Instructional Brief

Advanced Workflow Concepts Using SharePoint Designer 2010

SharePoint User Group
September 8th, 2011
About the Instructor

Susan Hernandez has over 11 years of development experience with extensive skills in Microsoft .NET technologies, including the Visual Studio Suite, Visual Source Safe, and Microsoft SQL Server 2005/2008. She has deep familiarity with all stages of the Software Development Lifecycle and solid expertise with custom Win32 client applications using SOA (Service Oriented Architecture). Specialized areas of interest include developing workflows using SharePoint Designer and Visual Studio 2010, creating forms with InfoPath 2007 and 2010, and the new Silverlight technology.

She also is skilled in Data Visualization using mapping technologies such as the Google Maps API and the Bin Maps API, and Business Integration including MOSS Enterprise Business Data Catalog (BDC) and Dashboards (KPI’s, Excel Services/graphing, and SSRS Reports). Ms. Hernandez also has experience supporting e-commerce web development, smart client applications, and windows service development experience, integrating with SQL Server 2000, 2005, 2008, and Oracle databases.

Susan is “fluent” in a number of languages, including C#, T-SQL, VB.NET; Web Services/SOAP, ASP.NET, HTML/DHTML, JavaScript, XML, and XSL/XSLT.
Introduction
During this session, we will cover a few concepts designed to peak your interest in SharePoint Designer 2010 and Workflows specifically. We will cover the following broad topics:

- Workflow Basics – A Recap
- Workflow Advanced Concepts
- Other Things to Consider

Workflow Basics – A Recap
Workflows using SharePoint Designer have evolved greatly since the days of 2007 workflows.

Out-of-the-Box Workflows
With SharePoint Server Standard or Enterprise license, you receive five out-of-the-box templates to use without needing to edit them in SharePoint Designer 2010.

Approval
An Approval workflow routes a document or list item to specified people for their approval or rejection. If you have Content Approval turned on in your list or library, you can also use an Approval workflow to control the content approval status.

TIP: There is also another, similar workflow type for use in web-publishing sites, the Publishing Approval workflow.

Collect Feedback
A Collect Feedback workflow routes a document or list item to specified people to receive their feedback. The Collect Feedback workflow consolidates all of the feedback and provides a record of the review process.

Collect Signatures
The Collect Signatures workflow routes a Microsoft Office document to specified people for their digital signatures.

IMPORTANT: The Collect Signatures workflow functions only with Word documents, Excel workbooks, and InfoPath forms.

Disposition Approval
The Disposition Approval workflow is designed to support records management needs within an organization. This workflow manages the document expiration and retention process by allowing participants to decide whether to retain or delete expired documents or items.
**Three-State**
The Three-state workflow is designed to track the status of a list item through three states (phases). It can be used to manage business processes that require organizations to track a high volume of issues or items — customer support issues, sales leads, or project tasks, for example.

**Workflow Basics – Steps, Conditions, and Actions**
Steps, Conditions, and Actions are the building blocks for any workflow using SharePoint Designer 2010.

**Steps**
Steps are simply a group of one or more actions and conditions which logically go together in a single unit. New in 2010 is also the concept of an “Impersonation Step”, which will run the step in the context of a user with higher privileges.

Steps can be nested within other steps to create Sub Steps.

**TIP:** Try to limit each step to only one Logical Set of actions. For example, don’t Send an Email and Set a Field in the same step; however do use a Utility Action (such as a string function) and Send an Email in the same step.

**Conditions**
I think of conditions as being like Rules in Outlook. Conditions are one or more limitations you would put on running the particular action. For example, you might wish to say Originator is domain\joeUser AND the field Authorizing Manager is Equal To domain\bossUser.

There are several types of conditions available for you to use:

- Comparing to a field in the current list or library
- Comparing any two data sources internal to the site
  - Does not include external data sources, such as XML files or Web Services
- Checking the title for a specific keyword
- Checking the created and modified, and created by and modified by fields for certain date ranges or certain people, respectively.
- Checking to ensure the user is a valid SharePoint user
- When workflow is created on a Document Library
  - Checking the file type
  - Checking the size of the file
- When inside an Impersonation Step
  - Checking list item permission or permission levels

**Actions**
Actions represent some processing you want to happen, such as Sending an Email, Checking in an Item, or Collecting Data from a User.
Actions can be run in parallel (performed at the same time) using a Parallel Block.

Actions are broken up into the following categories:

- **Core Actions**
  - Mostly manipulation of a particular piece of information such as a variable or list field.
- **List Actions**
  - Work on an entire list item at one time.
- **Task Actions**
  - Manipulate the associated Tasks list and pauses the workflow until complete.
- **Document Set Actions**
  - Work on complete document sets all at once.
- **Utility Actions**
  - Let you perform functions such as working with string values.
- **Relational Actions**
  - Includes one action – finding a user’s manager.

### Core Actions

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
<th>New in 2010?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add a Comment</td>
<td>Does not actually do anything – this allows you to place comments to describe what it is you’re doing in that area</td>
<td>Yes</td>
</tr>
<tr>
<td>Add Time to Date</td>
<td>Allows for addition or subtraction of Minutes, Hours, Days, Months, or Years to a Date field.</td>
<td>No</td>
</tr>
<tr>
<td>Do Calculation</td>
<td>Allows for simple arithmetic functions such as addition, subtraction, multiplication, division, and determining a remainder on two values in the workflow.</td>
<td>No</td>
</tr>
<tr>
<td>Log to History List</td>
<td>Enables you to write items to the hidden History List – is very useful for showing where you are in a workflow process. Use often.</td>
<td>No</td>
</tr>
<tr>
<td>Pause for Duration</td>
<td>Causes the workflow to wait a specified amount of time* before proceeding to the next action.</td>
<td>No</td>
</tr>
<tr>
<td>Pause Until Date</td>
<td>Causes the workflow to wait until a specified date and time* before proceeding to the next action.</td>
<td>No</td>
</tr>
<tr>
<td>Send an Email</td>
<td>Sends an HTML email to a user or set of users.</td>
<td>No</td>
</tr>
<tr>
<td>Send Document to Repository</td>
<td>Sends the document to another repository – for example another document library in a different site collection that functions as an archive.</td>
<td>Yes</td>
</tr>
<tr>
<td>Set Time Portion of Date/Time Field</td>
<td>Overrides the time portion of a Date/Time lookup value. It leaves the Date the same but changes the time, and stores the result in a new variable.</td>
<td>No</td>
</tr>
<tr>
<td>Set Workflow Variable</td>
<td>Directly sets the value of a variable or Initiation Form Field.</td>
<td>No</td>
</tr>
<tr>
<td>Set Workflow Status</td>
<td>Sets the status of the workflow to Canceled, Approved, Rejected, or a custom status you define.</td>
<td>Yes</td>
</tr>
<tr>
<td>Stop Workflow</td>
<td>Stops the execution of the workflow immediately.</td>
<td>No</td>
</tr>
</tbody>
</table>
* The timer job that runs this usually runs every 5 minutes by default. You will not get a precise amount of time, but rather that constitutes a minimum amount of time.

### List Actions

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
<th>New in 2010?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add List Item Permissions*</td>
<td>Specify Users and/or Groups and give them a specific level of permissions on any item in the site.</td>
<td>Yes</td>
</tr>
<tr>
<td>Check In Item</td>
<td>Checks in and sets Check In Comment.</td>
<td>No</td>
</tr>
<tr>
<td>Check Out Item</td>
<td>Locks an item to the person who performs action (Checks Out the document).</td>
<td>No</td>
</tr>
<tr>
<td>Copy List Item</td>
<td>Copies items between two compatible types of lists (i.e. doc lib to doc lib).</td>
<td>No</td>
</tr>
<tr>
<td>Create List Item</td>
<td>Creates a new item in any list in the site and returns the ID of the new list item into a variable.</td>
<td>No</td>
</tr>
<tr>
<td>Declare Record</td>
<td>Declares the current item as a record, which (depending on settings) will protect the record from being edited or deleted.</td>
<td>Yes</td>
</tr>
<tr>
<td>Delete Drafts</td>
<td>Deletes all minor versions (drafts) from the current item.</td>
<td>Yes</td>
</tr>
<tr>
<td>Delete Item</td>
<td>Removes the specified item from the list or library.</td>
<td>No</td>
</tr>
<tr>
<td>Delete Previous Versions</td>
<td>Deletes all versions that are not the current version of the item.</td>
<td>Yes</td>
</tr>
<tr>
<td>Discard Check Out Item</td>
<td>Undoes the check-out and reverts back to the state it was in before being checked out.</td>
<td>No</td>
</tr>
<tr>
<td>Inherit List Item Parent Permissions*</td>
<td>Inherit the permissions of the item’s immediate parent.</td>
<td>Yes</td>
</tr>
<tr>
<td>Remove List Item Permissions*</td>
<td>Specify Users and/or Groups and remove a specific level of permissions to any item on the site.</td>
<td>Yes</td>
</tr>
<tr>
<td>Replace List Item Permissions*</td>
<td>Replace the permission level of the specified item on the site.</td>
<td>Yes</td>
</tr>
<tr>
<td>Set Content Approval Status</td>
<td>Changes the status of the content approval of the current item and sets the comment.</td>
<td>No</td>
</tr>
<tr>
<td>Set Field in Current Item</td>
<td>Sets the value of the specified field to either a hard-coded value or a lookup value.</td>
<td>No</td>
</tr>
<tr>
<td>Undeclare Record</td>
<td>Undeclare the current item as a record – makes the item behave as any other items in the list or library.</td>
<td>Yes</td>
</tr>
<tr>
<td>Update List Item</td>
<td>Enables you to change the fields of an existing item in the site.</td>
<td>No</td>
</tr>
<tr>
<td>Wait for Change in Document Check Out Status</td>
<td>Waits for the document to be in a certain checkout state, such as Checked Out, Checked in, Unlocked by document editor, or Discarded.</td>
<td>Yes</td>
</tr>
<tr>
<td>Wait for Field change in Current Item</td>
<td>Pauses the workflow until a field in the current item matches a particular condition.</td>
<td>No</td>
</tr>
</tbody>
</table>

* Only available within an Impersonation Step.
## Task Actions

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
<th>New in 2010?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assign a Form to a Group</td>
<td>Lets you create a survey that everyone in a group must complete before the workflow continues to the next action.</td>
<td>No</td>
</tr>
<tr>
<td>Assign a To-do Item</td>
<td>Creates a standard task that must be completed before the workflow continues to the next action.</td>
<td>No</td>
</tr>
<tr>
<td>Collect Data from a User</td>
<td>Lets you query a single user with custom fields which the user must complete before the workflow continues to the next action. The ID of the task created is assigned to a workflow variable.</td>
<td>No</td>
</tr>
<tr>
<td>Start Approval Process</td>
<td>An entire tasking process based on the built-in Approval workflow template. This intense action allows you to define various pieces of the process using the Task Process Designer.</td>
<td>Yes</td>
</tr>
<tr>
<td>Start Custom Task Process</td>
<td>Similar to the Start Approval Process action, this however lets you define the process from scratch.</td>
<td>Yes</td>
</tr>
<tr>
<td>Start Feedback Process</td>
<td>An entire tasking process based on the built-in Collect Feedback workflow template. This intense action allows you to define various pieces of the process using the Task Process Designer.</td>
<td>Yes</td>
</tr>
</tbody>
</table>

## Utility Actions

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
<th>New in 2010?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extract Substring from End of String</td>
<td>Copies the specified number of characters from the end of a string and saves the result in a new variable.</td>
<td>Yes</td>
</tr>
<tr>
<td>Extract Substring from Index of String</td>
<td>Copies a part of the string from a certain place within the string and saves the result in a new variable.</td>
<td>Yes</td>
</tr>
<tr>
<td>Extract Substring from Start of String</td>
<td>Copies the specified number of characters from the start of a string and saves the result in a new variable.</td>
<td>Yes</td>
</tr>
<tr>
<td>Extract Substring of String from Index with Length</td>
<td>Copies a part of the string from a certain place with a certain length within the string and saves the result in a new variable.</td>
<td>Yes</td>
</tr>
<tr>
<td>Find Interval Between Dates</td>
<td>Determines an interval in Hours, Minutes, or Days, between two dates and saves the result in a new variable.</td>
<td>Yes</td>
</tr>
</tbody>
</table>

## Relational Actions

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
<th>New in 2010?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Look up a Manager of a User</td>
<td>Retrieves the manager of the specified user from the User Profile database.</td>
<td>Yes</td>
</tr>
</tbody>
</table>
**Document Set Actions**

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
<th>New in 2010?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capture a Version of the Document Set</td>
<td>Creates a new version for the Document Set.</td>
<td>Yes</td>
</tr>
<tr>
<td>Send Document Set to Repository</td>
<td>Sends the document set to another repository – for example a document library in another site collection.</td>
<td>Yes</td>
</tr>
<tr>
<td>Set Content Approval Status for the Document Set</td>
<td>Changes the content approval status and sets a comment.</td>
<td>Yes</td>
</tr>
<tr>
<td>Start Document Set Approval Process</td>
<td>An entire tasking process based on the built-in Approval workflow template, which works on an entire Document Set. This intense action allows you to define various pieces of the process using the Task Process Designer.</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Other Workflow Concepts**

Using SharePoint designer 2010 introduces new functionality you didn’t have in 2007, as well as using some of the same terminology, such as the action of Publishing, or the terms Steps, Conditions, and Actions.

**Save vs. Publish**

When you save a workflow, you are saving it in draft state, where you can come back to it later, but your users will not be able to use the workflow on the sites. To use a completed workflow on the site, you simply press the Publish button to make it live.

The workflow will check itself for errors, save the workflow, create any necessary forms needed for the workflow, and then finally publish it to the site.

**Else-If Branch**

The term **Condition** as used in SharePoint Designer workflows is like the programming term of an **IF** statement. The **Else-If** branch works just the way it does in programming – it provides a second set of conditions to match against. For example, to switch logic based on the color chosen, you might enter the following:

```csharp
If field ‘Color’ equals ‘Red’
    // Perform some actions
Else If field ‘Color’ equals ‘Blue’
    // Perform some actions
Else
    // Perform actions that hit any other color other than Red or Blue
```

With **Else-If** branches, you don’t have to use the same comparator twice – i.e. you don’t have to compare different values to the same field:
If field ‘Sub Total’ is greater than or equal to ‘500.00’
  // Perform some actions
Else If field ‘Shipping and Handling’ is greater than ’25.00’
  // Perform some actions
Else
  // Perform actions that hit any other condition than the ones stated

**Parallel Block**

By default, when you add actions and conditions to the workflow, they are run one after another – in serial. However, you can use a Parallel Block if you want conditions and actions to run at the same time.

You would use this functionality when you don’t wish for the operation to pause the workflow – for example, when assigning a task.

**Impersonation Step**

Impersonation Steps are placed in the workflow in the same way that you would place regular steps. They have conditions and actions within, just the same. However, the actions within the Impersonation steps will run as if it were the identity of the user who authored the workflow.

**CAUTION:** Using this feature gives the users of the workflow rights they wouldn’t normally have. They may be able to add list items to a list they don’t have rights to, for example. Use with extreme care.

There are some actions, as you’ve already seen, that only run under the elevated privileges of the Impersonation Step.

**Publishing Globally**

If you have full control permission at the top-level site on which you are working, you can choose to publish a workflow globally. Publishing globally places the workflow into the Global workflows catalog, making it reusable on every site within the site collection.

**Initiation Form Parameters**

There might be times when you wish to ask your user for information that isn’t stored anywhere, or to have your user choose from among several options presented to them, both of which will determine the course of the workflow. Initiation Forms and their parameters allow you to query the user for that information.

When you click on the Initiation button, you are given the choice to add one or more parameters that are questions you are asking your users. For each parameter you wish to have, you will type in the name of the parameter, you will choose the type of field it should be (i.e. Text, Number, Date, etc.), and the default value, if any. You also specify whether each parameter will be allowed to have nothing filled in (“Allow blank values”).
The form that is used to present the questions to the user is automatically created when you publish the form. If you are running SharePoint Server 2010, then all of the forms that are automatically generated are using InfoPath 2010. The forms are then later customizable in InfoPath.

**Local Variables**
Variables give you a place to store values, such as storing information to pass between steps, holding interim calculations, holding item IDs of items you create, etc. Information held in variables are accessed in the workflow in the same manner that initiation form parameters are accessed – the difference between the two are that variables are internal and not exposed to the users.

You can choose from among the following data types for your variables:

- Boolean
- Date/Time
- Integer
- List Item ID
- Number
- String

**Association Columns**
Columns (i.e. fields) called association columns can be declared within reusable workflows. When the workflow is deployed to a list or library, the association columns get added to that list or library as well.

**Start Options**
The start options are found in the Workflow General Settings page, which can be reached by pressing the Workflow Settings button on the Ribbon. Start options give you three choices:

- Whether or not you wish to allow users to start the workflow manually
- Whether or not to start the workflow automatically every time a new item is created
- Whether or not to start the workflow automatically every time an item is edited.

Keep in mind, that when you set the workflow to automatically start, no Initiation Form is presented to the user, and if you have any initiation parameters they are set to their default values.

**Best Practices**
- **BEFORE** creation of workflows:
  o Create all lists you will need all of the fields in each list.
  o Create and define any Document Sets you will be using.
  o If you wish to use or change Content Approval Status in your Workflow, you need to turn it on in the Settings of your list.
If you’re using a workflow on an InfoPath Forms library, make sure to “Promote” the fields in your InfoPath form that you want to be able to use in a Workflow. You won’t be able to update the fields’ data by default, but you will be able to use them in conditions, and in actions such as an Email TO field.

- BEFORE saving your workflow:
  - If you are using a List Workflow, change the association of the Task List to use a new task list.
  - Although you can, don’t ever modify the out-of-the-box workflows – instead make a copy and modify.

Tips and Tricks
- Try to use Reusable workflows – if you don’t then you can’t move the workflow.
- Use the “Log to History List” action often – at the beginning, at the end, and at least once per step. This way your users will always know where they are in the course of the workflow progress.
- Try not to hard-code any user names. Set up a list with the appropriate person’s user name or email address in order to use that in the workflow. That way if the position or person ever changes, you can just update it in the list.

Workflow Advanced Concepts
Although some of it is the same as in SharePoint Designer 2007, there are many new and exciting things you can accomplish with SharePoint Designer 2010 and Workflows.

Working with the “Define Workflow Lookup” Dialog
The “Define Workflow Lookup” dialog is one of the most potentially confusing dialog box. It is used, for example, when you are comparing values to any value in a condition, doing a variable or list lookup, or after a Task operation, to copy or compare the values resultant from that task.
This dialog works very much like a SQL Select statement:

SELECT [Assigned To] FROM Tasks WHERE Tasks.ID = 2

Let’s go through the different parts of this dialog.

Data Source
There are several choices you can use for the first dropdown box in the “Define Workflow Lookup” dialog. Using our SQL Statement analogy, these are like the tables that you choose from:

SELECT [Assigned To] FROM Tasks WHERE Tasks.ID = 2

- **Current Item**
  This choice allows you to retrieve values from the columns in the list you’re running the workflow on, and specifically on the very item that the workflow is currently running on. Selecting this choice will keep the dialog short and only ask you for the field you want to return, and how to return the field.

- **Workflow Variables and Parameters**
  If you have any Initiation Parameters or Variables in which you have data stored, this is where you would retrieve that information. Selecting this choice will keep the dialog short and only ask you for the parameter or variable you want to return, and how to return it.

- **Workflow Context**
  This choice allows you to choose Association information about the workflow, such as who associated the workflow with the list, and whether to start the workflow on Item Creation or Item Change. You can also choose some information about the workflow, such as when it was last run, who the current user is, or what the URL is to the item or to the Workflow Status page. Selecting this choice will keep the dialog short and only ask you for the field you want to return, and how to return the field.

- **Association: History List**
  This choice allows you to retrieve information from the item’s workflow history by looking directly into the Workflow History list associated with that workflow. Selecting this choice will expand the dialog and make you choose which item you want in that list by specifying a field and value to use to search for the right item.

- **Association: Task List**
  This choice allows you to retrieve information from the Task list associated with that workflow. For example if you used the “Collect Data from a User” action, that would have created a task for a user; you then look at the associated Task list to retrieve that data after. Selecting this choice will expand the dialog and make you choose which item you want in that list by specifying a field and value to use to search for the right item.

- **Current List**
  This choice allows you to retrieve information from the current list that you’re running the workflow on, but from an item other than the one you’re running the workflow on (Current Item). Selecting this choice will
expand the dialog and make you choose which item you want in that list by specifying a field and value to use to search for the right item.

- **User Profiles**
  This choice allows you to retrieve information in a user’s profile; for example, if you wanted to find a manager’s manager. Selecting this choice will expand the dialog and make you choose which person you want by specifying an Account Name (only).

- **[Other Lists and Libraries in your site]**
  This choice allows you to retrieve information from any list in the same site where you’re running the workflow. Selecting this choice will expand the dialog and make you choose which item you want in that list by specifying a field and value to use to search for the right item.

**Field From Source**
This second dropdown in the “Define Workflow Lookup” dialog is simply the choice of the field (or parameter or variable) that you want to get from the source you chose in the first dropdown. This Field (or parameter or variable) is what is used as the return value. Using our SQL analogy, this is the field that you selected:

```
SELECT [Assigned To] FROM Tasks WHERE Tasks.ID = 2
```

**Return Field As**
This third dropdown is very often disabled. I have only seen it enabled when you choose a field that represents a person. In that case you can choose to return the person in the following manners:

- As String (comes through as ID Number and then User Name – ex: 116;#Susan Hernandez)
- Display Name
- Email Address
- Login Name
- User Id Number

**Find the List Item / Field**
The first dropdown in the bottom section is the field from the source you chose above that you want to use to filter out exactly which item to retrieve. In our SQL analogy, it is the field immediately after the WHERE statement:

```
SELECT [Assigned To] FROM Tasks WHERE Tasks.ID = 2
```

**Find the List Item / Value**
The last text box (or sometimes dropdown) is where you choose or type in the value that you want to find from the field you chose immediately above (Find the List Item / Field). Note that you can press the button to use yet another lookup to find the value to use. In the SQL analogy, it is the value that you are comparing to in the WHERE statement:

```
SELECT [Assigned To] FROM Tasks WHERE Tasks.ID = 2
```
Define Workflow Dialog box In Action
Let’s go through an example which will show how this dialog box might be used to retrieve the results of a Task action – namely the Collect Data from a User action.

1. Insert the Collect Data from a User action

2. Set up the Collect Data from a User action
   a. Click data.
   b. Click Next.
   c. For the Name, type in Accounting cross-check
   d. For the Description, type in Please cross-check the dollar amounts being entered and validate that the total cost is correct.
   e. Click Next.
   f. Click Add…
   g. For the Field Name, type in Accounting Validation
   h. For the Description, type in Have you validated the total cost? Please choose the appropriate response.
   i. For the Information Type, choose “Choice (menu to choose from)”
   j. Click Next.
   k. Type in the following 2 choices, one on each line:
      Total Cost is Valid
      Total Cost is Invalid
   l. Uncheck “Allow blank values?”
   m. Click Finish.
   n. Click Finish.
   o. Click this user.
   p. Choose the Accounting Associate (choose a test account).
3. Set the Condition

   a. Create a new Step.
   b. Choose the “If any value equals value” condition.
   c. Click on the first value.
   d. Click \( \text{value} \).
   e. For the Data Source, choose “Association: Task List”.
   f. For the Field from source, choose “Accounting Validation”.
   g. In the “Find the List Item” section, for Field, choose “ID”.

   h. For Value, click \( \text{fx} \).
   i. For the Data Source in the “Lookup for Integer” dialog, choose “Workflow Variables and Parameters”.
For the Field from source, choose “Variable: collect”.

Click OK.

Click OK.

Click on the second value.

Choose “Total Cost is Valid”.

Here, you would complete the step by adding actions you wish to happen when the return value of the Task is “Total Cost is Valid”. You would further enter an Else clause to set the actions to run when the value of the Task is “Total Cost is Invalid”.

**Working with the Approval Process Action**

One of the best new additions to workflow, in my opinion, is the Start Approval Process action.

The first thing to notice when you create an approval process is that quite a few new variables are added for you automatically. One variable to point out is IsItemApproved (Boolean). That stores whether all participants have approved the item.

If you press the word Approval, you will be taken to the General Information screen. This is where you can set some General settings, add Task Form Fields, edit Task Outcomes, and change the behavior of the workflow.
**Task Information**

The Task Information section holds the Name of the workflow as well as the Owner of the process. If the Owner is set, that's the user that Impersonation Steps run as.

**Settings**

You have 3 checkboxes here:

- **Only allow task recipients and process owners to read and edit workflow tasks**
  
  This is an important one – if this is not checked (and by default it's not), then anyone with contributor rights to the task list can update someone else's task. This may be good and bad – it would work fine if the only contributors to the task list were alternate approvers, for example.

- **Reassignment**
  
  By checking this check box, you allow the assigned Approver to click the Reassignment command to send the task to another person.

- **Change Requests**
  
  By checking this check box, you allow the assigned Approver to make a request to the Originator to have the item be Changed before they will approve or reject it.
Task Form Fields
This section allows you to add custom fields to the Task Form that is displayed to the assigned Approvers when they are going to approve or reject an item. You can mark them as required (by unchecking “Allow blank values”) and you can add the new field to the Default view.

You can choose from among the following data types:

- Single line of text
- Multiple lines of text
- Currency
- Date and Time
- Choice
- Yes/No
- Person or Group
- Hyperlink or Picture

Task Outcomes
This allows you to change the outcomes or add new ones. Basically each Outcome is one button that the assigned Approver is allowed to click on the Task form. The amount of text you can put in is very limited.

NOTE: You have to change the completion conditions and/or the behavior of the overall task process to correctly reflect the right outcomes and approval statuses for the workflow item if you add new Task Outcomes.

Customization
I left the best for last – this is where things get interesting. Here we have 4 links. The first one is just to return to the workflow. But the next 3 open up entire processes for you to customize to your needs.
Change the completion conditions for this task process

This process is what happens every time a task has completed. By default it has all the conditions that need to be met for the entire workflow to be considered complete. For example, if the number of approved tasks equals the completed task count, then the approved variable is set to Yes. You can add more logic here so that a different set of conditions are required for the workflow to be marked as complete or you can leave the existing conditions as-is.

**NOTE:** When you are in the process editors for these task-specific processes, you can now choose from among some new actions that are related to tasks, depending on where you are in the processes:

- End Task Process
- Set Content Approval Status
- Set Task Field
- Append Task
- Delegate Task
- Escalate Task
- Forward Task
- Insert Task
- Reassign Task
- Request a Change
- Rescind Task
- Send a Task Notification Email
### Change the behavior of a Single Task

<table>
<thead>
<tr>
<th><strong>Before a Task is Assigned</strong></th>
<th>Run these actions before every individual task is created:</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Start typing or use the Insert group in the Ribbon.)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>When a Task is Pending</strong></th>
<th>Run these actions after every individual task has been created:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>If</strong> Current Task:External Participant is empty</td>
<td>Log Task created for $%$Current Task:Assign... to the workflow history list</td>
</tr>
<tr>
<td></td>
<td>then Email task notification to Current Task:Assigned To</td>
</tr>
<tr>
<td><strong>Else</strong></td>
<td>Log Task created for $%$Current Task:Assign... to the workflow history list</td>
</tr>
<tr>
<td></td>
<td>then Email task notification to Workflow Context:Initiator</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>When a Task Expires</strong></th>
<th>Run these actions every time an individual task is still incomplete past its due date:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Email task notification to Current Task:Assigned To</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>When a Task is Deleted</strong></th>
<th>Run these actions every time an individual task is deleted before it is completed:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Log Task assigned to $%$Current Task:Assign... to the workflow history list</td>
</tr>
<tr>
<td></td>
<td>then Log Task assigned to $%$Current Task:Assign... to the workflow history list</td>
</tr>
<tr>
<td></td>
<td>then Email Current Task:Assigned To</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>When a Task Completes</strong></th>
<th>Run these actions every time an individual task is completed:</th>
</tr>
</thead>
</table>
This process allows you to customize the various stages of a task’s lifecycle – when a task is assigned, when it’s pending, when it expires, when it’s deleted, and when it completes.

For example, here’s where you would change the email notifications for when the task is created or for when it expires. You may also choose to escalate the task if it has gone past its due date.

The section “When a Task Completes” is where you would check the outcomes and perhaps take other actions based on which outcome has been selected by the approver.

**Change the behavior of the overall task process**
In this process, you can change what happens when the task process starts, when it’s running (including deleted and changed tasks), when it’s canceled, and when it completes.

By default, there are a number of things happen when the task process completes; if the IsItemApproved variable equals Yes, then the workflow status is set to Approved. If the enable content approval parameter is Yes, then content approval for the workflow item is set to Approved.

This is also the area that you might have to change if you add different Outcomes to the process.

Approval Process In Action
Let’s go through an example of customizing the default Approval Process action.

1. Add the action “Start Approval Process”

2. Click Approval

3. Add a new variable
   a. Click on Local Variables
   b. Click on Add...
   c. For Name, type in ApprovalOutcome
   d. Click OK
   e. Click OK

4. Add an Outcome
   a. In the Task Outcomes section, click “New”
   b. For the new Name, type in Appr With IT Review
   c. For the Task Form Button, type in Appr. With IT Review

5. Change the Individual Task Behavior
   a. Click “Change the behavior of a single task”
   b. Scroll down to the “When a Task Completes” section
   c. Change Approved
i. Insert an “Set Workflow Variable” action just above the logging action in the condition if the Current Task’s Outcome is Approved

```
When a Task Completes

Run these actions every time an individual task is completed:

If Current Task:Outcome equals Approved

Set workflow variable to value

then Log Task assigned to %Current Task:Assign... to the workflow history list
```

ii. Click workflow variable  

iii. Choose “Variable: ApprovalOutcome”  

iv. Click value  

v. Type in Approved  

d. Change Rejected  

i. Insert an “Set Workflow Variable” action just above the logging action in the condition if the Current Task’s Outcome is Rejected

```
If Current Task:Outcome equals Approved

Set Variable: ApprovalOutcome to Approved

then Log Task assigned to %Current Task:Assign... to the workflow history list

Else if Current Task:Outcome equals Rejected

Set workflow variable to value

then Log Task assigned to %Current Task:Assign... to the workflow history list
```

ii. Click workflow variable  

iii. Choose “Variable: ApprovalOutcome”  

iv. Click value  

v. Type in  

Rejected  

e. Add a new condition
i. Add an Else If Condition underneath the If and Else If Conditions

```
Else if Current Task:Outcome equals Rejected

Set Variable: ApprovalOutcome to Rejected

then Log Task assigned to [Current Task:Assignee] to the workflow history list

If Variable: CancelonRejection equals Yes

Set Variable: CompletionReason to [Task Process:Process Name] on [Task Name]

then End Task Process
```

Else if [value] equals [value]
(Start typing or use the Insert group in the Ribbon.)

ii. Click `≠`

iii. Choose “Current Task: Approval”

iv. Choose “Outcome”

v. Click OK

vi. Click the second `≠`

vii. Type in “Approved With IT Review”

f. Add the Actions to the condition

i. Insert a “Set Workflow Variable” action in the new Else If branch

ii. Click `workflow variable`

iii. Choose “Variable: ApprovalOutcome”

iv. Click `value`

v. Type in “Approved With IT Review”

vi. Insert a “Log to History List” action

vii. Click on this message

viii. Type in “Task completed – Outcome: Approved With IT Review”

6. Change the Completion Conditions

a. On the Breadcrumb, click Approval
b. Click “Change the completion conditions for this task process”

c. Change the Completion Condition
   i. Click on Task Process Results: Number of Approved
   ii. Change the Data Source to “Workflow Variables and Parameters”
   iii. Set the Field from source to “Variable: ApprovalOutcome”
   iv. Click OK
   v. Click value
   vi. Type in Approved

![Check the Completion Conditions](image)

```plaintext
If Task Process:Future Task Count equals 0
and Task Process:Active Task Count equals 0

If Variable: ApprovalOutcome equals Approved

Set Variable: IsItemApproved to Yes

then Set Variable: CompletionReason to [%Task Process:Process Name%] on [%Ta...

then End Task Process
```

d. Add an Else If condition
   i. Add an Else If condition below the IF condition
   ii. Click on value
   iii. Click
   iv. Set the Data Source to “Workflow Variables and Parameters”
   v. Set the Field from source to “Variable: ApprovalOutcome”
   vi. Click OK
   vii. Click on the second value
   viii. Type in Approved With IT Review

e. Add the Action
   i. Add a “Set Workflow Variable” action
ii. Click on workflow variable

iii. Choose “Variable: IsItemApproved”

iv. Click on value

v. Click “Yes”

From here, you would use the IsItemApproved and the ApprovalOutcome variables in the section “Change the behavior of the overall task process”. From there, in the “When the Task Process Completes” section, you might wish to set the Content Approval of the item to Pending if the ApprovalOutcome is “Approved With IT Review”.

Finally, after the approval process is done, in the following step you can query the ApprovalOutcome variable to see if IT Review is required.

Other Things to Consider

- Workflow History is NOT SUITABLE for “legal” audit trails – they are stored in SharePoint Lists and therefore are able to be tampered with. Use SharePoint’s Audit Log feature instead.

- Things not covered in this session:
  - Writing Workflows using Visio
1) Exporting workflows from Visio
2) Importing Visio workflows into SharePoint Designer
   o Modifying Workflow Forms using InfoPath 2010
      1) Initiation Forms
      2) Task Forms
      3) Association Forms
   o Saving as a template – exporting to a WSP file (SharePoint Solution Package)

**Resources**