knowledge14
April 27–May 1, 2014 • Moscone Center • San Francisco, CA

Lab Guide

Orchestration 201:

Core Capabilities

Jun Zhou & Shawn Turnbow

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

Login / Passwords:

admin / Knowledge14

Lab Goal

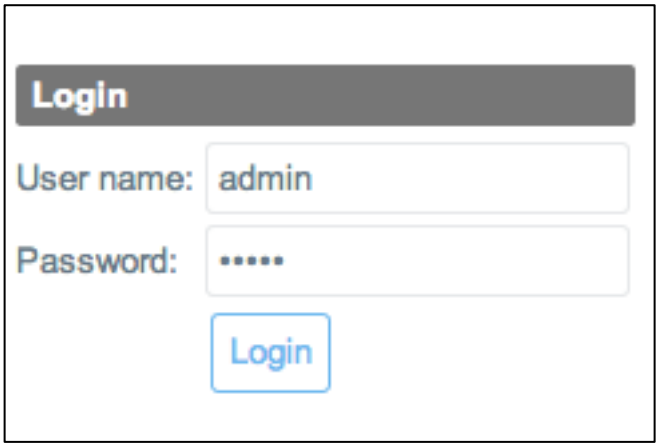
Build an Orchestration workflow to automate the manual installation of an application to a web server, customize the home page, and verify the installation visually.

Lab 1
Create Subflow

Log on to Your Instance

1. Your instance URL is on the cover page.

2. Login using the **admin** credentials provided on the cover.



®2014 ServiceNow, Inc. All rights reserved.

Understand Our Starting Point

Let's examine the initial workflow.

1. Navigate to **Orchestration > Workflow Editor**.

2. Click **Open**.

3. Select **K14 – Order Application** to open the workflow.

This workflow:

- Obtains approval from the requestor's manager.
- When the request is approved, it checks how many licenses are available and decides if it needs to acquire more licenses.
- If a license is acquired, it creates an application installation task.
- Once the application is installed, it creates an application verification task.

Our goal:

- Replace the last 2 manual tasks with automation to install the application and test that the application was installed successfully. We will create a subflow to perform the tasks automatically and incorporate the subflow into this workflow.

©2014 ServiceNow, Inc. All rights reserved.

Create the Subflow

A subflow is really just a workflow.

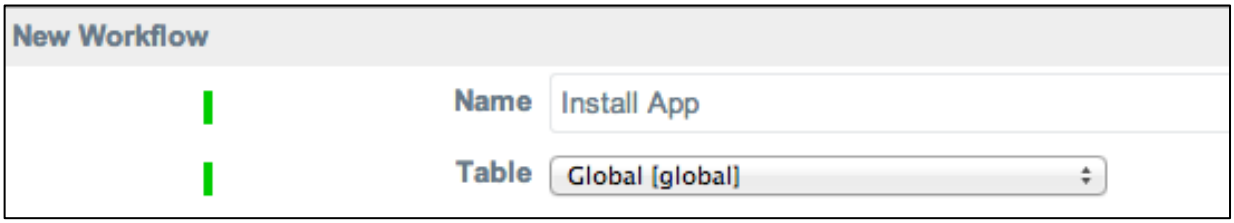
1. Create the subflow: navigate to **Orchestration > Workflow Editor**.

2. Click **New**.

3. Enter or select the following:

   Name: **Install App**
   Table: **Global [global]**

   Leave all of the other fields with their default values



4. Click **Submit**.

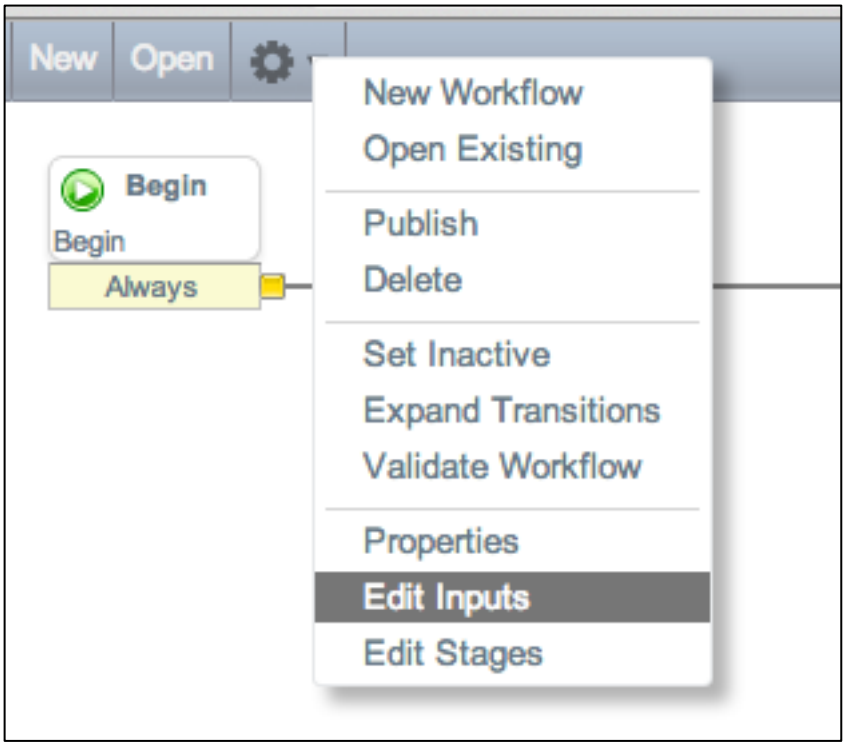
®2014 ServiceNow, Inc. All rights reserved.

Add Input Variables

The subflow takes 3 input variables:

- The IP address of the target server the application will be installed on.
- A string parameter required by the installation script that will be called on the target server.
- The web application home page.

1. Click the gear menu.
2. Select **Edit Inputs**.

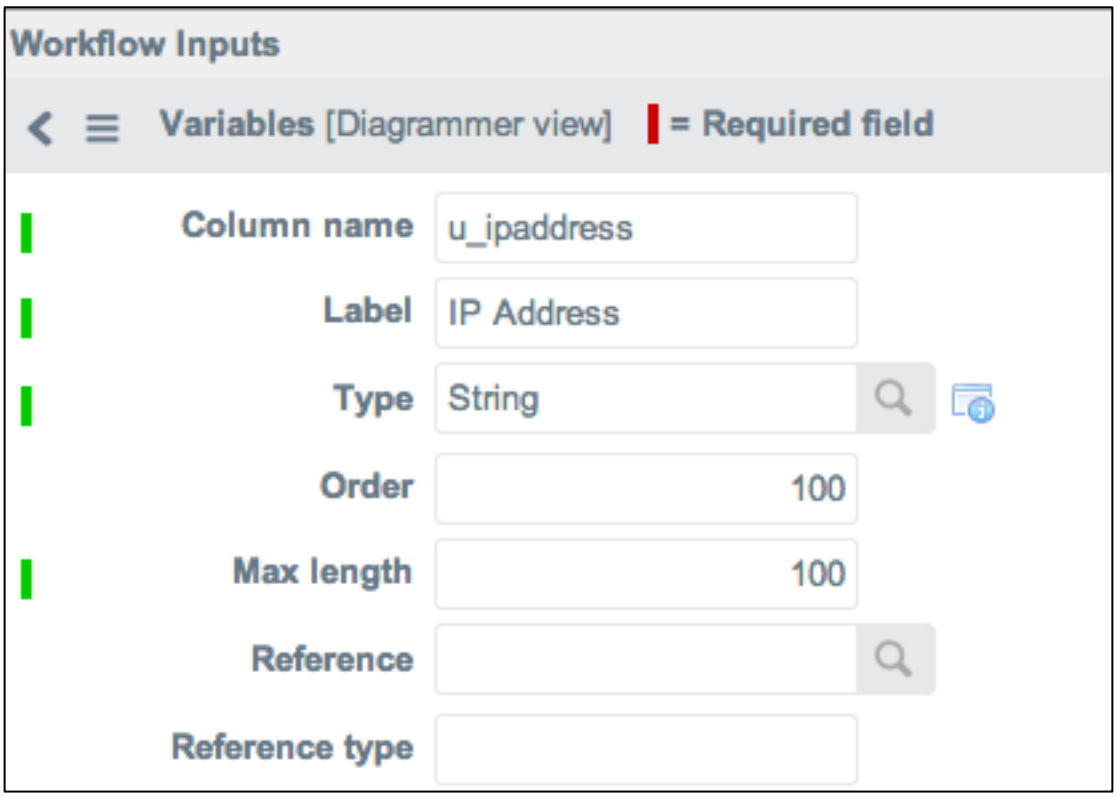


3. Click **New**.

©2014 ServiceNow, Inc. All rights reserved.

4. Enter or select the following:

   Column name: **u_ipaddress**
   Label: **IP Address**
   Type: **String**
   Order: **100** (leave alone)
   Max length: **100**

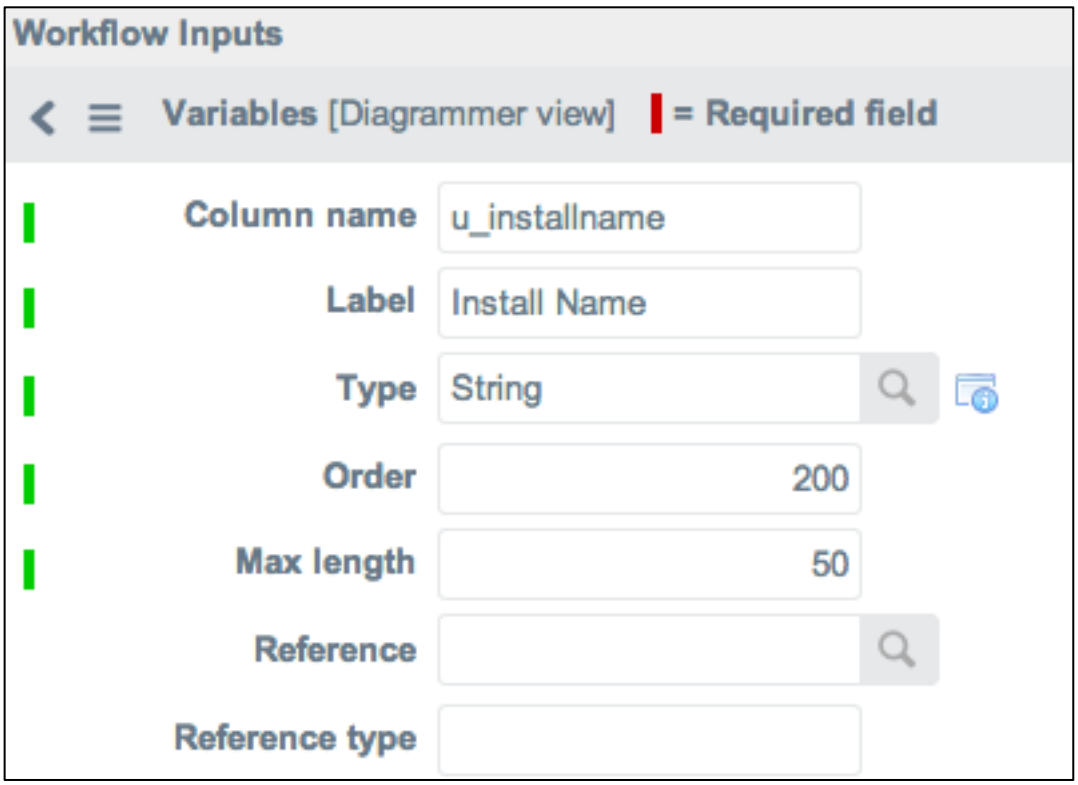


5. Click **Submit.**

®2014 ServiceNow, Inc. All rights reserved.

6. Repeat steps 3 through 5, entering:

   Column name: **u_installname**
   Label: **Install Name**
   Type: **String**
   Order: **200**
   Max length: **50**

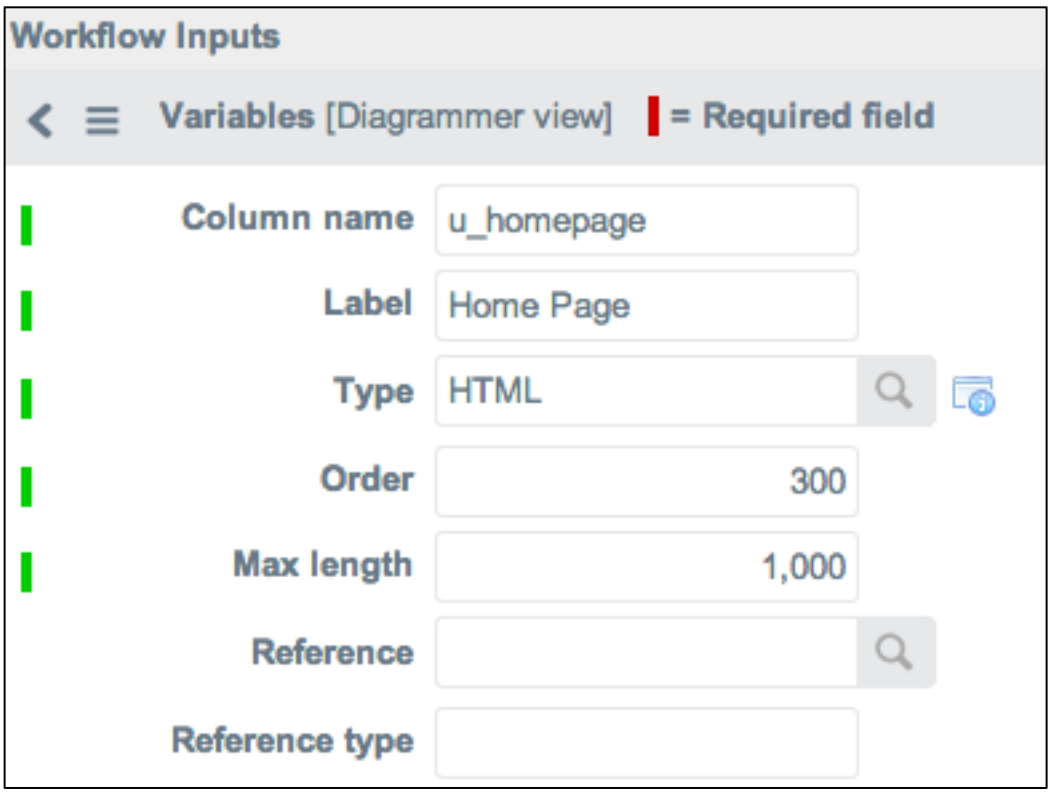


7. Click **Submit.**

®2014 ServiceNow, Inc. All rights reserved.

8. Repeat steps 3 through 5, entering:

   Column name: **u_homepage**
   Label: **Home Page**
   Type: **HTML**
   Order: **300**
   Max length: **1000**



9. Click **Submit**.

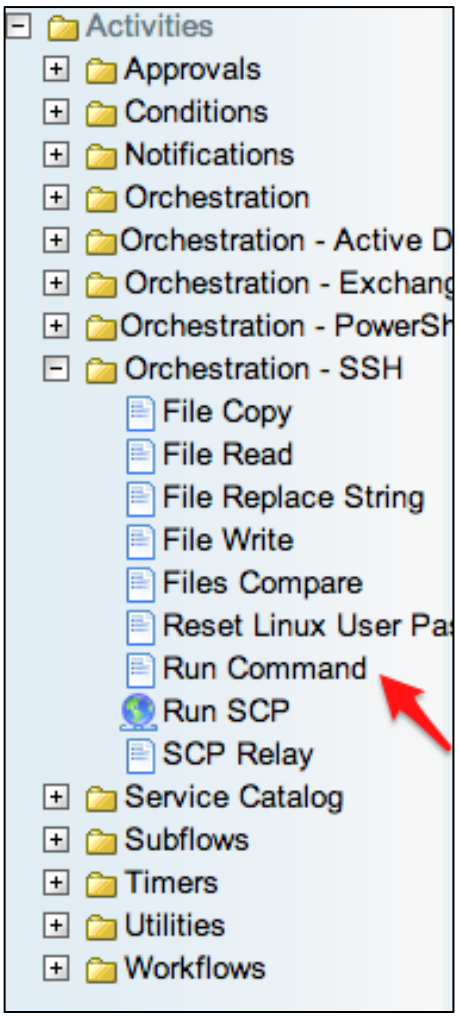
10. Click the little **X** in the upper-right of the Workflow Inputs window to close it and return to the workflow canvas.

®2014 ServiceNow, Inc. All rights reserved.

Add Run Command Activity

We configure this activity to run our install script on the target system.

1. Expand **Orchestration – SSH** activities on the right side of the workflow editor.



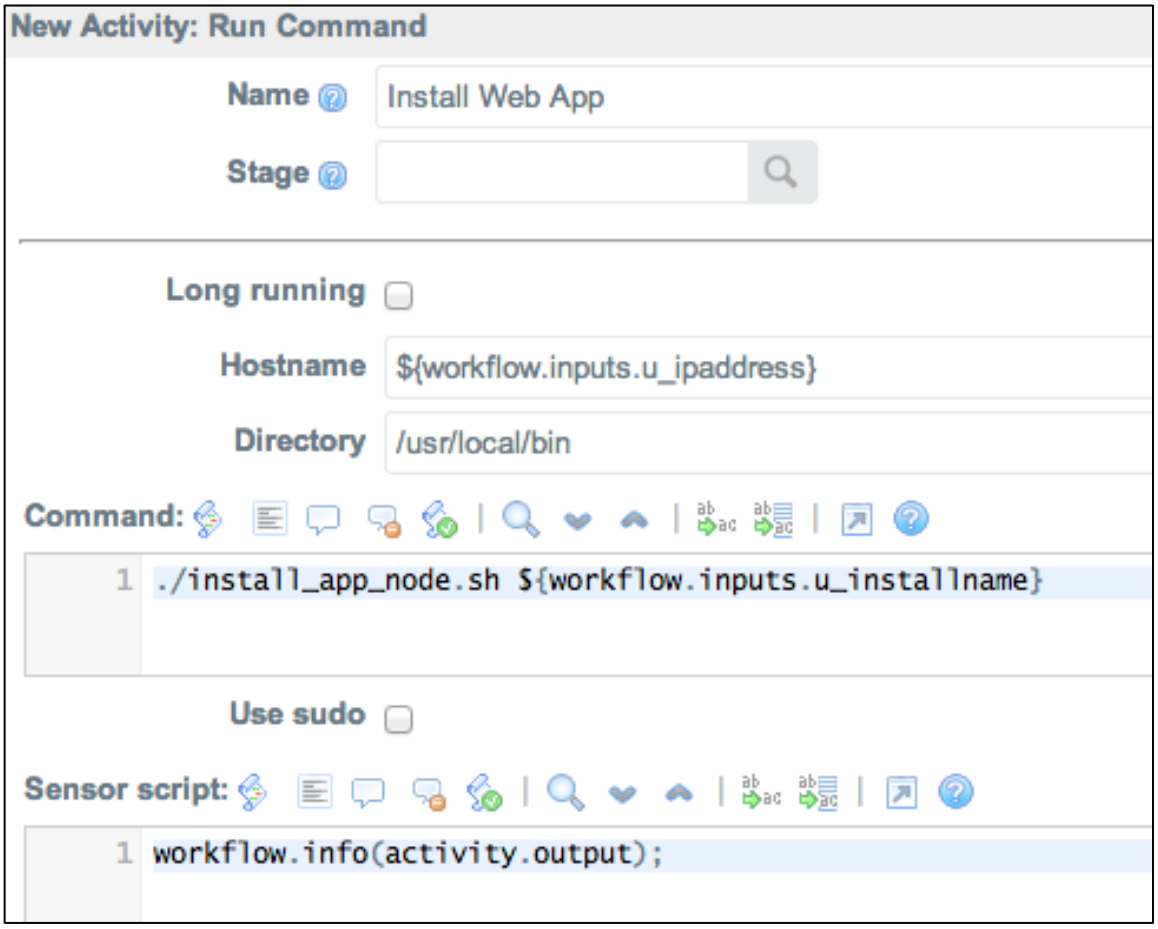
2. Drag **Run Command** onto the transition between **Begin** and **End**.

©2014 ServiceNow, Inc. All rights reserved.

3. Here we enter in the variable to run a local script in **/usr/local/bin called install_app_node.sh** and we parse the results with the corresponding sensor script.

   Enter the following for the activity variables:

   Name: **Install Web App**
   Hostname: **${workflow.inputs.u_ipaddress}**
   Directory: **/usr/local/bin**
   Command: **./install_app_node.sh ${workflow.inputs.u_installname}**
   Sensor script: **workflow.info(activity.output);**



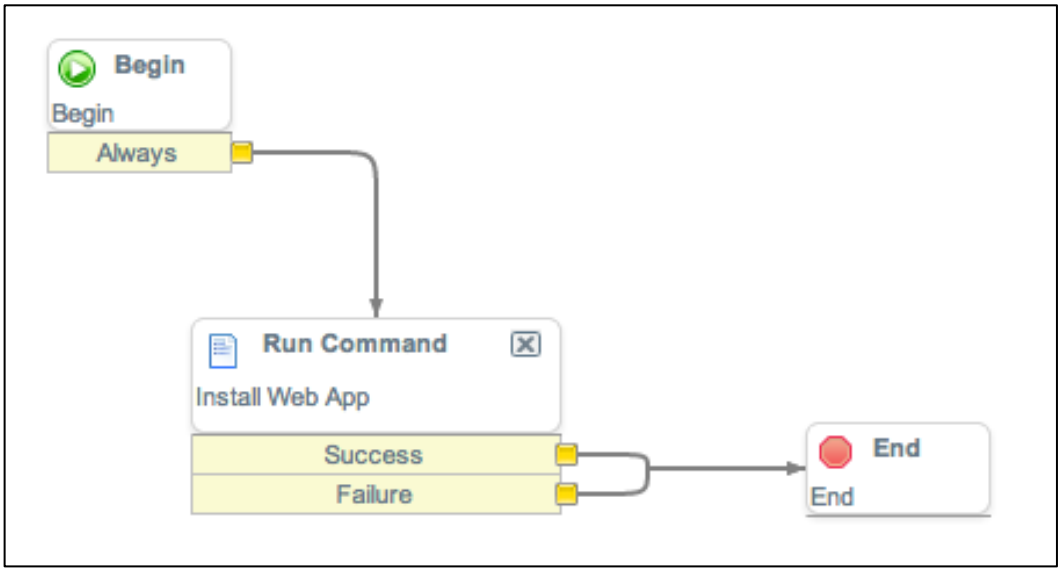
4. Click **Submit**.

©2014 ServiceNow, Inc. All rights reserved.

5. Make transitions from:

   Begin to **Run Command**
   Run Command Success to **End**
   Run Command Failure to **End**

   The workflow appears as shown.

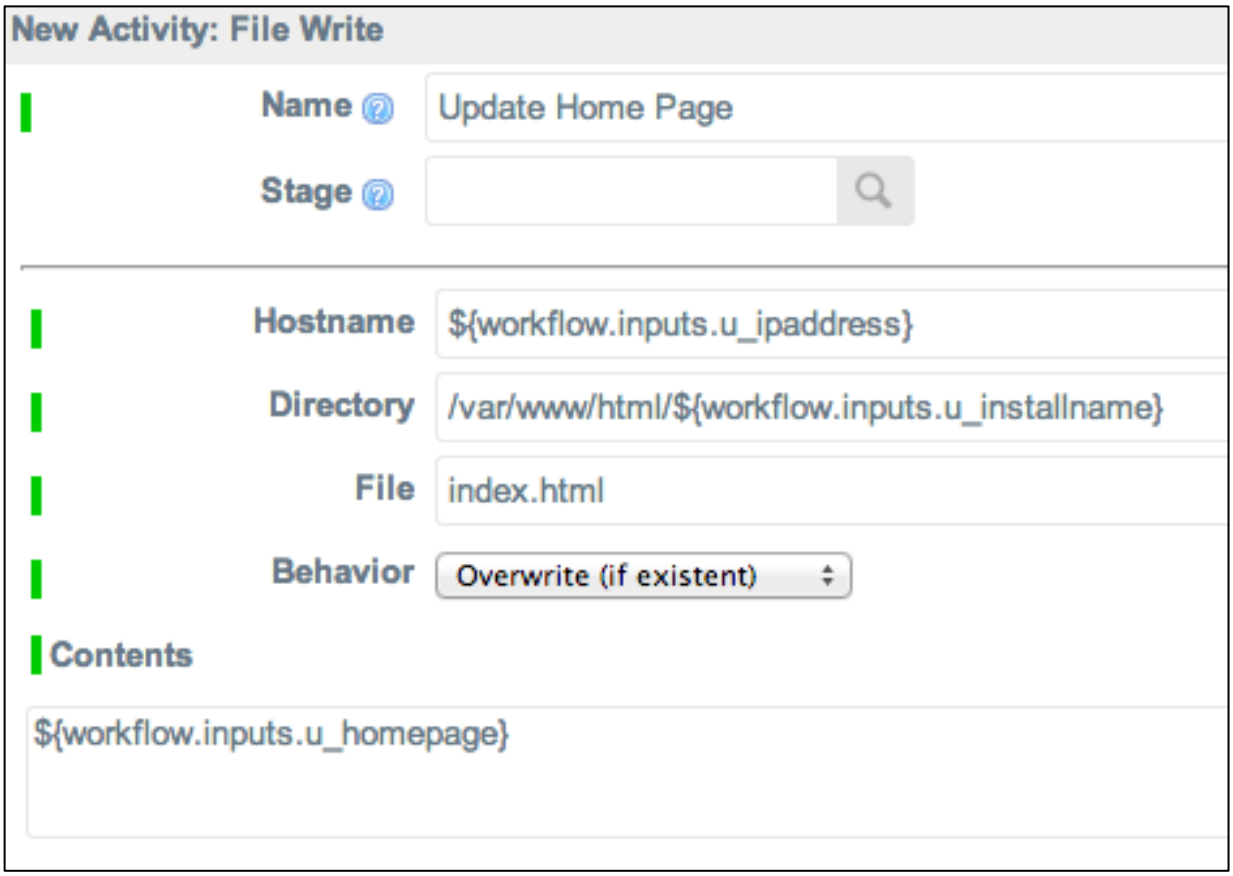


6. Drag **File Write** onto the transition between Success condition of **Run Command** and **End**.

©2014 ServiceNow, Inc. All rights reserved.

7. Enter the following for the activity variables:

   Name: **Update Home Page**
   Hostname: **${workflow.inputs.u_ipaddress}**
   Directory: **/var/www/html/${workflow.inputs.u_installname}**
   File: **index.html**
   Behavior: **Overwrite (if existent)**
   Contents: **${workflow.inputs.u_homepage}**



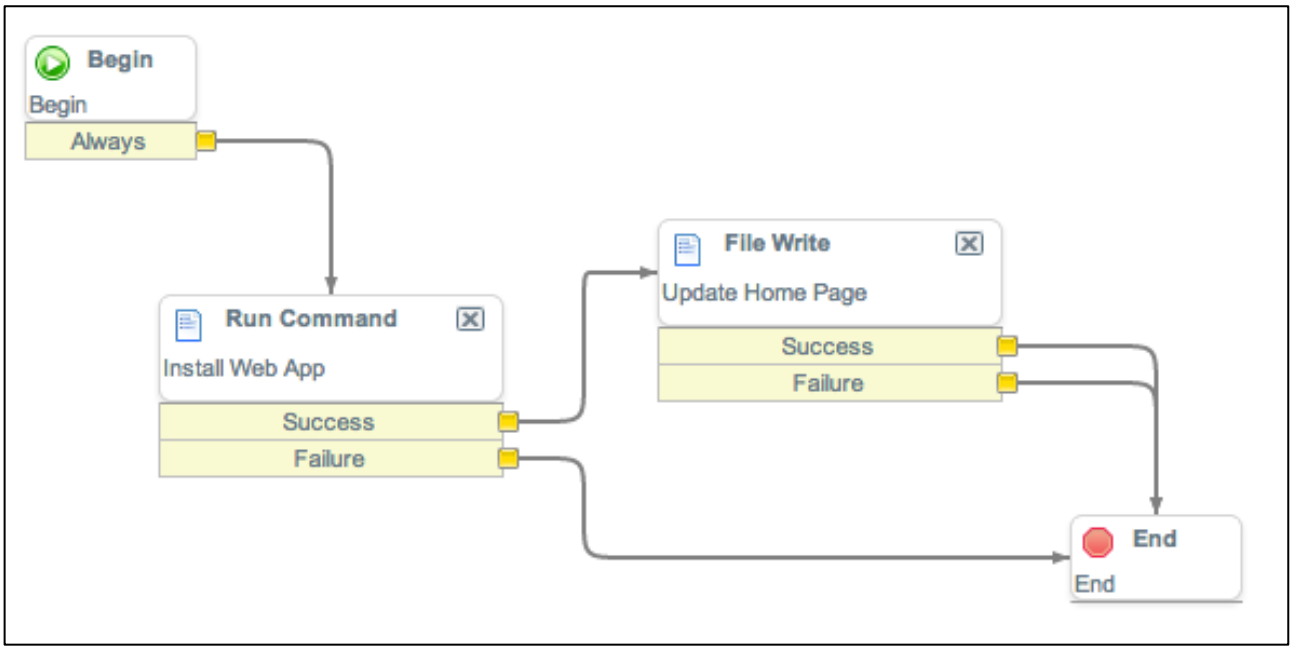
8. Click **Submit**.

©2014 ServiceNow, Inc. All rights reserved.

9. Make transitions from:

   Run Command Success to **File Write**
   File Write Success to **End**
   File Write Failure to **End**

10. Validate the workflow is connected and appears as shown.



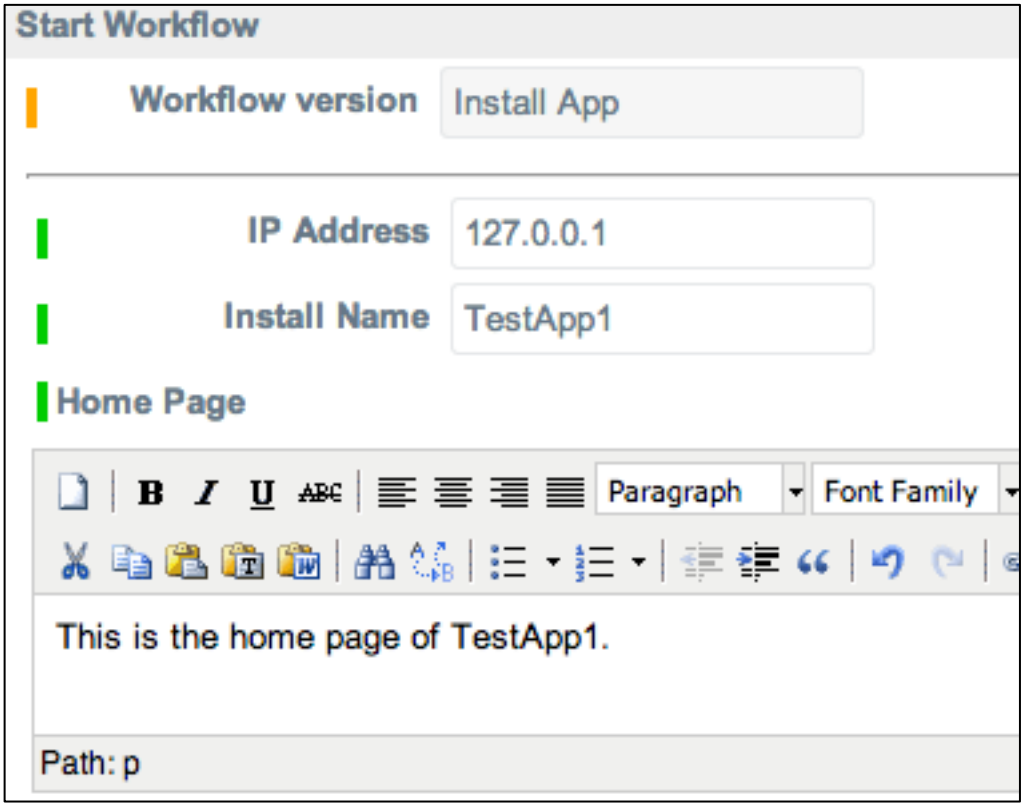
©2014 ServiceNow, Inc. All rights reserved.

Test Workflow

1. Click the green start arrow .

2. Enter the following:

   IP Address: **127.0.0.1**
   Install Name: **TestApp1**
   Home Page: **This is the home page of TestApp1**



3. Click **Submit**.

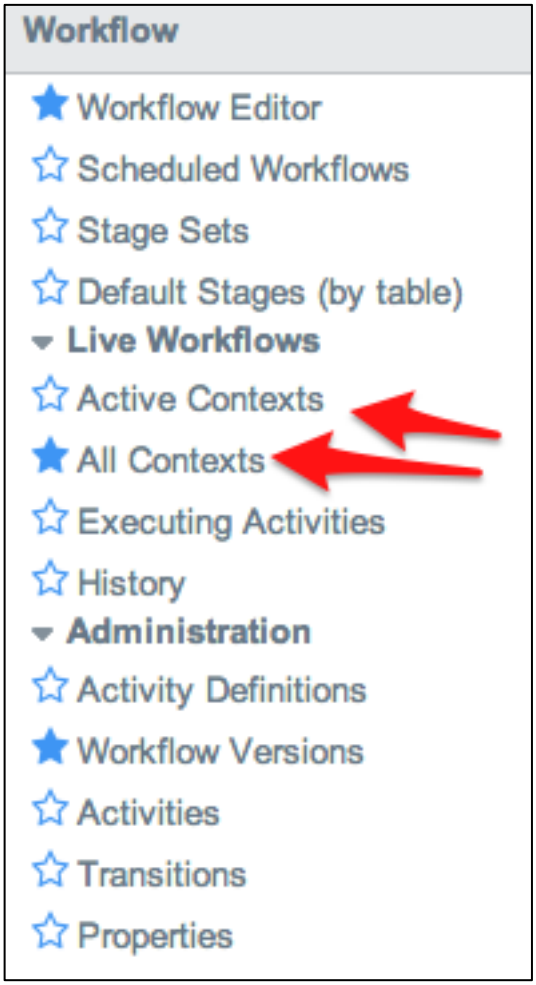
©2014 ServiceNow, Inc. All rights reserved.

Check Workflow Status and Getting Information About What Ran

There are multiple ways to observe the workflow status:

1. Refresh the visible workflow using the refresh button on the upper right . When the workflow has finished running, a transition will lead to End.

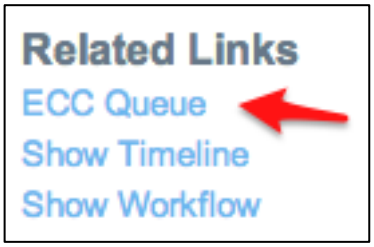
2. You can also check the workflow context. Navigate to **Workflow > Active Contexts**. While your workflow is still running, its context record will appear in this list. If it has already completed, you can find it in the **All Contexts** module. To get there, navigate to **Workflow > All Contexts.**



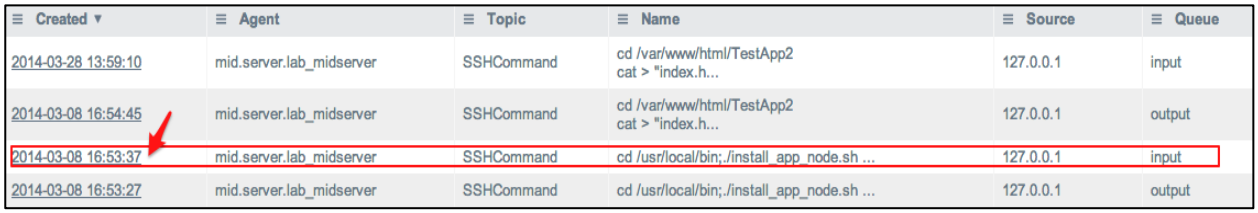
©2014 ServiceNow, Inc. All rights reserved.

3. Click into the context for the test run – the Workflow version is Install App.

4. Under **Related Links**, click the link for **ECC Queue**. The ECC Queue is the table that the MID Server and the ServiceNow instance communicate through. Output records contain instructions that the MID Server will execute. Input records contain the results of the instructions run.



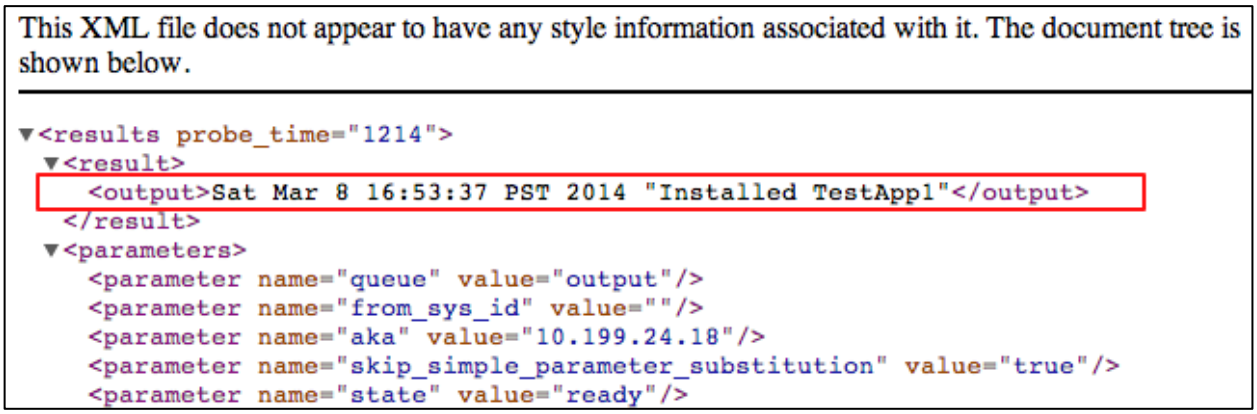
5. Select the input record from the viewpoint of the ECC Queue. The payload in ECC Queue input record tells the instance the result of execution.



Created	Agent	Topic	Name	Source	Queue
2014-03-28 13:59:10	mid.server.lab_midserver	SSHCommand	cd /var/www/html/TestApp2 cat > "index.h...	127.0.0.1	input
2014-03-08 16:54:45	mid.server.lab_midserver	SSHCommand	cd /var/www/html/TestApp2 cat > "index.h...	127.0.0.1	output
2014-03-08 16:53:37	mid.server.lab_midserver	SSHCommand	cd /usr/local/bin;./install_app_node.sh ...	127.0.0.1	input
2014-03-08 16:53:27	mid.server.lab_midserver	SSHCommand	cd /usr/local/bin;./install_app_node.sh ...	127.0.0.1	output

©2014 ServiceNow, Inc. All rights reserved.

6. For a formatted view of the payload, click the XML button next to the payload field.


This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<results probe_time="1214">
  ▼<result>
     <output>Sat Mar 8 16:53:37 PST 2014 "Installed TestAppl"</output>
   </result>
  ▼<parameters>
     <parameter name="queue" value="output"/>
     <parameter name="from_sys_id" value=""/>
     <parameter name="aka" value="10.199.24.18"/>
     <parameter name="skip_simple_parameter_substitution" value="true"/>
     <parameter name="state" value="ready"/>
```

The value of the output field contains what was written to **stdout** as a result of executing the command in the Run Command activity.

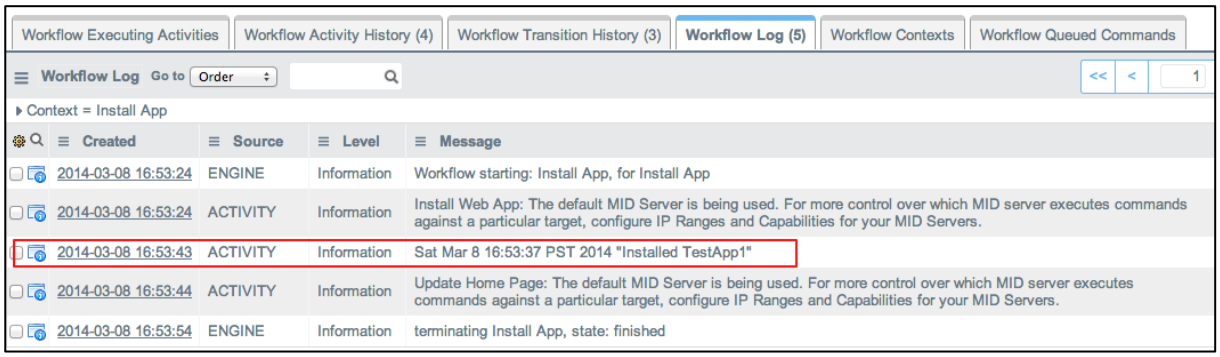
The content in this output field is stored in a variable called **activity.output** for use in the sensor script of the Run Command activity. Hence, the sensor script we wrote actually writes what was captured from **stdout** to the **workflow log**.

7. Go back to the workflow context (**Workflow > All Contexts** and find your context).

8. Look at the **Workflow Log** related list.

©2014 ServiceNow, Inc. All rights reserved.

9. The output from the command is visible in the log. You can click the **Toggle tabs on / off** button  on top right of the page to show tab view.



This is an info level message because we used **workflow.info** in the sensor script.
There is also a **workflow.error** to log error-level messages.
There is a **workflow.warn** to log warning-level messages.
There is a **workflow.debug** to log debug-level messages.
Logging messages of these levels are controlled by properties that can be configured by navigating to **Workflow > Properties**.

Visual Verification of Installation

The installation script we called installs a new web node to the Apache Server. Let's take a look at that page to validate it is successfully installed.

1. Open a new tab in the web browser.

2. Enter the URL for your lab instance using **http** instead of https and add "**:80/TestApp1**".
   seat1 is a placeholder for your instance number on the guide cover.
   For example: **http://orchestration201-seat1.lab.service-now.com:80/TestApp1**

3. You should see the home page display the message: **This is the home page of TestApp1**

©2014 ServiceNow, Inc. All rights reserved.

Lab Goal

Define REST Message to test the web application. Create REST Message activity and integrate into subflow to programmatically verify the installation.

Lab 2
Verify Web App Installation Programmatically

Defining the Verification Mechanism

Visually verifying the installation worked is nice, but we really want to do that automatically. You can do that using the ServiceNow REST web services capabilities.

1. Navigate to **System Web Services > REST Message**

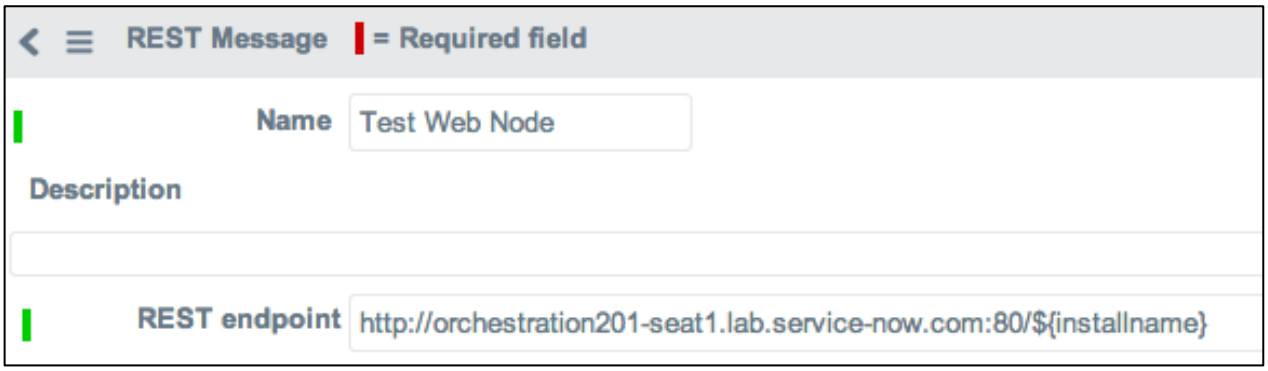
2. Click **New**.

©2014 ServiceNow, Inc. All rights reserved.

3. Our request is basic – all we really need is the endpoint. Enter the following:

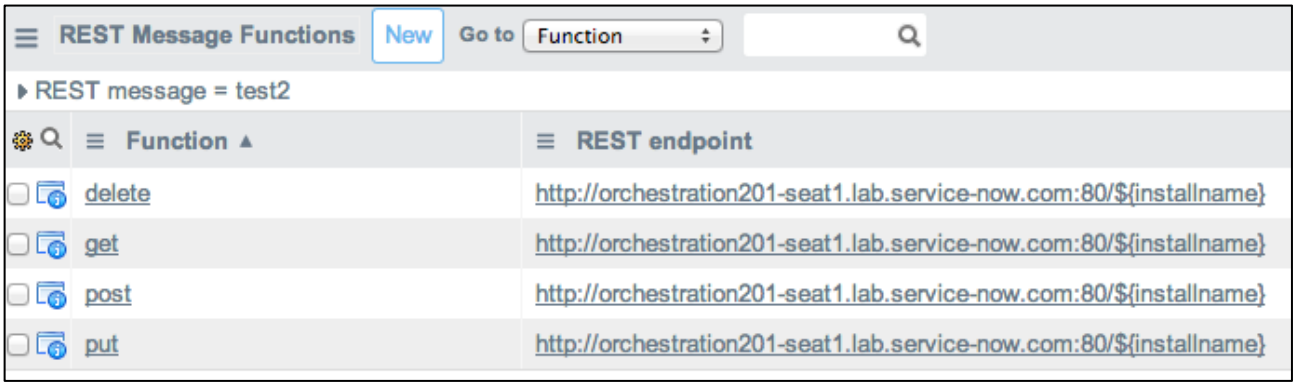
   Name: **Test Web Node**
   REST endpoint: **http://orchestration201-seat1.lab.service-now.com:80/${installname}**

   Note: Use **http**, not https. **seat1** is a placeholder for the instance number on your guide cover.



4. Click **Submit**.

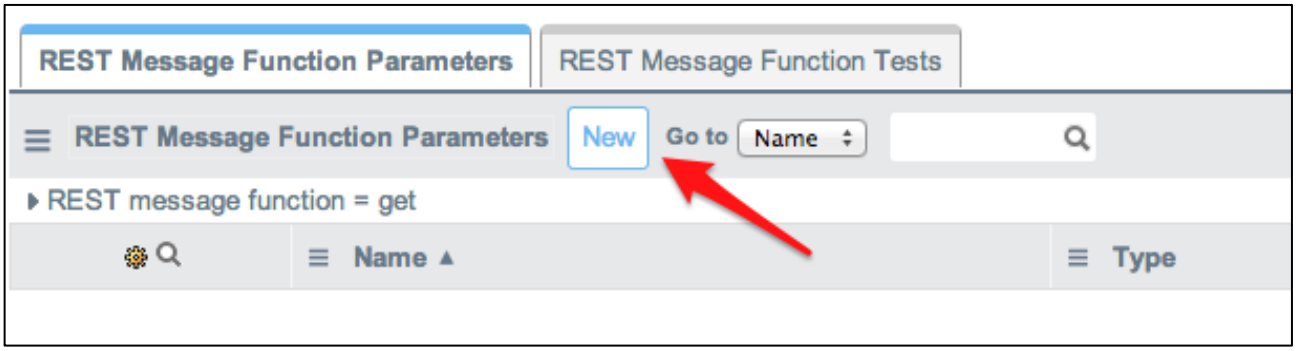
   Click on the newly created record. Note there are 4 REST Message Functions that have been automatically created.



5. Click **get**.

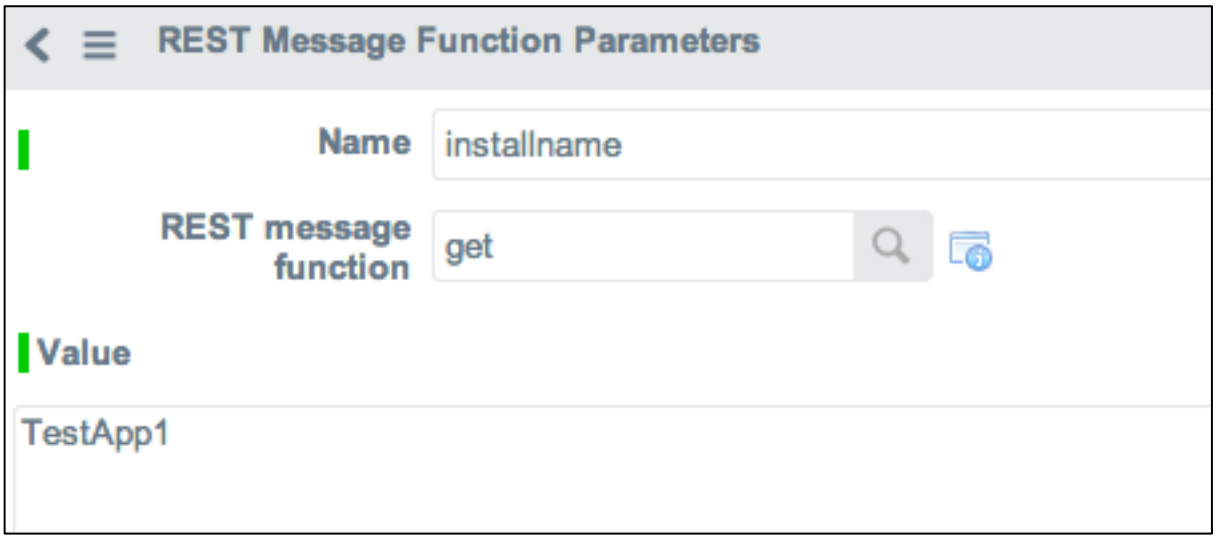
©2014 ServiceNow, Inc. All rights reserved.

6. On the related list **REST Message Function Parameters,** click **New**.



7. Enter the following:

   Name: **installname**
   Value: **TestApp1**



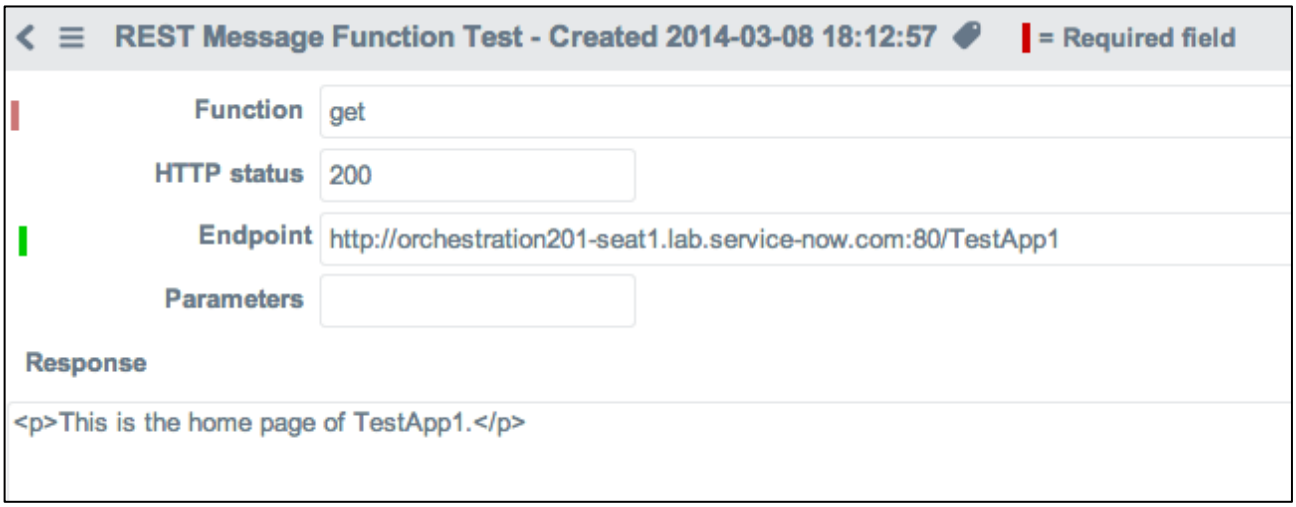
8. Click **Submit.**

©2014 ServiceNow, Inc. All rights reserved.

9. Under the **Related Links**, click **Test** UI Action.



You should get a result like the following:



What did this do?

We defined a **REST** Message that uses the get method to verify there is a webpage at that endpoint. Then we executed the method to run the test. The **${installname}** is a parameter in the **Endpoint** field that is substituted with the **Value** defined for testing this REST Message.

We see that the HTTP status returned is success (200), and the **Response** looks reasonable.

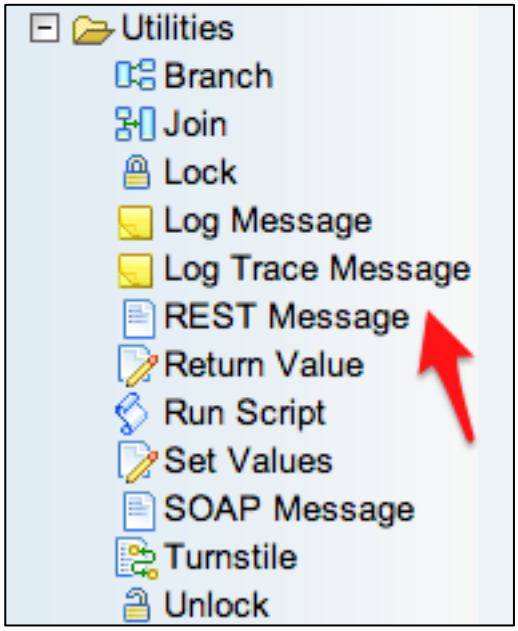
©2014 ServiceNow, Inc. All rights reserved.

Integrating the Verification Mechanism into the Subflow

We want the subflow to automatically perform this verification when the installation succeeded.

1. Go back to the **Workflow Editor**.

2. Expand the **Utilities** category on the right hand side of the editor.

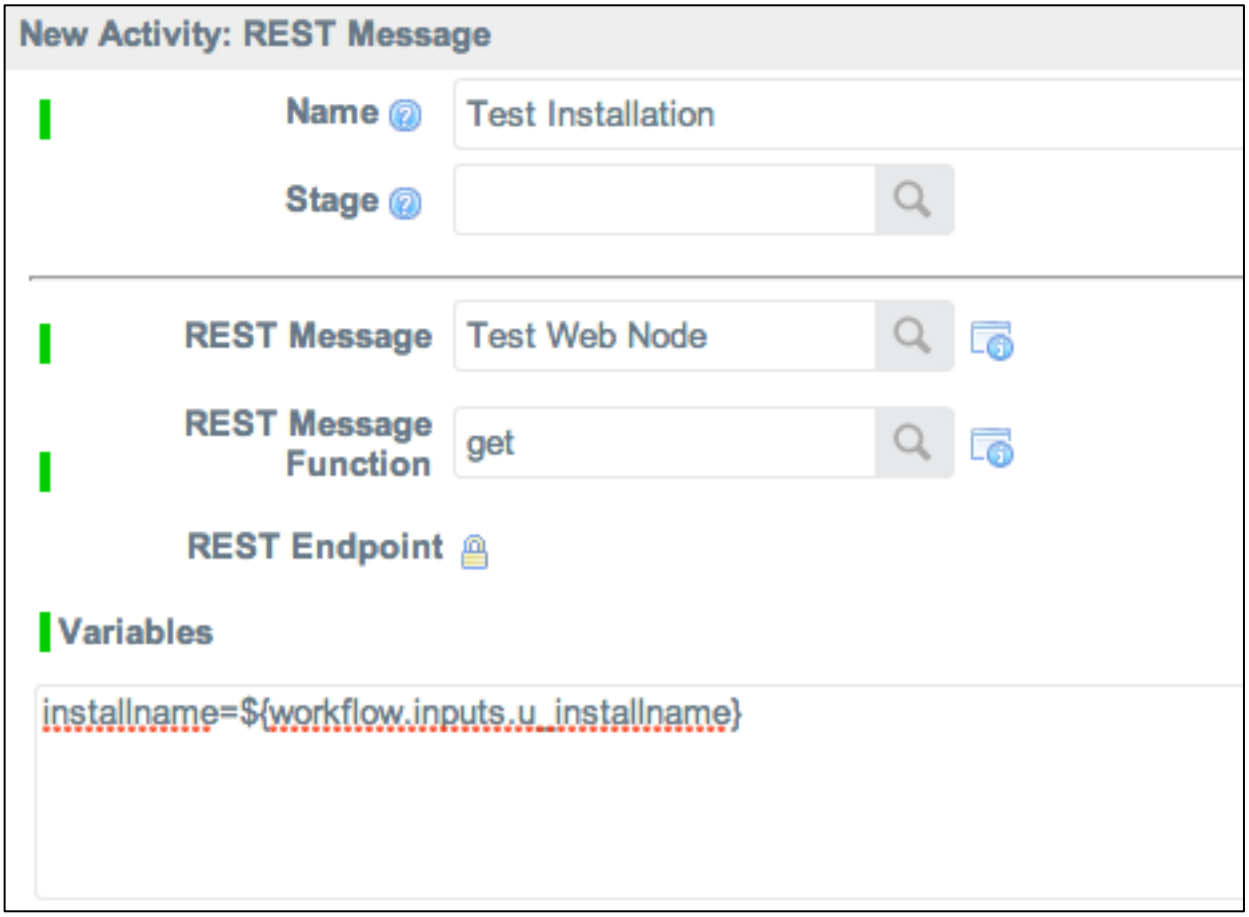


3. Drag a **REST Message** activity onto the transition between File Write's Success and End.

©2014 ServiceNow, Inc. All rights reserved.

4. Enter the following:

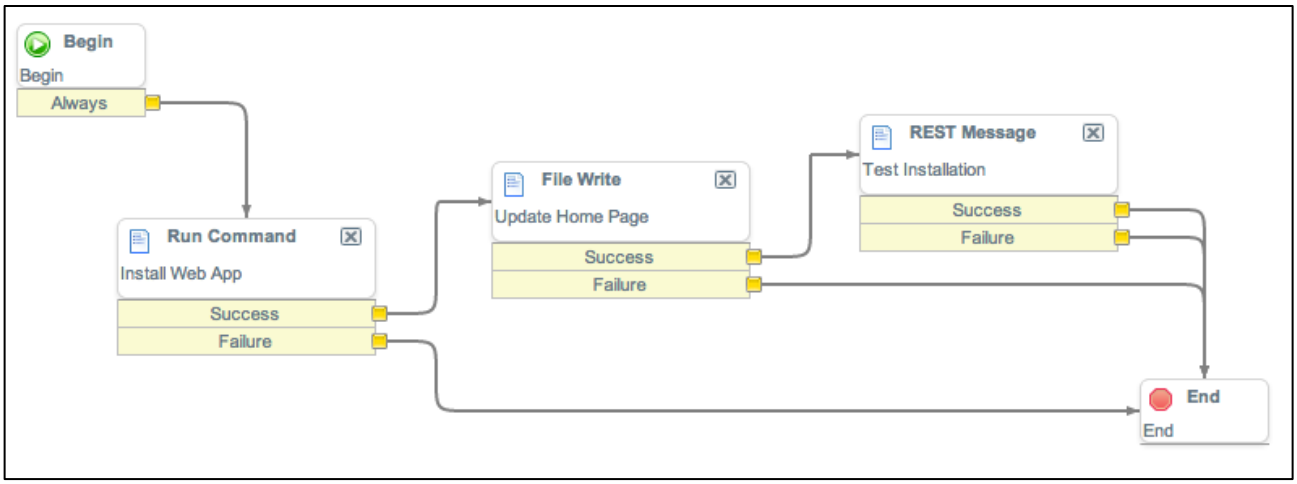
   Name: **Test Installation**
   REST Message: **Test Web Node**
   REST Message Function: **get**
   REST Endpoint: leave it alone – defaults to the endpoint on the REST Message Function
   Variables: **installname=${workflow.inputs.u_installname}**



5. Click **Submit**.

©2014 ServiceNow, Inc. All rights reserved.

6. Create a transition from both the **Success** and **Failure** conditions of the **REST Message** activity to **End**.



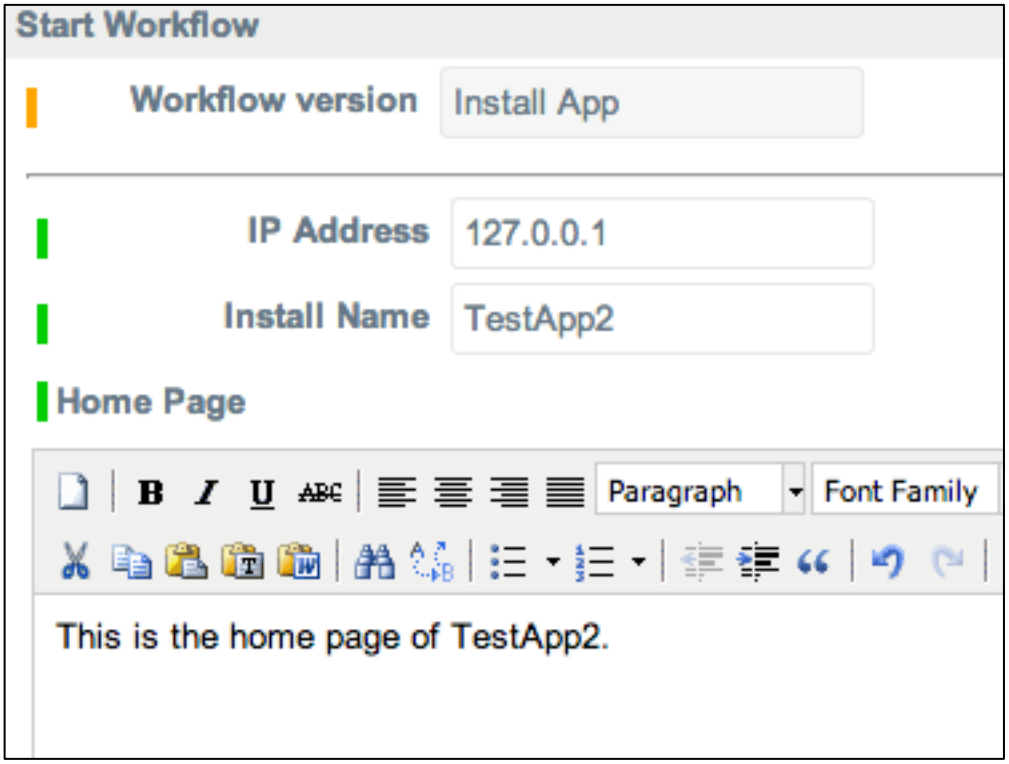
®2014 ServiceNow, Inc. All rights reserved.

Test the Updated Workflow

1. Click the green start arrow .

2. Enter the following:

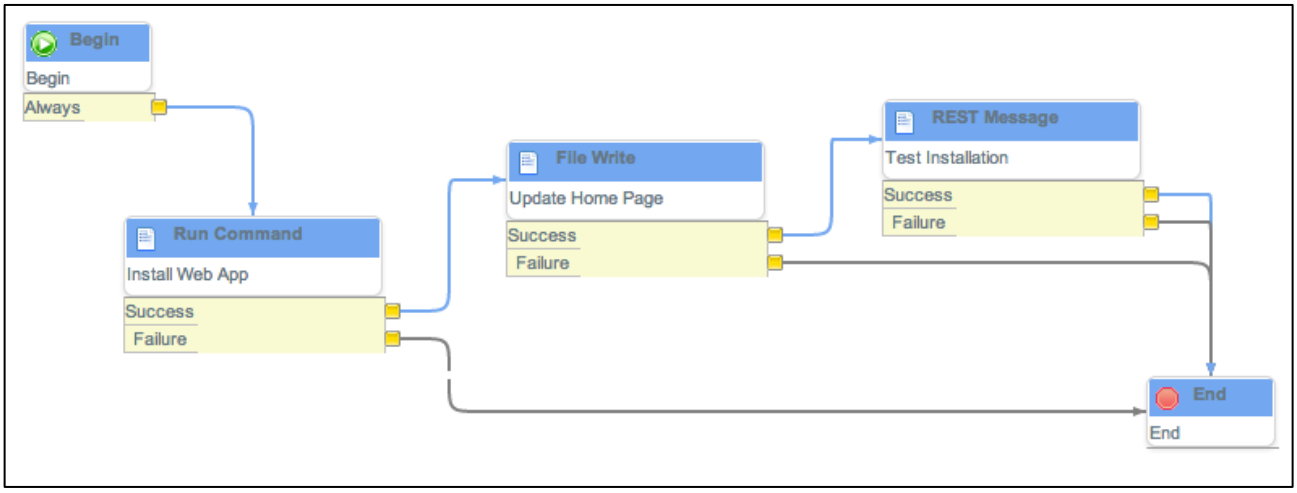
   IP Address: **127.0.0.1**
   Install Name: **TestApp2**
   Home Page: **This is the home page of TestApp2**



3. Click **Submit**.

©2014 ServiceNow, Inc. All rights reserved.

4. **Refresh** until the workflow completes. It should follow success transitions to End.



®2014 ServiceNow, Inc. All rights reserved.

Lab Goal

Handle installation failure scenario. Use File Read activity to grab the contents of log file and save the information to workflow log. Return the execution status of the subflow to the parent workflow.

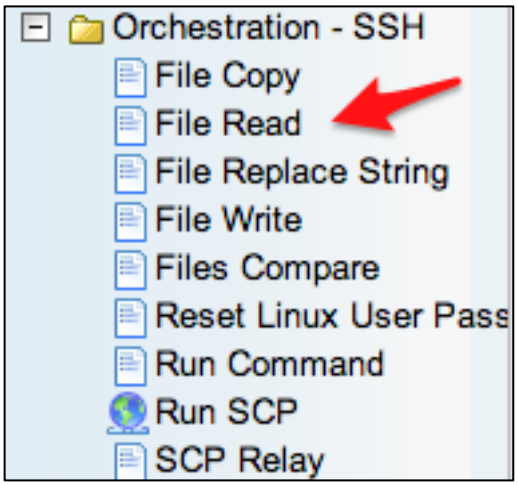
Lab 3
Handle Installation Failure

Grab Contents of a Log File

If the installation fails, we would like to retrieve contents of the log file generated by the install script. How do we do that? We have an activity to do exactly that – **File Read**! Let's use it to pull the contents of the file, and write the contents of the file to the workflow log.

1. Return to the **Workflow Editor**.

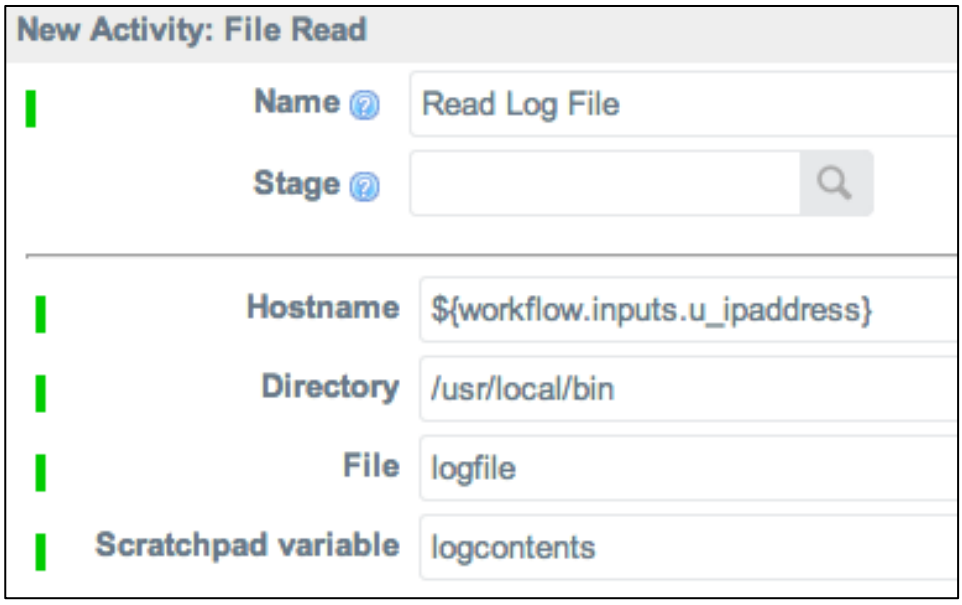
2. Drag a **File Read** activity from the **Orchestration – SSH** category on the right side of the editor onto the transition between the Run Command's Failure condition and End.



©2014 ServiceNow, Inc. All rights reserved.

3. Enter the following:

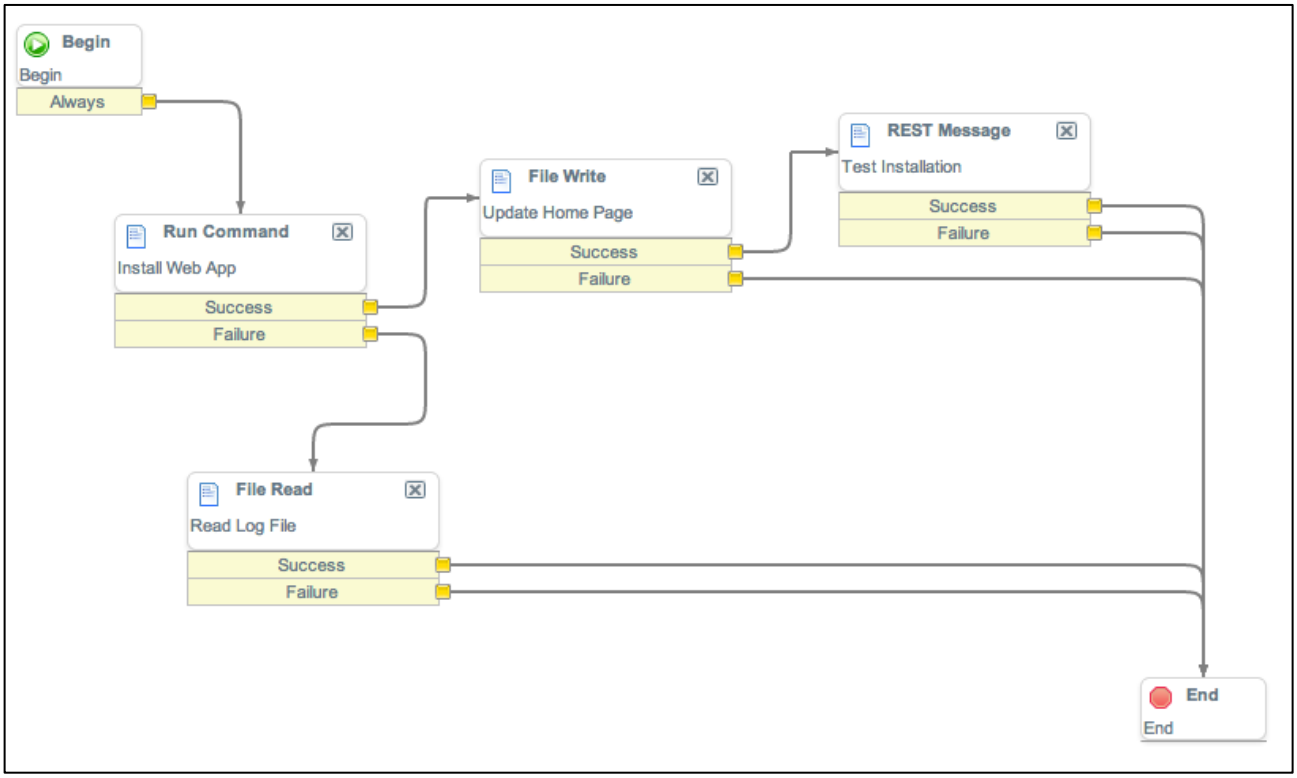
   Name: **Read Log File**
   Hostname: **${workflow.inputs.u_ipaddress}**
   Directory: **/usr/local/bin**
   File: **logfile**
   Scratchpad Variable: **logcontents**



4. Click **Submit**.

©2014 ServiceNow, Inc. All rights reserved.

5. Create transitions from the **Success** and **Failure** conditions of the **File Read** activity to **End**.

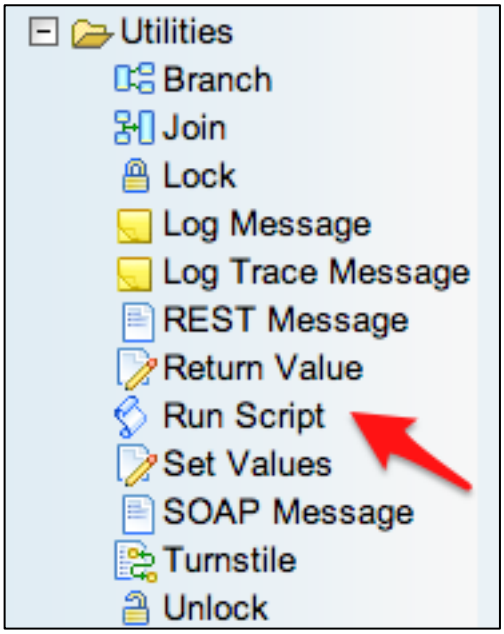


®2014 ServiceNow, Inc. All rights reserved.

Put Contents of a Log File in the Workflow Log

Where did the file content go? In the variables for File Read, we gave it the scratchpad variable **logcontents**. This means the content should be stored in the workflow scratchpad, accessible in script as **workflow.scratchpad.logcontent**. Let's put this content in the workflow log.

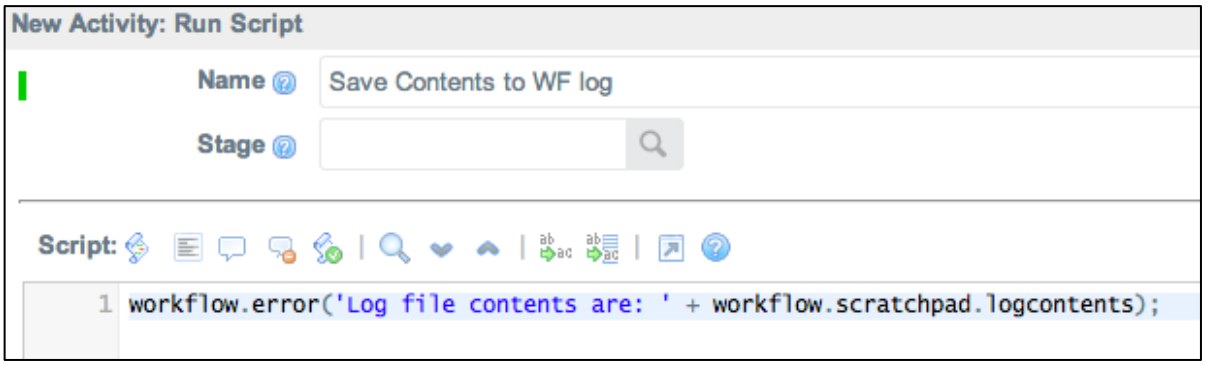
1. Drag a **Run Script** activity from the **Utilities** category onto the transition between the File Read's Success condition and End.



©2014 ServiceNow, Inc. All rights reserved.

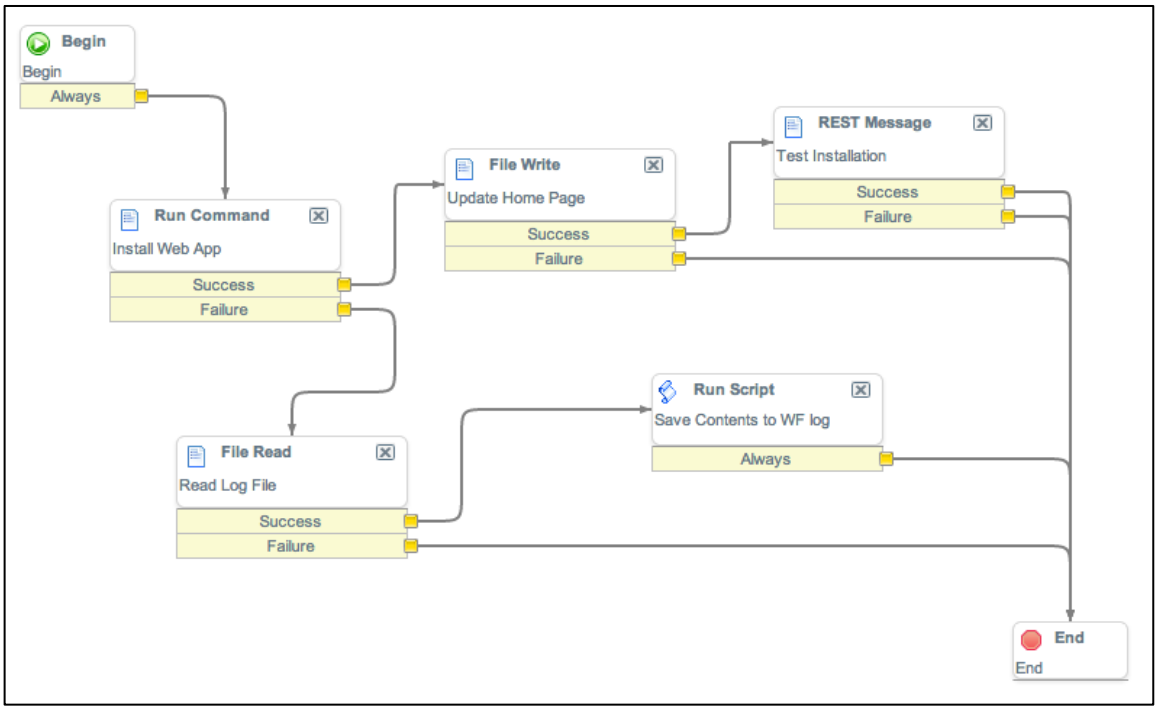
2. Enter the following:

   Name: **Save Contents to WF Log**
   Script: **workflow.error('Log file contents are: ' + workflow.scratchpad.logcontents);**



3. Click **Submit**.

4. Validate the workflow is connected and appears as shown.



©2014 ServiceNow, Inc. All rights reserved.

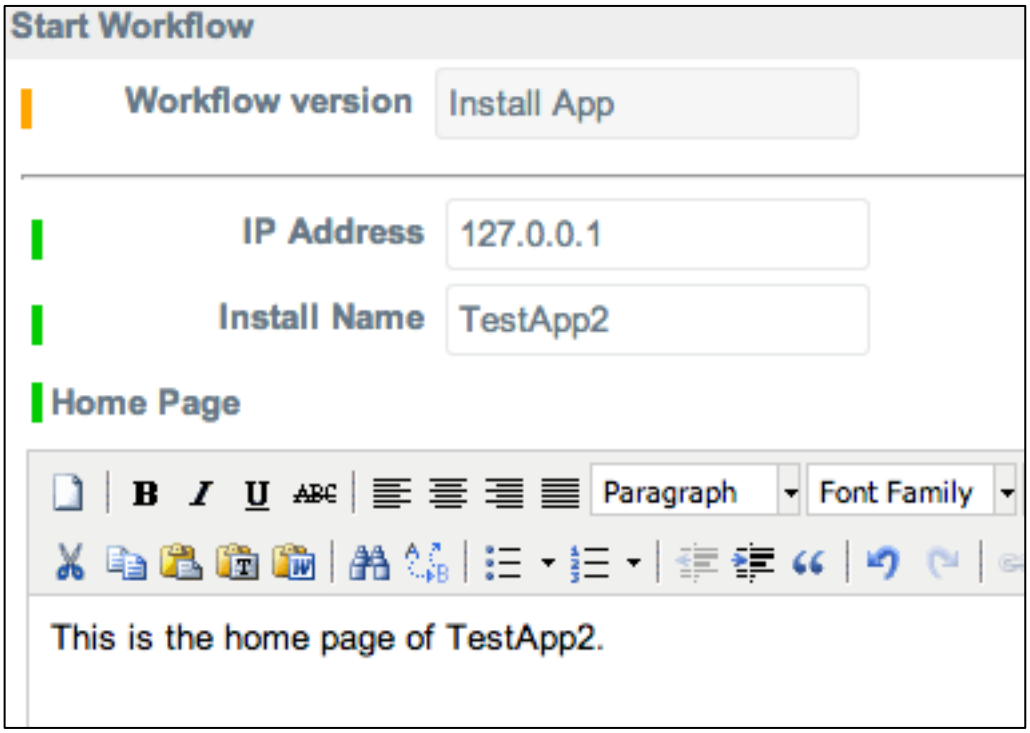
Test Updated Workflow (make it fail!)

We'll enter an install name we already entered, and expect a failure.

1. Click the green start arrow .

2. Enter the following:

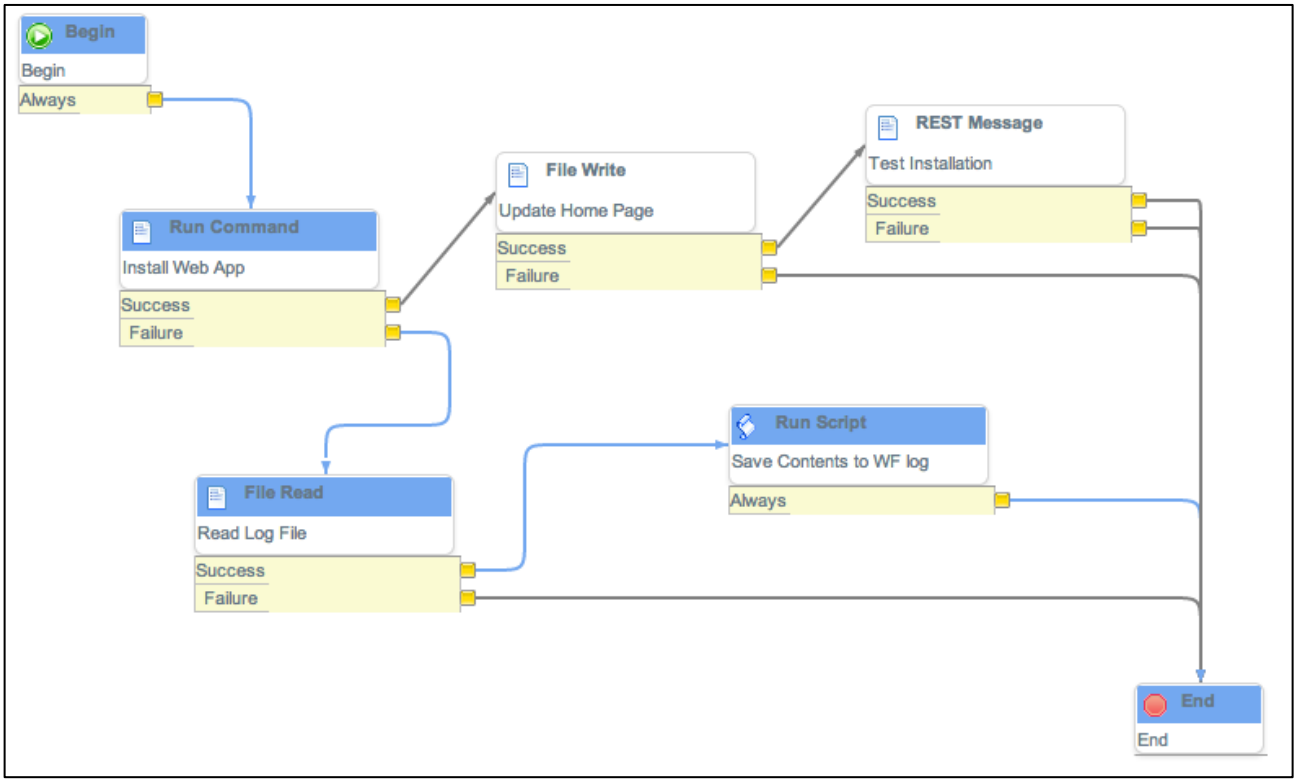
   IP Address: **127.0.0.1**
   Install Name: **TestApp2**
   Home Page: **This is the home page of TestApp2**



3. Click **Submit**.

©2014 ServiceNow, Inc. All rights reserved.

4. Refresh periodically until the workflow completes.

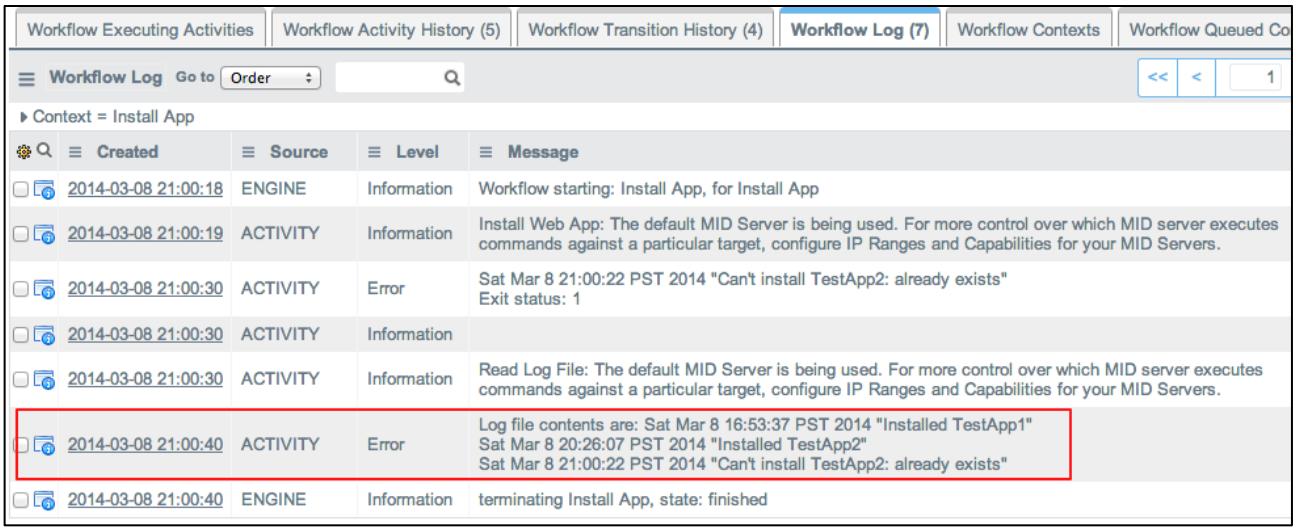


The Install Web App Run Command should fail, the File Read activity should succeed.

5. Examine the workflow context. Navigate to **Workflow > All Contexts** and open the most recent record.

©2014 ServiceNow, Inc. All rights reserved.

6. Look at the **Workflow Log** related list.
   The contents of the logfile are written in the log from the Run Script activity.



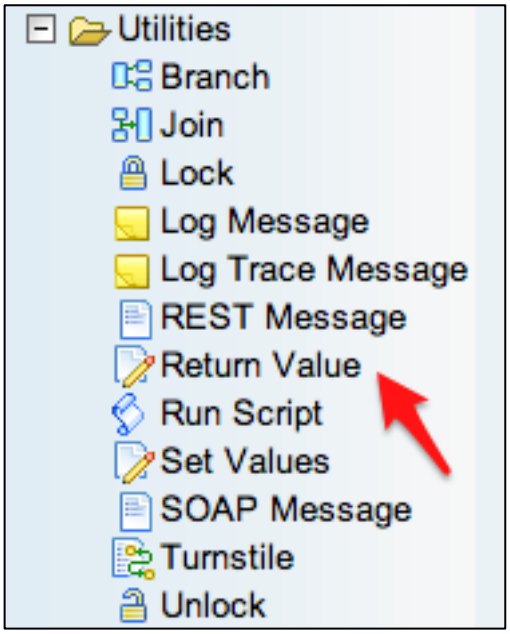
This is an error level message because we use workflow.error in the Run Script activity.

©2014 ServiceNow, Inc. All rights reserved.

Tell the Parent Workflow What Happened

So far, this is fine as a standalone workflow, but we plan to embed this as a subflow within a parent workflow. That parent workflow would want the subflow's status. For simplicity, we're just going to return the string **success** when it works, and **failure** when it does not.

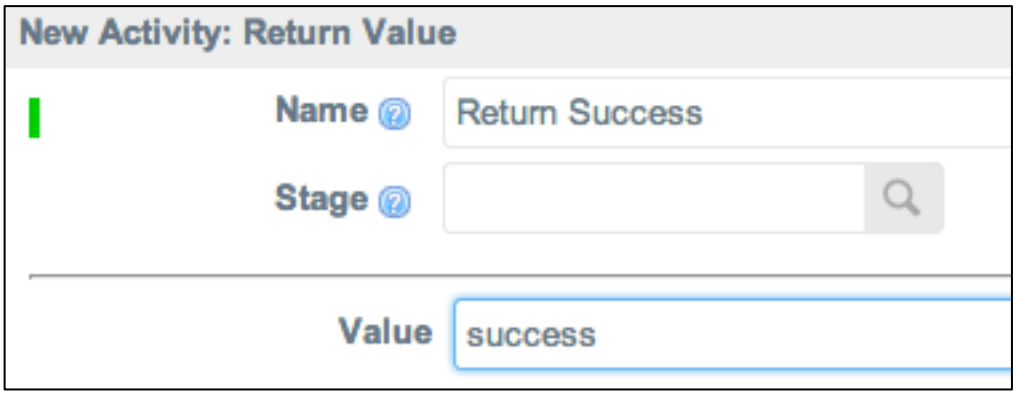
1. Back in the workflow editor, drag a **Return Value** activity from the Utilities category onto the canvas.



®2014 ServiceNow, Inc. All rights reserved.

2. Enter the following:

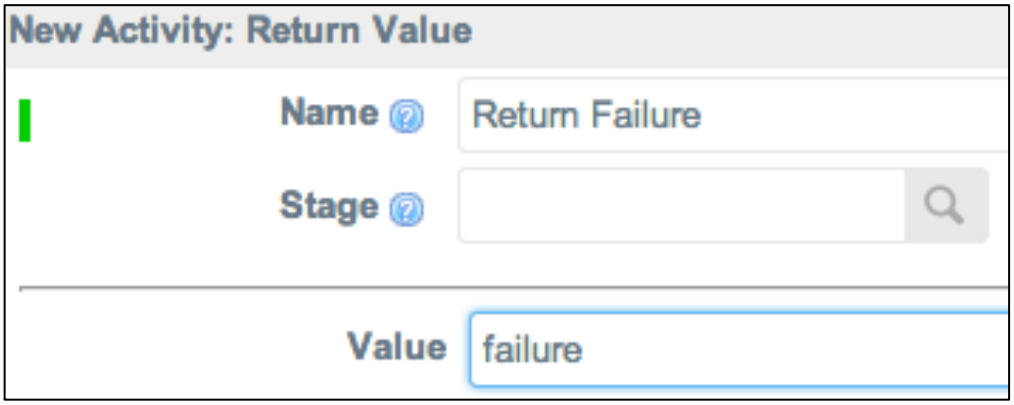
   Name: **Return Success**
   Value: **success**



3. Click **Submit.**

4. Repeat Steps 1 to 3, entering the following:

   Name: **Return Failure**
   Value: **failure**

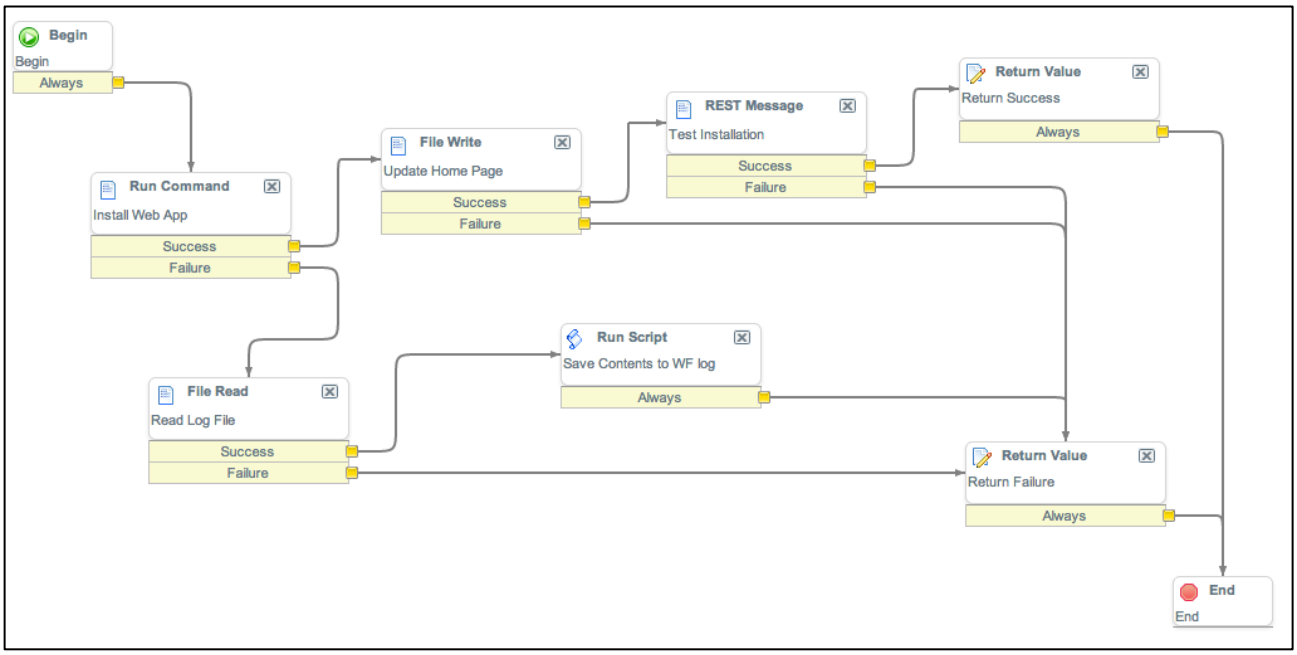


©2014 ServiceNow, Inc. All rights reserved.

5. Update the transitions as follows:

   From **REST Message Success** update transition to **Return Value Success**.
   From **REST Message Failure** update transition to **Return Value Failure**.
   From **File Write Failure** update transition to **Return Value Failure**.
   From **File Read Failure** update transition to **Return Value Failure**.
   From **Run Script Always** update transition to **Return Value Failure**.
   From **Return Value Success Always** update transition to **End**.
   From **Return Value Failure Always** update transition to **End**.

6. Validate the workflow is connected and appears as shown.

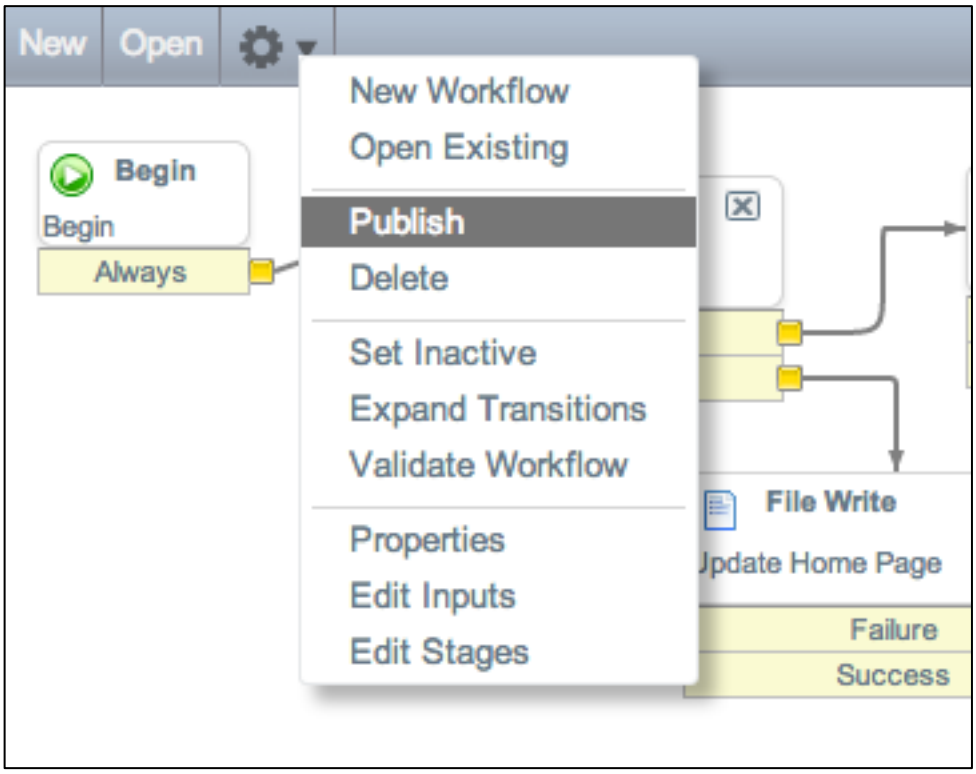


©2014 ServiceNow, Inc. All rights reserved.

Publish the Workflow

The subflow should be published before it can be used in the parent workflow. To publish the workflow:

1. Click the gear menu and select **Publish**.



©2014 ServiceNow, Inc. All rights reserved.

Lab Goal

Integrate the subflow to parent workflow to replace manual tasks and test the entire workflow from begin to end.

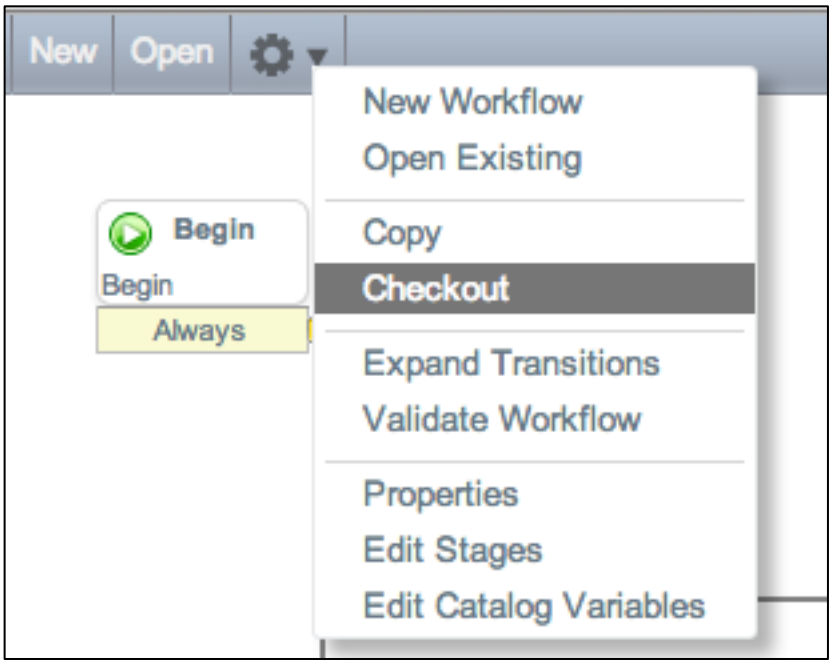
Lab 4 Integrate Subflow to Parent Workflow

Replace Manual Tasks in Parent Workflow with Subflow

Your instance has a pre-built workflow that contains approvals, license checks, and manual tasks to install the application. We looked at this workflow earlier. This is the parent workflow for the subflow we have built. Let's integrate the subflow into the parent workflow, replacing the manual tasks.

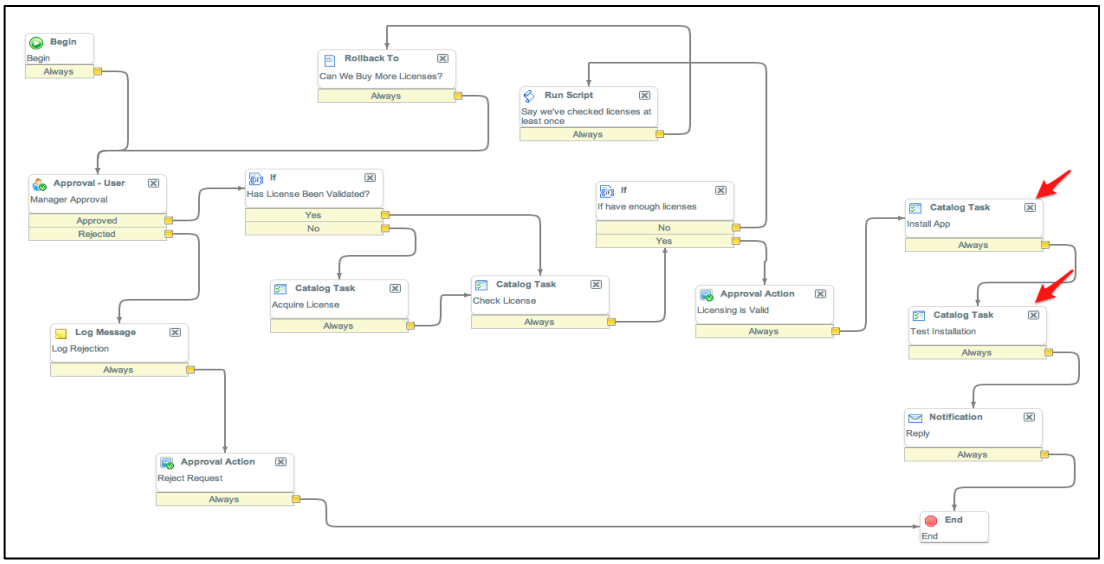
1. In the **Workflow Editor**, click **Open** and select **K14 - Order Application**.

2. **Checkout** the **K14 - Order Application** workflow.

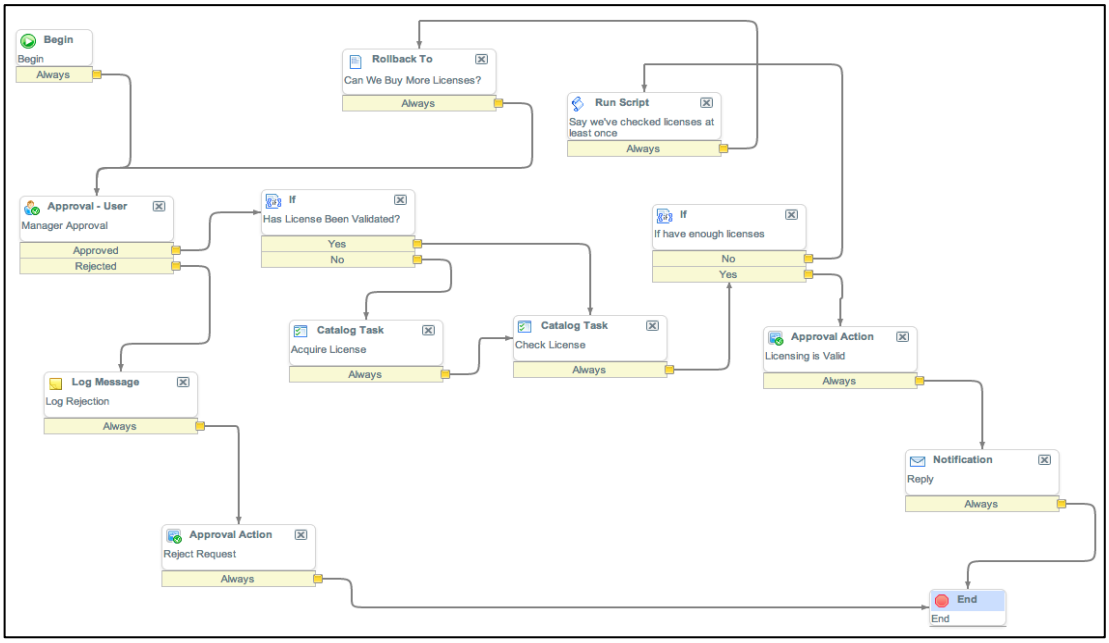


©2014 ServiceNow, Inc. All rights reserved.

3. Delete **Install App** and **Test Installation** Catalog Tasks by clicking X button on Catalog Task activity. We will replace these 2 manual tasks by the Install App workflow we created.

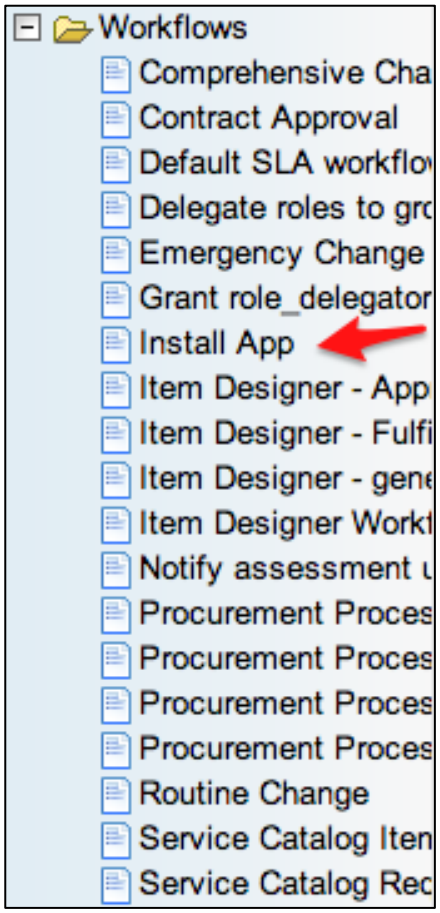


4. Validate the workflow appears as shown after deleting the 2 Catalog Tasks.



©2014 ServiceNow, Inc. All rights reserved.

5. Expand the **Workflows** category on the right side of the Workflow Editor. This shows all workflows in the system. You can drop a workflow into your current workflow as a subflow simply by treating one of these workflows in the list as an activity.



6. Drag the **Install App** workflow onto workflow canvas.

7. Enter the following:

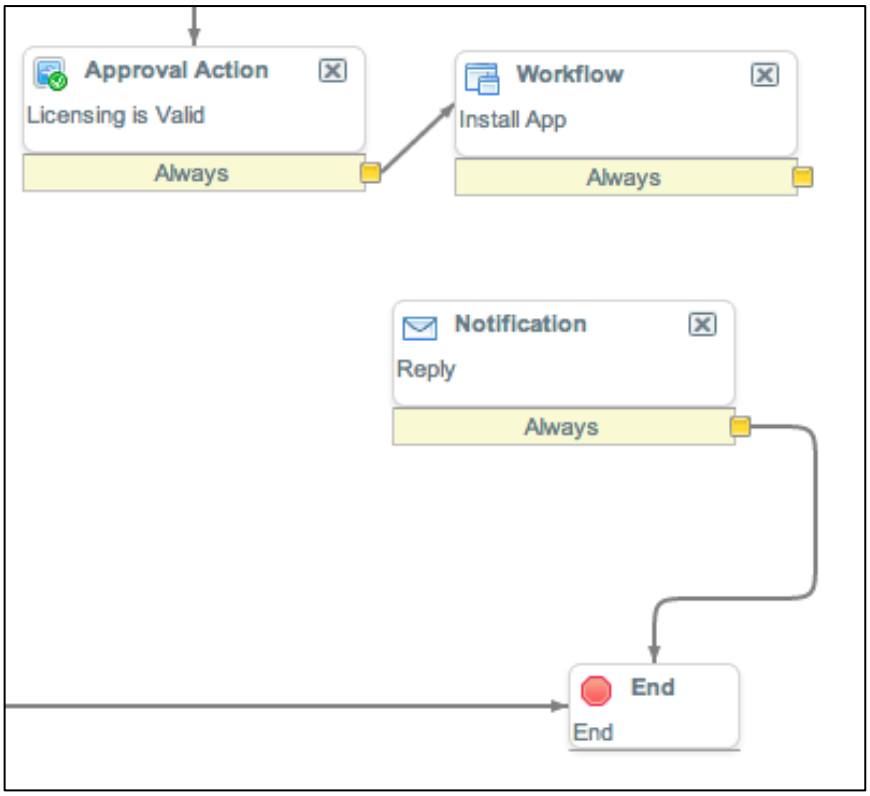
   Map return value to: **workflow.scratchpad.result**
   IP Address: **${current.variables.target_server.ip_address}**
   Install Name: **${current.variables.appname}**
   Home page: **${current.variables.home_page.web_page}**

©2014 ServiceNow, Inc. All rights reserved.

8. Click **Submit**.

9. Create a transition from **Always** in **Approval Action** to the new **Workflow** activity.

10. Validate that this part of the workflow appears as shown.



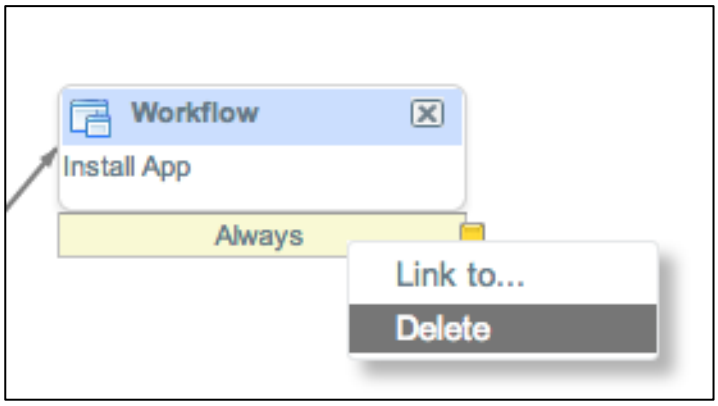
©2014 ServiceNow, Inc. All rights reserved.

Check Subflow Results

This workflow currently always sends a notification to the requesting user after the subflow is called, regardless of whether the installation succeeded or failed. The subflow we created returns success or failure. We should check that value and, on success, proceed to notify the user that their application has been successfully installed. On failure, we want to assign a task to someone to investigate and fix the problem before sending the notification of success. Let's modify the workflow to do this.

1. Right click on the **Always** condition of the Workflow (subflow) activity.

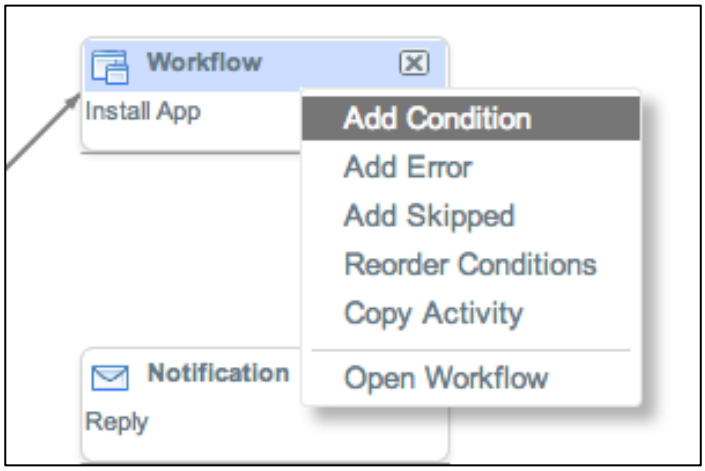
2. Click **Delete** on the context menu that appears.



3. Right-click on the body of the Workflow activity.

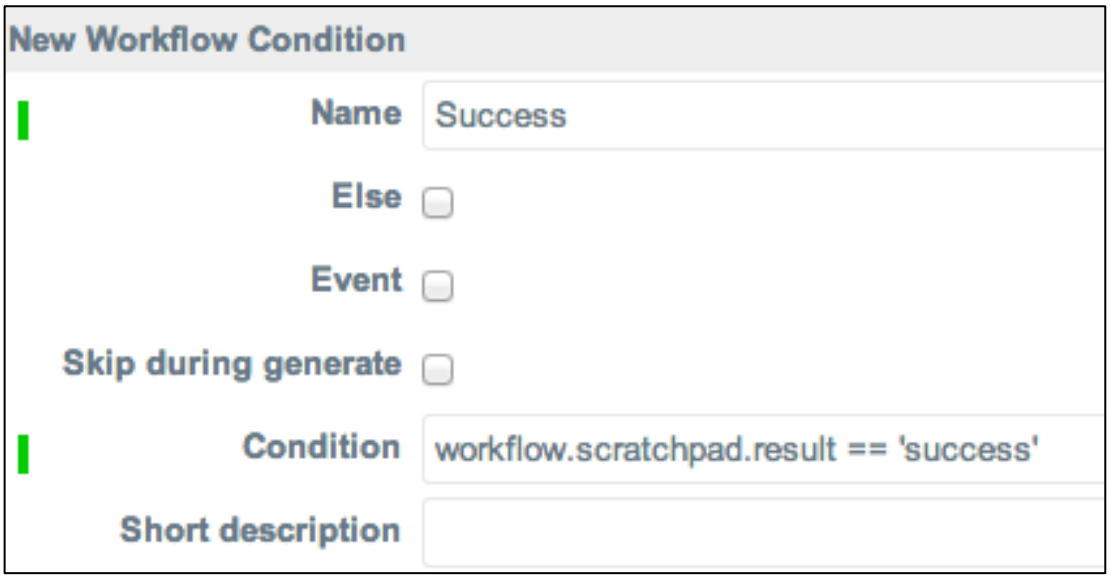
©2014 ServiceNow, Inc. All rights reserved.

4. Select **Add Condition**.



5. Enter the following:

   Name: **Success**
   Condition: **workflow.scratchpad.result == 'success'**

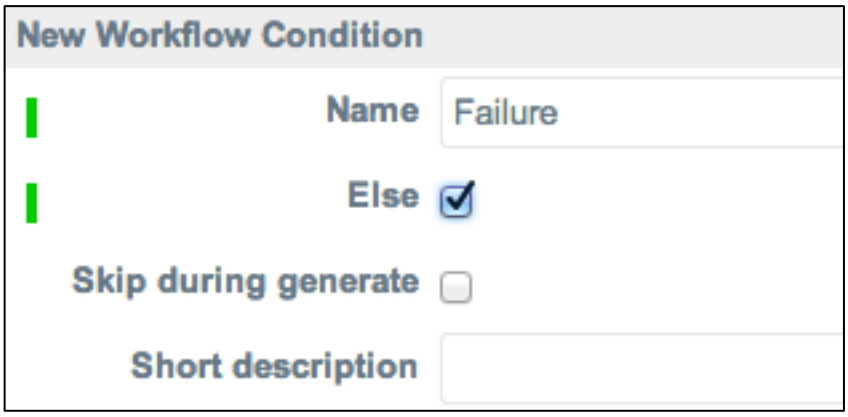


6. Click **Submit**.

©2014 ServiceNow, Inc. All rights reserved.

7. Repeat Steps 3-6, entering the following:

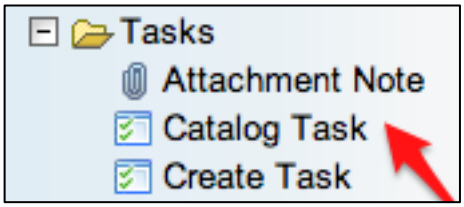
   Name: **Failure**
   Check the **Else** box



8. Create a transition from the **Success** condition on the **Workflow** activity to the **Notification** activity.

9. Add the **Catalog Task** activity on failure. Start by expanding the **Tasks** category on the right side of the workflow canvas.

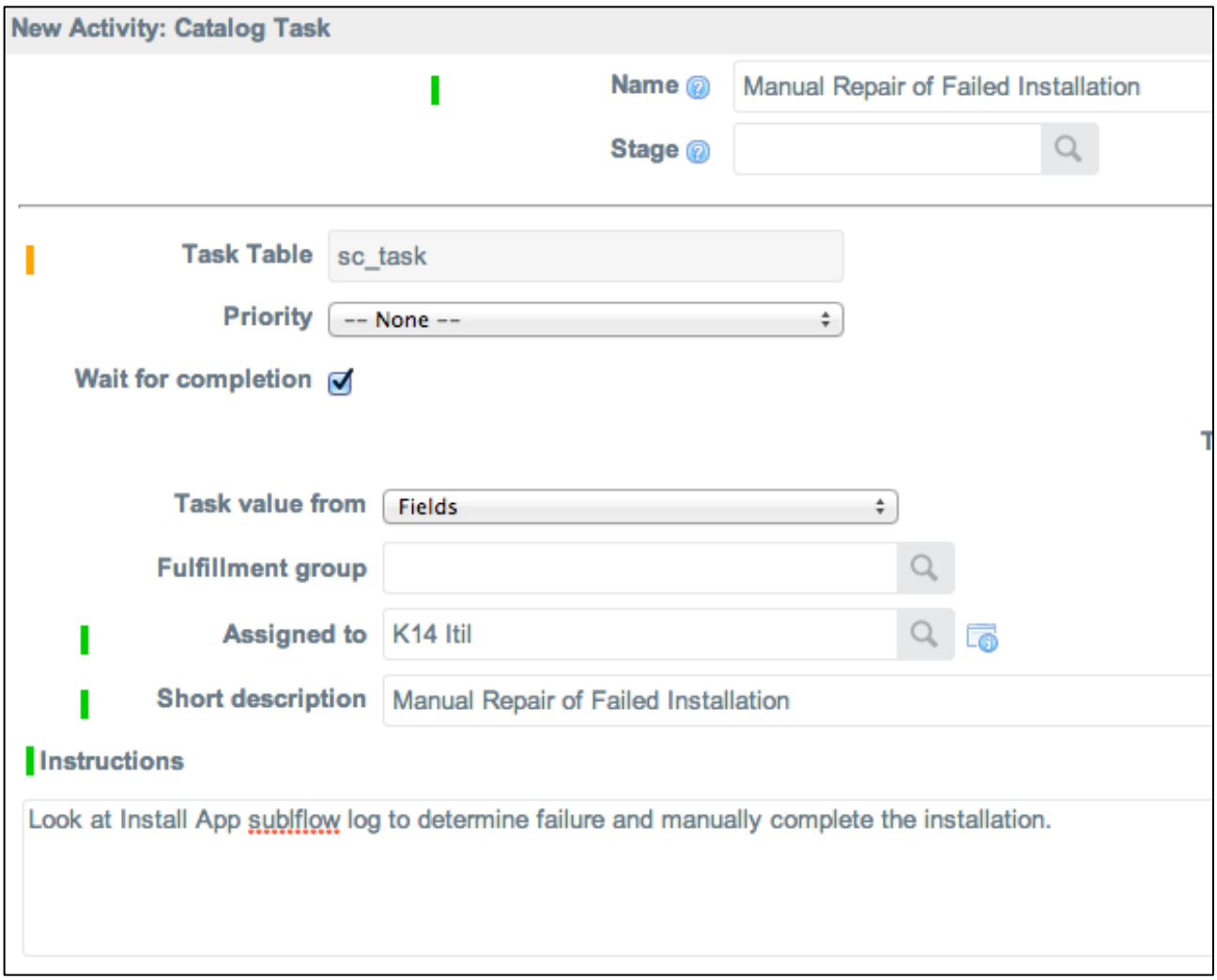
10. Drag a **Catalog Task** onto the canvas.



©2014 ServiceNow, Inc. All rights reserved.

11. Enter the following:

    Name: **Manual repair of failed install**
    Assigned to: **K14 Itil**
    Short description: **Manual repair of failed install**
    Instructions: **Look at Install App subflow log to determine failure and manually complete the installation.**



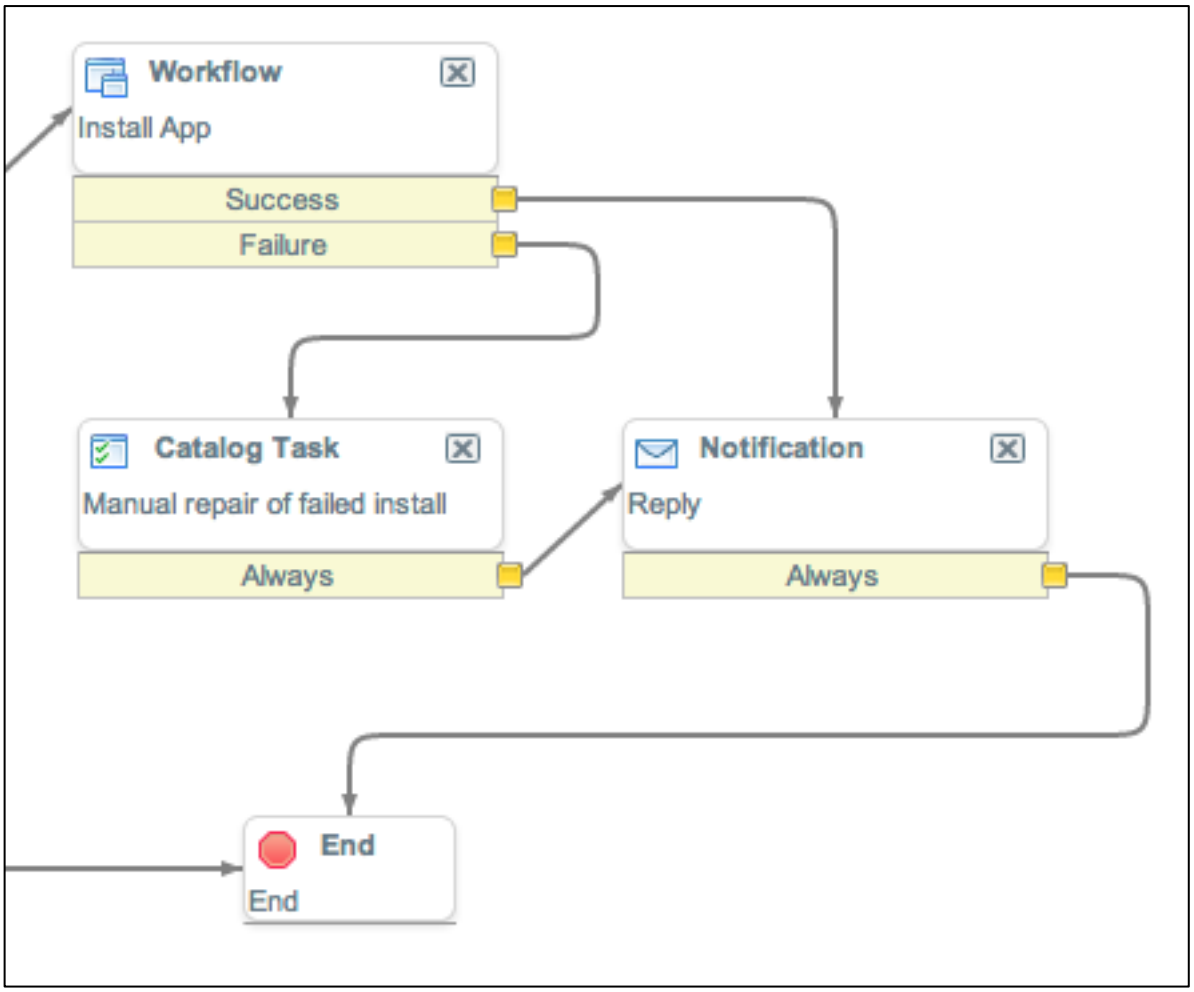
12. Click **Submit**.

©2014 ServiceNow, Inc. All rights reserved.

13. Create a transition from **Failure** on the **Workflow** activity to the new **Catalog Task**.

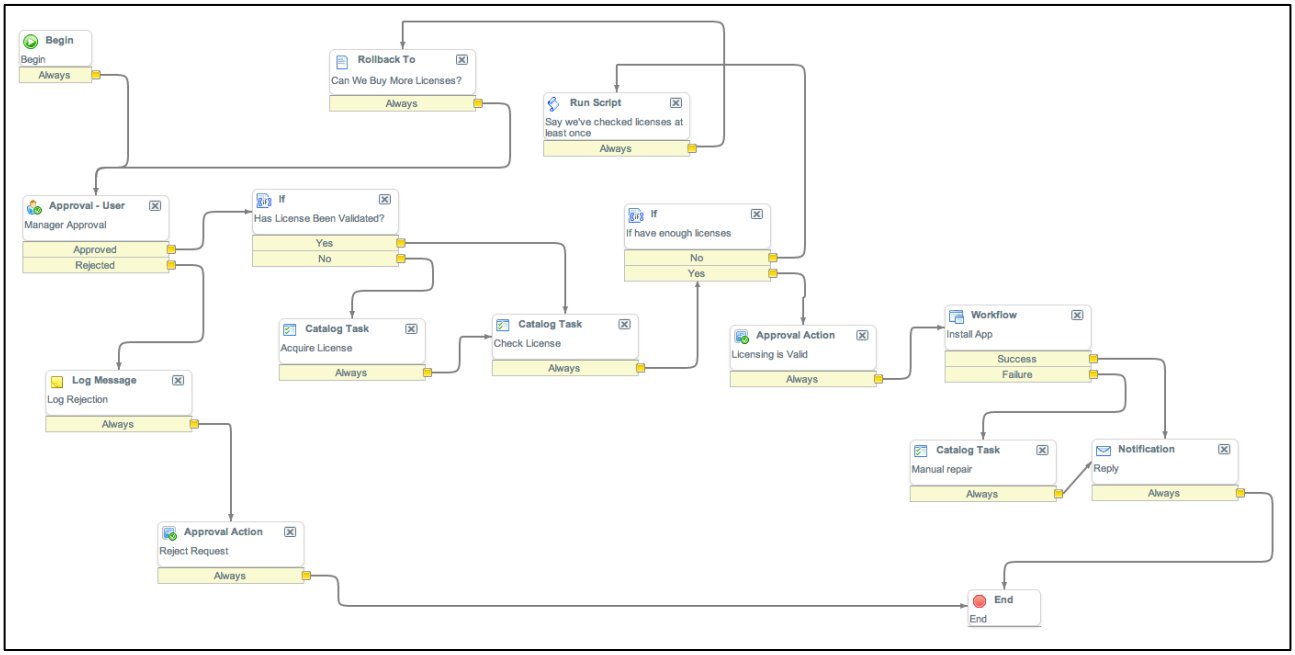
14. Create a transition from the **Always** of the new **Catalog Task** to **Notification**.

15. Validate this part of the workflow is connected and appears as shown.



©2014 ServiceNow, Inc. All rights reserved.

16. Validate **K14 – Order Application** workflow appears as shown.



17. Click the gear menu and select **Publish** to publish the workflow.

®2014 ServiceNow, Inc. All rights reserved.

Test the Entire Flow

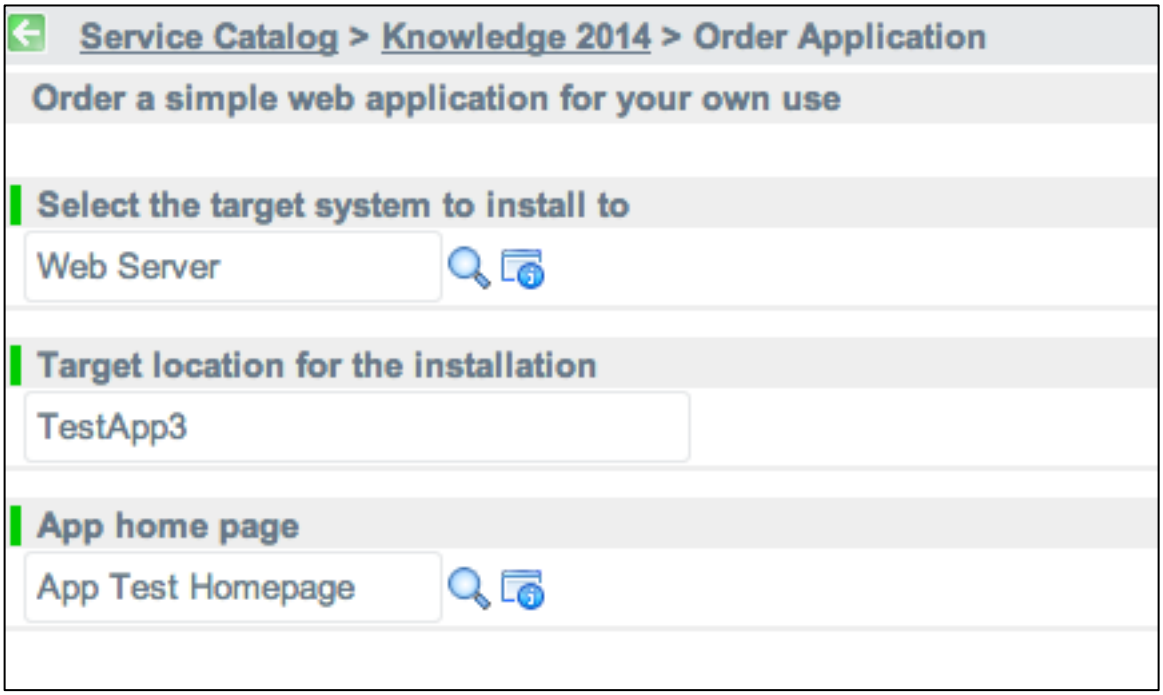
Let's test the entire end-to-end scenario.

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

2. Click on **Order Application** under the **Knowledge 2014** category.

3. Enter or select the following:

   Target system: **Web Server**
   Target location: **TestApp3**
   App home page: **App Test Homepage**



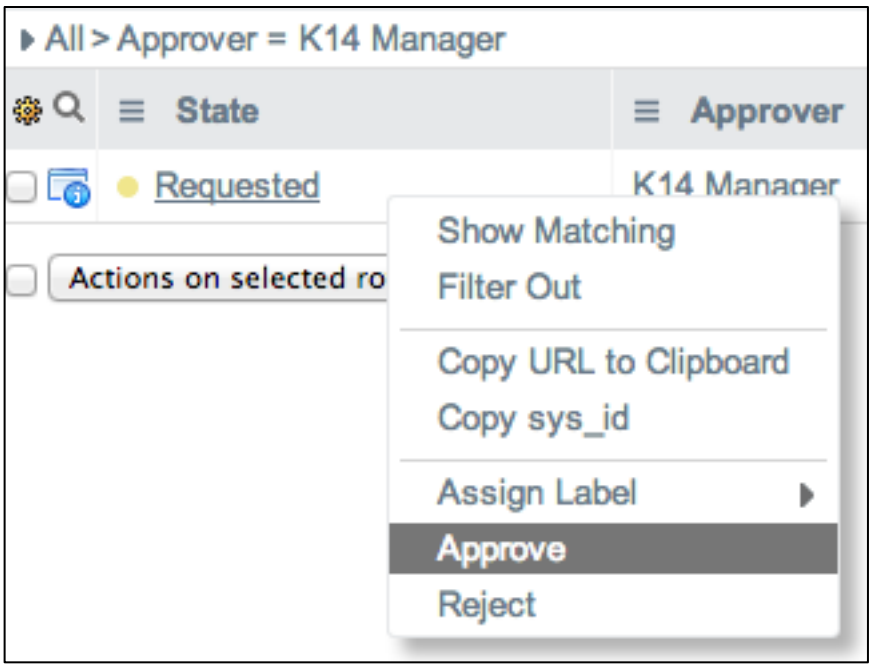
4. Click **Order Now**.

5. Impersonate **K14 Manager**.

©2014 ServiceNow, Inc. All rights reserved.

6. Navigate to **Service Desk > My Approvals**.

7. Right click on the requested item and click Approve.



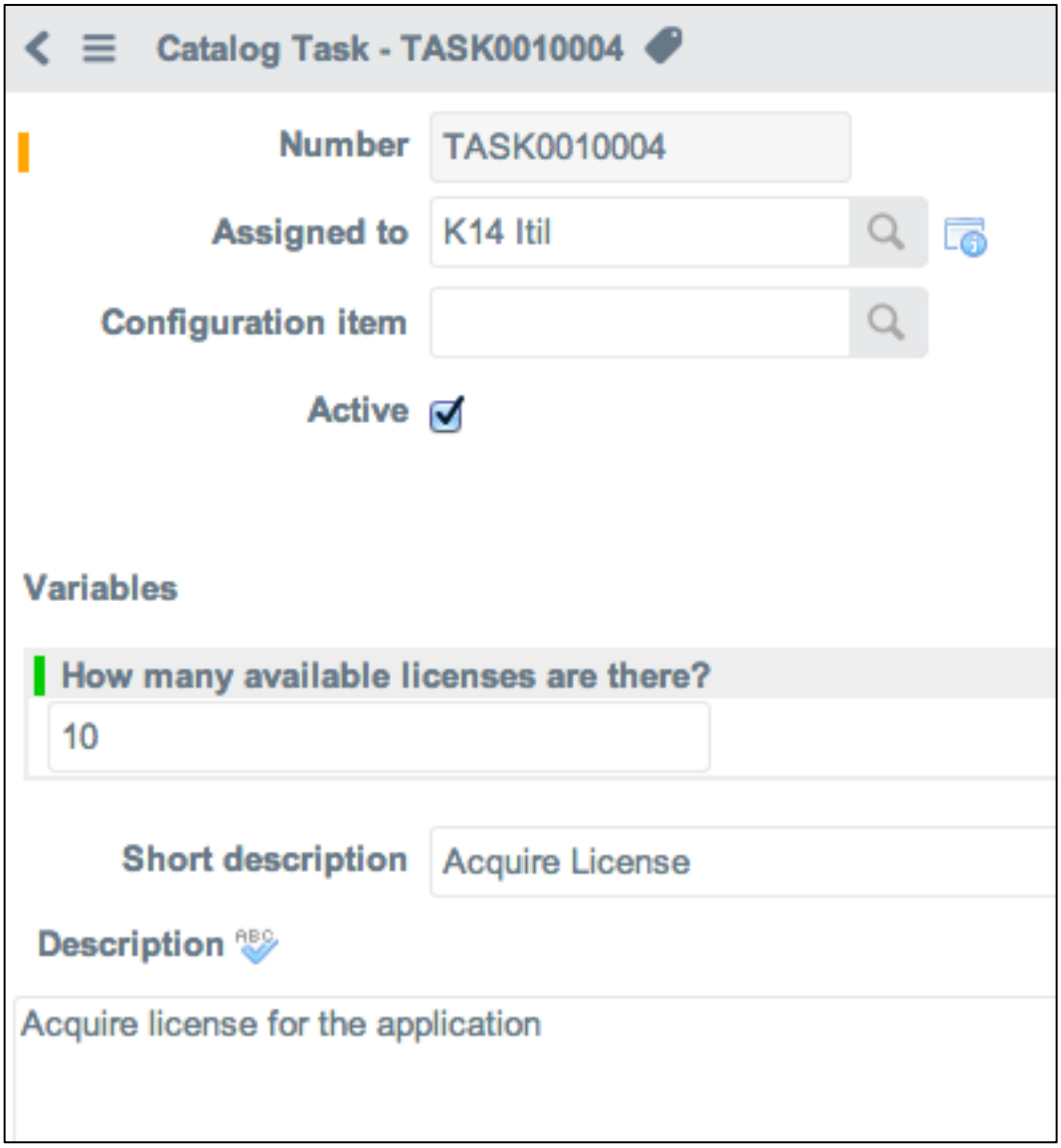
8. Impersonate **K14 Itil**.

9. Navigate to **Service Desk > My Work**.

10. Open up the task **Acquire License**.

©2014 ServiceNow, Inc. All rights reserved.

11. Enter **10** in the **How many available licenses are there?** variable field.



12. Click **Close Task**.

13. Navigate to **Service Desk > My Work**.

©2014 ServiceNow, Inc. All rights reserved.

14. Open up the task **Verify license for application installation**.

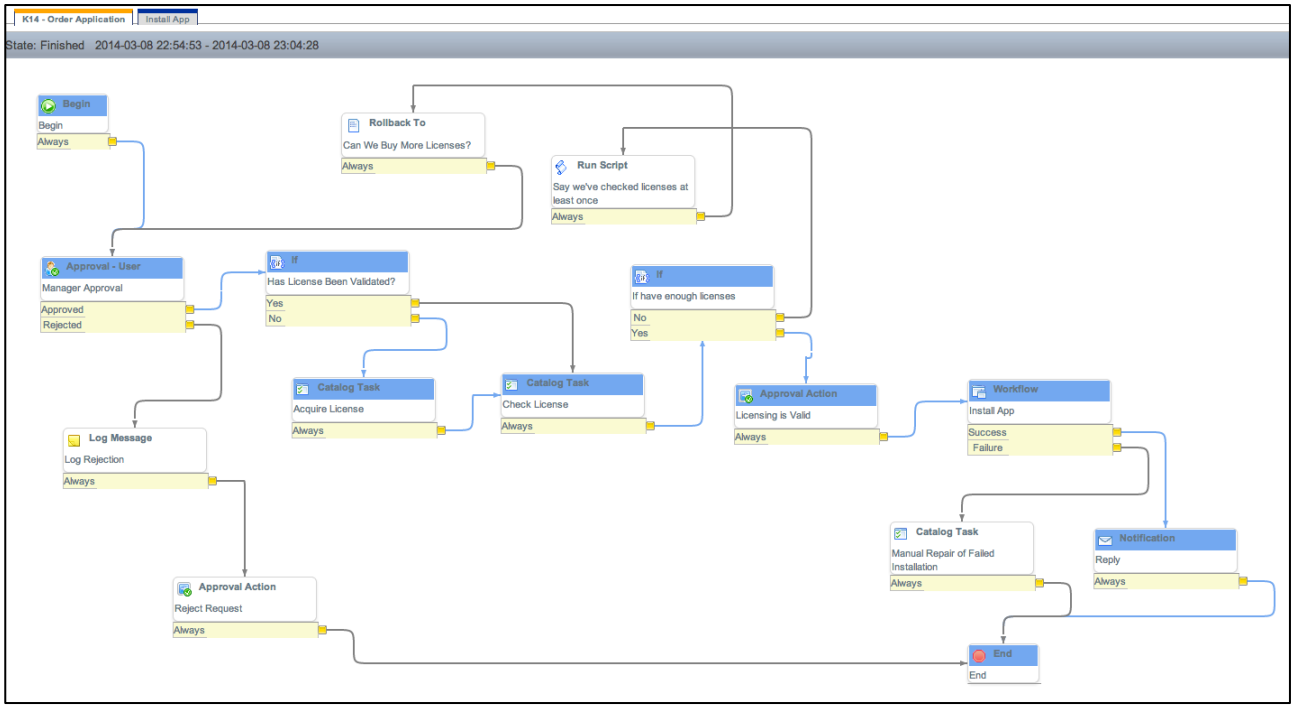
15. Select **Yes** in the **Have we already checked licenses?** option

16. Click **Close Task**.

17. Impersonate **System Administrator**.

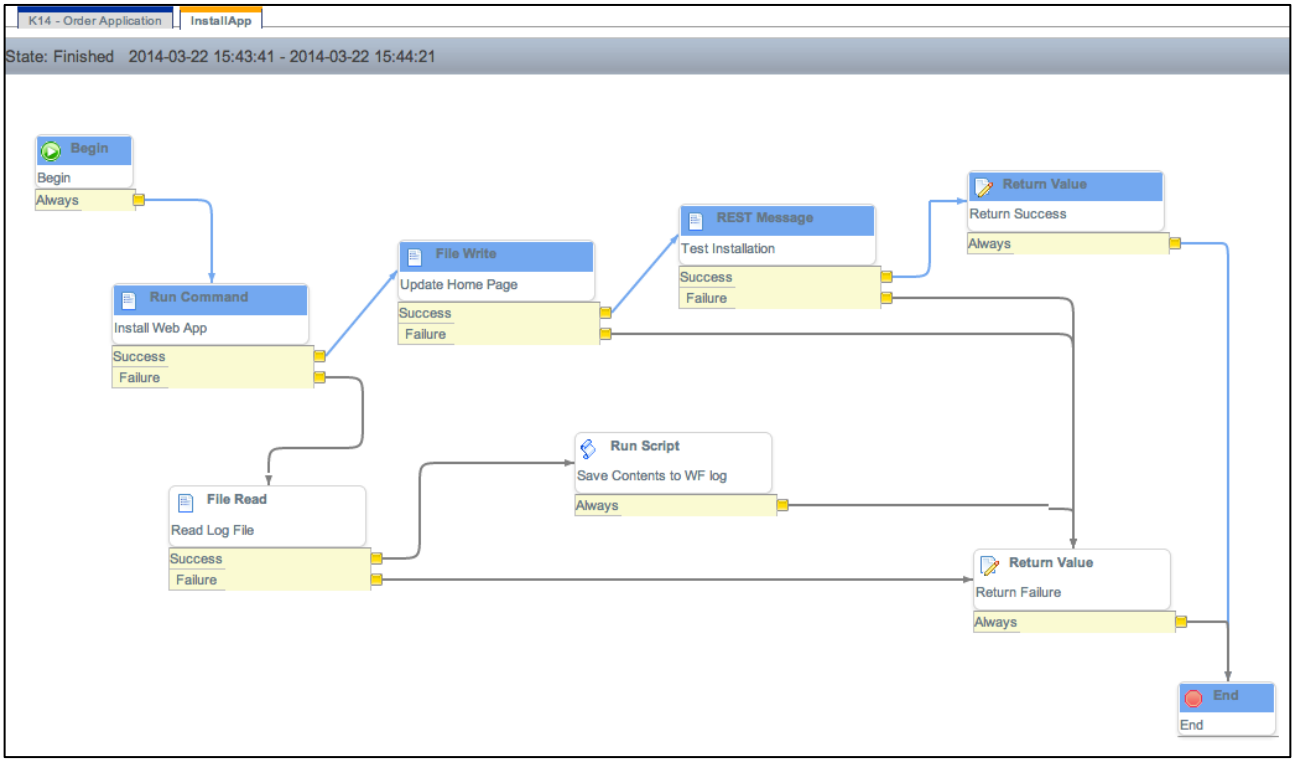
18. Navigate to **Workflow > All Contexts**, view the two workflow contexts. When the parent workflow has completed, navigate to the resulting webpage. Your request was successful!

    K14 Order Application workflow:



©2014 ServiceNow, Inc. All rights reserved.

InstallApp workflow:



®2014 ServiceNow, Inc. All rights reserved.