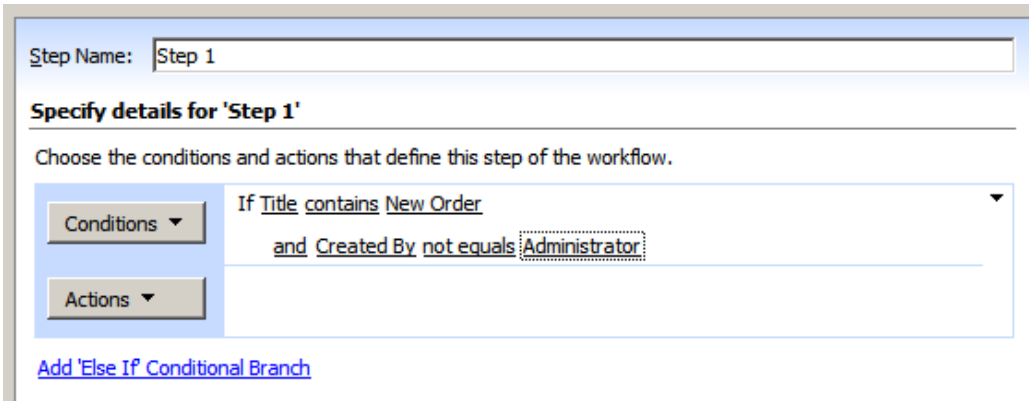



FIGURE 12. MULTIPLE CONDITIONS

The editor for selecting the value for the Created By field comparison is very different from the editor used to select the value for the Title field comparison. We will see more such editors later in this chapter and in the following chapters.

Finally, you can add a second branch in the same step. To do so, you click the down-arrow in the upper right of an existing condition and select *Add New Branch*. Doing so adds a second set of Conditions and Actions buttons in a new branch. In the second branch, you can set conditions as well. Figure 13 shows how this would look.

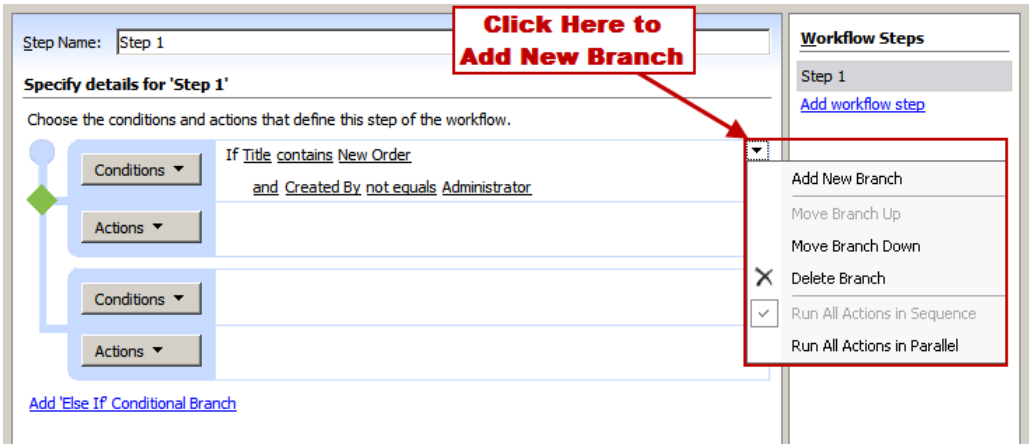



FIGURE 13. MULTIPLE BRANCHES

Let's get some work done and look at the actions.

## Workflow Actions

The Actions are the bread and butter of any workflow. Actions are what perform, well, actions. Without actions, you will have a very boring workflow.

To add actions to your workflow, click the Actions button of any branch. You will notice a similar popup menu as for the Conditions button, but the Actions menu s a bit different. See fig .

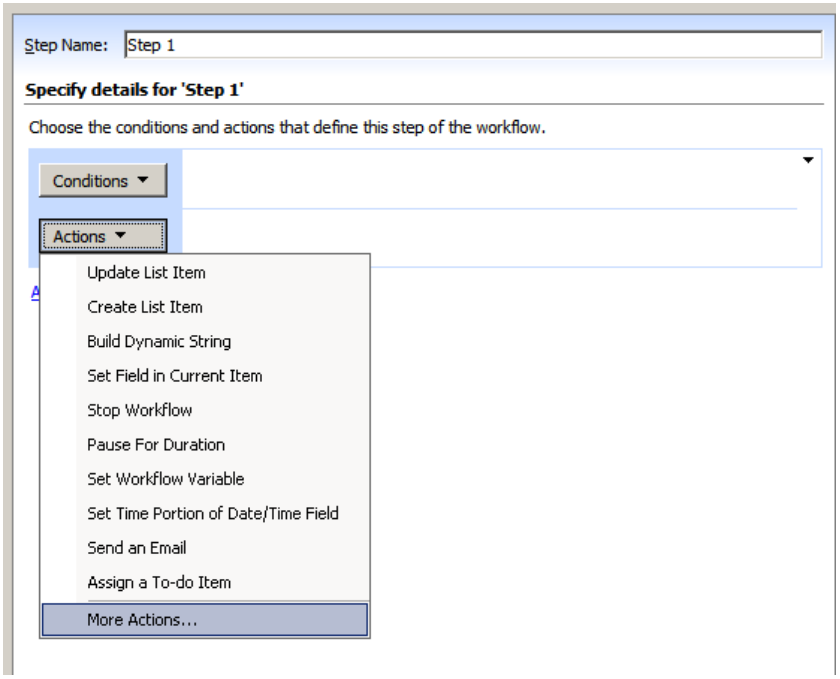



FIGURE 14. ACTIONS MENU

First, notice the *More Actions* item at the bottom of the menu. The range of available actions is greater than can fit in a single menu, so clicking this option will open a new dialog. More on that in a moment.

Second, and you will not see this from the figure, the items on the Actions menu are ordered in the order you last used them, so that the most recently used action is on the top of the list.

When you click any action in the Actions menu, that action is added to the branch. You then need to configure the action, with an editor that resembles the editor for the conditions. We will explore several of these actions and their editors during the course of the remaining chapters.

When you click the More Actions menu options, you get a second dialog box that displays all installed and available actions, as shown in Figure 15.

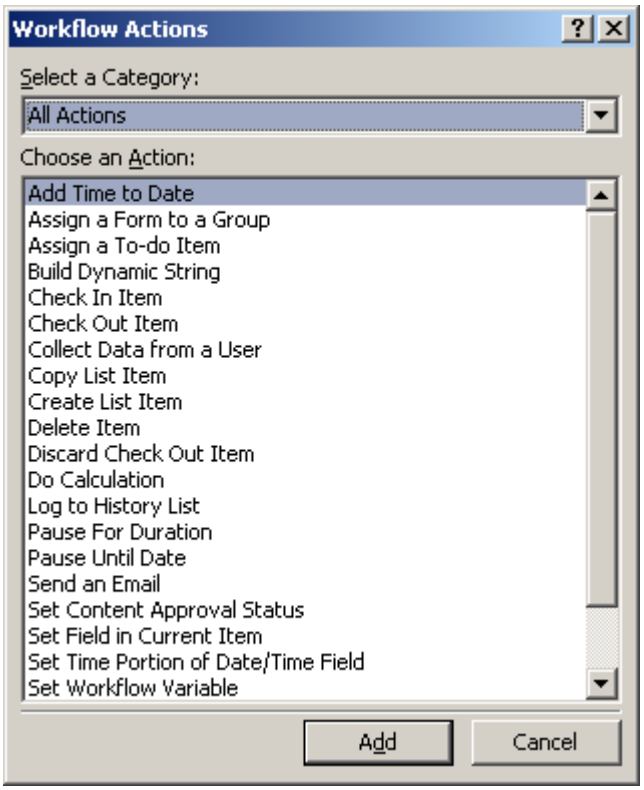



FIGURE 15. MORE ACTIONS DIALOG BOX

This list represent the complete actions buffet, categorized into three categories; core, list, and task actions. You can filter the actions by these categories in the top drop-down menu.

Adding an item from the More Actions dialog box is the same as adding from the Actions menu in the Workflow Designer window.

As for the conditions, you can also add several actions to a single branch. If you do, however, you can decide whether the actions should be performed in sequence or in parallel. If you look back at Figure 13, you will notice an option in the branch menu called 'Run All Actions in Parallel'. Checking this menu option will cause all actions to be started more or less at the same time, rather than run one after another.

Finally, back in the Workflow Designer window, notice that when you hover over an action you get a context menu drop-down, as shown in Figure 16. This menu allows you to reorder actions in a branch or delete actions you no longer need.

This context menu is also available for other items in the Workflow Designer, for example for steps on the right side of the designer. The options for each menu varies slightly, but the options aren't difficult to understand.

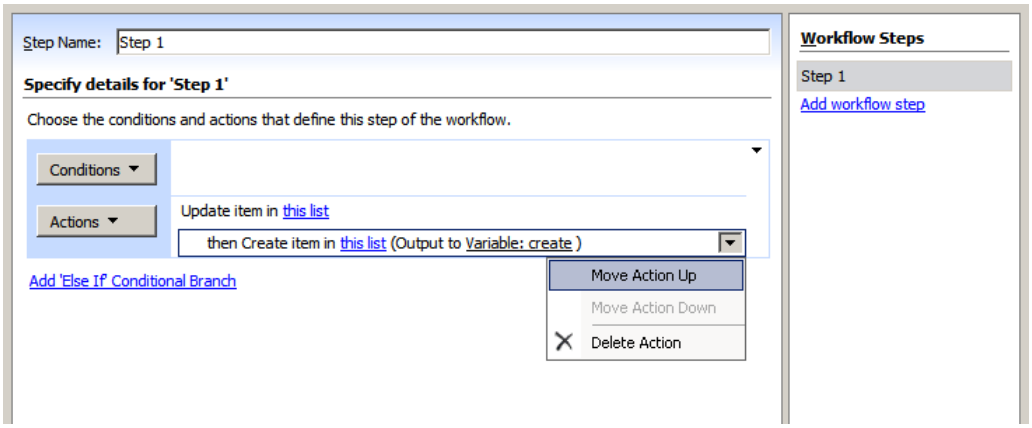



FIGURE 16. ACTION CONTEXT MENU

We will explore many of these actions in later chapter in this issue. Before we move on to those chapters, however, we need to look at another important concept of workflows.

## Working with Variables

Workflow variables are vitally important to any but the simplest of workflows. Variables allow you to store information retrieved either at the beginning or during a workflow run.

You have access to two different types of variables termed initiation variables and workflow variables. The biggest difference is how these variables are created. Initiation variables are created by asking the user to enter a value while workflow variables are created from the logic of the workflow itself.

The variables store data in one of five different data types:

| Workflow Data Type | Initiation Data Type | Description |
|---|---|---|
| Boolean | Yes/No | Stores a value that is either on or off, such as True/False, Yes/No, etc. |
| Date/Time | Date and Time | Stores a Date value |
| List Item Id | N/A | Store a reference to an item in a list. |
| Number | Number | Stores any number |
| String | Single line of text or Multiple lines of text | Stores non-formatted text. |
| N/A | Choice | Let's user select from a drop-down list box or a set of radio buttons |

Note that the Choice data type is only available in the initiation data types. The value selected by the user is stored as a string, though, so when you are working with that value in the workflow, you can treat it as a string.

## Initiation Variables

If you want to request the user to provide initial values for the workflow, you can add one or more initiation variables. When you do, SharePoint Designer will add controls to the workflow initiation page. In Figure 6 I have added a few initiation variables to a new workflow to show you how the different initiation variable data types are rendered to the user.

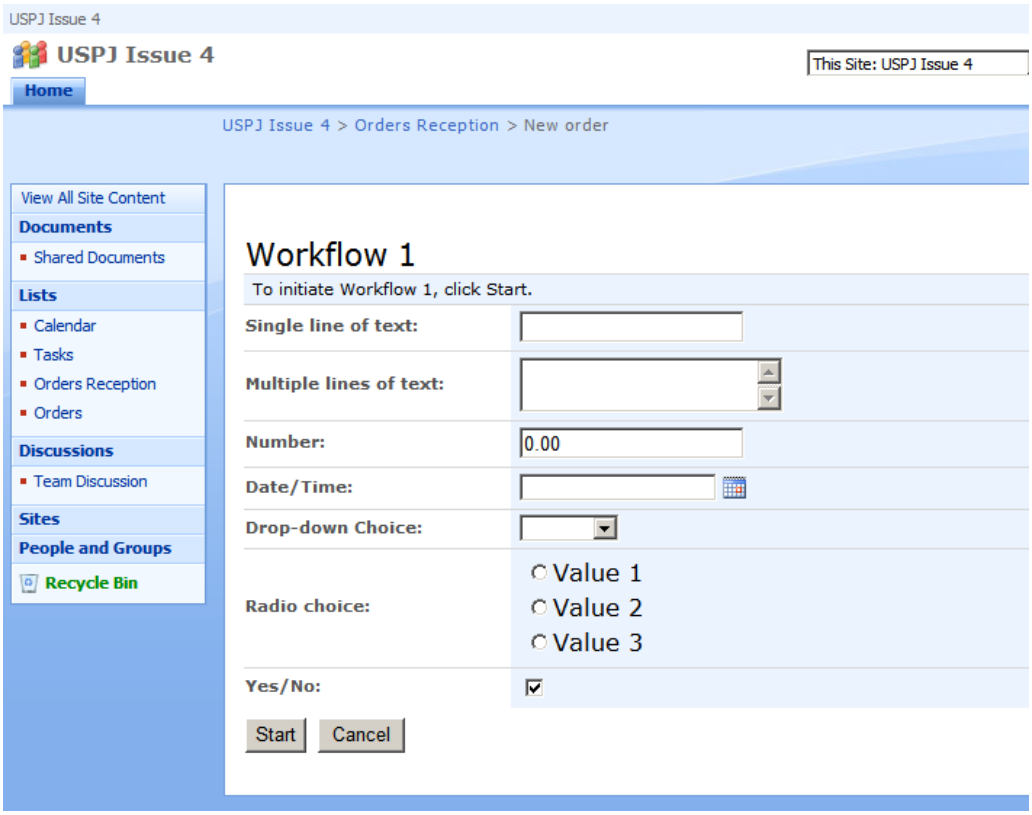



FIGURE 17. INITIATION FORM DATA TYPES

The really cool thing here is that SharePoint will create an ASPX form for you to hold the initiation values. This page is called the *initiation form*.

Once you are done with your workflow you can then modify the initiation form to change appearance or functionality. We will look into how to do so in Chapter 4.

***Default Initiation Values***

Another important concept with initiation variables is the default values. Consider what will happen if you need initiation variables for a workflow that starts automatically when a new item is created. No user is prompted to start the workflow and no one ever sees the initiation form.

This is where the default values are important. If you select default values for the initiation variables, these values are used when the workflow starts automatically. If you do not set default values, however, the initiation variables will have no value at all, which may cause problems for your workflow.

You are warned when creating an automatically triggered workflow to set default values for initiation variables. Also, when you save an automatic workflow where initiation variables do not have default values you are asked if you really want to associate the workflow, as shown in Figure 7.

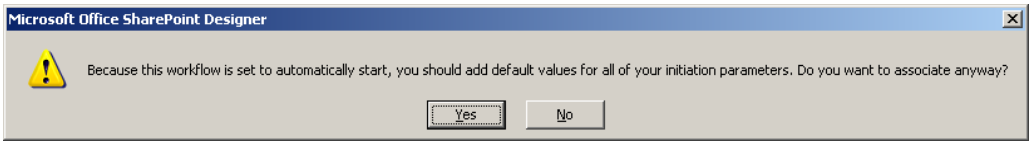



FIGURE 18. INITIATION VARIABLE WARNING

Don't worry, your workflow will still run, but if you rely on initiation variables having values, and your workflow is started automatically you may run into problems.

Before we move on, let's take a short break and stretch our bodies. Then, look at the review questions below to make sure you understand the key concepts discussed in this chapter.

### REVIEW QUESTIONS

? 1.

? 2.

? 3.

? 4.

## ✍ REVIEW QUESTIONS ANSWERED

- 1.
- 2.
- 3.
- 4.

Chapter 3

# Receiving Email

*Answering email was never simpler than this…*

So, we should get started on our first task of this mighty simple order management system. What we will create here is a system for receiving orders by email, assigning a task to process that order, and then sending a message to the customer 30 days after the order is process to make sure everything is OK.

Keep in mind that the components in this system are selected to show off various aspects of SharePoint Designer workflows, and are not necessarily the best methods for solving the problem at all.

For the first part of this solution, we need to accept a new order email. Then we need to create a new order object based on information from that email.

In the next chapters, we will then follow up the order and make sure that the order is processed, before we ask the customer for feedback after a certain period.

**Note**

In this chapter, we will use incoming email to trigger a workflow. The problem here is that automatically launching a workflow like this carries some serious security implications.

Microsoft has changed the way email triggered workflows behave, first by allowing it, then by denying it, and finally to allow it if the farm administrator makes some special configurations.

If you want to enable the automatic triggering of the workflow when you receive an email, consult the knowledge base article on http://support.microsoft.com/kb/953289.

## Task Overview

The task here is to receive an incoming email to a SharePoint list and based off that email, create a new order item in a separate list. SharePoint is capable of receiving emails to lists, a very useful trick to allow users to publish content to SharePoint from their email clients.

Also, this opens the option for SharePoint to connect to external and anonymous users, which can be a good thing, especially in our scenario where we preferably want people to send us orders as often and with as little effort as possible.

One problem with receiving emails in SharePoint is that only a few list templates support incoming email at all. Unfortunately, the custom list template is not one of those, which is a real shame since the custom list is a very versatile list template. None of the other list templates are well suited for creating a custom order handling system, so we want to use a custom list for the order processing.

Tip

**Tip**

Custom lists are quite capable of receiving incoming emails, but to make that happen, you need to develop an email receiver, which is done in .NET code. In my book *"Building the SharePoint User Experience",* I explain all the details of how email handling in SharePoint lists work, as well as show you how to create your own email receivers.

*Building the SharePoint User Experience* is now in stores. You can also order the eBook version online from the book web page at http://www.understandingsharepoint.com/userexperience.

Another problem is that each list only supports a single email alias. This may be sufficient in many cases, but if you are creating a more complex information routing system, for example with several entry points for incoming information, this may be a limiting factor. In our scenario, for example, we might want to have corporate orders received on one email address, while all other orders should be received on another address.

We need to overcome these two problems, and we will do so by having a separate list for receiving the email than for handling the order workflow. Therefore, we should create one order receiving list, which is email enabled, and another custom list, not email enabled, where we develop our workflow and handle the order through the various steps of our business process.

## Setting up the Environment

The first thing we should do is prepare the email reception list. In this scenario, I will use an announcement list to receive the email. You can use other lists as well, perhaps you want to enable scanned or electronic purchase orders to be attached to the incoming emails, in which case a document library might be a better choice.

**Note**

**Creating a Receiving List**

I won't go into details on setting up incoming email in SharePoint, as the settings vary greatly depending on your existing network infrastructure. As such, I will just assume that your SharePoint site is able to receive email, and that this option is enabled.

Start by creating a new list based off the Announcements list template. To do so, click Create on the Site Actions menu on your SharePoint site. You will find the Announcements list template under Communications, as shown in Figure 19.

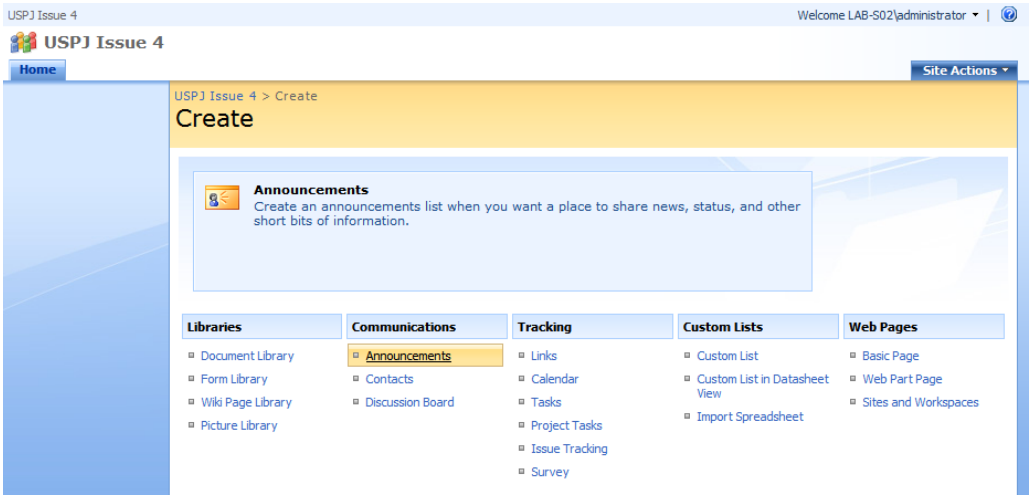



FIGURE 19. ANNOUNCEMENTS LIST TEMPLATE

Once you click to create a new announcement list, you will be directed to the New list page where you set the name, description, and Quick Launch options. If you have set up incoming email correctly, you will also see the incoming email settings on the bottom of this page, as highlighted in Figure 20.

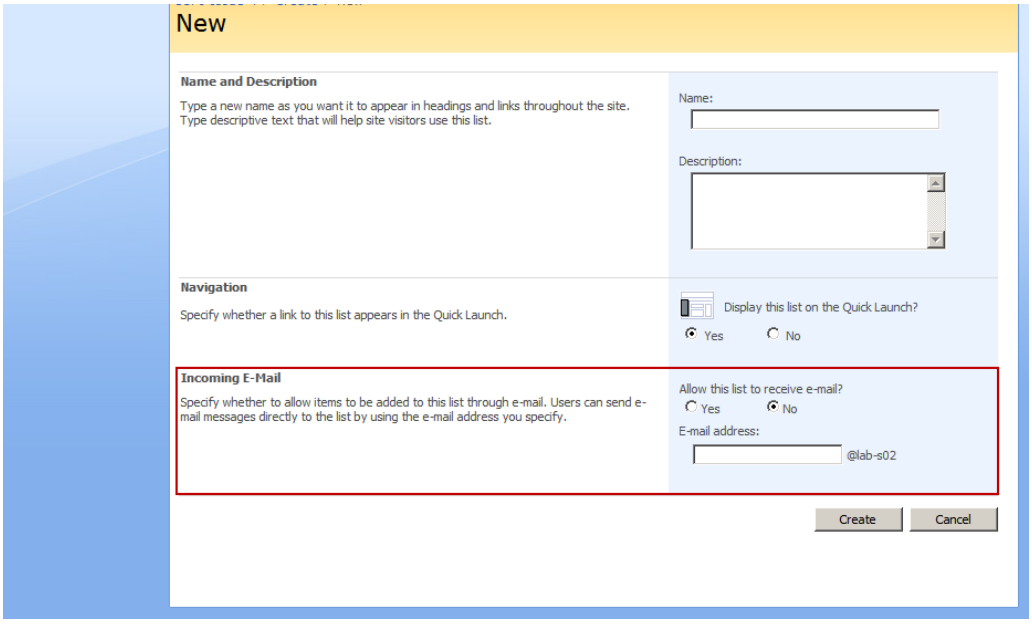



FIGURE 20. NEW LIST PAGE

**Note**

If you do not see the Incoming E-Mail settings at the bottom of the page, there is something wrong with your incoming email settings on your site. Verify your setup, or contact your administrator.

Yeah, and if you are the administrator, you should check those settings in the Central Administration, under Operations.

Select a name for your list, like "Orders Reception" and enter an email alias that you want to use, such as "orders@yourdomain". I am using the server name LAB-S02 in my lab, as you can see in the figure above. Click Create to, you guessed it, create your new list.

Even though you set the email alias on the New list page, you still have other options for incoming email that are accessible from the List Settings page. Once you list has been created, go to the Settings->List Settings and then to the Incoming e-mail settings link, shown in Figure 21.

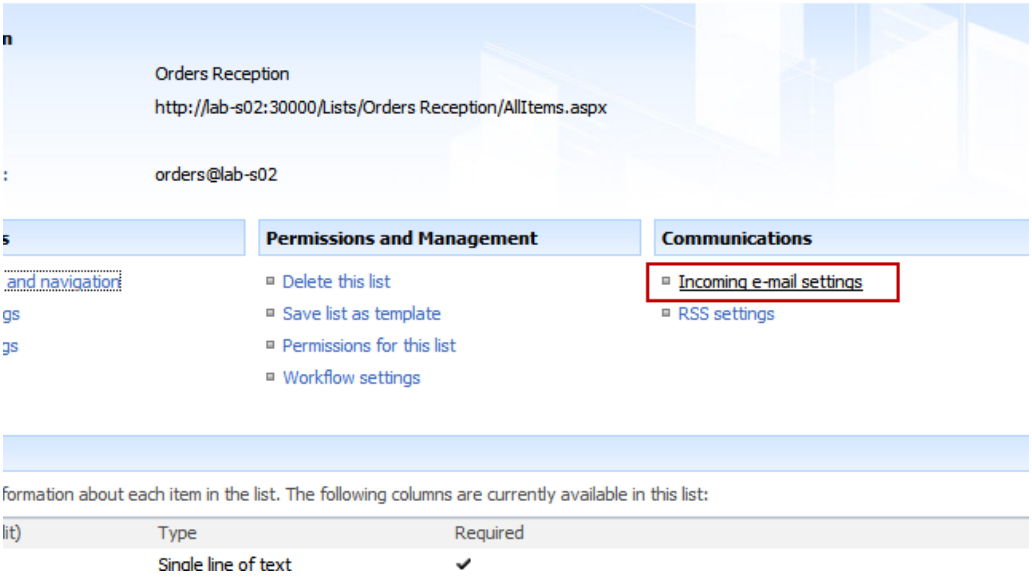



FIGURE 21. INCOMING E-MAIL SETTINGS LINK

On the Incoming E-Mail Settings page you can set up further options for your email reception, but the most important setting for us right now is to set the security to allow anyone to send us their orders. Go to the bottom of the page, and set the E-mail security policy to "Accept e-mail messages from any sender", as shown in Figure 22.

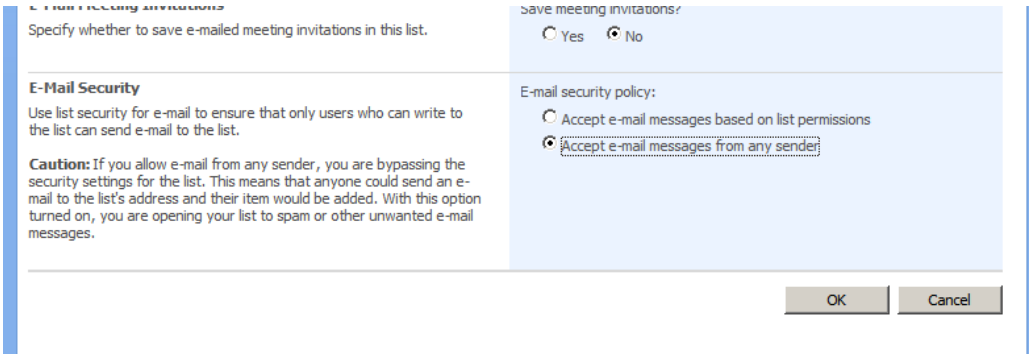



FIGURE 22. E-MAIL SECURITY POLICY

Hit OK to save you settings.

You may want to test your email reception as well. In my case, I have created a simple email message that I use to send using SMTP directly to the server. The SMTP message I send looks like this:

```
HELO LAB-S02
Mail from: customer@example.com
Rcpt to: orders@lab-s02
Data
```

```
Subject: New order
From: Customer Name <customer@example.com>
To: orders@lab-s02

Hello,

I would like to order 10 widgets.

.b

.

QUIT
```

Sending this through my local IIS SMTP server produces the result shown in Figure 23 in the Orders Reception list.

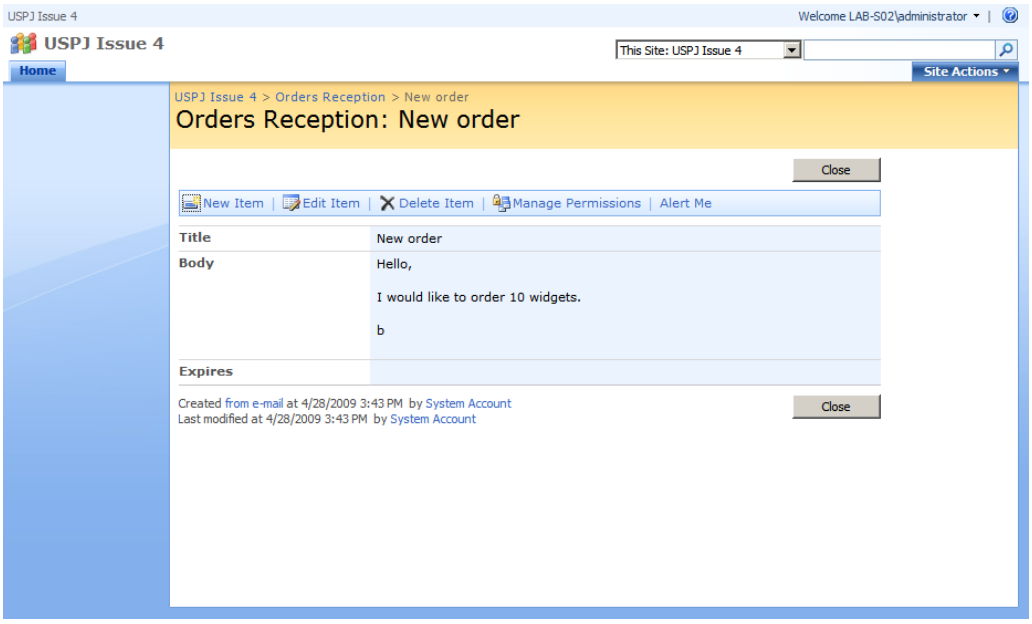



FIGURE 23. INCOMING EMAIL WORKS

Notice the text below the item metadata, which says "Created from e-mail at 4/28/2009 3:43 PM by System Account". The text "from e-mail" is a link, and clicking that link gives the result shown in Figure 24 that has more details on the email. These details are important to us later, but I just want to show you where you can find these properties.

Notice the difference in rendering of the E-Mail Sender column and the E-Mail From column. I'll explain more later in this chapter.

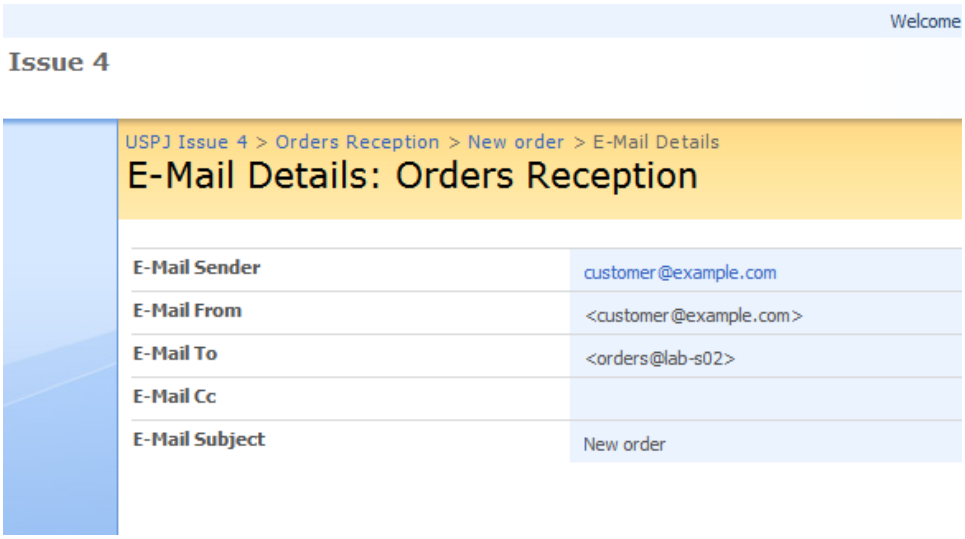



FIGURE 24. EMAIL DETAILS

Great, that's all we need to do to set up the list. Now it is time to move on to the workflow part of email reception.

## Receiving Orders Workflow

Now that you have a list to receive emails, we need to create a workflow to create a new order for the incoming email. Our incoming email should create a new order item in an order list.

**Creating the Orders List**

### Creating the Orders List

The first order of business is to create the orders list. For this list, we want to use a custom list. Create a new custom list, following the steps used to create the reception list. Name it something like Orders. Notice that on the New list page this time, no incoming e-mail settings are present.

Second, add three more columns to the list. To do so, go to the List Settings page, and click the Create column link, shown in Figure 25.

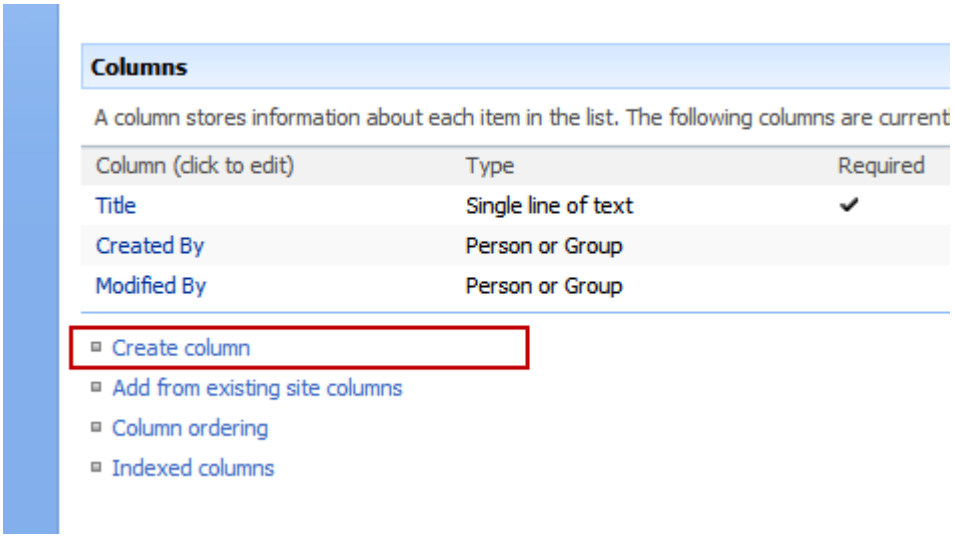



FIGURE 25. CREATE COLUMN

We need to add more columns to support the workflow later, but we'll add those when we need them. The columns you should add now are:

| Name | Type |
|---|---|
| **Email from** | Single line of text |
| **Order text** | Multiple lines of text |
| **Order sum** | Currency |

You can accept the default settings for these new columns. You may also wonder how we are supposed to find the order sum. I'll tell you later ☺

## Creating the Workflow

**Creating the Workflow**

The workflow we are about to create now will create a new order item for us in the orders list you just created. This task is well suited for SharePoint Designer workflows, as you will see in a moment.

Start SharePoint Designer and open the site where you have created your lists. The option to open a site is on the File menu.

Next, create a new workflow on the site. Refer to Chapter 2 if you need to refresh how to create a Workflow.

On the first page of the Workflow Designer, select the Orders Reception list in the lists drop-down. Give the workflow a nice name, such as 'Receive order'.

Then, set the workflow to start manually for now. Later we will set this to automatic, but for developing, setting the manual option is best. Your first page should now look like Figure 26. If it does, click Next to go to the main Workflow Designer page.

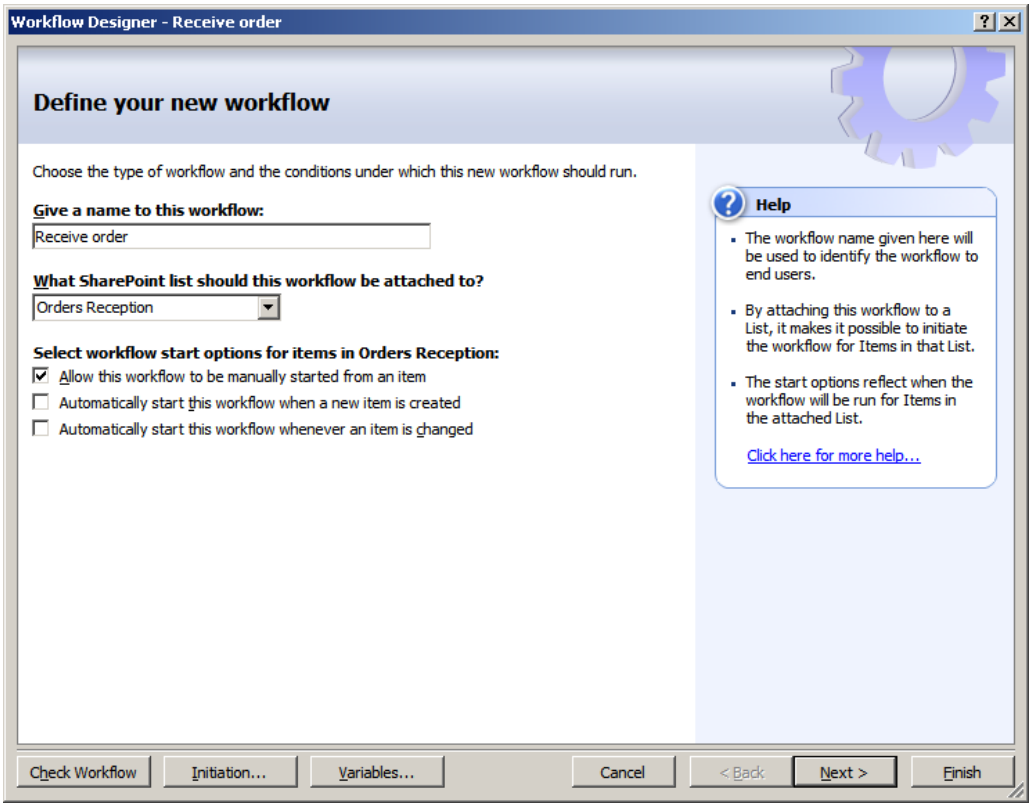



FIGURE 26. FIRST PAGE OF RECEIVE ORDER WORKFLOW

**Note**

At any point, you can click Finish on the bottom right to save your workflow. You can reopen the workflow later by opening the Workflows folder in the Folder List, locate your workflow, and double-clicking on the .xoml file named after your workflow.

### *Creating a New List Item*

When you get to the main page of the Workflow Designer, start by adding an action called 'Create List Item'. Remember, you may need to check the More Actions item on the Actions drop-down menu. Once you have added the action, click the parameter text *this list* as shown in Figure 27.

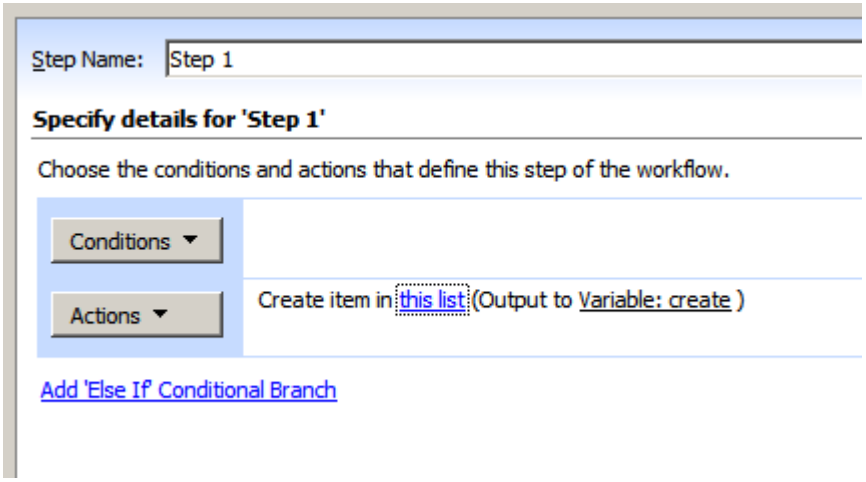



FIGURE 27. CREATE LIST ITEM ACTION

Clicking the *this list* parameter will open the editor for the item parameter that you will now configure.

Start by selecting the Orders list in the List drop-down. Doing so will populate the columns list with the required columns, marked by a start in parenthesis, as shown in Figure 28.

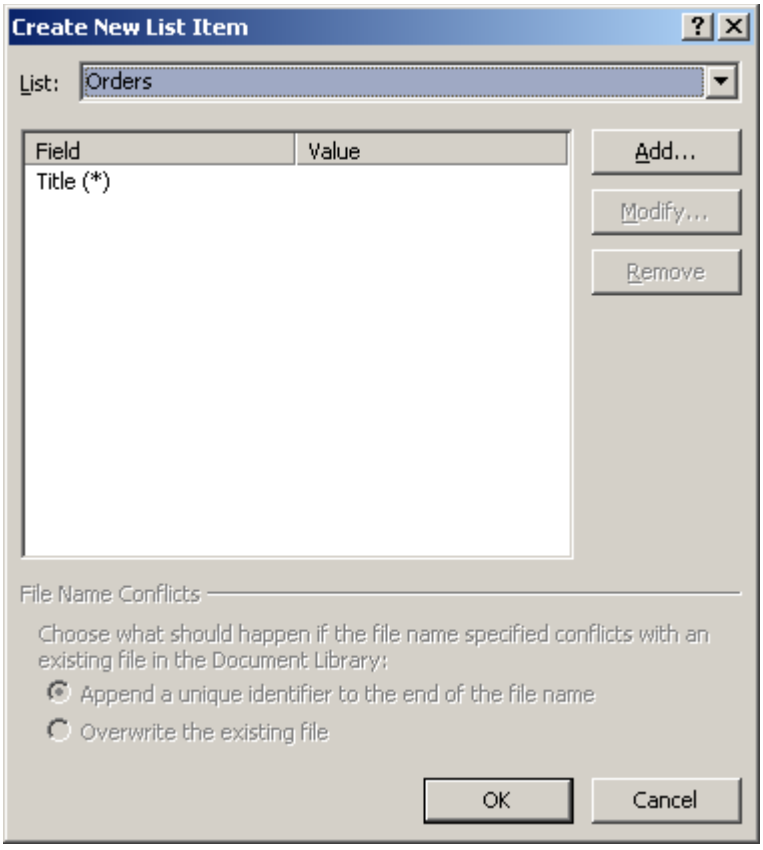



FIGURE 28. NEW LIST ITEM PARAMETER EDITOR

Click the Title column, and then Modify. This will open a new dialog box, called the Value Assignment dialog. Learning how this dialog works is important, as you will see similar dialog boxes often when working with workflows in SharePoint Designer. The Value Assignment dialog box is shown in Figure 29.

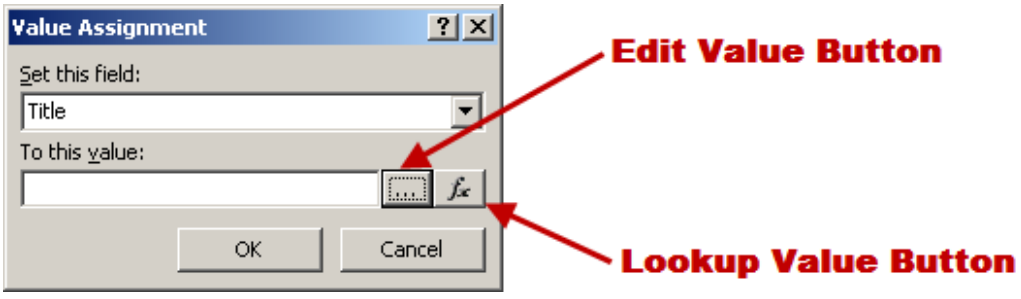



FIGURE 29. VALUE ASSIGNMENT DIALOG BOX

You will find the Edit Value button and the Lookup Value button in other dialog boxes and editors as well, so let me pause for a few moments to explain a bit how these work.

Both buttons will produce a second dialog box. The Edit Value button spawns a dialog box where you can edit a static value. The layout of the dialog box varies by value type. For example, the edit value dialog box for text is shown in Figure 30, while the edit value dialog box for dates is shown in Figure 31.

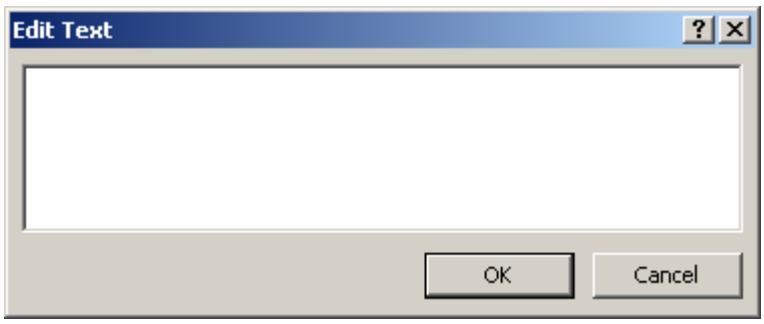



FIGURE 30. TEXT VALUE EDITOR

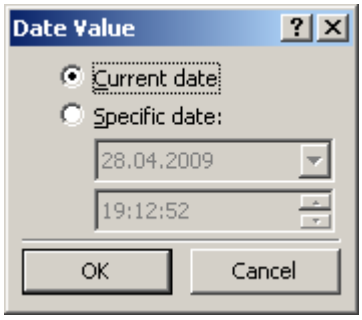



FIGURE 31. DATE VALUE EDITOR

These editors may vary slightly within the same value type as well. For example, if a text column supports rich text, you will see a notice about this at the top of the dialog box.

The lookup value assignment dialog, however, is very different. You would use this editor if you need to look up a value dynamically when the workflow runs. For example, you may want to look up a value in another list and use as a value in a new item, or you may want to use a value in the current item, which is just what we want to do with the title.

Click the Lookup button (  ) to start the Define Workflow Lookup dialog box. Open the Source drop-down to see the available options. You will see all your lists and libraries, as well as two additional options, the Current Item and the Workflow Data options. These options allow you to use values from the current item or from workflow variables, respectively.

For this exercise, however, we'll keep this simple, so just leave the Current Item source selected, and also, select the E-Mail Subject as the field. Click OK twice to return to the Create New List Item dialog box.

Next, click the Add button to add a new column mapping. In the Set this field drop-down, select Email from, and then click the Lookup Value button to map the Email from column to the Email Sender column in the Current Item source.

**Note**

When you look at the drop-down list of possible fields in the current item, notice that there is both an Email From and an Email Sender column. The Email Sender includes HTML formatting to make the address appear as a link and is not a valid address, so make sure you pick the Email From.

Not Email Sender.

Repeat this process for the order text, and map that to the Body column of the Current Item source. Your completed Create New List Item dialog box should resemble Figure 32. Note that I have adjusted the columns to display the entire text.

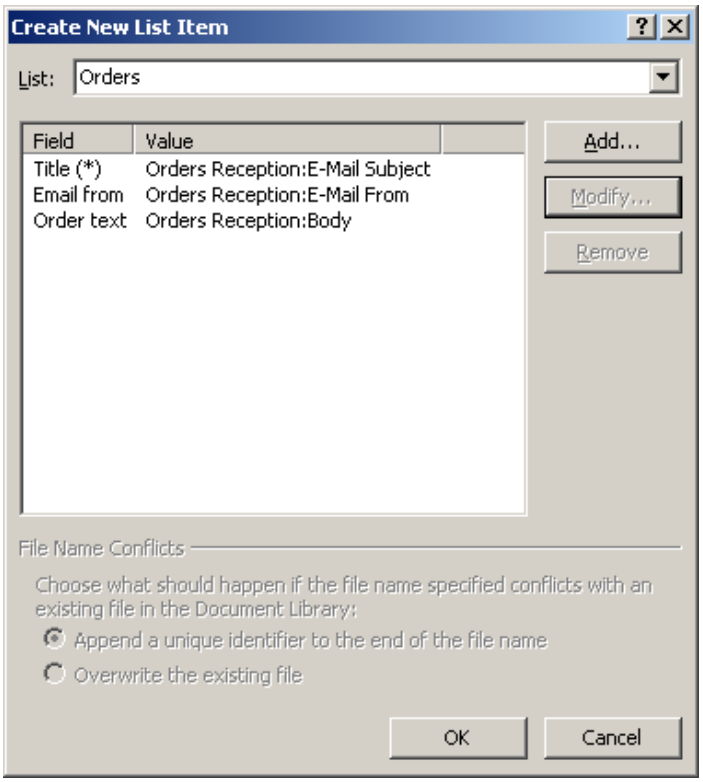



FIGURE 32. CREATE NEW LIST ITEM COMPLETED

That's it! Hit OK and you action has been configured.

For now, we can be content with our workflow, provided everything works as expected. If you think this was too simple, just relax, we will get a lot more complex and explore several other actions and scenarios in the following chapters.

Save and publish your workflow by clicking Finish. This will make your workflow available on the Orders Reception list, so open that list in SharePoint now.

Again, I will assume that you have the incoming email setup correctly, so send an email to the alias your created for the Orders Reception list, and wait for SharePoint to pick up the incoming email.

**Note**

Receiving email is not immediate, but controlled by a timer job that checks for new emails every few minutes. If your email does not arrive in your list right away, grab a cup of coffee and wait.

If you don't drink coffee, start now.

When your new email arrives, you will see a new option in the item drop-down in the SharePoint list called Workflows, as shown in Figure 33.

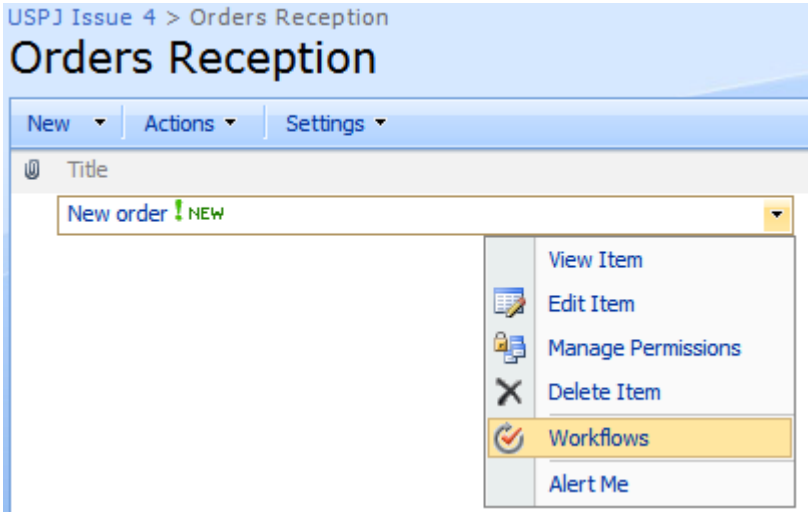



FIGURE 33. ITEM DROP-DOWN

Clicking the Workflows option will take you to the list of available workflows for the item. You should see your Receive order workflow here now. Click the workflow icon or name to get to the workflow initiation form, as shown in Figure 34.

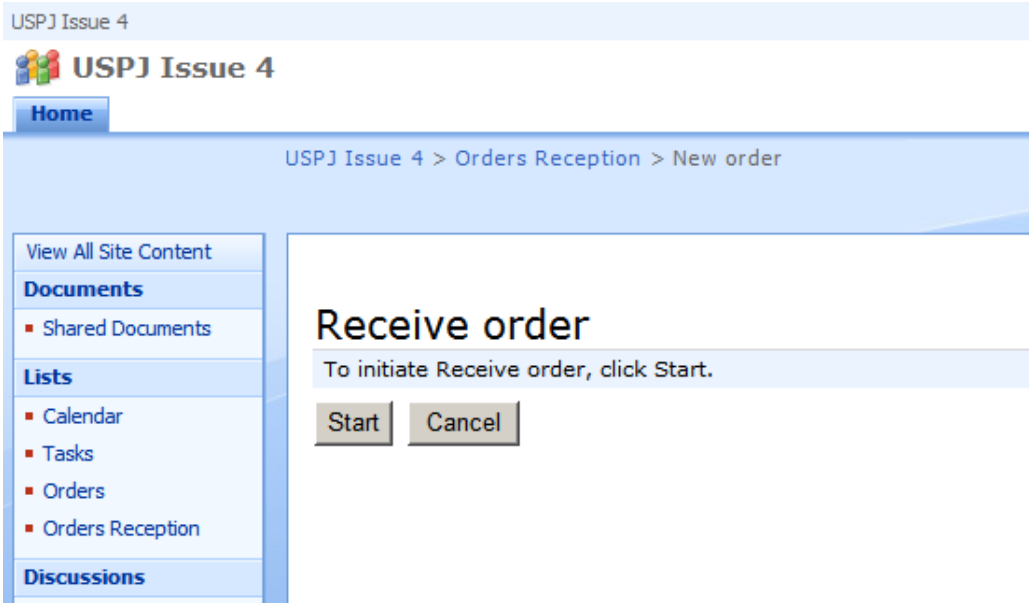



FIGURE 34. RECEIVE ORDER INITIATION FORM

When you are ready, take a deep breath, and click the Start button. After a short while, you should be sent back to the list view page where you will be greeted by a slightly modified list view that now includes a column called *Receive order*.

SharePoint adds the Receive order column the list the first time you run a new workflow on a list. For each item on which the workflow has run you will see a status message. If the workflow completed without errors you will see the text Completed, and you can click that link to see the results of your workflow run on what is called a Workflow Status page, shown in Figure 35.

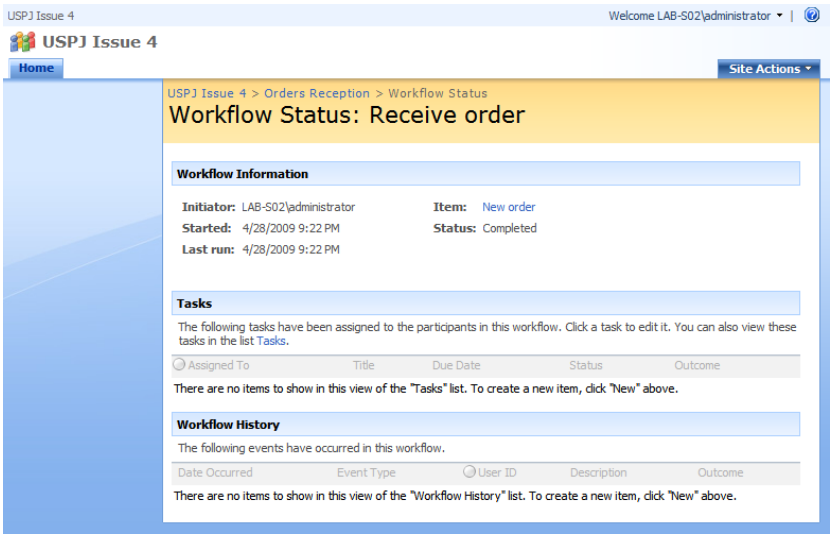



FIGURE 35. WORKFLOW STATUS PAGE

The Workflow Status page is useful both after a workflow has run and during the running of a workflow. After the workflow has completed you can review information about tasks created, events that happened, as well as any errors or problems that have occurred.

During a workflow you can review the status and see what current activities are holding up the workflow, for example what tasks are currently awaiting completion. We will explore this more in the next chapter.

Finally, go to the Orders list and see if your new order item is there. Figure 36 shows my results.

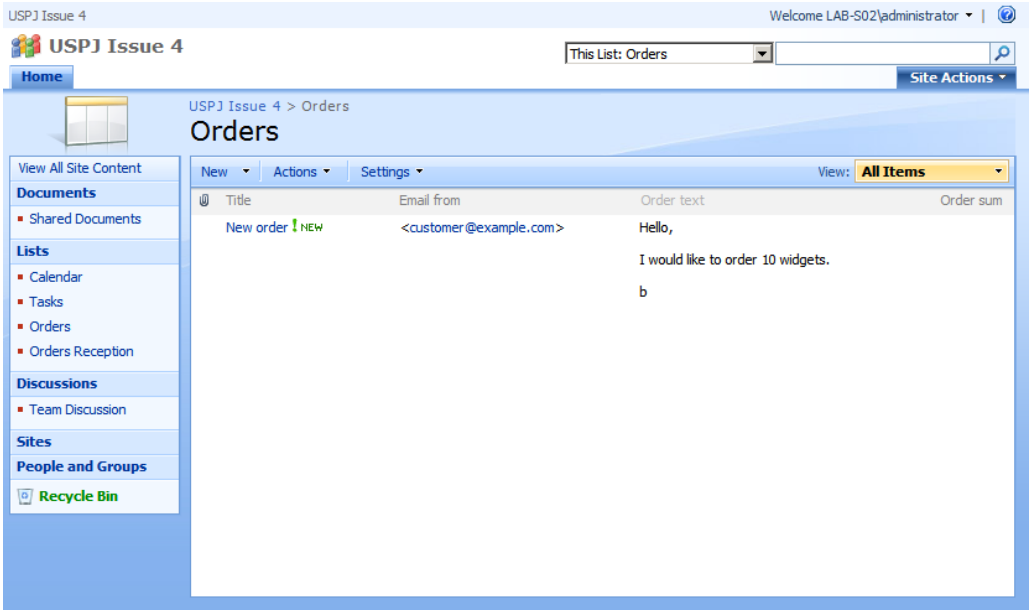



FIGURE 36. NEW ORDER CREATED

Congratulations! You have created your first workflow! Simple, eh? We will get more funky later in the chapters, though, so don't let that new confidence go too far to your head.

If you want, you can now reopen the workflow and set the start options to automatic. To do so, go into SharePoint Designer and browse to the Workflows folder in your Folder List. Inside, locate the Receive order workflow and double-click the .xoml file to open the Workflow Designer. Change your start options and click Finish to deploy your updated workflow.

Remember the note at the beginning of the chapter as well. You need to consult the knowledge base article to learn how to enable email triggered workflows.

Let's review the key concepts and make sure you got the most important parts of this chapter, shall we? The answers are on the next page.

### ✍ REVIEW QUESTIONS

? 1.

? 2.

? 3.

? 4.

## ✍ REVIEW QUESTIONS ANSWERED

- 1.
- 2.
- 3.
- 4.

Chapter 4

# Data Collection

*Got to know what to do.*

OK, so far, so good. We have created the first of several workflows we will need to complete our order management system. Now it is time to make everything a bit more complicated, because this time we will need to handle several steps in our workflow.

## Task Overview

In this chapter, we will create a new workflow that triggers when a new order item appears in the Orders list. The workflow will first send an email to the order sender to inform them that the order has been received.

Then, the workflow will create a new task of processing the order by updating the Order sum column. To accomplish this, we need to use a recursive lookup; one of the most powerful workflow tools in SharePoint Designer.

Let's get right down to business.

## Creating the Process Order Workflow

**Creating the Email Workflow Step**

The rest of this issue will deal with a new workflow, attached to the Orders list. So, you first task is to create a new workflow for the Orders list. Refer to the previous chapters if you need a refresher on how to do so.

Name your workflow something like 'Process order' and leave it with manual launch as the only start option for now. You can always change this later. Click Next to get to the main Workflow Designer window.

In this chapter, we will create the first three steps of this workflow. The first step will be to notify the customer that their order is now processing. You should name your first step accordingly, such as 'Send customer receipt'.

From the Actions button on the first step, find the Send Email action and click to add that action to the step. Remember that actions may be located on the More Actions option in the Actions menu. Once you do, click the text *this message* to start the action editor for sending emails, shown in Figure 37.

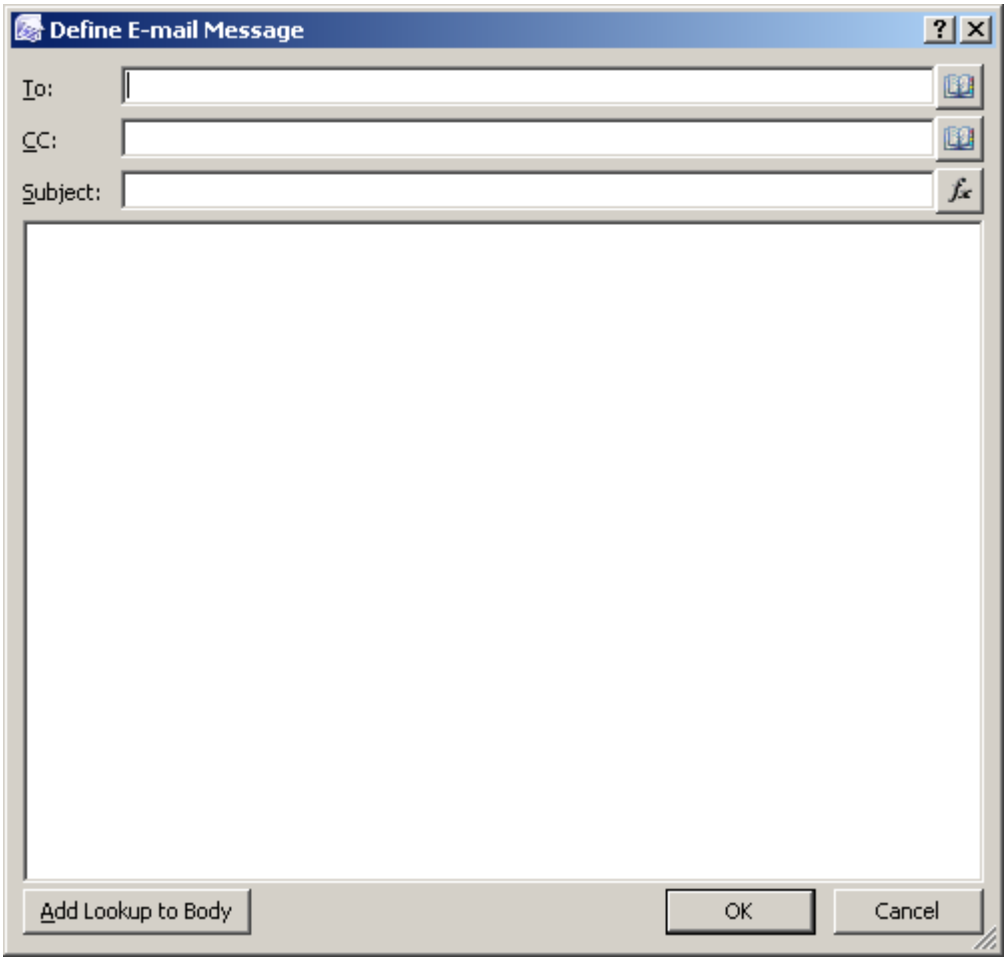



FIGURE 37. DEFINE E-MAIL MESSAGE EDITOR

The editor looks a lot like your regular email client, but beneath the surface, the complexity can be bewildering. We will re-visit this editor later as well, so we'll cover some of the basics now.

Next to the To field you have an address book lookup button. Clicking that button brings up a Select Users dialog box, shown in Figure 38. Notice the three special items at the top of the users list on the left.

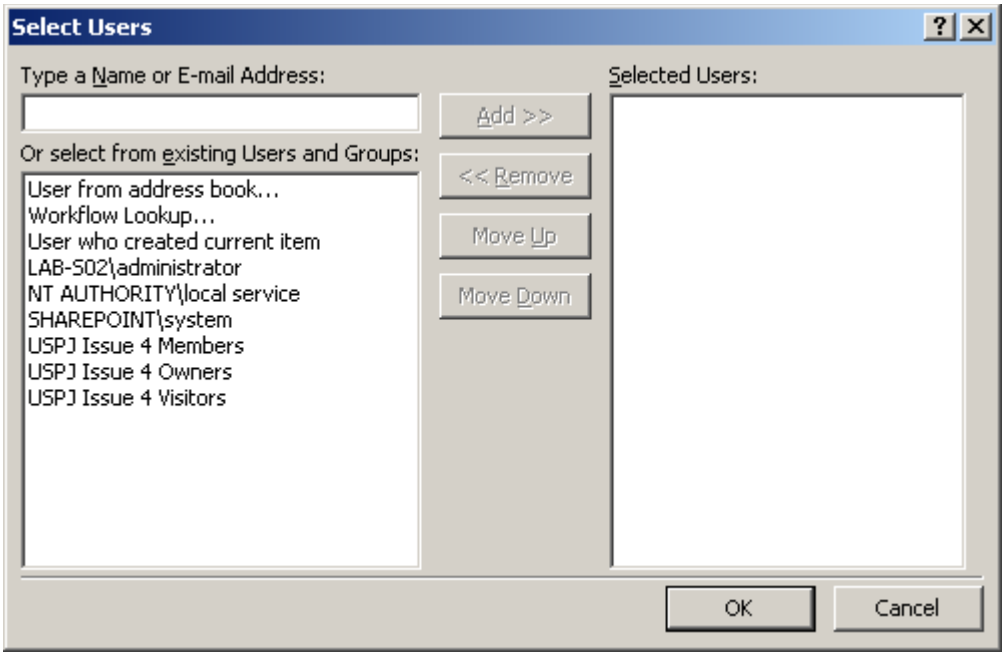



FIGURE 38. SELECT USERS DIALOG BOX

The three special items are used to look up users in other parts of the workflow, the site, or the organization.

The simplest option is the 'User who created current item'. This option selects the user who created the current item, plain and simple. Oh, and of course, that user must have an email address defined in SharePoint for any email to reach that person.

The User from address book… is just a simpler way of finding users within an organization. Again, this relies on users having email addresses added, and that you have a working local address book, which is the case if you are using email clients such as Outlook.

The Workflow Lookup… option is more interesting. Select that option now, and click Add>> to start the Workflow Lookup dialog. You may recognize the dialog box from the previous chapter. Again, we just want to pick a value from the current order item, so leave the Source set to Current Item and select Email from as the Field, as shown in Figure 39.

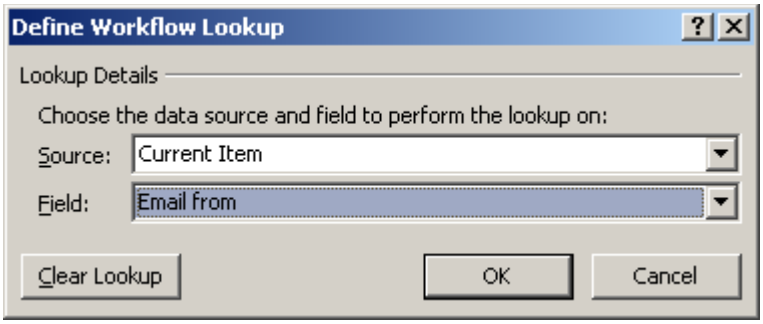



FIGURE 39. WORKFLOW LOOKUP

We will explore this dialog more in the next exercise, so for now, just click OK and notice that the right side of the Select Users dialog now has an item called Orders:Email from.

You can add multiple recipients to an email message by selecting other options on the left and then click the Add>> button. For example, you may want to store a copy of the outgoing customer receipt, or even send that receipt to another SharePoint list. Just make sure you are not sending an email to the Orders reception list, as this will spawn a new order and throw you into an endless loop.

In the Select Users dialog box, click OK to get back to the Define M-mail Message editor window.

In the subject, you can write a static text, such as 'Your order received'. Your other option is to use a workflow lookup, finding the information somewhere else. A common scenario is to build a dynamic string prior to send the email and use that string as the subject. That string can then be stored in a workflow variable and added using the workflow lookup option.

Sadly, this is an either/or field. Either you need to define a static text or you need to use a single existing value from the lookup. We will explore the latter option in the next chapter, where we will build a dynamic string to serve as the subject of the receipt email.

The body is more flexible, though, and allows you to create a template with lookup data.

Inside the large area below To, CC, and Subject, type a message, such as the following. Feel free to :

```
Hello,

We just wanted to let you know we have received your order dated

Your order will be processed and you will receive a confirmation
with the total sum.
```

```
Thank you,

Company
```

Place your cursor after the word dated in the second line, and click the Add Lookup to Body button below the text area. The workflow lookup should appear.

Still keeping with the simple stuff, leave Current Item selected, and select Created from the Fields drop-down. Hit OK and notice that the second line now reads:

[…] know we have received your order dated [%Orders:Created%]

The [%Orders:Created%] token will be replaced with the date that the order item was created in the list.

Finally, click OK to complete the email setup. Your first step is now complete, and should resemble Figure 40.

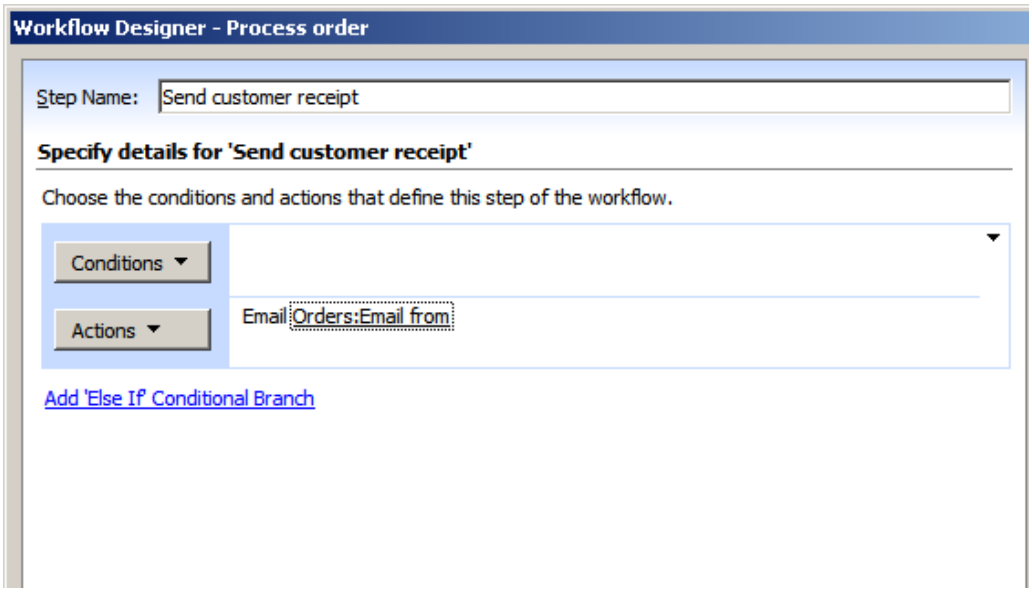



FIGURE 40. SEND CUSTOMER RECEIPT STEP

Remember that you can save your workflow at any time by hitting Finish and then reopening the .xoml file in the Workflows folder.

Ok, let's move on and look at the next step of our workflow.

## Collecting Data

Collecting data from a user is a very common task in a workflow. You may want to get feedback from users or managers, you may need to gather information from different departments for a report, or you may have other reasons for wanting data entered into a workflow.

**Collecting Workflow Data**

In SharePoint Designer, a dedicated action handles the task of collecting data from a user. We will use this action to request the order total from a user in this exercise.

First, however, create a new step by clicking the *Add workflow step* link on the right side of the main Workflow Designer page. Name the new step something like 'Retrieve order total sum'.

Next, add a new "Collect Data from a User" action. Again, remember that you may need to look under More Actions. The Collect Data action has two parameters that need configuring; the *data* parameter and the *user* parameter.

When you click the text *data* in the action, a new type of editor appears, called the Custom Task Wizard, as shown in f.

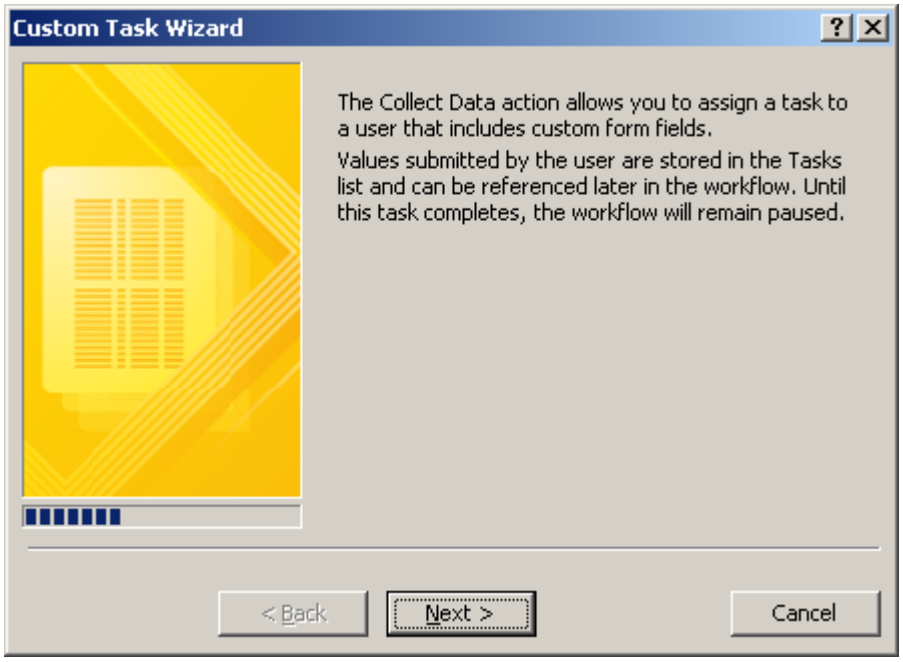



FIGURE 41. CUSTOM TASK WIZARD, PAGE ONE

The Custom Task Wizard helps you set up a new task for a user. You will see other versions of this dialog in SharePoint Designer, for example if you add a to-do item action. In the case of a Collect Data action, however, the wizard will also let you set up what data should be collected.

Click next to go to the title and description page of the wizard.

The first thing you need to understand here is how these tasks work. When you assign a task to someone as part of a workflow that task will be added to the default task list in SharePoint. If no task list exists, SharePoint creates one for you.

Each task will have an associated page where the user will be instructed on how to complete the task. For a plain to-do item action, your only options are to use the title and the description to explain to the user what to do. However, once the Task Wizard creates the new task action, it also creates a regular ASP.NET page that you can configure in any way you like. We will explore that in a moment.

The file name and title of that page is defined from the name you give to the task. Considering that many tasks are similar the potential for naming conflicts is definitely present in more complex workflows. As such, take heed to pick a name that is distinct, not only to this task but also to this action. A name such as 'Collect data' may not be unique enough to prevent file name collisions.

In my example, as shown in Figure 42, I have named the task *Retrieve order total* and entered a description text which is *This text describes the task.* I have used these names so you will recognize where they are used in the task page.

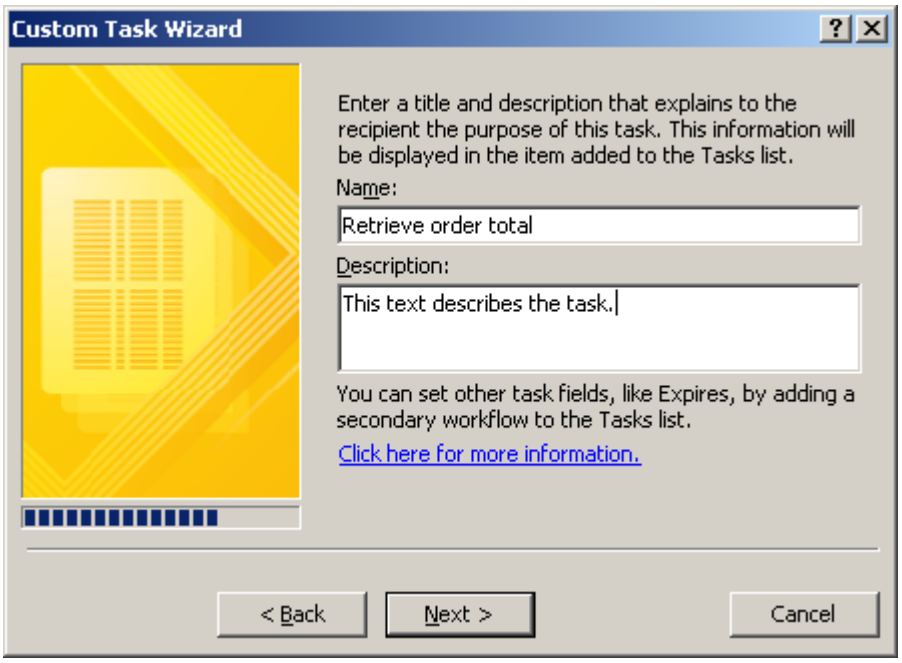



FIGURE 42. CUSTOM TASK WIZARD, PAGE TWO

Click Next to get to the custom fields configuration page of the wizard.

On this page, you will define what information you want the user to enter. The process is similar to that of adding columns to a list or variables to a workflow.

Each piece of data you add has a name and a data type. Each data type in turn has a dialog box to support that data type. Feel free to explore the different data type dialogs by adding various fields.

After you are done exploring, add a field named Order total of type Currency. The dialog box for setting this field up is shown in Figure 43.

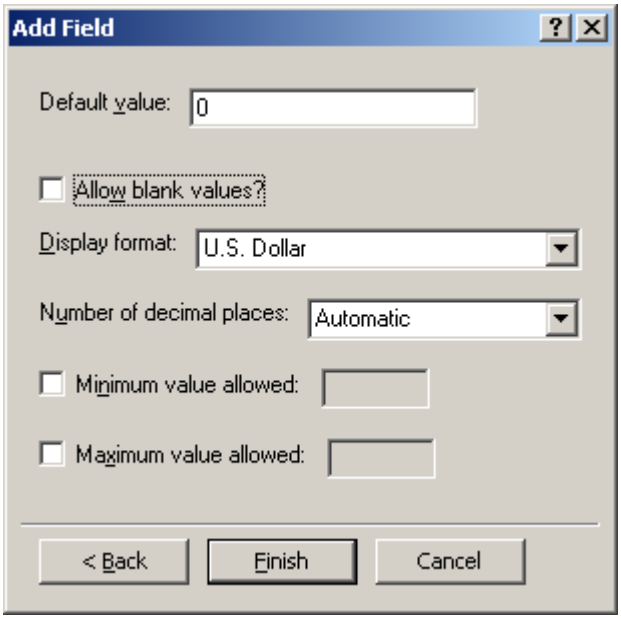



FIGURE 43. ORDER TOTAL

Hit Finish and make sure your custom fields page of the task wizard only has that field added, as shown in Figure 44before you click Finish to complete the first part of the action configuration.

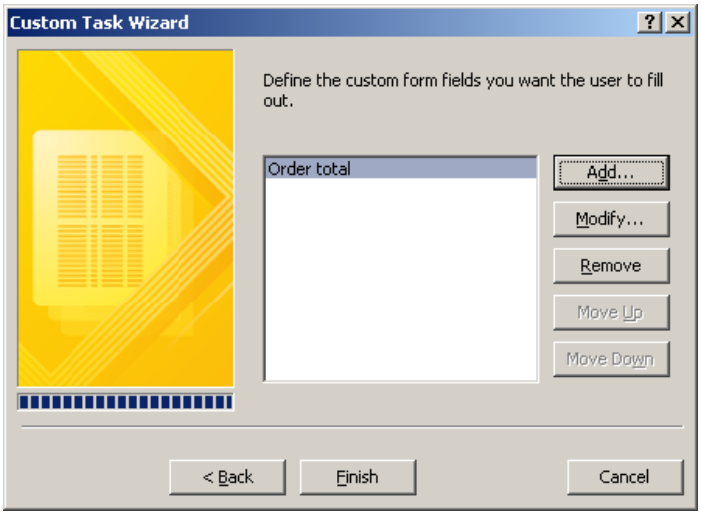



FIGURE 44. CUSTOM TASK WIZARD, LAST PAGE

Back in the Workflow Designer, click the *this user* text. The editor here is the same you saw for selecting the recipient of the email in the last exercise.

In a production environment, you would likely have more options available. For example, you may have dedicated SharePoint groups for various roles. If so, feel free to assign tasks to one or more person, role, or group. In my example shown in Figure 45, however, I have assigned the task to the administrator, to demonstrate how the task will behave.

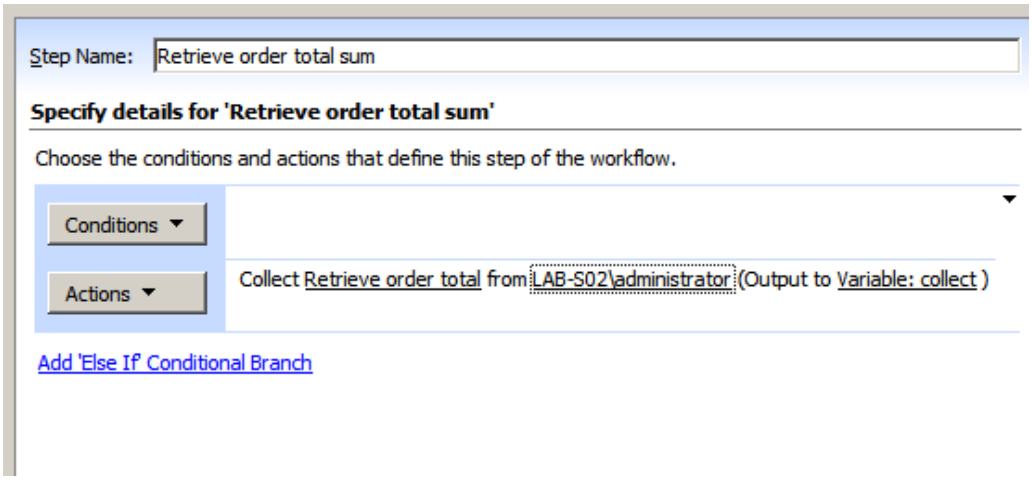



FIGURE 45. COLLECT DATA TASK CONFIGURED

Notice the (Output to Variable: collect ) text. This is very important. The Collect Data action you just created will create a new task in the default task list. The ID of that task will be stored in a workflow variable called collect. We will use this now to retrieve the data that the user assigned to the task eventually enters for the order total sum.

## Customizing Workflow Forms

First, however, I want you to take a short break, hit Finish to save your workflow, and look in the Folder List where you will now find a new .ASPX file named after the Collect Data action you created, as shown in fig.

If you double-click the .ASPX file, the file will open inside SharePoint Designer. You can customize the page and update the design, or if you are as skilled in web design as I am, leave it to someone else. Your designers will likely enjoy working with these aspects of your workflow far more than I do.

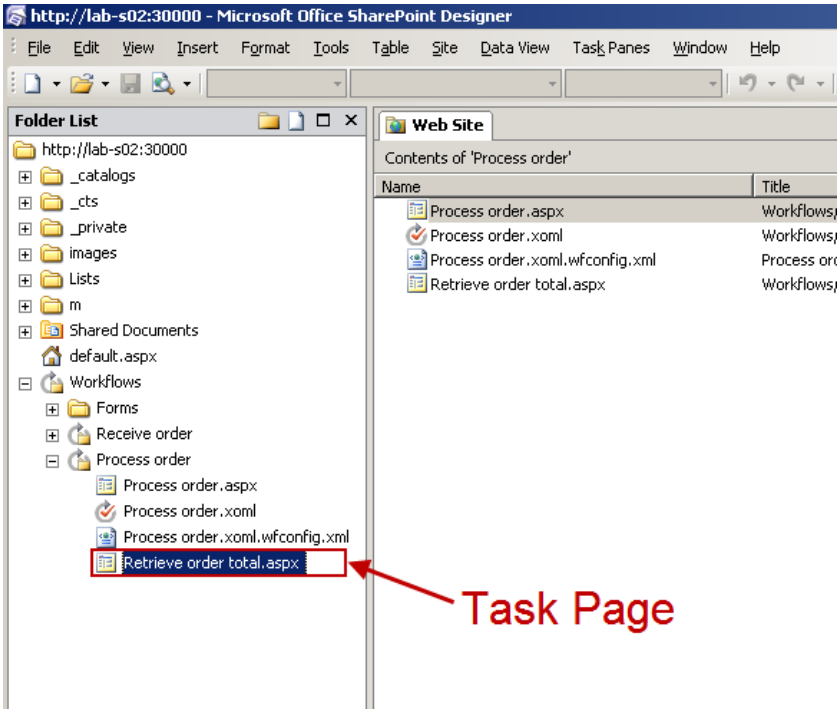



FIGURE 46. COLLECT DATA TASK PAGE LOCATION

The same also applies to the initiation form, by the way. The initiation form is the page shown when starting a workflow manually. You may decide to modify this page as well, for example to provide instructions for users initiating the workflow, as shown in Figure 47.

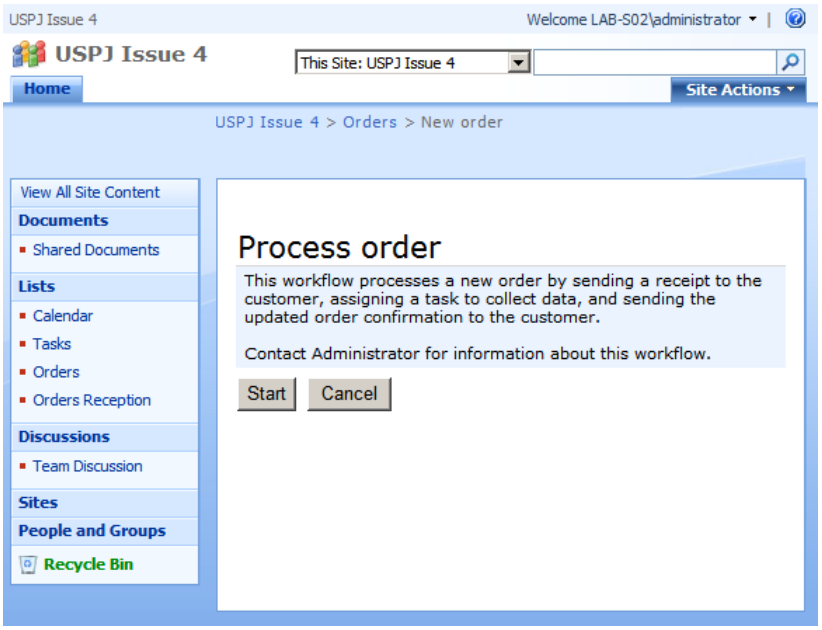



FIGURE 47. CUSTOMIZED ININTIATION FORM

Ok, let's move on to the next exercise, and use the data collected to update the order.

## Exploring Workflow Lookups

**Using Collected Data in Workflow**

The next task we need to accomplish is to update the order with the data the user entered. However, the data is stored in the task object, not in the workflow. The only reference we have is the ID of the task item, so we need to look up the correct item before we can retrieve the data.

Stick to the task branch where you added the Collect Data action. Click the Action menu to add a second action. This time, you want to select the Set Field in Current Item action.

The first parameter is very simple, you just need to select which column or field in the current item your want to update, which would be the Order sum column. Notice that you do not get a dialog box but rather just a drop-down menu listing the available columns in the Orders list.

The value parameter, however, require a bit more work. Once you click the *value* text you will get a text input box if you want to add a static value. However, we want to look up the value entered by the user for the Collect Data action, so click the Workflow Lookup button instead, as shown in Figure 48.

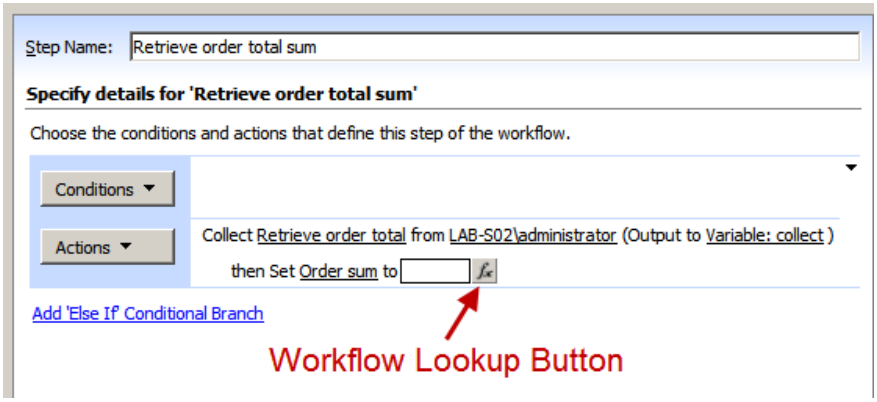



FIGURE 48. FIELD VALUE EDITOR

The Define Workflow Lookup should be familiar by now, but so far, we have only used the dialog to look up data in the current item. This time, we need to look up data in another list, so select the Tasks list from the Source drop-down. When you do, the dialog box changes, as shown in Figure 49.

**Note**

Remember that the ID of the task created by the workflow is stored in a workflow variable called collect, as you can see in the above figure.

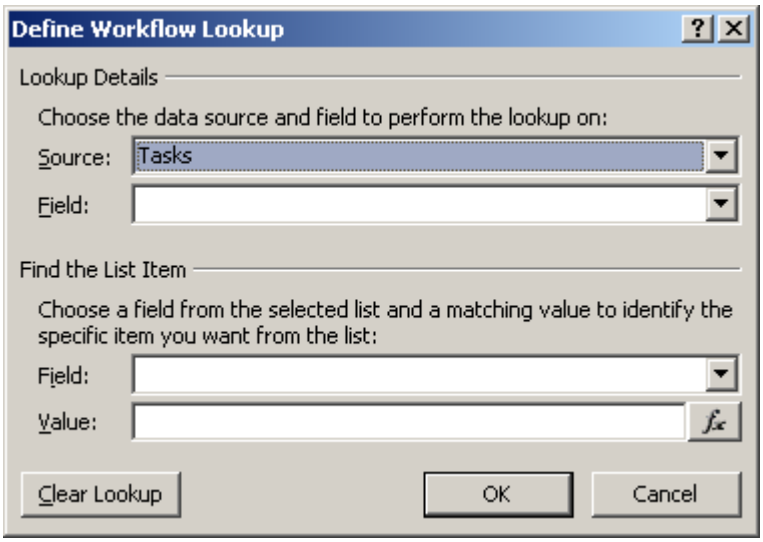



FIGURE 49. FOREIGN LIST LOOKUP EDITOR

The top two drop-down lists are familiar, and since we want to retrieve the Order total from the task, you should select that field in the Field drop-down now.

The second part of this extended dialog holds the means by which we will locate the correct task. The logic is to select an item from the Tasks lists where the Field matches a certain Value. The field we know is the ID, so select the Tasks:ID field from the drop-down.

Next, we need to define the value, which is stored in the workflow variable collect. So, click the Workflow Lookup button located next to the Value input box. This launches a second Workflow Lookup dialog.

The second Workflow Lookup dialog works exactly the same as the Workflow Lookup dialog we just left. This time, our source is the workflow variable *collect*, so select Workflow Data as the source, and Variable: collect as the Field.

Click OK to return to the previous Workflow Lookup, and make sure your dialog box matches Figure 50.

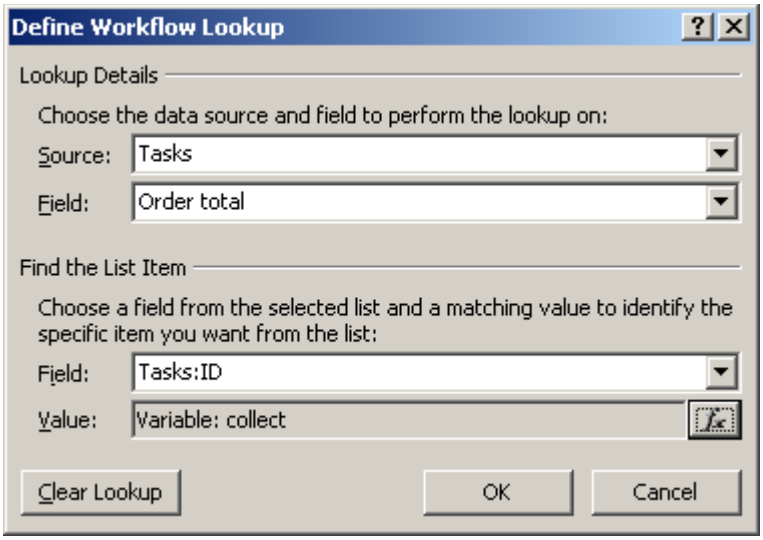



FIGURE 50. COMPLETED WORKFLOW LOOKUP

What you just witnessed, or actually performed, is a very powerful technique in SharePoint Designer workflows. You just created a recursive workflow lookup. Such recursive lookups can potentially span several levels and be incredibly complex, but once you understand how the recursive lookup works, you will be able to look up data anywhere in a SharePoint site and use as part of your workflow.

Feel free to retrace your steps here to make sure you understand fully what you just did. Doing so will greatly increase your skills as a SharePoint Designer workflow developer.

## Workflow Review

Before we to the final step of this chapter, a step you will take on your own, let's see how the workflow behaves so far. Hit OK in the Define Workflow Lookup dialog box, and hit Finish once you return to the main Workflow Designer window.

Next, go to the Orders list of your SharePoint site, and start the workflow you just created. Refer to Chapter 1 for information on how to do this if you need a refresher.

Once the Process order workflow starts, SharePoint sends an email to the originator of the order email. My sample order came from the invalid domain example.com, so the email will just drop dead in the outbound queue-folder of my mail server. In Figure 51, I have opened the outbound email in Notepad so you can see that the text is indeed correct, with the order date added to the body.

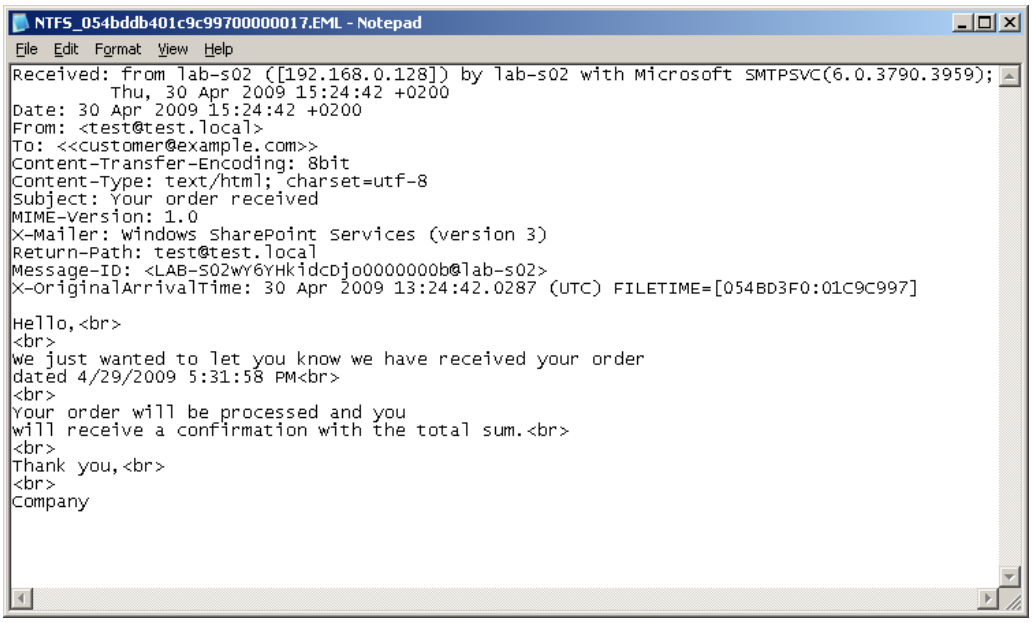



FIGURE 51. OUTGOING EMAIL

Note that the email is formatted as an HTML email. Changing this is not possible in SharePoint Designer with the default email action.

Next, go to the Tasks of you SharePoint site, and note the new task created, as shown in Figure 52.

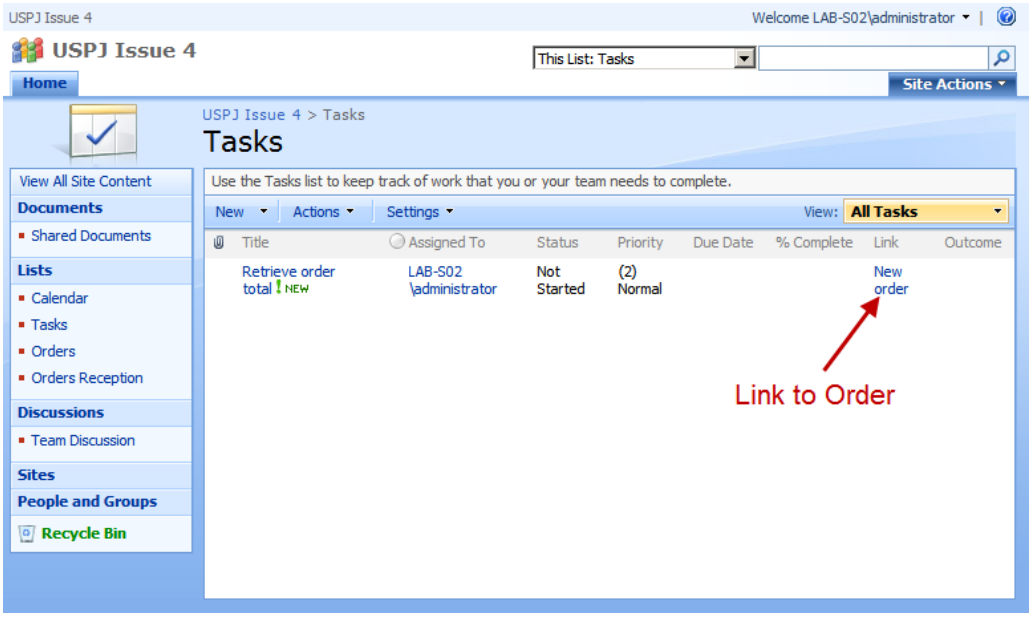



FIGURE 52. TASK CREATED

If you click the link to the order, you will jump to the order, to review what the customer wanted to buy. If you open the task by clicking the title, you will see the task, like any normal task in SharePoint.

However, note also that the task has a column added for the Order total. To complete this task you click the Edit Item link shown in Figure 53, which will lead you to the collect data task page created by SharePoint Designer as shown in Figure 54.

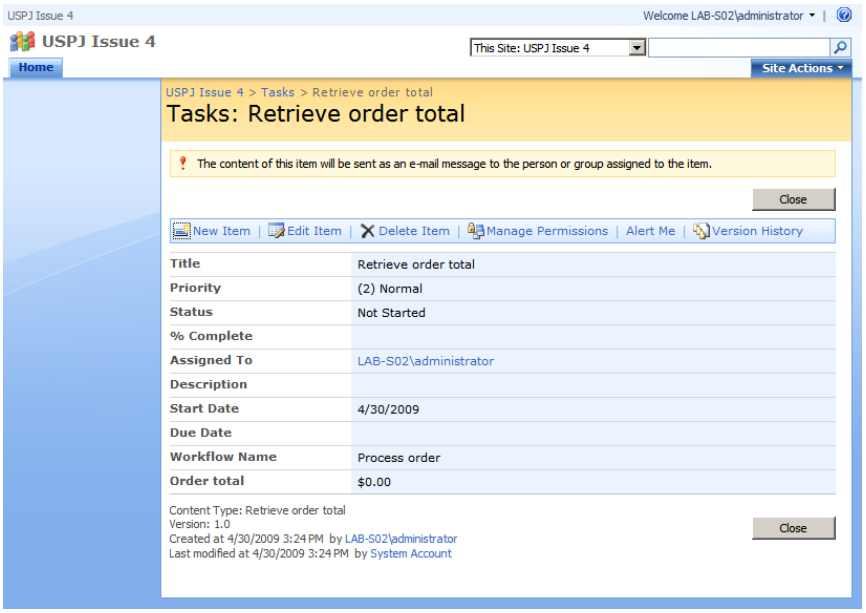



FIGURE 53. RETRIEVE ORDER TOTAL TASK DISPLAY PAGE

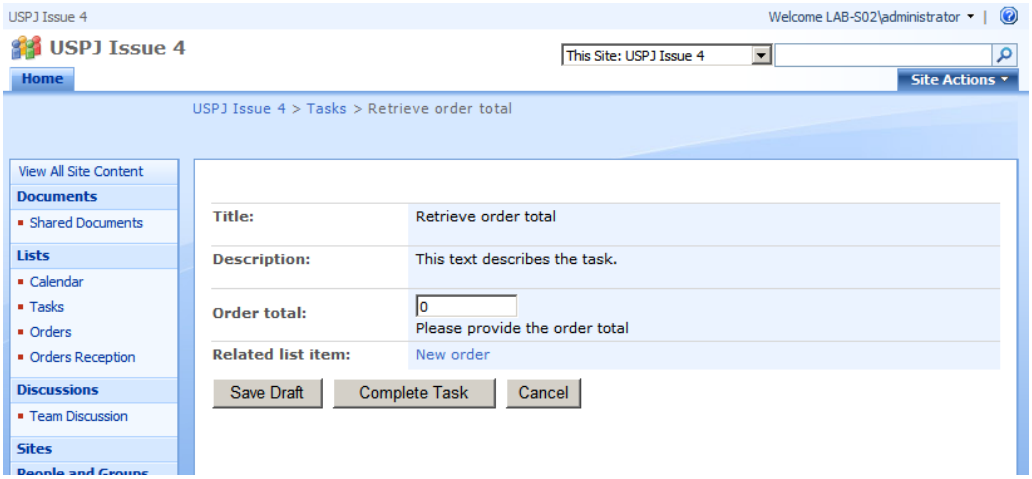



FIGURE 54. RETRIEVE ORDER TOTAL TASK EDIT PAGE

To complete the task, fill in the Order total number and click Complete Task. Once you do, you can go back to the order from the link in the tasks list and verify that your Order sum displays the correct value, as shown in f.

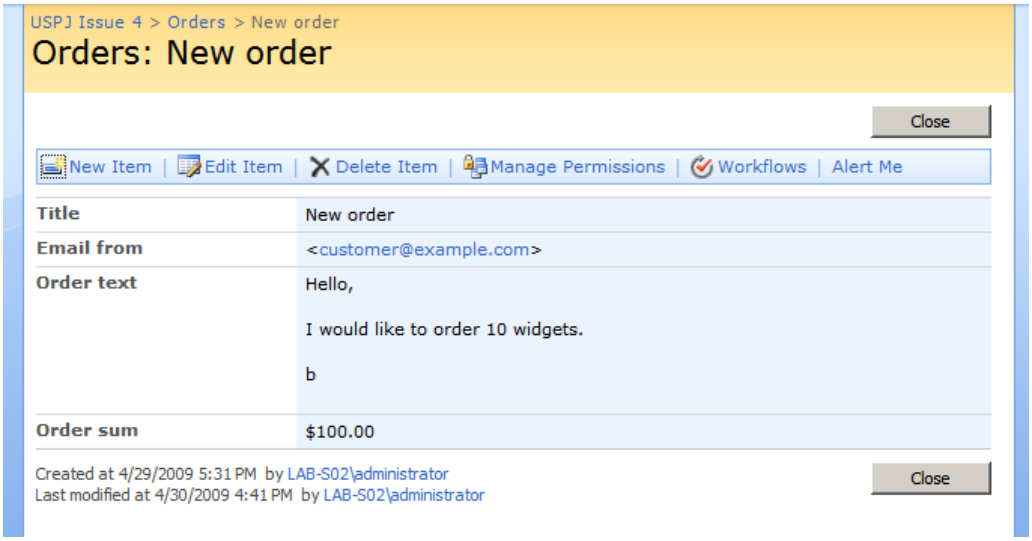



FIGURE 55. WOHOO!

Finally, you may want to review the workflow history, accessible from the Workflows link. Click on the Process order link under Completed Workflows to review. Figure 56 shows my result.

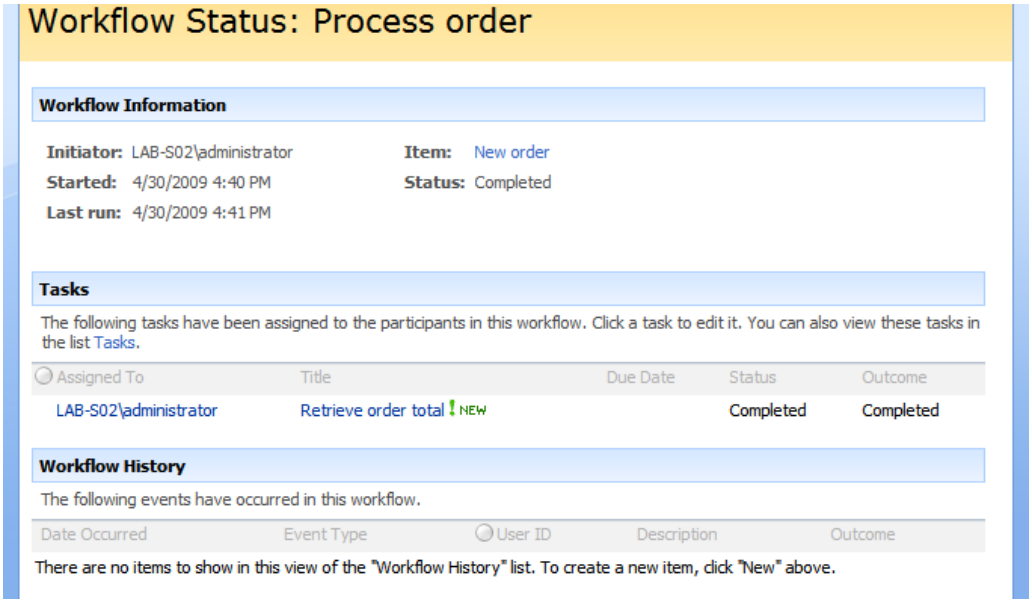



FIGURE 56. WORKFLOW HISTORY

As you can see in the figure above, the Workflow History is empty. Some default actions will log information to the workflow history, but not all. In the next chapter, we will augment our workflow to include logging of events.

**Note**

When a workflow creates a new task, the workflow pauses until that task completes, regardless of how long that takes.

Before we move on, review the following questions to make sure you grasp the key concepts from this chapter.

### ✍ REVIEW QUESTIONS

? 1.

? 2.

? 3.

## ✍ REVIEW QUESTIONS ANSWERED

- 1.
- 2.
- 3.

Chapter 5

# Following up on New Orders

*People like attention… So give it to them!*

We are nearing completion of our workflow, but I would still like to show off some interesting concepts.

In this chapter, we will follow up on the order and send a second email to the customer. Then, after a certain period, we should follow up the customer and make sure our handling of the order was satisfactory. This will introduce us to a few new concepts, although the sending of email was introduced in Chapter 3

## Returning the Order Confirmation

The first step of this chapter is to send an order confirmation to the customer. We sent an email as part of the last chapter to avoid repeating everything here I will keep this simple. However, I do want to introduce you to a few new actions and provide you with an exercise for you to complete on your own.

In this step, I want to show you how to build dynamic strings in SharePoint Designer workflows. Dynamic strings are useful in a range of situations. For example, the email subject of an email can be either a fixed, static string or a workflow lookup value, but you cannot combine the two. So, if you want to include a text such as 'Your order, dated %date%' where %date% is replaced by the order received date, you are out of luck.

That is where dynamic strings come into play. Using dynamic string, you can create a flexible string from bits and pieces of runtime information, and combine those bits and pieces with static text to create a new variable that you can then use, for example, as the subject of an email.

Let's see how.

**Build Dynamic String Action**

Start by creating a new step in your Process order workflow. Name it something like 'Send order confirmation'.

Add a new "Build Dynamic String" action to the step and then click the text *dynamic string* to open the String Builder editor. The String Builder is similar to the editor used for the body of an email, so you shouldn't have too many problems understanding what goes on.

In the text area or the editor, enter 'Your order received ', and make sure to include the last space before you hit Add lookup.

In the Define Workflow Lookup dialog, make sure you select Current Item as the source and select Created as the field. Finally, hit OK to close the lookup and ensure that your String Builder looks like Figure 57.

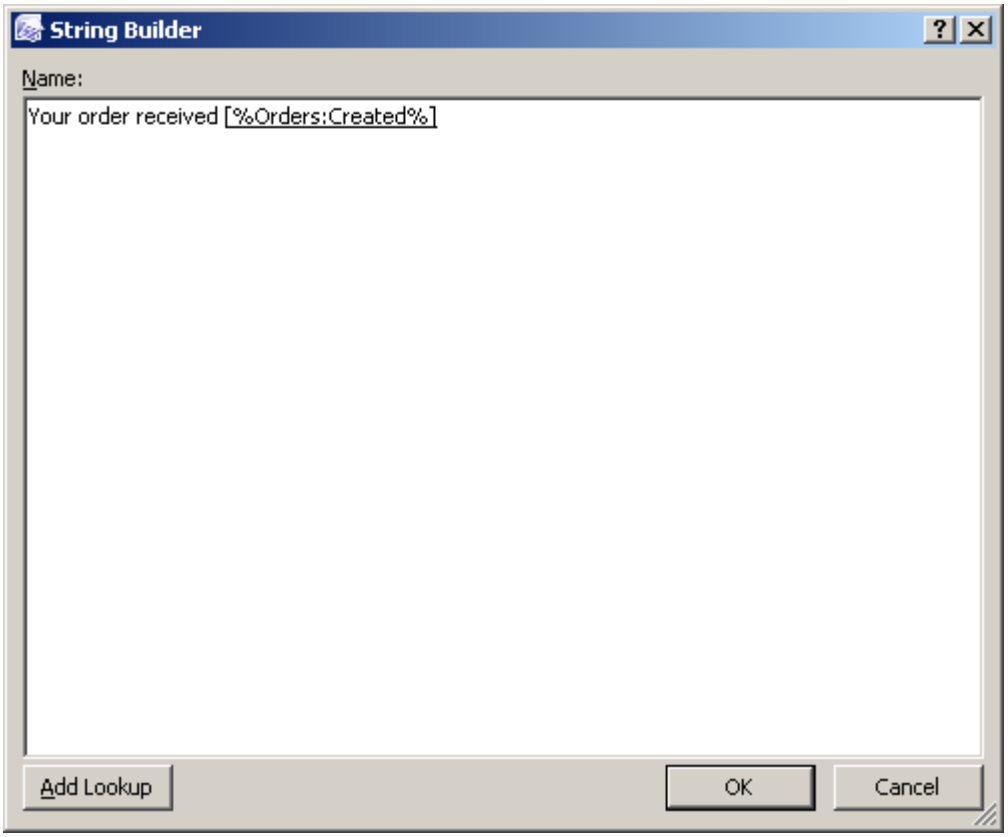



FIGURE 57. STRING BUILDER

If everything looks OK, then say so by clicking the OK button to return to the main Workflow Designer window.

Note that the action states it will store the string in a variable called variable as shown in Figure 58. Such a name isn't really good, since it tells us nothing about what the variable does or contains. In complex workflows, you may need to work with a range of variables, and remembering after 6 steps what the variable named 'variable' contains is difficult. Consider also how well you will remember this after 6 months when your boss drops by and asks you to change the workflow, or when someone else needs to work with the workflow.

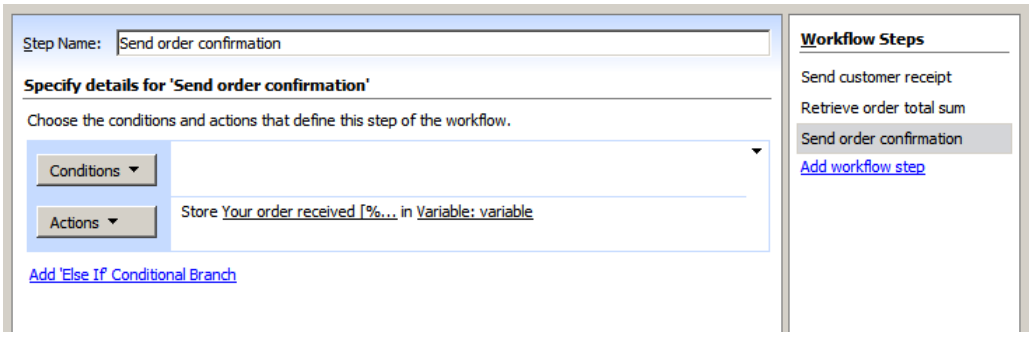



FIGURE 58. ALMOST COMPLETE DYNAMIC STRING ACTION

You can change all such variables, however, in the Workflow Local Variables dialog box shown in f. To open this dialog, click the Variables button in the lower part of your Workflow Designer. You will see two variables used in your workflow, the collect variable, created during the Collect Data action in the last chapter, and your new variable string.

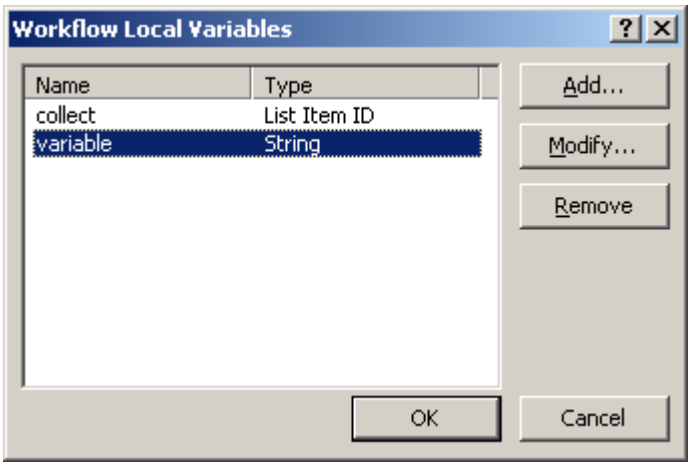



FIGURE 59. WORKFLOW LOCAL VARIABLES

To change the variable name, select the variable and click Modify. You can then change the name and type of the variable into something a bit more descriptive, such as 'Confirmation subject'.

**Caution**

Note that if you change the name of a variable that is already in use, the workflow steps using the variables will not be updated with the new name. For example, if you update the collect variable name to "Collect data task ID" as shown in f, the previous step in your workflow is not updated to reflect the change, as shown in f.

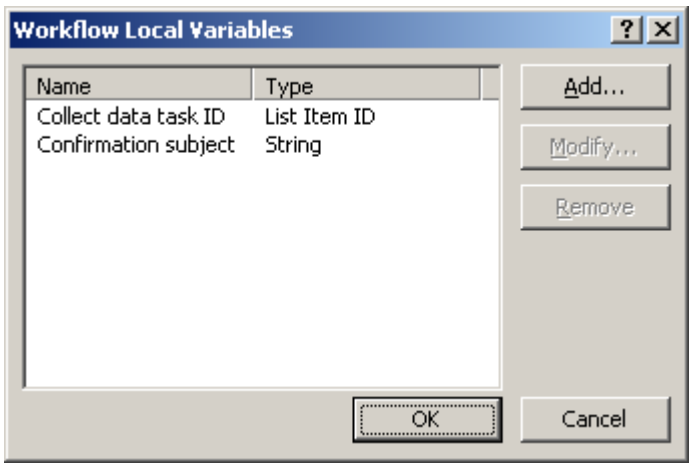



FIGURE 60. UPDATED VARIABLE NAMES

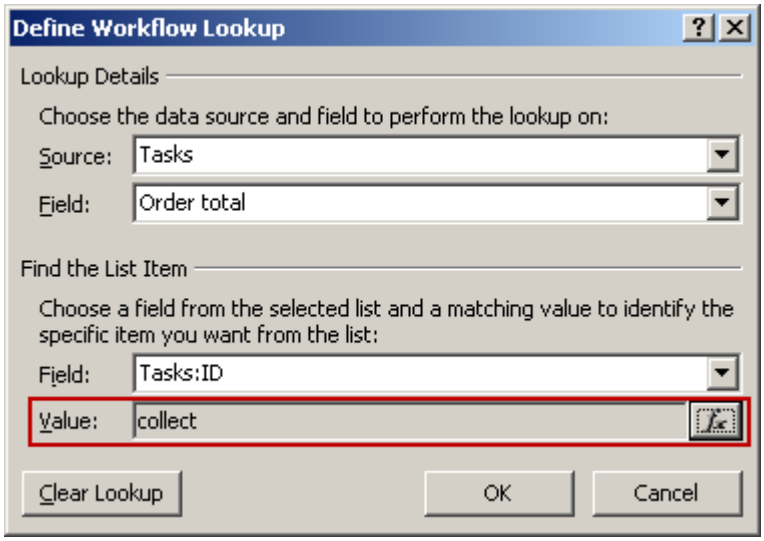



FIGURE 61. BAD VARIABLE!

If you change variable names you should click the Check Workflow button in the lower left part of your Workflow Designer window. This will give update the workflow and place an icon wherever there are errors in the workflow, including references to old variable names, as shown in Figure 62. Within each step, non-existent variables will be enclosed in **.

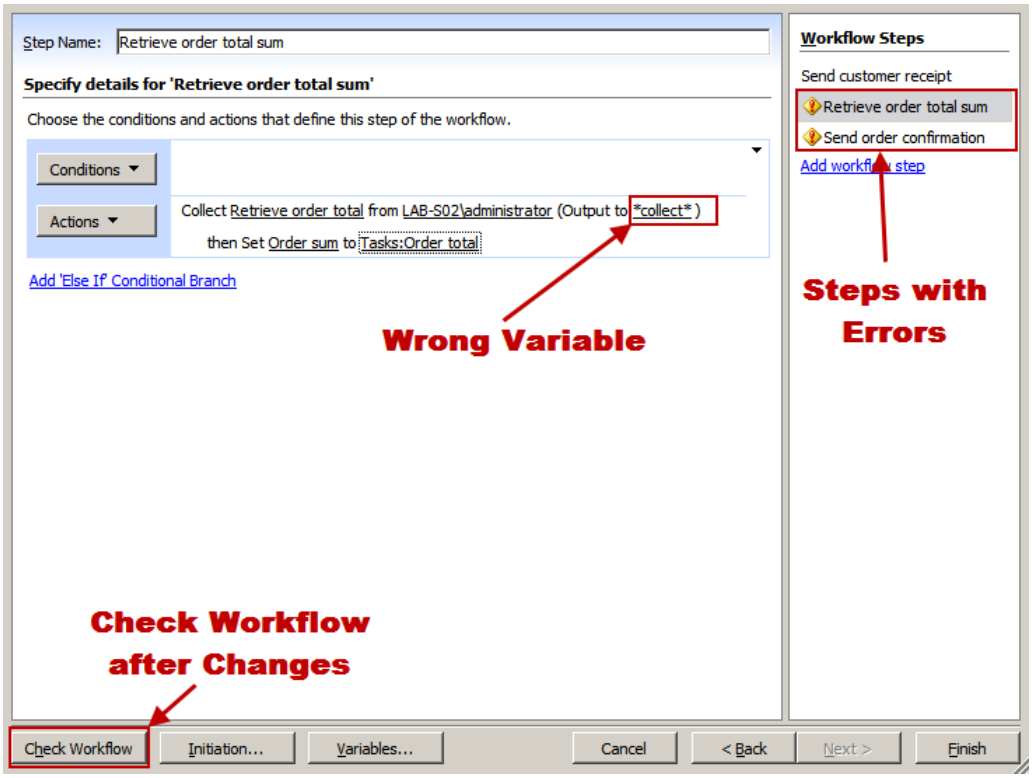



FIGURE 62. WORKFLOW ERRORS

To fix these errors, simple go through the steps and click the old variable names. A drop-down will list available variables, and you can simply select the correct variable directly from the task editor, as shown in Figure 63.

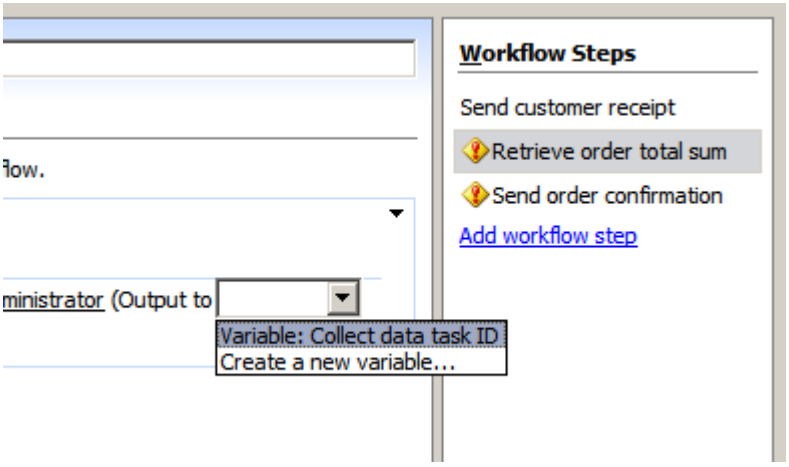



FIGURE 63. AVAILABLE VARIABLES

Some variables are not directly accessible through the task editor, such as when we are using a variable as part of a workflow lookup, and for these variables you need to modify the names through the appropriate editor.

Once you have corrected all errors, click Check Workflow again to ensure that no further errors exist.

**Sending Order Confirmation**

Now, here's a challenge for you. For the next exercise I will not tell you the steps you need to take, but simply tell you what you need to accomplish. You will need to figure out which steps to take yourself.

What you need to do is set up the rest of this step to send an email to the customer with a confirmation that the order has been processed, and include the order total as part of the message.

Using what you have learned in this and the previous chapters, create an email task to send a confirmation message to the customer. The resulting email should resemble Figure 64.

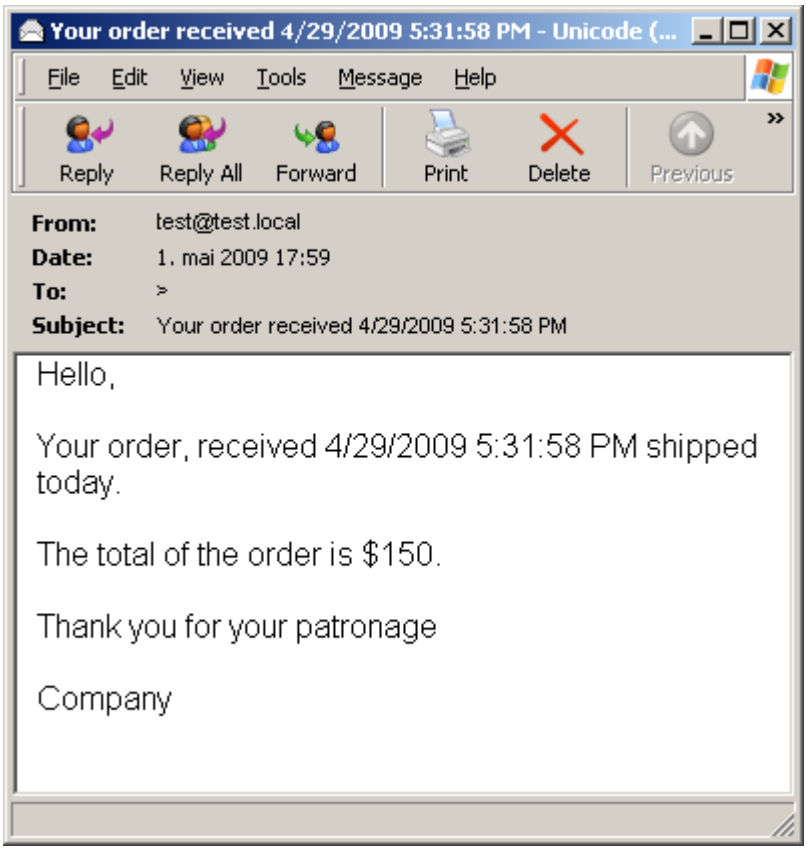



FIGURE 64. RESULTING EMAIL

Have a go at it, and don't give up, I am sure you can get everything setup correctly ☺

## Pausing a Workflow

For this next trick, I will teach you how to control the very flow of time itself. Well, not quite, but I will show you how to control your workflow execution by pausing the flow of steps for a period or until a date is reached.

> **Note**
>
> I will assume you have completed the previous exercise, but if you haven't, it will not affect the exercise. Simply imagine the correct email being sent.

A trick I learned from an online retailer who wanted customers to be reminded about their order close to receiving the goods is to delay the "order shipped" message until just before the order would actually arrive.

**Pause Workflow for a Period**

I am not condoning such trickery, but it makes a good exercise for us. What we will do now is delay the "order confirmed" message for one day after we have actually processed the order.

The first thing you need to do is to add a new step to your workflow. Name the step "Pause for one day" or something similar.

Of course, when you add a new step, the step will appear at the end of all the steps, and this is not where we want to pause. So, the next thing you need to do is move the step to the correct position in the workflow.

To do so, hover your mouse pointer over the newly created step, and open the menu that appears, as shown in Figure 65.

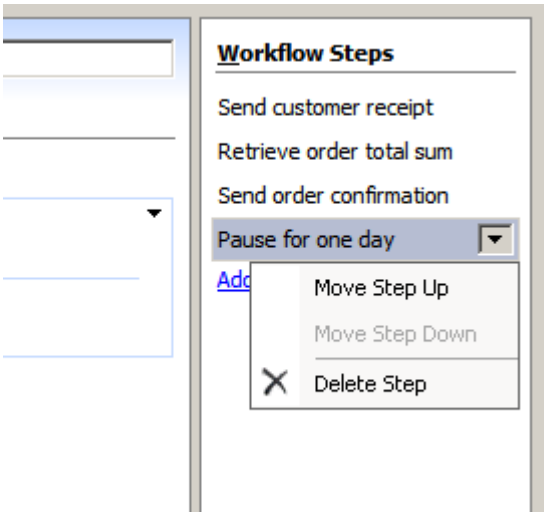



FIGURE 65. CHANGE WORKFLOW ORDER

Click the Move Step Up option to put the pausing step in the right position. You could also have moved the Send order configuration step down to achieve the same effect.

Next, add a Pause for Duration action to the new step. The task editor will let you enter values for days, hours, and minutes. Set the number of days to 1 and the other values to 0 as shown in Figure 66.

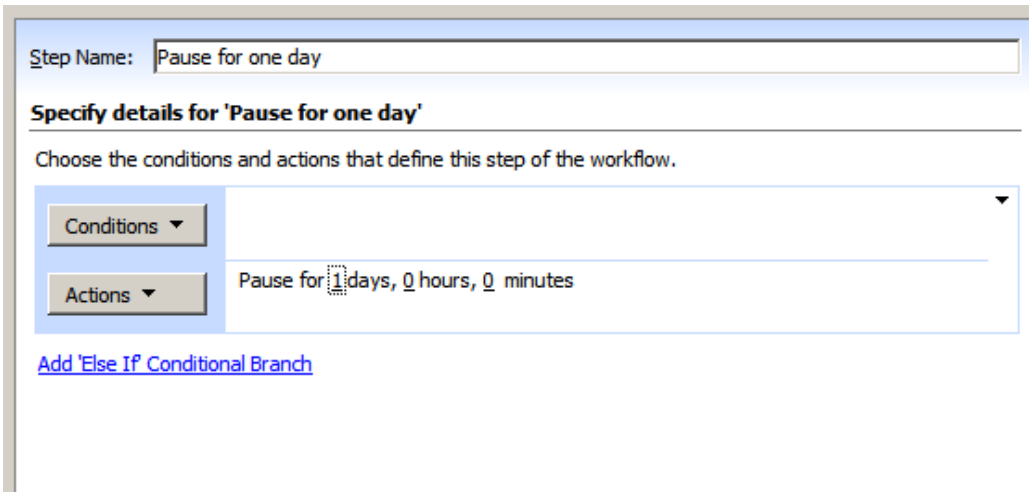



FIGURE 66. DELAY ACTION CONFIGURED

Feel free to test your delay by setting a shorter duration, for example 5 minutes. The thing you need to understand about delays in workflows is that they are only processed with a resolution of about 5 minutes. This resolution is controlled by a timer job in SharePoint that runs every 5 minutes.

So, setting a short delay may not be accurate, but be offset by 1-4 minutes.

A Pause for Duration action is one of the actions that logs information to the history log. If you start your workflow now, and then enters the workflow status page on your site, you will see a message like figure Figure 67.

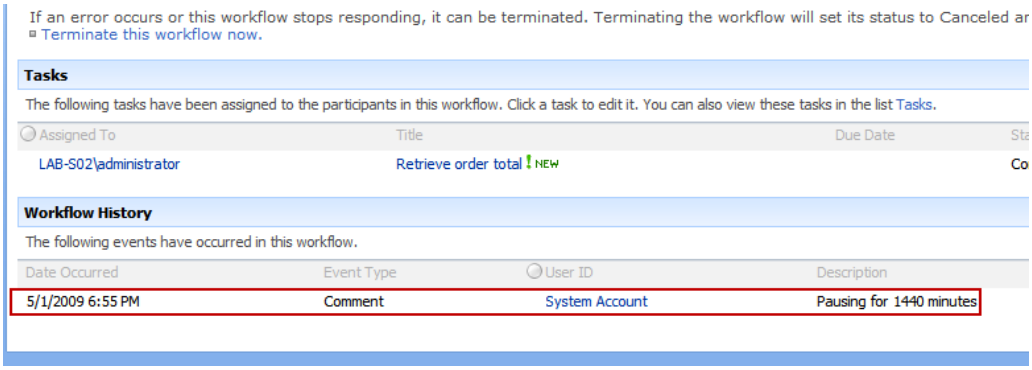



FIGURE 67. PAUSED WORKFLOW

Your other option for pausing a workflow is to pause until a certain date is reached. We won't explore this in an exercise, but for the sake of completeness I would like to explain briefly how that action works.

The action for this is called Pause until Date, and allows you to either set a fixed date or use a workflow lookup value. The latter becomes a powerful option if you combine other actions. You can set a workflow variable to the create date of an item using the Set workflow variable action, and then use the Add date to Set Time Portion of Date/Time Field action to add a period of time to that variable.

## Branching

I want to show you another useful option for improving your workflows. Let's say you want to handle large orders in a different way than smaller orders, and notify a Key Account Manager if the order is larger than $10,000.

To do so you can use conditional branching, and I will show you how now.

Add a new step to your workflow called 'Optionally notify KAM'.

From the Conditions button, select "Compare Orders field". The name Orders reflect the name of the list, so if you used a different list name, that part of the option will differ.

Next, select Order sum from the field drop-down menu. Then, click the text *equals* and select the 'is greater than' option from the comparison drop-down list, as shown in Figure 68.

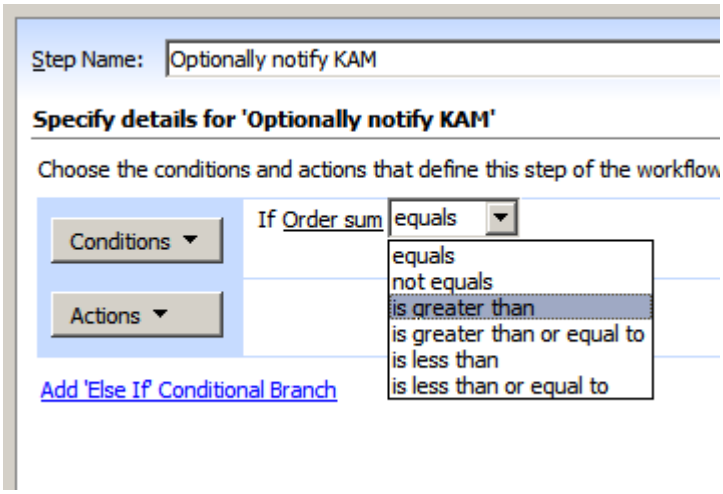



FIGURE 68. COMPARISON DROP-DOWN

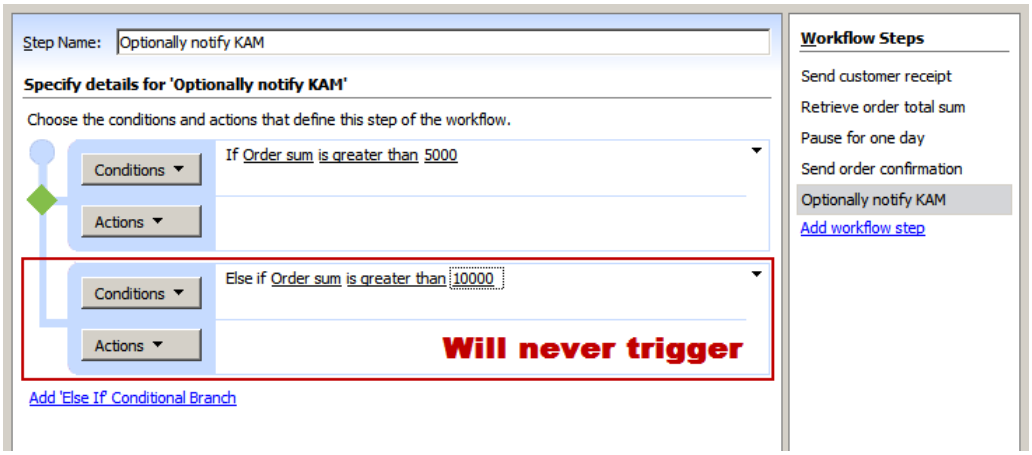
The content of the comparison drop-down changes with the data type of the column selected as the field. For example, had you selected a text type field you would see options such as 'starts with' and 'contains' that makes no sense in number fields such as the Order sum.

Finally, enter a value in the value parameter of the comparison editor. In my example, I have used 10,000. Note that you must enter the value without commas. This concludes the branch condition.

We won't add any notification action, as this would be an email, and I think we have created enough email actions for now. However, in the next section we will use logging to log that we enter the correct branch.

You may want to add additional branches, however. For example, if the order is above $5,000 but below $10,000, you may want to send a special thank you message to the customer.

When you add multiple branches to a step, you must pay attention to the logic and in which order you define the branches. Each branch is exclusive inside the same step, so if you set the first step to trigger on 'Order sum is greater than $5,000' and the second branch to trigger on 'Order sum is greater than $10,000' then the second branch will never trigger, even if the Order sum is greater than $10,000. F shows this.




Remember that you can click the down-arrow in the top right of a branch to add a new branch. You can also click the Add 'Else If' Conditional Branch link at the bottom of your current branches.

You can circumvent this problem by adding a second condition to the first branch. Just click the Conditions button again and add a new 'Compare Orders field' condition. When you do, you will get a second row in the condition section, as shown in Figure 69.

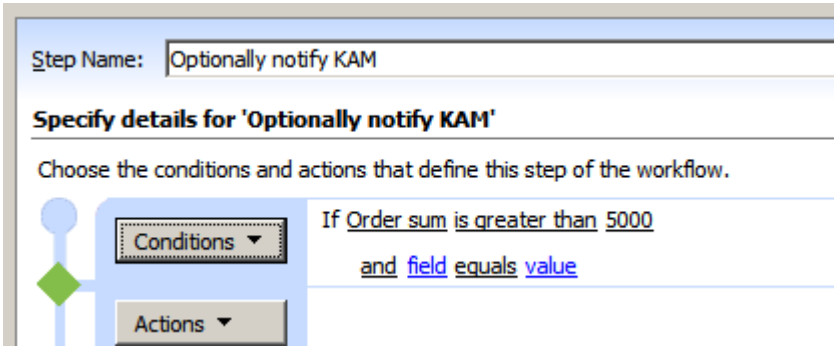



FIGURE 69. SECOND CONDITION

The text *and* is clickable, and clicking changes the condition to or. A second click changes the condition back to and.

Set the Order sum as the field, change the comparison to 'is less than or equal to' and set the value to 10000, as shown in Figure 70. Now, your second condition will trigger if the Order sum is greater than 10000.

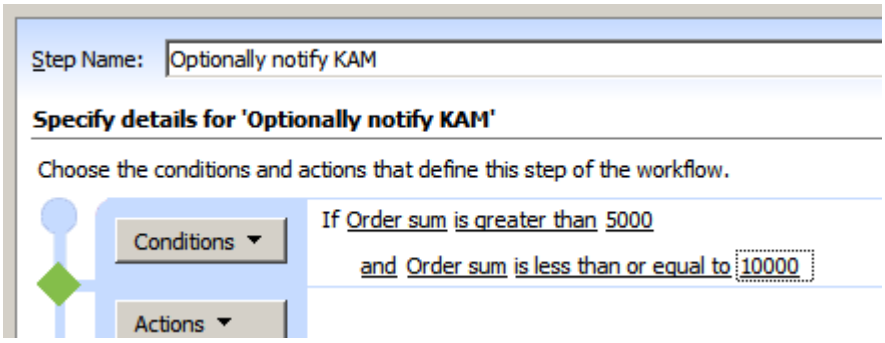



FIGURE 70. COMPLETED CONDITION

Great, your condition is set up, so let's add a default branch as well.

If you have conditions in all of the branches you can optionally add a branch with no conditions at all. If none of the other branches trigger, the branch with no conditions will trigger by default. So, add a new branch, but do not add any conditions.

Now we should complete the workflow by adding a bit of logging information.

Yes, I know we haven't added any actions to the branches. If you are not tired of setting up email actions, feel free to add a few messages to the branches to test.

## Logging Events to Workflow History

The final task I want to show you is how to log information to the workflow history. This is very useful for tracking what goes on inside a workflow, especially if you are using complex workflow control, such as branching and conditions.

SharePoint has a dedicated action for this purpose called Log to History List. Add one such action to each of the branches in your 'Notify KAM' step. Oh, and if you like, rename the step as well, perhaps to 'Special order events'.

For each of the Log to History actions, enter a descriptive phrase that will be logged to the workflow history depending on the Order sum. Do not forget to add a default message as well to the branch with no conditions.

Figure 71 shows my setup and my sample messages.

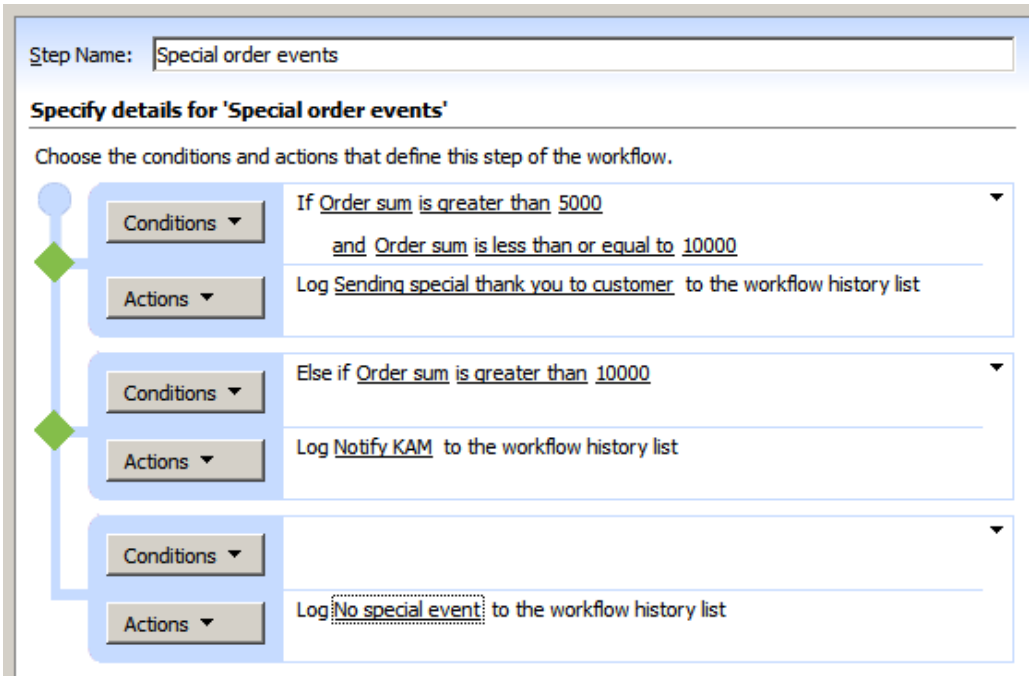



FIGURE 71. BRANCHES AND MESSAGES SET UP

Once you now click Finish, your workflow will be complete, and you can test your marvel.

Try running the workflow with varying values for the "Order sum". In Figure 72, I have run the workflow three times with varying order sums.

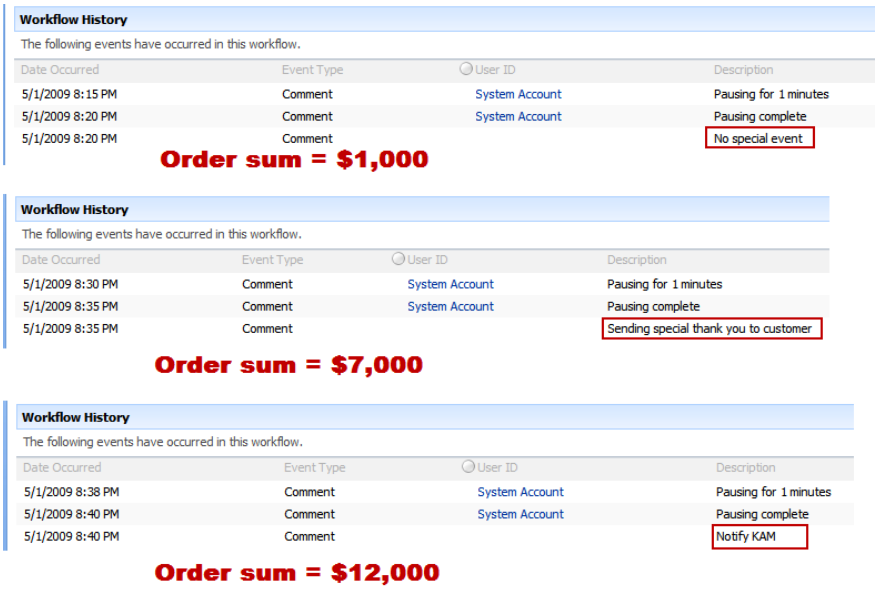



FIGURE 72. CONDITIONAL RESULTS

Adding logging actions is very useful and definitely a best practice. By logging important events, such as the result of branching or the progress into new steps of the workflow, is useful both during development and during debugging.

I think we have explore enough for now, but don't touch that dial, in the last chapter I will show you some additional information that you may find useful.

First, though, there are some review questions.

### ✍ REVIEW QUESTIONS

? 1.

? 2.

? 3.

## ✍ REVIEW QUESTIONS ANSWERED

- 1.
- 2.
- 3.

Chapter 6

# Where to Next?

*Some additional concepts you may find interesting*

While you have now been introduced to SharePoint Designer workflow development and should be capable of creating your own workflows, you will quickly find limitations in the default actions available in SharePoint Designer.

Even so, you do have at your disposal a set of building blocks in the form of actions that can accomplish impressive results when combined and used somewhat… creatively.

In this chapter, I will demonstrate some such applications, using the default actions to accomplish tasks that are otherwise not possible.

## Loops in SharePoint Designer

One question pops up in many forums, emails, and articles on SharePoint Designer workflows. The question is: Can you make SharePoint Designer workflows loop through all items in a list?

The short answer is: You can't.

The longer answer is: Sure, no problem.

The thing about making loops in SharePoint Designer is that you need to think outside the box. Before we start, however, I would like to make the following very clear:

**Caution**

**Caution**

What you are about to see should never be attempted at home. The stunts performed here are done by professional SharePoint geeks in a controlled lab environment. Performing these stunts in your own environment is likely to make your administrators or operations people very angry.

The following experiment has serious weaknesses, and is likely never usable in a real-world scenario. I will get back to these weaknesses later in this chapter; I don't want to ruin your joy of learning and experimenting.

**Looping Through Items in a List using SharePoint Designer Workflows**

With that caution in mind, I'll demonstrate how you can utilize some of the peculiarities of SharePoint Designer to, well, go where no SharePoint Designer has gone before.

On such peculiarity is the way workflow lookups behave when you are searching for an item using a non-unique value in a column. For example, if you were to look up the ID of items where a Boolean property Processed is set to No, you may set up your workflow lookup as shown in Figure 57.

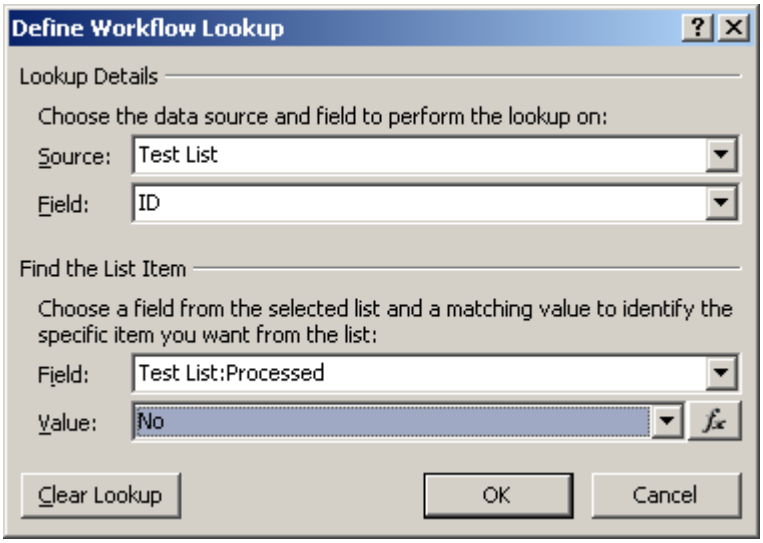



FIGURE 73. MULTIPLE RETURN VALUE LOOKUP

Now, what would happen if multiple items in the Test List had the Processed column set to No? When a lookup finds multiple items, the lookup returns only the first item found. You will be warned by SharePoint Designer, though, with a warning as shown in Figure 58.

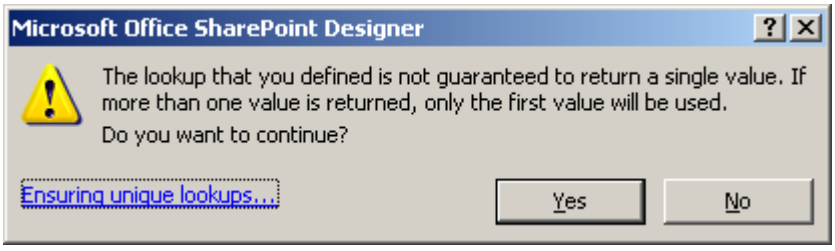



FIGURE 74. POSSIBLE MULTIPLE RESULTS WARNING

We can use this fact to our advantage.

Start by creating a list based on the Custom List template. To that list, add a new column of type Yes/No and name the column something like 'Processed' and make sure you set the default value to No. Then, add some sample data, but leave all the Processed check-boxes off.

This concludes the list preparation for our workflow.

In your production envi… Sorry, strike that.

In your lab environment, since you are wise enough never to perform anything like this in a production environment, you would likely have very different data. The important thing here, though, is that a Yes/No column is present, as we will use this to check the progress of our loop.

Start a new workflow and attach it to your new list. Set the start options to both manual and automatic on change.

For the first step of this workflow, you can just add any action you want to happen for each item in the list. For example, as I have done in Figure 59, just update the title by adding '(Done)' to the existing title, if it is not already part of the title.

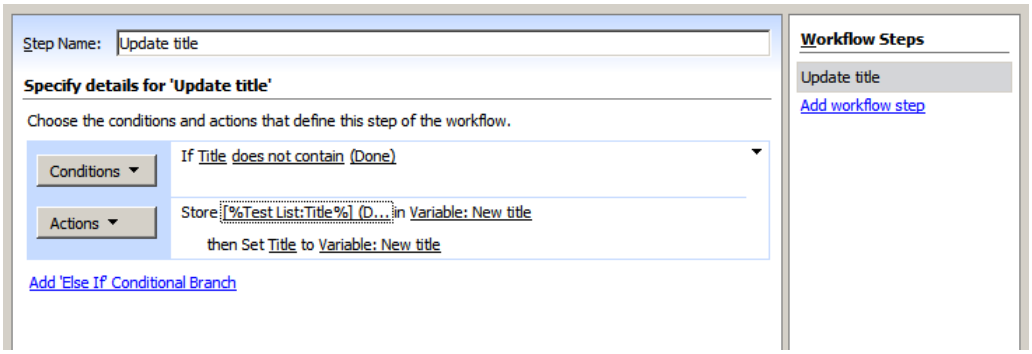



FIGURE 75. STEP 1, WHAT YOU WANT TO DO

Hint: In this step, I have used the Build Dynamic String action to store the current item's title along with the string '(Done)' in a new variable called *New title*, and then I have used the Set Field in Current Item to set the Title column to the *New title* variable.

Next, add a new step, called 'Update process status'. In this step, add an Set Field in Current Item action, and set the Processed column to Yes, to ensure that the same item is not processed repeatedly. The step is shown in f.

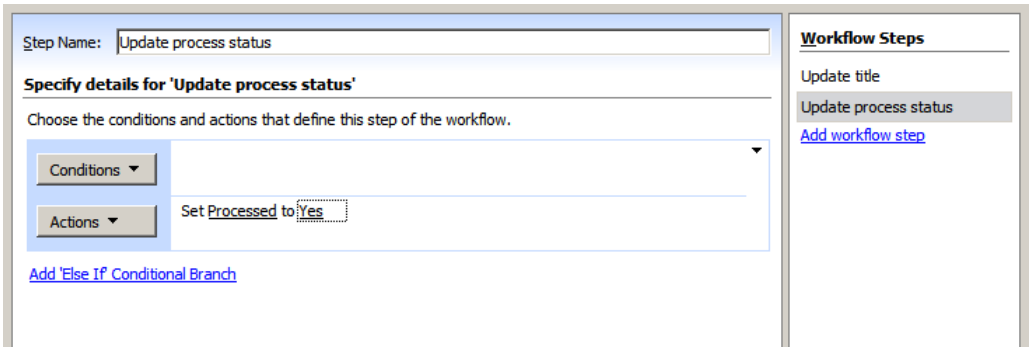



FIGURE 76. UPDATE PROCESS STATUS STEP

Your second step is complete, but we still need one more step.

Add a new step called 'Launch next item update'. If there is a trick to this workflow, and I'm not saying there is, it is in this step. Here we will update the first item in the list where the Processed column is set to No. Since we just set the Processed of the current item to Yes, we know that this item will not be updated again.

Add a new Update List Item action to your new step. Click the *this list* text to launch the Update List Item dialog.

We want to look for the first item in the Test List having Processed set to No, so select the Test List from the List drop-down.

You do not need to actually change any column or field in the item, but you need to locate the correct item. In the Find the List Item, select Test List:Processed and then No as the value.

The purpose here is to utilize the fact that any update will trigger this workflow again. By updating the first item having Processed set to No, right after ensuring that the current item has Processed set to Yes, we are in effect updating the *next* item in the list.

Hit OK and then hit Yes to confirm that you know the lookup will return multiple values. After all, we are depending on this for our workflow to work. Finally, hit Finish to save and deploy your new workflow.

Go to your Test List, click any item and then start your workflow. If everything works you should see that all the items in the list will progressively receive the (Done) text added to the title, as shown in Figure 61.

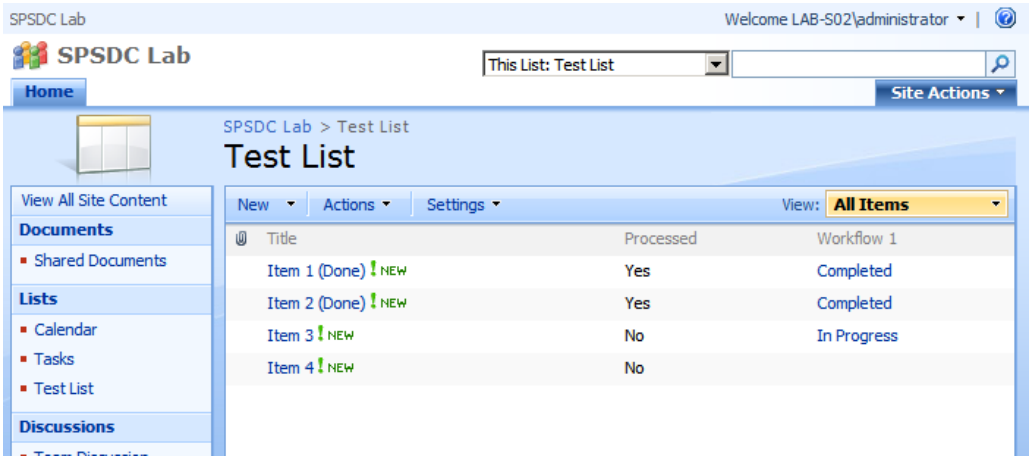



FIGURE 77. LIST ITEMS UPDATING

Now, I know your mind boggles at this point from the sheer thrill of the power you have now learned how to master. Therefore, I feel it is my duty to pour some cold water into your bloodstream.

First, I didn't show you the final result after all the workflows were completed. Figure 78 shows that result, and pay particular attention to the last item in the list.

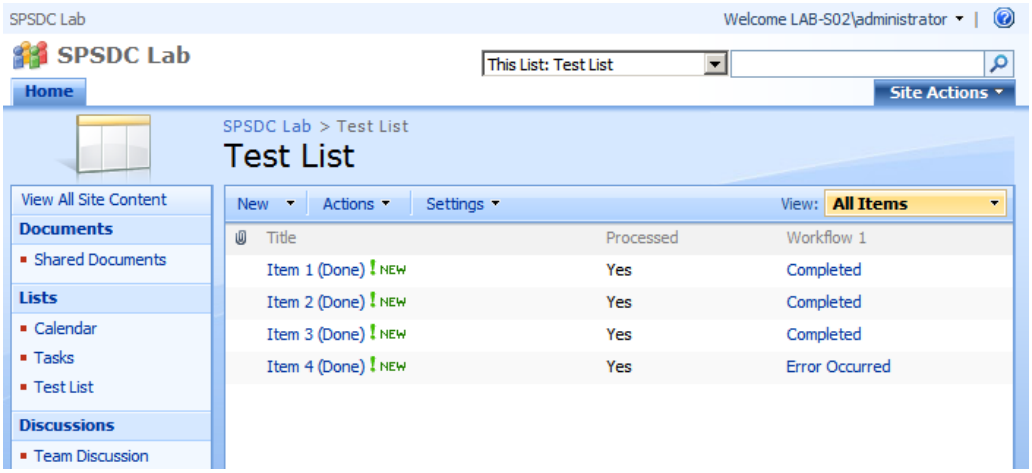



FIGURE 78. WHOOPS!

The error that occurs here happens because when no more items having Processed set to No, the workflow will fail because it does not find the item for which it is searching.

Note that the workflow will still complete its main task in step 1, and only fail in the last step of the last item on the list.

The second problem appears when you try to make any change to any of the items in the list. Since the workflow needs to spawn on any change to an item, and still fail from lack of unprocessed items, your list would quickly be populated by erroneous workflows.

The simple solution to both these problems is to delete the workflow after it has performed its mission. Consider this technique usable only for one-off updates to a list.

A more complex, but similar solution, is to use what is known as a secondary workflow. A secondary workflow is a workflow that is triggered by actions performed in one workflow, called the primary workflow. An example of this is if you had a separate workflow that was set to start automatically on any new item in the tasks list, and then created a new task.

You can utilize a secondary workflow to keep track of items in the primary workflow list and accomplish a similar result as in the above example. The added benefit of the secondary workflow method is that you really need just one item to handle all the primary workflows, and that your primary workflows to not fail.

The drawback is that maintaining two workflows increases the complexity of your solution. Again, this is bordering on using SharePoint Designer in scenarios where it is not well suited, and you may be better off taking the more demanding route of developing a Visual Studio solution or utilize other third-party workflow tools.

Still, this is possible, despite suggestions to the contrary.

## Other Useful Information

### Extending SharePoint Designer

You can extend SharePoint Designer workflows by adding your own custom actions. You can have actions perform any task at all, including much better methods of looping, security control, workflow control, or anything your heart can imagine.

However, I will not cover custom actions in this issue. The main reason is that custom actions require programming in .NET, and this issue is aimed at administrators and end users.

**Note**

Ironically, Paul Galvin has written a very good overview of custom actions with pointers to existing and downloadable third-party custom actions for EndUserSharePoint.com. You can read that article on http://www.endusersharepoint.com/?p=1146.

The possibility of custom actions is a magnet to smart people who want to help improve SharePoint Designer. You will find a variety of both free and commercial add-ons that will extend the activities available to you.

One such example is the *Useful Sharepoint Designer Custom Workflow Activities* project available on CodePlex. That project adds the following set of actions, taken from the project home page at http://www.codeplex.com/SPDActivities:

**Send Email with HTTP File attachment** - Allows sending emails with attachments retrieved using a web request

**Send Email with List Item attachments** - Allows sending list item attachments as files attached to an email

**Start Another Workflow** - Starts another workflow associated with a list item

**Grant Permission on Item** - Allows granting of specified permission level on a spicified item

**Delete List Item Permission Assigment** - Allows deleting of specified permission level assigment for a given user

**Reset List Permissions Inheritance** - removes any unique permissions assigned to an item by inheriting list permissions

**Is User a member of a SharePoint group** - Checks if a given user is part of given sharepoint group

**Is Role assigned to User** - Checks if a user role is already assigned on the current list item

**Lookup user info** - allows to lookup properties in site's user information list for a given login

NEW! **Copy List Item Extended Activity** - Allows copying/moving list items and files cross site.

NEW! **Send Email Extended** - Enhaced version of the OOTB activity. Allows you to specify the sender. Also does not break links in body.

The links in the titles go to the CodePlex site showing more information about the actions.

If you want to learn how to create custom activities, let me know, and I will put it down on the possible topic for future journal issues. Send an email to journal@understandingsharepoint.com and I will consider it ☺

## Nested Branching

In Chapter 5, you used branching to log different messages to the history log depending on the sum of the order. This was a fairly simple branching exercise, using only a single level of branching.

What if you wanted to further sub-divide each of the branches? For example, you may want to send one type of Thank you mail to the customer if they have previously purchased goods from you, and another type of Thank you email if they are first-time buyers.

Sadly, SharePoint does not support such nested branching. You can only set conditions on the top level of the step. However, you can work around the issue in several ways.

First, the simplest form is just to apply more branches with more targeted conditions. Each condition would target both the order sum as well as the sub-branch such as the purchase history of the customer.

The problem, of course, is that the number of conditions quickly become confusing, even in relatively simple conditions such as in Chapter 5. In Figure 79, I have set up such a set of conditions, and as you can see, the complexity quickly increase.

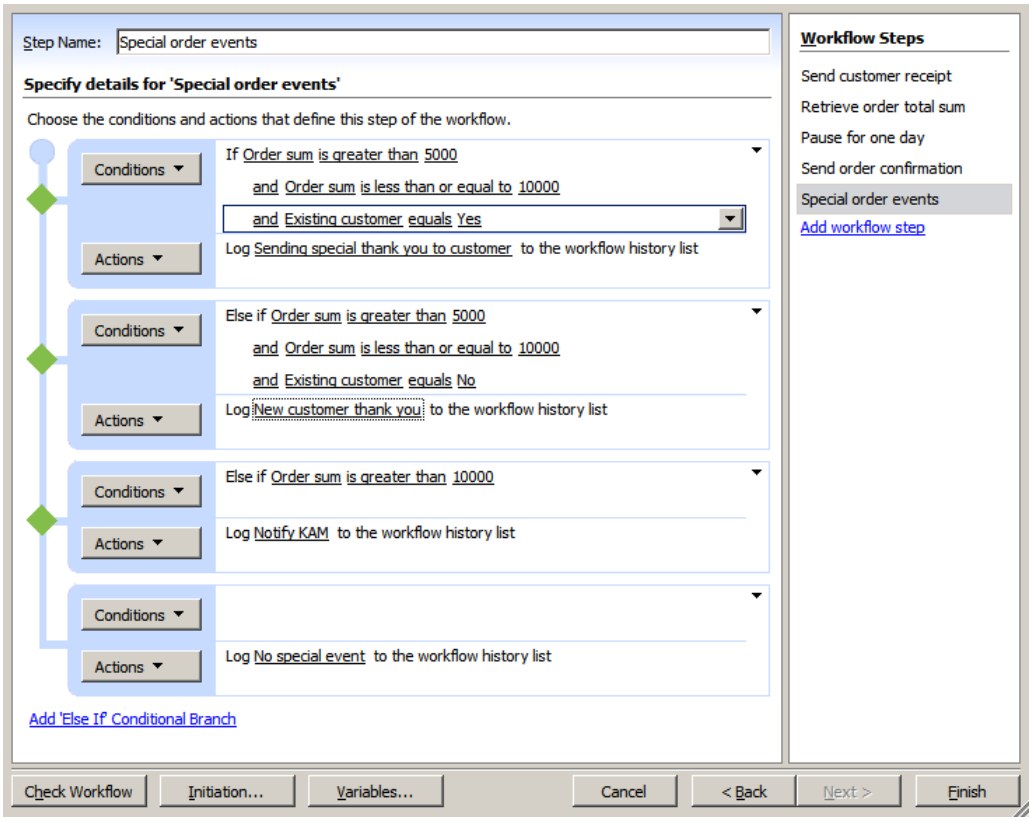



FIGURE 79. NESTED BRANCHING. SORTOF.

Another option is to use a separate step to handle the nesting of branches. In the first step, add a Boolean variable to indicate that the outer branch has been triggered. Then, add a separate step where you check both the Boolean variable and the secondary condition.

While conceptually similar, the latter method may give you a better overview if you use complex conditions in each branch.

I think we have had enough for now. Let's review and complete this issue of Understanding SharePoint Journal.

### ✍ REVIEW QUESTIONS

? 1.

? 2.

? 3.

## ✍ REVIEW QUESTIONS ANSWERED

- 1.
- 2.
- 3.

# Final Thoughts and Additional Resources

*Never stop learning*

Great, we are done with this issue, but I hope I have inspired you to learn even more. Keep in mind that what you have seen here is just the surface of SharePoint Designer, even just a part of workflows in SharePoint Designer. Consider this an introduction to what can be accomplished with the right tools and knowledge of how to use those tools.

This issue is the first in a series of SharePoint business process management issues, focusing on workflow and automation of business processes. BPM is an extremely useful feature of SharePoint and likely to induce a high WOW! factor in any demonstration.

The next issue of Understanding SharePoint Journal will… (not decided yet).

To get information about the upcoming issues of *USP Journal*, make sure you sign up for the mailing list on the *USP Journal* website, http://www.understandingsharepoint.com/journal. Members of the list receive special offers, discounts, and previews of new issues.

You may also want to check out my book *Building the SharePoint User Experience*, which deals with every aspect of the user experience, including a thorough understanding of the SharePoint architecture. You can read more about the book on http://www.understandingsharepoint.com/userexperience.

Finally, check out my blog at http://www.understandingsharepoint.com/url/1000 where you will find shorter articles, tips, questions and answers, and downloadable content.

Thank you for reading this issue, and I hope to see you next time,

.b

# 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 easy of mind from regular payments, handled by E-junkie and PayPal.

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