# SharePoint 2013 Demo – Workflow Stages

## Estimated time to complete this demo

20 minutes

## Objectives

Demonstration of following capabilities in SharePoint 2013

* Create a workflow that uses the new stages capability in SharePoint 2013

## Computers in this demo setup

This demo uses virtual machines as described in the following table. Before you begin the demo, 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}.

## Getting started with demo

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](http://intranet.contoso.com/sites/Workflow%20) .

In addition also ensure that a user **Dan Jump** (**CORP\danj**) has been added to the **Site Collection Administrators** of this site. You must create & run workflows under this user rather than the administrator account because workflows cannot be started using a SYSTEM account.

# Using Stages in SharePoint Designer 2013 Workflows

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.

* Begin this task logged on to **SP as CORP\danj**.

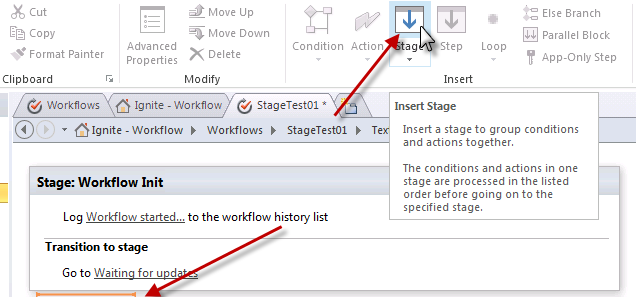
1. Open **SharePoint Designer 2013** (Start ⮚ All Programs ⮚ SharePoint ⮚ Microsoft SharePoint Designer 2013).
2. After SharePoint Designer 2013 launches, click the **Open Site** button. When prompted, enter the **URL** of the site to connect to: <http://intranet.contoso.com/sites/Workflow> and click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

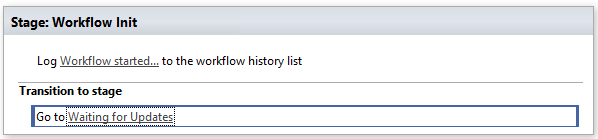
* **Name:** My First Staged Workflow
* **Platform Type:** SharePoint 2013 Workflow

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.
   * 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:



* + 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**.

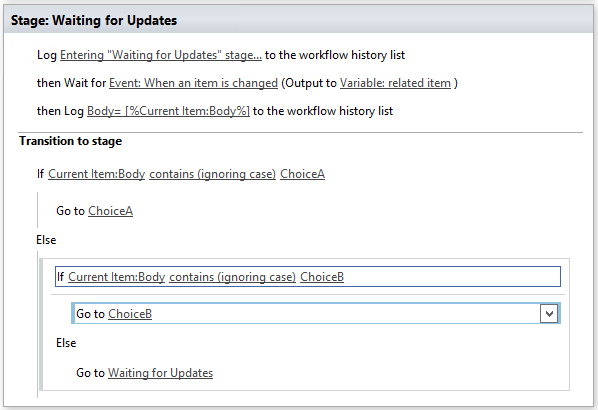


## Task 2: Complete the Waiting for Updates Stage

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

* Begin this task logged on to **SP as CORP\danj**.

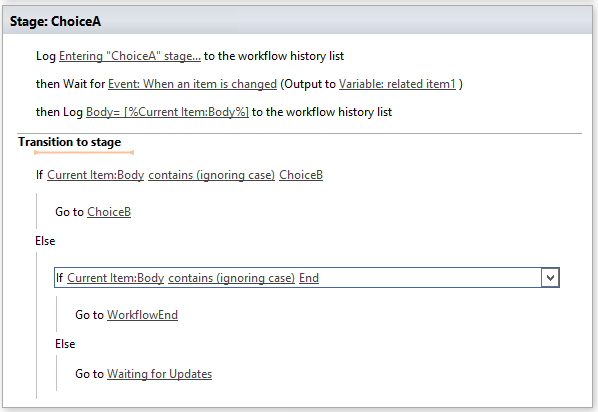
1. Within stage **Waiting for Updates**, add three actions in the following order:
   * Log to History List
   * Wait for Event in List Item
   * Log to History
2. For the first **Log to History List**, change the **message** to **Entering “Waiting for Updates” stage…**.
3. For the **Wait for Event in 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.
   * Add **Body =** to the text area
   * Click the **Add or Change Lookup** button
   * In the Lookup for String dialog, enter the following:
     1. Data Source: **Current Item**
     2. Field from Source: **Body**
     3. 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.

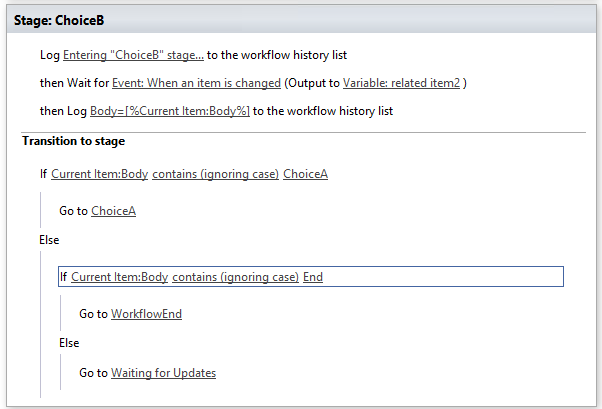
1. Within stage **ChoiceA**, add three actions in the following order:
   * Log to History List
   * Wait for Event in List Item
   * Log to History
2. For the first **Log to History List**, change the **message** to **Entering “ChoiceA” stage…**.
3. For the **Wait for Event in List Item**, click the **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.
   * Add **Body =** to the text area
   * Click the **Add or Change Lookup** button
   * In the Lookup for String dialog, enter the following:
     1. Data Source: **Current Item**
     2. Field from Source: **Body**
     3. 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.
   * Add **Body =** to the text area
   * Click the **Add or Change Lookup** button
   * In the Lookup for String dialog, enter the following:
     1. Data Source: **Current Item**
     2. Field from Source: **Body**
     3. 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.

* Begin this task logged on to **SP as CORP\danj**.

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

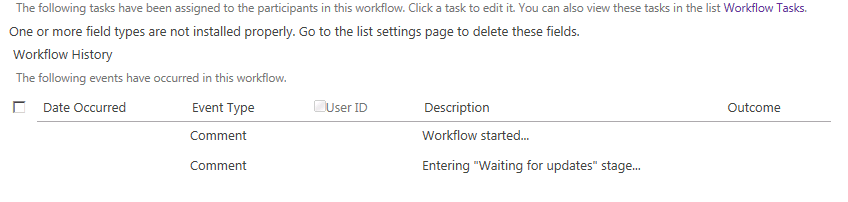
In this task you save, publish and test the workflow you just created.

* Begin this task logged on to **SP as CORP\danj**.

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.

* This is the end of the demo