knowledge14

April 27–May 1, 2014 • Moscone Center • San Francisco, CA

Lab Guide

Workflow 301:

Advanced Workflow for Application Development

Frankie Thompson & Michelle Corona

Lab instance: https://**<<name>>**.lab.service-now.com

User credentials: K14.WFAdmin/ k14wfadmin

Lab Goal

- Define the Stages of a Business process in a Set.
- Import Stage Sets into Workflows.
- Use the existing stages of a Workflow to create a Stage Set.

Lab 1
Workflow Stage Sets

Workflow Stage Sets in ServiceNow

The Workflow Engine uses Stages to reflect the progress of a workflow through notable aspects of a process.

In many cases the same set of stages can be used to define multiple processes or can be used as a starter set for similar processes. The Workflow Editor provides a tool to define and share Stage Sets.

Stage Sets are new in Dublin and provide an easy way to create a group of Stages that represent all or portions of a business process.

In previous versions of the product, lists of default Stages were associated with a table and would be auto assigned to Workflow Versions as soon as the Workflow Version was saved. Stage Sets provide all the ease of having default Stages associated with a table without imposing them on a Workflow Version by default.

Stage Sets offer far more flexibility than the previous default Stages. Even though they can be associated with a table; they are not required to be. In this way, Stage Sets can be written and shared across multiple workflows and tables that support similar processes.

©2014 ServiceNow, Inc. All rights reserved.

1. **Workflow > Stage Sets.**

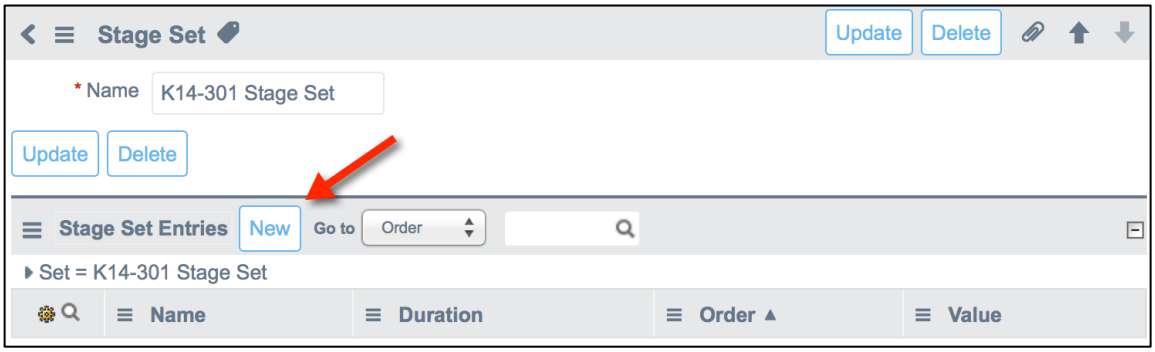
2. Select the **New** button.

3. Fill out the form as follows:
   Name: **K14-301 Stage Set**



4. Right-click and **Save.**

5. In the Stage Set Entries list, click **New**.



   On the form, notice the **Orde**r field. This field will auto-order in increments of 100 as the **Stage Set** is developed. The Order can be changed at any time. The auto-ordering provides a recommended order based on the order of creation.

6. Fill out the form as follows:
   Name: **Assignment**
   Value: **assign**

7. Click **Submit**.

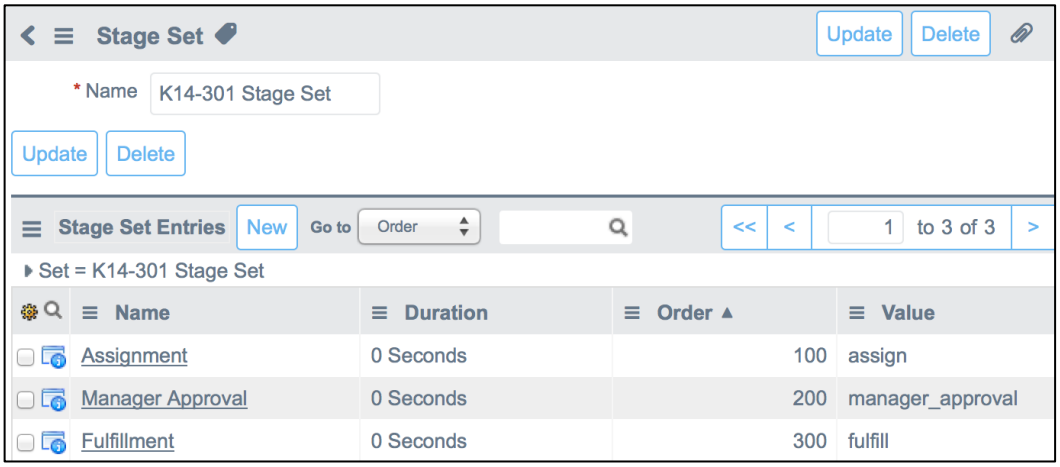
©2014 ServiceNow, Inc. All rights reserved.

8. Create another **Stage** as follows:
   Name: **Manager Approval**
   Value: **manager_approval**

9. Click **Submit**.

10. Create another **Stage** as follows:
    Name: **Fulfillment**
    Value: **fulfill**

11. Click **Submit**.



12. Click **Update** to save the K14-301 Stage Set.

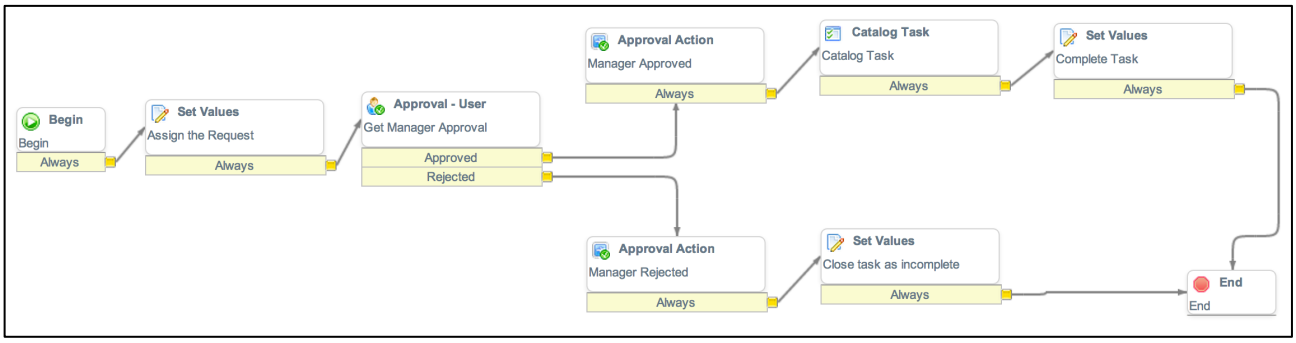
13. Navigate to **Workflow > Workflow Editor**.

14. Click **Open**.

15. Select **K14 WF 301 - Stages**.

©2014 ServiceNow, Inc. All rights reserved.

Your workflow should look like this:



16. Select the **Gear Menu > Check Out.**

17. Select the **Gear Menu > Edit Stages.**

Stage Sets Related Links

The Stage Sets Related Links are utilities for managing Stage Sets.

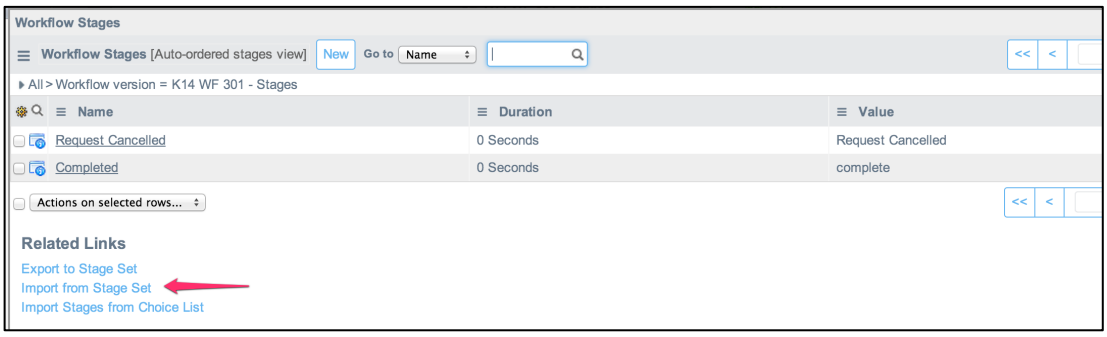
Export Stage Set: Will take all the Stages in Workflow Stages and export them to the Stage Set list. Once exported, the Stage Set can be imported into other workflows. This is particularly useful if there are multiple workflows that participate in the same process. Stages in one workflow can be exported and shared with peer workflows or subflows.

Import from Stage Set: Will take an established Stage Set, for example the K14-301 Stage Set we just built, and import the stages into the Workflow Stages list of the current Workflow Version.

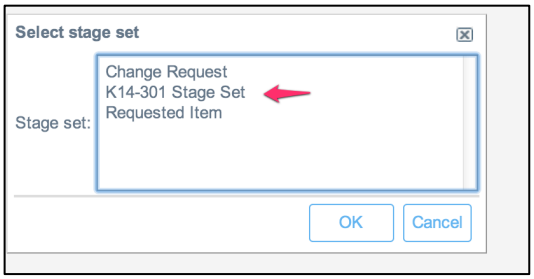
Import Stages from Choice List: Current records have columns that are made up of choices. These choices often represent stages of a process. This link facilitates translating Choice Lists into Workflow Stages. This reduces duplicate entry, eliminates mistakes in entry, and provides a fluid connection between the current record and the Workflow Stages.

©2014 ServiceNow, Inc. All rights reserved.

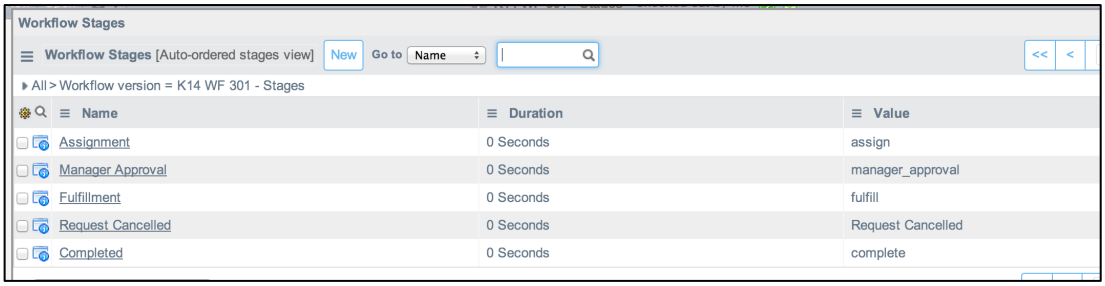
18. Select **Import from Stage Set**.



19. Select the **K14-301 Stage Set**.



20. Click **OK**.

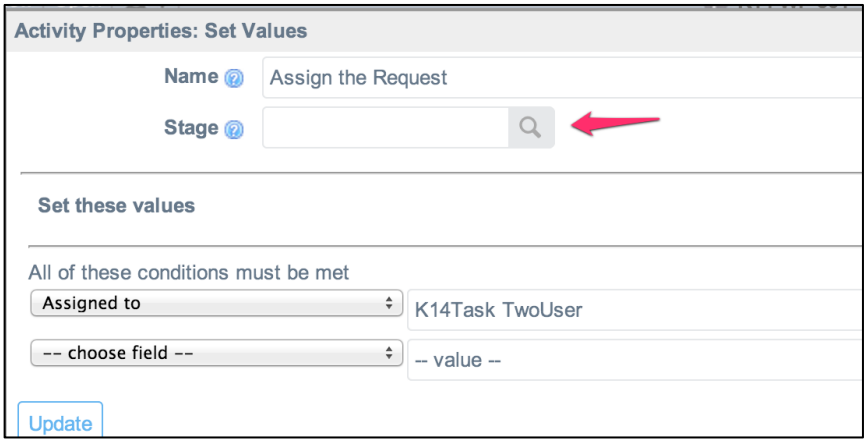


21. Close the Workflow Stages list.

©2014 ServiceNow, Inc. All rights reserved.

Assign Stage to Activity

1. Double-click the **Assign the Request Set Values** Activity.

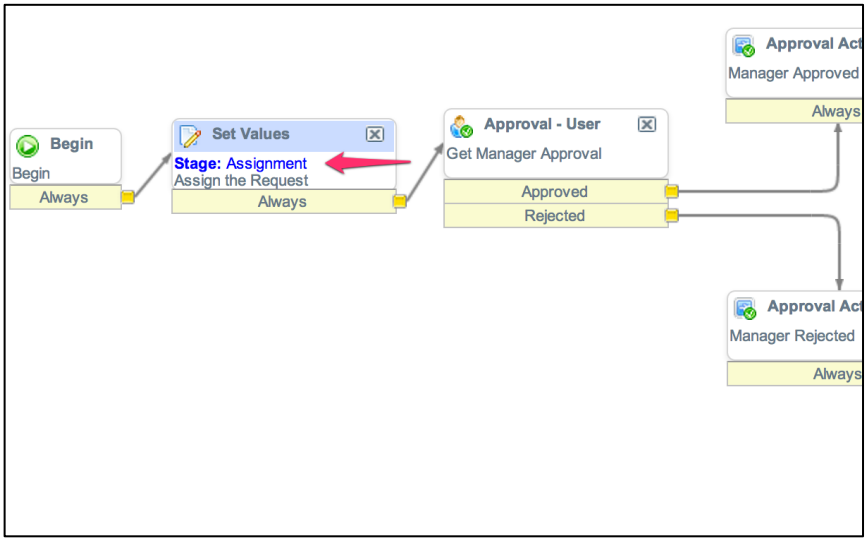


2. Select the Stage field **Looking Glass** search icon**.**

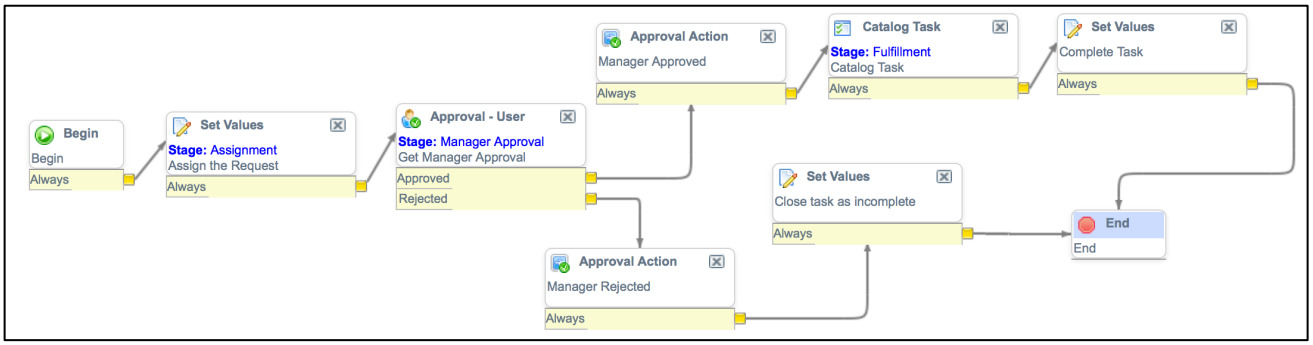
3. Select **Assignment**.

4. Click **Update**.

®2014 ServiceNow, Inc. All rights reserved.



5. Assign Stages to these Activities:
   To the Get Manager Approval, assign the **Manager Approval** stage
   To the Catalog Task Activity, assign the **Fulfillment** stage



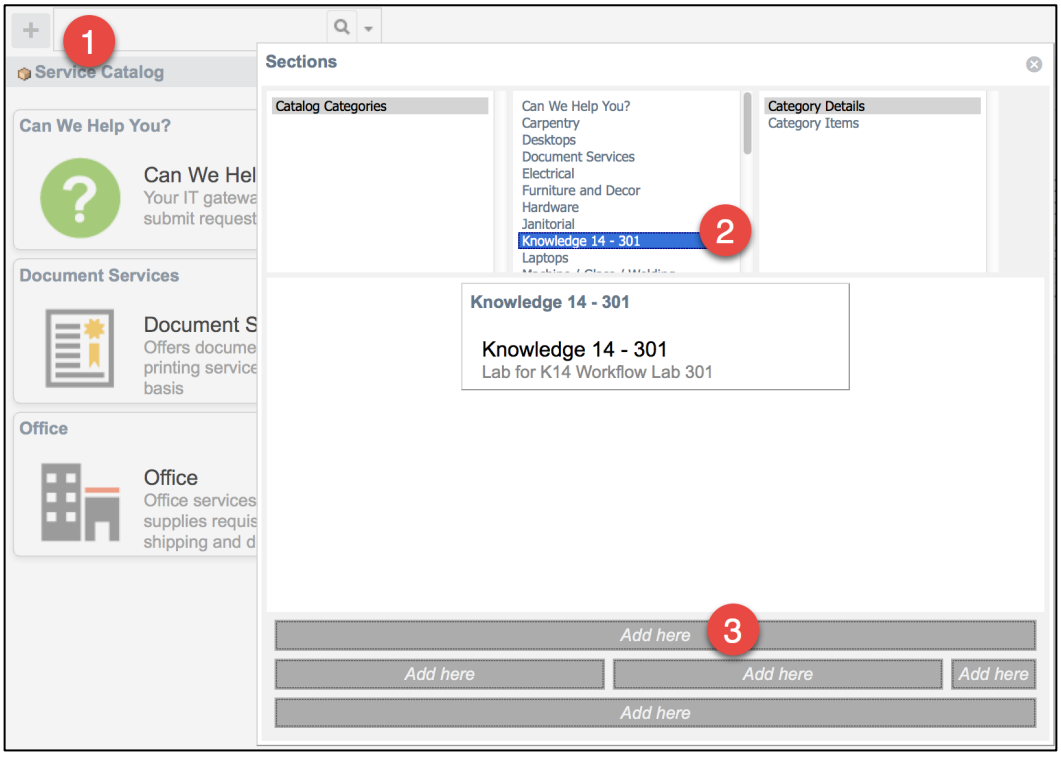
6. Return to the main ServiceNow tab.

©2014 ServiceNow, Inc. All rights reserved.

Make Service Catalog Request to Review Stages

1. **Self-Service > Service Catalog**.

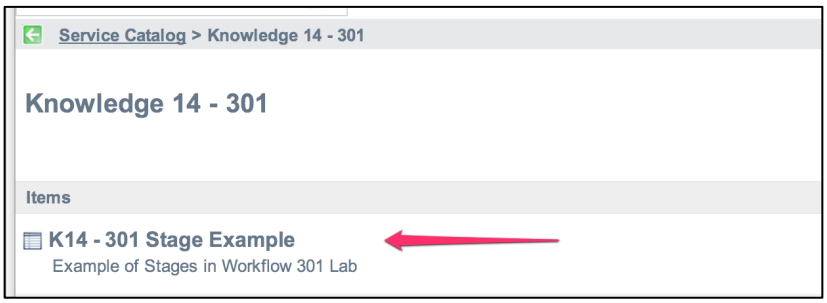
2. Add the **Knowledge 14** Category.
   Step 1. Click **+**.
   Step 2. Select **Knowledge 14 – 301**.
   Step 3. Add the **Knowledge 14** Category to the top dropzone.
   Step 4. Close the **Add menu**.



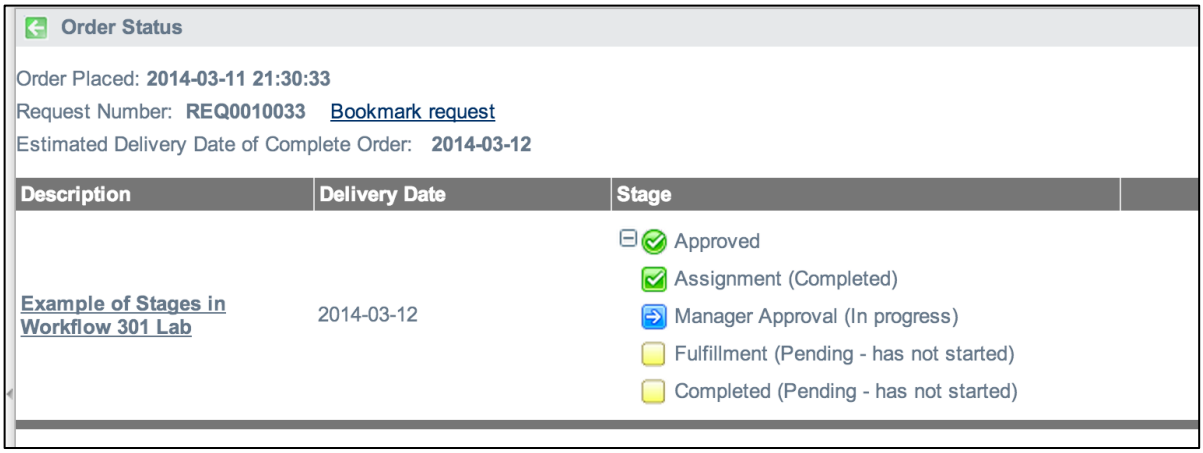
3. Select the **Knowledge 14 - 301** Category.

©2014 ServiceNow, Inc. All rights reserved.

4. In the Items list, select **K14 – 301 Stage Example**.



5. Click **Order Now.**

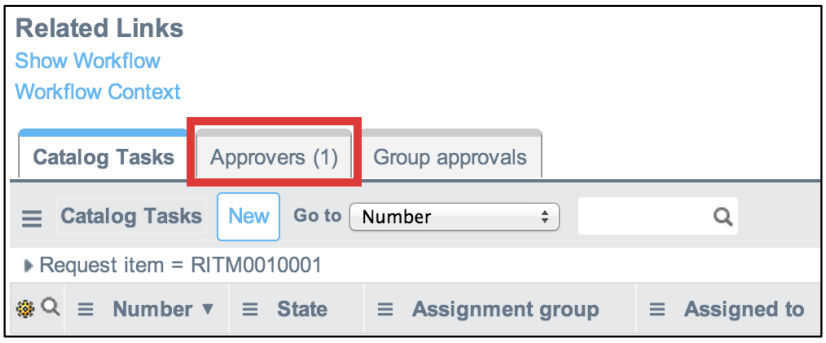


6. In the Description column, click the **Example of Stages in Workflow 301 Lab RITM** link.

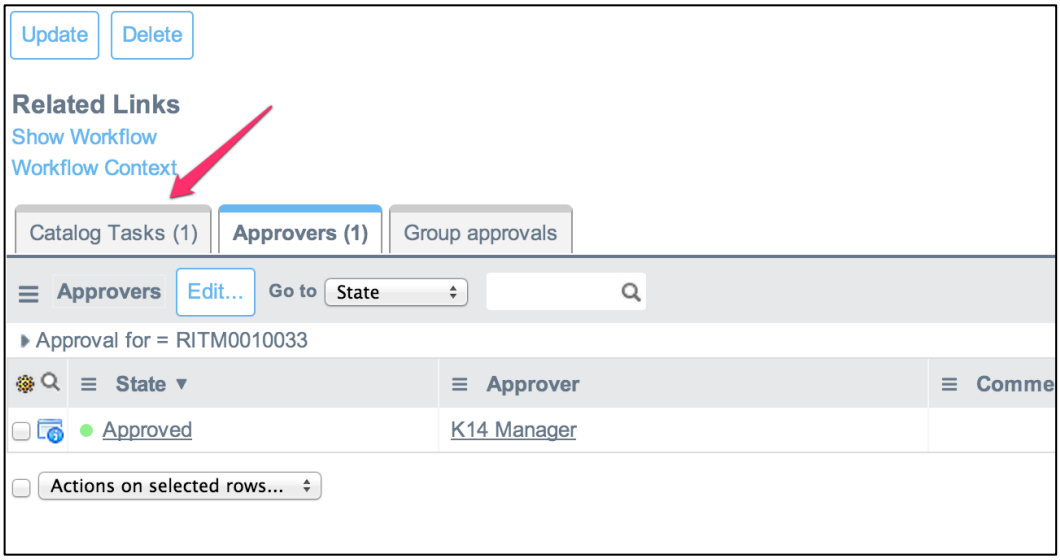
7. Change to the **Catalog** View. You can do this by clicking on the View name. Also, click on the Tabs icon for easier access to the Approval record.

©2014 ServiceNow, Inc. All rights reserved.

8. Select the **Approvers** Related List.



9. Open the Approval record, **Approve** the request, then **Update** the record. Upon updating, a Catalog Task record is generated.



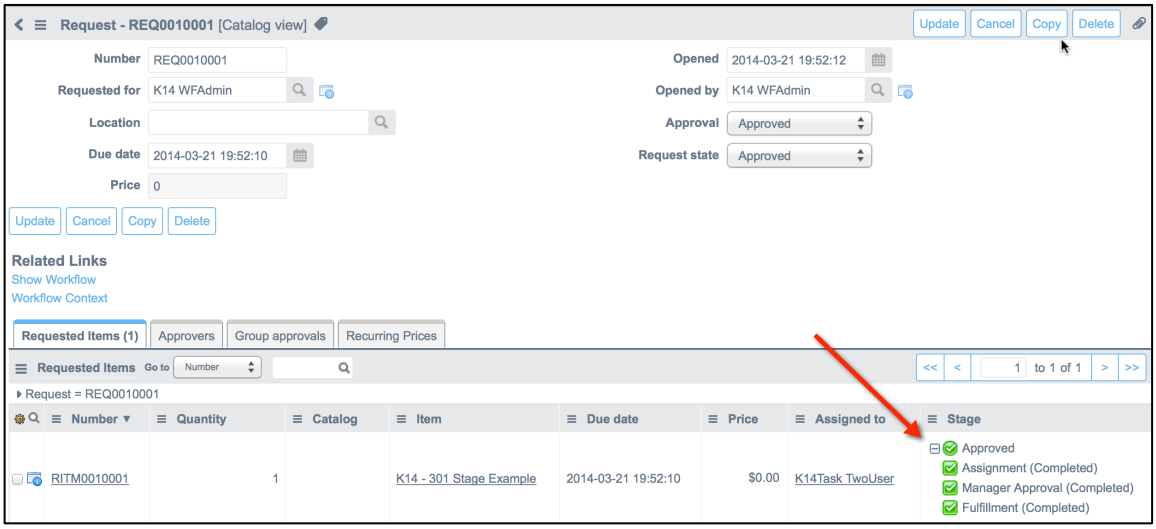
10. Open the Catalog Task record, set the State to **Closed Complete**, and **Update** the record.

11. Scroll to the top of the form and locate the **Request**.



©2014 ServiceNow, Inc. All rights reserved.

12. Click the form icon next to the Request. The Stage icons reveal the progress of the business process. Use the **Show Workflow** Related Link to see the completed context.



Summary

For processes that share significant points in their processing, incorporating Stage Sets into their workflow provides a way to recognize, define, and share a common set of Stages.

Stage Sets are imported into a Workflow Version using the Related Link Import from Stage Sets on the Workflow Stages list accessed through the **Workflow > Edit Stages** control of the Workflow Gear menu.

Stages that are added directly to the Workflow Stage List of a workflow as part of the design of a workflow may be exported for sharing using the Related Link Export to Stage Set on the Workflow Stages list accessed through the Workflow > Edit Stages menu option of the Workflow Gear menu.

Often times the table associated with a workflow will have a state or stage column that already defines a list of states. In this case, those pre-defined states can be added to the Workflow as a Stage set using the Related Link Import Stages from Choice List on the Workflow Stages list accessed through the Workflow > Edit Stages menu option of the Workflow Gear menu.

©2014 ServiceNow, Inc. All rights reserved.

Lab Goal

- Learn how to add a custom stage column to a table.
- Learn how to have a workflow write stages to that column.

Lab 2 Custom Stage Column

Adding a Custom Stage Column

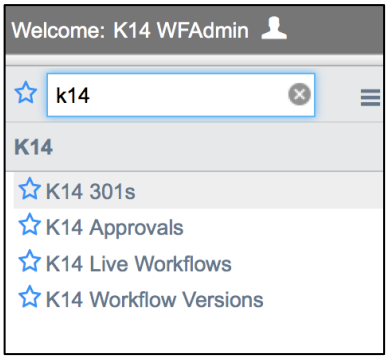
When designing applications, using the Stage column is an excellent way to reflect process progress to a user while abstracting away from the complexity of the Workflow.

The Element type of **Workflow**, when associated with a **Workflow Version** will receive the stage state as the workflow executes.

For this lab, a custom application named **K14** has been added to your instance.

1. Return to the main ServiceNow tab.

2. In the Application Navigator Text filter type, **K14.**

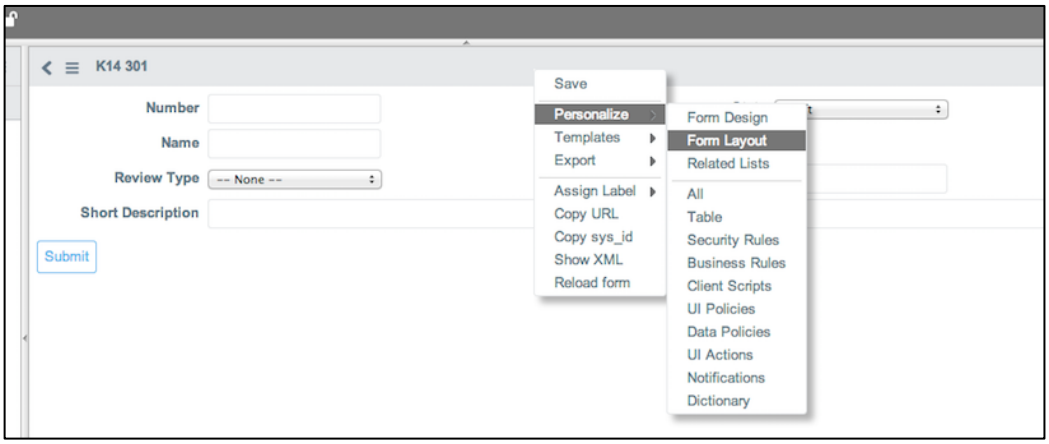


3. Select the **K14 301s** link.

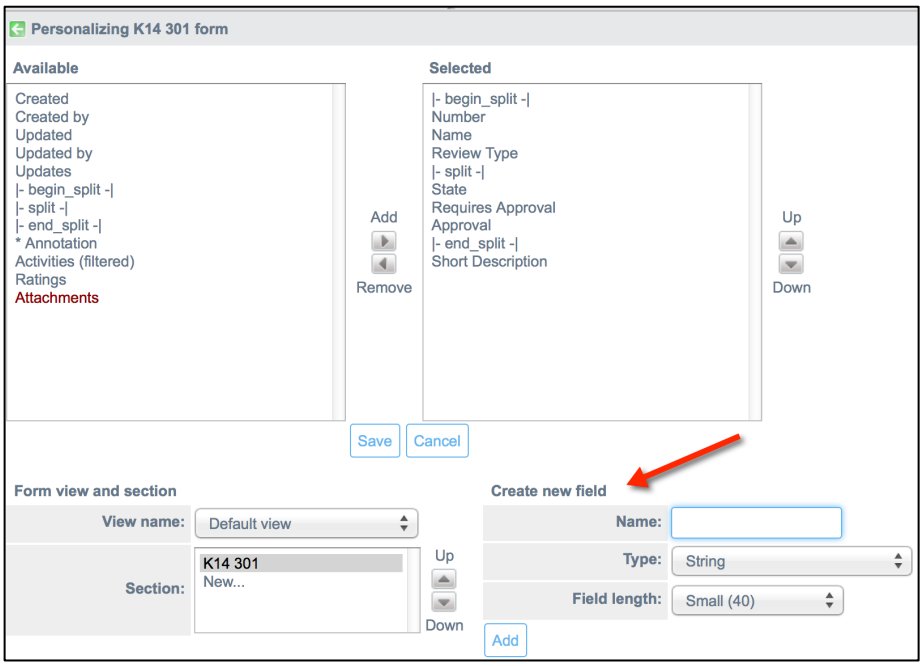
©2014 ServiceNow, Inc. All rights reserved.

4. Select the **New** button.

5. In the header of the form, right-click and **Personalize > Form Layout.**



6. Find the **Create new field** section on the form.



®2014 ServiceNow, Inc. All rights reserved.

7. Fill out the **Create new field** section as follows:
   Name: **Stage**
   Type: **Workflow**

8. Click the **Add** button.

9. Use the Up Arrow to move the new **Stage** field just under **State**.

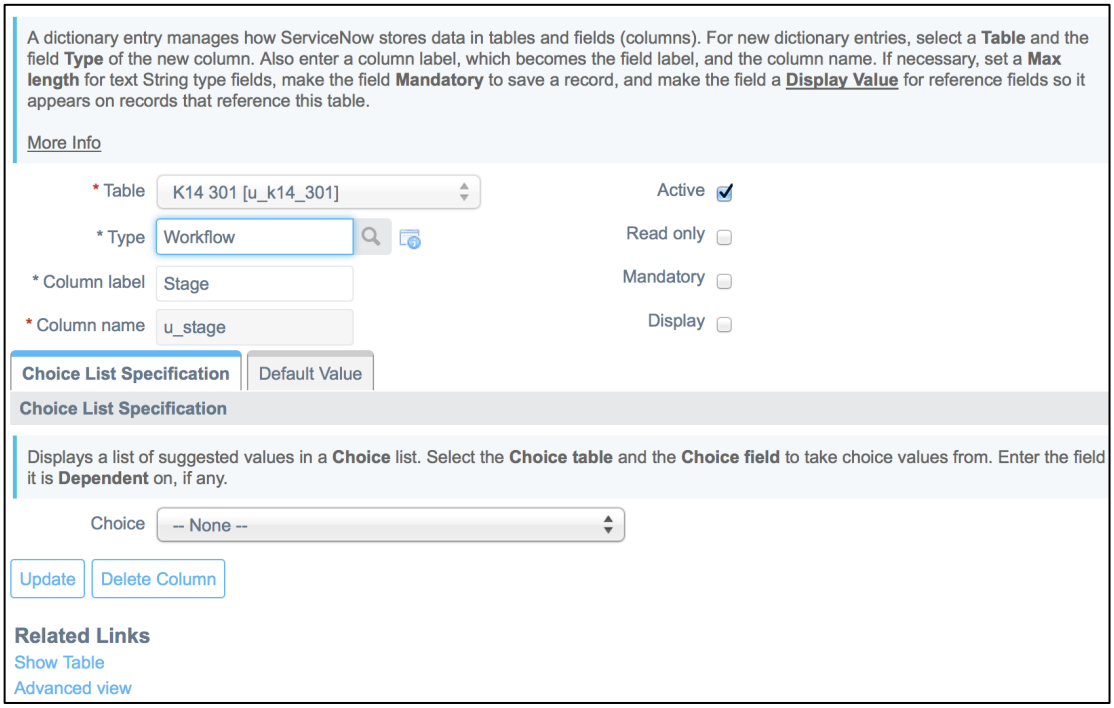
10. Click the **Save** button.



11. Right-click on the **Stage** label.

12. Select **Personalize Dictionary.**

®2014 ServiceNow, Inc. All rights reserved.



NOTE: When adding a Stage column to a table in an application that is specific to rendering the Stage progression icons, it is best practice to make that field Read only. In this way, conflicts between the workflow process and what the user sees are not possible. The Workflow Engine *writes* to the Stage field, it does not *read* or *react* to it.

13. Check **Read only**.

14. Click **Update**.

Now that there is a Workflow field that can accept Stages from a workflow, configure a workflow to write to that field.

©2014 ServiceNow, Inc. All rights reserved.

Associate Stage Column to Stages

15. Return to the Workflow Editor tab.

16. Click **Open.**

17. Select **K14 – 301 Custom Stage Demo**.

18. Select **Gear Menu > Check out**.

19. Select **Gear Menu > Properties**.



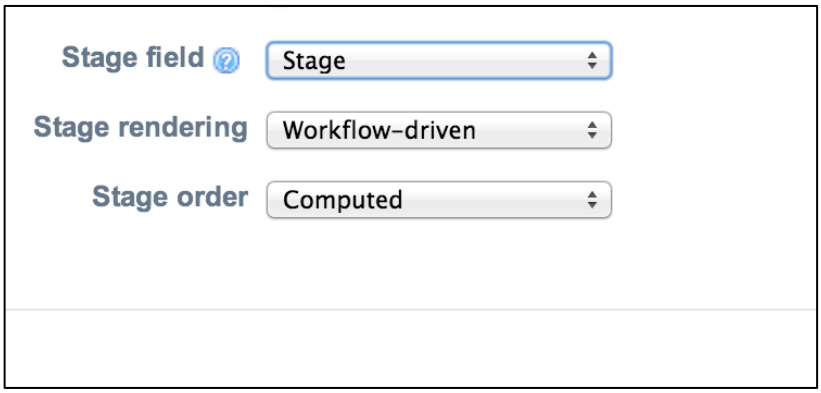
Notice:

- This workflow is written against the K14_301 table just modified.
- The workflow runs only when the **Name** starts with **Custom Stage.**
- The **Stage** field is the field that tells the Workflow Engine to write the current Stage value to the current record. It is possible to have Stages in a workflow and not have them communicate to the current record. They can be on the workflow just for clarity when viewing the Workflow on the Canvas. However, if the designer intends to reflect the progress of the workflow in **Stage** icons, this field must be selected.

©2014 ServiceNow, Inc. All rights reserved.

20. Select the **Stage** column from the choice list.

21. The stage section of your screen should look like this:



22. Notice the two additional Stage Fields. These are fields that will determine how the Workflow Engine follows the stages through the workflow execution and how it reports them.

©2014 ServiceNow, Inc. All rights reserved.

Stage Rendering

The Workflow Engine determines how to render the stages in the icon list based on one of these selections.

Legacy: This is the least desired selection. The legacy rendering will continue to support all the icon script includes and Business Rules that are already deployed in previous versions of the product. This is specifically for backwards compatibility.

Linear: The Workflow Engine will render all the icons in the Workflow Stage list, in the order that is assigned to the individual Stages. With the Linear renderer, the Workflow Engine will maintain the Stage state even if the Stages are changing in subflows. With this selection, the designer has the most control over how the Stages will appear in the list. Every Stage will appear, regardless of whether or not the stage was executed in the Workflow.

Main flow: The Workflow Engine will render only the icons that represent Stages executed in the Main flow. If there are Stages in subflows they will not reflect in this icon list. If there are stages on execution paths not executed in a workflow, they will not show in the stage icon list.

Progress Bar: The Workflow Engine will not render icons as all. It will render a single progress bar that shows the overall progress of the Workflow.

Workflow-Driven: This is the default selection. With this selection the Workflow Engine will initially render icons that represent Stages that are on the 'happy path' of execution. That is the Stages on **Approved**, **Yes**, **Always**, and **Complete** condition paths. However, if during execution an "unhappy path" begins (for example an Approval Rejection), the Workflow Engine will re-calculate and redraw the icon tree. In the adjusted rendering, the "happy path" no longer displays and the Workflow Engine attempts to predict the next likely path the workflow could transition towards.

Stage Order

The Workflow Engine will determine the order in which stages will appear in the icon list based on one of these selections.

Computed: Icons by default are ordered in the order of execution of a workflow and based on the expected or surmised future path found in all transitions. This default ordering is called the **Computed** order and is defined in the workflow properties under the field Stage order.

User Specified: The workflow designer may determine the exact order of icon rendering in a list, regardless of execution order by selecting **User Specified** in the Stage order: field of workflow properties and appropriately adding an order value to the defined Workflow Stages

©2014 ServiceNow, Inc. All rights reserved.

For this lab, we will choose Linear, since we have the most control over what we see.

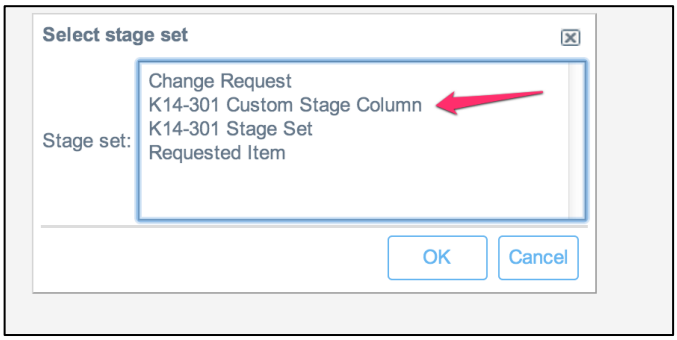
1. Select Stage Rendering: **Linear**.

2. Stage Order: **User Specified.**

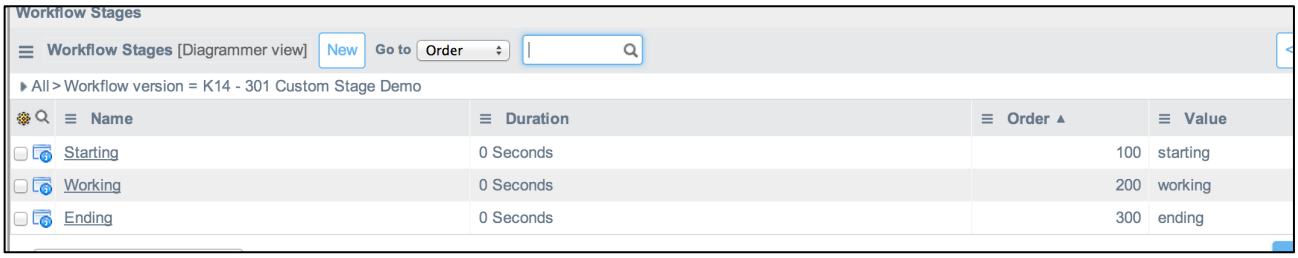
3. Click **Update.**

4. Select **Gear Menu > Edit Stages.**

5. Select **Import from Stage Set**.



6. Select **K14-301 Custom Stage Column** .



7. Close the Workflow Stages list.

8. To the **Custom Stage Log Starting** Log Message Activity, assign the **Starting** stage.

©2014 ServiceNow, Inc. All rights reserved.

9. To the **Waiting** Timer Activity, assign the **Working** stage.

10. To **Log Ending** Log Message Activity, assign the **Ending** stage.

11. Return to the main ServiceNow tab.

12. In the Application Navigator Text filter type, **K14**.

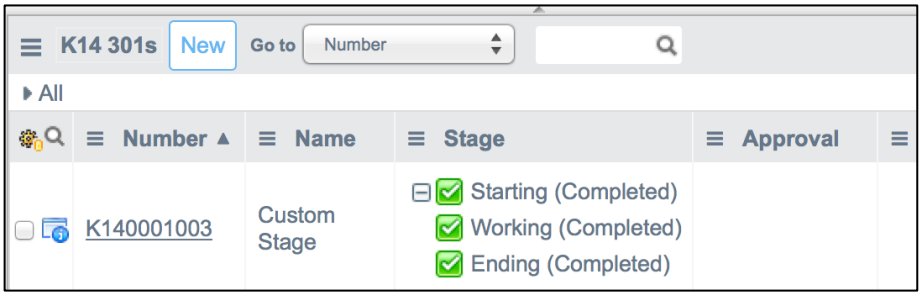
13. Select the **K14 301s** link.

14. Click **New**.

15. Fill out the form as follows:
    Name: **Custom Stage**

16. Click **Submit.**

17. Personalize the List Layout and add the **Stage** column after the Name field.



If it does not look like that right away, refresh you list. It could just be your timer has not come back and completed the stage yet.

18. Return to the Workflow Editor.

©2014 ServiceNow, Inc. All rights reserved.

19. **Gear Menu > Edit Stages** .

20. In the **Working** Stage, change the value in the Order column to **600**.

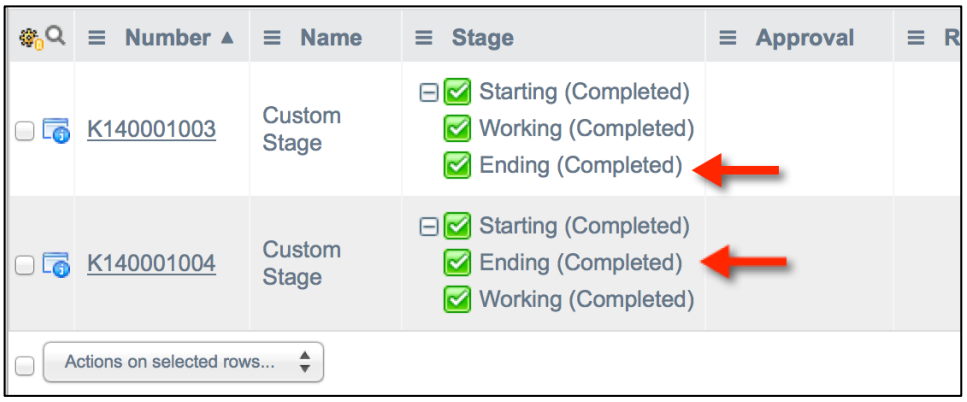
21. Close the Stage List.

22. Return to the **K14** application. You can see the difference in progression in the workflow by submitting additional records.

23. Click **New**.

24. Fill out the form as follows:
    Name: **Custom Stage**

25. Click **Submit.**



Notice that the order of the icons has changed according to the user ordering specified in the Workflow Stage List. Ordering is one way the designer has to convey the status of a workflow. It is also possible to change the rendering format completely.

26. Return to the Workflow Editor.

©2014 ServiceNow, Inc. All rights reserved.

27. **Gear Menu > Publish.**

28. **Gear Menu > Check out.**

29. **Gear Menu > Properties.**

30. In the Stage Rendering field select **SimpleProgressBar.**

31. Click **Update**.

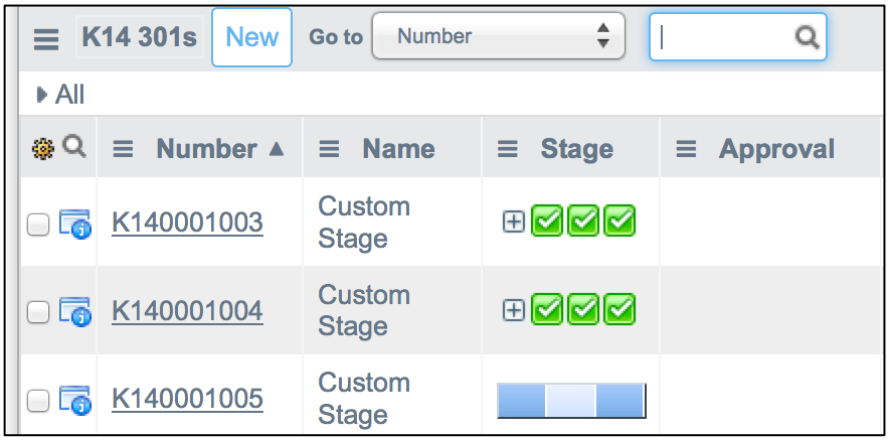
32. Return to the **K14** application. You can see the difference in progression in the workflow by submitting additional records.

33. Click **New**.

34. Fill out the form as follows:
    Name: **Custom Stage**

35. Click **Submit.**

    Your list should look like this:



The icons and the progress bar are the currently available base instance rendering options for displaying the progress of a workflow.

©2014 ServiceNow, Inc. All rights reserved.

Summary

Workflow Stage icons are a powerful way to reflect the progress of a workflow to users.

The Workflow Engine can be configured to reflect icons in a variety of ways:

- In the order the process designer specifies
- In the order the Workflow Engine discovers stages as it executes
- Stages only in the main flow with multiple subflows
- As a progress bar that reflects overall progress.

The order and rendering of Stage icons is configured in the Workflow Properties form.

The Workflow Engine will publish icons if:

- A table has a field of Workflow Type.
- A Workflow is written against that table.
- The Stage column has been identified in the Workflow Properties.
- Stages have been assigned to activities in a Workflow.

©2014 ServiceNow, Inc. All rights reserved.

Lab Goal

Learn how to share stages sets for workflows in a shared process.

Lab 3
Stage Sets for Applications

Stage Sets for Applications

Recall that the purpose of Stage icons is to place the user in the progression of a business process. The user does not need to be aware of where they are in a workflow execution – being in a subflow or the main workflow is not really relevant to them.

Stage Sets elevate the notable aspects of a business process to the user and communicate using icons. As a designer involved in service automation, you will know the meaningful business process stages independent of the design of the workflow. The value of designing Stage Sets to identify the stages then apply them to all workflows involved in a process.

This lab provides a simple example of sharing a stage set across a main flow that uses a subflow within a single business process.

When designing applications it is typical to put common re-usable behavior in subflows. As was learned in Lab 1, the significant elements of a business process may be defined in a single stage set.

For this lab, a simple main flow and subflow have been added to the **K14** application. has been added to your instance.

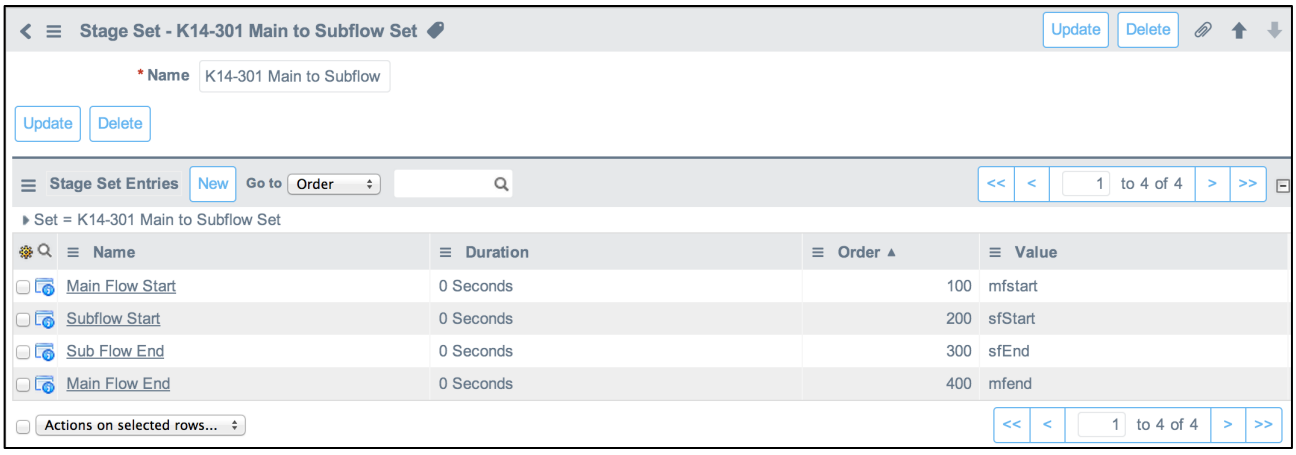
1. Return to the main ServiceNow tab.

2. In the Application Navigator Text filter type, **Stage Sets.**

3. Select **K14-301 Main to Subflow Set Stage**.

©2014 ServiceNow, Inc. All rights reserved.

4. Notice there are four elements in this stage set. Notice the order of the stages. In the example we will use Linear Renderer.



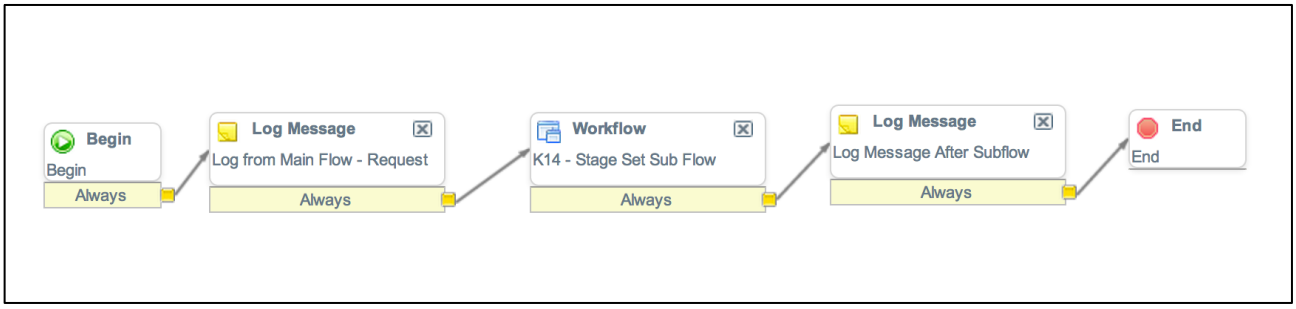
QUESTION: What is relevance of the Order column when using the Linear Renderer?

5. Return to the main Workflow Editor.

6. Select **Open**.

7. Open **K14 – Stage Set Main Flow**.

Your workflow should look like this:



©2014 ServiceNow, Inc. All rights reserved.

8. Select the **Gear Menu > Check Out**.

9. Select the **Gear Menu > Properties**.



QUESTION: Looking at the properties, what is the Condition start workflow execution?

QUESTION: What property needs to be updated so that the workflow will update the Stages icons?

QUESTION: What does User Specified mean in relation to the Linear Stage Renderer?

10. In the Stage field, assign the **Stage** column if not already assigned.

11. Close the Workflow Properties form.

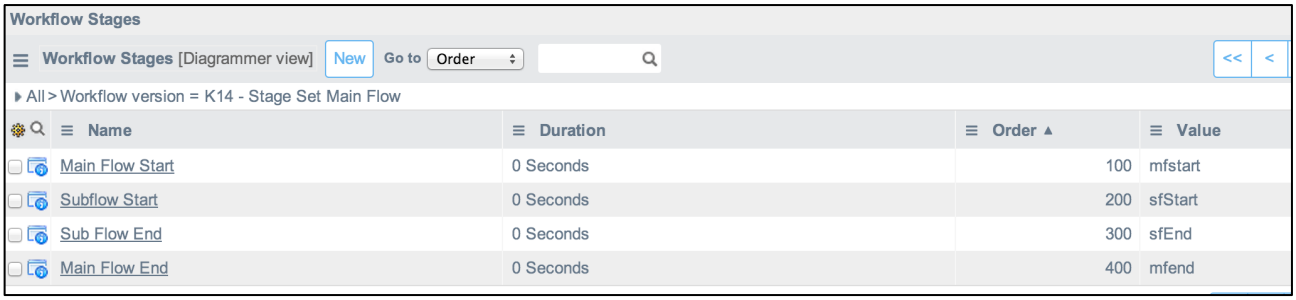
12. Select the **Gear Menu > Edit Stages.**

13. Select **Import from Stage Set.**

©2014 ServiceNow, Inc. All rights reserved.

14. Select **K14-301 Main to Subflow Set.**

    Your Workflow Stages should look like this:

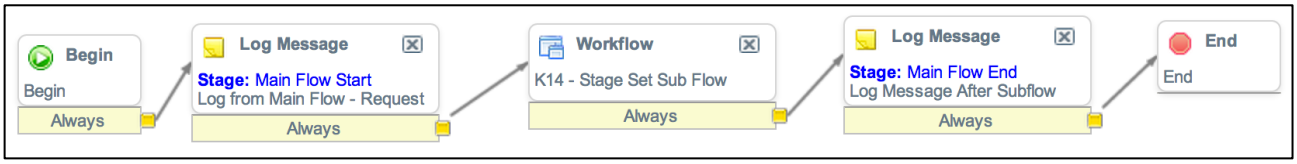


15. Close the Workflow Stages list.

16. To the **Log from Main Flow - Request** Log Message Activity, assign the **Main Flow Start** stage.

17. To the **Log Message After Subflow** Log Message Activity, assign the **Main Flow End** stage. Note that the Stage displays on the Activity.

    Notice in the workflow there is a subflow right in the center. We will check out that subflow next and add the same Stage Set, but assign it different Stages of the Stage Set to subflow activities.



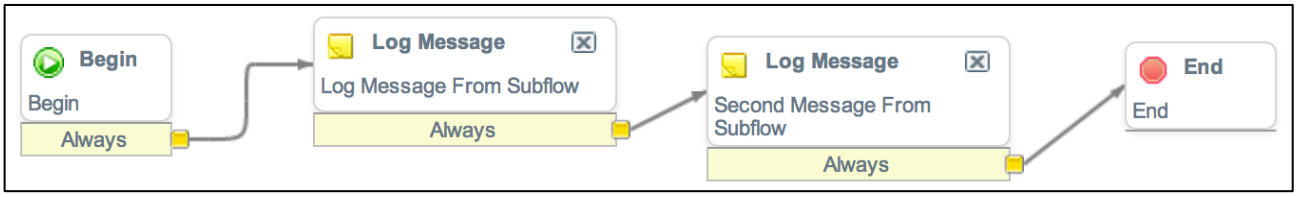
18. Select the **Gear Menu > Publish.**

19. Select **Open**.

20. Open **K14 – Stage Set Sub Flow**.

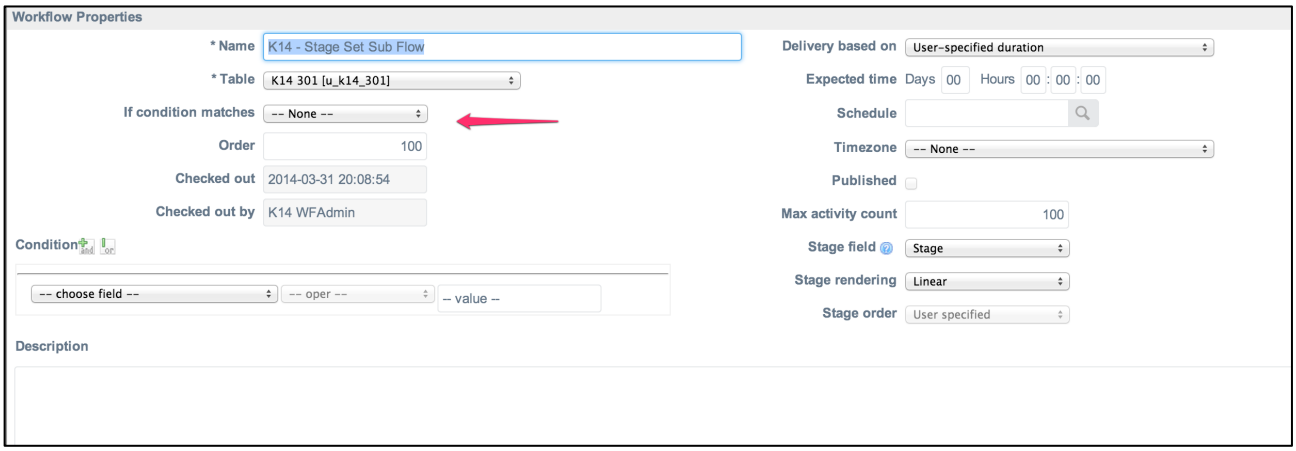
©2014 ServiceNow, Inc. All rights reserved.

21. Your workflow should look like this:



22. Select the **Gear Menu > Check Out**.

23. Select the **Gear Menu > Properties**.



Notice the **If Condition Matches** property is set to **None**.

QUESTION: Why do you think this Condition is set to None?

When None is the selection the Workflow Engine will not run the workflow. Since this is a subflow, that is perfect. In this scenario, it will only run when called from the main workflow.

QUESTION: If we want this workflow to report stages, what do we need to make sure is defined in the Workflow Properties?

24. Select the **Stage** column, if not set in the Stage field.

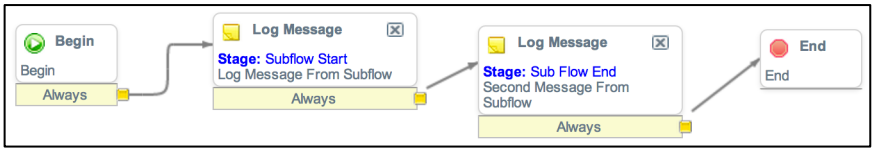
©2014 ServiceNow, Inc. All rights reserved.

25. Click **Update**.

26. Import the **K14-301 Main to Subflow Set** Stage Set into the subflow.

27. To the Log from Subflow Flow Log Message Activity, assign the **Sub Flow Start** Stage.

28. To the Second Message from Subflow Log Message Activity, assign the **Sub Flow End** Stage.



29. Select the **Gear Menu > Publish.**

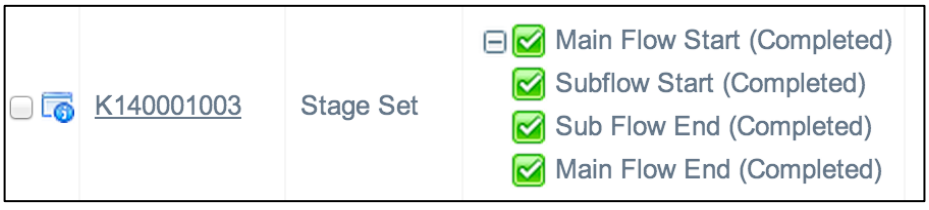
30. Return to the Main ServiceNow tab and navigate to **K14 301s**.

31. Select the **New** button.

32. Fill out the form as follows:
    Name: **Stage Set**

33. Click **Submit**.

    Your Stage Set icons appear in the Stage column. Expand the Stage icons.



    Notice the stages from both the main flow and the subflow are represented in a single icon list.

©2014 ServiceNow, Inc. All rights reserved.

Summary

As a designer involved in service automation, you will know the meaningful business process stages independent of the design of the workflow. Once defined, a Stage Set can be shared across all workflows and subflows in the Application simply by importing it into workflow.

The Linear is the most effective Renderer when designing applications because the order that the icons appear to the user will remain consistent throughout the process, even when executing over multiple workflows.

©2014 ServiceNow, Inc. All rights reserved.

Lab Goal

Understand the Activities available for making decisions in a Workflow based on the values of the Current Record.

Lab 4 Routing Workflows Using Switch

Routing Workflows

One of the most powerful aspects of the Workflow Engine is its ability to interact with the platform. For the most part, this interaction is through the manipulation of the Current Record.

The Workflow Editor provides Activities that will read specified fields on the Current Record and route the workflow along specific transition paths based on the values of those fields.

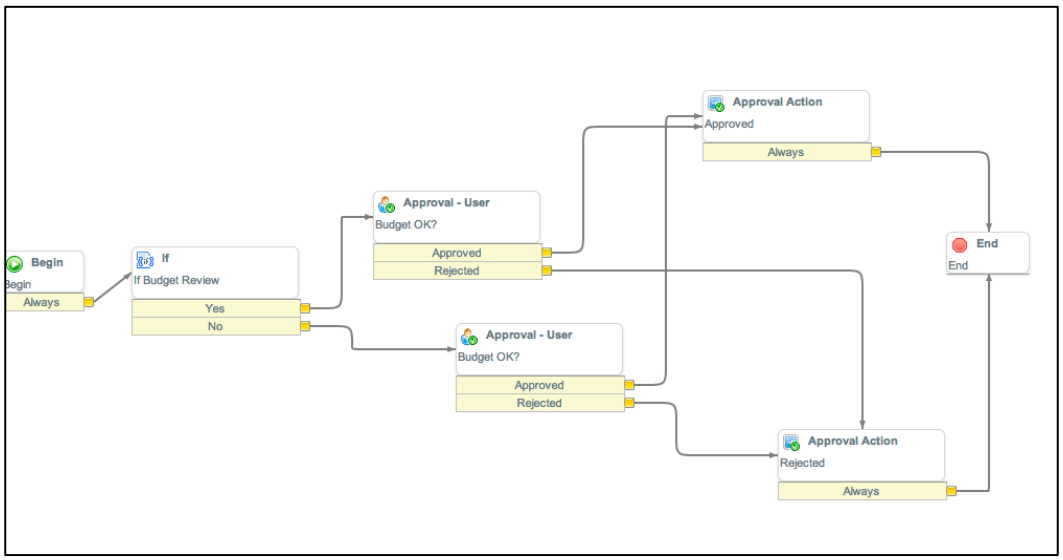
In the Conditions Category folder, the **Switch** and **If** Activities that assist the designer with routing. In the Utilities Category folder, the **Branch** can be used.

1. Return to the Workflow Editor.

2. Click **Open.**

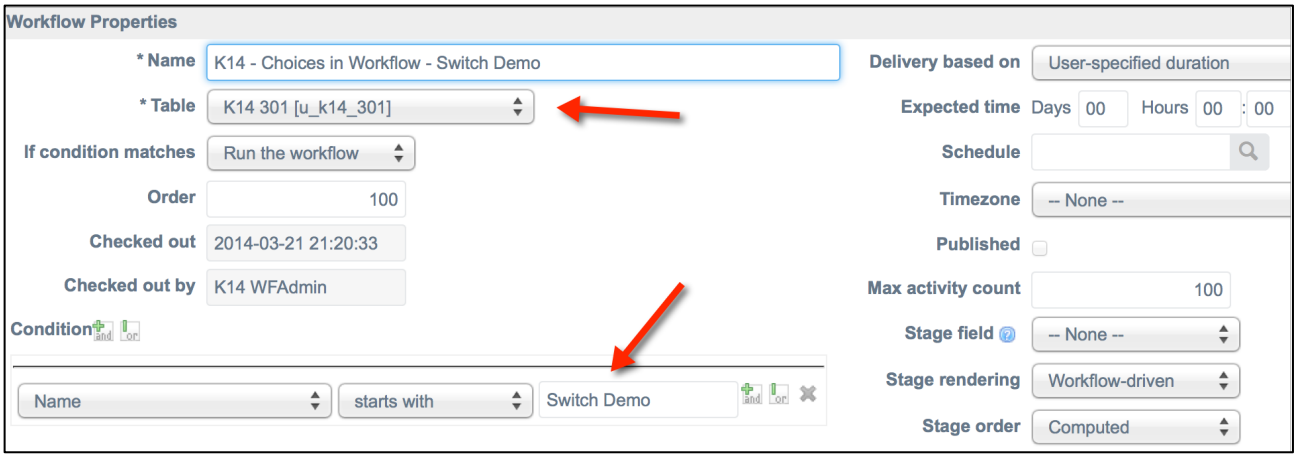
3. Select **K14 – Choices in Workflow – Switch Demo**.

©2014 ServiceNow, Inc. All rights reserved.



4. Select the **Gear Menu > Check Out**.

5. Select the **Gear Menu > Properties**.

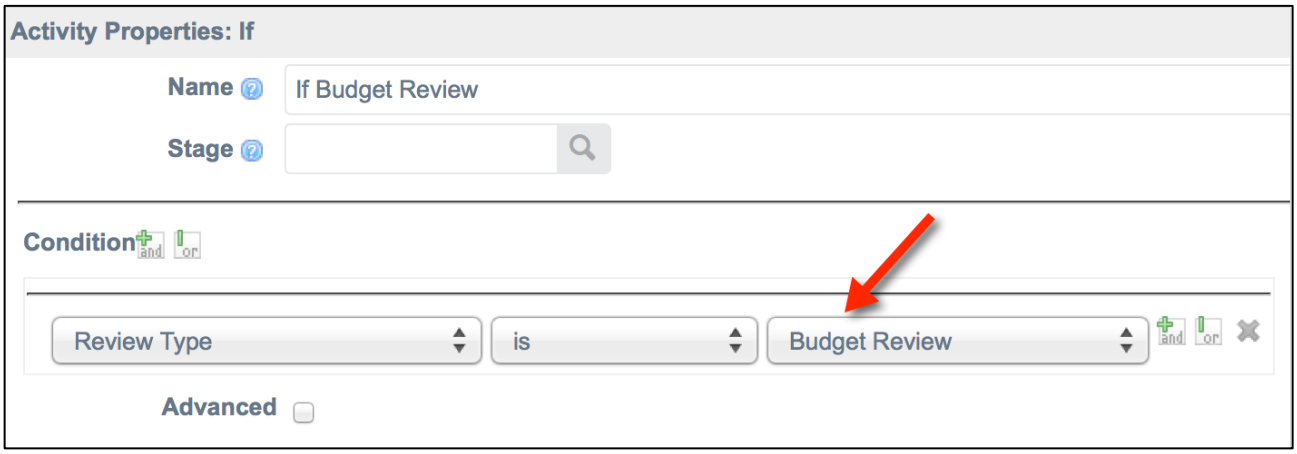


Notice that this workflow is on the Same K14 301 table used in earlier labs. In this demo, condition that runs the workflow is **Name** starts with **Switch Demo.**

6. Close the **Properties** form.

©2014 ServiceNow, Inc. All rights reserved.

7. Double click on the If **Budget Review** Activity.



The **If** Activity contains a simple condition builder. The condition choices are based on the fields that are available in the table associated with the workflow. In this case **K14 301 Demo**. Take a minute to look back at the form and look at the choices for Review type.

8. Close the **Activity** form.

9. Double-click on the If **Budget Review** Activity.

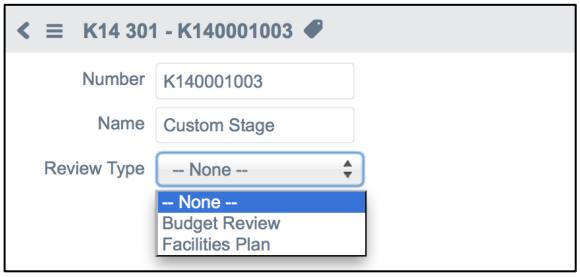
10. Return to the main ServiceNow tab.

11. In the Application Navigator Text filter type, **K14**.

12. Select a record from the K14 301 list.

13. Select **Review Type** control.

©2014 ServiceNow, Inc. All rights reserved.

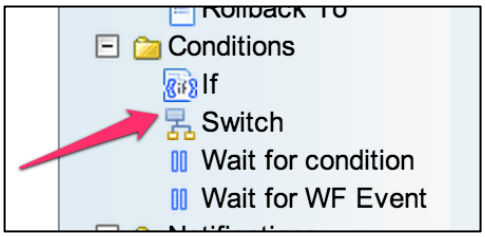


Notice that the Review type has 3 options, Budget Review, Facilities Review and a default of None.

14. Return to the Workflow Editor.

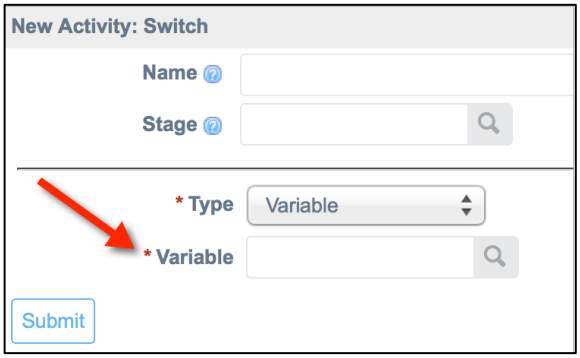
15. Close the Activity.

16. In the **Activities Tree**, expand the Conditions Category to locate the **Switch** Activity.



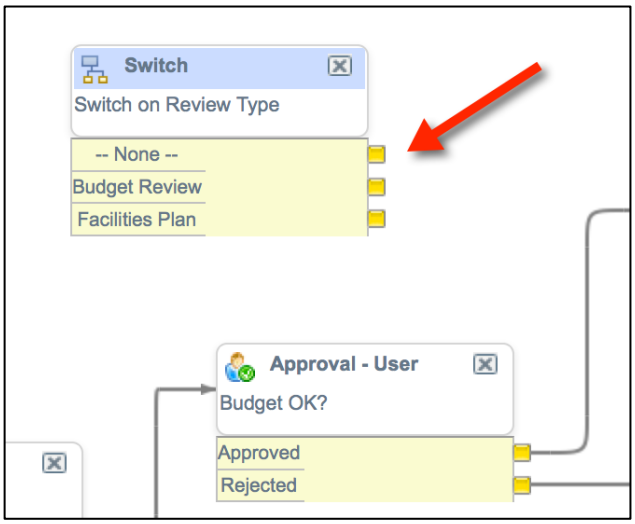
17. Drag the **Switch** Activity onto the Canvas.

©2014 ServiceNow, Inc. All rights reserved.



18. Fill out the form as follows:
    Name: **Switch on Review Type**
    Type: **Field**
    Field: **Review Type**

19. Click **Submit**.



Notice that the auto-generated conditions are the same values as the items in the choice list on the K14 301 forms. Using the Switch instead of the If in this situation allows us to route the workflow differently on a **None** selection.

20. Delete the If by clicking the X on the Activity.

©2014 ServiceNow, Inc. All rights reserved.

21. Route the **Begin** to the **Switch**

22. Route the **Budget Review** Condition to the **Budget Ok?** Approval.

23. Route the **Facilities Plan** Condition to the **Plan OK?** Approval.

24. In the **Utilities** folder in the right panel, find the **Log Message**.

25. Drag the **Log Message** onto the canvas.

26. Fill out the form as follows:
    Name: **Error**
    Message: **No Review Type Selected**

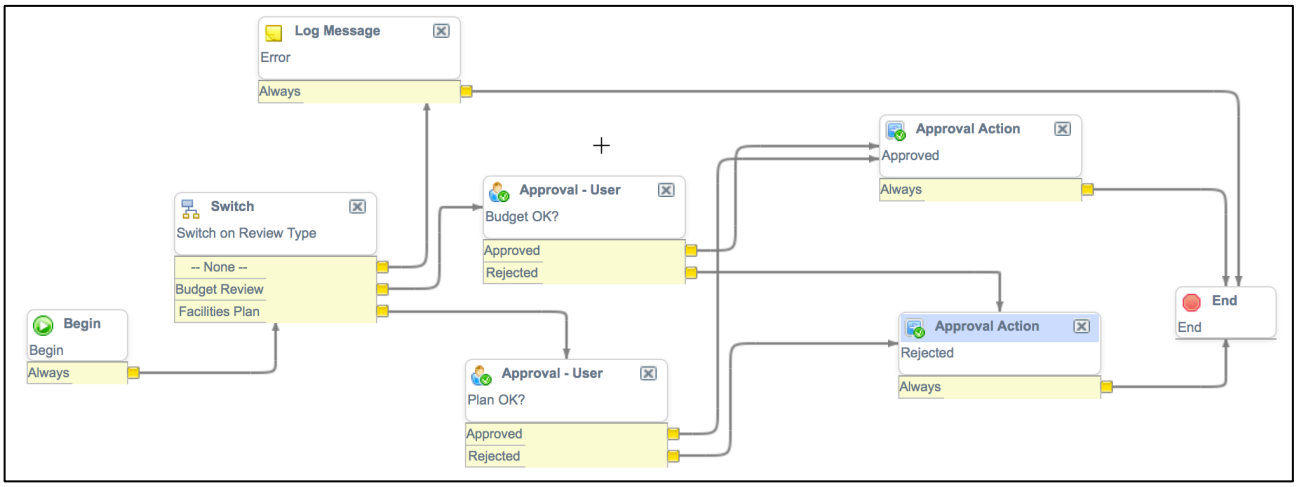
27. Click **Submit.**

28. Route the **None** Condition of the Switch to the **Error** Log Message.

29. Route the **Error** Log Message to **End**.

©2014 ServiceNow, Inc. All rights reserved.

Your workflow should look like this:



30. Return to the main ServiceNow tab.

31. Select **New** on the K14 301 list header.

32. Fill out the form as follows:
    Name: **Switch Demo**
    Review Type: **Budget Review**

33. Click **Submit**

34. Select **K14 Approvals.**

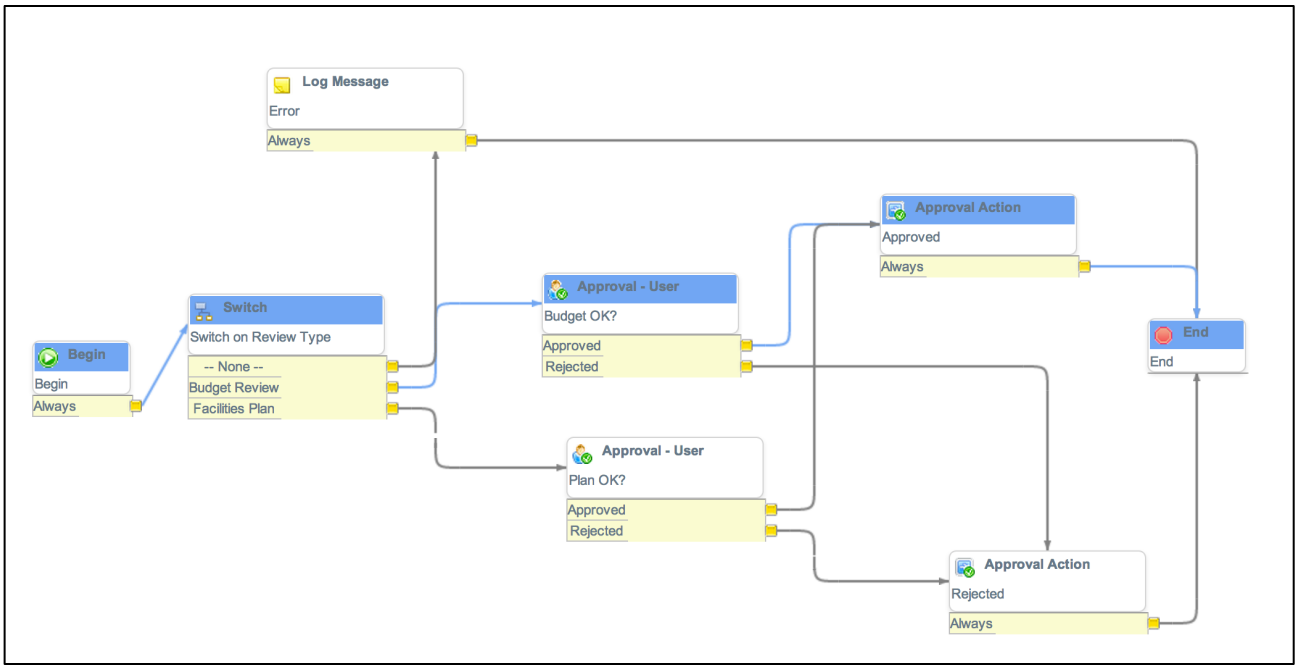
35. **Right click > Approve** the Requested Approval.

®2014 ServiceNow, Inc. All rights reserved.

36. Select **K14 Live Workflows**.

37. Select the Context that matches the Number of the last K14 301 record.

38. Select **Show workflow** from the Related Lists.



Notice that the Workflow was routed through transitions of the Workflow according the selection in the Review Type selection.

39. Repeat steps 31 - 38 for the **None** selection. Fill out the form as follows:
    Name: **Switch Demo**
    Review Type: **None**

40. Repeat steps 31 to 38 for the **Facilities Plan** selection. Fill out the form as follows:
    Name: **Switch Demo**
    Review Type: **Facilities Plan**

41. Validate the routing matches the selection.

©2014 ServiceNow, Inc. All rights reserved.

Summary

The Workflow Editor offers several ways to route a workflow through transitions based on the data in the Current Record.

The **If** Activity can be configured to evaluate values of fields in the Current Record using a Condition builder.

The **Switch** Activity auto-generates conditions based on the values of a choice list in the Current Record.

©2014 ServiceNow, Inc. All rights reserved.

Lab Goal

Demonstrate options in workflow Routing.

Lab 4
Other Routing Workflows

Routing Workflows with Conditions

As we saw in the previous lab, a workflow can be routed using conditions of activities such as **If** and **Switch**. Routing off conditions is the most common way to route a workflow but it is not the only way.

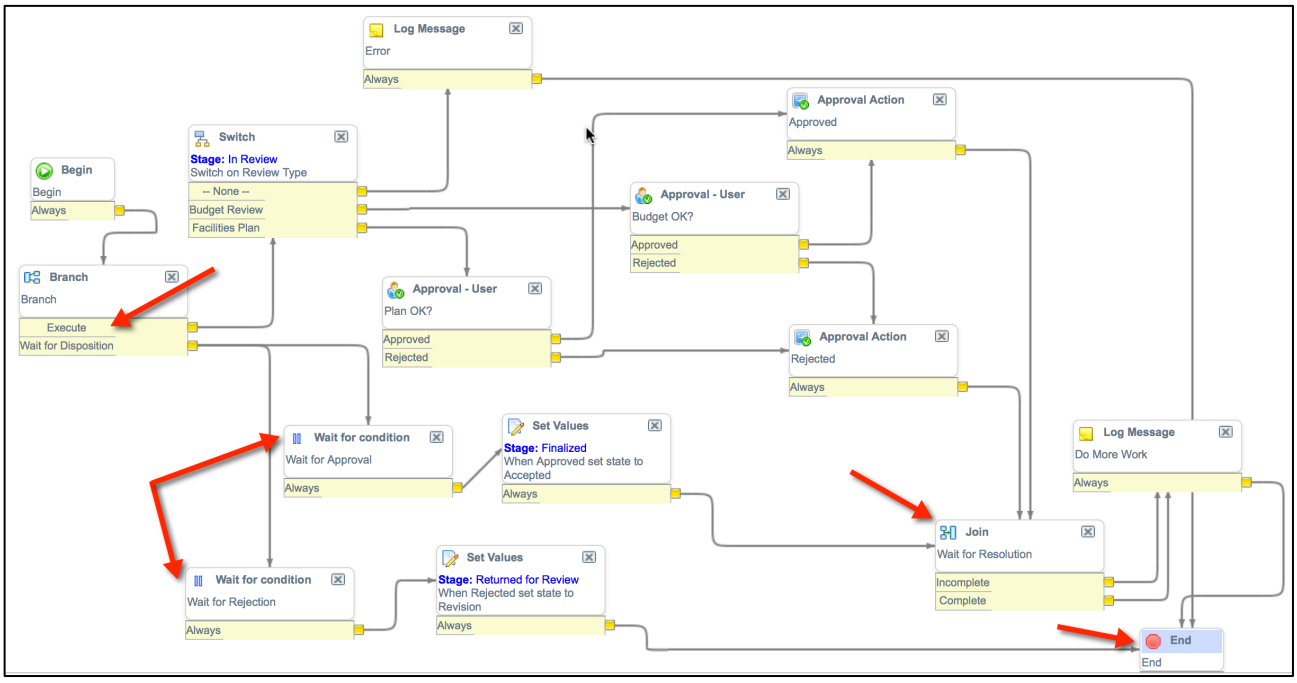
Workflows can have execution paths that wait for an event to fire or wait until for a state change on a specified field. The activities **Wait for Event** and **Wait for Condition** both provide a way for a workflow to wait.

Waiting for events and conditions can simplify the layout of a workflow and can provide a way to visually isolate transition paths in a workflow with complicating the main execution path of a workflow.

1. Return to the Workflow Editor.
2. Click **Open**.
3. Select the **K14 – Choices in Workflow – Wait for Condition** workflow.
4. **Gear Menu > Check Out**

©2014 ServiceNow, Inc. All rights reserved.

Your workflow should look like this:



This is essentially the same example workflow as the Switch example but with added decision making to provide a more complete process. We will detail the Activities pointed to by the arrows.

The process being defined is as follows:

- Determine the Request Type
- Route the work to the correct Approver.
- Based on the Approve | Reject status of the Approver
    - Update the State of the record to **Final** on **Approve**
    - Update the State of the record to **Revision** on **Reject**

Branch

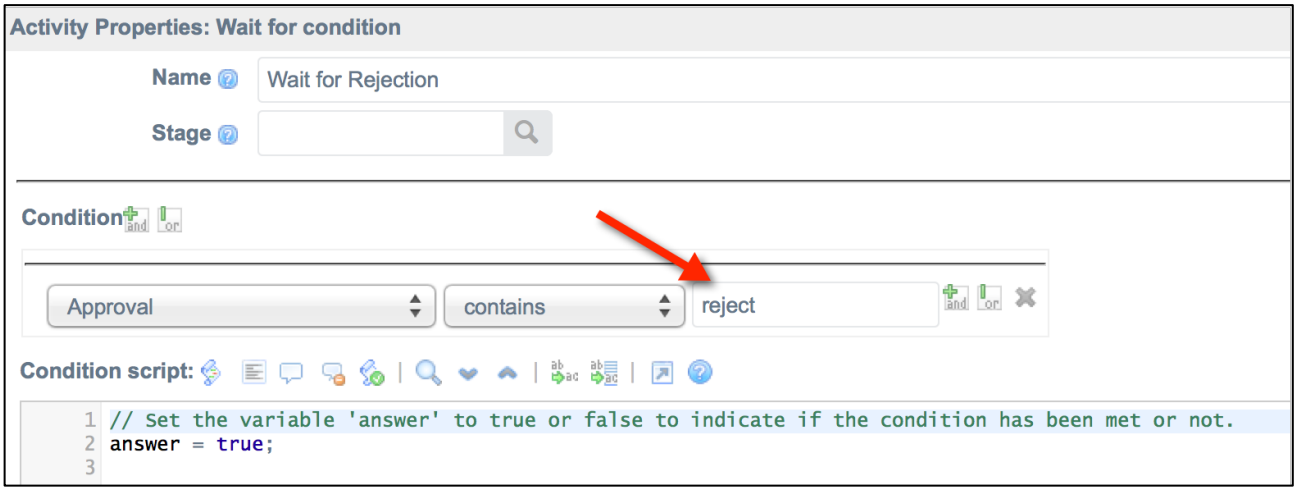
The Branch activity does **not** by default evaluate the Conditions. Its primary purpose is to direct the workflow down multiple paths and provide a label for them. You can script conditions, but this example does not.

- **Execute** condition routes to the same process flows as the switch example in the prior lab.
- **Wait for Disposition** condition will route the workflow to two **Wait for Condition** Activities.
    - These activities are waiting for mutually exclusive conditions, so when designing the workflow, they were routed off the same Branch condition.

©2014 ServiceNow, Inc. All rights reserved.

Wait for Condition
The **Wait for Condition** Activity will execute and then wait for a specified field to be assigned a value. Once that value is assigned the Activity will transition. If that condition is not met in the process of executing the workflow, the Activity will never transition.

5. Double-click on **Wait for Rejection** Activity.



Notice the condition builder specified the field on the Current Record to listen to and the value to listen for.

6. Close the Activity.

©2014 ServiceNow, Inc. All rights reserved.

7. Double-click on **Wait for Approval** Activity and see that it is waiting for the same field but with a different value.

   By listening to the same field for different values, the designer can use the **Wait for Condition** to route the workflows down two mutually exclusive paths. This same thing could have been done with an **If** Activity. Using an If however, would require the conditional logic to be intermingled with the execution logic. Using the **Wait for Condition** allows a designer to separate the transition paths into isolated, testable execution scenarios.

   End
   Any transition to the **End** Activity ends the Workflow. Though this seems obvious, the most important note is that because any transition ends the workflow, the first transition to the workflow will cancel any executing activities. In the Context Diagram we see a record was created that did not specify a **Review Type.** The workflow logs a message and then routes directly to **End** notice that the two Wait for Conditions are cancelled. It was the execution of the first transition to **End** that cancelled those activities.

   There are however, many reasons to have concurrently transitioning paths do some work and then wait for other transition paths before continuing or before ending a workflow. The **Join** Activity provides a meeting place for different transition paths to route to.

   Join
   The **Join** Activity can accept any number of transitions as inputs. Each time a transition routes to a Join the activity does an evaluation of all transitions that are assigned to it. The evaluation determines if all **possible** routes to decide if it is finished waiting. A Join can finish on a **Complete** or an **Incomplete** condition.
   - **Complete** indicates that all transitions arrived to the **Join** successfully.
   - **Incomplete** indicates that not all transitions arrived, but all **possible** transitions arrived.
     - In the example workflow, **Approval Action Approved** and **Approval Action Rejected** both transition to the same **Join**. Because these are mutually exclusive transitions, the Join will always be Incomplete. On Incomplete, the **Join** will progress to the next activity.

8. Return to the main ServiceNow tab.

9. Create a new K14 301 record.

©2014 ServiceNow, Inc. All rights reserved.

10. Fill out the form as follows:
    Name: **Wait for Condition**
    Review Type: **Budget Review**

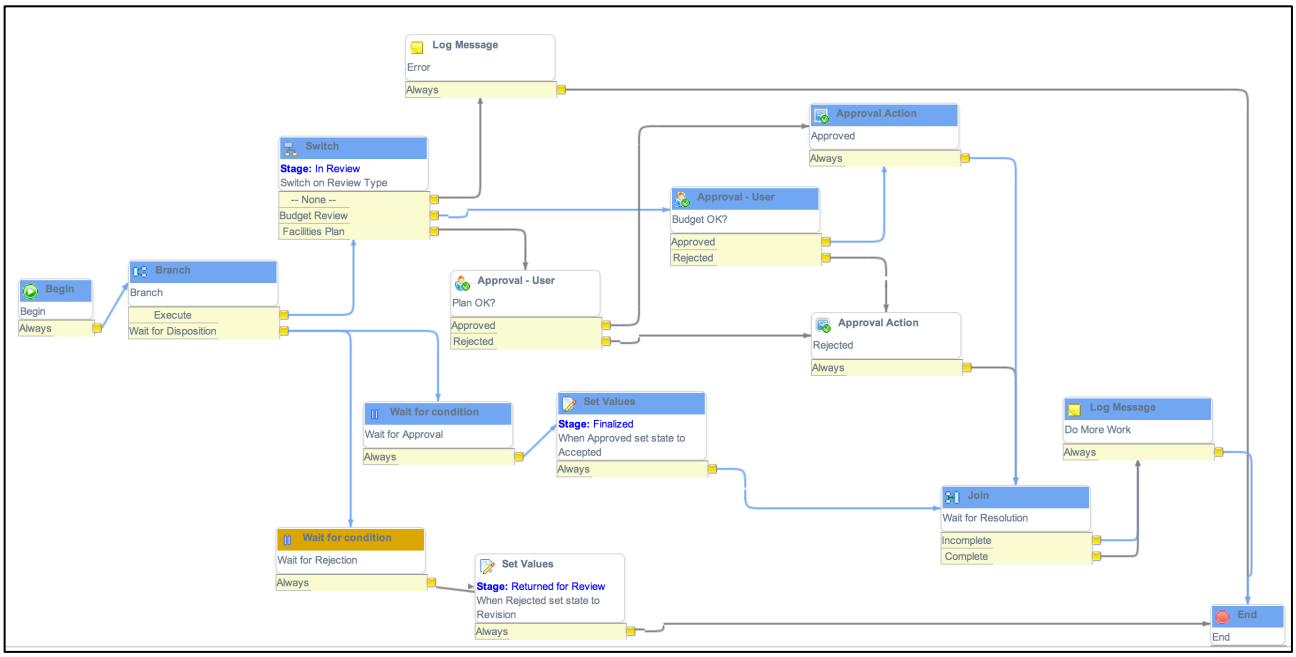
11. Click **Submit** .

12. **K14 > K14 Approvals**.

13. Approve the associated **Approval.**

14. Examine the Workflow Context**.**

15. **K14 > K14 Live Approvals**.



Notice that the Wait for condition was cancelled. This is because the **Do More Work** activity got to the **End**. The first one to **End** will always win.

16. Using the **Branch** Activity and the **Wait for Conditions** as your guide, take a few moments to test some other routings in the workflow and examine the difference.

®2014 ServiceNow, Inc. All rights reserved.

Lab Goal

Understand Workflow within a Glide Record.

Lab 5
Workflow Engine

The Workflow Engine in ServiceNow

The Workflow Engine runs part of the Glide Engine. To design applications or extend existing applications with Workflow Processes it is important to understand where the Workflow Engine is as part of the Glide record transaction.

This lab introduces the Glide Transaction and highlights the place and role of the Workflow Engine within that Transaction.

The Workflow Engine will always guarantee updates to the current record at the end of each process point.

The Glide Engine is data drive and so reacts to **events** that signal a change in the state of some data.

Glide events are **insert**, **update**, **delete** or **cancel**. These events are fired into the engine in various ways. Some examples are:

- UI Actions:
  - A user clicks a **Submit** button -> an **insert** event fires into the Glide Engine.
  - A user clicks an Update button -> an **update** event fires into the Glide Engine.
  - A user clicks a Delete button -> a **delete** event fires into the Glide Engine.
  - A user clicks the Cancel Transaction icon -> a **cancel** event fires into the Glide Engine.

®2014 ServiceNow, Inc. All rights reserved.

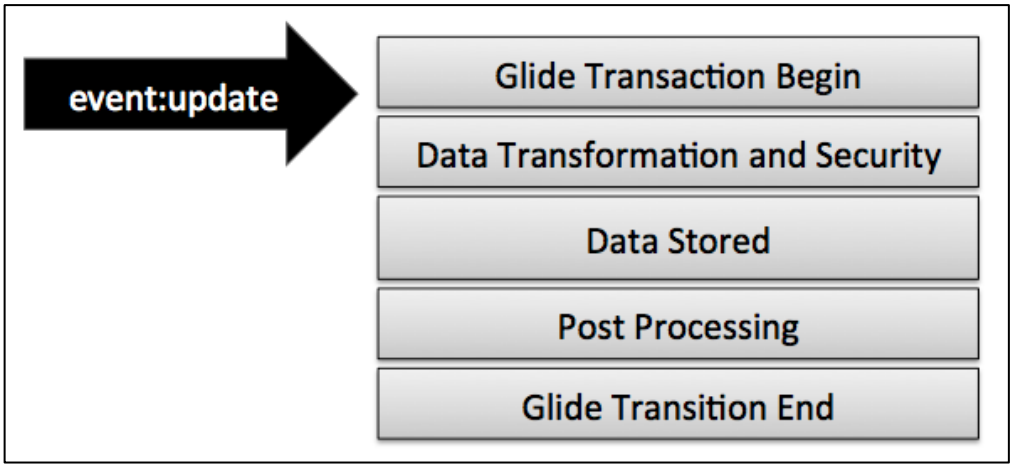
- Within Scripts:
  - A script that initializes a GlideRecord -> an **insert** event fires into the Glide Engine.

```
GlideRecord gr = new GlideRecord ('incident');
gr.initialize();
gr.insert();
```

  - A Business Rule that deletes a **current** -> a **delete** event fires into the Glide Engine.

```
GlideRecord gr = new GlideRecord ('incident');
gr.get('specify_sys_id');
gr.delete();
```

For each of these events fires into the Glide Engine a series of transformations and processes begins. **One event = One Glide Transaction**. As each of these events fires into the Glide Engine a series of transformations and processes begins.



Workflow is part of both the Data Transformation layer as a **Default Workflow** and part of Post Processing as a **Deferred Workflow**. In either case the Workflow Engine, along with other engines such as Field Normalizaiton and Data Policy, are sandwiched between two sets of business rules.

The Order value of Business Rules determine their order of execution around the sandwiched engines. Business Rules that are ordered less than 1000 will run before the other engines, including Workflow. Business Rules that are order greater than or equal to 1000 will run after other engines, including Workflow.

Both Business Rules and Workflows can be configured to run before data is stored (transformation) or after data is stored (post processors).

©2014 ServiceNow, Inc. All rights reserved.

To configure a Business Rule to run before data is stored, check the **Before** check box on the form.

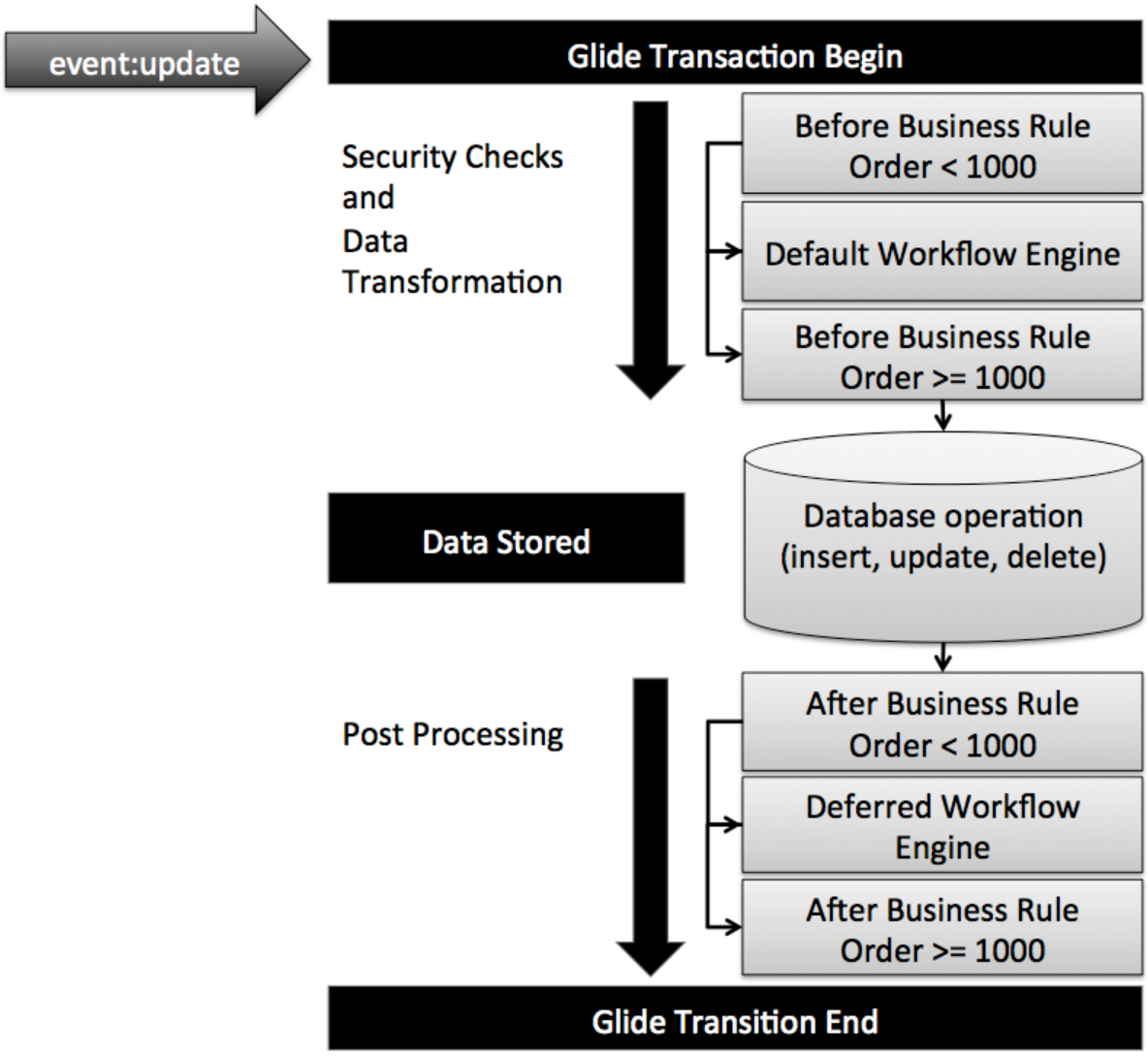
By default, Workflows will run before data is stored.

To configure a Business Rule to run after data is stored, check the **After** check box on the form.

To configure a Workflow to run after data is stored, check the **Run After Business Rules** box on the Workflow Properties form.

NOTE: We will be looking at each of this specifically in just a bit.

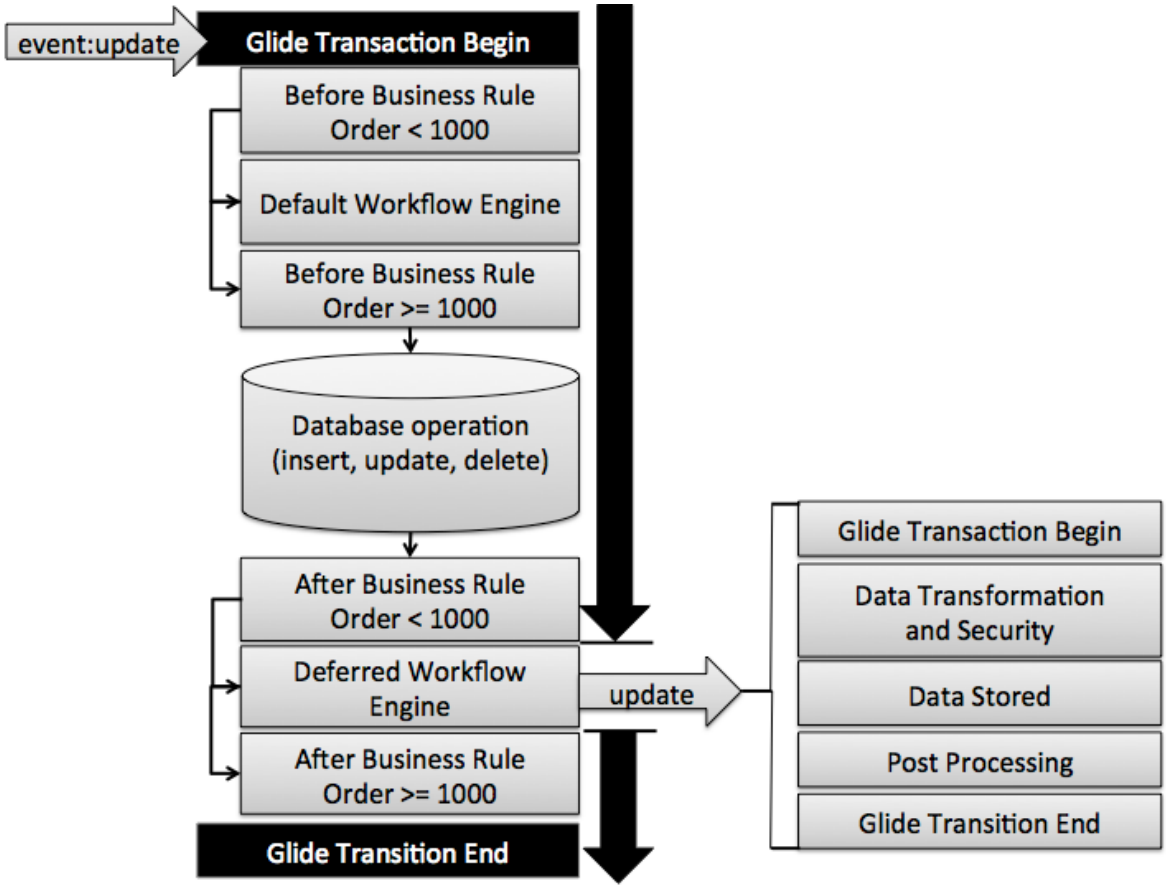
If we were to add more detail to the graphic of a transaction, it would look like this.



©2014 ServiceNow, Inc. All rights reserved.

As a Workflow Designer it is important to understand the timing of each participant in the Glide Transaction.

A **Deferred Workflow**, that is one that runs after the database operation, will still ensure the preservation of changes to the current record, that preservation is **not** part of the same database transaction. Remember: **One event = One Glide Transaction.** Since the data is already stored when a **Deferred Workflow** is called, the Workflow Engine forces a second **update** event. However, this update event will begin an independent and nested transaction, which will complete before the After Business Rules ordered greater than or equal to 1000 of the original event fire.



NOTE: Because the Default Workflow runs as part of a database event and because a **Deferred Workflow** performs its own nested update of current, **it is never necessary to call current.update(); anywhere within a Workflow Activity Script field**.

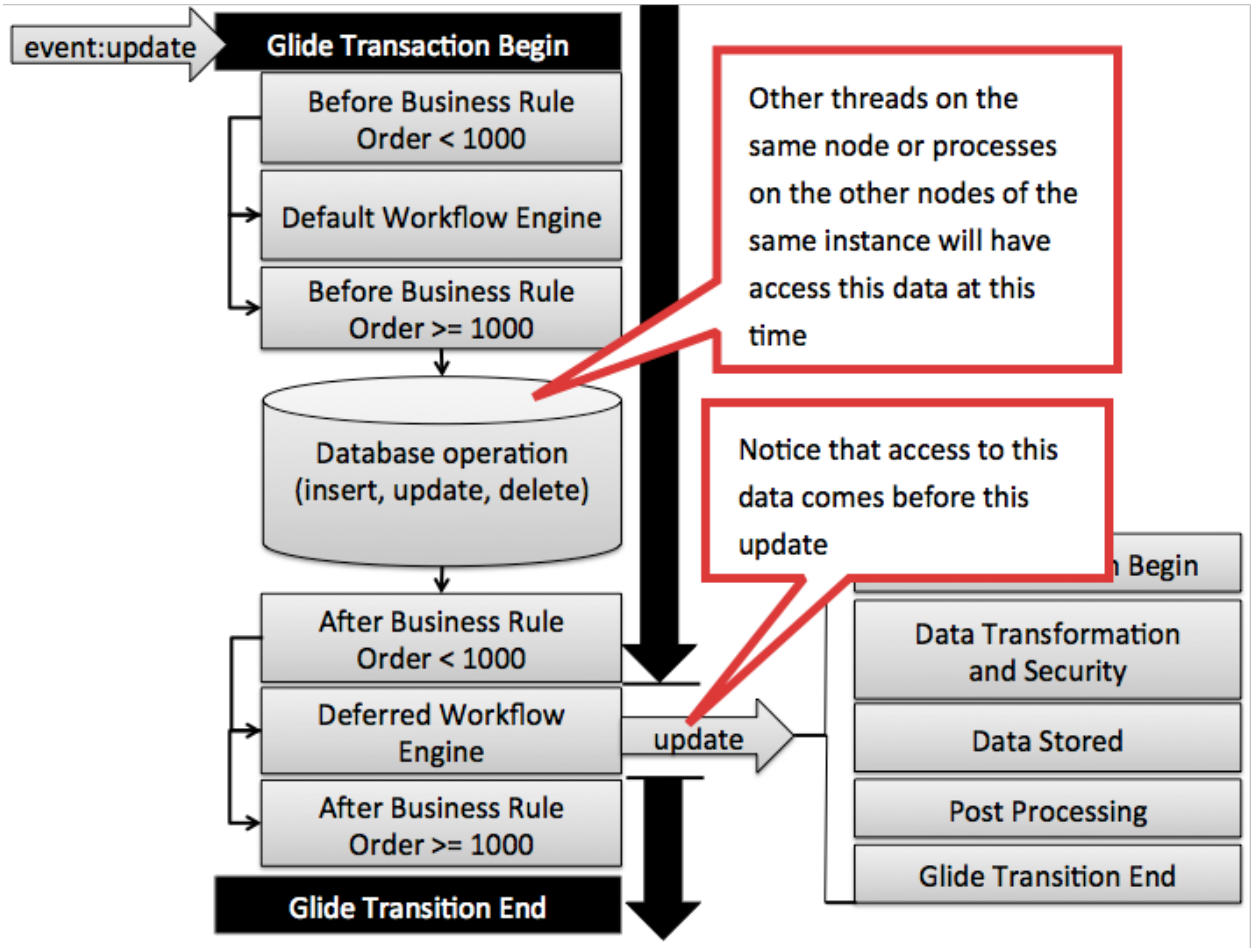
Keep in mind that every call to update() creates an independent **update** event. As an independent event it will cause an unnecessary nested update that will affect the performance of your application.

©2014 ServiceNow, Inc. All rights reserved.

One event = One Glide Transaction

It is important as a Workflow Designer within an application to be aware of what Business Rules and Engines are also in the Application and what effect each may have on the data that drives the process.

A Workflow Designer needs to be able to predict when and how data is being transformed and by which process player. It is equally important, depending on the complexity of the application, to know when other nodes or threads will have access to that data.



NOTE: Most business processes work very well with **Default Workflows**. If you choose to configure a Workflow to be a **Deferred Workflow**, beware of the timing of the two updates on the current record.

If there is not a **Deferred Workflow** assigned to a Glide Record in the Post Processing phase of the Glide Transition, there will be **no update** to the current record by default. The best use of the Post Processing phase of the Glide Transaction is to react to the data change not continue to change it.

©2014 ServiceNow, Inc. All rights reserved.

Lab Goal

Some practice in understanding how data is transformed in the process of a Glide Record transaction.

Lab 6
Engine Ordering and Data

Practicing Engine Ordering

The following exercises are to become familiar with the transformation of data as it moves through the Glide Transaction.

In the following tables there are a series of Business Rules and Workflows presumed. Each one simply does a modification to the same field throughout the Glide Transaction.

The script in each is very simple: `current.desc += 'some value';`

For example:

```
current.desc += 'K14';     // the value of current.desc = K14

current.desc += '-WF 301'; // the value of current.desc = K14-WF 301
```

and so on.

©2014 ServiceNow, Inc. All rights reserved.

1. Walk through following scenario:

Process Player	order	Action on desc field
Business Rule	100	current.desc = 'BLB';
DATABASE TRANSACTION		

QUESTION: What is the value of **current.desc** in the database at the end of the Glide Transaction?

2. Walk through the following scenario:

Process Player	Order	Action on desc Field
Business Rule	100	current.desc = 'BLB';
Workflow(Default)	100	current.desc += '-WFDefault';
DATABASE TRANSACTION		
Business Rule	1200	current.desc += '-BHA';

QUESTION: What is the value of **current.desc** in the database at the end of the Glide Transaction?

QUESTION: Will the value of **current.desc** ever equal **BLB-WFDefault-BHA** in the database? Why or Why not?

©2014 ServiceNow, Inc. All rights reserved.

3. Walk through the following scenario:

Process Player	Order	Action on desc Field
Business Rule	100	current.desc = 'BLB';
Workflow(Default)	100	current.desc += '-WFDefault';
Business Rule	1200	current.desc += '-BHB';
DATABASE TRANSACTION		
Business Rule	5	current.desc += '-BLA';
Workflow (Deferred)	100	current.name += 'K14';

QUESTION: What is the value of **current.desc** in the database at the end of this Glide Transaction?

QUESTION: How many updates were performed to get to that place?

©2014 ServiceNow, Inc. All rights reserved.