Lab Guide

Source Control:

Tracking Your App Development in Git

Chris Maloy & Chuck Tomasi

Default Login / Password:

admin / Knowledge16

itil / Knowledge16

employee / Knowledge16

This
Page
Intentionally
Left
Blank

Lab Goal

Lab 1.1
Basics

This lab explains the basics of tracking application development in the ServiceNow Studio IDE with Git. Highlights of this lab include:

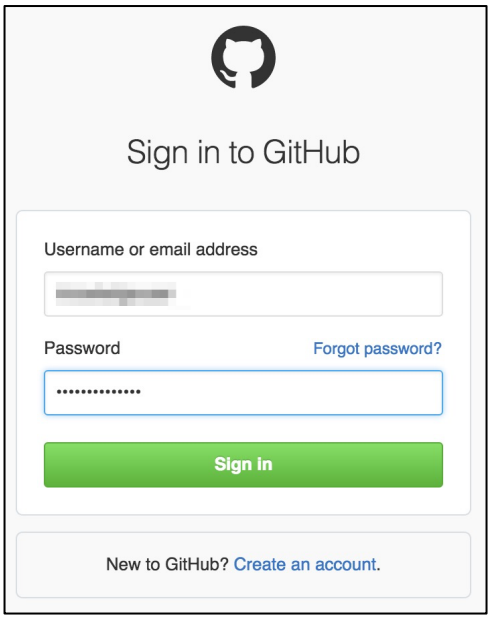
- Download and install lab materials
- Create a Git repository
- Create a simple app in ServiceNow
- Link your app to source control
- Commit your changes
- Review your changes on Git

Prerequisite

This lab requires an active GitHub account. If you have not already set up a GitHub account, follow the instructions in the document **http://bit.ly/GitHubSetup**.

Create a Copy of and Import the Lab Materials

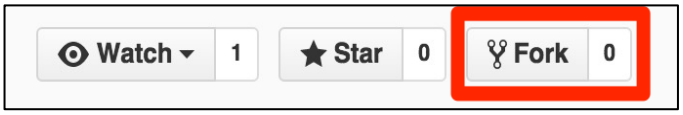
1. Log in to your GitHub account at **https://github.com/login**.



®2016 ServiceNow, Inc. All rights reserved.

2. Navigate to **https://github.com/chucktomasi/17LA01-Scripts**.

3. Click **Fork**.



4. Note in the upper left that the repository has been copied to your account. You now have a copy of the lab material for reference after the conference!

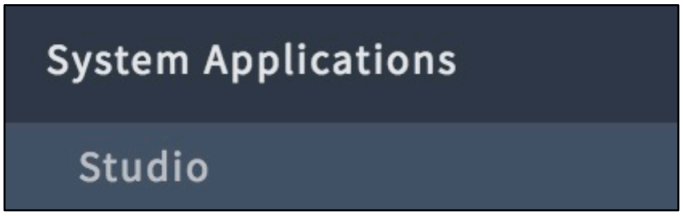


5. Locate the **HTTPS** field and click the clipboard to the right. This action copies the URL in the clipboard.



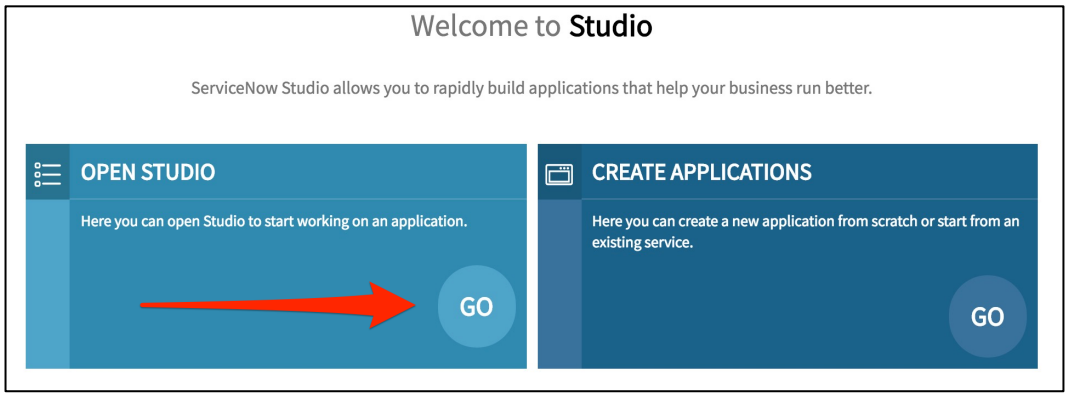
6. Log in to your instance with the credentials provided.

7. Navigate to **System Applications > Studio**.

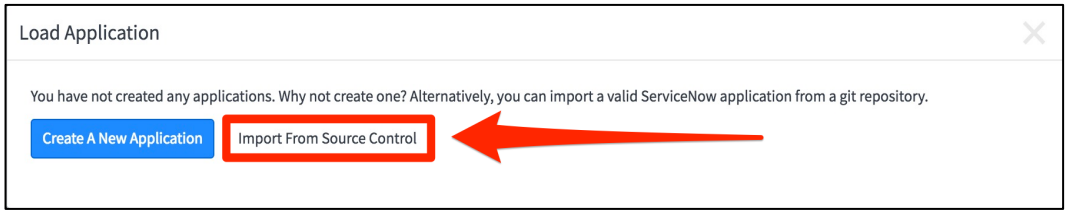


©2016 ServiceNow, Inc. All rights reserved.

8. Click **Go** in the **Open Studio** section on the left.



9. Click **Import From Source Control**.

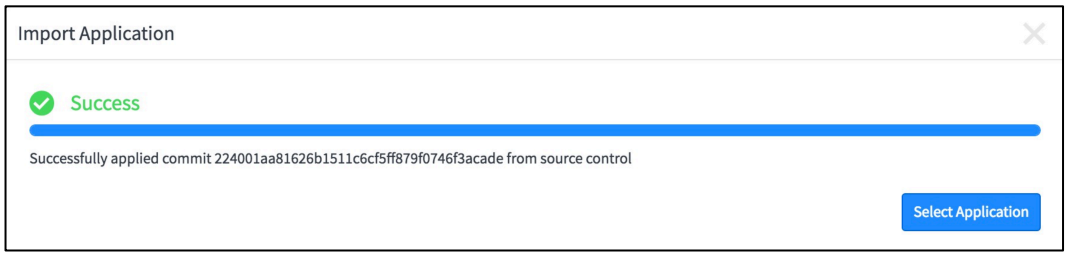


10. In the Import Application window, paste the URL copied in step **5** and provide your GitHub credentials. Click **Import**.



©2016 ServiceNow, Inc. All rights reserved.

11. When the import completes, click **Select Application**.



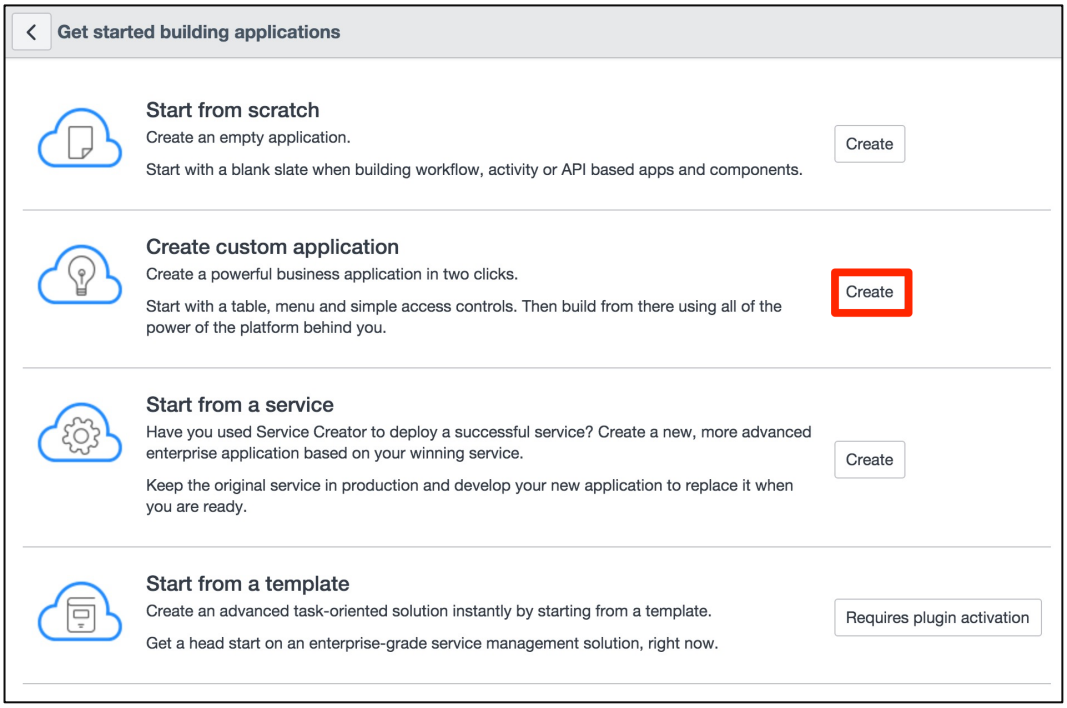
12. Click **Create a New Application**.



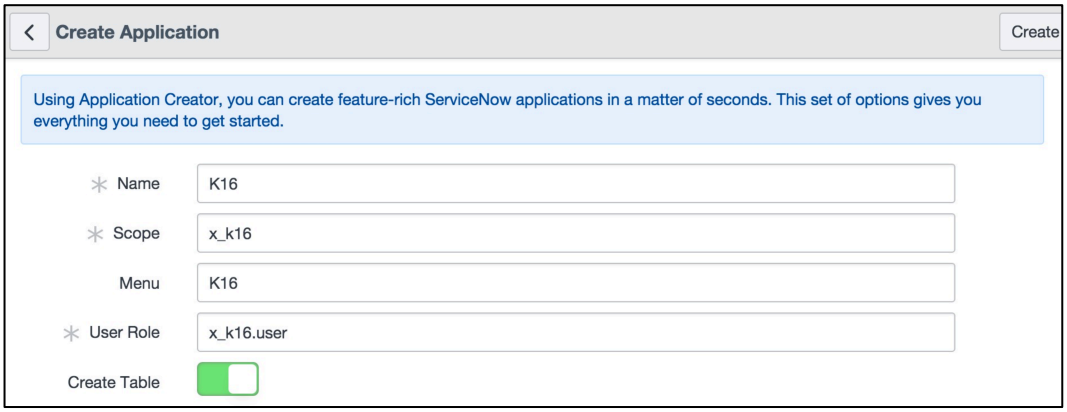
®2016 ServiceNow, Inc. All rights reserved.

Create a Simple App

1. Locate the section labeled **Create custom application**. Click **Create** as shown in the example.



2. On the Create Application screen, fill in the form with the following information:
   Name: **K16**
   Scope: (automatically populated)
   Menu: **K16**
   User role: (automatically populated)
   Create table: **On**



®2016 ServiceNow, Inc. All rights reserved.

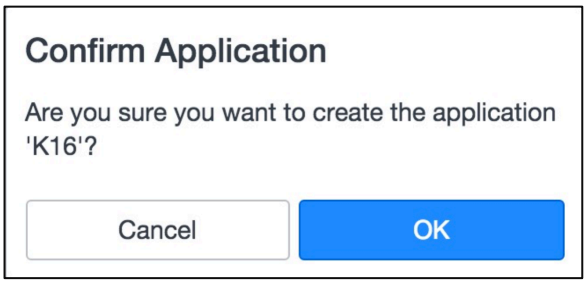
3. Scroll down and fill in the remainder of the form with the following information:

   Label: **Lab**
   Name: (automatically populated)
   Module: (automatically populated)



4. Click **Create**.

5. Click **OK** on the Confirm Application window.

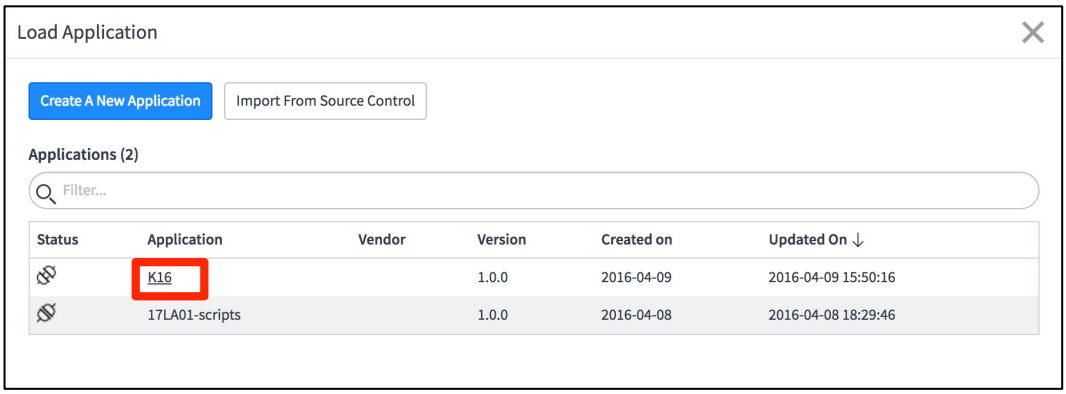


6. Click **Back to list** in the Application Creation window.



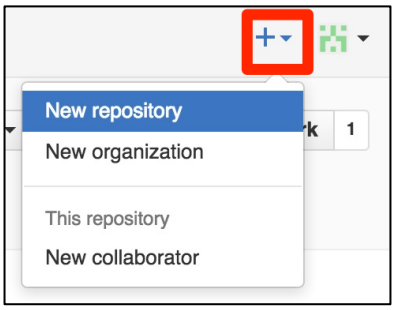
©2016 ServiceNow, Inc. All rights reserved.

7. Select **K16** in the Application column.



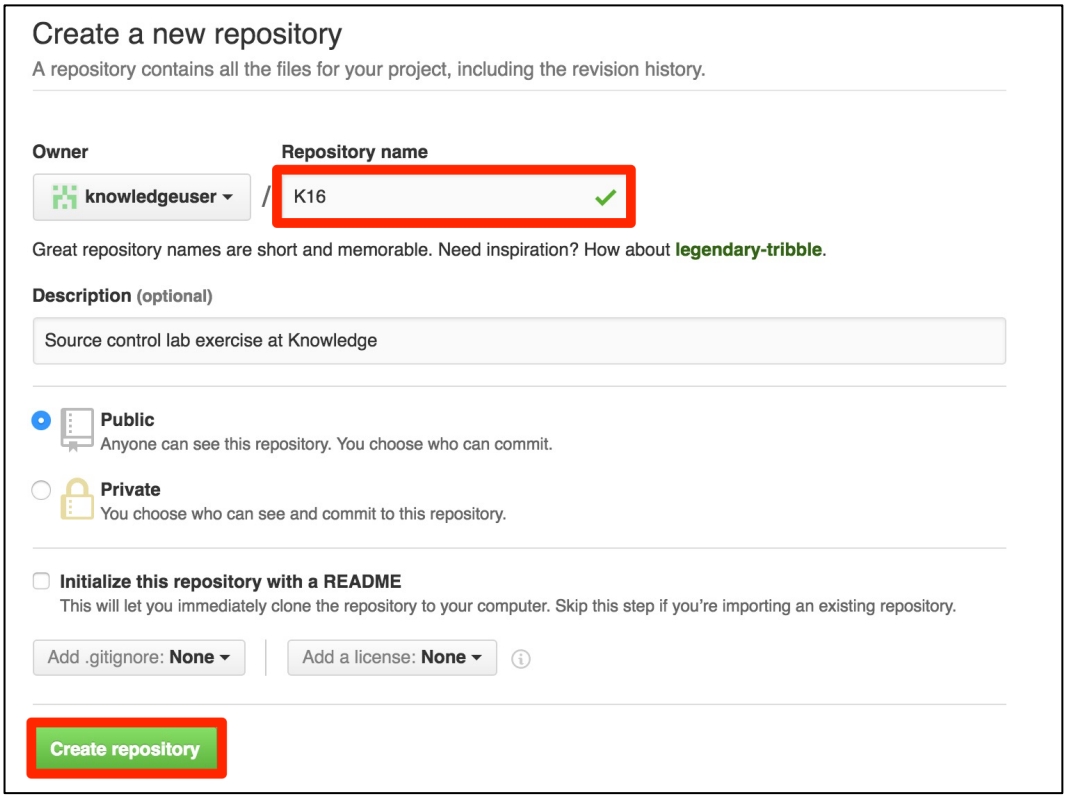
Create the Git Repository

1. In the Git browser tab, click the + icon in the upper right and choose **New repository**.



©2016 ServiceNow, Inc. All rights reserved.

2. In the Create a new repository form, fill in the Repository name field with **K16**. Click **Create repository**.

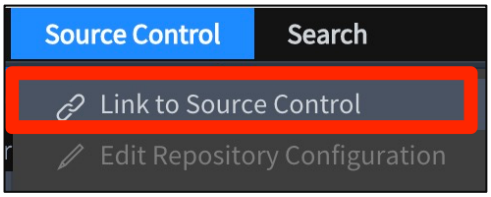


3. Locate the **HTTPS** field and click the clipboard to the right. This action copies the URL in the clipboard.



Link the Application to Source Control

1. From the Studio screen, navigate to **Source Control > Link to Source Control**.

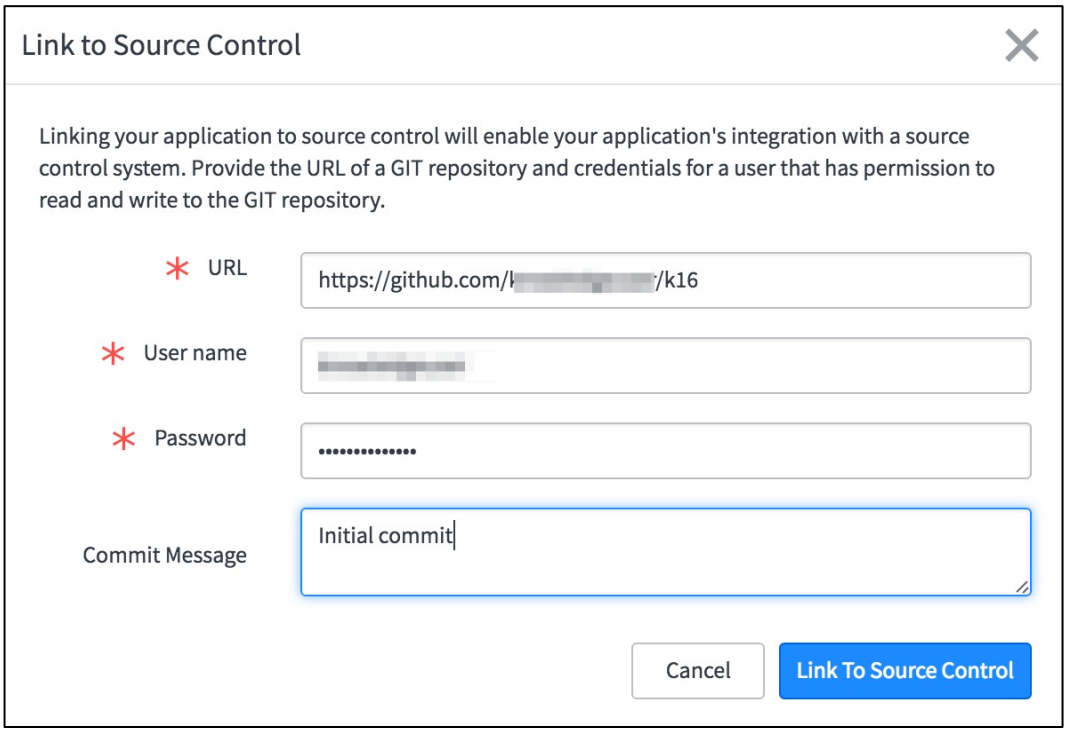


©2016 ServiceNow, Inc. All rights reserved.

2. In the form that appears, fill in the form with the following information:

   URL: (paste the value you copied from Git)
   User name: (your Git user name)
   Password: (your Git password)
   Commit Message: **Initial connection**

3. Click **Link to Source Control**.



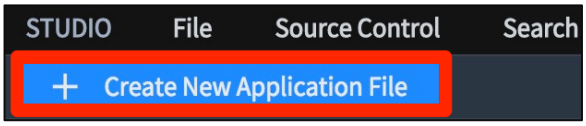
4. Confirm that you connected successfully. If not, make necessary corrections to the fields above and click **Close Dialog**.



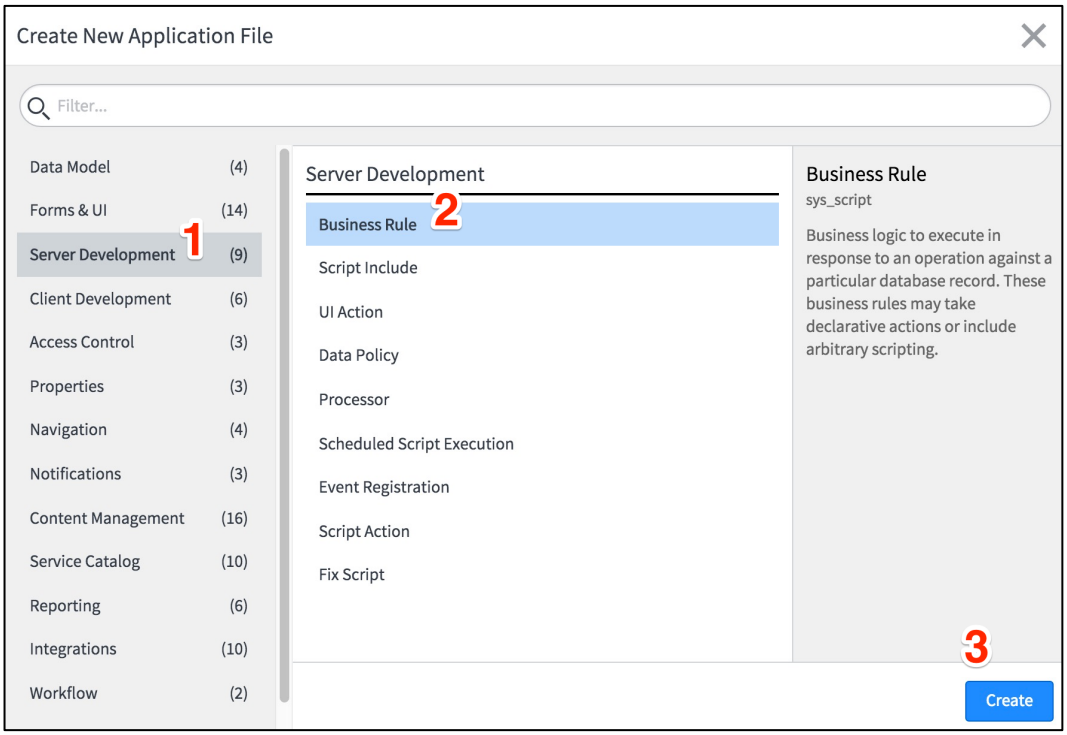
©2016 ServiceNow, Inc. All rights reserved.

Make and Commit Changes

1. In Studio, click **Create New Application File**.



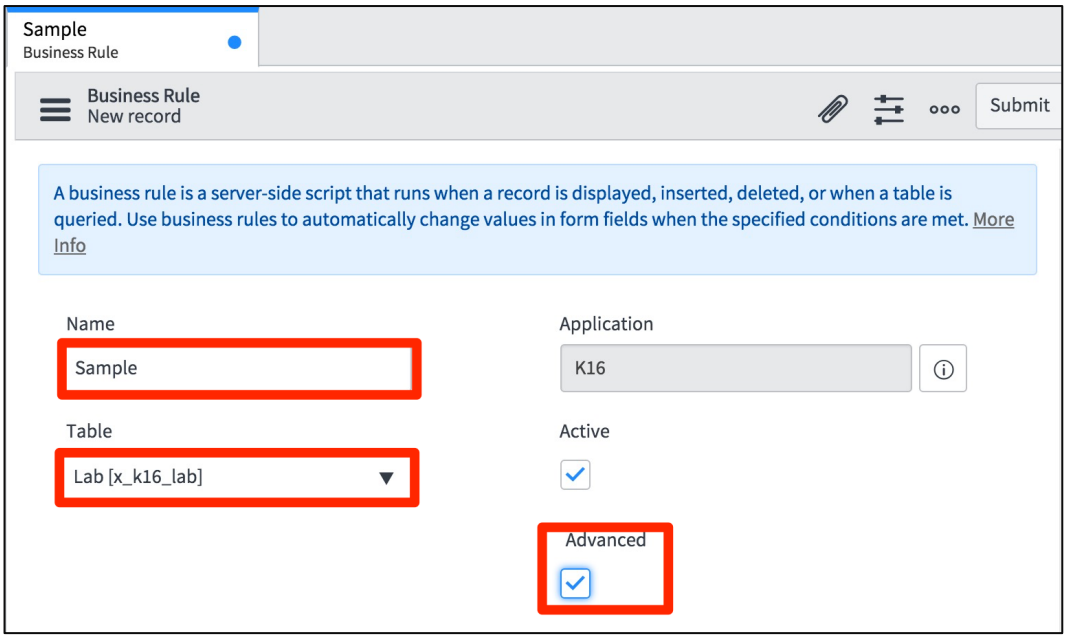
2. Click **Server Development** (1), **Business Rule** (2). Click **Create** (3).



©2016 ServiceNow, Inc. All rights reserved.

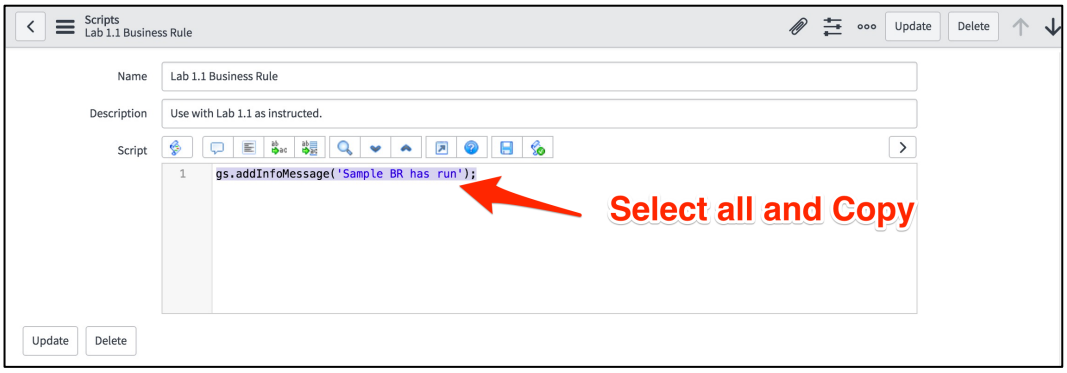
3. In the Business Rule form, complete the form with the following information:

   Name: **Sample**
   Table: **Lab**
   Advanced: **true**



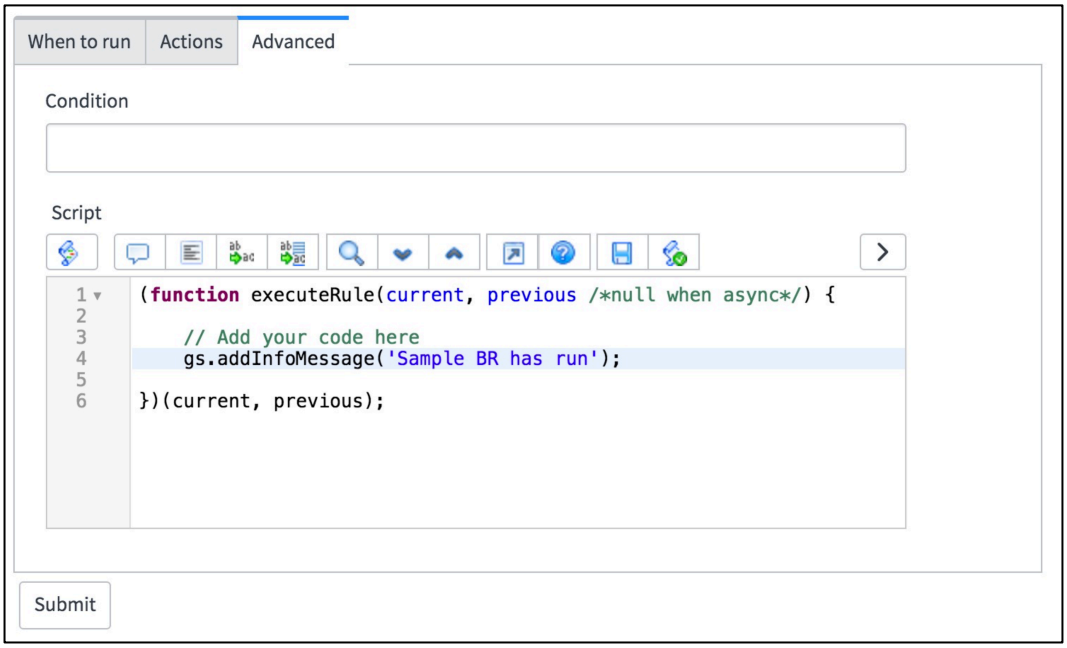
4. Scroll down the form. Click the **Advanced** tab.

5. In the main ServiceNow browser window, refresh your browser then navigate to **Source Control Lab > Scripts**. Open the record labeled **Lab 1.1 Business Rule**. Copy the contents of the Script field.



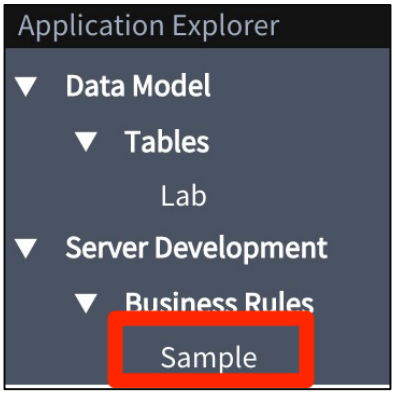
©2016 ServiceNow, Inc. All rights reserved.

6. In Studio, paste the script contents just below the line **//Add your code here**. Your business rule script should appear similar to the example.



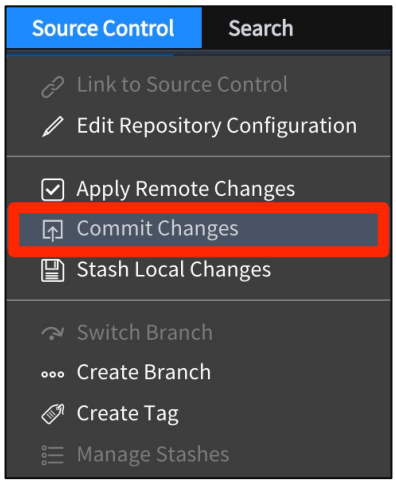
7. Click **Submit**.

8. Note that the new business rule appears in the studio navigator on the left.

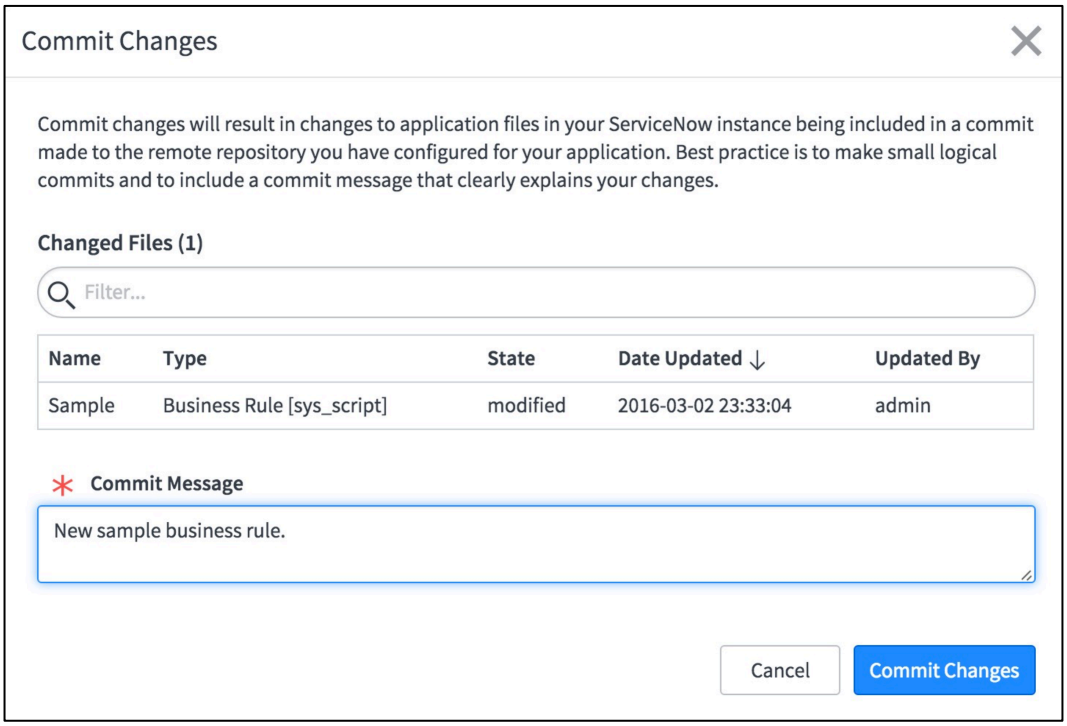


©2016 ServiceNow, Inc. All rights reserved.

9. Under the **Source Control** menu, click **Commit Changes**.



10. In the Commit Changes window, enter text in the **Commit Message** field as shown in the example. Click **Commit Changes**.



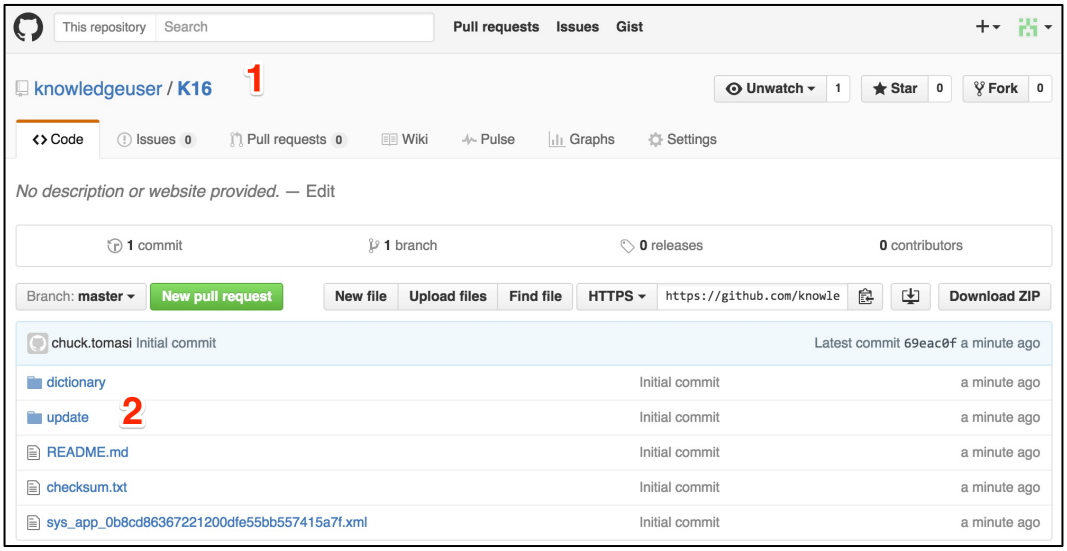
©2016 ServiceNow, Inc. All rights reserved.

11. Verify your commit was successful and click **Close Dialog**.



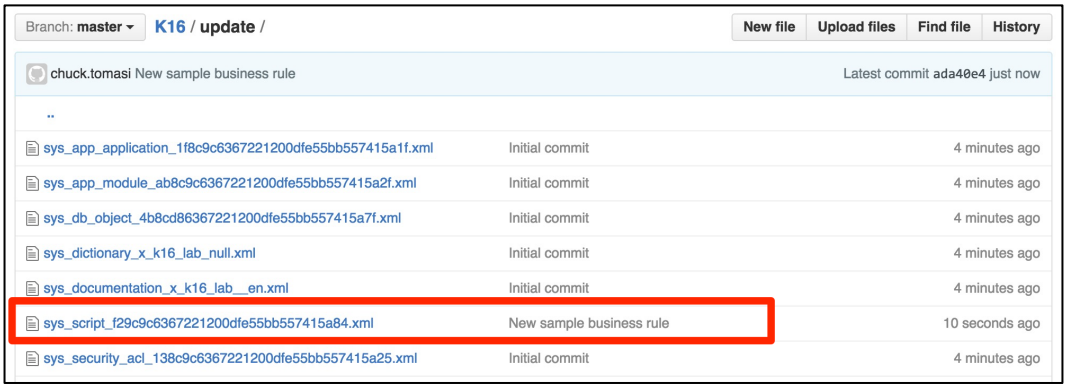
Lab Verification

1. Switch to the GitHub browser tab and click the **project** link (1) then click the **update** folder (2).



©2016 ServiceNow, Inc. All rights reserved.

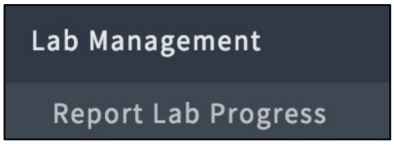
2. Note that your business rule has been committed by locating the list entry beginning with **sys_script_** and has a comment "**New sample business rule**."



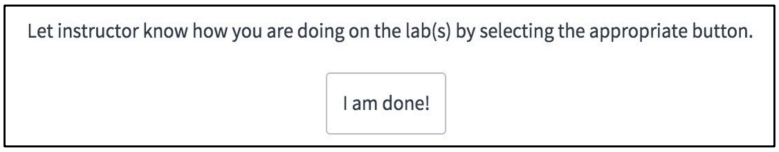
Progress Report

1. Switch to the ServiceNow main window.

2. Navigate to **Lab Management > Report Lab Progress**.



3. Click **I am done!**



©2016 ServiceNow, Inc. All rights reserved.

Lab Goal

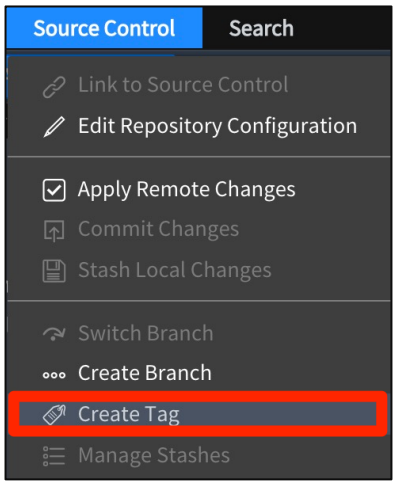
Lab 1.2 Tagging & Branching

This lab explains the basics of managing different releases of your application. Highlights of this lab include:

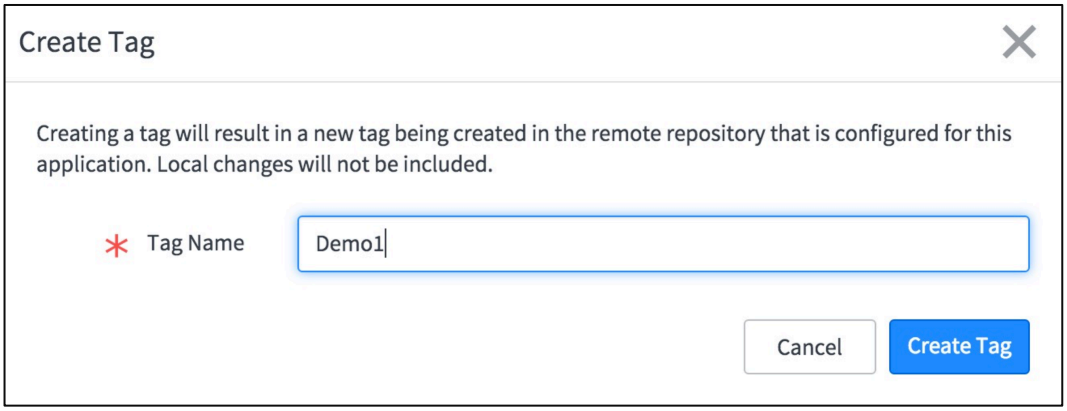
- Create tags
- Create branches
- Switch between branches

Create a Tag

1. In the Studio window, navigate to **Source Control > Create Tag**.



2. Enter the tag name **Demo1** in the pop-up window and click **Create Tag**.



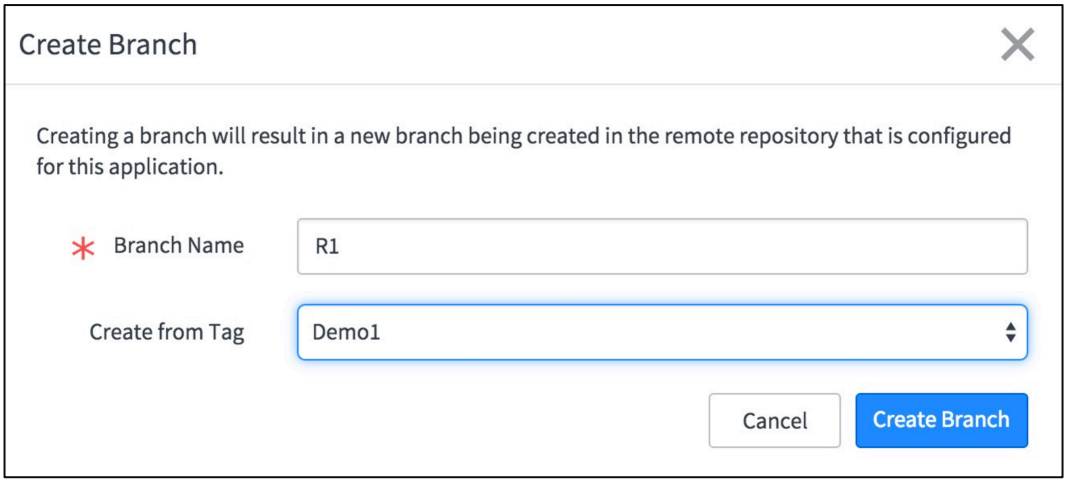
©2016 ServiceNow, Inc. All rights reserved.

3. Click **Close Dialog** in the Create Tag pop-up window.



Create a Release 1 Branch from a Tag

1. Navigate to **Source Control > Create Branch.**
2. In the Create Branch pop-up window, enter the Branch Name **R1** and select **Demo1** from the **Create from Tag** drop-down list.
3. Click **Create Branch**.

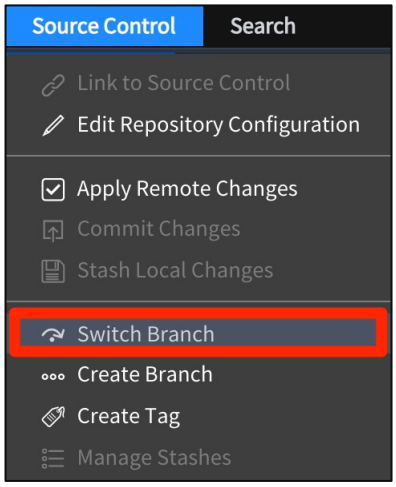


4. Click **Close Dialog** in the Create Branch pop-up window.

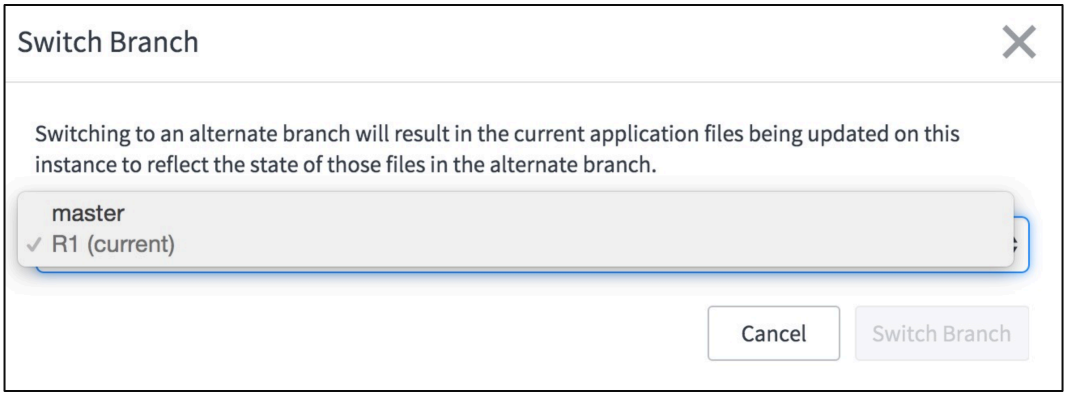
©2016 ServiceNow, Inc. All rights reserved.

Create a New Change on the Master Branch

1. Navigate to **Source Control > Switch Branch**.



2. Select **master** from the drop-down list and click **Switch Branch**.



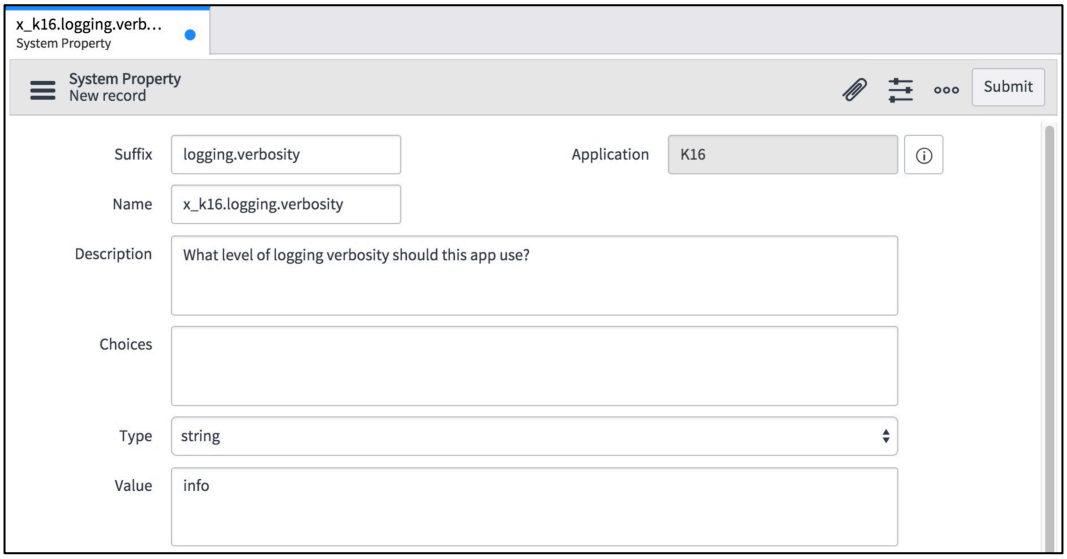
3. Click **Close Dialog** on the Switch Branch pop-up window.
4. Click **Create New Application File** in Studio.
5. Navigate to **Properties > System Property** and click **Create**.

©2016 ServiceNow, Inc. All rights reserved.

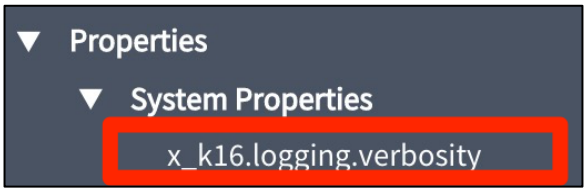
6. Complete the properties form with the following values:

   Suffix: **logging.verbosity**
   Name: (auto populated)
   Description: **What level of logging verbosity should this app use?**
   Type: **string**
   Value: **info**

7. Click **Submit**.

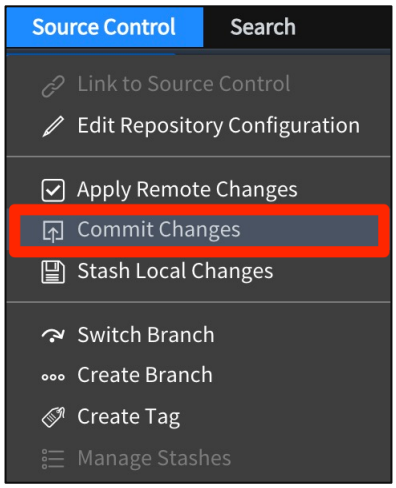


8. Note the new property appears on the left.



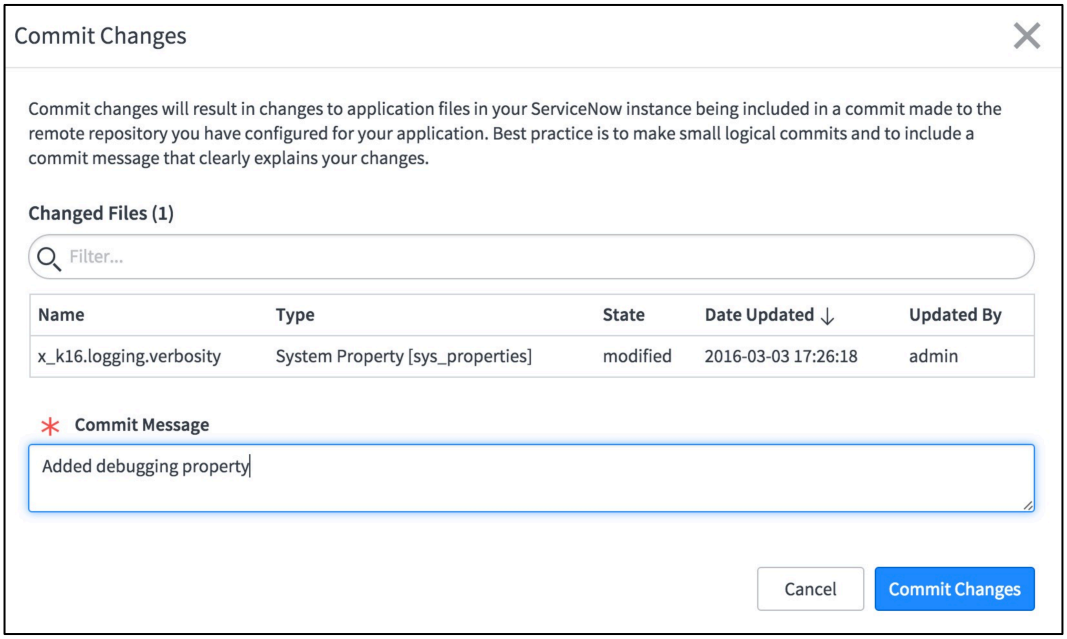
©2016 ServiceNow, Inc. All rights reserved.

9. Navigate to **Source Control > Commit Changes** to push the local changes to the Git repository.



10. Review the changes being pushed (in this case only one new property) and add the comment message **Added debugging property**.

11. Click **Commit Changes**.

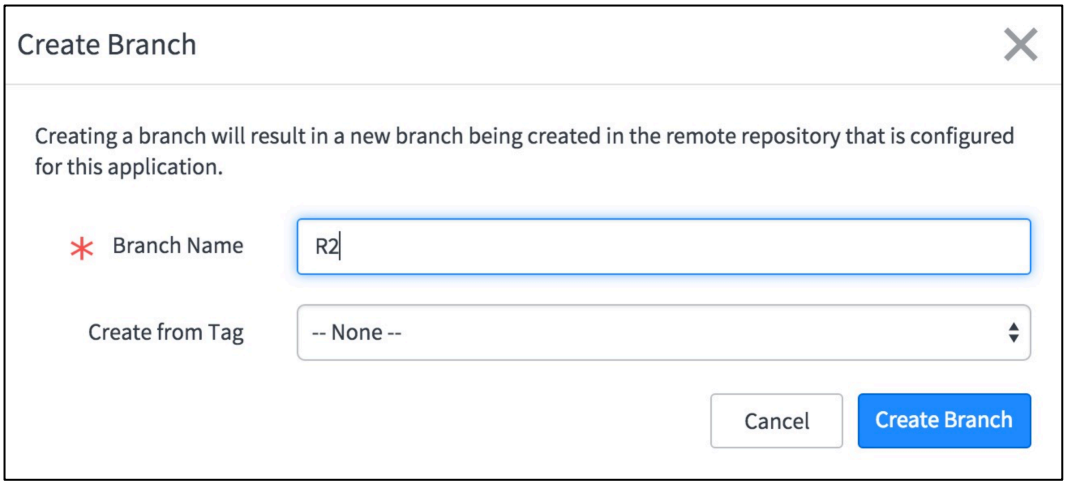


12. Click **Close Dialog** on the Commit Changes pop-up window.

®2016 ServiceNow, Inc. All rights reserved.

Create a Release 2 Branch

1. Navigate to **Source Control > Create Branch** in Studio.
2. Create a new branch called **R2** and set the **Create from Tag** drop-down list as **-- None --**.
3. **Create Branch**.



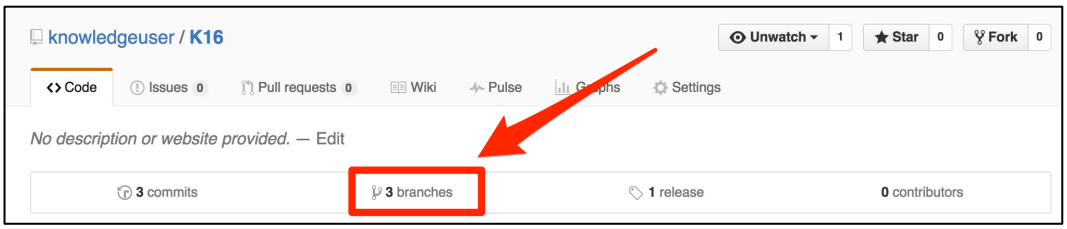
4. Click **Close Dialog** on the Create Branch pop-up window.

Lab Verification

1. Switch to the GitHub browser window and click **K16**.

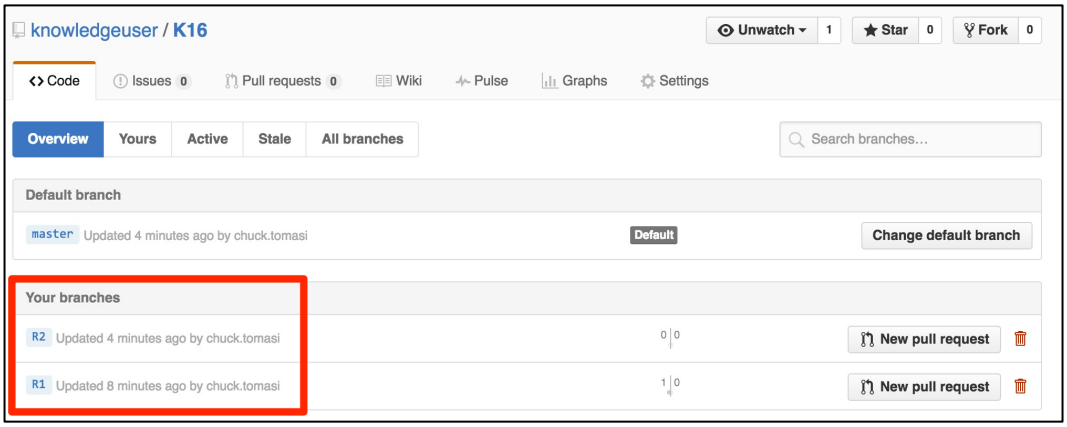


2. Locate and click the **branches** link in the header section.

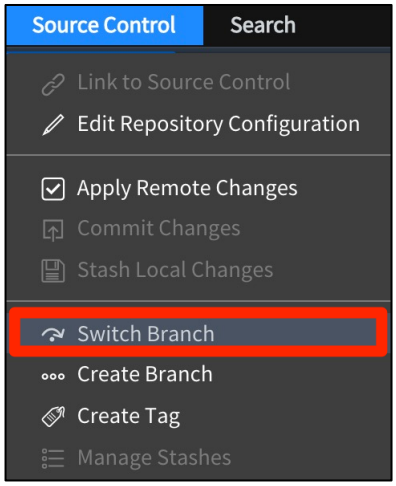


©2016 ServiceNow, Inc. All rights reserved.

3. Confirm the commits and branch operations in ServiceNow worked properly by noting R1 and R2 exist in the **Your branches** section.



4. In the Studio window, navigate to **Source Control > Switch Branch**.

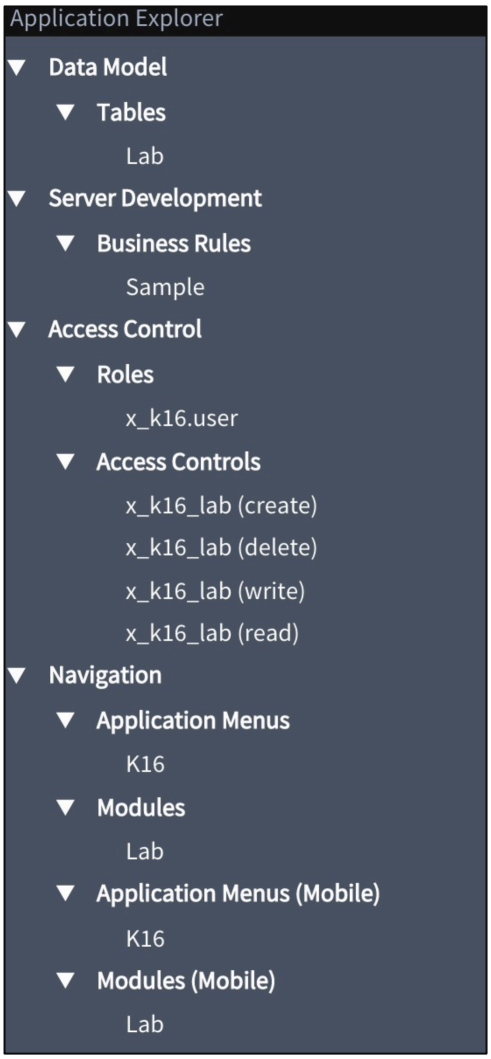


5. Select the **R1** branch from the drop-down list and click **Switch Branch**.

6. Click **Close Dialog** on the Switch Branch pop-up window.

©2016 ServiceNow, Inc. All rights reserved.

7. Note that your property does not exist in R1. This confirms that you can successfully make changes in a later release (R2) and switch back to review or make changes in an earlier release (R1) at any time.



8. Switch to the **R2** branch and notice your property is available.

9. Switch to the **Master** branch and notice your property is available.

©2016 ServiceNow, Inc. All rights reserved.

Progress Report

1. Switch to the ServiceNow main window.
2. Navigate to **Lab Management > Report Lab Progress**.
3. Click **I am done!**

©2016 ServiceNow, Inc. All rights reserved.

Lab Goal

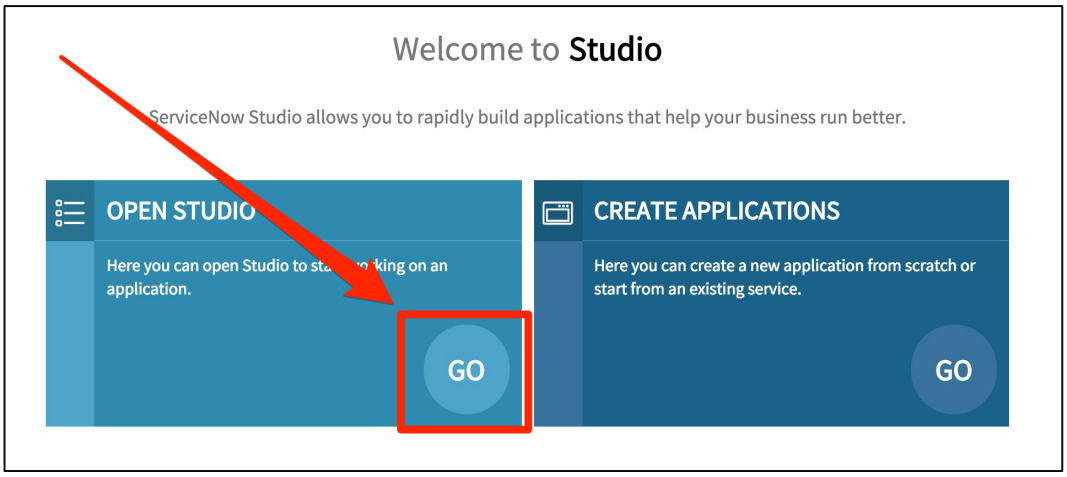
Lab 2.1 Multiple Instances

This lab explains how to manage an application when multiple development instances are involved. Developers may make local changes to a file that conflict with work done by other developers on the same file. This lab explains how to recognize and resolve conflicts. Highlights of this lab include:

- Import an existing app from the repository
- Manage collisions

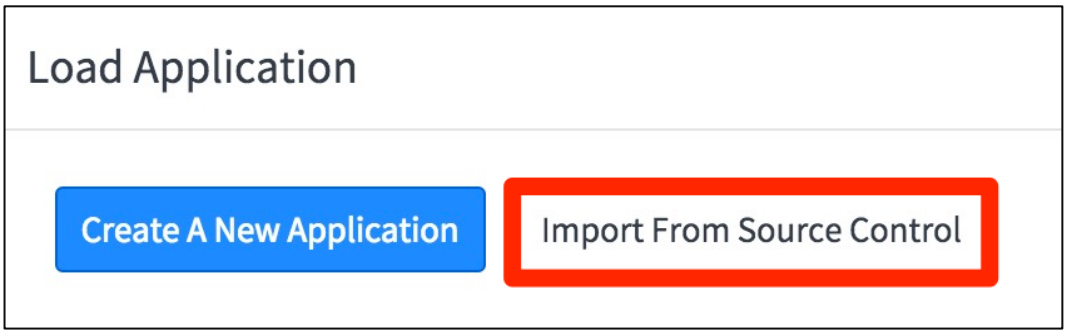
Import an Existing App

1. Log in to your second instance with the admin credentials provided.
2. Navigate to **System Applications > Studio**.
3. On the following screen, click the **Go** button under Open Studio.



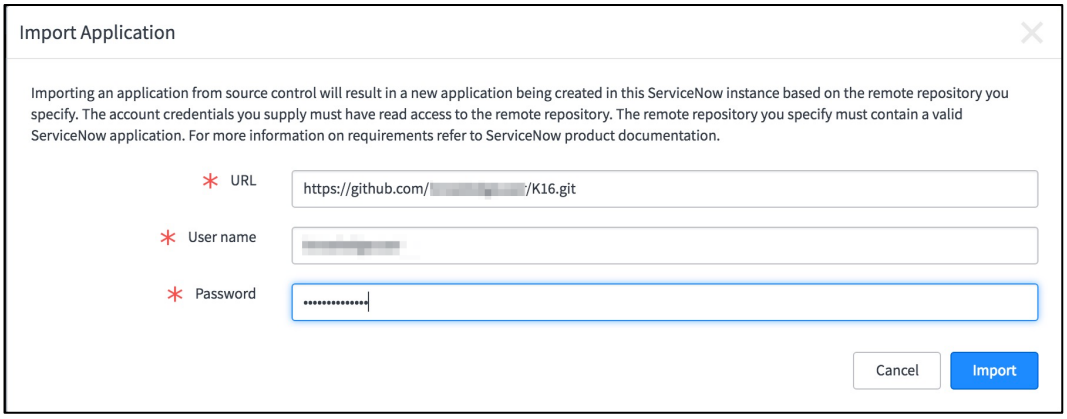
©2016 ServiceNow, Inc. All rights reserved.

4. Click **Import From Source Control** in the pop-up window.



5. Enter the URL and credentials to your **K16** Git repository. If needed, go to the repository, select the project and copy the URL again.

6. Click **Import**.



7. After the import completes, click **Select Application**.

8. Click **K16** in the Application column.



©2016 ServiceNow, Inc. All rights reserved.

Create a Conflict

1. Open the **Sample** business rule in Studio.
2. Check the **Insert** and **Update** check boxes on the tab labeled **When to run**.



3. Copy/paste the contents of the **Lab 2.1 - Hello** record (in **Source Control Lab > Scripts**) into the **Script** field. Place the **gs.debug()** statement just after the **gs.addInfoMessage()** line as shown in the example.

```
(function executeRule(current, previous /*null when async*/) {

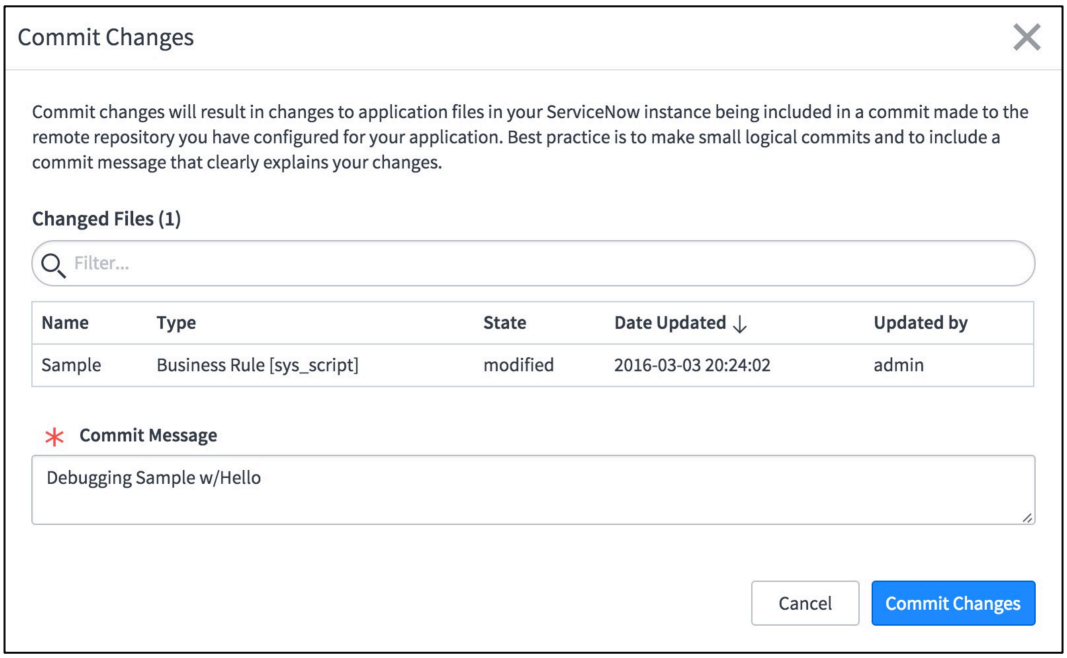
    // Add your code here
    gs.addInfoMessage('Sample BR has run.');
    gs.debug('Hello');

})(current, previous);
```

4. Click **Update**.
5. Navigate to **Source Control > Commit Changes**.

©2016 ServiceNow, Inc. All rights reserved.

6. Provide a commit message and click **Commit Changes**.



7. Click **Close Dialog** on the Commit Changes pop-up window.

8. Switch to your first instance.

9. Ensure you are on the **master** branch by looking in the upper-right corner of the studio window. You should see the word **master** as pictured below.



10. Open the **Sample** business rule.

11. Ensure the **Input** and **Update** check boxes are **NOT** checked.

©2016 ServiceNow, Inc. All rights reserved.

12. Copy/paste the contents of the **Lab 2.1 - Goodbye** record (in **Source Control Lab > Scripts**) into the **Script** field. Place the **gs.debug()** statement just after the **gs.addInfoMessage()** line as shown in the example.

```
(function executeRule(current, previous /*null when async*/) {

    // Add your code here
    gs.addInfoMessage('Sample BR has run.');
    gs.debug('Goodbye');

})(current, previous);
```

13. Click **Update**.

14. Navigate to **Source Control > Commit Changes**.

15. Add a commit message to the pop-up window and click **Commit Changes**.

16. Note that the committed changes from your second instance conflict with the recent changes from your first instance.

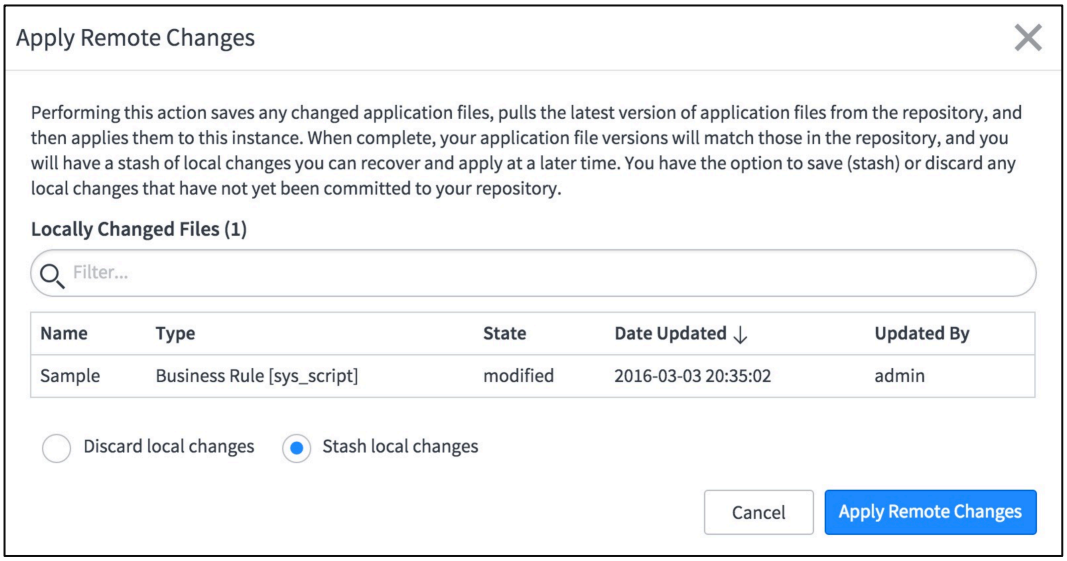


17. Click **Apply Remote Changes**.

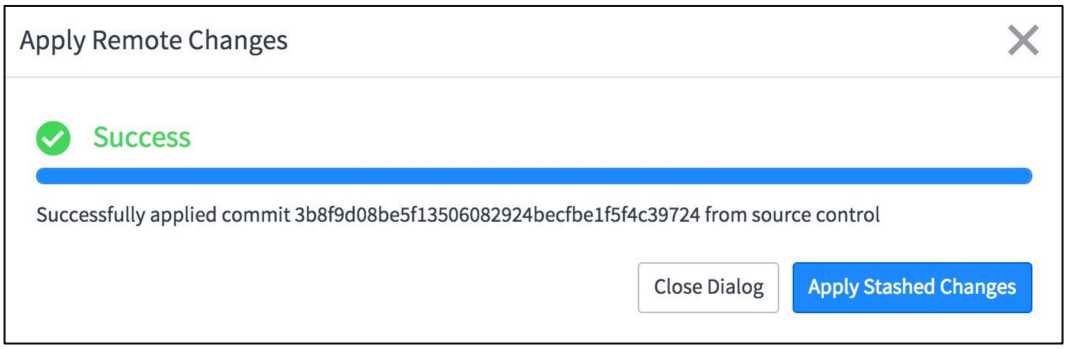
©2016 ServiceNow, Inc. All rights reserved.

Resolve the Conflict

1. Select the option **Stash local changes** and click **Apply Remote Changes** to get your local instance in sync with the repository.



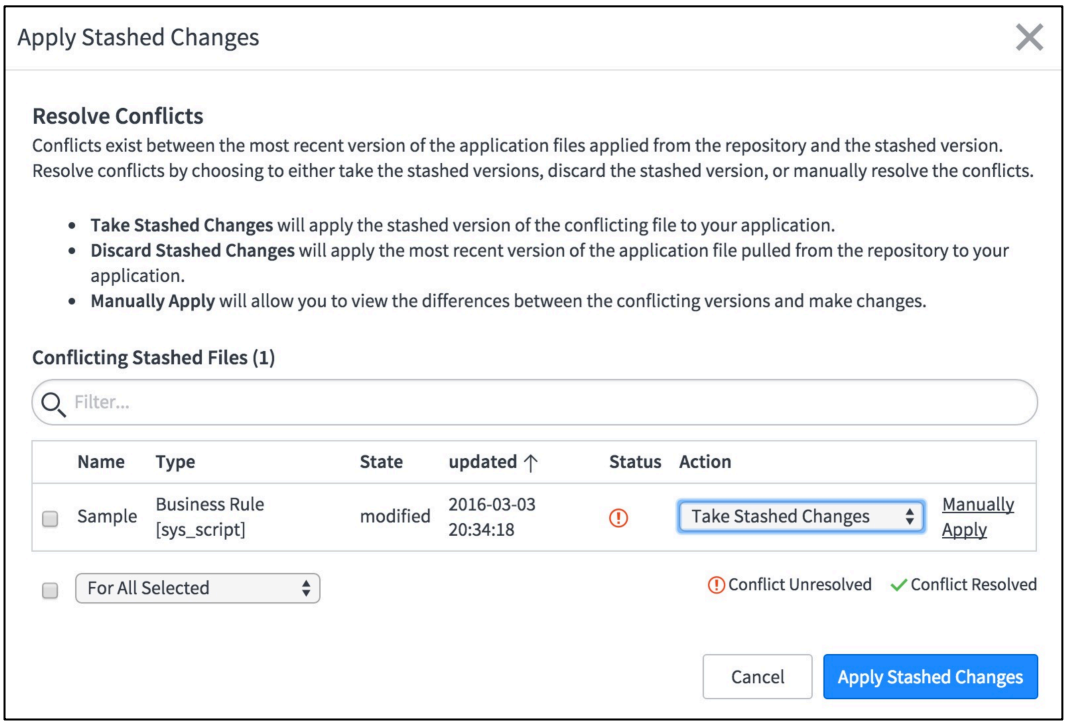
2. When the remote changes are pulled from the repository, click **Apply Stashed Changes**.



3. In this case, you want to keep the local changes you stashed for the business rule (the ones with the Goodbye message) and apply them locally. Use the drop-down list in the **Action** column and select **Take Stashed Changes**.

©2016 ServiceNow, Inc. All rights reserved.

4. Click **Apply Stashed Changes**.



5. Click **Close Dialog** in the Apply Stashed Changes pop-up window.

Lab Validation

1. At this point the local changes contain the **Goodbye** message and the repository for the master branch still says **Hello**. Commit the changes to update the repository.

2. On the second instance, navigate to **Source Control > Apply remote changes** in Studio to apply the remote changes. Note the Sample business rule has the two check boxes deselected and the script debug message is **Goodbye**.

Progress Report

1. Switch to the ServiceNow main window.

2. Navigate to **Lab Management > Report Lab Progress**.

3. Click I am done!

©2016 ServiceNow, Inc. All rights reserved.

Lab Goal

Lab 3.1 Manual Stashing

This lab expands your knowledge of stashing. You have seen how stashes are used to resolve conflicts. This lab explains how stashes can be used to move changes between branches. Highlights of this lab include:

- Create a stash
- Create a new branch
- Manually apply a stash

Create a Stash

1. Log in to your original instance.
2. Go to **Studio** and ensure you have the **master** branch selected.
3. Create a new property with the following attributes using the same method as the earlier lab exercise:

   Prefix: **logging.destination**
   Name: (auto-filled)
   Description: **Destination of debugging messages**
   Type: **string**
   Value: **file**

4. Click Submit.
5. Create another property with the following attributes:

   - Prefix: **lab.author**
   - Name: (auto-filled)
   - Description: **Who wrote this awesome lab?**
   - Type: **string**
   - Value: **(your name)**

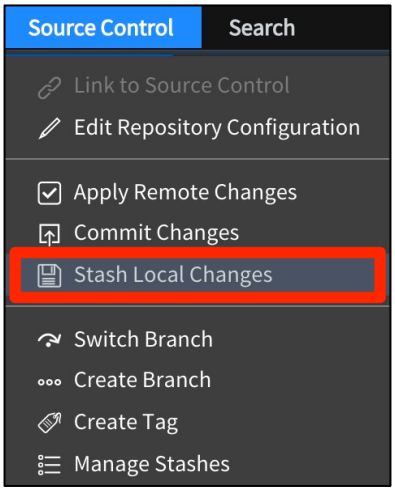
6. Click **Submit**.

You have now created two local changes. Before committing them to the repository, you can stash them away to be applied to one or more branches, almost like a patch. This is only available for local

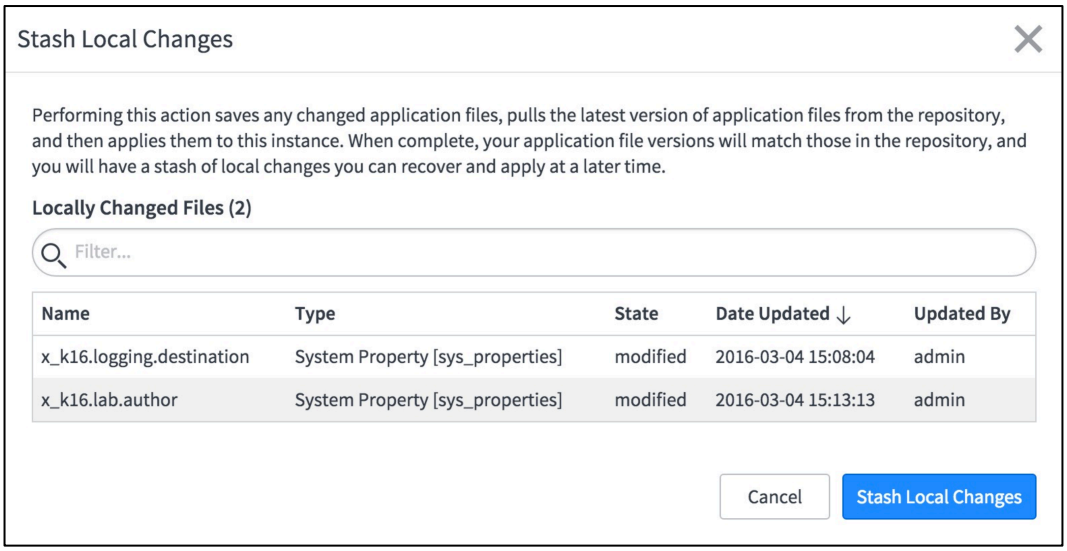
©2016 ServiceNow, Inc. All rights reserved.

changes. Note that **ALL** local changes in a stash are applied at once. If you create multiple stashes at a time, it may be difficult to identify which stash contains which changes.

7. Navigate to **Source Control > Stash Local Changes**.



8. A confirmation screen (below) appears to let you know what local changes are being stashed. Click **Stash Local Changes**.



9. As part of the stash operation, Studio reloads the latest commit to the current branch from the repository. This is evident by the two new properties missing from the left application explorer window.

©2016 ServiceNow, Inc. All rights reserved.

10. When the stash operation completes, click **Apply Stashed Changes** to bring the local changes back up to date with the two new properties. **Note**: These local changes still have not been committed and are still available in the stash.



11. Click **Close Dialog** in the Apply Stashed Changes pop-up window.

12. Note that the two new properties are available in the current branch.

13. Navigate to **Source Control > Commit Changes** to commit the two new properties to the master branch.

Apply the Stash to an Older Branch

1. Navigate to **Source Control > Switch Branch** and switch to the R1 branch.

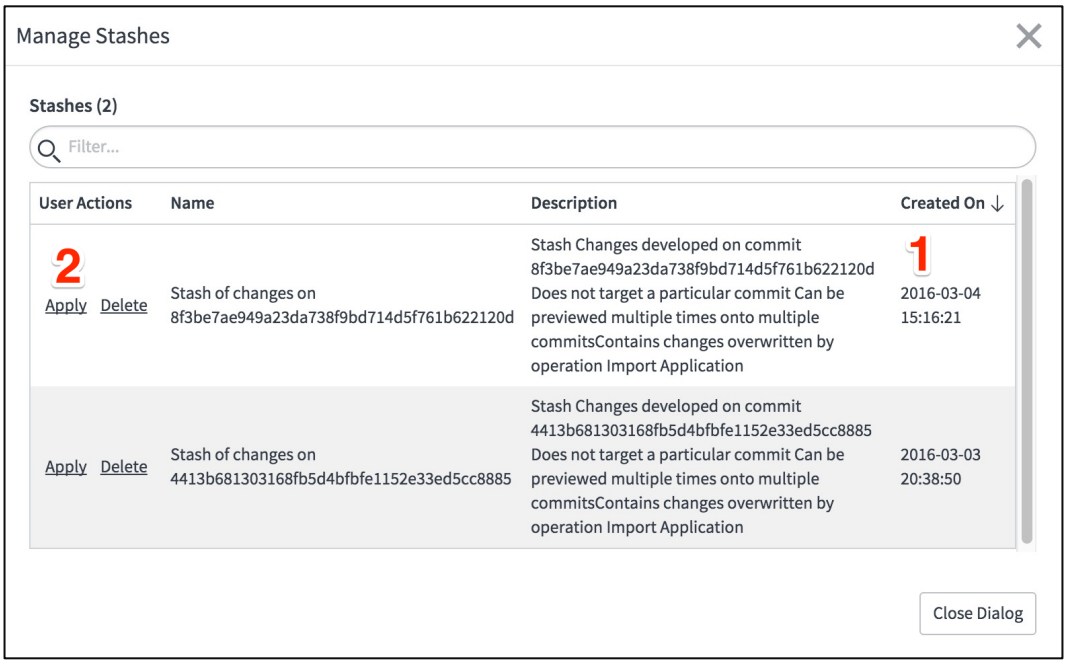
2. Note which properties are available on this branch.

3. Navigate to **Source Control > Manage Stashes**.

4. Identify the stash with the most recent Created On date/time (1).

©2016 ServiceNow, Inc. All rights reserved.

5. Click the **Apply** link (2) associated with that line item.



6. Click **Close Dialog** on the Apply Stashed Changes pop-up window.

7. Note which properties are now available on the R1 branch.

8. Commit the changes to save the new properties to the R1 branch.

Lab Verification

1. Validate the following using **Source Control > Switch Branch**:

   Branch R1 has two properties: **x_k16.lab.author and x_k16.logging.destination**.
   Branch R2 has one property: **x_k16.logging.verbosity**.
   Master branch has three properties**: x_k16.lab.author, x_k16.logging.destination, x_k16.logging.verbosity**.

Progress Report

1. Switch to the ServiceNow main window.

2. Navigate to **Lab Management > Report Lab Progress**.

3. Click I am done!

©2016 ServiceNow, Inc. All rights reserved.