PART 1

A New Beginning

I'll be the first to admit it—I never read the first chapter of a technical book. Introductions are often boring and rarely contain much useful information.

I'd like to ask your indulgence here, though. Read the first two chapters that make up this part. As the author, I'll admit to being biased, but I think there's a lot of good information in these chapters. We're going to be wading deep into new waters as we progress through the later chapters and it's important that everyone speak the same language and start from the same base.

Chapter 1 is an introduction to all things workflow and then an overview of Windows Workflow Foundation and Workflow in the 2007 Microsoft Office System. It provides information that lays the foundation for what we will cover later.

Chapter 2 covers establishing our environment. It talks about all of the pieces and how to get them installed and configured properly. Chapter 2 then wraps up with an overview of the scenarios we are going to cover later in the book. If you're going to skip a chapter, skip this one. Naturally, as the author, I hope you don't, but, hey, you paid for the chapter whether you read it or not. Also, don't blame me if something doesn't work for you later—everything you need to get things working properly is covered in Chapter 2.

That's it, so let's get started…

CHAPTER 1

Introduction

Imagine waking up one morning and finding everything you knew, or thought you knew, had been flipped over 180°—the sun was green, the sky orange, and you were suddenly no longer a seasoned computer professional but instead a 10-year-old kid sitting in elementary school again. That's kind of how most of us felt when we sat down in front of Office 2007 for the first time—or the first few dozen times.

It's a whole new ballgame.

Once the shock begins to wear off, however, you can see some glimmer of hope. Things are different, certainly, but not so different that you can't function. There are some interesting new kids on the block but most of your old friends are still around—just pimped out in some fancy new clothes. You still use Word to produce documents, Excel to crunch numbers, and SharePoint to collaborate and share content. They just look and act a little differently.

Hopefully, that's where you are when you start reading this book. Are the Office 2007 client applications different from what came before them? Yes, absolutely. Is the Office 2007 System—server and client—better than what came before? Again, the answer is a resounding *yes*.

Continuing on with my somewhat lame analogy, Workflow is just one of those new kids on the block. Workflow is the kid who somehow makes everything else better. Sure, you've got an Xbox 360, but the new kid has the wireless controllers and the big-screen HD TV to play it on. Could you play Xbox before? Sure. But once you've sipped from the fountain, there's no going back. Workflow is the same way. There's very little in Office 2007 Workflow that you couldn't do before, either manually, with a third-party product, or with a homegrown solution. Office 2007 Workflow just makes it all easier, faster and better in every way—*everything tastes better on an Office 2007 workflow…*

Introducing Workflow

All bluster and pontificating aside, Workflow truly is the single most exciting new feature in Office 2007. It is, perhaps, not as immediately noticeable as the changes to the client interface, but it is going to have the most impact on business productivity.

Note One thing that is important to understand early on is that we're talking about workflow in the Office 2007 System—this includes both the client products typically referred to as "Office," as well as the server products—primarily SharePoint.

You would think, perhaps, that Workflow is new technology Microsoft has developed to fill what had been a huge hole in the Office system. In reality, though, Workflow is as old as the hills. It existed before computers were invented and will exist after your souped-up gaming rig has been replaced with something that makes the HAL 9000 look like the ENIAC.

■ **Note** Before going any further, I'd like to make sure that we are all on the same page with regard to the definition of workflow. After all, the rest of this book is going to be about workflow so we should get a definition out of the way right here at the beginning. *Workflow* is a term that means different things to different people. For our purposes, we'll start defining the word by breaking it down:

> *work*: a task to be completed
>
> *flow*: a process

We'll add one other important aspect and that is a *goal*. Every workflow has an identified end result it is targeted to achieve. So, our full definition of workflow is as follows: *the process that defines and controls the completion of one or more tasks in order to bring about the realization of an identified goal*. The key parts of this definition are "process," "tasks," and "identified goal." Everything else typically associated with workflow—notification, reporting, tracking, escalation, etc.—merely support the process and the realization of the goal.

Before computers, workflows were handled manually, usually by a secretary or low-level manager. For example, back in the dark ages (i.e., when I was in college in the '80s) the secretary of the MIS department for the college played the role of what we now call the workflow engine. She would take in documents and manually deliver paper copies to professors or computer operators for review or approval. She had a separate calendar on her desk where she would write herself reminders to follow up on documents and remind people to review them in a few days. To remind people, she would walk to their office and talk to them, talk to them in the hallway, or place a note (an actual physical piece of paper—gasp!) in their mailbox. For what we now call long-running workflows, she had a tickler file—a collection of file folders in a filing cabinet into which she would place copies of documents that she needed to do something with at some point in the future. Every Monday morning she would check the tickler file for the current month to see if there was anything she needed to act on. If there were, she would pull out the paper copy and route it or act on it appropriately. The system worked great—with only the occasional hiccup when she was out sick—until she and her husband moved out of state. Suddenly, the entire department fell apart for a few weeks while the new secretary got up to speed on the system. Documents didn't get circulated, people felt out of touch, and work fell through the cracks. It was really ugly for about a month. Then the new secretary had a handle on everything; she had learned the process and caught up on everything that had fallen behind. All was right again with the world.

A computer's role in workflow is merely to automate that manual process. Like most other computerized processes, the computer isn't doing anything that couldn't be done by a human being, as you saw earlier. It is just doing it more quickly, efficiently, and without cigarette breaks. You also don't have to worry about the computer moving out of state. However, in a sense, something is missing from computerized workflows—human intelligence and adaptability.

Until HAL is installed in your office, computers can only follow a prescribed series of steps. Those steps can be flexible and account for many exceptions and situations within a process, but they will not replace a human being's capability to adjust for a new situation. For example, Nancy (the original secretary from my college) knew that a certain professor would not be at work the day after his alma mater won the NCAA basketball championship and so she had to not only cancel all of his classes, but also follow up with the department head for approval on a document due back to the president of the college that day. While not impossible, computerized processes generally don't account for sudden, unplanned, celebratory drinking binges. Keep that in mind as you plan and execute your workflows.

So, trips down memory lane aside, where are we? At a high level, we know that workflows automate a business process, handling all touch points, routing, escalation, and so forth. Office 2007 fits into this picture as the tool used by most workflow participants to interact with the workflow. This chapter presents some theory on workflow as well as architectural details and some high-level information on how workflows are implemented in the new Office system.

A WORKFLOW BY ANY OTHER NAME…

Business process management. Business process automation. In most cases, these terms are just fancy names for a workflow. Somewhere, someone decided that *workflow* just didn't sound impressive enough, so they coined a new set of terms. When you break it all down to its core, though, it's just workflow.

Workflow Scenarios

Still staying at a high level, workflow basically comes in two flavors:

- *Human-centric*: People are the primary participants and completers of tasks.
- *Machine-centric: Computers* are the primary participants and completers of tasks.

There will almost always be some mixing of human versus machine participation in a workflow, but we classify them based on who does *most* of the work.

Starting with the machine-centric, the following are both examples of machine-centric workflows:

- *Assembly-line robotics*, for example, assembling cars. This is a workflow because there are tasks to be completed (rivets, welding, electrical connections, etc.) and they must be completed in a certain order (you can't install the seats before the floor) in order to achieve a goal (a new car). Human beings do not get involved in most of this work because we are too slow and it is dangerous.
- *Credit card approvals for online purchases*. This is also a workflow because there are tasks (verify card number, verify address, check credit limit, etc.) to be completed in a certain order (you need to verify the account number and access the account before you can verify the billing address) to achieve a goal (sell a product). Human beings do not get involved in this process for fraud-prevention reasons and because we are too slow—imagine what Amazon's sales would look like if a human being had to review and approve every transaction.

These examples are very different and yet both fit well within our definition of a workflow. Similarly, these are machine-centric workflows for different reasons, but at a very basic level it is because the process can be defined and codified to a degree that does not require human intervention. All seats are riveted to the floor in the same way for a given car. All credit card transactions are approved following the same well-defined set of rules and conditions. There is no reason for a human being to be involved for any reason other than exception handling—which brings up a good point. Most machine-centric workflows exist because the process can be defined well enough for someone to write code to enforce the process. However, no matter how well defined the rules and process are, there must *always* be a final piece to handle unplanned-for conditions. That final step is usually to stop and pass the process off to a person to take care of the problem—whatever it may be. A well-written machine-centric workflow will always have this step for unforeseen circumstances because there is no way to code for the unknown.

Human-centric workflows are different—they *start* with preparing for the unknown and support the human participants in whatever tasks they need to perform in order to complete the process. Human-centric workflows generally need some sort of advanced reasoning, comparison, or abstract thinking that cannot be codified. Also common to a human-centric workflow is some sort of approval decision. Whether for accountability or opinion, many human-centric workflows include a step where someone makes a judgment call on whether to proceed. The following are examples of human-centric workflows:

- *Document approval*: The stereotypical human workflow example. No two documents are alike. Each requires advanced reasoning and a high level of abstract thinking in order to be approved. In the vast majority of scenarios, there is no way this can be fully automated.

- *Design approval*: Machines cannot assess aesthetics. For example, there is no way for a computer to determine which of three designs is best suited for a web site, a brochure, or some other marketing material.

- *Document translations*: Machines cannot yet capture all of the nuances of human language. This requires a human being who understands context, cultural implications, and often very precise domain-specific knowledge.

Most human-centric workflows are similar to these examples. Machines are involved for the routing, storage, and notification of task assignments—in effect, the mechanics of the process—while humans are responsible for the actual work performed at most of the steps. In some cases, machines may play a bigger role; for example:

- *Retrieving data* from external sources to augment information contained in a Workflow step. During a purchase requisition workflow, for instance, the computer may retrieve purchase history, budget, and other information to provide additional data to the person responsible for approving the purchase request

- *Automated document creation* based on content supplied during the workflow steps. A scenario for this would be where a salesperson supplies information via a workflow form as part of a sales order workflow and the computer automatically creates a contract document based on a template prefilled with the appropriate details and then routes that document for approval.

- *Automated document manipulation*, for example, removing macros from Word documents before they are circulated for approval to combat the spread of viruses.
- *Process automation* based on information supplied during the workflow. An example of this is the creation of Exchange and Active Directory accounts as part of a new employee workflow.

These steps, however, are still typically secondary to the human tasks in the process. The machine is still following a prescribed set of steps in support of, or based on, the human pieces of the process.

Human-centric workflows are the focus of Workflow in Office 2007. Office is a human-productivity tool and so this makes perfect sense. Office is all about documents (*documents* in a generic sense as this includes not only Word documents but also Excel spreadsheets, PowerPoint presentations, InfoPath forms, etc.). These documents are often referred to as the *payload* for the workflow; they are the chewy nougat at the center. The whole reason the workflow exists is to move this payload through a process.

■**Note** All of this *machine-centric* and *human-centric* talk makes me feel like I'm in a bad *Terminator* sequel—*T99: Rise of the Workflow*. Since we're focusing on Office 2007 Workflow, which is all about human-centric processes, I'm going to drop the human-centric bit from here on out. Just remember that there is a whole other world of workflow out there that has very little to do with documents and people. When we talk about Windows Workflow Foundation later in this chapter, we'll revisit this dark side just a little bit.

Types of Workflow

Workflows fall into one of two broad types, based on how the tasks are processed:

- *Sequential workflows*: Typically depicted as a flowchart, in which the process has a beginning, a prescribed path (which could include parallel branches, criteria-based branching, and loops, but is nonetheless a defined path) and an end. Figure 1-1 shows a generic example of a sequential workflow. It includes most of the typical structures found in a standard flowchart. Starting from the top, it is possible to trace the execution logic from beginning to end without much knowledge of even what the process represents.
- *State machine workflows*: State machines are a significantly different beast from sequential workflow and often harder to bend your mind around. They are, however, much better at modeling complex human activities. Essentially, a state machine is based on the concept of conditions and transitions. A *condition* is a set of circumstances that indicate the current status or situation of the process being modeled. Events occur and cause a *transition* from one condition to another. Unlike sequential workflows, there is no prescribed path through the workflow. Instead, the path taken by the workflow is determined by the events that occur as the workflow is processing. Figure 1-2 shows a generic state machine workflow.

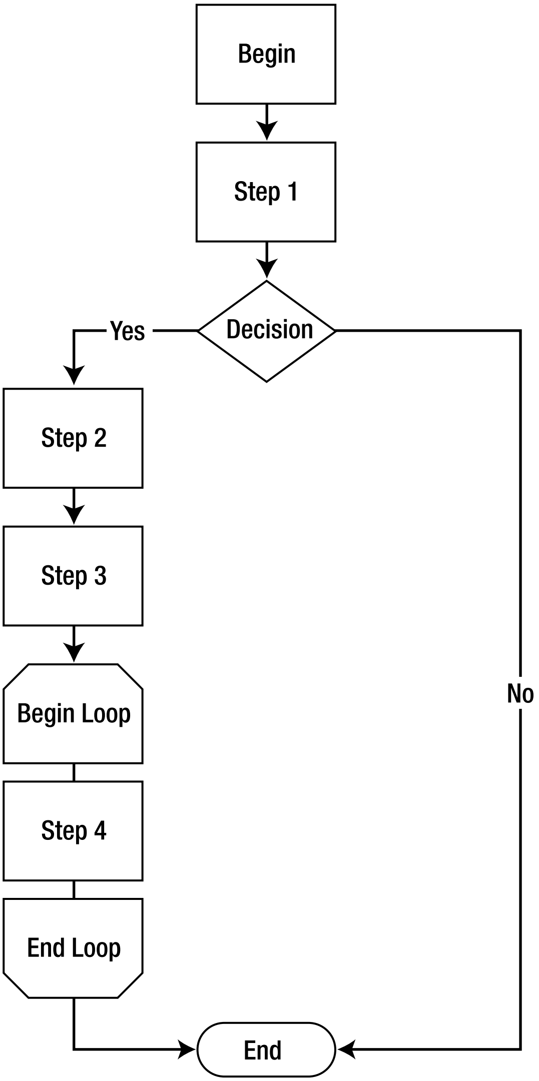


Figure 1-1. *A generic sequential workflow has a prescribed path through the process.*

Looking at Figure 1-2, you can see that on the one hand, the representation of the workflow is much simpler; there are only two structures—states and events. On the other hand, the process seems more complex because there is no way to start from the beginning and step through to the end—there is no prescribed path to follow. Office 2007 supports either type of workflow. We'll cover each in more detail and with concrete examples in the next sections.

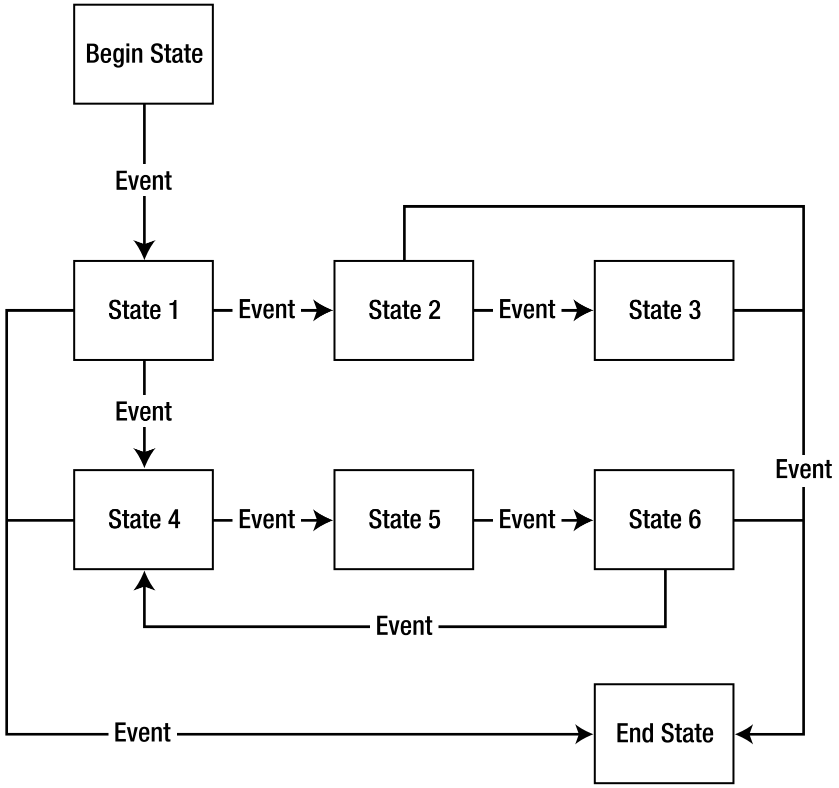


Figure 1-2. *A generic state machine workflow*

Sequential Workflows

As alluded to earlier, and as indicated by the name, sequential workflows follow a sequence of prescribed steps to move from beginning to end. They are easy to understand and follow when presented graphically; consequently, most people start their exploration of workflows with sequential workflows.

For those of you old enough to remember life before event-driven programming, sequential workflows are like the first programs you learned to write in high school or college—linear. They started, did something in the middle, and stopped. You could read from line 1 of the source code all the way through to the end and get a good understanding of what the program did without much (or any) jumping around. There were likely conditions and branches that might send the code down one path or another, but it was all laid out cleanly and in a way that was easy to step through. Sequential workflows are like this.

One key tenet of sequential workflows is that the participants do not typically determine the next step in the workflow. Each participant merely indicates to the workflow engine that they have completed their step. They may or may not have any knowledge of what happens after they have done their task. As I have said before, there is a prescribed path to the sequential workflow so the workflow engine determines the next step in the process based on how it was constructed. The workflow engine is in control.

A sequential workflow works very well in many scenarios you are likely to encounter when implementing Workflow in Office 2007:

- Approval
- Translation
- Feedback
- Collaboration

Most of these scenarios, however, can be generalized into a single category known as *basic routing*, in which documents need to be moved from one person to another. Each person reviews the document and passes it along either with or without adding or changing some content. A sequential workflow handles that scenario very well.

Unfortunately, most business processes are not quite that simple. Sequential workflows are not well suited to modeling complex business processes. They begin to break down with the introduction of significant exception handling, multiple (often arbitrary) execution paths, and nontrivial external factors. In these scenarios, it's time to explore state machine workflows.

State Machine Workflows

Revisiting my programming analogy, state machines are to workflow what object orientation is to programming—a way to simplify the development of something that is complex. As mentioned before, only two elements make up a state machine:

- *States*: A condition that represents the current status of your workflow
- *Events*: Responsible for managing the movement of your workflow from one state to another

Therefore, the definition of a state machine workflow is simply the identification of the possible states and the allowable events that signal a transition to and from each state.

To help understand this better, let's look at an example. A common workflow scenario in Office 2007 will be the approval of a document, so we'll explore a state machine implementation of an approval workflow. Figure 1-3 shows a basic document approval process as a state machine. Not counting the begin and end states, which are really just labels for the diagram and will not actually be developed as part of our workflow, there are three states:

- *DocumentSubmitted*: A document is in this state from the moment the workflow is initiated until the moment someone approves or rejects the document.
- *DocumentRejected*: A document is in this state when the reviewer has rejected it.
- *DocumentApproved*: A document is in this state once someone has approved it.

and three events:

- *OnDocumentSubmitted*: This event occurs when the workflow is initiated. The handler for this event will need to determine who needs to approve the document and notify them in some way that there is a document awaiting their review.
- *OnDocumentRejected*: This event occurs when the reviewer rejects the document. The handler for this event will need to notify the author that their document was not approved.
- *OnDocumentApproved*: This event occurs when a user approves the document. The event handler for this event will need to perform any finalization tasks for the workflow.

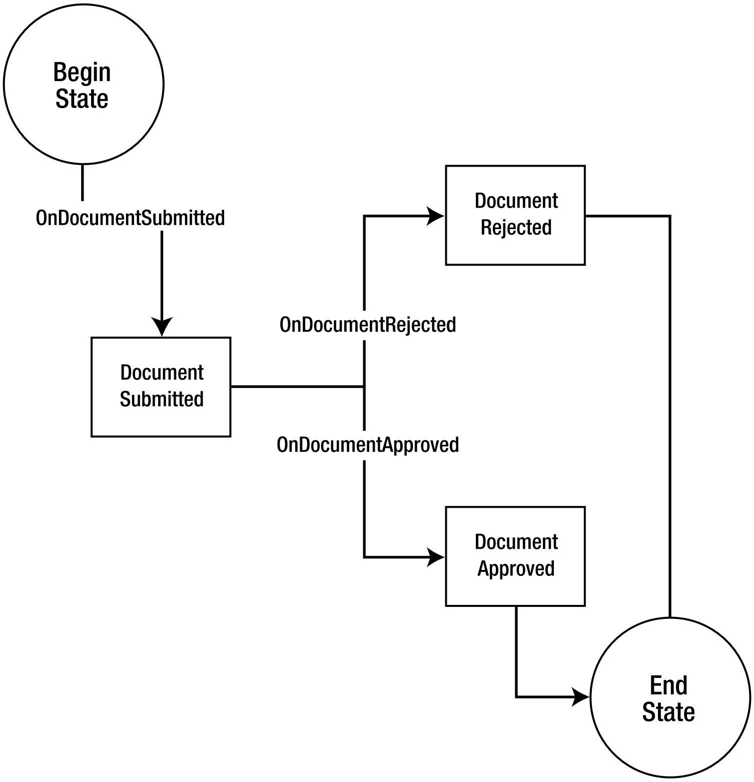


Figure 1-3. *A simple state machine for document approval*

State machine workflows can be used for any scenario you are likely to encounter in Office 2007. As shown earlier, even the simplest workflow can be handled by a state machine. But the real power of a state machine comes in handling complex scenarios. To help you understand this, the next section will look at a single scenario modeled as both a sequential and a state machine workflow.

Sequential vs. State Machine: A Workflow Smackdown

If each style of workflow can handle any scenario we are likely to encounter in our Office 2007 Workflow lives, how do we know which style to use in a given scenario? We already know that simple workflows can be modeled as either sequential or state machine, so the answer there is really whichever model you are more comfortable with. Complex scenarios are a different story, however. I've stated a few times that state machines are more suited to complex scenarios, but there is nothing inherent to Workflow in Office to stop you from handling complex scenarios with a sequential workflow. I'll show you an example of a complex scenario modeled as both sequential and state machine to help you grasp why a state machine is a better choice for complex situations.

First, let's present our scenario. We'll start with the same basic scenario we saw earlier for the state machine—document approval. While not uncommon, a simple scenario such as this is not typical. More common is a situation in which complexity is added in the form of requirements like those listed here:

- The document is time-sensitive so approvals need to happen in a timely fashion or be escalated to another approver.
- There are multiple potential approvers of a document, each with their own area of focus. For example, a document might need approval from several departments.
- Not all reviewers are created equal—some reviewers can trump another's approval or rejection with their own super-approval or super-rejection.
- There are levels of approval, occurring either in series or parallel. What happens when a document is rejected by a second-level reviewer—does it go back to the first level, back to the original author, or continue on for a final review?
- A document requires a subset of the reviewers to approve or reject it before it moves on or falls back. For example, a document is sent to five reviewers and requires any three of the five to approve it before it can be published.
- The document is modified by one approver—does it go back to the original author, start the workflow over again, or just continue on?

For our scenario, we're going to take on just a few of these additional complexities. Our approval is going to require the fourth and fifth items from the previous list. Figure 1-4 shows our process modeled as a sequential workflow. As you can see, even a process that is only slightly more complex gets unwieldy in a sequential model.

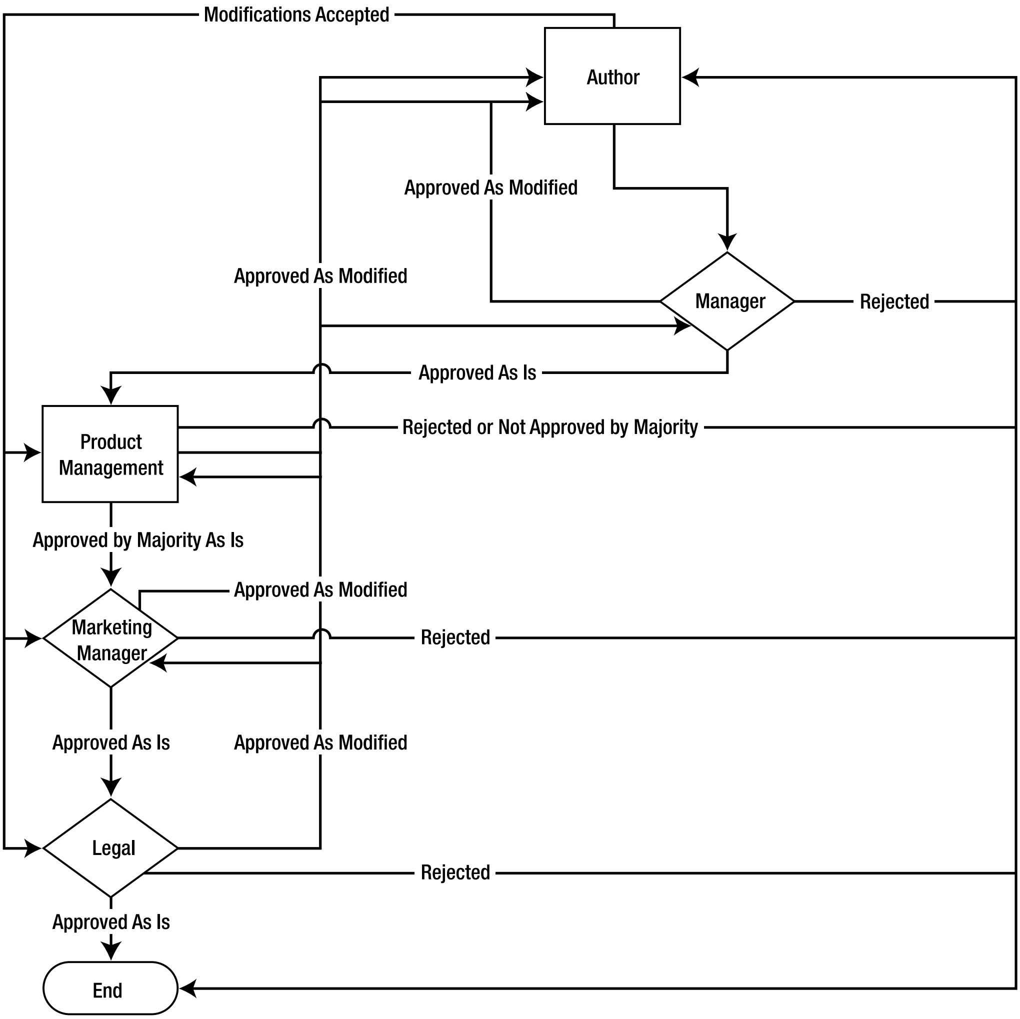


Figure 1-4. *A complex process modeled as a sequential workflow*

Looking at Figure 1-4, we can describe the process as follows:

1. An author creates a document and initiates the workflow.

2. The manager reviews the document and takes one of three paths:

   - *Rejects document outright*: The document is returned to the author as rejected.

   - *Approves document but makes some modifications*: The document is returned to the author with suggested modifications.

   - *Approves document as is*: The document moves on to the Product Management step.

If the document is returned to the author with suggested modifications, the author can accept the modifications, in which case the document moves on to product management, or else reject the modifications, in which case the document dies.

3. Product management has the same three options as the manager—reject, accept as modified, or accept as is. The only difference here is that anything except *rejected* requires a majority of the Product Management Committee. If less than the majority of the committee approves or approves with modifications, the document is returned to the author.

4. The Marketing Manager and Legal steps are similar to the Manager step as well, except that if they reject or accept with modifications, they can opt to send it back to anyone in the process earlier than them—depending on what their objection or suggestion is related to.

Figure 1-4 is not the easiest process to follow. It loops back on itself, branches multiple ways, and generally is difficult to understand. The process depicted is not really that complex and yet the representation is quite complex. The code to implement this process will be equally intricate. It must account for many situations and track the current status of the document through a maze of possibilities. Making this situation worse is the fact that if this process needs to change in the future, making any changes will require someone with significant knowledge of how the process was coded in the first place. None of this makes for a good situation.

Now let's take a look at the same process modeled as a state machine. Figure 1-5 shows this same process.

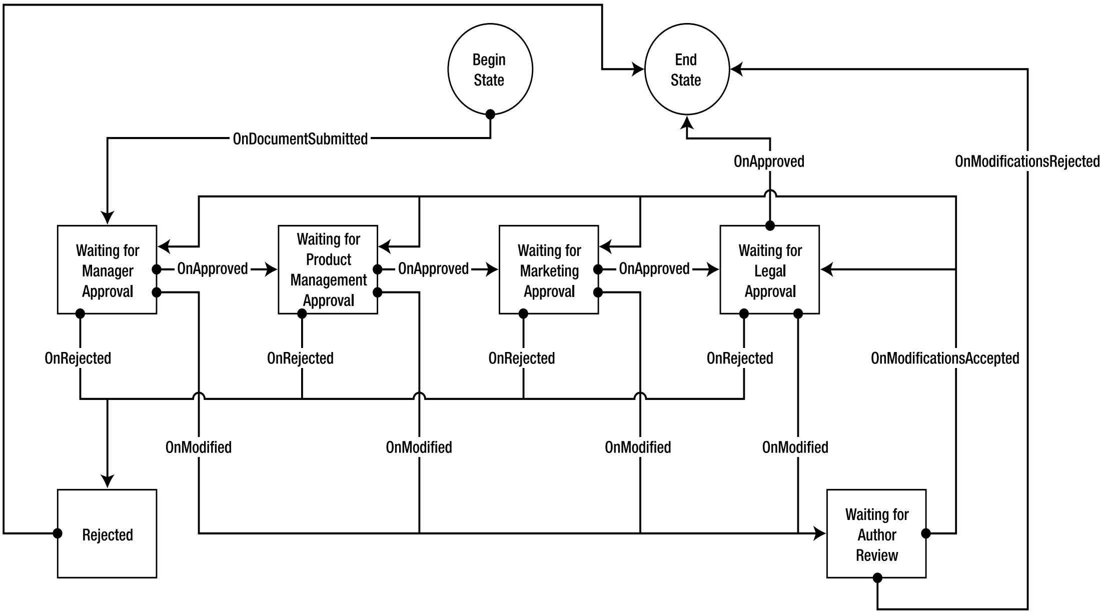


Figure 1-5. *A complex process modeled as a state machine*

At first glance, Figure 1-5 may not seem any simpler. However, once you come to grips with how state machines are modeled, it is much easier to understand. Remember, too, that this is only a somewhat more complex process—fortunately, however, a state machine workflow does not get progressively more complicated as the process it models does. In many ways, once you get over the initial learning curve, very, very complex state machines are not much harder to follow than simpler ones. Take a few minutes to review Figure 1-5 and you'll begin to see what I mean. You'll notice that there are five states (excluding the placeholder begin/end states):

- Waiting for Manager Approval
- Waiting for Product Management Approval
- Waiting for Marketing Approval
- Waiting for Legal Approval
- Waiting for Author Review

and six events:

- OnDocumentSubmitted
- OnApproved
- OnRejected
- OnModified
- OnModificationsAccepted
- OnModificationsRejected

Each participant in the workflow has the same options as in the sequential model. However, looking at Figure 1-5, you can see that it is easier to follow—each state has a set number of events that it supports to transition to or from another state. Future modifications are simply a matter of adding new states or events and updating which are supported at each part of the workflow. The object-oriented nature of a state machine workflow brings all of the same benefits to our workflows as it does to our programs—modularity, ease of maintenance, isolation, and so forth.

As we progress through the book, we'll look at examples of both state machine and sequential workflows. However, we're going to begin our exploration of workflows in Office 2007 with sequential workflows for three reasons:

- Sequential workflows are what most people think of when they think about workflow. They are simpler to understand and so we can focus our time on understanding Workflow's tools and constructs in Office 2007 rather than a new processing paradigm.
- The out-of-the-box workflows (detailed in Chapter 3) are all sequential models.
- The SharePoint Designer—the software formerly known as FrontPage (presented in Chapter 4)—only supports the creation of sequential workflows.

As we delve deeper into Workflow in Office 2007, we'll revisit state machines and see the power and simplicity they bring to a complicated process.

Workflowasaurus: Workflow in the Pre–Office 2007 Mesozoic

With a pretty good understanding of workflow now under our belts, it's time to start honing our knowledge of Office 2007. Before we start, however, you need to understand a bit about where I'm coming from. I alluded earlier to the fact that the lack of workflow has been a huge hole in the Office System for quite some time. Does this mean that the work didn't happen, or that it didn't follow a process? No, naturally not. What it means is that until now, the solution was harder than it could have been. Before Office 2007, workflow in an Office environment was handled in one of three ways:

- With a third-party tool
- With custom code
- As a manual process

Of the three of these possibilities, the last is by far the most common. People route paper copies of documents or email electronic copies. Just last week I spoke to two clients about their processes. Both use folders with the tabs marked as "Return to <*Name*>" to make sure that the document inside eventually made it back to the author. In one case they placed sticky notes inside the folder with the document listing the names of the people who had to review it. In the other case, they just sent the document out and hoped it made the appropriate rounds. The apparent benefit—the cost of implementing this "solution" is zero—is more than outweighed by the ongoing costs of lost productivity and incomplete, incorrect, undocumented, and inconsistent processes. The amount of time, paper, and network or email storage wasted routing, tracking, and following up on documents in a manual process is staggering. In addition, in today's ever-more-regulated business environment a good automated workflow implementation can help ensure regulatory compliance that could save millions of dollars in legal fees, penalties, or settlements.

For those companies that implemented some sort of automated workflow process, most were homegrown solutions. While certainly better than a fully manual process, the time, effort, and cost spent developing, managing, and extending this solution often reduced its business value to a marginal sum. True, the compliance benefits were still there—if the developers did their jobs right—but the costs were often immense.

Finally, workflows built using a third-party tool—and there are many—were often nearly as costly (or in some situations more costly) than a homegrown solution. The benefits of support, reliability, and compliance were usually better than in the homegrown solution, but again, the costs were often a fairly high barrier to entry. Compliance benefits were generally very good and solid, but there's still that niggling little detail of cost.

Windows Workflow Foundation and Workflow in Office 2007 allow you to take care of all of the same compliance issues but at a software cost that is easier for most companies to handle—free (well, at least for the workflow engine). Yes, there is still development, configuration, and administration that needs to happen, but that's no different than any third-party solution and better than a homegrown one. Add to that the tight integration with the most popular productivity suite on the planet, the tight integration with SharePoint, and a few other bells and whistles and you'll begin to understand why the 2007 Microsoft Office system is rocking the workflow world.

■**Note** In Chapter 9, we'll burn a few cycles on integrating Office 2003 with our Office 2007 Workflow experience. We won't come close to the full-blown Office 2007 experience, but we'll at least make Office 2003 a player in the game.

Ladies and Gentlemen...Windows Workflow Foundation

The core of every workflow in Office 2007 is the Windows Workflow Foundation (WF). WF is part of the .NET Framework 3 that also includes the Windows Presentation Foundation and the Windows Communication Foundation. WF consists of

- A collection of classes and objects that form the in-process workflow engine
- Add-on designers for Visual Studio 2005
- A programming model for delivering workflow-enabled applications

In the rest of this chapter, we focus on the core workflow engine provided by WF. The other two aspects will be covered throughout the book as we discuss various development tasks and scenarios.

■**Note** Workflow in Office 2007 is deeply intertwined with the new Office servers and the significant changes that have been made to the Office client applications. While this book is not an authoritative source for either the Office servers or the Office client applications, a brief overview of the salient points of each is provided later in this chapter.

Windows Workflow Foundation Architecture

The workflow engine part of WF is responsible for providing several core services to all workflows built utilizing Windows Workflow Foundation. These services are classified into several categories. Figure 1-6 shows a high-level view of the WF architecture. To better understand the architecture of Workflow, we'll discuss each piece of Figure 1-6 in the following sections.

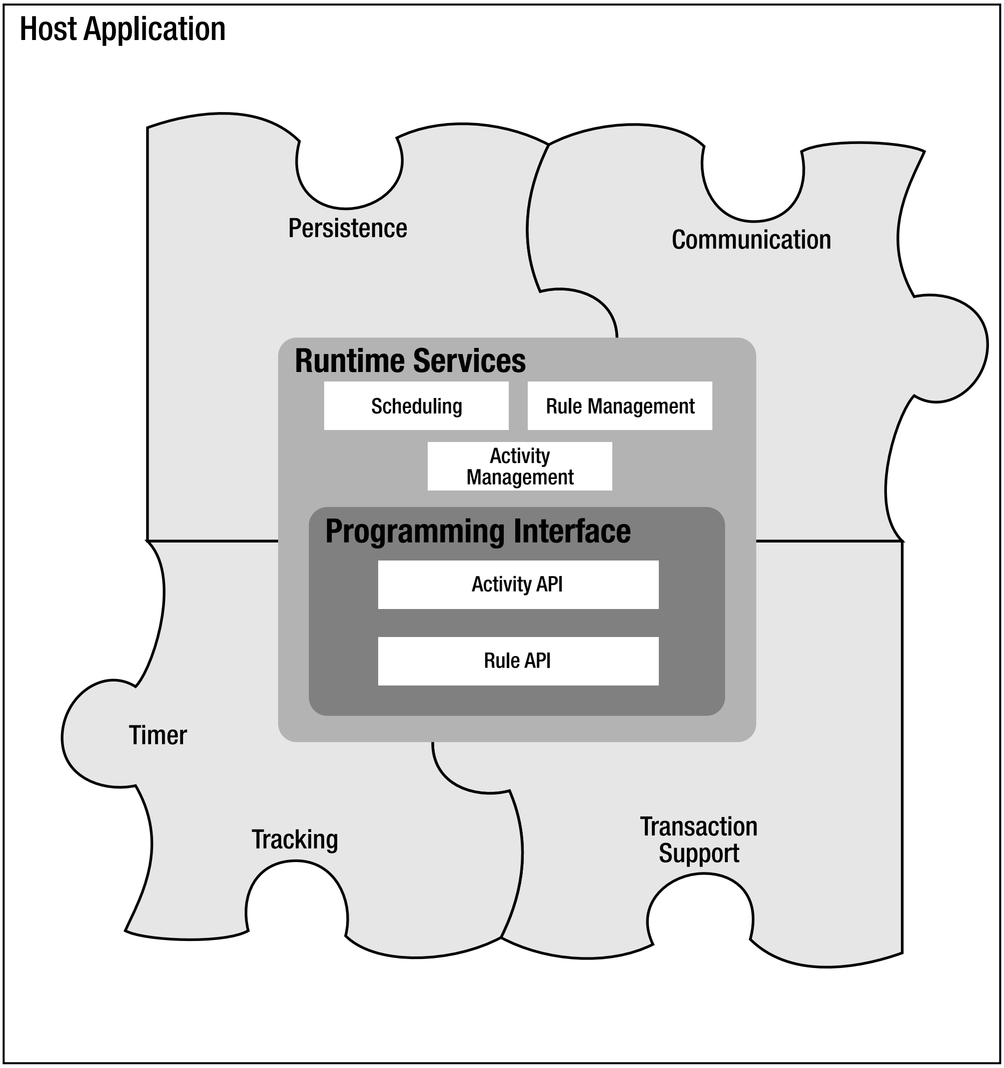


Figure 1-6. *The Windows Workflow Foundation architecture*

Host Interface

One thing that it is important to understand about WF is that it is not directly executable. It is designed only to exist within another process. That *other process* is called the *host*. For Office 2007, the host is SharePoint. We'll talk more about the host functionality provided by SharePoint later. For now, it's only important to understand the concept of a host and the role it plays in our workflows. For more information on workflow hosts, see the sidebar "Every Party (Workflow) Needs a Host."

Definition A *host* is an executable process that serves as the interface between the workflow engine and workflow participants. The host provides common mechanisms for all WF workflows to provide core services.

WF provides a common set of interfaces that allows hosts to directly interact with the core engine. Specific implementation details are left up to the host. As shown in Figure 1-6, the services provided by the WF host interface are like sockets that a specific host application can plug into. The services provided include

- *Persistence*: Even if the various human participants in a workflow responded immediately to notifications and dropped everything to complete their workflow tasks, each workflow would still be considered a *long-running process*. Let's face it, from the computer's point of view, wetware is slow. Combine the fact that people are rarely able to respond to workflow tasks immediately with the fact that some workflows are by nature going to require days, weeks, or months (the design and approval of a new multimillion-dollar widget is not an overnight process) and we could have a major problem on our hands. If the server processing our workflow had to keep all of the details concerning each running workflow in memory, it would soon run out of memory. Furthermore, if all the details about a workflow are stored in memory and the server were to go down for any reason, all of the workflow information would be lost. To get around these problems, WF provides for workflows to be stored and unloaded from memory in the middle of processing. As WF is meant to be hosted inside another application, it does not specify precisely how the information is persisted; it merely provides a mechanism for signaling when the workflow details need to be dehydrated and stored or rehydrated and activated (for details on dehydration/rehydration, see Chapter 9). Each workflow host determines the specific storage mechanism appropriate for its workflows and responds accordingly when signaled by WF. So, for example, one host may store its information in a set of XML files, while another may write them to a set of tables inside a database. In Office 2007, our workflow host is SharePoint, and it stores persistence information in SQL Server.
- *Timer*: It is not uncommon for a workflow to wait for a specified interval before proceeding. For example, a workflow may wait three days and then send out a reminder for a task that has not been completed. It may then wait an additional two days and then escalate the task assignment to a manager. In some cases, this delay information may need to be persistent as well. A typical Windows timer would not serve in all cases because it would not survive a system restart. Each workflow host must be able to specify how it handles delays and other time-based events. WF, therefore, needs to provide a mechanism for each host to implement its own Timer subsystem but still hook into the core workflow processing.
- *Communication*: Each host implementation for WF is going to require a different mechanism for communicating messages (events, information, etc.) from the core workflow engine to the specific host implementation or from the host implementation back into the core. If the WF were to limit communications to only one channel, it would severely limit the flexibility of the platform. For this reason, WF provides a communication interface that can be extended to support whatever mechanism is required by each individual host—from web services to Microsoft Message Queue (MSMQ)-style messages to whatever is required by the host implementation.

- *Tracking*: As workflow is the representation of a business process, it is not at all unusual for the business to need to monitor the progress of a workflow. Companies may need to know where in the process each workflow is at a given point in time. Similar to the Timer process, the details of how this information is stored and retrieved could depend on a number of factors. It would be impossible for the developers of WF to predict each case in advance. Instead, they provide an event system inside WF that raises events during the course of a workflow. Each host implementation is responsible for responding to or ignoring those events as it deems appropriate.

■ **Note** Unfortunately, SharePoint does not implement the Tracking provider from WF and does not support third-party tracking providers. It does implement various types of reports and status windows but nothing as extensible as what would be possible were this interface supported. If we need to track our workflow progress, we will need to handle all of that manually.

- *Transaction support*: Transactions in a workflow can be a tricky thing. Some workflows will require full rollback and transaction isolation functionality whereas others will require no transactions. Every workflow is going to be different. WF therefore can't be all things to all workflows. Instead, it supports a transaction interface that allows the host to determine how to handle transactions. Workflow merely signals when transaction functionality is required and lets the host do the rest.

EVERY PARTY (WORKFLOW) NEEDS A HOST

Much like a host at a party, a workflow host is responsible for the entire experience of its guests. In this case, those guests are the workflow activities as well as the end users. As mentioned earlier, WF by itself is simply a set of objects that expose functionality. It does not expose a user interface (UI) that would allow you to create a workflow, manage a workflow, or respond to workflow events or notifications.

That is all the job of a workflow host. The host presents the interface and the functionality that allows

- Document owners to select and assign a workflow to a document
- Administrators to manage and track workflows
- End users to receive notifications of workflow events that concern them
- End users to update the status of tasks assigned to them

Important tasks of a workflow host include

- *Security*: WF is a generic engine. There is no way that the developers at Microsoft could write a security model into a workflow engine that would suit every application's needs. This is left to the host to implement as it needs to. Some hosts will require no security; others will require Department of Defense–level security. Most will be somewhere in between.

- *User interface*: Similar to security, building one all-encompassing UI into the WF engine would be a nearly impossible task. It would mean that the UI would need to be so generic as to be almost useless, or else that the process of customizing the UI be so open-ended and flexible that it would be exceedingly difficult to work with. So the UI is left entirely to the host developer.

There is nothing in WF that would stop you from writing your own host. Microsoft wholeheartedly supports and encourages this, and WF contains a wealth of information and processes to simplify this task. Unfortunately, writing a workflow host is well beyond the scope of this book.

■ **Note** For information and resources on writing a workflow host, you can visit `http://wf.netfx3.com`.

The next version of SharePoint is a workflow host and for our purposes will serve all our needs nicely. Any customization work that we need to do can be handled with custom *activities*, which will be covered in more detail in Chapter 5.

■ **Definition** An *activity* is a discrete unit of functionality used to build a workflow. Send Email, Create Task, and Write to Log are all examples of potential activities. We'll look at activities in much greater detail, and even build one of our own, in Chapter 5.

Runtime Services

The next set of services provided by WF that we're going to discuss are Runtime Services, shown as the first interior box in Figure 1-6. Unlike the host interface in Figure 1-6, this layer of functionality is isolated from the world outside of WF and will be identical for all hosts and workflow implementations. This layer provides the core services that allow individual workflows to execute. It includes these services:

- *Scheduling*: The Scheduling service is responsible for controlling the runtime execution of activities. It determines when each activity runs.
- *Activity Management*: This service provides the functionality for execution of activities—event management, exception management, transaction management, tracking, and persistence. As necessary, this service interfaces with the host interface layer services.
- *Rule Management*: WF provides for the external application of policy to a workflow via a rules engine. Rules will be addressed in greater detail in Chapter 8.

Programming Interface

The final layer of Figure 1-6 that we need to discuss appears in the innermost shaded box—the programming interface. As workflow developers, this is the functionality that interests us the most; it allows us to write our own custom activities that do what we need. This is the primary means of developing a highly custom workflow. Most of the rest of this book covers the programming interface.

■ **Note** One final item to be aware of with regard to WF is that it can support machine-centric workflows as easily as it supports human-centric workflows. We talked earlier about the distinction between these two types of workflows and discussed how most of this book will focus on human-centric workflows. I just want to make sure that you remember that WF is not limited to only human-centric workflows.

Office 2007 Workflow Technology

So far, we've been talking about workflow in a generic sense, with little if any specific references to Office 2007. That trend stops here. We've covered all of the basics, and everyone should have a pretty good idea of what workflow is, the different types of workflows available, the benefits of workflow, and so forth. From here through the end of the book, we're all about workflow specifically in the Office 2007 system.

Before we get started, you need to understand the Workflow-specific features and additions in the Office 2007 system. We're not going to cover them in painful detail—we'll touch on the various aspects in more detail as we cover topics throughout the rest of the book—for now we're just going to talk about them at a fairly high level so everyone has a basic understanding. Because we're only covering the Workflow-specific features of Office 2007, this should not be considered an exhaustive dissertation on Office 2007. For a more detailed look at Office 2007 in general, or any specific topic not related to Workflow, you'll need to look elsewhere.

■ **Note** There will be plenty of books on Office 2007, some good, some not so good. The pundits will heap praise and scorn on Office 2007, sometimes at the same time. Just about the only thing you can say with *absolute certainty* about Office 2007 is that it will not be a nonevent. For my thoughts on some of the various Office 2007 books that come out both before and after this one, please visit `www.kcdholdings.com` and peruse the book lists.

We'll start with the Office 2007 client applications and then cover the server-side elements of the Office 2007 System.

Introducing the Office 2007 Client

Unless you've been living under a rock (what a strange expression!) for the last several months, you have likely heard all of the chatter about the new UI in the Office 2007 client applications. It is different, I'll grant you that. But once you get used to it, you look at Office 2003 the way a Ferrari owner looks at a minivan—*I'm not going near that…* While we won't specifically walk through the new interface, we will see various parts of it as we explore Workflow.

The full list of Office 2007 client applications is stunningly impressive:

- Access
- Business Contact Manager
- Communicator
- Excel
- Groove
- InfoPath
- OneNote
- Outlook
- PowerPoint
- Project
- Publisher
- SharePoint Designer
- Visio
- Word

Fourteen applications that do just about anything you need, short of laundry and grocery shopping. Wonderful, simply wonderful; no wonder Microsoft has such a lion's share of the productivity suite market. However, and I hate to rain on your parade, but the most important thing to realize about the Office System is that not all of the client applications are created equally. In the client-side world of the Office System, there are

- The Fabulous Four: Word, Excel, PowerPoint, and Access
- Outlook
- A young upstart known as InfoPath
- A bunch of other applications

Please understand that this is not a slight on those “other applications.” I love OneNote—it’s probably my most-used Office application. I use Project frequently, and Visio, Publisher, and the SharePoint Designer are all great at what they do. However, none of these “others” gets the new UI in this release. Personally, I can see why and it makes sense to me—these applications fall into one or more of several exception categories:

- They are not part of what most people use in Office.
- They are more specialized in their focus than the rest of Office.
- They are not document-centric (*document* in a generic sense of a cohesive file that is easily portable and shared among multiple users).
- They are focused on developers.

Having the two UIs presents a problem in a workflow scenario, however. As mentioned earlier, some of the new constructs in the Office UI are intended to inform users of workflow information, for example, the tasks assigned to them. This works beautifully if the document they are working with sports the new UI. But what if it doesn’t? Presumably, this is only a significant problem in Publisher, Project, and Visio as the rest of the “others” are not so file-centric. However, from a workflow point of view, there are two very different experiences:

- If the user opens a Word document in Word 2007, they will be informed of their workflow tasks. The new UI will inform them of the task and support them in managing and completing that task.
- If the document they open, however, is a Project plan, or a Visio or Publisher file, they will receive no indication in the client itself that they have to review and approve that document—even if they open the file in the 2007 version of the program.

Workflow builders and administrators will need to keep this fact in mind and must be sure to support these other file types as special cases should the need arise in their environment. Presumably, a future release of Office will at least partially remedy this situation.

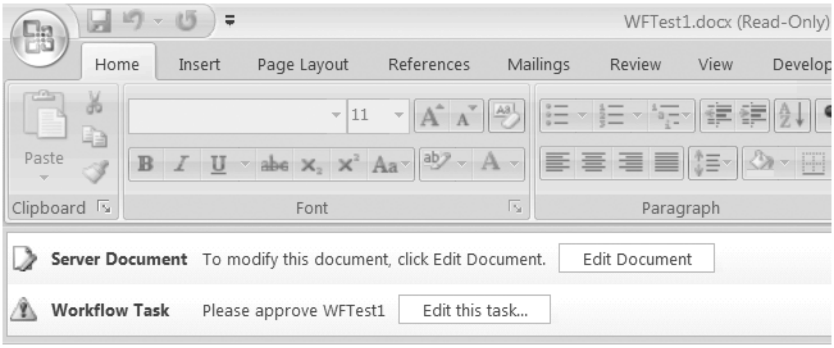
Workflow Features in the Office Client Applications

Hidden among the myriad changes to the Office 2007 client applications are a handful of features that impact our workflows. We’re going to touch on them briefly here. As mentioned previously, we’ll drill into them in more detail as we step through the examples and scenarios later in this book.

Business Bar

First and foremost among these elements is a construct known (during the beta cycle, at least) as the *Business Bar*. The Business Bar integrates the client application with a Microsoft Office 2007 SharePoint Server (MOSS) that is storing the document.

Realizing that there are multiple ways for a user to get access to a document (through email, through a browser, through the Office client, etc.), Microsoft has made sure that users will still be aware of workflow information by placing it directly within the client application itself. If a user has a task assigned to them for the current document, the Business Bar will be shown just under the Ribbon. It will contain information regarding workflow tasks related to the currently opened document. Figure 1-7 shows an example of the Business Bar displaying information about a pending workflow task as well as a notification about editing the document.



WFTest1.docx (Read-Only)

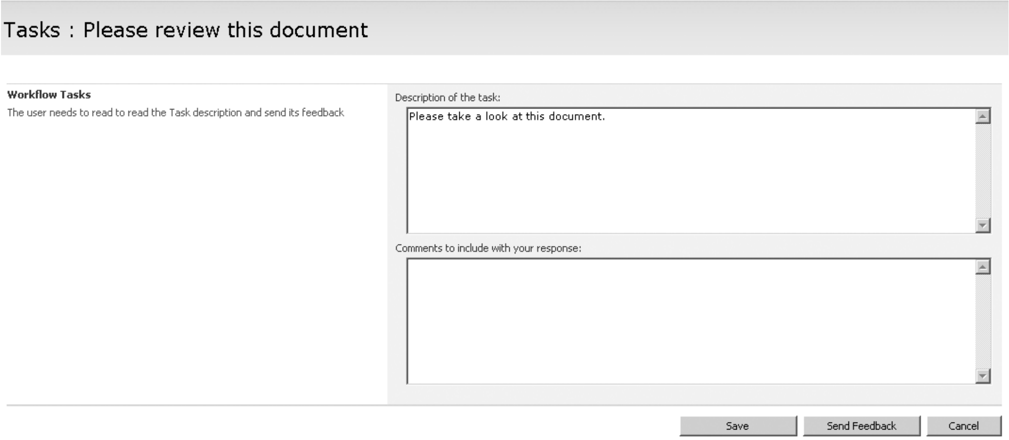
Server Document To modify this document, click Edit Document.

Workflow Task Please approve WFTest1

Figure 1-7. *The Business Bar shows information about pending workflow tasks so the user always knows what they need to do.*

Clicking the Open link on the Business Bar will present the user with information related to the workflow task they have been assigned and give them the opportunity to directly complete and modify that task. The user's experience will depend on which components of the Office 2007 System are installed in their environment.

- Clicking the Edit this task button will launch a browser. The task information will be presented as an ASPX page, as shown in Figure 1-8. This is the only option available if only WSS is installed.



Tasks : Please review this document

Workflow Tasks

The user needs to read to read the Task description and send its feedback

Description of the task:

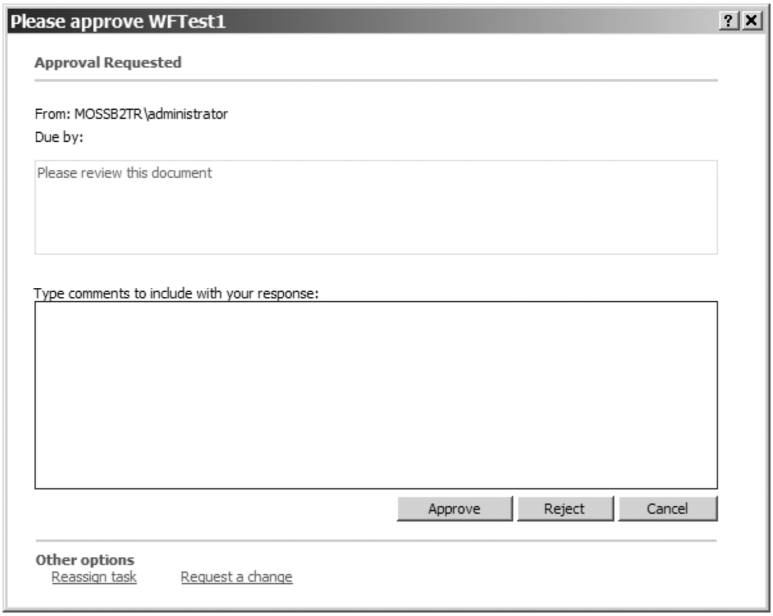
Please take a look at this document.

Comments to include with your response:

Figure 1-8. *The Workflow information dialog box shown as an ASPX page*

- We have not discussed the new part of Office 2007 known as the Forms Server. However, it is part of the 2007 Office System and is responsible for converting InfoPath forms to HTML. MOSS ships with core components of the Forms Server that allow for the conversion of our Workflow forms (even if the full Forms Server is not installed). In this case, the workflow task information will be presented as a form in a dialog box, seamlessly displayed by the Office client application as though it were a native dialog box. An example of this is shown in Figure 1-9. We'll cover the Forms Server in much more detail in Chapter 7.

Note There is another component of MOSS that allows our workflow forms to be rendered directly in the Office client applications—a set of web services that *only* ship with MOSS. This means that even if you install the full Forms Server in an otherwise WSS-only installation, you will still not get this level of integration.



Please approve WFTest1

Approval Requested

From: MOSSB2TR\administrator

Due by:

Please review this document

Type comments to include with your response:

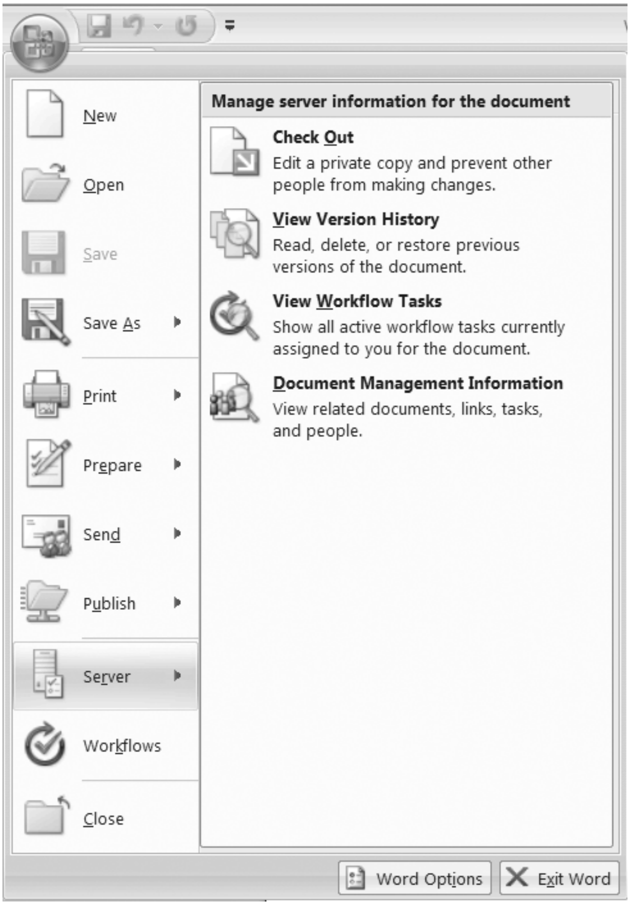
Figure 1-9. *The Workflow information dialog box shown utilizing the new Office 2007 Forms Server*

If the Office Forms Server is not installed in the environment, then the only option available to us as Workflow Builders and Administrators will be the ASPX forms.

File Menu Integration

OK, so technically there is no *File* menu anymore—another one of those little changes in the Office client applications that will take a little getting used to. In place of the File menu, we now have a construct known (again in beta-terminology) as the *Office Button*. Clicking the Office Button will reveal what is effectively the old File menu on steroids, as shown in Figure 1-10.

We'll see the Workflow features of the Office Button in action later as we walk through the various scenarios (although you can get a sneak preview of some of the functionality in Figure 1-10).



Manage server information for the document

Check Out
Edit a private copy and prevent other people from making changes.

View Version History
Read, delete, or restore previous versions of the document.

View Workflow Tasks
Show all active workflow tasks currently assigned to you for the document.

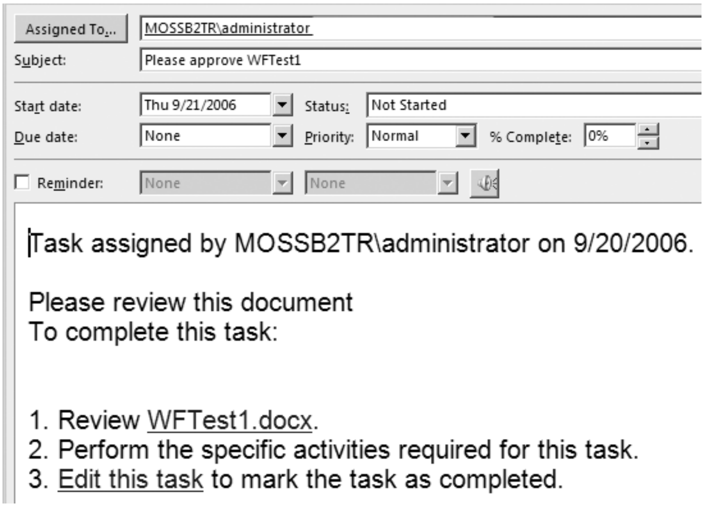
Document Management Information
View related documents, links, tasks, and people.

Figure 1-10. *The new Office Button menu replaces the File menu and provides access to Workflow functionality right from within the Office client applications.*

Outlook

Outlook is likely the most often used Office client application. It's typically the first application opened in the morning and the last closed at the end of the day. Wouldn't it be nice if we could get our Workflow information to show up in Outlook, too? With a lead-up like that you shouldn't be surprised to find out that the level of integration between Outlook and SharePoint has gone through the roof in the 2007 release.

Sticking just with the Workflow-related aspects, the main thing we're looking at is that our Workflow tasks will now show up in our Outlook task list. But it gets even better—the synchronization is two-way. If you edit the task (mark it as complete, etc.) in Outlook, the change will be reflected in the SharePoint task list. Figure 1-11 shows the same Workflow task we saw in Figures 1-8 and 1-9 as an Outlook task.



Assigned To...: MOSSB2TR\administrator

Subject: Please approve WFTest1

Start date: Thu 9/21/2006 Status: Not Started

Due date: None Priority: Normal % Complete: 0%

□ Reminder: None None

Task assigned by MOSSB2TR\administrator on 9/20/2006.

Please review this document
To complete this task:

1. Review WFTest1.docx.
2. Perform the specific activities required for this task.
3. Edit this task to mark the task as completed.

Figure 1-11. *A Workflow task synchronized into Outlook*

Note Later in the book, we'll make use of this two-way functionality to get our Workflow tasks to be accessible from a mobile device.

Microsoft Office SharePoint Designer 2007

No, this is not another client application; it is simply FrontPage renamed and all spruced up. There is some new functionality added that impacts Workflow. The SharePoint Designer includes a wizard interface that allows power users to build a workflow without writing any code. It's called the Workflow Designer and we'll cover it in detail in Chapter 4.

Introducing the Office 2007 Servers

The changes that have been made server-side in the Office 2007 system are, in my opinion, even more significant than the client-side changes. Not all of them will impact our workflows, and they won't be as noticeable to most users, but to a company as a whole they offer much, much more. As before, this section will barely scratch the surface of the new features and functionality. I will cover only those elements that are relevant to Workflow.

Microsoft Office SharePoint Server 2007/Windows SharePoint Services v3

We'll start with the biggie—you can't do workflow right in Office 2007 without doing SharePoint. The next version of SharePoint contains significant changes and will have the most impact on our workflow world. Dozens of books on SharePoint 2007 will flood the market. Hundreds of white papers will be written covering the details of specific features—even the most seemingly insignificant. Thousands upon thousands of blog entries will be written with tips and tricks on how to work with the new features. We're going to cover the couple of items that relate directly to Workflow. Trust me, this barely scratches the surface of the new SharePoint.

Note Before we really begin, a quick word about SharePoint. First and foremost, there really is no product known as *SharePoint.* What people refer to as *SharePoint* is really two distinct products—just as in the last version. First there is Windows SharePoint Services (WSS). This product is free and forms the basis for the next product, Microsoft Office SharePoint Server (MOSS) 2007. WSS ships as part of Microsoft's server products (Windows Server 2003, Longhorn, etc.). MOSS, on the other hand, is a separate product and is not free. It extends WSS in several important ways, primarily due to its lineage—it is a combination of the old SharePoint Portal Server and Microsoft Content Management Server products. When I refer to *SharePoint*, I mean both products. If I'm talking about just one product, I'll call it be name.

We will discuss the specific feature breakdowns of the two SharePoint products in Chapter 3 when we discuss the out-of-the-box functionality of Workflow in Office. For now, a high-level overview of the differences between the two products follows.

WSS is a platform component central to all of Microsoft's collaborative applications. The functionality provided by WSS includes

- Document management
- Core collaboration
- Core Workflow capabilities
- User management
- Security

As stated previously, MOSS is built on WSS. In the future, more and more Microsoft products will build on this base and extend it for their own needs. Microsoft has already announced that future versions of many of their products will utilize WSS as their core collaboration platform. MOSS is simply the first product to build on that base. The additional functionality provided by MOSS includes

- Enterprise content management—drawing on its Content Management Server heritage
- Business Intelligence capabilities
- Expanded search and management functionality
- Integration with line-of-business applications
- Internet publishing

Workflow Features in WSS and MOSS

SharePoint (remember, that means both WSS and MOSS) contain a host of new features that impact our workflows and allow us to build robust, Workflow-enabled applications. Many of them are incidental to Workflow itself, since they are more focused on SharePoint as an application platform and as a robust product. We'll touch on the two most significant of these features here and cover them all in later chapters as we walk through our scenarios and examples.

Content Types A *content type* in the next version of SharePoint is a means of packaging functionality and metadata into a single manageable and deployable unit. Content types are focused on the developer's mantra of *write-once, use-often*. Because we define the functionality, features, and information in one place, it can be managed centrally. That's the *write-once* part. The *use-often* comes in when you deploy the content type and associate it with multiple lists. Each list gets the full power of the content type without having to define it over and over again. Changes made to the single central copy can be immediately reflected in the lists that use it.

Note Rather than typing "lists and document libraries" repeatedly here, I'm just using *lists* in a generic sense. In SharePoint 2007, the distinction between lists and document libraries blurs considerably. Document libraries are merely specialized lists with some additional functionality.

A related benefit is the fact that in the 2007 release of SharePoint, lists can contain more than one type of item. By storing multiple types of documents in a single document library, we can reduce the complexity of our sites, for both administrators and end users. For example, a site dedicated to a specific project can now contain a single document library that houses all of the documents related to that project—designs, requirements, status, project plans, etc. Within a list itself, content types show up on the *New* menu, as shown in Figure 1-12, which show two new content types—*Expense Report* and *Status Report*.

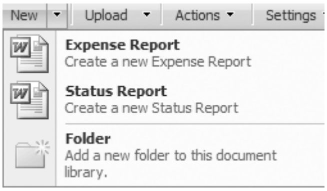

New
Upload
Actions
Settings
Expense Report
Create a new Expense Report
Status Report
Create a new Status Report
Folder
Add a new folder to this document library.

Figure 1-12. *Content types are a powerful new feature in SharePoint.*

Now that you have a basic understanding of what a content type is, let's talk a little about what you can put into a content type. The answer is just about anything you would need to:

- *Metadata*: Each content type can contain any number of columns of any data type available in SharePoint. By adding the content type to the list, you can automatically add the columns.
- *Policy*: Each content type can have its own information management policy defined and enforced.
- *Workflow*: Content types can carry with them their own workflows to enforce business policies specific to the content type. This is obviously the part that interests us the most.
- *Document Information Panel*: Each content type can define a custom form to be displayed in the Business Bar within the Office client applications. This form can be used to keep users informed or to collect information from them.

By default, SharePoint ships with a single content type per list, but it is certainly possible to create your own and add them to your lists as necessary to meet your requirements.

Note For lists of resources that cover creating content types, please visit `www.kcdholdings.com`.

An example here will help clarify the power of the new content types feature. In SharePoint 2003, each list contains one type of information—announcements, links, events, etc. Each document library contains one type of document—contracts, status reports, budget reports, etc. If you need to store multiple types of information and documents, you're forced to create a list for each and put them all in the same site. Although this is certainly better than simple file shares, it is nonetheless problematic. Each site contains the full definition of lists, which means that initially you had to create them all each time, or use site or list templates. In either event, each site still contains the full definition of the list and because each list can only effectively contain a single type of content, you have multiple lists to manage.

Six months pass and you've created dozens or hundreds of sites, each containing your multiple lists. Suddenly the business need changes and you need to start tracking additional information. You need to add one column to every list on every site. Because each list contains its own copy of its definition, you need to manually or programmatically change each.

Ugh.

Now let's look at this using content types in SharePoint 2007. Instead of having to create multiple lists to store all of the various types of content, you can centrally create a content type for each type of item that you need to store and add them to a single list on each site. Not only is your site design easier and cleaner for the end user, but six months, or six days, down the road, you've built hundreds of sites utilizing your new content types. So when the business needs change, you make the necessary changes to the single centralized content type, push those changes out to all lists using the content type, and all is right with the world. It's time for happy hour.

Perhaps this is a *little* more work up front to identify and design the content types, but the payoff in the long run is more than worth any extra initial effort.

Content types are one of the strongest additions to the next release of SharePoint—regardless of whether you are looking from a Workflow point of view. The ability to wrap our business needs up into a nice, neat package and deploy that package as a unit across multiple sites, but still manage it centrally, is quite powerful.

Note It doesn't have to be the case, but to my mind, content types are most useful with documents.

Features While customizing a site definition in SharePoint 2003 was possible, it wasn't exactly the easiest thing in the world. Even seasoned veterans would not look forward to opening `ONET.xml` or `Schema.xml` and wading through line after line of raw XML. In SharePoint 2007 this task is considerably less onerous because site definitions are now broken apart into *Features*.

A Feature is essentially a package of functionality that can be developed, deployed, managed, and activated as a unit. Once deployed to a server, Features can be turned on or off for individual sites.

Features are certainly used for more than Workflow in SharePoint 2007, but from a Workflow point of view, Feature packaging is a useful means of encapsulation and deployment. As you'll see in Chapter 6, we will package our custom workflow into a Feature and deploy it to our SharePoint server.

Microsoft Office Forms Server 2007

Another big addition to the Workflow family is the Forms Server. This Office 2007 server is responsible for dynamically generating HTML forms for data collection. This is going to come into play in two primary ways with regard to Workflow:

- As a mechanism for displaying and collecting information intrinsic to our workflow. We've already seen this use of the Forms Server in action. Flip back a few pages to Figure 1-9. The Workflow task information dialog box shown earlier is an example of a dynamically generated form using the Forms Server delivered right in the Office client applications. The integration with the Office 2007 clients is impressive—it looks like an integrated piece of the application.
- As documents stored in a SharePoint List, forms can have their own workflows assigned directly to them. The possibilities for this scenario are practically limitless. Any form used in your business can be delivered via the web and have a Workflow associated with it. When an instance of that form is submitted by a user, the workflow can automatically begin the appropriate business process.

There is naturally a lot more to this server than that, and the use outside Workflow is equally as important. For now, however, we'll leave it at that. We'll cover the Forms Server in more detail in Chapter 7.

The Cast of *Ben-Hur*

Well, no, not really. There are really only five user types that come into play in a workflow:

- *Initiator*: This is the person who originates or kicks off a workflow. Typically, this will be owner or author of the document or list item but it doesn't have to be. Anyone with sufficient privileges can start a workflow on an item.
- *Participant*: This person receives the tasks of a workflow and completes them.
- *Server administrator*: The server administrator is responsible for establishing and maintaining the environment in which the workflows operate. Primarily this means installing workflows on the server so that they are available to individual sites.

- *Site administrator*: The site administrator is responsible for associating our workflow with a specific list or document library. Often, this will be the same person as the server administrator, though it does not have to be.
- *Builder*: This role is tasked with creating new workflows. This role could alternately be named *designer* to cover the declarative workflows created with the SharePoint Designer (covered in Chapter 4) but for simplicity's sake, I'll just stick with builder.

The majority of this book focuses on the builder and two administrator roles. The initiator and participant roles are typically end user focused (though certainly at one point or another, both administrators and builders will find themselves as participants in a workflow). The primary audience for this book is not end users, so we will not spend a lot of time on those roles. As we walk through examples in the later chapters, however, we will step through the end-user experience. Among other things, it is important for builders and administrators to walk the proverbial mile in the end user's shoes if we are going to create and support an efficient workflow system for them.

Key Facts Summary

That wraps up our whirlwind introduction to Workflow, WF, and Workflow in Office 2007. We've covered a lot of ground pretty quickly. Here's a quick review of what we've seen:

- Terminology: A few key terms were presented in this chapter:
  - *Workflow*: The process that defines and controls the completion of one or more tasks in order to bring about the realization of an identified goal.
  - *Payload*: The document or other piece of content that a workflow is assigned to. For example, in an instance of an approval workflow, the document that a reviewer must approve or reject is called the payload.
  - *Host*: The interface between the workflow engine and workflow participants.
  - *Activity*: A discrete unit of functionality used to build a workflow. *Send Email*, *Create Task*, and *Write to Log* are all examples of activities.
  - *State*: In a state machine workflow, a condition that represents the current status of the process.
  - *Event*: In a state machine workflow, the action responsible for managing the movement of our workflow from one state to another.
- Windows Workflow Foundation, part of .NET Framework 3.0, is the basis for Workflow in Office.
- SharePoint is a Windows Workflow Foundation *host*.

- Despite all of the fun that could be had with a WWF acronym, the preferred acronym for Windows Workflow Foundation is simply WF. Somehow the World Wildlife Fund is drawn into the fight and even Microsoft doesn't mess with the Panda. I'm still on the lookout, though, for a Windows Workflow seminar that calls itself "SMACKDOWN"!
- Windows Workflow Foundation provides a pluggable architecture to facilitate specific implementations of the following services within each workflow host:
  - *Persistence*: Management of long-running processes
  - *Communications*: Integration of myriad disparate components through a common communication subsystem
  - *Tracking*: Monitoring workflow progress in real time (unfortunately, this is not supported in SharePoint)
  - *Transaction support*: The ability to commit or roll back workflow actions as a unit
  - *Timer*: Support for delay or time-triggered processing
- Office 2007 focuses on human-centric workflows, as opposed to machine-centric workflows. Windows Workflow Foundation, however, supports either.
- Office 2007 supports both sequential and state machine style workflows.
- The primary Office client applications (what I termed the Fabulous Four—Word, Excel, PowerPoint, and Access, plus portions of Outlook) support Workflow through a variety of customizations and user interface constructs.
- Some of the new aspects of SharePoint that support Workflow are content types and Features.

Sample Office 2007 Workflow Scenarios

To help everyone understand Workflow in Office 2007 better, straight from the home office in Ottumwa, Iowa, here is my list of the top-10 scenarios best suited to Workflow in Office 2007:

1. *Document lifecycle*: Controlling a document from creation to archiving to destruction.
2. *Document translation*: In our increasingly global economy, support for multiple languages is a business necessity.
3. *Feedback*: Submitting an idea (list item or document) for review and formal or informal suggestions for improvement (as opposed to approval).
4. *Collaboration*: Working with a team to build a document collectively.
5. *Design approval*: Whether for a web page, a brochure, a logo, or some other type of marketing material, aesthetic reviews and feedback are an integral part of this creative process.

6. *Form processing*: Forms strongly imply a process. Any electronic form that generates a discrete document can likely be routed and processed by a workflow. Though certainly not limited to these types, here are a few examples of forms that would lend themselves well to Workflow:

   - Requisition forms
   - Expense reports
   - Travel requests
   - Timesheets

7. *Web content management*: Moving content from concept to publication is often a precise and intense process requiring multiple levels of review and approval. Often, in highly regulated environments, there are prescribed steps to follow prior to publication.

8. *Knowledge management*: Maintaining a knowledgebase is an increasingly important part of corporate life. The ability to have multiple people submit content is often one way to improve the relevancy and quantity of information available. However, without a process to manage those submissions, the knowledgebase will quickly become overgrown and unwieldy.

9. *Document approval*: Not quite as formal or involved as a whole document lifecycle process, this is just a simple review and approval.

10. *<Insert Your Favorite Scenario Here>*: I'd like to say I was being considerate of my audience and letting them supply an entry to this "Top-10" list that met *their* needs, but the truth of the matter is that I got stumped at 9. Somehow, though, a "Top-9" list just doesn't sound right. So go ahead and supply your own entry for number 10.

Summary

I'll close out this chapter with a quick synopsis of a conversation I had with my CTO and one of our salespeople. We were discussing a few projects we were either bidding on or had recently won. After I had mentioned Workflow as an option for the first three brought up, they both asked me what was up with this sudden Workflow-based approach. The end result of our little tangential conversation was a realization that Workflow represents one third of all business applications. Think about it this way: nearly every business application ever written consists of only three things:

- Data storage
- Data presentation
- A business process

Every application differs somewhat in the degree of complexity involved in each of these areas, but any way you slice it, Workflow is roughly 30 percent of every business application. You've been writing Workflow applications for years likely without ever realizing it. Wait and see how much easier your life is going to become.