

# 4

## Helpdesk Ticketing

### WHAT'S IN THIS CHAPTER?

---

- ▶ Allowing users to submit new problems issues or requests
- ▶ Enabling helpdesk employees and managers to respond to and resolve those problems
- ▶ Providing end users with a view into current problem status
- ▶ Providing metrics for analysis
- ▶ Helping to facilitate and grow institutional knowledge by creating a memory bank of solved problems, organized for easy categorization and retrieval

You don't always need to dive deeply into SharePoint Designer to create a workflow solution. End users and helpdesk managers can manage a workflow process quite well with little more than the clever use of a status field on a key custom SharePoint list.

### WHAT IS THE HELPDESK TICKETING PROCESS?

As anyone that works in the IT department of a company of any size knows, technology fails outright on occasion, or end users find creative ways to cause problems for themselves. It often falls to the IT department to unravel these problems and get end users back up on their feet, happy and productive. The goal of a helpdesk ticketing process is to help the IT department manage end user reported problems efficiently, with high quality and with high satisfaction. At the same time, a good solution helps end users figure out things on their own, and when they can't, report the problem and receive updates as IT addresses it for them.

The helpdesk ticketing process loosely follows the pattern you see in crowded delicatessens. You pull a numbered ticket from the ticket machine and wait for someone to call your

number. This is where the “ticket” part enters the picture. Unlike at a deli, however, you’re not placing an order, you’re reporting a problem or requesting action.

Lastly, the helpdesk system helps your organization build a durable base and repository of knowledge from which your team can leverage past effort. The solution that took hours to research this month should be next month’s easy answer. You can do this by marking closed tickets with an eye to building an easily searchable knowledge base.

Tickets start with end users. They can’t get a printer to work or have lost an important document and need someone to retrieve it from the long-term backup storage. The helpdesk isn’t just for problems’ it can be the launching point for ordinary business processes. For instance, a hiring manager could create a ticket, requesting that HR begin searching for a new hire. For example, assume the manager lost an important file.

The manager logs into the intranet to find the helpdesk link and clicks Report a Problem. SharePoint opens up an online form for the manager to fill out. The form prompts for information such as the following:

- Short description
- Longer description
- Problem category and subcategory
- Urgency

A smart form (using InfoPath, for instance) would also prefill certain fields of information, including the name of the user reporting the problem and the date the problem was logged and set some kind of initial status on the ticket. Smart forms are beyond scope for this book, but see the sidebar for more on InfoPath and Chapter 11 for a section describing InfoPath in enough detail to get you started. When you begin to solve this problem, the ticket’s status is important because it drives list views and enables management by exception (MBE) principles.



*InfoPath is a Microsoft office product that works much the same as Excel or Word. InfoPath enables you to create a form by dragging and dropping fields onto the screen. This form is then published to SharePoint, and providing you have the correct edition (Enterprise), SharePoint renders the form in a web browser. InfoPath enables you to create interactive forms that include such things as cascading drop-downs, a conditional display of information, additional buttons beyond OK and Delete, and overall, goes a long way to create a smart online data entry form. This book does not describe using InfoPath much beyond these few sentences because it does not, by itself, necessarily address workflow problems that you can’t manage without it. However, InfoPath can provide much better data entry forms, and you should plan to use it for that purpose in your solutions. Chapter 11 covers InfoPath in more detail and should be enough to get you started using that tool.*

After the user submits the ticket (the originator), SharePoint sends an email to the default owner. Who is the default owner? The category helps you to identify that individual. For example, the

director of HR may handle new hire requests whereas a designated helpdesk manager solves printer problems. Real-life complicates this otherwise simple activity because the usual (default) owner may be on vacation. You need to implement a delegation solution to account for this. When you figure out this owner, the system assigns the ticket to him.

The assignee receives an email (or as you see later, works off a dashboard). The assignee researches the issue, perhaps goes back to the originator for more information and ultimately solves the problem. The assignee then changes the status, which triggers an email back to the originator stating the problem is solved.

You're not quite finished yet. Say that the problem is actually complicated and took up a lot of research time. If it happens again, you hardly want to spend another dozen hours on the same problem. Instead, the system saves the solved ticket and provides enough metadata so that it's easy to locate in the future.

With SharePoint you can design a full-blown ticket management system. That's what this chapter provides.

## PROBLEM CLASS

The helpdesk ticketing problem falls into the real-time response problem class. Random events occur (who can tell when an end user will report a problem?) and someone needs to fix that problem or respond to the request as quickly as possible.

All problems are not created equal. Some problems truly must be resolved ASAP. For example, a sales representative deleted his PowerPoint presentation and needs it restored for a critical sales presentation later that afternoon. These can be time-sensitive issues with dollars (and jobs!) on the line. On the other hand, some helpdesk tickets don't represent problems, but rather a request to start a process, for instance, the new employee on-boarding process described in Chapter 3, "HR On-Boarding." The on-boarding process itself can take some time and is managed with its own workflow. However, it has to be started in a timely manner.

The ticketing process also produces durable information. Helpdesk problems, when resolved, become a history of validated solutions. Over time, the helpdesk ticketing system becomes a treasure trove of institutional problem-solving knowledge.

## TECHNICAL PATTERNS

While solving the helpdesk ticketing problem, you implement the following technical patterns:

- **Basic CRUD (Create/Read/Update/Delete):** Tickets are created, they are updated (many times in this case) and they are "deleted." Deleted is in quotes because the system shouldn't normally delete ticket requests but instead archive them. This is a critical component to the long-term repository aspect of the problem. One thing to take away is that the CRUD pattern which was explained in such detail in Chapter 3 shouldn't be taken literally; the "D" in CRUD could just as easily be an "A" for Archive.

Why save them? Helpdesk metrics can be valuable. Keeping the tickets helps with metrics, as you see later in the chapter.

- **Long-lived data:** Some tickets stick around for a long time. Not all tickets are valuable, so you want to save valuable tickets to a long-term knowledge storehouse. The long-lived data pattern identifies when those items should be moved to long-term storage or instead marked for early archiving.
- **Assignment and delegation:** When tickets arrive into the system, someone has to look at them. The assignment and delegation pattern ensures that 1) tickets never languish because they are always assigned properly and 2) if the usual assignee is on vacation, the system understands that fact and takes the appropriate action in response.
- **Dashboard feeder:** The bedrock of manage by exception (MBE) dashboards show just the right information in an easily digestible format that requires little or no interpretation. Dashboards represent multiple business processes as a snapshot in time. The dashboard feeder pattern ensures that your solution provides appropriate status information so that you can provide meaningful dashboards to the people entrusted and responsible for making sure the business process works smoothly.

## BUSINESS PATTERNS

While solving the helpdesk ticketing problem, you need to implement the following business patterns:

- **Routing:** Who is assigned any given ticket when it arrives in the system? This pattern provides for both a default and fallback assignee.
- **Delegation:** Similar to the technical pattern of the same namesake, delegation enables users to meet their obligations by identifying a responsible team member to handle their responsibilities while they are away. If I'm the guy on the IT team responsible to provision a network security account for a new hire and I'm going to be on vacation, I need to have someone take care of that while I'm away. Delegation lets me do that.
- **Knowledge builder:** Routine issues, such as printer paper jams, probably don't merit special attention. After all, there are only so many ways to unjam a printer. Other issues, however, do warrant careful consideration, require a lot of effort to solve, and may come up again in future. It would be a shame to duplicate that investigative effort again if it's reported a second time. Follow the knowledge builder pattern to implement a solution that helps prevent wasted effort.
- **Metrics:** Management loves metrics, and the helpdesk solution is a perfect place to leverage this pattern. How many helpdesk calls are placed? How long does it take to resolve them? What systems are most affected?

## BUSINESS PROBLEM DESCRIPTION

You cannot define the technical solution without first delving into the details behind the business problem. To do that, you begin by discussing requirements with all the potential stakeholders in the solution. You then identify actors and the actions they take. Follow this process to craft a suitable solution. Let's begin with the requirements.

## Requirements Discussion

In many cases, organizations already have some kind of helpdesk ticketing system in place. If they don't, odds are good that someone in the IT support department has worked with such systems in the past. In both cases, your company probably has one or a few helpdesk "super stars" (who you can probably name without thinking as you read this). All these are good sources for business requirements.

Helpdesk systems serve the needs of the IT department: printer problems, email issues, security setup, and so on. However, this is a limiting view. The facilities group could and should use a specialized helpdesk system to manage facilities requests, ranging from "change my light bulb" to "move my office." HR could use a helpdesk system to manage information requests on benefits and launch new hire processes.

The point of this discussion is that you should cast a wide net when thinking through the potential users (as always). If you narrow your focus to pure IT interests, you can miss an opportunity to extend the solution outward and drive more value for your company and prestige for yourself as a "big thinker." When you canvass other departments for potential helpdesk needs, keep in mind that a department may tell you on Monday that it is not interested, but a change in leadership or edict from high may change situations. In which case, you want to plan for it in advance.

A helpdesk solution needs to provide certain fundamental services to end users:

- Simple and quick entry
- Answer the question, "What's the status of my problem?"
- Notify the end user when important events occur, such as when the issue is assigned, when the issue is resolved, if additional information is required to solve it, and so on
- Self-help: A quick-and-easy way to look up the issue and solve it before a ticket is created

As important as they are, end users are just part of the picture. The helpdesk system also supports its managers:

- Notification when a new issue is created
- Dashboard to view and manage
- Gathering raw data for metrics
- Delegation

If you can meet these requirements, you have a solid solution.

With these basic requirements in mind, schedule meetings with the end user community to conduct the all-important end user interviews. Consider the following questions:

- Which department within the company could use a helpdesk system to manage department-specific requests?
- What kinds of processes make sense to include within the helpdesk universe? They may launch other workflow processes. For instance, it could make a lot of sense to integrate this process with the HR on-boarding process from Chapter 3, "HR On-Boarding," because that could be a separate ticket.

- Do you want to use this support for both internal and external user communities?
- What are security requirements? Should issues be kept secret, and should they be protected from other users' prying eyes?
- What kind of metrics do you want this solution to produce? Total tickets entered? How many tickets opened and closed on a daily basis? What else?
- What will it take to convince the end user community to use it?

This last question is particularly interesting and should be asked of any solution you plan to deliver in SharePoint. Helpdesk issues can be tricky, particularly if you don't already have a helpdesk process live and operational today. End users tend to follow old patterns rather than embrace a new solution. It's almost always easier to pick up the phone and ask for assistance rather than fill in an online form. This can lead to a troublesome disconnect between the "official" system (the helpdesk solution) and reality (what people are actually doing). The best way to solve this problem is to publish a rule: "If a problem isn't in the helpdesk ticketing system, the problem will not be solved." It can cause short-term pain, but your organization can realize a great deal of value from this harsh stance.

For the purposes of this chapter, the helpdesk solution needs to meet the following high-level requirements:

- Any validated user can enter a helpdesk ticket.
- Only employees are considered "valid" users. In other words, you aren't going to allow customers to enter helpdesk tickets.
- Users can see their own tickets but no other users' tickets.
- Helpdesk administrators may see any ticket.
- Users can see the status of their outstanding and closed helpdesk tickets at any time via a convenient dashboard.
- The helpdesk system automatically assigns tickets to a member of the helpdesk administrator based on ticket data whenever possible.
- The helpdesk system supports delegation.
- The helpdesk system supports and enables MBE.
- The helpdesk system generates a database of metrics — not that the requirement is to generate metrics, but to not necessarily report against them. You learn about this distinction later.

## Actors

The model for this solution that meets the requirements includes the following actors:

- An end user who has a problem or wants to kick off a business process.
- One or more users identified as helpdesk administrators. These users work on tickets with an eye toward solving whatever problem prompted the ticket in the first place.
- SharePoint, providing self-service dashboards, email notification, and the automated workflow.

## Actions

The helpdesk solution responds to the actions on behalf of its supported actors. These actions include the following:

- Create a helpdesk ticket.
- Manage existing helpdesk tickets to closure.
- View ticket status.
- Mark valuable solved tickets as such.

## Challenges

This chapter doesn't provide great details on how to address certain nice-to-have functions, including the following:

- **Duplicate tickets:** Two common scenarios arise leading to duplicate tickets (two or more tickets for the same issue). First, end users are simply confused, and they enter the same issue twice. Sometimes, they enter it twice by mistake; sometimes they enter it twice because they feel like the first ticket was "lost." Second, systemwide failures might prompt many different users to enter the same issue. The former issue is easier to detect and manage automatically. However, this chapter does not walk through a solution to either situation. It does suggest an approach to solving them in the conclusion.
- **Urgency:** Most of the time, end users believe their issue is urgent. In practice, if you allow users to specify priority or urgency, they almost always mark their ticket with the highest available priority. This is a social issue more than a technical issue and is not addressed technically.
- **Self-service:** You would like your end users to check out previous solutions in the knowledge base first. Failing that, it would be nice to analyze the ticket programmatically and email the end user potential solutions before the helpdesk administrator starts working on the issue.

## HIGH-LEVEL SOLUTION

The high-level solution is your first attempt to match up SharePoint features and capabilities against the identified business requirements. This solution continues the pattern set from the last chapter where you use content types and custom lists to define data entry screens and the backend data that your workflow process uses to meet the process requirements. A swim lane diagram organizes these different elements into a cohesive picture that you can bring back to the business for validation.

## Solution Overview

To implement the helpdesk application, create a solution that leverages a number of out-of-the-box SharePoint components. In the end, the helpdesk solution is rather light on the workflow side of

things and rather heavier on lists for dashboard management. Workflow is a critical component. However, workflow provides interesting utility functions instead of managing the entire process.

The solution begins with a SharePoint custom list backed by a content type. In combination, these represent a helpdesk ticket. SharePoint's default Create/Update/Delete forms for custom lists provide the user with an interface. This custom list, with some judicious list views, also provides a handy data source for dashboards built using web part pages.

SharePoint's content approval function meets the security needs of the application. You can configure the list to require content approval, and you can create a SharePoint group, Helpdesk Administrators. You can configure helpdesk administrators as approvers on the list. This enables users to create items (helpdesk tickets) in the list but have read-access only to those items they create. Because helpdesk administrators can approve items in the list, they can also view all items in the list.

SharePoint workflow assigns helpdesk tickets automatically based on list metadata. It can do so by looking up values in another SharePoint list that matches ticket types to appropriate helpdesk administrators (experts in the type of issue the user reported). This workflow is smart enough to account for vacations or other planned absences (that is, delegation).

Helpdesk administrators review and take ownership of tickets. As they work on the ticket, they set the status of the ticket. This status mechanism drives where the ticket shows up on dashboards provided for both end users and administrators alike.

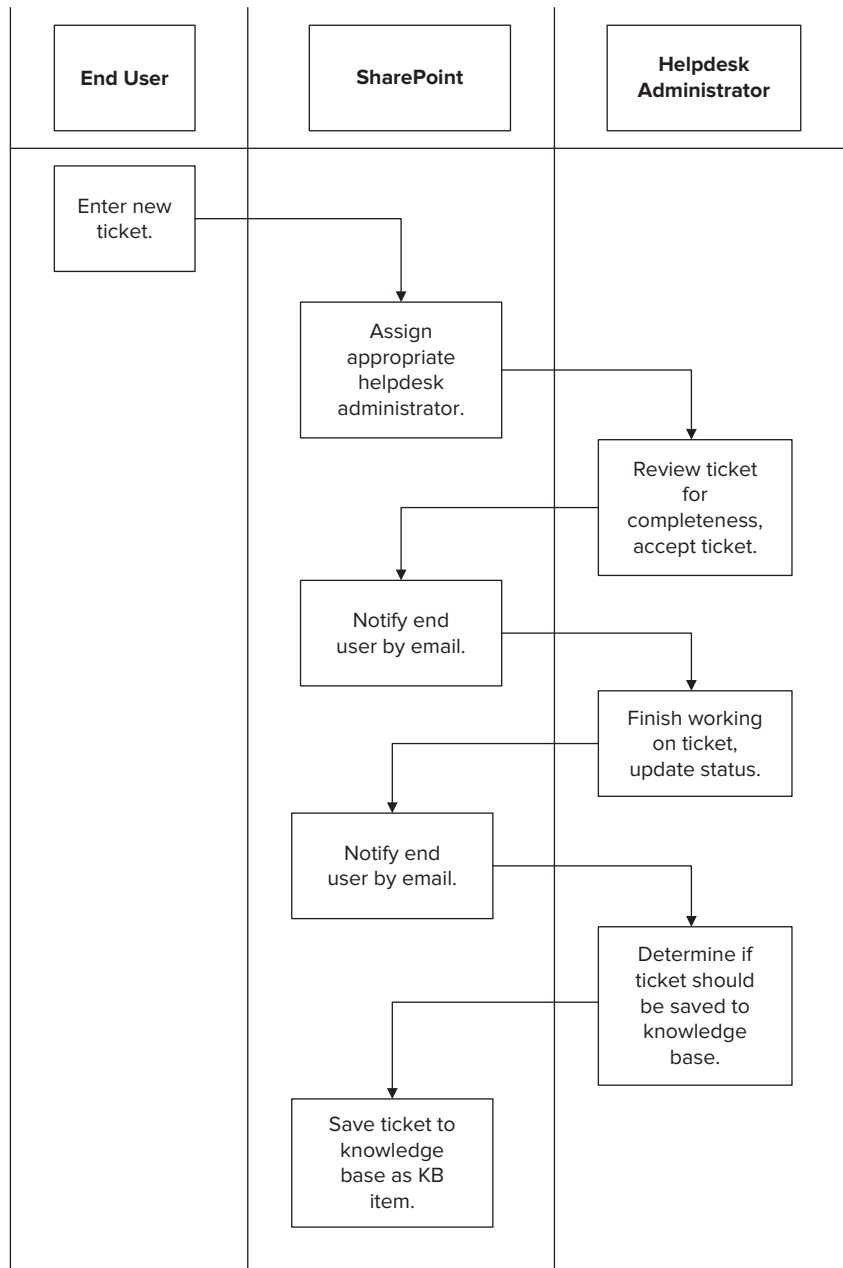
Lastly, the solution marks certain closed tickets to provide a knowledge base of previously solved problems. This solution envisions creating a SharePoint Search Scope on that list (see sidebar for information on scope).



*What is a custom search scope in SharePoint? The entire body of information in SharePoint is called the corpus, which includes all documents, all pages, all list items — anything and everything in SharePoint. If you think of the corpus as a pizza pie, a scope is a slice of pizza. SharePoint search administrators define scopes via a series of rules, and you can create a custom scope that includes only closed helpdesk ticket items. Search scopes help you to meet the knowledge base requirements of the helpdesk solution.*

## Swim Lane Diagrams

The swim lane shown in Figure 4-1 depicts the actors and actions they take in a typical request/response cycle.



**FIGURE 4-1**

The process begins when an end user enters a new ticket. SharePoint routes the ticket to the appropriate helpdesk administrator who reviews the ticket for completeness. End users commonly fail to provide enough information for the assigned helpdesk administrator, who can send the ticket back to the end user for this additional information.

After the ticket is accepted, SharePoint notifies the end user of this fact, and the helpdesk administrator begins to research the issue. The administrator may update the ticket with additional information and research notes. Eventually, the ticket is resolved, and SharePoint emails the end user that it's time to open up the Bubbly.

Lastly, the helpdesk administrator decides whether this ticket and its documented solution should be marked as a knowledge base item for future reference.

## Technical Implementation Overview

Now that the solution framework is well understood, it's time to get into the weeds and start designing the SharePoint solution. Begin by identifying content types and their lists for the general solution. Beyond that, you'll implement a delegation framework that you'll re-use over and over again in the future.

### Content Types

Like all your solutions, site columns and content types lie at the core of the solution. To meet the helpdesk system requirements, you need content types that represent the following:

- **A helpdesk ticket:** This content type has columns such as Short Description, Long Description, and Category.
- **A general purpose delegation content type:** This content type provides the foundation upon which you'll build a delegation model that recognizes the fact that people take vacations or don't show up for work for one reason or another (e.g. sick days). In such cases, the system can't be sitting around waiting for someone to complete an assigned task. Instead, the task must be delegated to another user or group of users. This content type supports that process.
- **Categories:** A list of categories from which the user can select. Categories include items such as Hardware, Software, Email, and so on.
- **Assignment mapping:** Based on a given helpdesk category, determine which helpdesk administrator should work on the ticket by default.

### Custom Lists

Use the following custom lists to implement the solution:

- **Helpdesk ticket:** A list associated with a content type of the same name.
- **Helpdesk ticket categories:** A simple one-column list for managing categories of problems/issues on helpdesk tickets.
- **Solved tickets:** When helpdesk administrators determine that a specific helpdesk ticket reflects a problem that could reasonably arise in the future, they create a "solved ticket." These solved tickets support self-service objectives.
- **Delegate mapping:** Much like assignment matching, delegate matching enables administrators to tell SharePoint how to delegate support requests.

## Delegation

This chapter introduces a technical solution to manage delegation, which helps to create a seamless transition when the regular assignee is unavailable. Consider this scenario:

- Paul typically handles all requests that have anything to do with email.
- The system automatically assigns email requests to Paul. (You see how shortly.)
- Paul begins a one-week vacation on Monday.

If the system goes ahead and blindly assigns the task to Paul, it obviously won't be looked at until Paul returns. That is a problem. Use delegation to solve this.

The delegation problem is easy to understand. The technical solution is also easy to understand at a high level, although as you see, SharePoint does its best to make the actual solution fairly challenging. Now tackle the easy part first.

The delegation solution has at its root a simple custom list with three columns. The first column represents the specific function. Using the preceding example, this might be called Helpdesk Email. The next two columns represent the usual assignee and his delegate. So, if the usual assignee is Paul and Samantha is Paul's backup, you'd have tuple like this:

- **Function:** Helpdesk Email
- **Default Assignee:** Paul
- **Delegate Assignee:** Samantha

This table structure serves the business purpose well. Any time the usual assignee for a given helpdesk specialty is out sick or on holiday, you can simply add them to this list. Obviously, you need to do something in the workflow itself. This is where it becomes challenging.

The challenge arises due to SharePoint Designer workflow's ability (or lack, really) to look up information in lists. SharePoint Designer is perfectly capable of looking up items with a unique key. This wouldn't be a problem if there were only possible delegates for all helpdesk functions. In this case, you'd have a simpler table structure with just two columns: the default assignee and the delegate. Whenever you need to find a delegate, just look up the default assignee and find his/her delegate. This approach may be sufficient in many cases. However, this chapter deliberately complicates matter by adding that third column. This provides a more general-purpose solution that you can use in other business scenarios and to introduce a useful technical technique.

## Dashboard

This chapter introduces the first process dashboard. The dashboard provides a one-stop shopping experience for all matters related to the helpdesk. There are many approaches to creating dashboards. In this chapter, you create a dashboard that leverages a web part page to present views into the overall process.

## DETAILED TECHNICAL IMPLEMENTATION

In this detailed walk-through, you'll finally get your hands dirty. You'll create the content types and lists required to support the workflow. You'll create four different SharePoint Designer

workflows, including the relatively complex delegation management workflow. You'll begin with the setup tasks.

## Setup

To implement the helpdesk solution, you need to create content types, SharePoint custom lists that use those content types and feed your dashboard and, of course, SharePoint Designer workflows to glue it all together.

## Supporting Lists

The helpdesk solution is built upon the following set of custom lists:

- **Helpdesk Tickets:** The core list around which the rest of the solution is built.
- **Helpdesk Categories:** Lookup list that includes values such as Email, Printer, Missing File, and so forth. For each category, it specifies the usual (default) helpdesk administrator assigned to solve these kinds of problems.
- **Helpdesk Status Codes:** A list of helpdesk status codes and corresponding descriptions of each. These codes include, for instance, Open, Assigned, In Process, and Closed.
- **General Purpose Delegates:** This list supports delegation, mapping default assignees to their delegate. This list benefits from its own SharePoint Designer workflow to solve the compound key problem previously mentioned.

## Content Types

Every supporting list is backed by a content type. As with every other content type in each solution, it consists of two broad types of site columns: business and technical. End users enter data directly into business columns, which appear on dashboards. Technical columns, on the other hand, do not generally appear to end users and instead support the technical needs of the solution. This solution calls for the following content types:

- Helpdesk Status Codes
- Helpdesk Category Codes
- Helpdesk Ticket
- General Purpose Delegates

### Helpdesk Status Codes

This simple content type contains two site columns. Both are business columns.

**Business columns:**

- **Status Code:** Status code.
- **Status Code Description:** A longer textual description of the code. This text appears in list views and displays when the end user clicks the status code.

Table 4-1 shows the helpdesk status site columns.

**TABLE 4-1:** Helpdesk Status Site Columns

COLUMN NAME	COLUMN TYPE	ADDITIONAL SETTING
HD_StatusCode	Single Line of Text	None
HD_StatusCodeDescription	Single Line of Text	None

Having created the site columns, add them to a new content type named Helpdesk Status Codes. Create the new Helpdesk Status Codes based off the standard Item system content type.

Create a new SharePoint custom list named “Helpdesk Status Codes”. Via this new list’s settings, enable content types (via advanced settings), and then add the content type you just created to it (i.e. Helpdesk Status Codes).

### Helpdesk Category Codes

This simple content type contains four site columns. Three columns are business level columns, and there is one lone technical column. They follow:

#### Business columns:

- **Category Code:** A short code that describes the category of the helpdesk ticket.
- **Category Code Description:** A longer textual description of the code. This text should help end users accurately select the correct category. The category drives automatic assignment to a specific helpdesk support person, and this description ensures that automatic assignment works as best as it can.
- **Default Assignee:** This is a specific helpdesk support person to whom all tickets in this category are assigned by default.

#### Technical columns:

- **Category Unique Key:** You need a unique key even for categories. A utility workflow updates this column at run-time, and the helpdesk assignee workflow (described soon) uses it to find the assignee.

Table 4-2 shows the helpdesk categories site columns.

**TABLE 4-2:** Helpdesk Categories Site Columns

COLUMN NAME	COLUMN TYPE	ADDITIONAL SETTING
HD_CategoryCode	Single Line of Text	None.
HD_CategoryCodeDescription	Single Line of Text	None.
HD_CategoryDefaultAssignee	Person or Group	Select just a single user. Not mandatory: Some categories don’t have a default assignee.)
HD_CategoryUniqueKey	Single line of text	Technical column that enables to look up this category’s default assignee at run-time.

Having created the site columns, add them to a new content type named Helpdesk Category Codes. Create a new SharePoint custom list for this new content type. Name it Helpdesk Category Codes and associate the site content type, Helpdesk Category Codes, with this new list.

## Helpdesk Ticket

This is the meat and potatoes of the solution. Despite its central role in the solution, it too contains only business columns. They follow:

- **HD\_TicketNumber:** A unique ID that identifies this ticket
- **HD\_ShortSummary:** Short summary of the issue reported. This short summary displays on the dashboard and list views.
- **HD\_OnBehalfOf:** This column supports the notion that administrative assistants may enter an issue on behalf of their manager or other situations in which the person entering the issue is not actually affected by the issue.
- **HD\_AssignedTo:** The actual helpdesk support person assigned to resolve the issue
- **HD\_LongDescription:** The full-blown description of the issue. End users should be encourage to provide more, rather than less, information here to help resolve the issue.
- **HD\_Category:** The user-supplied category. SharePoint Designer workflow tries to use this category to automatically assign a support person to this ticket. This category links to another list for its values. This is called a lookup column. A detailed explanation is presented next.
- **HD\_TicketStatus:** The status of the ticket. This is also a lookup column.

Table 4-3 shows the helpdesk ticket site columns.

**TABLE 4-3:** Helpdesk Ticket Site Columns

COLUMN NAME	COLUMN TYPE	ADDITIONAL SETTING
HD_TicketNumber	Single Line of Text	None.
HD_ShortSummary	Single Line of Text	None.
HD_LongDescription	Multiple Lines of Text	None.
HD_OnBehalfOf	Person or Group	Not mandatory.
HD_AssignedTo	Person or Group	Not mandatory.
HD_Category	Lookup	Selects values from the Categories custom list. See next for more details.
HD_TicketStatus	Lookup	Selects values from the Helpdesk Status Codes custom list you created earlier.
HD_ResolutionNotes	Multiple Lines of Text	None.

Having created the site columns, add them to a new content type named Helpdesk Ticket. Once you've created site content type, create a custom SharePoint list for it. Name this custom list "Helpdesk Ticket".

## GP\_Delegate

This is the most complicated content type in the solution, and you spend a lot of time working with it. It is the only content type that contains a technical site column.

### Business columns:

- **Delegate Function:** Identifies the specific function to which this delegation applies
- **Delegate From:** This is the person who would normally be assigned the item. For instance, if Paul normally handles all helpdesk tickets where the category is Email, then the Delegate From is Paul.
- **Delegate To:** For the business function, delegates tasks normally assigned to the Delegate From to this user

### Technical columns:

- **Delegate Unique Key:** To work around SharePoint Designer's inability to work with compound keys, you need to generate a unique key. This column holds it. Read more about this in the next section.

Table 4-4 shows the general purpose delegate site columns.

**TABLE 4-4:** General Purpose Delegate Site Columns

COLUMN NAME	COLUMN TYPE	ADDITIONAL SETTING
GP_DelegateFunction	Single Line of Text	None
GP_DelegateFrom	Person or Group	Specify just one user
GP_DelegateTo	Person or Group	Specify just one user
GP_DelegateUniqueKey	Single Line of Text	None

Create a new site content type named "General Purpose Delegate" based off the default SharePoint Item content type. Add the GP columns from Table 4-4 to this new site content type.

Next, Create a new custom list named GP\_Delegation. Configure this custom list to use the General Purpose Delegate site content type.

## Create Workflows

To implement the solution, create four separate workflows.

- **Update Delegate Key:** A utility workflow that helps to solve SharePoint's lack of multicolumn primary key support.
- **Determine Delegate:** Used on the helpdesk list to determine the delegate.
- **Update Category Unique Key:** Sets the value of the category's unique key column and used by the Assign default owner workflow.

- **Assign Default Owner:** This utility workflow assigns a default helpdesk administrator to own the ticket until it is closed.

## Update Delegate Key

This chapter introduces the first utility workflow of the book by way of the delegate management workflow. A utility workflow is similar to a “technical site column.” It fills a critical role in the solution, but it’s not directly tied to the business process. The delegate management workflow’s utility role is to make up for SharePoint’s inability to do a multicolumn lookup in a SharePoint Designer workflow. What does that mean? Consider the following example. Paul and Samantha both work for the IT department. Paul normally handles paper jams, and Samantha normally handles lost file recovery. When Paul is on vacation, Aidan takes on Paul’s responsibilities, and when Samantha is on vacation, Sydne handle’s Samantha’s responsibilities. You can see a clear delegation relationship in the Delegate Relationships table, as shown in Table 4-5.

**TABLE 4-5:** Delegate Relationships

TASK	PRIMARY	DELEGATE
Clear paper jams	Paul	Aidan
Recover lost files	Samantha	Sydne

Things become more complicated, however, when someone else is also responsible for one of Paul’s tasks. For example, assume that Peter is also responsible for clearing paper jams. (There are a lot of paper jams at this company.) This time, Martha takes on that responsibility when Paul is on vacation. Now see how Table 4-6 looks.

**TABLE 4-6:** Delegate Relationships

TASK	PRIMARY	DELEGATE
Clear paper jams	Paul	Aidan
Recover lost files	Samantha	Sydne
<b>Clear Paper Jams</b>	<b>Peter</b>	<b>Martha</b>

This solves the problem in a business sense. If Paul is on vacation, the helpdesk ticketing workflow simply has to look up Paul’s delegate to solves the delegate problem generally. Furthermore, this is easy to model in a SharePoint custom list. SharePoint Designer workflow, however, can’t use a table like that directly. Why not and how do you get around this shortcoming?

Now imagine that you defined a custom list as previously shown. To find the delegate for the Clear Paper Jams task, you need to find the row in the list that matches both the task and the primary person responsible for that task. You need both pieces of information to find the specific row in the list. When you find that row, you know the delegate.

SharePoint Designer does enable you to look up information in a list at run-time. However, SharePoint Designer allows you to use only a single column to specify the target row. You can use

only a single column to lookup data in a list. Use a *utility workflow* to solve this problem. In this case, a utility workflow creates a unique lookup value in the delegate lookup list. It does this by concatenating the task with the primary. To implement this solution, follow these steps:

1. Fire up SharePoint Designer, and connect to the GP\_Delegation list you previously created, as shown in Figure 4-2.

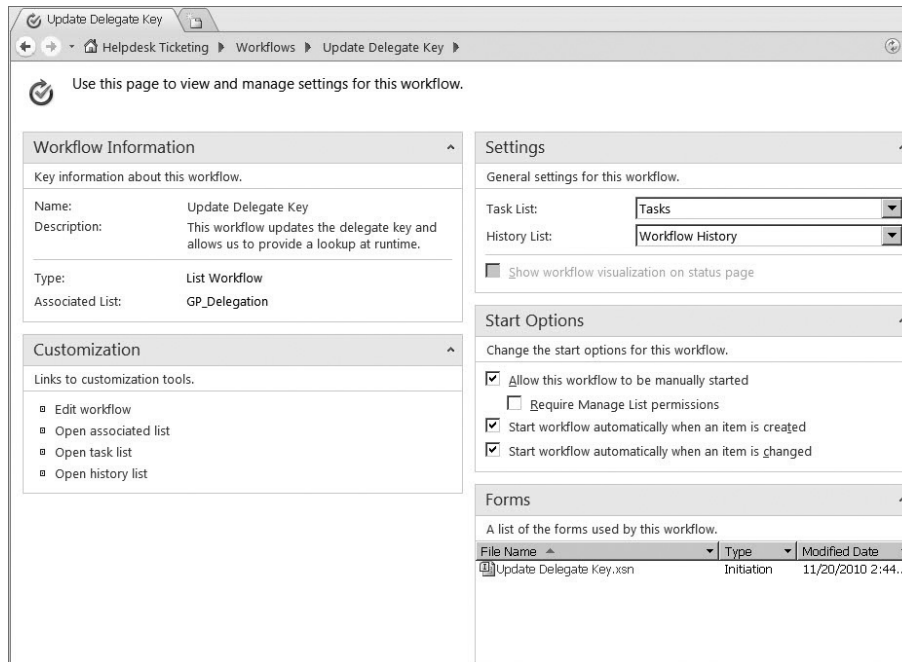


FIGURE 4-2

2. Attach the workflow to the GP\_Delegation custom list and name it Update Delegate Key. This workflow should start automatically when an item is created or changed. For testing purposes, it's also handy to allow the workflow to be started manually.

The workflow itself consists of one step. Name the step “Update Delegate Key.” This step has the following two simple actions:

1. Assign a variable equal to the task plus the primary (no spaces).
2. Assign that result to the GP\_DelegateUniqueKey field.

It looks similar to Figure 4-3 when completed.

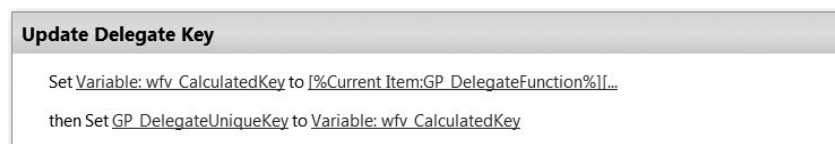


FIGURE 4-3

The first set action is of particular interest. That action creates the unique key and saves it into a workflow variable named `wfv_CalculatedKey`. A workflow variable is the temporary storage place for you to use in your SharePoint Designer workflow.

1. To create a variable, click Local Variables in the ribbon, as shown in Figure 4-4.

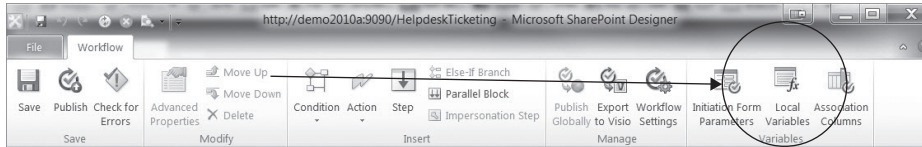


FIGURE 4-4

2. The resulting dialog box enables you to create local variables for use in the workflow. Click the Add button, and create the variable `wfv_CalculatedKey`, as shown in Figure 4-5.
3. With `wfv_CalculatedKey` ready, it's time to calculate the actual unique key. Add a new line to your workflow, and type `set` and press Return. SharePoint Designer responds similar to Figure 4-6.
4. Select Set Workflow Variable. SharePoint Designer displays the set line as shown in Figure 4-7.

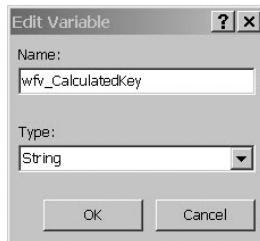


FIGURE 4-5

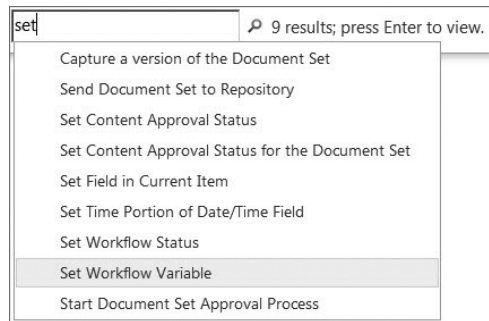


FIGURE 4-6

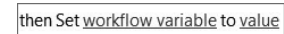


FIGURE 4-7

5. Click Workflow Variable, and select `wfv_CalculatedKey` from the drop-down list.
6. Click Value and then the ... button (not the Fx button). SharePoint Designer responds with the String Builder dialog, as shown in Figure 4-8.

Figure 4-8 shows SharePoint Designer's general purpose String Builder dialog. You can type any text here that you want, but in this case, you want to build a unique key from the two columns `GP_DelegateFunction` and `GP_DelegateFrom`. To do this, click the button labeled Add or Change

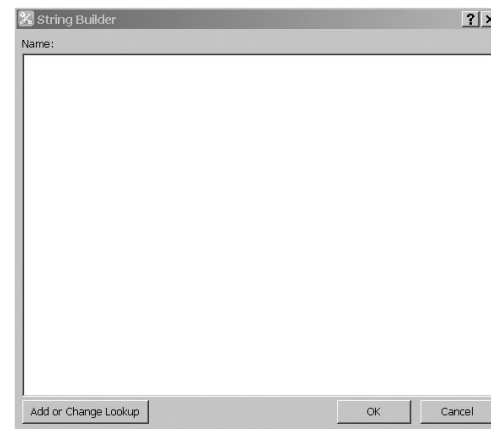


FIGURE 4-8

Lookup on the lower-left side of the String Builder dialog. Welcome to SharePoint Designer's general purpose lookup dialog!

SharePoint Designer's general purpose lookup dialog is powerful and useful. It is also a bit tricky to understand for first-time users especially as it changes itself depending on what values you select. Begin by examining the initial dialog, as shown in Figure 4-9.

Use the first two fields as follows (ignore Return field As for the time being):

- **Data Source:** The SharePoint custom list from which you want to retrieve a value. The more general label Data Source is appropriate as you can find data from the other places, including other workflow variables.
- **Field from Source:** The actual field that you want to extract from the source.

In this case, select Current Item as the data source and GP\_DelegateFrom as the Field from Source. Your screen should look similar to Figure 4-10.

Finally, click the OK button. SharePoint Designer then displays this rather cryptically in the String Builder dialog box, as shown in Figure 4-11.

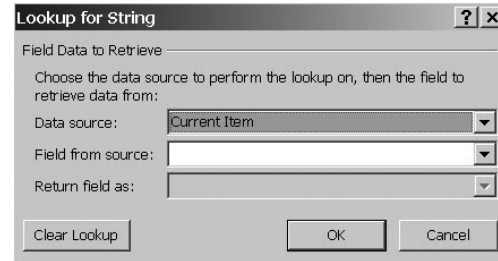


FIGURE 4-9

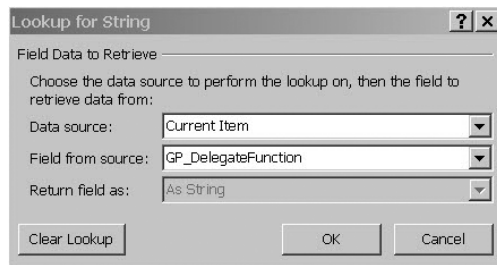


FIGURE 4-10

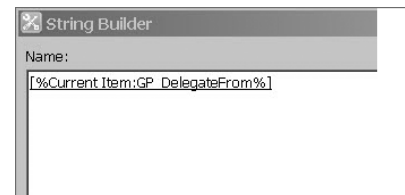


FIGURE 4-11

SharePoint Designer has inserted “[%Current Item:GP\_DelegateFrom%]”.

Next, repeat the same steps but this time select GP\_DelegateFrom instead of GP\_DelegateFunction. In the end, your String Builder dialog box looks like Figure 4-12.

Finally, assign the value of the unique key stored in wfv\_CalculatedKey to the current item's GP\_DelegateUniqueKey. Follow these steps to do that:

1. Type Set into the Update Delegate Key workflow and press Enter.
2. Select Set Field in Current Item.

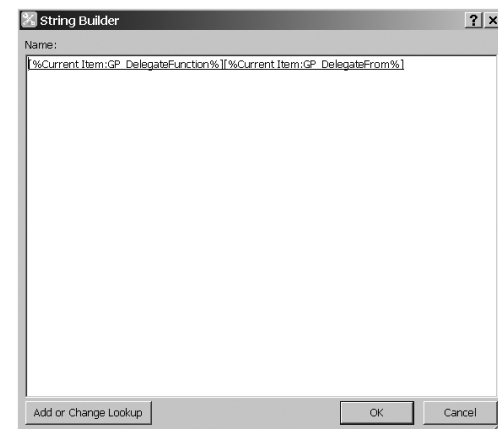


FIGURE 4-12

3. Click the underlined word “field,” and select `GP_DelegateUniqueKey` from the drop-down.
4. Click the underline word “value,” and then the `Fx` symbol that SharePoint Designer displays in response. You’re already familiar with dialog box.
5. Select `Workflow Variables and Parameters` from the `Data Source` drop-down and `Variable:wfv_CalculatedKey` from the `Field from Source` drop-down. Your dialog box now looks like Figure 4-13.
6. Press the `OK` button and you’re done with the dialog.
7. Finalize the workflow by clicking the `Publish` button in the Ribbon and you’re finished!

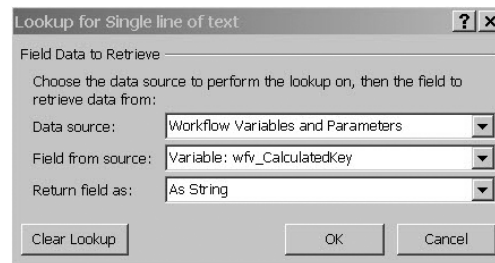


FIGURE 4-13

## Delegate Workflow Summary

The Update Delegate Key utility workflow creates a single column that uniquely identifies a row in the list. Consider Table 4-7. You create entries in `GP_Delegation`. When you do that, you enter values for `GP_DelegateFunction`, `GP_DelegateFrom`, and `GP_DelegateTo`. The workflow runs each time you create a row (three times if you’re following along with the book). That workflow populates the site column `GP_DelegateUniqueKey` as shown in Table 4-7. The values look “funny” because they are mashed together (no spaces). However, this is fine since it’s a technical column and only meant to be used by the workflows themselves.

TABLE 4-7: Fully Populated `GP_Delegation` Custom List

<code>GP_DELEGATEFUNCTION</code>	<code>GP_DELEGATEFROM</code>	<code>GP_DELEGATETO</code>	<code>GP_DELEGATEUNIQUEKEY</code>
Clear paper jams	Paul	Aidan	Clear paper jamsPaul
Recover lost files	Samantha	Sydne	Recover lost filesSamantha
Clear paper jams	Peter	Martha	Clear paper jamsPeter

For the helpdesk workflow to take advantage of this unique key, it must also concatenate the task with the person primarily responsible for that task. When two separate workflows agree on the same definition of a primary key, they are said to cooperate with each other.

You regularly use this technique, both as a utility workflow and an agreed-upon definition of a primary key.

## Lookup Delegate

It takes two to tango, and it takes two workflows to demonstrate delegation: one to manage the delegates (previous section) and one to take advantage of delegates. This section covers the latter.

The lookup delegate workflow kicks in when the helpdesk ticket is first created. This is the only workflow guaranteed to run on the helpdesk ticket. Like the Update Delegate workflow, the Lookup

Delegate workflow is also a utility workflow whose only responsibility is to assign the correct person to work on this ticket.

This workflow runs against the Helpdesk Ticket custom list that you created earlier. Just like the Update Delegate Key workflow you just finished creating, this utility workflow also runs automatically. Again, it's also useful to allow a manual start for debugging purposes.

This utility workflow is almost a mirror image of the update Delegate Key workflow. Instead of calculating a key and then saving it to the current item, you calculate the key and look up the delegate. There are a few twists, however, so now walk through it.

1. First, create two workflow variables named `wfv_CalculatedKey` and `wfv_DelegatedUser`.
2. Using the same technique previously described, assign this variable equal to first `HD_Category` and then `HD_AssignedTo`.
3. With this calculated workflow field, you can now look up the delegate. Use the lookup dialog as previously described. When you finish, your lookup dialog should look similar to Figure 4-14.

Figure 4-14 shows that you are looking for `GP_DelegateTo` in the `GP_Delegation` list by finding a row in the `GP_Delegation` list whose column `GP_DelegateUniqueKey` equals the calculated workflow key, `wfv_DelegatedUser`.

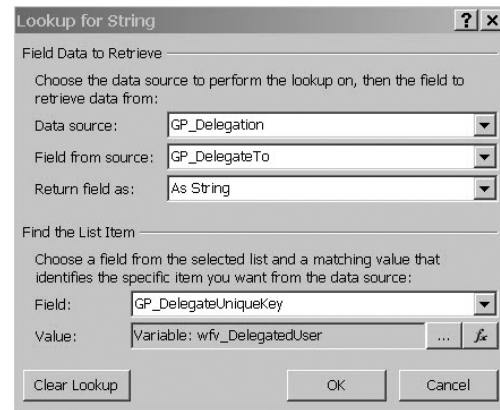


FIGURE 4-14

Two things can happen at this point. Either the lookup succeeds and it does find a row in `GP_Delegation` or it does not succeed. What happens in the latter case? Quite simply, the value of your workflow variable will be blank. You use this fact to assign the delegate key. You do this using a conditional step in your workflow.

1. Create a conditional step by clicking the Step button in the Ribbon, as shown in Figure 4-15.

SharePoint designer responds by adding a new step into the workflow pane, as shown in Figure 4-16.

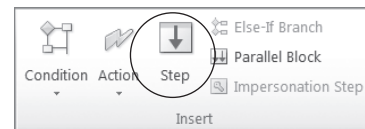


FIGURE 4-15

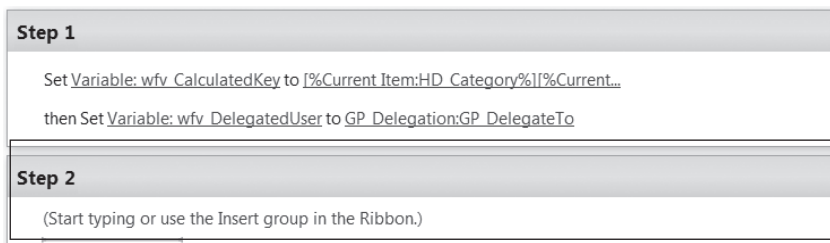


FIGURE 4-16

2. The purpose of creating this conditional step is to assign the delegate you found in the previous list to become the new assigned helpdesk administrator for this ticket. However, you only want to do that on the condition that SharePoint found a delegate. SharePoint Designer workflow always executes steps in the order in which they display. You can control whether a given step executes by adding conditions. To do so, click the Condition button in the tool bar, as shown in Figure 4-17.

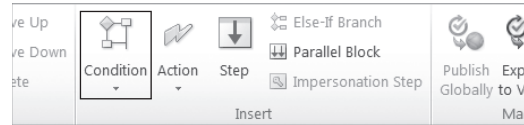


FIGURE 4-17

3. Add a condition to a specific workflow step by clicking the workflow step and then clicking the Condition button. SharePoint Designer responds, as shown in Figure 4-18.

You use a number of these conditions in later chapters. This time, select If Any Value Equals Value. This brings up the now-familiar lookup dialog.

4. This time the data source is Workflow Variables and Parameters, and the Field from Source is Variable: wfv\_DelegatedUser.

5. Click Value, and SharePoint Designer responds with the options shown in Figure 4-19.

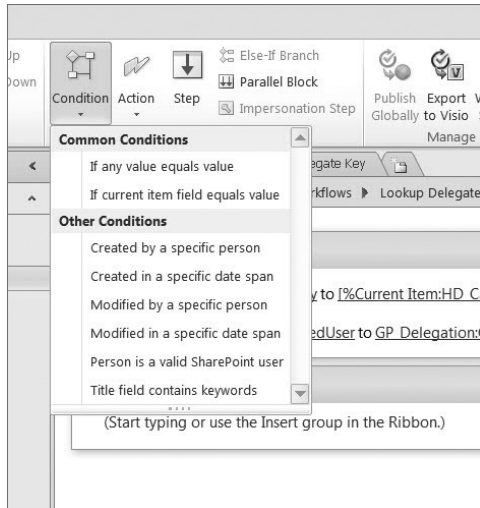


FIGURE 4-18

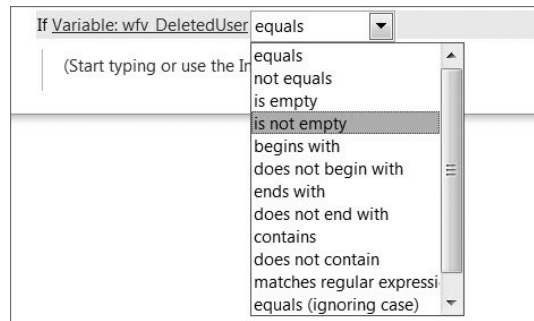


FIGURE 4-19

6. Select Is Not Empty to finalize this condition. This now ensures that the statements in this list will execute only if a delegate were found in the previous series of steps.

7. Finally, assign the delegate. Use the Set Field in Current Item activity. Click Field and select HD\_AssignedTo. Click Value; then click the Fx button. SharePoint Designer responds with the familiar dialog. Select wfv\_DelegatedUser from the Workflow Variables and Parameters data source.

Congratulations, you completed the delegation cycle. You can manage delegates and unique keys. You can look up delegates at run-time. Of course, delegation can't kick into drive until you assign the ticket to someone.

## Update Category Unique Key

The Update Category Unique Key workflow assigns a value to the `HD_Category`'s site column, `HD_CategoryUniqueKey`. There are often good reasons to assign unique keys, as you just read about vis-à-vis delegates. This time, however, it's for a technical reason that allows you to get around a quirk in how SharePoint represents the value of lookups internally.

Recall that the `HD_Ticket` content type has a site column named `HD_Category`. This site column is backed by the custom list, Helpdesk Categories. When users fill out a helpdesk ticket and pick a category, they are actually picking the category from Helpdesk Categories and saving the selected category into the helpdesk ticket. If you have a category named Lost Files, you'd expect that SharePoint stores the word Lost Files into the `HD_Category` site column. Actually, SharePoint treats lookup columns differently from other columns and stores additional information. Specifically, SharePoint stores both the internal ID of the helpdesk category from the source list and the category itself. Therefore, if the Lost Files category has internal ID 5, then the internal value of the `HD_Category` on the helpdesk ticket list is `5;#Lost Files`. (SharePoint inserts the `;` delimiter between the ID and the string value.)

SharePoint is smart enough to strip out the leading characters in list views and elsewhere in SharePoint. However, SharePoint Designer workflows get the internal value, not the display value — therefore, the need for this workflow.

The objective of this workflow is to assign a unique key to the `HD_CategoryUniqueKey` site column so that the Assign Default Owner workflow (described next) can find it.

This is a simple one-line workflow. Create a workflow named Update Category Unique Key with one action, Set Field in Current Item. Click Field and select `HD_CategoryUniqueKey`. Click value and the ... button. SharePoint responds with the String Builder dialog box. Follow these steps:

1. Click Add or Change Lookup.
2. Select ID from the Current Item, as shown in Figure 4-20.
3. Add a `;` character after the `[%Current Item:ID%]` symbol.
4. Click Add or Change Lookup again.
5. Select `HD_CategoryCode` from Current Item.

In the end, your String Builder Dialog will look similar to Figure 4-21.

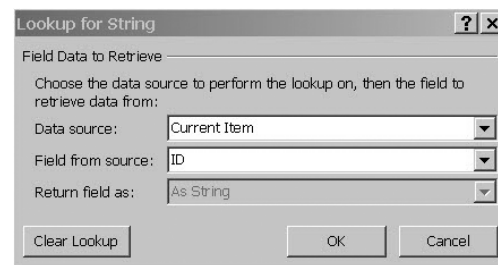


FIGURE 4-20

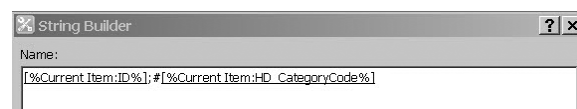


FIGURE 4-21

When this workflow runs, it can now update the unique key column in HD\_Categories. This is going to be useful in the next workflow.

## Assign Default Owner

The Assign Default Owner workflow leverages the Helpdesk Category Codes list to look up and assign a helpdesk administrator to work on the ticket. This workflow doesn't introduce any new concepts and instead leverages techniques described in the previous two utility functions. This workflow runs on the HD\_Ticket custom list.

You only want to assign a default owner when no owner has been assigned. Use this by adding a condition that checks to see if the current item's HD\_AssignedTo is empty.

The one action in this workflow assigns a value to HD\_AssignedTo by looking at a value in the Helpdesk Category Codes custom list. This lookup function is a little different from the preceding because there are a wider array of possible data sources when user IDs (HD\_AssignedTo) are added to the equation, as shown in Figure 4-22.

In this case, you want to select a value from the Helpdesk Category Codes list, so click Workflow Lookup for a User and then Add>>. SharePoint Designer now responds with the same lookup function as before. Use that dialog to select HD\_CategoryDefaultAssignee from the HD\_Category custom list. Your dialog will look similar to Figure 4-23.

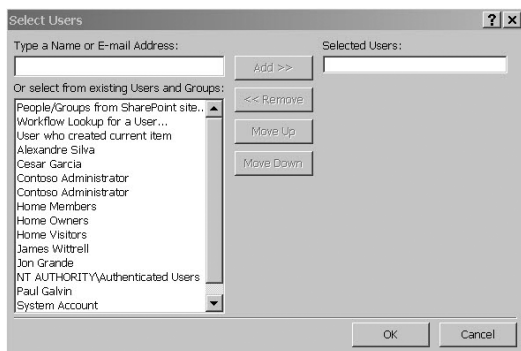


FIGURE 4-22

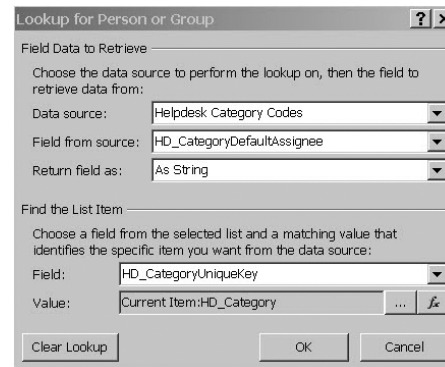


FIGURE 4-23

Click OK and publish the workflow.

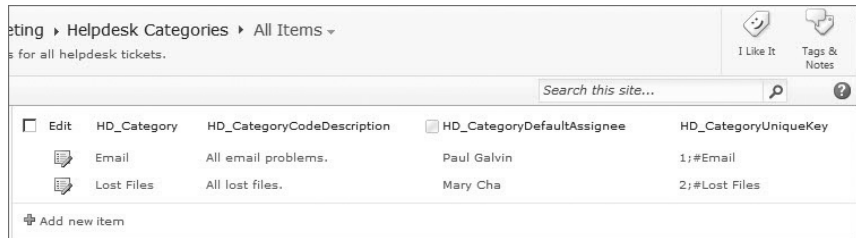
## Managing the Process

This section glues all the discrete pieces together and walks you through the helpdesk process.

### Setup

At this point, you may be wondering how a few content types, custom lists, and some utility workflows make for a comprehensive helpdesk solution. This section describes how these discrete components work together to provide that solution. Begin with a new helpdesk ticket item.

To create a helpdesk ticket, you need to have the supporting data lined up and ready to go. Create several `HD_Categories`, for instance Email and Lost Files. Assign a default owner to each so that you can test this functionality. Figure 4-24 shows an example.



**FIGURE 4-24**

Figure 4-24 shows two helpdesk problem categories. Email problems are assigned to Paul Galvin and Lost Files are assigned to Mary Cha.

In addition to `HD_Categories`, you need to create Helpdesk Ticket Statuses. These statuses lie at the core of what most people would consider the “workflow” in this solution. Create Helpdesk Status as follows, as shown in Table 4-8.

**TABLE 4-8:** Helpdesk Status

STATUS CODE	STATUS CODE DESCRIPTION
1. Open	This is the initial value for all helpdesk tickets.
2. In Process	The assigned helpdesk administrator is actively working on the issue.
3. Awaiting Initiator	The helpdesk administrator needs additional information from the helpdesk ticket originator and cannot proceed until he receives it.
4. Closed	The issue is resolved (or closed for some other reason).
5. KB	Indicates that the item should be added to the knowledge base.

You want to prefix status codes with a number to organize helpdesk tickets by status on list views and dashboards. The leading number ensures that tickets sort and group together. This also enables you to create views on the helpdesk list called In Process or Waiting based on status.

## Entering Tickets

Now that the setup is out of the way, it’s time to create a ticket. End users navigate to the Helpdesk Tickets list, or you can create a link to the list from an appropriate place in your intranet. You should provide instructional content via a wiki or minimally an MS Word document. This content should address topics such as the following:

- How to create tickets
- How to view ticket status

- How to sign up for alerts
- Best practices for tickets (good information helps expedite solutions)

A well-organized document like this can help with adoption because it eliminates mystery and engenders confidence.

Armed with this information, end users should sign up for alerts on the custom list. None of the preceding workflows send emails. Workflow-generated email was omitted on purpose. Alerts on tickets are the best way to handle email communication for a few reasons. First, it's a simple, out-of-the-box approach. Second, alerts go out to end users as helpdesk administrators update the ticket. This helps end users gain confidence that their issue is actually being managed. Lastly, it helps end users learn an important function within SharePoint. Alerting is useful in many other areas, and end users will be the better for it.

## Managing Tickets

Helpdesk administrators manage tickets by controlling individual tickets' status. There is nothing automated here and there is no need for automation. Every ticket needs to be looked at by a human being, and that human being can and should change the status as they are worked on.

Administrators know which ticket to work on by way of views on the Helpdesk Ticket custom list. You can probably imagine quite a few useful views. Consider the following:

- **“My Tickets”**: A filtered view that lists all tickets sorted by status where `HD_AssignedTo = [Me]`. Helpdesk administrators use this view to see any tickets assigned to them.
- **“Blocked Tickets”**: A list of all tickets that require a response from the end user that originated it.
- **“All Open Tickets”**: A filtered view that lists all tickets, grouped by status code. Exclude tickets whose status is “4. Closed” or “5. KB.”
- **“Solved Tickets”**: A filtered view that shows only tickets whose status is “5. KB” grouped by Helpdesk Category. This view can help to quickly identify tickets that previous helpdesk administrators determined could be useful in the future.

These views can help your helpdesk administrators stay on top of tickets and provide the basis for quick answers to status questions from end users.

## Advanced Topics

The techniques and approaches described to this point provide a good basis for a working solution. There are many opportunities to improve upon this. This section describes some of those ideas.

### Dashboards

A helpdesk ticket dashboard extends the fairly simple listview-based approach to ticket management. You can create a web part page and add the Helpdesk Ticket custom list to the dashboard. You can add the list multiple times and specify a different view each time. Using this approach, you can quickly have a dashboard that shows all open tickets, all stalled tickets, tickets

grouped by assignee, and many others. In the end, you have a dashboard that shows open tickets, in process tickets, and blocked tickets.

### Searchable Knowledge Base

Helpdesk administrators decide on a case-by-case basis which helpdesk tickets should be marked as KB, meaning that they are considered valuable references to use when working on future problems. Helpdesk administrators can then use a KB view on the Helpdesk custom list to find these entries. However, this can become ungainly over time as more and more solved helpdesk problems are classified as KB.

SharePoint provides a highly effective search solution out-of-the-box. Use SharePoint search to provide a more powerful solution that can ferret out prior solutions more quickly and with greater fidelity than eyeballing items in a SharePoint list view. SharePoint provides a “free” search tailored to all items in the list. However, you can create a search scope that includes only Helpdesk Ticket items whose status is equal to 5. KB.

## SUMMARY

This chapter walked you through the thinking process, technical design, and actual implementation of a simple but effective helpdesk ticketing solution. Following are the key points you should take away from this chapter:

- **Delegation:** How to set up, manage, and use defined delegates for business processes.
- **Utility workflows galore:** Most SharePoint Designer workflow solutions are purely technical. This chapter used technical utility workflows to support delegation and default helpdesk administrator assignments on new tickets.
- **Status-driven manual workflows:** This helpdesk process does not provide a great deal of automation. Instead, it populates statuses and relies upon helpdesk administrators actively managing the issues. This is often good enough.

You can use these techniques for a wide variety of related business problems, some of which are covered by the following chapters and others that you create in the future.

