|  |
| --- |
| Microsoft SharePoint 2013 - Hands-on Lab |
| Workflow |
| Verified Against Build #15.0.4420.1017 |

|  |
| --- |
| Microsoft  Version 1.0  August 14, 2012 |

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

The names of manufacturers, products, or URLs are provided for informational purposes only and Microsoft makes no representations and warranties, either expressed, implied, or statutory, regarding these manufacturers or the use of the products with any Microsoft technologies. The inclusion of a manufacturer or product does not imply endorsement of Microsoft of the manufacturer or product. Links are provided to third party sites. Such sites are not under the control of Microsoft and Microsoft is not responsible for the contents of any linked site or any link contained in a linked site, or any changes or updates to such sites. Microsoft is not responsible for webcasting or any other form of transmission received from any linked site. Microsoft is providing these links to you only as a convenience, and the inclusion of any link does not imply endorsement of Microsoft of the site or the products contained therein.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Copyright © 2012 Microsoft Corporation. All rights reserved.

Microsoft and SharePoint are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

# Technical Background

This section is intended for labs with technical background information that would be of interest to lab consumers.

## Introduction

SharePoint 2013 represents a major shift in the architecture of SharePoint’s interaction with Workflow Foundation. New in this release is the incorporation of Windows AppFabric, both in the cloud and on the server. In this lab you will learn how to create workflows using the various tools provided in this new framework.

## Create Lab SharePoint Site Collection

In the files provided with the hands on lab, run the batch file called **SetupModule.bat** by double clicking it. This file will execute a PowerShell script that will create a new site collection at <http://intranet.contoso.com/sites/Workflow>.

|  |
| --- |
| Description: C:\Users\vesaj\Pictures\DVD_ART36\Artwork_Imagery\Icons - Illustrations\_ SUPER VISTA STYLE\yield sign red white exclamation point.png **Important** |
| *It is important you run this batch file to create the site collection before working through any of the exercises as the exercises contain instructions for working with this specific site collection at the specific URL created by the script.*  *Further, while this lab requires you to be logged in using a non-administrator account, you should ensure you are logged in as the administrator account when running this script and granting the user access to the site as that user does not have rights to login to the site.* |

The script will first check to see if there is already a site collection at the specified address. If there is it will delete the site collection before creating it. Therefore if you run into problems with the lab, feel free to rerun the batch file to reset the environment and restart the exercise.

# Introduction

## Estimated time to complete this lab

30 minutes

## Objectives

After completing this lab, you will be able to:

* Learn the fundamentals of workflow in SharePoint 2013
* Learn to use SharePoint Designer to create workflows in SharePoint 2013
* Learn to use Visual Studio to create workflows in SharePoint 2013

## Computers in this lab

This lab uses virtual machines as described in the following table. Before you begin the lab, you must start the virtual machines and then log on to the computers.

|  |  |
| --- | --- |
| **Virtual Machine** | **Role** |
| {Supplied by Instructor} | Domain Controller |
| {Supplied by Instructor} | Actual SharePoint environment with Office client and other required software. |

All user accounts in this lab use the password {Supplied by Instructor}.

## Lab Precursor

Before performing this lab, the SharePoint farm’s User Profile Service Application must contain valid accounts, specifically one used to create and execute the workflow. In the lab the user Dan Jump (CONTOSO\danj) is used. This issuer must also be a member of the **Site Collection Administrators** group.

## Lab Setup Script

In the files provided with the hands on lab, run the batch file called **SetupModule.bat** by double clicking it. This file will execute a PowerShell script that will create a new site collection at <http://intranet.contoso.com/sites/Workflow>.

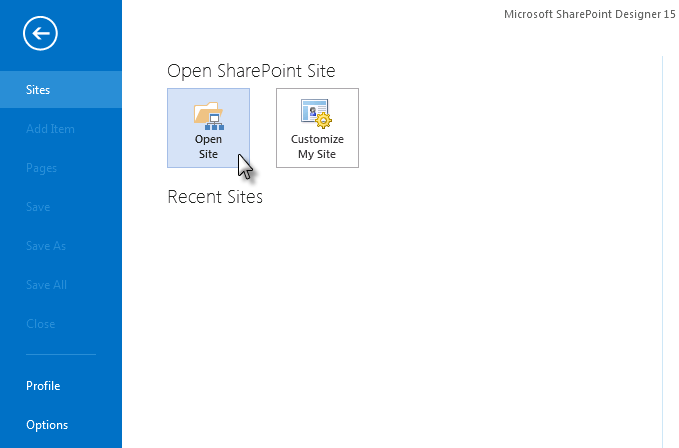
# Exercise 1: Creating a Workflow using a Task Action

In this exercise you will create a workflow using SharePoint Designer 2013 that uses the task action.

## Task 1: Open SharePoint Designer 2013

In this task you will open a site using SharePoint Designer 2013.

1. Open SharePoint Designer 2013 (**Start ⮚ All Programs ⮚ SharePoint ⮚ Microsoft SharePoint Designer 2013**).
2. After SharePoint Designer 2013 launches, click the **Open Site** button. When prompted, enter the URL of the site to connect to: <http://intranet.contoso.com/sites/Workflow> and click **OK**.

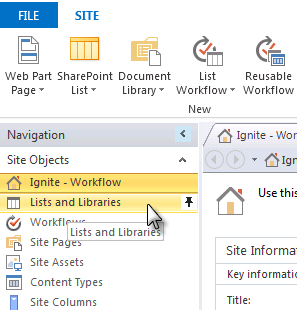


1. Click the **Workflows** item in the left-hand **Navigation** pane.

## Task 2: Create Announcement List

In this task you will create an Announcement list using SharePoint Designer 2013.

1. In the **Navigation** pane in the left-hand side of SharePoint Designer, select **Lists and Libraries**.



1. Using the ribbon, click the **SharePoint List** button and select **Announcements** from the **Generic List** category.

* **Name:** Announcements
* **Description:** List used for testing workflows.

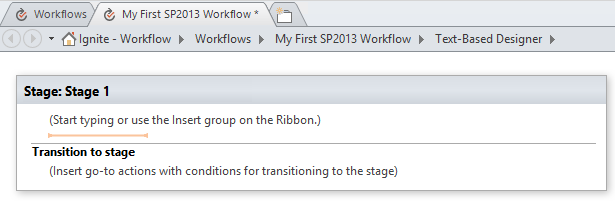
## Task 3: Create a Workflow

In this task you will create a workflow using SharePoint Designer 2013.

1. In the **Navigation** pane in the left-hand side of SharePoint Designer, select **Workflows**.
2. From the ribbon’s **New** group, select **List Workflow ⮚ Announcements**.
3. In the **Create List Workflow – Announcements** dialog, set the following values and click **OK**:

* **Name:** My First SP2013 Workflow
* **Description:** This is my first SharePoint 2013 workflow.
* **Platform Type:** SharePoint 2013 Workflow

1. In the **Text-Based Designer**, ensure the orange flashing horizontal line is located immediately after the text “(Start typing or use the Insert group on the Ribbon.)”… if it isn’t use the mouse to click just below that text to place it there:



1. Add a Log to History List action by selecting the **Workflow** tab to activate the ribbon, click the **Action** button in the **Insert** group and select **Log to History List** from the **Core Actions** group:

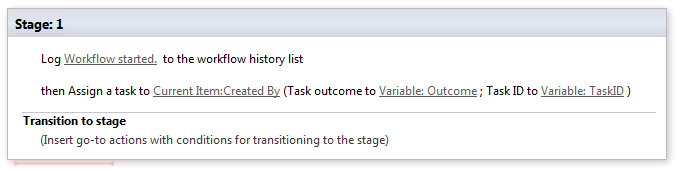


1. In the designer, click the blue underlined **“message”** text and enter the following: **Workflow Started**.

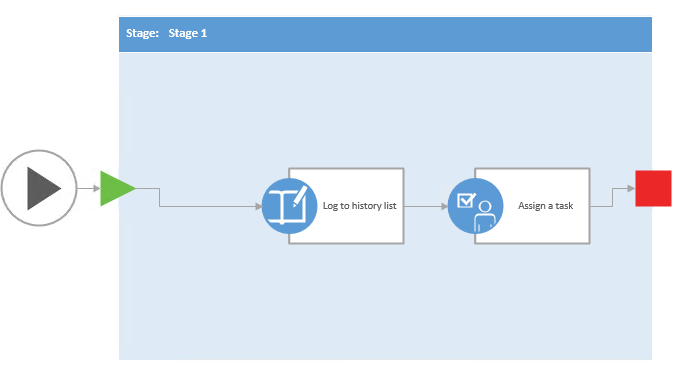
## Task 4: Add a Task Action

Add a task action to the workflow that was just created in the previous task.

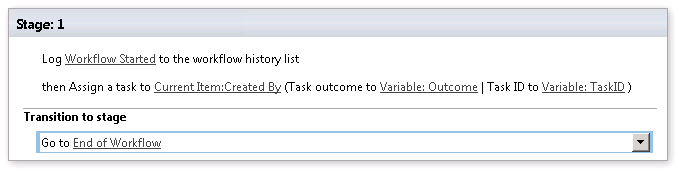
1. Make sure the orange horizontal line is below the **Log Workflow Started to the workflow history list** action added in the last task and add an **Assign a task** action by selecting the **Workflow** tab, click the **Action** button in the **Insert** group and select **Assign a task** from the **Task Actions** group.
2. Click the **this user** text to create the task:
   * Click the **[…]** button to the right of the **Participant** textbox.
   * In the **Select User** dialog, select **User who created current item** and click the **Add** button followed by **OK**.
   * Enter **Approval Requested Task** in the **Task Title** textbox.
   * Click **OK**.



1. Switch the view of the workflow by clicking the **Workflow** tab to activate the ribbon and click the **Views** button in the **Manage** group.
2. Notice the workflow is now displayed in a graphical representation rather than a text view:



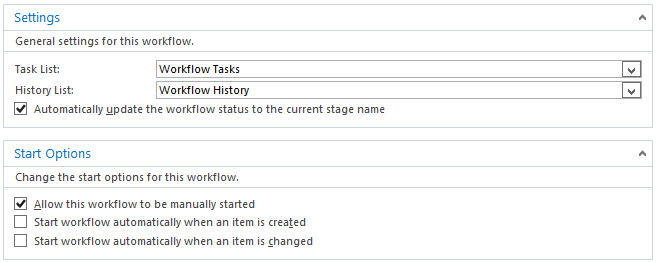
1. Go back to the **Text-Based Designer** by clicking the **Views** button in the **Manage** group and place the cursor in the **Transition to Stage** section.
2. Add a **Go To a stage** action.
3. Click **a stage** and select **End of Workflow**.



1. Go back to the workflow summary page to view the settings by clicking the **My First SP2013 Workflow** in the navigation above the designer:



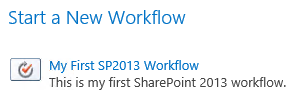
1. On the summary page:
   * In the **Start Options** group, ensure the **Allow this workflow to be manually started** is checked but the two other **Start workflow automatically when…** are unchecked.



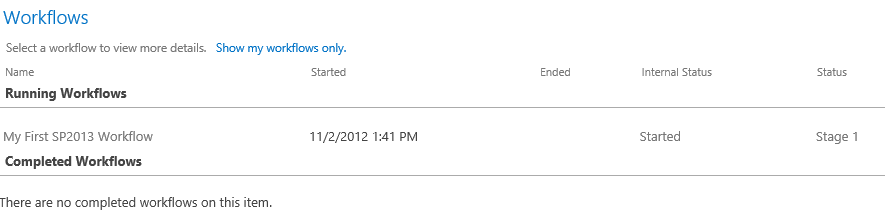
## Task 5: Deploy & Test the Workflow

In this task you will deploy and test the workflow in an existing SharePoint site.

1. Save all changes to the workflow by clicking the **Workflow Settings** tab in the ribbon and then clicking the **Save** button in the **Save** group.
2. After the workflow has been saved, click the **Publish** button in the **Save** group.
3. Open the site in a browser by clicking the name of the site in the left-hand **Navigation** pane (*the entry just above the* ***Lists and Libraries*** *entry*). In the ribbon click the **Site** tab and in the Manage group, click the **Preview in Browser** button.
4. In the browser, navigate to the Announcements list by clicking the **Site Contents** link in the **Quick Launch** and then the **Announcements** app in the lists of Apps.
5. Add a new item to the list by clicking the **new announcement** link. Set the title of the announcement to **Testing My Workflow** and click **Save**.
6. Initiate the workflow by selecting the item in the list. Within the ribbon click the **Items** tab and in the **Workflows** group, click the **Workflows** button.
7. In the section **Start a New Workflow**, click the **My First SP2013 Workflow** to initiate your workflow.



1. Go back to the workflow status page by selecting the item again and click the **Items** tab and in the **Workflows** group, click the **Workflows** button.
2. This page will show not only the workflows that can be started (at the top), but also the running & completed workflows. You should see one instance of your workflow that has started. If not, refresh the page every few seconds until you see it listed:



1. To complete the workflow, navigate to the Workflow Tasks list, find the task that was just created, and mark it as complete by clicking the **Approved** button.
2. Now go back to the workflow status page.
3. Once the workflow has completed which happens when the task has been completed you will see the status has changed to **Completed**.



# Exercise 2: Using Stages in SharePoint Designer 2013 Workflows

* 1. In this exercise you will create a new workflow that leverages the new stages capability. In this scenario the simple workflow will be used to demonstrate transitioning between stages.

## Task 1: Create a Workflow and Initial Stages

In this task you will create a workflow and add the initial stages to the workflow.

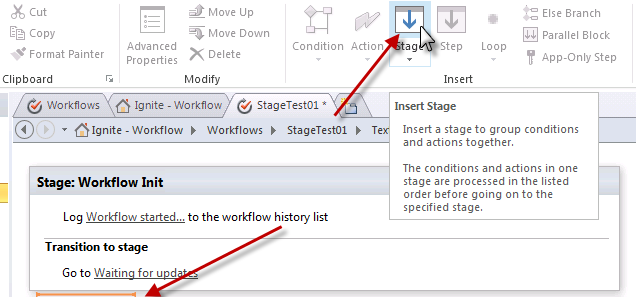
1. Open **SharePoint Designer 2013** (Start ⮚ All Programs ⮚ SharePoint ⮚ Microsoft SharePoint Designer 2013).
2. After SharePoint Designer 2013 launches, click the **Open Site** button. When prompted, enter the **URL** of the site to connect to: <http://intranet.contoso.com/sites/Workflow> and click **OK**.

|  |
| --- |
| Description: C:\Users\vesaj\Pictures\DVD_ART36\Artwork_Imagery\Icons - Illustrations\_ SUPER VISTA STYLE\yield sign red white exclamation point.png **Important** |
| *If the site does not contain an Announcements list, create it by following Task 2 in Exercise 1 above.* |

1. Click the **Workflows** item in the left-hand **Navigation** pane.
2. From the ribbon’s **New** group in the **Workflow** tab, select **List Workflow** ⮚ **Announcements**
3. In the **Create List Workflow – Announcements** dialog, set the following values and click **OK**:

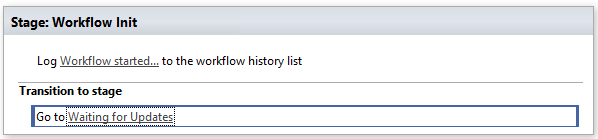
* **Name:** My First Staged Workflow
* **Platform Type:** SharePoint 2013 Workflow

1. Rename the first stage of the workflow by clicking **Stage: 1** header bar and renaming Title to **Workflow Init** and click **OK**.
2. Add a **Log to History List** action by click the **Actions** button in the ribbon and finding the activity in the **Core Actions** category.
   1. Set the **message** of the activity to “Workflow started…”
3. Add a new stage by clicking the area below the **Workflow Init** stage and then click the **Stage** button in the ribbon:



* 1. Rename the stage to **Waiting for Updates**

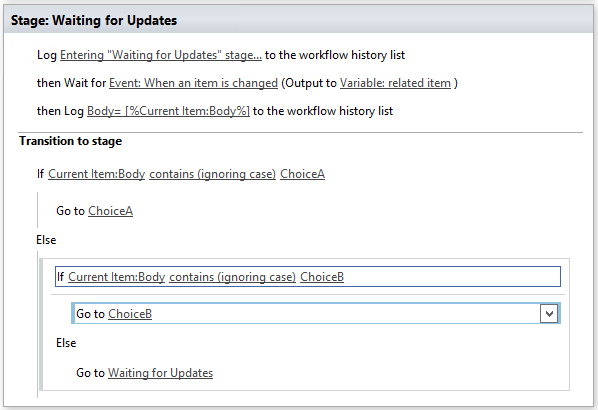
1. Repeat this process of creating a stage by creating the following additional stages:
   * Stage: **ChoiceA**
   * Stage: **ChoiceB**
   * Stage: **Workflow End**
2. Place the cursor in the **Transition to Stage** section of stage **Workflow Init**.
3. Add a **Go to a stage** action.
4. Click **a stage** and select **Waiting for Updates**.



## Task 2: Complete the Waiting for Updates Stage

In this task you complete the body of the Waiting for Update stage.

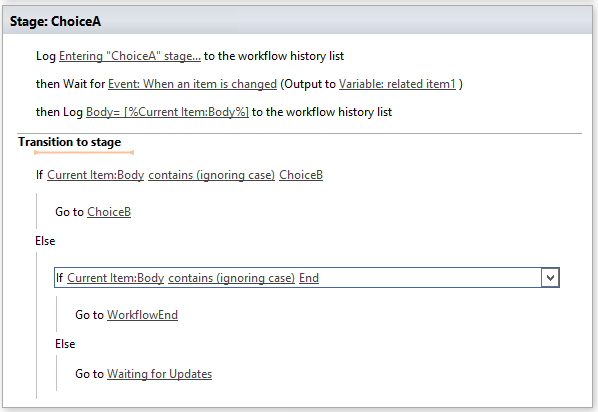
1. Within stage **Waiting for Updates**, add three actions in the following order:
   * Log to History List
   * Wait for Event in List Item
   * Log to History
2. For the first **Log to History List**, change the **message** to **Entering “Waiting for Updates” stage…**.
3. For the **Wait for Event in List Item**, click the **this item event** link.
4. In the **Choose List Item Event** dialog, enter the following values:
   * Event: **Event: When an item is changed**
   * List: **Current List**
5. For the second **Log to History List**, click **[…]** button next to the **message** link.
   1. Add **Body =** to the text area
   2. Click the **Add or Change Lookup** button
   3. In the Lookup for String dialog, enter the following:
      * Data Source: **Current Item**
      * Field from Source: **Body**
      * Return field as: **As String**
6. Within the same stage, go to the section on **Transition to stage**. Add an if condition by clicking the cursor in that section and click the **Condition** button in the ribbon, selecting **If any value equals value**.
7. In the **if** condition, click the first **value** link and enter the following values:
   * Data source: **Current Item**
   * Field from source: **Body**
8. In the **if** condition, click the **equals** link and select **contains (ignoring case)**.
9. In the **if** condition, click the second value link and enter **ChoiceA**.
10. Under the **then** part of the if condition, add a **Go to a stage** action, click the **a stage** link and select **ChoiceA**.
11. Under the **else** part of the **if** condition, add another if statement similar to the previous one that checks for the value of **ChoiceB** and routes to either the **ChoiceB** or **Waiting for updates stage**.
12. The **Waiting for Updates** stage should now look like the following figure:



## Task 3: Complete the ChoiceA Stage

In this task you complete the body of the ChoiceA stage.

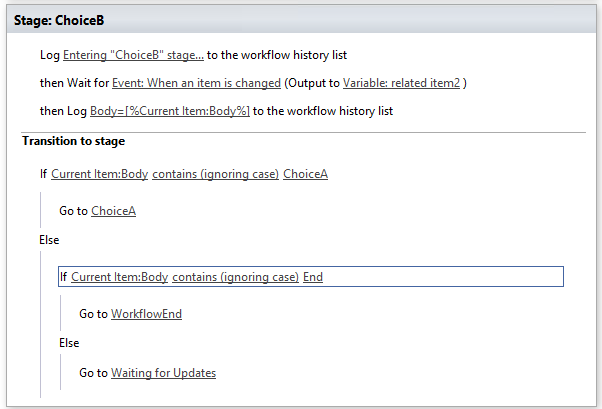
1. Within stage **ChoiceA**, add three actions in the following order:
   * Log to History List
   * Wait for Event in List Item
   * Log to History
2. For the first **Log to History List**, change the **message** to **Entering “ChoiceA” stage…**.
3. For the **Wait for Event in List Item**, click the **this item event** link.
4. In the **Choose List Item Event** dialog, enter the following values:
   * Event: **Event: When an item is changed**
   * List: **Current List**
5. For the second **Log to History List**, click **[…]** button next to the **message** link.
   1. Add **Body =** to the text area
   2. Click the **Add or Change Lookup** button
   3. In the Lookup for String dialog, enter the following:
      * Data Source: **Current Item**
      * Field from Source: **Body**
      * Return field as: **As String**
6. Within the same stage, go to the section on **Transition to stage**. Add an if condition by clicking the cursor in that section and click the **Condition** button in the ribbon, selecting **If any value equals value**.
7. In the **if** condition, click the first **value** link and enter the following values:
   * Data source: **Current Item**
   * Field from source: **Body**
8. In the **if** condition, click the **equals** link and select **contains (ignoring case)**.
9. In the **if** condition, click the second value link and enter **ChoiceB**.
10. Under the **then** part of the if condition, add a **Go to a stage** action, click the **a stage** link and select **ChoiceB**.
11. Under the **else** part of the **if** condition, add another if statement similar to the previous one that checks for the value of **End** and routes to either the **Workflow End** or **Waiting for updates stage**.
12. The **ChoiceA** stage should now look like the following figure:



## Task 4: Complete the ChoiceB Stage

In this task you complete the body of the ChoiceB stage.

1. Within stage **ChoiceB**, add three actions in the following order:
   * Log to History List
   * Wait for Event in List Item
   * Log to History
2. For the first **Log to History List**, change the **message** to **Entering “ChoiceB” stage…**.
3. For the **Wait for Event in List Item**, click the **this item event** link.
4. In the **Choose List Item Event** dialog, enter the following values:
   * Event: **Event: When an item is changed**
   * List: **Current List**
5. For the second **Log to History List**, click **[…]** button next to the **message** link.
   1. Add **Body =** to the text area
   2. Click the **Add or Change Lookup** button
   3. In the Lookup for String dialog, enter the following:
      * Data Source: **Current Item**
      * Field from Source: **Body**
      * Return field as: **As String**
6. Within the same stage, go to the section on **Transition to stage**. Add an if condition by clicking the cursor in that section and click the **Condition** button in the ribbon, selecting **If any value equals value**.
7. In the **if** condition, click the first **value** link and enter the following values:
   * Data source: **Current Item**
   * Field from source: **Body**
8. In the **if** condition, click the **equals** link and select **contains (ignoring case)**.
9. In the **if** condition, click the second value link and enter **ChoiceA**.
10. Under the **then** part of the if condition, add a **Go to a stage** action, click the **a stage** link and select **ChoiceA**.
11. Under the **else** part of the **if** condition, add another if statement similar to the previous one that checks for the value of **End** and routes to either the **Workflow End** or **Waiting for updates stage**.
12. The **ChoiceB** stage should now look like the following figure:



## Task 5: Complete the Workflow End Stage

In this task you complete the body of the Workflow End stage.

1. Within stage **Workflow End**, add a **Log to History** **List** action.
2. Click the **message** link on the **Log to History** **List** action and change it to **Ending workflow**.
3. Within the **Transition to stage** section, add a **Go to a stage** action and set it to the **End of Workflow** stage.

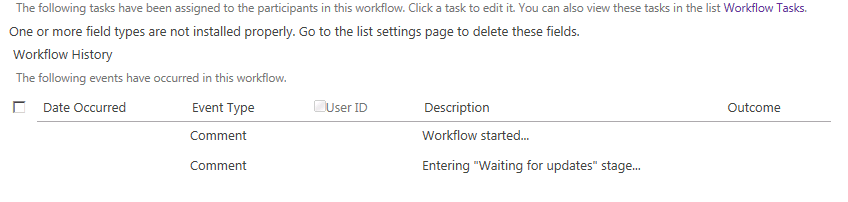
## Task 6: Save, Publish and Test the Workflow

In this task you save, publish and test the workflow you just created.

1. With the workflow still open, click the **Save** button in the ribbon.
2. After saving the workflow, publish it by clicking the **Publish** button. This will make it available for use in the SharePoint announcement list and push the workflow to Windows Azure Workflow.
3. Next it is time to test the workflow. Navigate in the browser to the <http://intranet.contoso.com/sites/Workflow> site and go to the **Announcements** list.
4. Add a new item to the list entering only the **Title** field but leave the **Body** field blank.
5. After adding the item, start the workflow. Do this by selecting the item and clicking the **Workflows** button under the **View** tab in the ribbon.
6. On the **Announcements: Workflows: [item title]** page, click the **My First Staged Workflow** to start the workflow.
7. After a brief delay the browser will redirect you to the Announcements list page again. The workflow will start but this may not be seen for a few seconds so be patient.

|  |
| --- |
| Description: note_ddNote |
| *To verify the workflow has started you can always go to the item that triggered the workflow. Look in the footer of the page where the “Created by” and “Last Modified By” information is. The Last Modified will say something to the effect of “Last modified at [date time] by Workflow on behalf of [user].” If the workflow has started.* |

1. View the details on the workflow by clicking the item in the Announcements list page and then the **Workflows** button in the ribbon.
2. On the **Announcements: Workflows: [item title]** page (*also known as the* ***workflow status page***), scroll down to the section Running Workflows and click the **Started** link.
3. Notice the comments that have been logged from the **Log to History List** activity:



1. Now, go back to the item and enter **ChoiceA** in the **Body** field.
2. Navigate back to the workflow status page to see how the workflow is responding going through the different stages.
3. Next, edit the item to enter **ChoiceB** in the **Body** field and check the status page how it responded.
4. Finally, edit the item to enter **End** in the **Body** field to have the workflow complete.

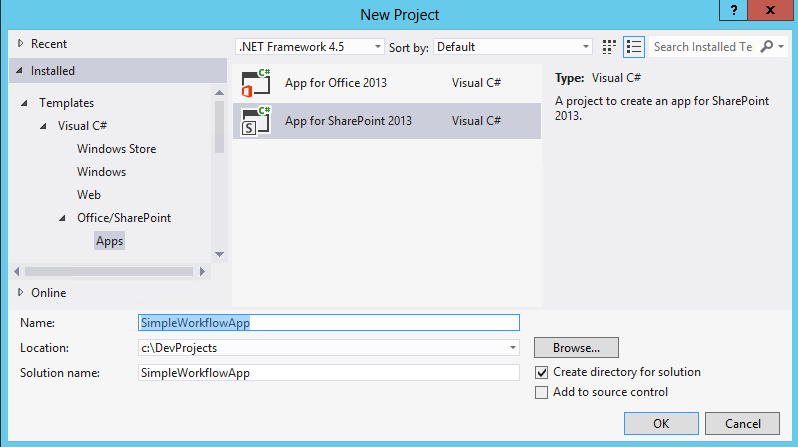
# Exercise 3: Creating a Workflow with Visual Studio

* 1. In this exercise you will create a new workflow using Visual Studio that updates a field in a list item. In this scenario the workflow you create will be entirely declarative unlike previous versions of SharePoint where Visual Studio workflows had a programmatic component to them.

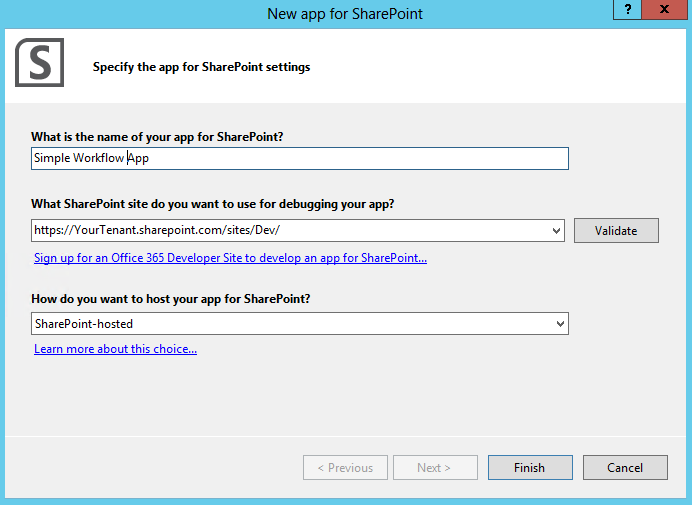
## Task 1: Create Workflow App in Visual Studio

In this task you will create a new SharePoint 2013 project in Visual Studio and add a workflow project item to it.

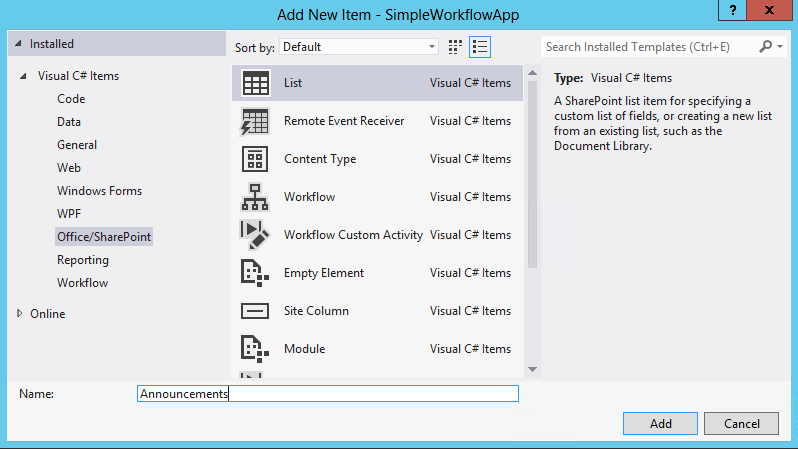
1. Open Visual Studio 2012 as an administrator: **Start ⮚ All Programs ⮚ Microsoft Visual Studio 2012 ⮚** right-click **Visual Studio 2012** and select **Run as Administrator**.
2. Select **File ⮚ New ⮚ Project**.
3. In the **New Project** dialog, select the **Visual C# ⮚ Office/SharePoint ⮚ Apps** category. Pick the project template **App for SharePoint 2013 Project**. Give the project the name **SimpleWorkflowApp**.



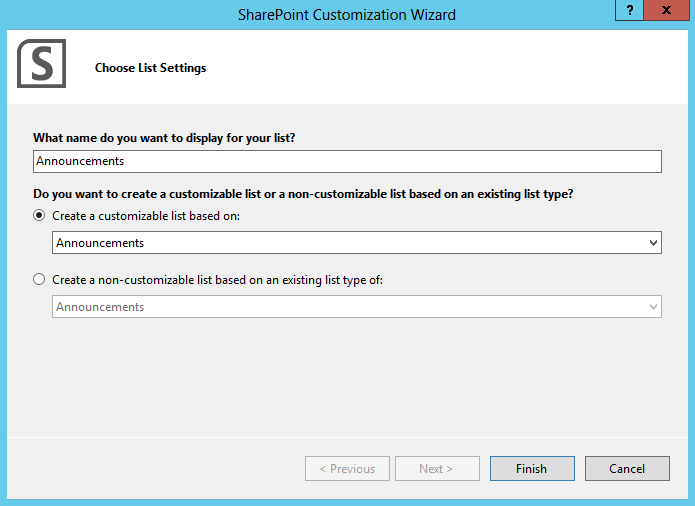
1. In the **New App for SharePoint Wizard** dialog, change the name of the app to **Simple Workflow App.** Enter the URL of the workflow site (<http://intranet.contoso.com/sites/workflow>) as the site to use in debugging, and change the hosting time to **SharePoint-hosted,** finally clicking **OK**.



1. Once the project is created, add a new list to the project. Right-click the project in the **Solution Explorer** tool window and select **Add ⮚ New Item**:



1. In the Add New Item dialog, select **List.** Give it a name of **Announcements** and click **Add**.
2. In the Choose List Settings dialog, choose **Create a customizable list based on:** and choose the **Announcements** list template.



1. The Announcements list will be deployed to the app web, we need to provide a way for the user to navigate to it.
   * Edit the **Default.aspx** page and add the following within the **PlaceHolderMain** content placeholder.

<WebPartPages:WebPartZone runat="server" ID="full"></WebPartPages:WebPartZone>

* + Under the Pages node you will find a file **elements.xml**. Edit that file, replacing the **File** element with the following:

<File Path="Pages\Default.aspx" Url="Pages/Default.aspx" >

<AllUsersWebPart WebPartZoneID="full" WebPartOrder="0">

<![CDATA[

<webParts>

<webPart xmlns="http://schemas.microsoft.com/WebPart/v3">

<metaData>

<type

name="Microsoft.SharePoint.WebPartPages.XsltListViewWebPart,

Microsoft.SharePoint,Version=14.0.0.0,Culture=neutral,

PublicKeyToken=71e9bce111e9429c" />

<importErrorMessage>

Cannot import this Web Part.

</importErrorMessage>

</metaData>

<data>

<properties>

<property name="Title"

type="string">Announcements</property>

<property name="ListDisplayName"

type="string">Announcements</property>

<property name="ChromeType"

type="chrometype">TitleOnly</property>

</properties>

</data>

</webPart>

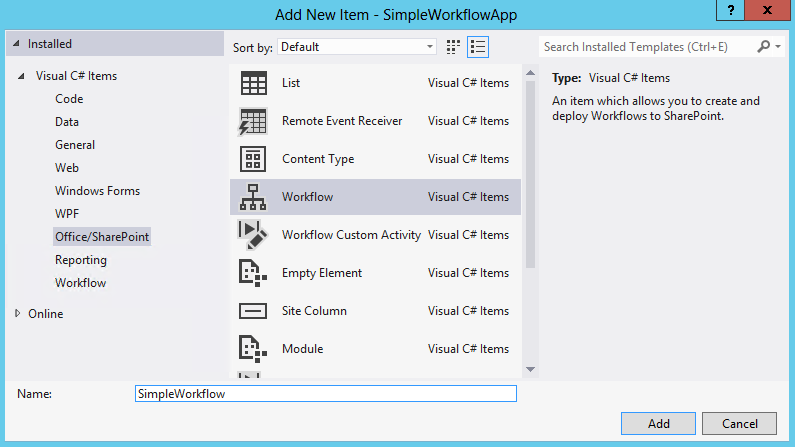
</webParts>

]]>

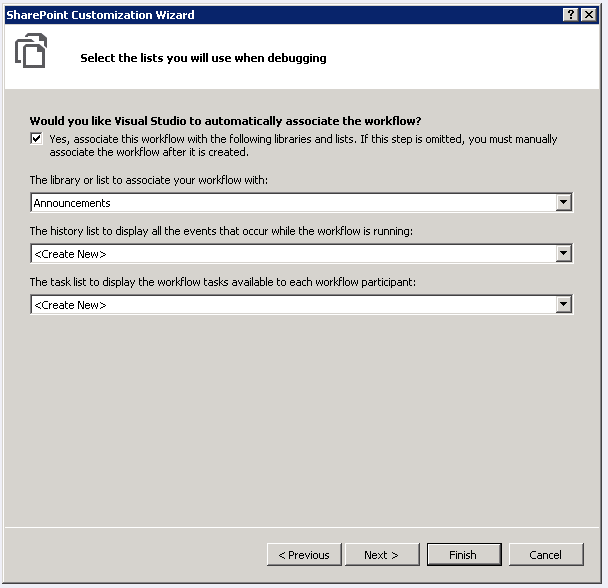
</AllUsersWebPart>

</File>

1. Add a new workflow item to the project. Right-click the project in the **Solution Explorer** tool window, select **Add ⮚ New Item**, and choose **Workflow**.



1. Give it a name of **SimpleWorkflow** and click **Add**.
2. In the **SharePoint Customization Wizard**, select **List Workflow** and change the name to **Simple Workflow**.
3. On the next step of the **SharePoint Customization Wizard**, check the box for **Would you like Visual Studio to automatically associate the workflow?** as well as the following values and click **Finish**.
   * The library or list to associate… : **Announcements**
   * The history list to display…: **Workflow History** or **<Create New>** if Workflow History isn’t an option.
   * The task list to display…: **Workflow Tasks** or **<Create New>** if Workflow Tasks isn’t an option.

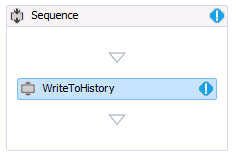


1. In the next window, choose to start the workflow when **a user manually starts the workflow** and click Finish.

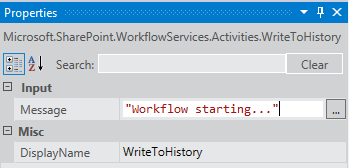
## Task 2: Add Activities to Workflow

In this task you will add and configure activities to the workflow.

1. Add a **WriteToHistory** activity to the workflow by opening the **SP – Current Context** section within the **Toolbox** tool window. To add the workflow click and drag it into the **Sequence** in the designer where it says **Drop activity here**.



1. Change the message by selecting the activity and within the **Properties** tool window, enter **“Workflow starting…”** for the **Message** property.



1. Now add an activity to update the field on the list item. Add a **SetField** activity (found in the **Utility Actions** section of the toolbox) to just after the **WriteToHistory** activity.
2. Select the **SetField** activity and using the **Property** tool window, update the following properties:
   * FieldName: **Body**
   * FieldValue: **“Updated by workflow”**

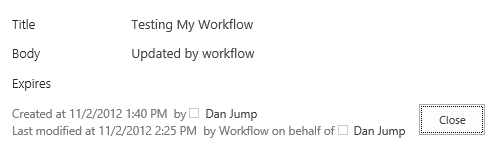
## Task 3: Deploy and Test the Workflow

In this task you will deploy and test the workflow as a sandbox solutions.

1. Since we are working with SharePoint Online, we are not yet able to debug workflows remotely. Instead, we will deploy our changes to test our work.
   1. Save the app by clicking **File ⮚ Save All**.
   2. Deploy the app in debugging mode by choosing **Build / Deploy.**
   3. Once the app has been deployed, launch a browser and navigate to the homepage of the test site.
   4. Click **Site Contents** and choose the **Simple Workflow App**.
2. Verify the **Announcements** list appears as a web part in the main page of our app. Do not add an item using this page. Instead, click the **Announcements** title to navigate to the **Announcements** list.
3. Add an item to the **Announcements** list but leave the **Body** field empty.

|  |
| --- |
| Description: C:\Users\vesaj\Pictures\DVD_ART36\Artwork_Imagery\Icons - Illustrations\_ SUPER VISTA STYLE\yield sign red white exclamation point.png **Important** |
| *At this point the new item may not appear in the Announcements web part. Simply click the title of the Announcements list to navigate to the Announcements list where the new item will appear.* |

1. After adding the item, start the workflow. Do this by selecting the item and clicking the **Workflows** button under the **View** tab in the ribbon.
2. On the **Announcements: Workflows: [item title]** page, click **SimpleWorkflow - WorkflowStart** to start the workflow.
3. After a brief delay the browser will redirect you to the Announcements list page again. The workflow will start but this may not be seen for a few seconds so be patient.
4. Click on the item in the list view to view the details page. You’ll notice that the **Body** field has been updated by the workflow.



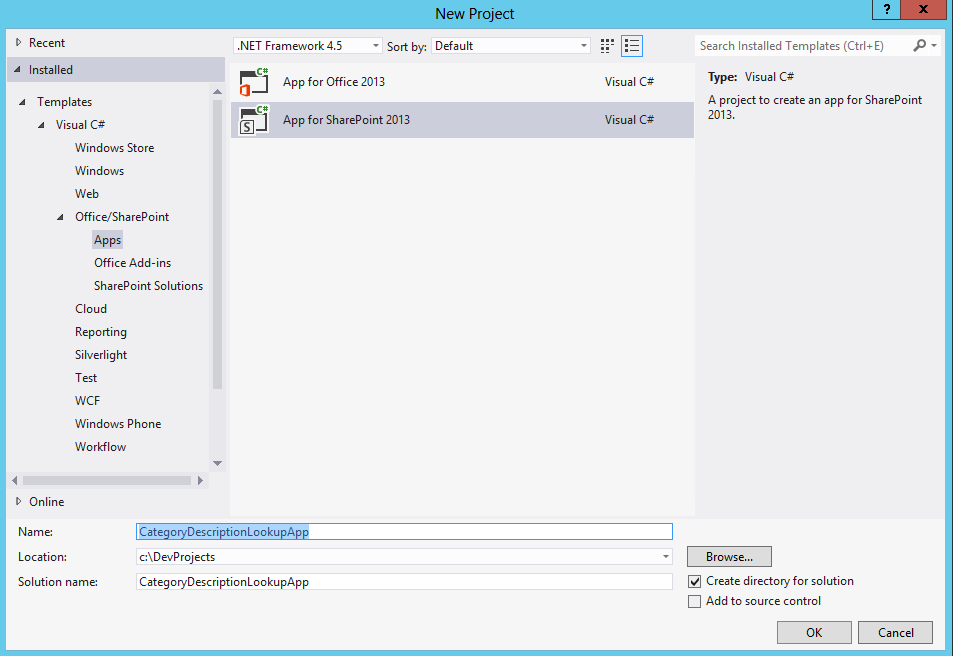
# Exercise 4: Using DynamicValues Visual Studio Workflows

* 1. In this exercise you will create a workflow using Visual Studio. This workflow will use the new HTTP and DynamicValue activities to issue a call from workflow app that gets a description of a specified category from a publically available OData service.

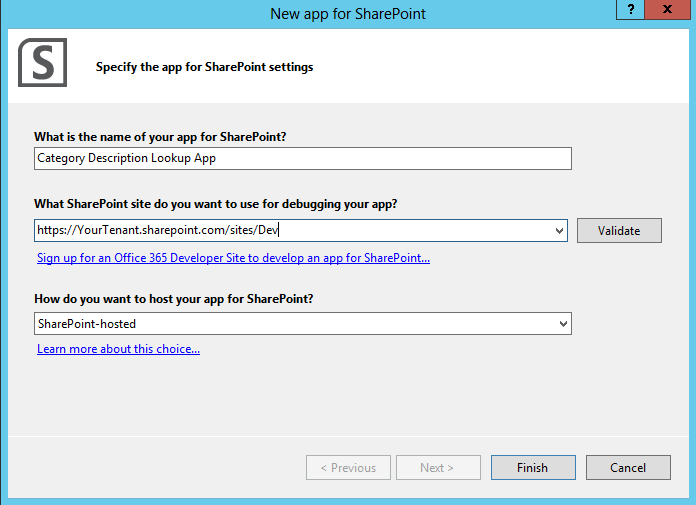
## Task 1: Create Workflow Project Variables

In this task you will create a workflow in Visual Studio and add some variables that will be used throughout the workflow.

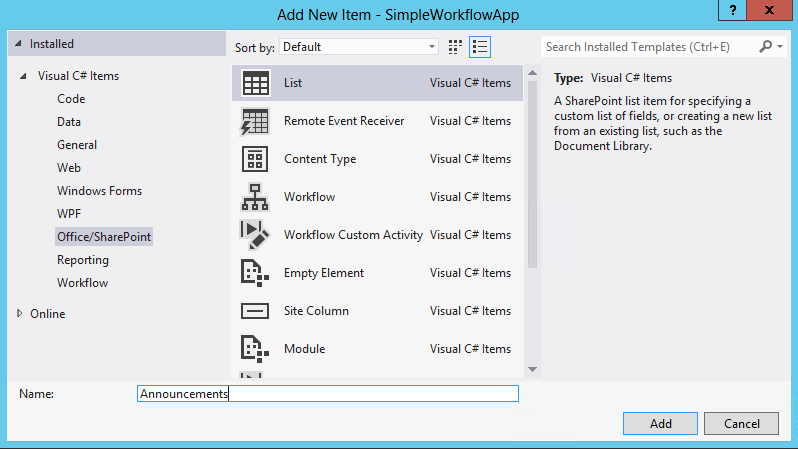
1. Open Visual Studio 2012 as an administrator: **Start ⮚ All Programs ⮚ Microsoft Visual Studio 2012 ⮚** right-click **Visual Studio 2012** and select **Run as Administrator**.
2. Select **File ⮚ New ⮚ Project**.
3. In the **New Project** dialog, select the **Visual C# ⮚ Office/SharePoint ⮚ Apps / Apps** category. Pick the project template **App for SharePoint 201**. Give the project the name **CategoryDescriptionLookupApp**.



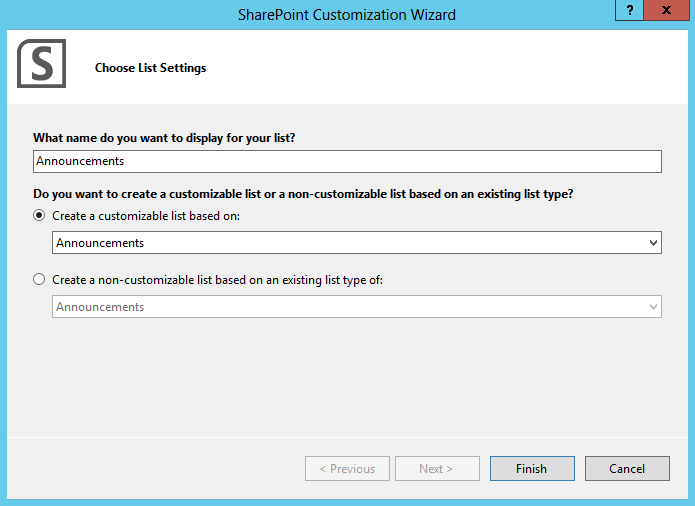
1. In the **New app for SharePoint** dialog, change the title to **Category** **Description Lookup App.** Enter the URL of the workflow site ([http://intranet.contoso.com/sites/workflow](http://intranet.contoso.com/sites/workflow/)) as the site to use to deploy the app, and choose the host type as **SharePoint-hosted**, finally clicking **OK**.



1. Once the project is created, add a new list to the project. Right-click the project in the **Solution Explorer** tool window and select **Add ⮚ New Item**:



1. In the Add New Item dialog, select **List.** Give it a name of **Announcements** and click **Add**.
2. In the Choose List Settings dialog, choose **Create a customizable list based on:** and choose the **Announcements** list template.



1. The Announcements list will be deployed to the app web, we need to provide a way for the user to navigate to it.
   1. Edit the **Default.aspx** page and add the following within the **PlaceHolderMain** content placeholder.

<WebPartPages:WebPartZone runat="server" ID="full"></WebPartPages:WebPartZone>

* 1. Under the Pages node you will find a file **elements.xml**. Edit that file, replacing the **File** element with the following:

<File Path="Pages\Default.aspx" Url="Pages/Default.aspx" >

<AllUsersWebPart WebPartZoneID="full" WebPartOrder="0">

<![CDATA[

<webParts>

<webPart xmlns="http://schemas.microsoft.com/WebPart/v3">

<metaData>

<type

name="Microsoft.SharePoint.WebPartPages.XsltListViewWebPart,

Microsoft.SharePoint,Version=14.0.0.0,Culture=neutral,

PublicKeyToken=71e9bce111e9429c" />

<importErrorMessage>

Cannot import this Web Part.

</importErrorMessage>

</metaData>

<data>

<properties>

<property name="Title"

type="string">Announcements</property>

<property name="ListDisplayName"

type="string">Announcements</property>

<property name="ChromeType"

type="chrometype">TitleOnly</property>

</properties>

</data>

</webPart>

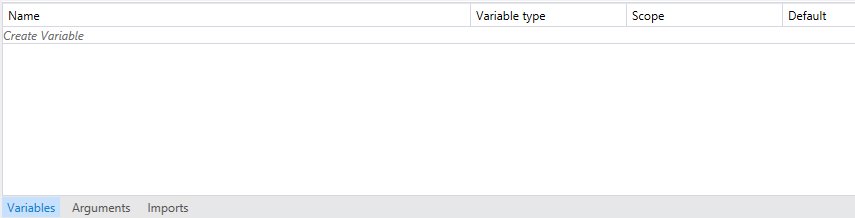
</webParts>

]]>

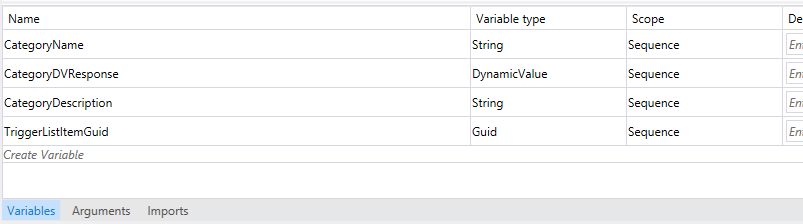
</AllUsersWebPart>

</File>

1. Add a new workflow item to the project. Right-click the project in the **Solution Explorer** tool window, select **Add ⮚ New Item**, and choose **Workflow**.
2. Add a new workflow item to the project. Right-click the project in the **Solution Explorer** tool window and select **Add ⮚ New Item**:
3. In the Add New Item dialog, select **Visual C# Items ⮚ Office/SharePoint** category and the template **Workflow**. Give it a name of **CategoryDescriptionLookupWorkflow** and click **Add**.
4. In the SharePoint Customization Wizard, select **List Workflow** and change the name to **Category Description Lookup Workflow**.
5. On the next step of the **SharePoint Customization Wizard**, check the box for **Would you like Visual Studio to automatically associate the workflow?** as well as the following values and click Finish.
   1. The library or list to associate… : **Announcements**
   2. The history list to display…: **Workflow History** or **<Create New>** if Workflow History isn’t an option.
   3. The task list to display…: **Workflow Tasks** or **<Create New>** if Workflow Tasks isn’t an option.
6. Visual Studio will automatically create and open in the designer the Workflow.xaml file that you will use to design the workflow. At the bottom of the designer you will find three tabs: Variables, Arguments & Imports. Click **Variables** to open the **Variables** window:



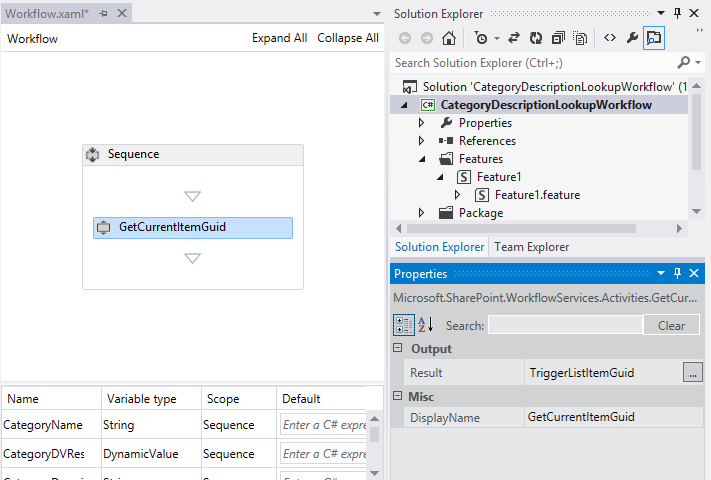
1. Once the **Variables** window it opens, click the **Sequence** activity in the designer to set the scope of the variables to the entire workflow.
2. Create three new variables that will be used throughout the workflow using the following options:
   1. Variable:
      * **Name:** CategoryName
      * **Variable Type:** String
      * **Scope:** Sequence
   2. Variable:
      * **Name:** CategoryDVResponse
      * **Variable Type:** Microsoft.Activities.DynamicValue
      * **Scope:** Sequence
   3. Variable:
      * **Name:** CategoryDescription
      * **Variable Type:** String
      * **Scope:** Sequence
   4. Variable:
      * **Name:** TriggerListItemGuid
      * **Variable Type:** System.Guid
      * **Scope:** Sequence



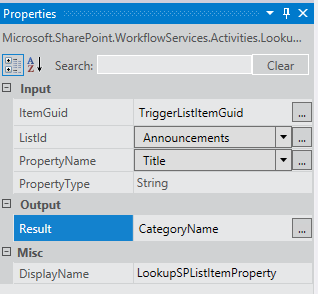
1. Save your changes by clicking **File ⮚ Save All**.

## Task 2: Add and Configure Activities to Request Data from an OData Service

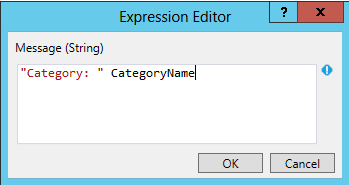
1. While still in the workflow created in the previous task, add a new **GetCurrentItemGuid** activity (found in the **Toolbox** under the **Utility Actions** category) to the workflow within the **Sequence** activity in the designer.
2. Select the **GetCurrentItemGuid** activity in the workflow, press **[F4]** to open the **Property** tool window and set the Result property to **TiggerListItemGuid**.



1. Next, add a new **LookupSPListItemProperty** activity (found in the **Toolbox** under the **SP – List Items** category) to the workflow within the **Sequence** activity in the designer.
2. Select the **LookupSPListItemProperty** activity and press **[F4]** to open the **Property** tool window. Set the following properties on the activity to extract the value in the **Title** field of the list item that triggered the workflow and store it in a local variable created in the previous task:
   1. **ItemGuid:** TriggerListItemGuid
   2. **ListId:** Announcements
   3. **PropertyName:** Title
   4. **Result:** CategoryName



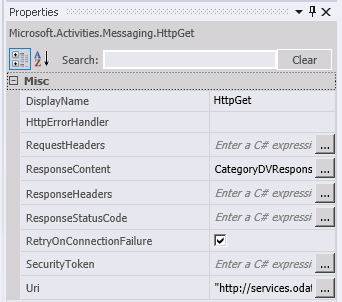
1. Add another activity (**WriteToHistory** found in the **Utility Actions** category in the **Toolbox**) to the workflow immediately following the previous activity added to write the value extracted from the item to the history list.
2. Select the **WriteToHistory** activity and press **[F4]**. Use the builder to create a string that writes out the name of the title field:



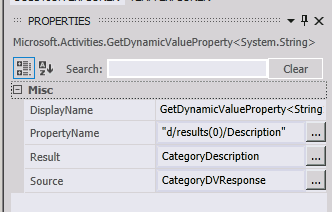
1. Save your changes by clicking **File ⮚ Save All**.

## Task 3: Call & Process the OData Service Response

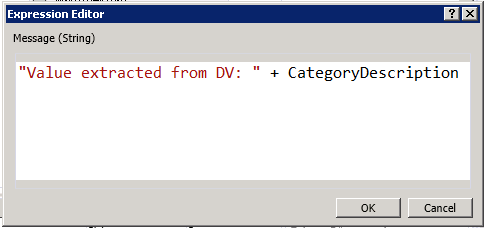
1. Now it is time to add an activity to the workflow that will make a call to the sample Northwind OData service hosted on <http://www.odata.org>. Add an **HttpSend** activity found in the **Messaging** category of the Toolbox.
2. Select the **HttpSend** activity and press **[F4]**. This activity has a few properties that you can ignore… many are used to authenticate against the service but in this exercise the service you will call is anonymous.
3. Store the results of the OData Web service call, which in the JSON format, to the **CategoryDVResult** DynamicValue variable created in a previous task by entering the variable name in the **ResponseContent** property.
4. Set the **Uri** property of the **HttpSend** activity to the following Web service address. You will have to craft the address by concatenating some strings together:
   * **Uri:** "http://services.odata.org/Northwind/Northwind.svc/Categories?$format=json&$filter=substringof('" + CategoryName + "', CategoryName) eq true&$select=Description"



1. Finally, for some additional debugging add another **WriteToHistory** activity immediately following the **HttpGet** activity and set its **Message** property to the following in order to see the raw results returned by the service in the workflow status page:
   1. **Message:** "Response from service: " + CategoryDVResponse.ToString()
2. With the service request stored in a variable, the next step is to extract it to another variable for use later in the workflow. From the **DynamicValue** category in the **Toolbox**, add a **GetDynamicValueProperty<T>** activity to the end of the workflow. When prompted for the type, select **String**.
3. Modify the properties of the **GetDynamicValueProperty<String>** activity to extract the **Description** of the category requested from the **CategoryDVResponse** (the Source) variable and store it in the **CategoryDescription** (Result) variable.



1. Add another activity (**WriteToHistory** found in the **SP – Current Context** category in the **Toolbox**) to the workflow immediately following the previous activity added to write the value extracted from the item to the history list.



1. Save your changes by clicking **File ⮚ Save All**.

## Task 4: Update SharePoint List Item

1. The last step in creating this workflow is to update the list item with the results from the OData service. From the **Utility Actions** category, add a **SetField** activity to the end of the workflow.
2. Set the properties of the **SetField** workflow to update the Body of the list item (an announcement) to the value extracted from the JSON OData response:
   1. **FieldName:** Body
   2. **FieldValue:** CategoryDescription
3. Save your changes by clicking **File ⮚ Save All**.

## Task 5: Deploy and Test the Workflow

1. Since we are working with SharePoint Online, we are not yet able to debug workflows remotely. Instead, we will deploy our changes to test our work.
   1. Save the app by clicking **File ⮚ Save All**.
   2. Deploy the app in debugging mode by choosing **Build / Deploy.**

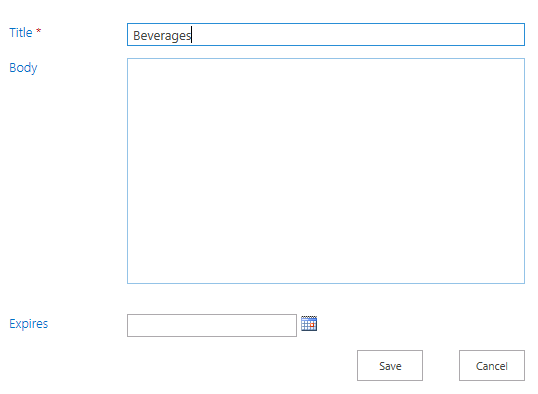
|  |
| --- |
| Description: C:\Users\vesaj\Pictures\DVD_ART36\Artwork_Imagery\Icons - Illustrations\_ SUPER VISTA STYLE\yield sign red white exclamation point.png **Important** |
| *If you are unable to deploy the app due to an error message “side loading of apps is not enabled on this site”, you can enable the Developer site collection feature using PowerShell.*  Enable-SPFeature -Identity e374875e-06b6-11e0-b0fa-57f5dfd72085 -Url http://intranet.contoso.com/sites/workflow |

* 1. Once the app has been deployed, launch a browser and navigate to the homepage of the test site.
  2. Click **Site Contents** and choose the **Category Description Lppkup App**.

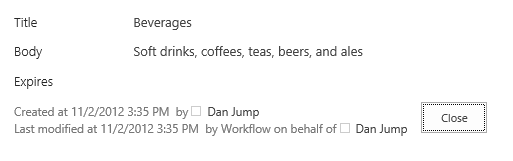
1. Verify the **Announcements** list appears as a web part in the main page of our app. Do not add an item using this page. Instead, click the **Announcements** title to navigate to the **Announcements** list.

|  |
| --- |
| Description: C:\Users\vesaj\Pictures\DVD_ART36\Artwork_Imagery\Icons - Illustrations\_ SUPER VISTA STYLE\yield sign red white exclamation point.png **Important** |
| *At this point the new item may not appear in the Announcements web part. Simply click the title of the Announcements list to navigate to the Announcements list where the new item will appear.* |

1. Add an item to the **Announcements** list:
   1. **Title**: Beverages
   2. **Body:** <blank>



1. After adding the item, start the workflow. Do this by selecting the item and clicking the **Workflows** button under the **View** tab in the ribbon.
2. On the **Announcements: Workflows: [item title]** page, click the **CategoryDescriptionLookupWorkflow - WorkflowStart** to start the workflow.
3. After a brief delay the browser will redirect you to the Announcements list page again. The workflow will start but this may not be seen for a few seconds so be patient.
4. Click on the item in the list view to view the details page. You’ll notice that the **Body** field has been updated by the workflow.



* This is the end of the lab

# Lab Summary

In this you performed following tasks.

* Created a new workflow using SharePoint Designer 2013.
* Created and worked with stages in SharePoint Designer 2013.
* Created and worked with workflows and DynamicValues in Visual Studio 2012