

# Lab Guide

## Team Development: Beyond Update Sets

Joanna Nelson & Andrew Martin

Default Login / Password:

admin / Knowledge15

itil / Knowledge15

employee / Knowledge15

This  
Page  
Intentionally  
Left  
Blank

## Lab Goal

This lab explains how to set up and work with Team Development

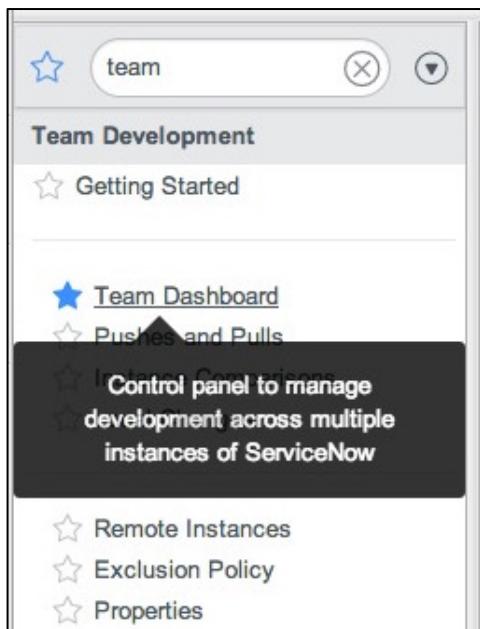
- Set up your ServiceNow instances for Team Development
- Get to know the Team Dashboard
- Understand Local Changes

## Lab 1 Setup and the Basics

### Access Your Instances

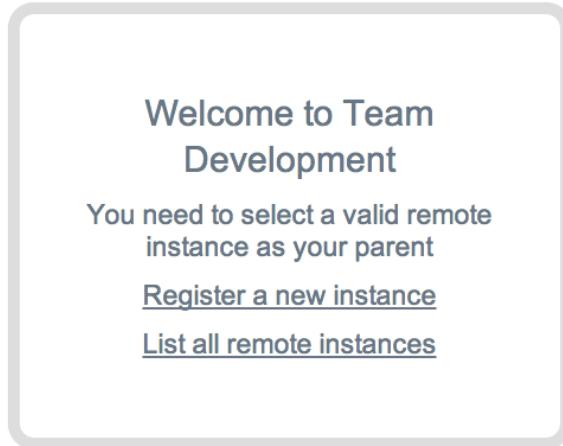
You will be provided two instances to use for this lab.

1. Identify which instance is your child development instance and which is your parent instance.  
Log on to your **child** development instance using the admin credentials.
2. Navigate to **Team Development > Team Dashboard**.



## Register Your Parent Instance

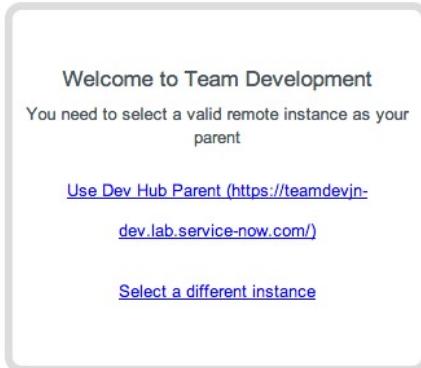
1. Click the **Register a new instance** link.



2. Enter the **parent** instance values provided when you received your lab instances, URL, and credentials. Specify the name as "Dev Hub". Click the **Test Connection** to confirm a successful connection with your parent instance. Click the **Submit** button.

Name	<input type="text"/>
Type	Development
Active	<input checked="" type="checkbox"/>
Short description	<input type="text"/>
<input type="button" value="Submit"/>	<input type="button" value="Test Connection"/>

3. The instance is now registered. Click **Use Dev Hub (<https://<yourparentinstance>.lab.service-now.com>)** to select it as the parent.



4. The child instance compares itself with the parent and produces a list of Local Changes. This Reconcile process takes longer if there are more versions your parent instance. In this lab, you have almost no existing versions; the process completes quickly.

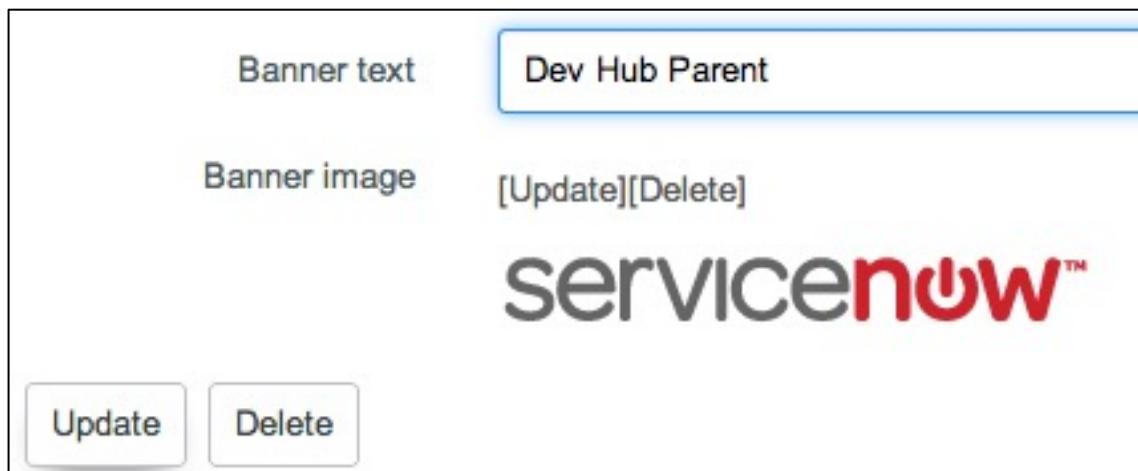


**Local Changes** are tracked relative to the current parent instance. If you switch parent instances, the new parent may already have your local changes. Thus a reconciliation process is needed whenever you select or switch parents.

## Customize Your Dev Hub (Parent) Instance

It is important to know on which instance you are working. Use System Properties and CSS to make that clear.

1. Log on to your **parent** instance.
2. Navigate to **My Company**.
3. Change banner text to **Dev Hub Parent**:



4. Select the **Update** button to save the change.
5. Navigate to **System Properties > CSS**.

6. Change **Banner text color** to **#D3242D** (the bright red used by ServiceNow).

**CSS Properties**

Please edit your changes and press Save

**Customization Properties for overriding values in the CSS file**

Banner text color

#D3242D 

7. Click **Save**, then refresh your browser window to see your changes reflected in the banner.



## Customize your Child Instance

Repeat the steps above on your child instance, using the title **Dev Hub Child** and the color **#339933**.

1. Log on to your **child** instance.
2. Navigate to **My Company**.
3. Change **Banner text** to the title **Dev Hub Child**, and click **Update**.
4. Navigate to **System Properties > CSS**.
5. Change **Banner text color** to **#339933**, click **Save**, and refresh your browser once more.

## Understand Your Changes (Child Instance)

The changes made on your parent instance and your child instance affects the Team Dashboard. This lab section explores the Dashboard changes.

1. On your child instance, navigate to **Team Development > Team Dashboard**.

The screenshot shows the Team Dashboard interface. On the left, there are two sections: 'Parent' (Dev Hub Parent) and 'Local' (teamdevam-dev). The 'Local' section shows 'Ready to Pull:' with 2 changes waiting to pull from parent. Below it, 'Local Changes:' shows 2 changes waiting to queue for push or ignore. On the right, a table lists system properties: 'css.version' and 'css.banner.description.color', both of which were changed by 'System Administrator' on March 9, 2015, at 15:24:05. The table includes columns for Record name, Type, Application, Changed by, Update set, and Updated.

2. Now under **Local Changes**, you see the system property changes you just made. You also see in the left-side control panel some interesting controls and notifications.

The screenshot shows the Team Dashboard with several callouts highlighting different parts of the interface:

- Parent dev hub**: Shows 'Ready to Pull:' with 409 changes waiting to pull from parent.
- Local empjnelson**: Shows 'Local Changes:' with 1423 changes waiting to queue for push or ignore.
- Action Buttons**: Callouts point to the 'Pull', 'Push...', and 'Refresh' buttons.
- Parent instance information and status**: Callout points to the 'Actions to change your current parent, or trigger a reconcile against the current parent' section.
- Number of changes to parent since list reconcile**: Callout points to the 'Number of changes to parent since list reconcile' section.
- Actions to pull and push changes**: Callout points to the 'Actions to pull and push changes' section.
- Local instance information and status**: Callout points to the 'Local instance information and status' section.
- Number of changes not found on Parent**: Callout points to the 'Number of changes not found on Parent' section.

As a result of your System Properties customizations on both the child and the parent instances, we see both local changes and remote changes are ready to pull.

## Lab Success Verification

- ✓ You should now have your instances configured with your Dev Hub parent and are now ready to start developing in parallel.
- ✓ You have a few changes ready to pull from your parent to your child and a few local changes available to push from your child to your parent.
- ✓ You understand the basic features and controls of the Team Dashboard and are ready to do some actual work!

## Lab Goal

This lab explains how to **back out** changes you do not wish to keep.

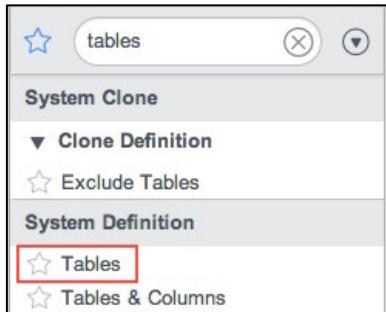
## Lab 2 Back Out Unwanted Changes

### Create a Field (Child Instance)

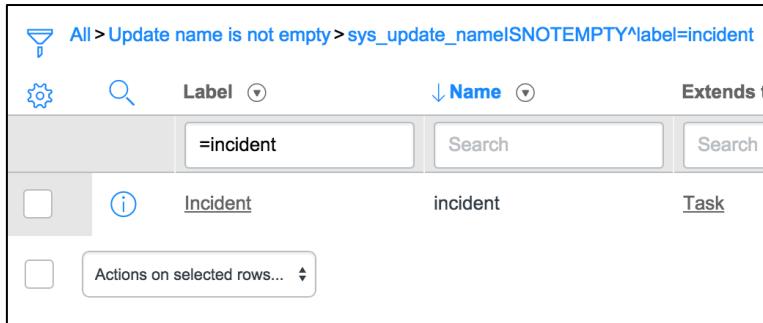
In this example, add a new True/False field to the Incident form to identify showstopper incidents.

**Note:** You will be purposely be making a mistake here.

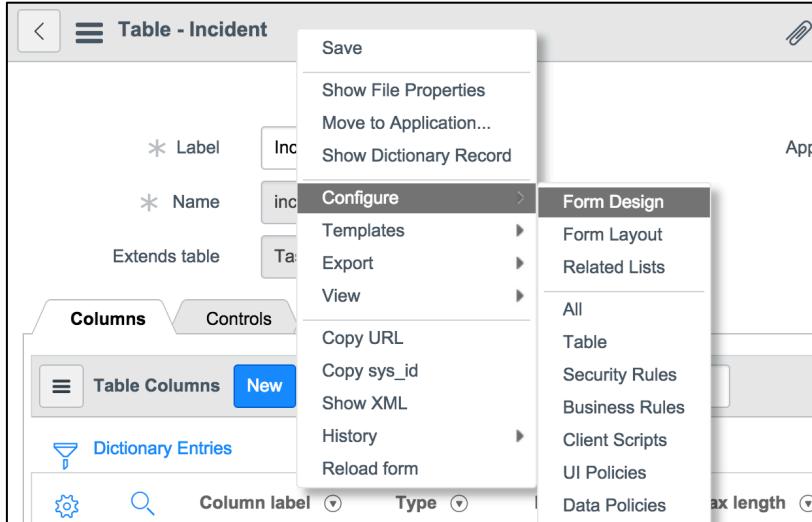
1. On your **child** instance, navigate to **System Definition > Tables**.



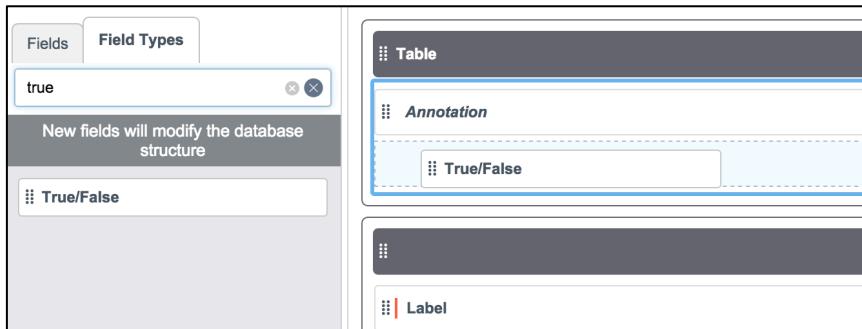
2. Locate the **Incident** table in the list by filtering by label. Open the Incident table form by clicking its name.



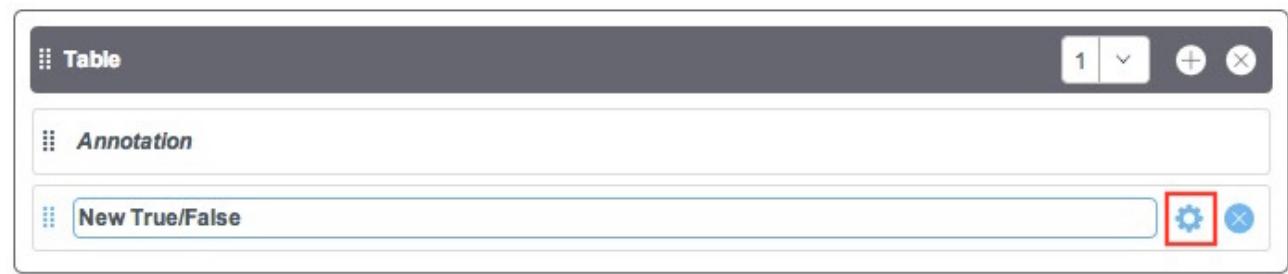
3. On the table form, right-click the header and select **Configure > Form Design**.



4. Under **Field Types**, enter **true** and drag and drop a **True/False** field under the **Annotation** field



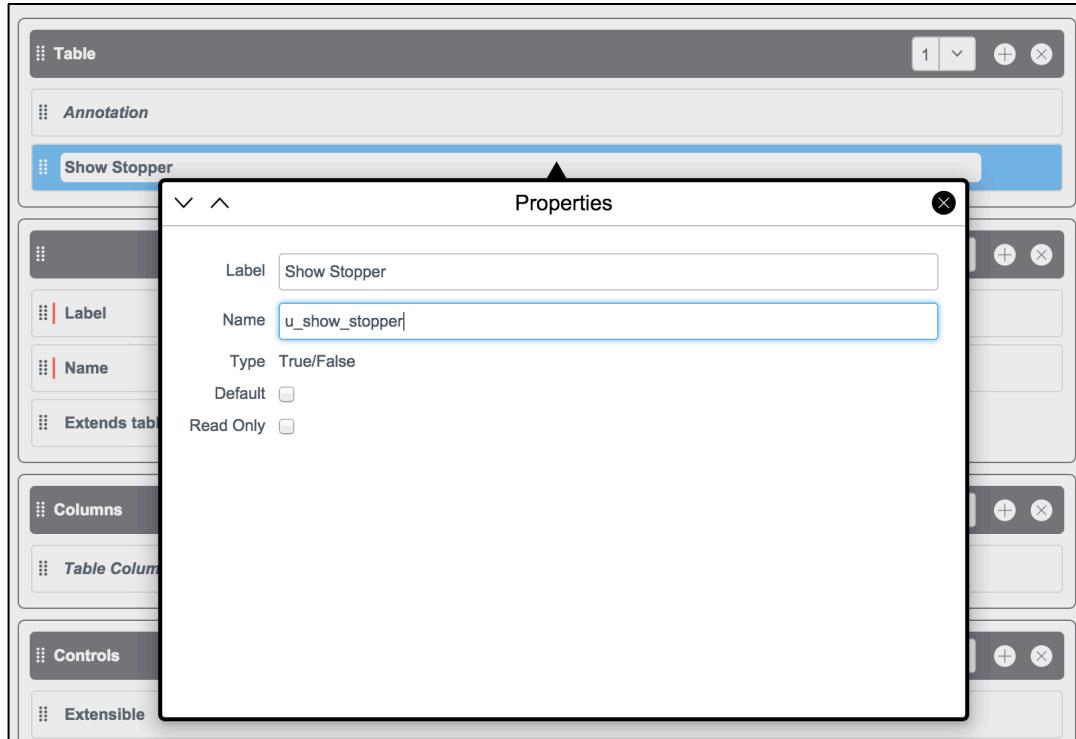
5. Click the **Cog Wheel** to personalize the new True/False field.



6. Edit the True/False field with the following:

**Label: Show Stopper**

**Name: u\_show\_stopper**



7. Close the properties dialog by clicking the X in the upper right corner and then click Save.
8. At this point, you realize that instead of customizing the Incident table you've customized the system dictionary – whoops! Continue with the remaining steps in the lab to undo this mistake.

9. Navigate to **Team Development > Team Dashboard**. Find the changes just made in the **Local Changes** list.

All > State = New > Remote Instance = dev hub

	Record name	Type	Application	Changed by	Update set	Updated
	Search	Search	Search	Search	Search	Search
<input type="checkbox"/>	Table	Form Layout	Global	System Administrator	Default	03-02 10:57
<input type="checkbox"/>	Table	Form Layout	Global	System Administrator	Default	03-02 10:57
<input type="checkbox"/>	Table.u_show_stopper	Dictionary	Global	System Administrator	Default	03-02 10:56
<input type="checkbox"/>	Table.Show Stopper	Field Label	Global	System Administrator	Default	03-02 10:56

10. If there are more changes than needed, sort the list by clicking the **Updated** column to display most recent updates first. Then right-click the cell that has a **Type of System Property** and select **Filter Out**. (Or filter out changes you wish to keep by other criteria.)

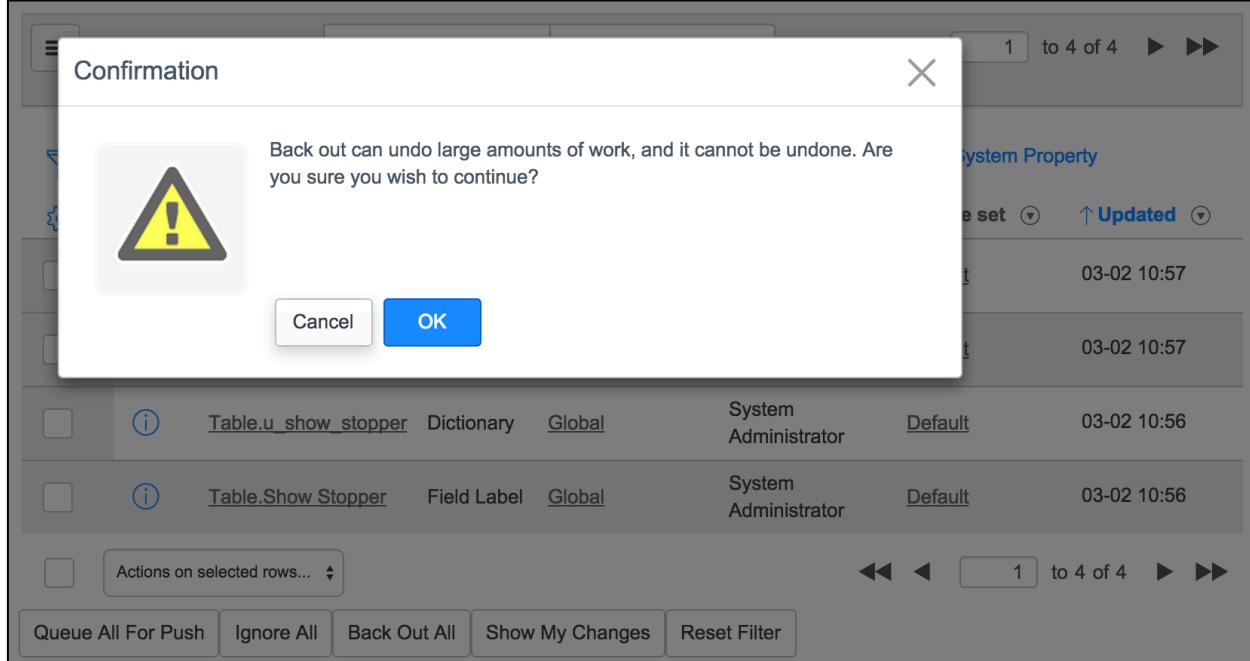
All > State = New > Remote Instance = dev hub > Version Updated on 2015-03-02

	Record name	Type	Application	Changed by	Update set	Updated
	Search	Search	Search	Search	Search	Search
<input type="checkbox"/>	Table	Form Layout	Global	System Administrator	Default	03-02 10:57
<input type="checkbox"/>	Table	Form Layout	System	Default	Default	03-02 10:57
<input type="checkbox"/>	Table.u_show_stopper	Dictionary	Show Matching	Filter Out	Default	03-02 10:56
<input type="checkbox"/>	Table.Show Stopper	Field Label	Copy URL to Clipboard	Default	Default	03-02 10:56
<input type="checkbox"/>	css.version	System Property	Copy sys_id	Default	Default	03-02 10:15
<input type="checkbox"/>	css.banner.description.color	System Property	Assign Tag	Default	Default	03-02 10:15

11. When the local change list shows only those changes you wish to back out, click **Back Out All**.

**Warning: Back Out All will back out all changes displayed in list view regardless of whether they are “checked” or not. Confirm list is only displaying changes you want to back out.**

When prompted to confirm, double-check that you have indeed filtered for the changes you wish to back out. This cannot be undone. When ready, click **OK** on the Confirmation dialog.



12. Navigate to **System Definition > Tables**.

13. Right-click the list header and select **Configure > List Layout**. Verify that the **Show stopper** field accidentally added to table is no longer present.

**Back Out Change** does more than just revert versions. It determines the last version pulled from or pushed to your parent instance, if any, and revert to that. Or, if no such version can be found, the record and its version history are completely erased.

## Lab Success Verification

- ✓ You learned how to back out one or more local changes.

## Lab Goal

This lab explains how to work with changes and resolve change issues. In the lab, you complete the following actions:

- Learn how to **pull** changes from your parent
- **Resolve collisions** when other people change the same files
- **Ignore** local changes that you never want to push upstream.
- **Create** an application you can develop in parallel

## Lab 3 Pulling Changes

### Pull Changes from your Dev Hub (Child Instance)

Before getting started on your application, we will pull changes from your parent and resolve any collisions we find.

1. From **child** instance navigate to **Team Development > Team Dashboard**.
2. In the Team Dashboard control panel, click the **Pull** action. The standard progress worker dialog appear to report the pull status.



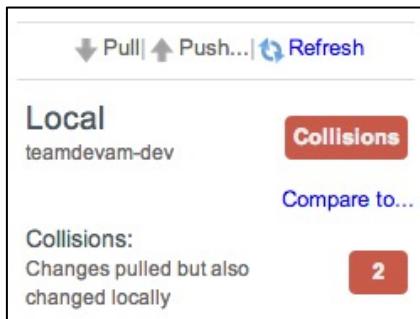
- When the progress worker finishes, you should see a message, explaining that while the pull itself was successful, you encountered some anomalies. Your message should contain wording similar to the example (numbers will vary based on the actual changes you have made on the instance).

*Pulled 10 versions of 8 files from parent but loaded 0 changes. The remaining changes were not loaded for the following reasons: 8 were private properties, 2 were collisions. Please resolve any collisions before continuing.*

This message means:

- Ten new versions were detected on your parent since the last time you pulled or reconciled. Those modifications were made to distinct files, which means the same record was modified multiple times, and you pulled both the current and the historical versions.
- Of all ten versions, none were actually loaded. (**Loaded** means the change was actually committed to the target table in the local instance.)
- Of the remaining, two versions resulted in collisions. This makes sense, since you modified the same two system properties on both child and parent instances. These are not loaded unless you decide to do so while resolving each collision (which we will do later).
- The remaining three were all new historical versions, inserted locally but not loaded. Only current versions can be loaded. History versions are pulled and stored purely for historical reasons.

- Click the **Team Dashboard** related link to return to the dashboard. Your control panel now shows the collisions to resolve.



5. The red **Collisions** box is a link. Click it to navigate to the list of colliding versions.

6. Right-click on one of the versions and select **Resolve Collision**

Type	Version	Record name	Application	Source
System Property	sys_properties_7619c651d7013100ea9e47eb5...	css.version	Global	Push or Pull: Pulled from Dev Hub Parent
System Property	sys_properties_0fd0370nc611228e019a8978b...	css.banner.description.color	Global	Push or Pull: Pulled from Dev Hub Parent

7. In the comparison popup window both versions appear: the pulled version from the parent, and the local version. In order to resolve this collision, click either the **Use pulled version** or the **Use local version** buttons.

Pulled version	Use pulled version	Local version	Use local version
sys_updated_on 2015-03-09 22:20:09		2015-03-09 22:24:04	
value #D3242D		#339933	

Since these are the system properties you customized to uniquely identify your local instance, you want to keep the local changes. Since you do not want the child instance to look like the parent, click **Use local version**.

8. There should be one remaining collision on the list. Since you already know what you want to do with it, without looking at them. In this case, you can use the choice list actions to resolve the collision without launching the comparison dialog. Select all rows and click **Use local version** from the list action menu.

The screenshot shows a software interface titled "Push and Pull Versions". At the top, there's a navigation bar with "Push and Pull Versions", "Go to", "Version Type", and a search bar. Below the navigation is a breadcrumb trail: "All > State = Collision > Push or Pull Remote Instance Sys ID = 6a6e29000c3231007f445c2b806ce11c". The main area is a table with columns: "Type" (System Property), "Version" (sys\_properties\_7619c651d7013100ea9e47eb5...), "Record name" (css.version), "Application" (Global), and "Source" (Push or Pull: Pulled from Dev Hub Parent). A context menu is open over the first row, listing options: Actions on selected rows..., Delete, Use Pulled Version, Use Local version (which is highlighted in blue), Create Application File, Assign Tag:, and New... . The bottom right corner of the interface shows a small circular icon with a question mark.

9. Navigate to **Team Development > Team Dashboard**, and look at the list of local changes.

You see the same two properties here because you chose to keep your local versions. However, not only do you *not* want to load these properties *from* your parent, you also do not want to ever *push* these properties *to* your parent. You also, however, do not want these appearing in your Local Changes list. In this case, choose to *ignore them*.

- Right click on 'System Property' and select Show Matching so that the two system properties (`css.banner.description.color` and `css.version`) are the only items in the list and click the 'Ignore All' button (although the screen shot shows the items checked, the buttons act on what shown in the list, regardless of whether they are checked or not).

The screenshot shows a 'Local Changes' interface. At the top, there are filters for 'Record name', 'Type', 'Application', 'Changed by', 'Update set', and a sorting column for 'Updated'. Below the filters, a table lists two entries:

	Record name	Type	Application	Changed by	Update set	Updated
<input checked="" type="checkbox"/>	<a href="#">css.version</a>	System Property	Global	System Administrator	Default	2015-03-09 15:24:05
<input checked="" type="checkbox"/>	<a href="#">css.banner.description.color</a>	System Property	Global	System Administrator	Default	2015-03-09 15:24:05

At the bottom of the table, there is a button labeled 'Actions on selected rows...' followed by 'Queue All For Push', 'Ignore All', 'Back Out All', 'Show My Changes', and 'Reset Filter'. The 'Ignore All' button is highlighted with a red box.

- Once the page reloads, you should now see the control panel appears with no incoming changes, no local changes, no collisions, and three ignored changes.

The screenshot shows the 'Control Panel' with the following sections:

- Ready to Pull:** Changes waiting to pull from parent (0)
- Local:** teamdevam-dev (OK button)
- Local Changes:** Changes waiting to queue for push or ignore (0)
- Ignored:** Changes you have decided not to push (2)

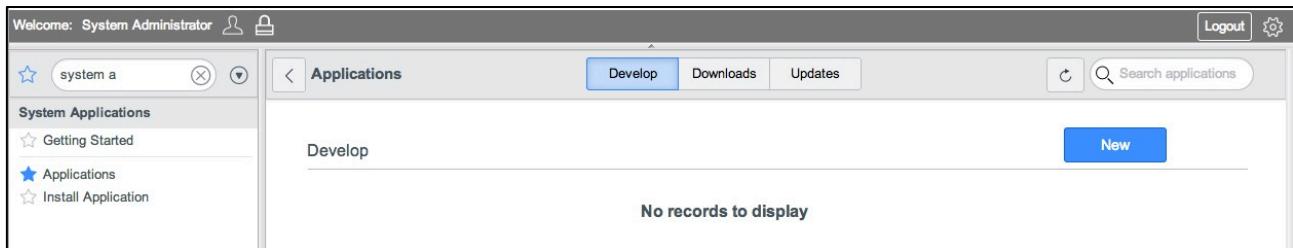
**Ignored changes** are not lost. If you ignore a change, it is moved to the Ignored state and appears in the Ignored tab in your dashboard.

If you change your mind, find the change in the Ignored list and click **Do Not Ignore** from the list action menu.

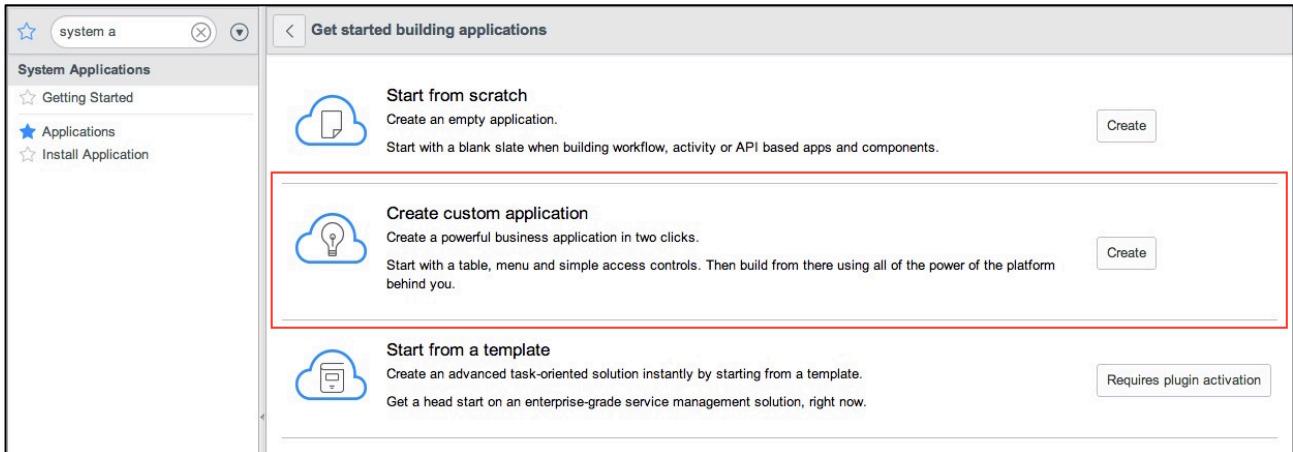
## Create a New Application in your Dev Hub (Parent Instance)

In this exercise, you will create a new application in the parent instance (DEV HUB) and then pull it down into your child instances.

1. Log on to your **parent** instance (DEBV HUB PARENT) with admin credentials.
2. On your Dev Hub **parent** instance, navigate to **System Applications > Applications**.
3. Click **new**.



4. Select **Create custom application**



5. Create an application called **Ideation** with a table Included that **Extends Table Task**.

The screenshot shows the 'Create Application' screen in the ServiceNow Application Creator. The application is named 'Ideation' with a scope of 'x\_ideation'. It includes a menu item 'Ideation' and a user role 'x\_ideation\_user'. The 'Create Table' toggle switch is turned on. Below this, the 'Table' section is configured with a label 'Ideation Table', name 'x\_ideation\_table', and module 'Ideation Table'. The 'Extends Table' field is set to 'Task'. There are also options for 'Extensible', 'Live Feed', and 'Auto-number', all of which are turned off. A 'Create' button is visible at the bottom left.

6. Click **Create** to save the application.
7. Find the new application table in the navigator by typing **Idea**.



**Adding or changing files upstream** has historically been frowned upon. If you make a change in production, for example, it is very important to make the same change in all of your sub-production instances as well. This is relatively expensive and error-prone if done manually or by creating an update set in production to be distributed elsewhere.

With Team Development, adding or changing files upstream is simple. Any change made upstream, such as creating the **Ideation** application, can be pulled to downstream instances with a single click as you will see below.

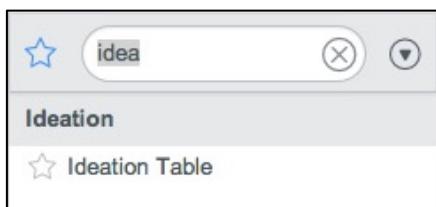
## Pull your Application into your Child Instances (Child Instance)

Now that the application is available in your Dev Hub Child, you can pull it into your child instance.

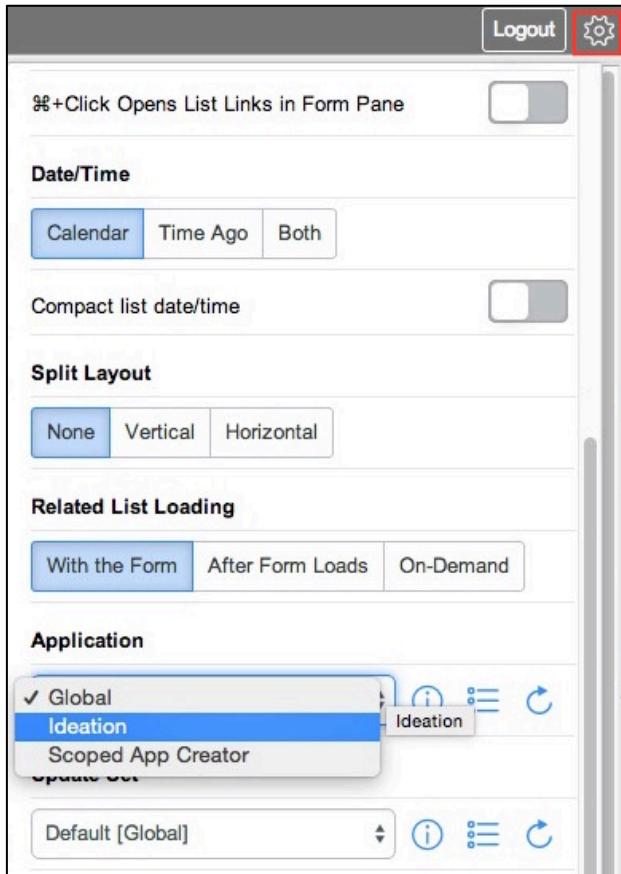
1. On your own **child** instance, navigate to **Team Development > Team Dashboard**.
2. Click **Pull** in the dashboard control panel.



3. When finished, click the **Close** button to return from the progress worker results screen. There should be no further changes waiting to pull.
4. Navigate to **Ideation > Ideation Table** to see the application table pulled to your child instance.



5. In the banner system menu, scroll to the bottom and select **Ideation** in the **Application** picker. This ensures all files you add going forward will be associated with your Ideation application.



## Lab Success Verification

- ✓ You learned how to **pull changes** made on your parent instance to your child instance with a single click.
- ✓ You understand how **collisions** occur and how to resolve them.
- ✓ You learned how and when to **ignore** local changes and how to prevent them from being pushed up to the parent.
- ✓ Your Ideation application, originally created on your parent Dev Hub is pulled, loaded, and functional on the child instance.

## Lab Goal

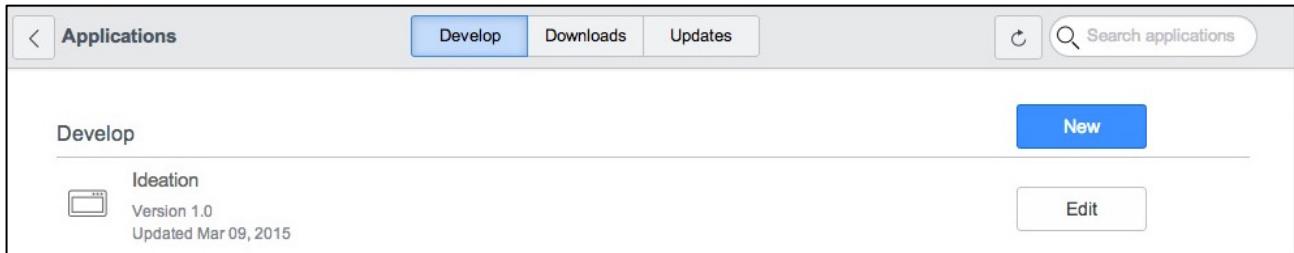
This lab explains how to make a few changes to an application in parallel on child instances and learn how to **push** them to the parent.

## Lab 4 Pushing Changes

### Learn How to Push Local Changes (Child Instance)

Read through the steps here to learn how to push changes then follow these steps.

1. Navigate to System **System Applications > Applications**.
2. Open **Ideation** application by clicking **Edit**.



3. Navigate to the ‘Application Menu’ tab/section and click into the Application Menu item with the Title ‘Ideation’ (The screenshot below shows a tabbed form. By default your instance will show an un-tabbed form. The names shown in the tabs will appear as sections in your form).

The screenshot shows a list of application menus. There is one entry named 'Ideation'.

Active	Order	Roles	Name	Updated
true		x_ideation_user	Ideation	2015-03-09 16:05:11

4. Add a module by clicking **New**.

The screenshot shows a list of modules. There is one entry named 'Ideation Table'.

Title	Table	Active	Filter	Order	Link type	Device type	Roles	Updated
Ideation Table [x_ideation_table]		true			List of Records		x_ideation_user	2015-03-09 16:05:19

5. Enter the values shown in the example.

This form has annotations - click  to toggle them - ([click here](#) to never show this again)

X

Title	Create New	Application	Ideation	
Application menu	Ideation	 		
Order				
Hint				

**Visibility**

Roles		Override application menu roles	<input type="checkbox"/>
Active	<input checked="" type="checkbox"/>		

**Link Type**

Select the type of link for the module or select separator to create a horizontal line. The fields below change depending on your selection.

Link type	New Record	 
* Table	Ideation Table [x_ideation_table]	 

6. Click **Submit** to save the module.
7. Navigate to **Team Development > Team Dashboard**.

8. The **Local Changes** list contains the changes just made, and possibly others. Right-click just over the “Create New” record name and select Show Matching so that only that record is shown in the list. Click **Queue All for Push**.

The screenshot shows a list titled "Local Changes" with a single record displayed. The record details are:

Record name	Type	Application	Changed by	Update set	Updated
Create New	Module	Ideation	System Administrator	Default	2015-03-09 16:21:47

At the bottom of the list, there are several buttons: "Actions on selected rows...", "Queue All For Push" (which is highlighted with a red box), "Ignore All", "Back Out All", "Show My Changes", and "Reset Filter".

The Local Changes list has a few buttons that operate on large sets of changes: **Queue All For Push**, **Ignore All**, and **Back Out All**. These "All" actions operate against all records that match the current filter in the local changes list. If the filter matches more records than fit on one page of results, clicking these actions operates against all of them. To perform these actions on only a few records in the list, select the check box and use the **List Menu** action to perform these.

9. Now there is at least one change queued for push. The **Push (blue up arrow [↑])** action is enabled on the dashboard control panel. Click **Push** to proceed.

The dialog box displays the following information:

- Top buttons: Pull | Push... | Refresh
- Section: Local  
Demo Server
- Action button: OK
- Text: Compare to...
- Section: Ready to Push:  
Changes ready to push to parent
- Count: 1

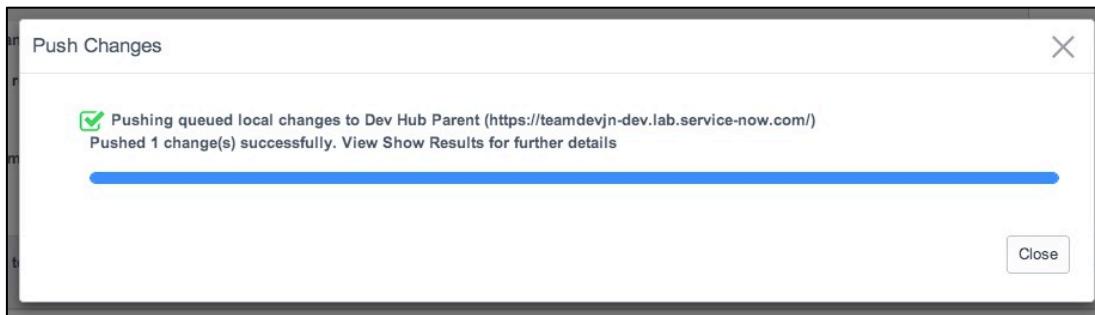
10. On the **Push** screen, enter this name to identify the change set: **Ideation - Update 1**.

The screenshot shows the 'Push Changes' interface. At the top, it says 'Pushing changes to: https://teamdevjn-dev.lab.service-now.com/'. Below that, 'Changes ready to push: 1' is displayed. A table lists one change with the following details:

Record name	Type	Application	Changed by	Update set	Updated
Create New	Module	Ideation	System Administrator	Default	2015-03-09 16:21:47

At the bottom of the screen, there are 'Push Changes' and 'Cancel' buttons.

11. Click **Push Changes**. When the progress worker screen shows complete, click **Close** to check the final status of the push in the **Team Dashboard**.



12. Check the **Pushes and Pulls** related list to see the individual states and possible log messages.

The screenshot shows the ServiceNow interface with the following components:

- Left Panel (Parent Instance):** Shows "Ready to Pull" status with 0 changes waiting to pull from parent, and "Pushes and Pulls" section with 2 ignored changes.
- Top Bar:** Local Changes, Go to, Version Updated dropdown, Search input.
- Search Bar:** Record name, Type, Application, Changed by, Update set, Updated dropdown, and search fields.
- Message Area:** No records to display.
- Pushes and Pulls List:** A table with columns: Created, Type, Name, Remote Instance, URL, Stage, Message, and Created by. It shows one entry: 2015-03-09 16:29:08 Push, Ideation - Update 1, Dev Hub Parent, https://teamdevjn-dev.lab.service-now.com/, Complete, Complete, admin.

13. Log onto your Parent Dev Hub instance and confirm that the new ‘Create New’ module appears under the ‘Ideation’ application. You may need to refresh your navigator (left-click on drop-down arrow next to the navigation text area and select Refresh Navigator).

## Lab Success Verification

- ✓ You learned how to track local changes and **queue them for a push**.
- ✓ You know how to **push changes** you have queued on your child instance with a single click.
- ✓ Your Ideation application is up to date and live across your parent and child instances.

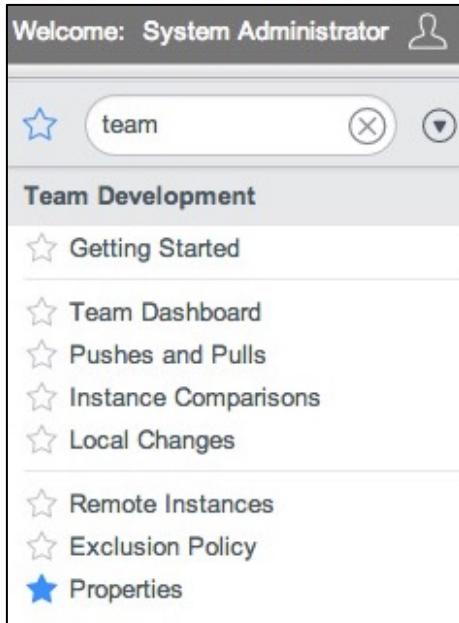
## Lab Goal

To add more protection from promoting unwanted changes it is possible to enable an approval workflow on your parent to support reviewing, approving and rejecting any push.

## Lab 5 Code Review

### Enable Code Review on Parent (Parent Instance)

1. Log on to your **parent** Dev Hub instance.
2. Navigate to **Team Development > Properties**.



3. Note: The **Parent** instance is still in the *Ideation* Scope. This causes records in the *Global* Scope to be read-only. Select **Switch to Global** in order to edit this property.

This record is in the [Global application](#), but [Ideation](#) is selected in your application picker. [Switch to Global](#).

**Team Development Properties**

Please edit your changes and press Save

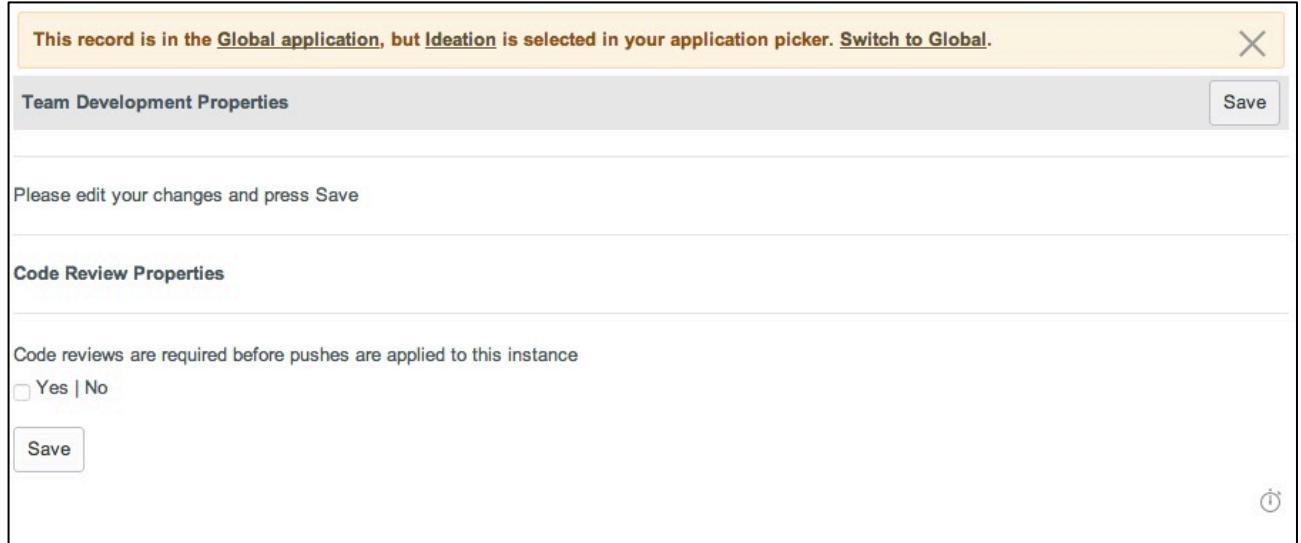
**Code Review Properties**

Code reviews are required before pushes are applied to this instance

Yes | No

**Save**

(i)



4. Check the box to enable Code Review on this parent instance, and click **Save**. This ensures that any push made to this parent is staged pending approval.

**Team Development Properties**

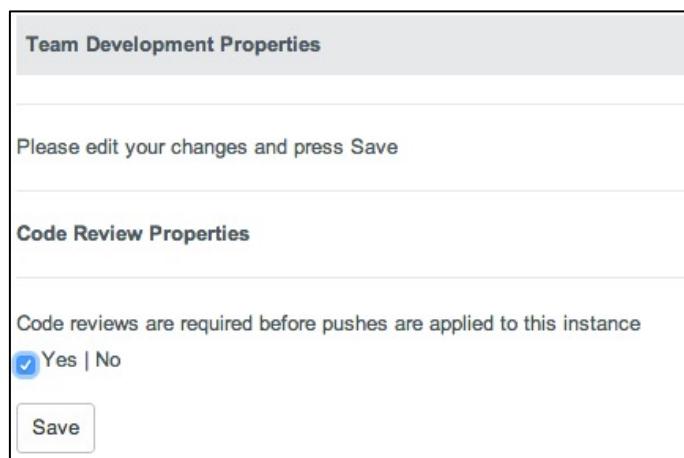
Please edit your changes and press Save

**Code Review Properties**

Code reviews are required before pushes are applied to this instance

Yes | No

**Save**



5. Navigate to **User Administration > Groups**.

- Locate the Team Development Code Reviewers group by name and click to open it.

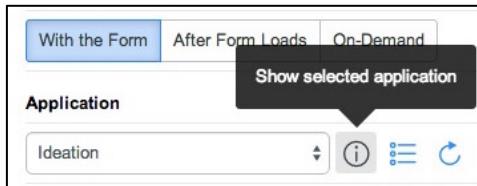
The screenshot shows a list of groups in the ServiceNow interface. The top navigation bar includes 'Groups' (selected), 'New', 'Go to', 'Name' (dropdown), and a search bar. Below the header, there's a filter bar with 'All > Name contains Team'. The main table has columns for Name, Description, Active, Manager, Parent, and Updated. One row is selected, showing 'Team Development Code Reviewers' with a description of 'Review, approve and/or reject the code p...', 'Active' status, and 'true' under Manager. The 'Updated' field shows '2013-12-01 15:47:36'. At the bottom, there are buttons for 'Actions on selected rows...' and navigation icons.

- Add the User David Loo to the group by clicking 'Edit...' next to the 'Group Members' label, and double-clicking on his name so it appears on the right-hand-side, then clicking Save. The default code review workflow relies on this group to distribute approval requests for and notify the approvers. (System Administrator is added by default.)

## Add Dictionary Override for State (Child Instance)

The final task is to add an override to Task **State** field.

- Log on to your **child** instance with admin credentials.
- Navigate to the Ideation application form using the application picker, ensuring that the **Ideation** application is selected.



- In the related **Tables** list, click the **Ideation table** to open it.

The screenshot shows the 'Tables' list view. The top navigation bar includes 'Tables (1)', 'Roles (1)', 'Application Menus (1)', 'Application Files (17)', 'Dependencies (1)', 'Update Sets (1)', 'Tables' (selected), 'New', 'Go to', 'Name' (dropdown), and a search bar. Below the header, there's a filter bar with 'Application = Ideation'. The main table has columns for Label, Name, Extends table, Extensible, and Updated. One row is selected, showing 'Ideation Table' with a label of 'x\_ideation\_table', 'Extends table' set to 'Task', 'Extensible' set to 'false', and 'Updated' on '2015-03-09 16:05:12'. At the bottom, there's a button for 'Insert a new row...'.

4. In the **Table Columns** list, filter for and open the **State** column.

The screenshot shows the 'Table Columns' list interface. At the top, there are tabs for 'Columns', 'Controls', and 'Application Access'. Below the tabs is a search bar with 'Search for text' and a 'Search' button. On the right side of the search bar are navigation icons and a status message '1 to 1 of 1'. The main area displays a table with one row. The first column is labeled '=State' with a dropdown menu icon. The second column is labeled 'Search' with a dropdown menu icon. The third column is labeled 'Search' with a dropdown menu icon. The fourth column is labeled 'Search' with a dropdown menu icon. The fifth column is labeled 'Search' with a dropdown menu icon. The sixth column is labeled 'Search' with a dropdown menu icon. The seventh column is labeled 'Display' with a dropdown menu icon. The row itself has columns for 'Column label', 'Type', 'Reference', 'Max length', 'Default value', and 'Display'. The 'Column label' column contains '=State', 'Type' contains 'Integer', 'Reference' contains 'Search', 'Max length' contains '40 1', 'Default value' contains 'false', and 'Display' contains 'Search'. There is also a small info icon next to 'State' in the 'Column label' column. A plus sign '+' is at the bottom left, and a message 'Insert a new row...' is at the bottom right.

Note: You will see the following message:

This record is in the [Global application](#), but [Ideation](#) is selected in your application picker. [Switch to Global](#).



You may be inclined to switch to Global. **DO NOT**. You want this dictionary override contained in your application scope.

- In the related **Dictionary Overrides** list, click **New** and enter the values below. This ensures that new Idea records default their state to **Pending** [-5] instead of **Open**.

**Table: Ideation Table [x\_ideation\_table]**

Override default value:

Default value: -5

Override the dictionary settings for the 'Task State' field in extended tables  
 Dictionary settings for Task State are:

Reference qualifier:  
 Dependent  
 Attributes:  
 Default value: 1  
 Calculation:  
 Mandatory: false  
 Read only: false  
 Display: false

Task display column: number

Application	Ideation	(i)
* Base table	Task [task]	
* Table	Ideation Table [x_ideation_table]	▼
Column name	state	
Override reference qualifier	<input type="checkbox"/>	
Override dependent	<input type="checkbox"/>	
Override attributes	<input type="checkbox"/>	
Override default value	<input checked="" type="checkbox"/>	
Default value	-5	

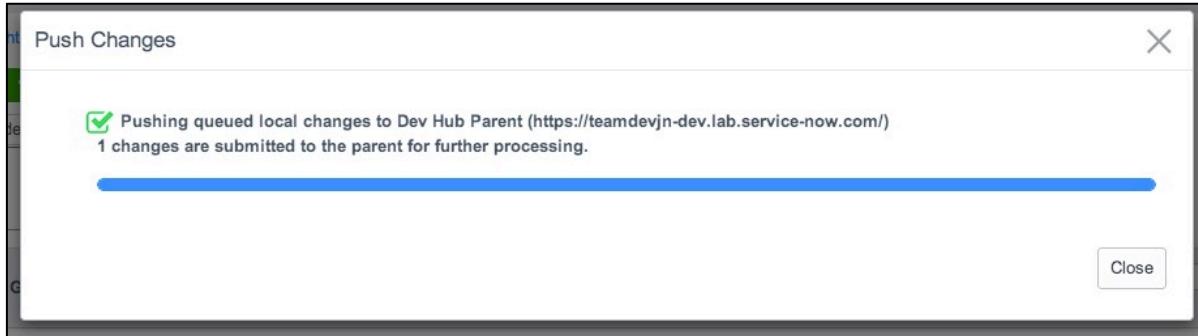
- Click **Submit**.

- Push the changes as described in Lab 4, Steps 8-12.

All > State = New > Remote Instance = Dev Hub Parent

Record name	Type	Application	Changed by	Update set	Updated
state	Dictionary Entry Override	Ideation	System Administrator	Default	2015-03-09 16:48:27
<input type="checkbox"/> Actions on selected rows... ▾ <span style="float: right;">◀◀ ▶▶ 1 to 1 of 1</span>					
<input type="button" value="Queue All For Push"/> <input type="button" value="Ignore All"/> <input type="button" value="Back Out All"/> <input type="button" value="Show My Changes"/> <input type="button" value="Reset Filter"/>					

- With code review enabled on the parent, the progress worker status message will be slightly different:



The pushed changes are now staged on the parent and are not committed until someone approves them.

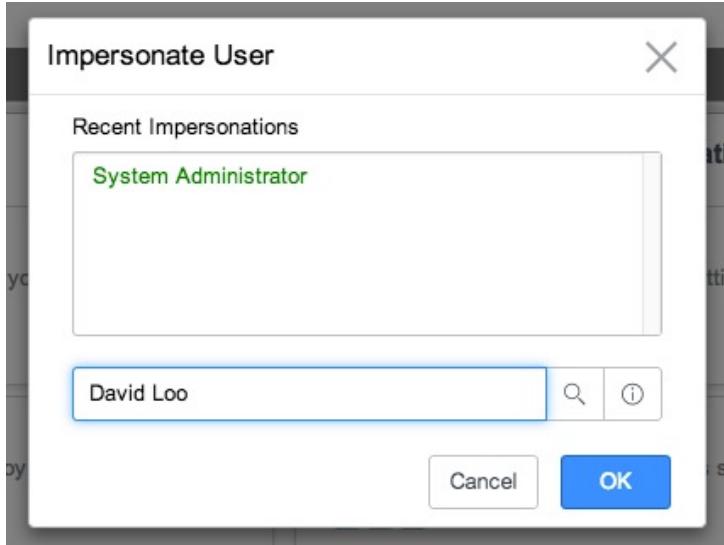
- Click **Close** to return to the **Team Dashboard** and click the **Pushes and Pulls** tab. The push history is listed with the stage *Awaiting Code Review*.

The screenshot shows the 'Pushes and Pulls' tab of the Team Dashboard. The table has columns for Created, Type, Name, Remote Instance, URL, Stage, Message, and Created by. One row is highlighted, showing a push from '2015-03-09 16:51:53' to 'Ideation - Update 2' at 'Dev Hub Parent' with the URL '<https://teamdevjn-dev.lab.service-now.com/>'. The 'Stage' column shows 'Awaiting Code Review' with a red border around it. The 'Message' column shows 'In progress'. The 'Created by' column shows 'admin'.

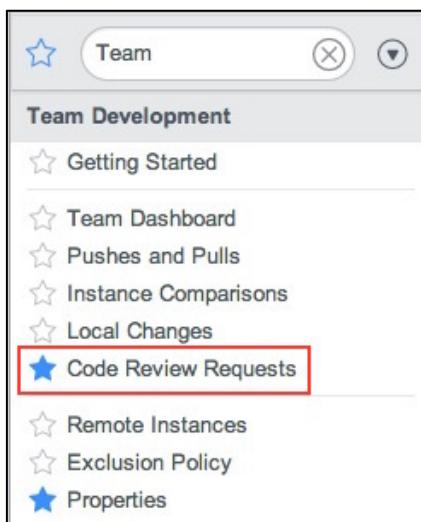
Created	Type	Name	Remote Instance	URL	Stage	Message	Created by
2015-03-09 16:51:53	Push	Ideation - Update 2	Dev Hub Parent	<a href="https://teamdevjn-dev.lab.service-now.com/">https://teamdevjn-dev.lab.service-now.com/</a>	Awaiting Code Review	In progress	admin

## Review and Approve the Push (Parent Instance)

1. Log on to your **parent** Dev Hub instance with admin credentials.
2. Impersonate User **David Loo**.



3. Navigate to **Team Development > Code Review Requests**.

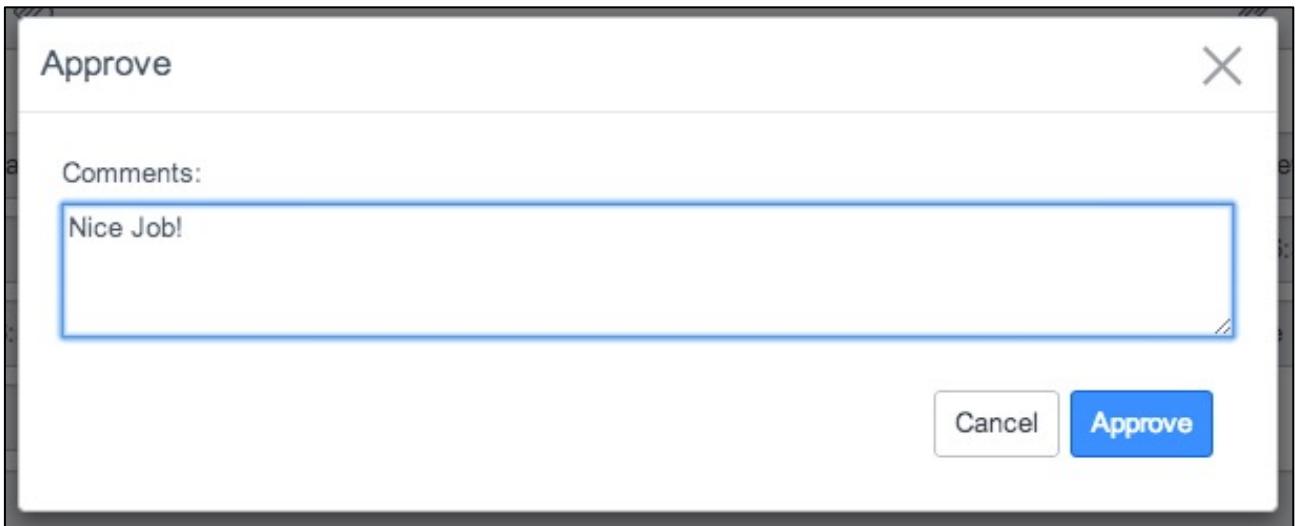


4. Open the push history record awaiting code review.

The screenshot shows a ServiceNow interface for managing pushes and pulls. The top navigation bar includes 'Pushes and Pulls', 'Go to', 'Name', and a search bar. Below the header is a breadcrumb trail: 'All > Stage = Awaiting Code Review > Type = Push Commit'. The main area is a grid with columns: Created, Type, Name, Remote Instance, URL, Stage, Message, and Created by. A single row is selected, showing the following details:

Created	Type	Name	Remote Instance	URL	Stage	Message	Created by
2015-03-09 16:51:53	Push Commit	Ideation - Update 2	teamdevam-dev	<a href="https://teamdevam-dev.lab.service-now.com:443">https://teamdevam-dev.lab.service-now.com:443</a>	Awaiting Code Review	In Progress	admin

5. Review the related changes as needed and click **Approve**. Enter an encouraging comment in the approval dialog and again click **Approve**.



- Within **parent** instance, navigate to **Team Development > Pushes and Pulls** and locate the approved push by name or created date. Confirm that the stage is now **Complete**.

The screenshot shows the 'Pushes and Pulls' list view in the ServiceNow interface. The table has columns for Created, Type, Name, Remote Instance, URL, Stage, Message, and Created by. Two rows are visible:

Created	Type	Name	Remote Instance	URL	Stage	Message	Created by
2015-03-09 16:29:08	Push Commit	Ideation - Update 1	teamdevam-dev	<a href="https://teamdevam-dev.lab.service-now.com:443">https://teamdevam-dev.lab.service-now.com:443</a>	Complete	Successfully pushed 1 versions to parent...	admin
2015-03-09 16:51:53	Push Commit	Ideation - Update 2	teamdevam-dev	<a href="https://teamdevam-dev.lab.service-now.com:443">https://teamdevam-dev.lab.service-now.com:443</a>	Complete	Successfully pushed 1 versions to parent...	admin

Below the main list is a 'Push and Pull Versions' detail view for the second push. It shows a single row with state 'Push Committed' and a 'Dictionary Entry Override' field.

- Navigate to the **Ideation** Application and click **Create New**. Confirm that the **State** field is now set to Pending by default.
- Log back on to your **child** instance from where you pushed the changes. Navigate to **Team Development > Pushes and Pulls** and locate the approved push by name or created date. Confirm that the stage is also now listed as **Complete**, and thus synchronized with the parent.

The screenshot shows the 'Pushes and Pulls' list view on a child instance. The table has columns for Created, Type, Name, Remote Instance, URL, Stage, and Message. One row is visible:

Created	Type	Name	Remote Instance	URL	Stage	Message
2015-03-09 16:51:53	Push	Ideation - Update 2	Dev Hub Parent	<a href="https://teamdevin-dev.lab.service-now.com/">https://teamdevin-dev.lab.service-now.com/</a>	Complete	Complete

Below the main list is a 'Push and Pull Versions' detail view for the push from the parent instance. It shows a single row with state 'Pushed' and a 'Dictionary Entry Override' field.

9. Open the record and view the Reviews related list to see any comments from the review process.

Approver Name	Approver ID	Decision	Comment
David Loo	david.loo	approved	Nice Job!

## Lab Success Verification

- ✓ You configured your parent instance to enforce approval before committing a change set using Code Review.
- ✓ You implemented a change, pushed it to your parent and approved it before it was committed to the instance.

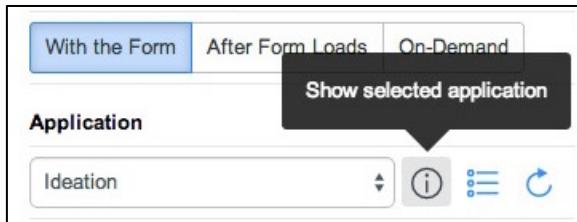
## Lab Goal

Team Development operates apart from Update Sets, but integrates with them in some useful ways. This lab covers how you can use local update sets to organize your local changes, and explains how update sets are generated during a push.

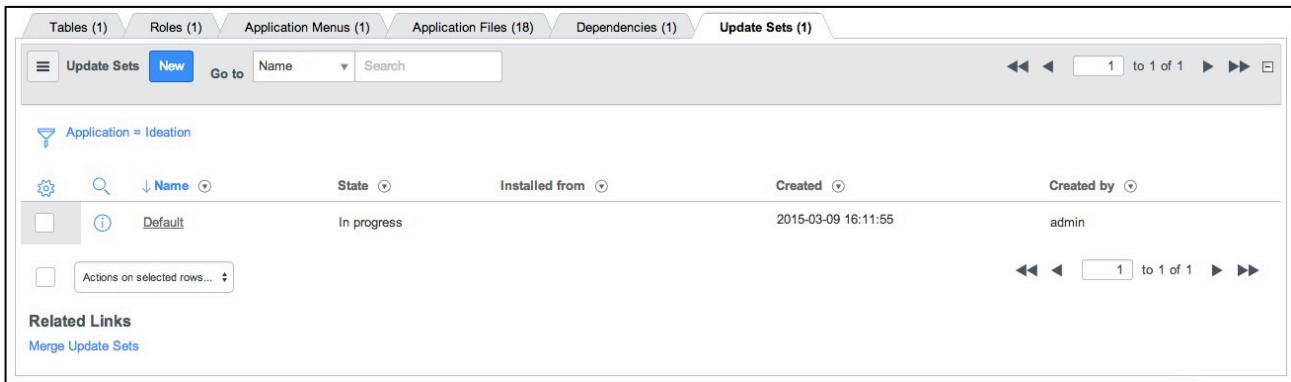
## Lab 6 Update Sets in Team Dev

### Create a Local Update Set (Child instance)

1. Log on to your **child** instance with admin credentials.
2. Navigate to the Ideation application form using the application picker, ensuring that the **Ideation** application is selected.



3. Click the related list **Update Sets**. Click **New**.



- Enter the name **Ideation Patch 1**, and click **Submit and Make Current**.

Name	Ideation Patch 1
State	In progress
Release date	<input type="text"/>
Description	<input type="text"/>
<input type="button" value="Submit"/> <input type="button" value="Submit and Make Current"/>	

- Now, make a change or addition to the application. Some ideas:

*Add a UI Policy to make the field Short Description mandatory.*

*Add a UI Action on your Idea form called **Submit Another Idea** that opens a new Idea form.*

If you are not familiar with either of these customizations you can do the following: Create a new Ideation Module by following Lab 4 Instructions 1 through 6 but in Step 5, specify the Title as 'Another Create New' instead of 'Create New'.

- When finished with the changes, navigate to **Team Development > Team Dashboard** and look at the **Local Changes** list.

Record name	Type	Application	Changed by	Update set	Updated
short_description	UI Policy Action	Ideation	System Administrator	Ideation Patch 1	2015-03-09 17:10:33
Make Short Description Mandatory	UI Policy	Ideation	System Administrator	Ideation Patch 1	2015-03-09 17:10:19

The column **Update set** displays the update set this local change is associated with. This is a standard reference field that you can use to sort and filter the list.

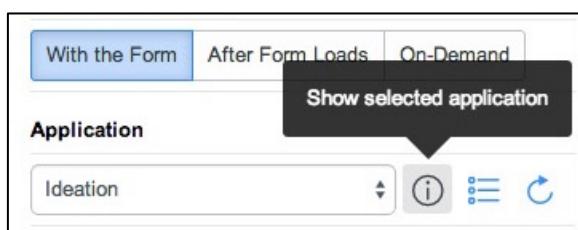
7. Right-click one of the cells in the **Update set** column referencing the update set, and select **Show Matching**. This ensures that only those updates associated with that update set are listed.
8. **Click Queue All For Push**.
9. **Click Push** in the dashboard control panel. Enter the name **Ideation 2.0** with a comment related to the change you made.

Record name	Type	Application	Changed by	Update set	Updated
<a href="#">short_description</a>	UI Policy Action	Ideation	System Administrator	Ideation Patch 1	2015-03-09 17:10:33
<a href="#">Make Short Description Mandatory</a>	UI Policy	Ideation	System Administrator	Ideation Patch 1	2015-03-09 17:10:19

10. **Click Push Changes**.

## Approve Change and Merge Update Sets (Parent Instance)

1. Log on to your **parent** Dev Hub instance with admin credentials.
2. Confirm that you are in the Ideation Application.



3. Navigate to **Team Development > Code Review Requests** and approve the pending push, as shown above.
4. Navigate to **System Update Sets > Local Update Sets**. Here you see one update set for each push you have made to your parent in this lab.

	Name	Application	State	Installed from	Created	Created by
<input type="checkbox"/>	Default	Global	In progress		2015-03-01 07:08:07	system
<input type="checkbox"/>	Default	Ideation	In progress		2015-03-09 16:05:11	admin
<input type="checkbox"/>	Ideation - Update 1	Ideation	Complete	https://teamdevam-dev.lab.service-now.co...	2015-03-09 16:29:22	admin
<input type="checkbox"/>	Ideation - Update 2	Ideation	Complete	https://teamdevam-dev.lab.service-now.co...	2015-03-09 16:58:28	admin
<input type="checkbox"/>	Ideation 2.0	Ideation	Complete	https://teamdevam-dev.lab.service-now.co...	2015-03-09 17:13:46	admin

Actions on selected rows... ▾

Related Links

Merge Update Sets

Always use **well-structured names** for your pushes, just as you should when creating update sets. The name coupled with good comments help anyone looking at either Push and Pull History or Update Set records understand what you were thinking at the time. You also have a much better chance that your change set will be approved.

5. At this point you no longer want to track such small change sets; you want to consolidate them into larger set. Click **Merge Update Sets**.

Ideation 2.0

Ideation

Complete

Actions on selected rows... ▾

Related Links

Merge Update Sets

6. Enter a suitable **Name** and **Description** for the new update set, and filter the **Update Sets** list to only include update set records where **Application = Ideation**.

**Note:** You can only merge update sets that have the same Scope.

The screenshot shows the 'Merge Update Sets' interface. At the top, there are fields for 'Name' (Ideation 3.0) and 'Description' (Revision to Data Policy and State fields). Below this, a message says 'Select the update sets you wish to merge by filtering them below'. A search bar at the top right shows the filter 'All > State != Ignore > Application = Ideation'. The main area is a grid of update sets with the following columns: Name, Application, State, Installed from, Created, and Created by. The grid contains four rows of data:

Name	Application	State	Installed from	Created	Created by
Default	Ideation	In progress		2015-03-09 16:05:11	admin
Ideation - Update 1	Ideation	Complete	https://teamdevam-dev.lab.service-now.co...	2015-03-09 16:29:22	admin
Ideation - Update 2	Ideation	Complete	https://teamdevam-dev.lab.service-now.co...	2015-03-09 16:58:28	admin
Ideation 2.0	Ideation	Complete	https://teamdevam-dev.lab.service-now.co...	2015-03-09 17:13:46	admin

At the bottom left are 'Merge' and 'Reset Filter' buttons.

7. Click **Merge**. When prompted to confirm, click **OK**. The new update set **Ideation 3.0** now contains all of the updates from the original update sets.
8. The original update sets should now be marked with **State** as **Ignore** and be empty.

## Lab Success Verification

- ✓ You used a local update set to keep your changes to the application organized.
- ✓ You learned that the system creates a completed update set on the parent instance for every change set pushed and committed; and you understand the importance of using good naming conventions when creating update sets and pushing changes in Team Development.
- ✓ You merged the completed update sets from all of your pushes made to your Ideation application into a single new update set. This allows you to consolidate related changes into one larger set that is easier to track and migrate.

## We Thank You For Playing

Those are the key components of Team Development in ServiceNow. We are building on these foundations and are striving to simplify the complicated world of code and customization management on our platform in the cloud.

We hope you enjoyed your time here, hope you learned more than you forgot, and hope you feel free to ask questions of any of us today or throughout the conference. Enjoy and cheers!