



University of Colorado

Boulder | Colorado Springs | Denver | Anschutz Medical Campus



## OnBase Guide – Workflow – FormPop Notifications

---

**Goal:** To configure workflow notifications using FormPop tokens to provide the recipient with a link to a Unity form

**Complexity Level:** Departmental Workflow Developers, Departmental Administrative Users

7/26/2023

---

## Table of Contents

Background .....	3
Prerequisites.....	3
Resources .....	4
OnBase Configuration Client - Add Document Type to User Group .....	4
OnBase Studio - Create the Workflow Notification .....	5
For public form modification (non-OnBase users): .....	7
For users authenticating to their own OnBase accounts: .....	8
Testing.....	14
Securing FormPop Content After Usage .....	15
Custom Action Recommendations to Control Visibility/Editing.....	15
Verifying Identity .....	15
Form Submission.....	17
Form is No Longer Available .....	18
FormPop and Document Locks .....	18
Document Type Setting.....	19
Script Option .....	20
Configuring Form Fields .....	20
Troubleshooting Form Scripts .....	21
Custom Action to Execute Script .....	21
Custom Action to Handle Script Result.....	22



---

## Background

This guide covers the process of setting up notification templates to include a FormPop URL to provide the recipient of the notification with a link directly to an existing form.

FormPop allows users to view and edit existing Unity Forms using a simplified Web Client viewer interface, without any extra OnBase functionality or access. This can be useful or necessary in a workflow process to collect additional information or obtain signatures or approvals within a form.

Typically, fixed service account credentials are used to allow access to the form, rather than an individual's OnBase account. This means the form will be available to anyone with the FormPop URL.

This means that beyond the simple setup of the notification and URL, there are important considerations for ensuring modifications can only be made at the appropriate time and by the intended audience.

FormPop does not allow users to create new forms, only to modify existing forms. To allow public submission of Unity forms, please refer to these guides instead:

- [Unity Form - Sharing for Public Use \(Basic URL\)](#)
- [Unity Form - Sharing for Public Use \(Portal\)](#)

NOTE: Only production FormPop URLs can be accessed publicly. For non-production, you'll need to be on a campus network or VPN to access forms via FormPop. Please keep this in mind when testing.

---

## Prerequisites

To create a FormPop Workflow Notifications, you must have OnBase Configuration and OnBase Studio installed. Refer to the [OnBase Client Guides](#) for installation instructions.

Only [OnBase Certified Administrators](#) have access to the OnBase Configuration Client. If you need assistance with the Configuration Client steps please contact [UIS\\_DM\\_SUPPORT@cu.edu](mailto:UIS_DM_SUPPORT@cu.edu).

---

## Resources

For more information, refer to:

- The Unity Form MRG
- The Pop Integrations topic from [OnBase Integrations Overview](#)
- The Workflow MRG

If you need assistance, please contact [UIS\\_DM\\_SUPPORT@cu.edu](mailto:UIS_DM_SUPPORT@cu.edu).

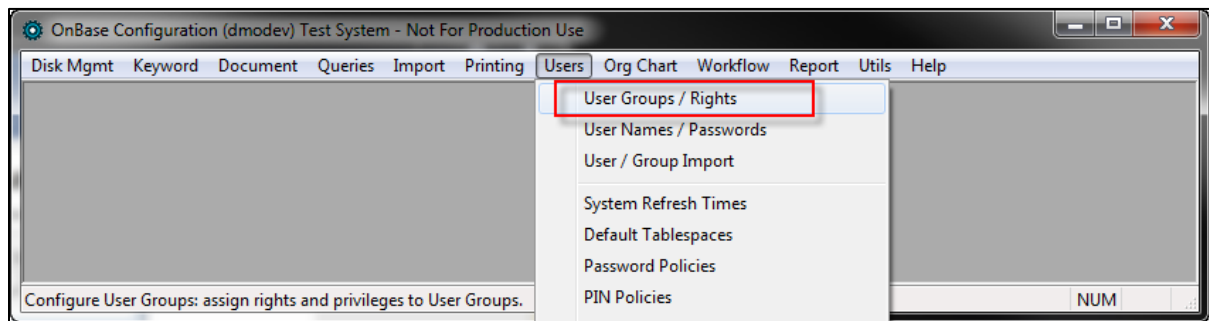
---

## OnBase Configuration Client - Add Document Type to User Group

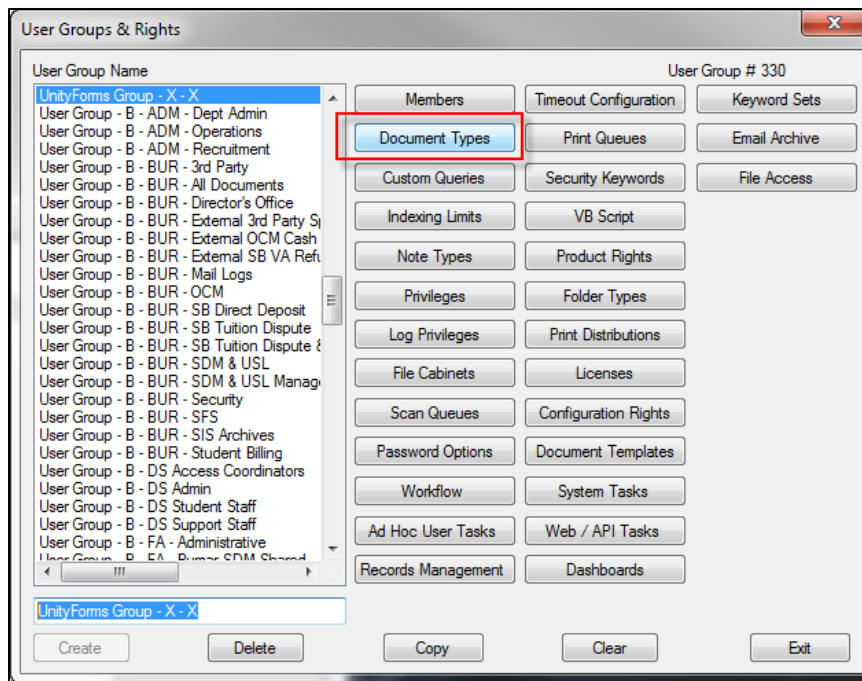
Add the document type that contains the Unity Form you wish to share via FormPop to the *UnityForms Group - X - X* group. This is necessary so that adequate permissions are available to view and modify the form without granting any additional access.

**NOTE:** This does not apply if you are having users authenticate to their own user accounts. In that case, you will still need to make sure the document type is available to the appropriate user groups but it does not need to be assigned to this group used for public submission/modification.

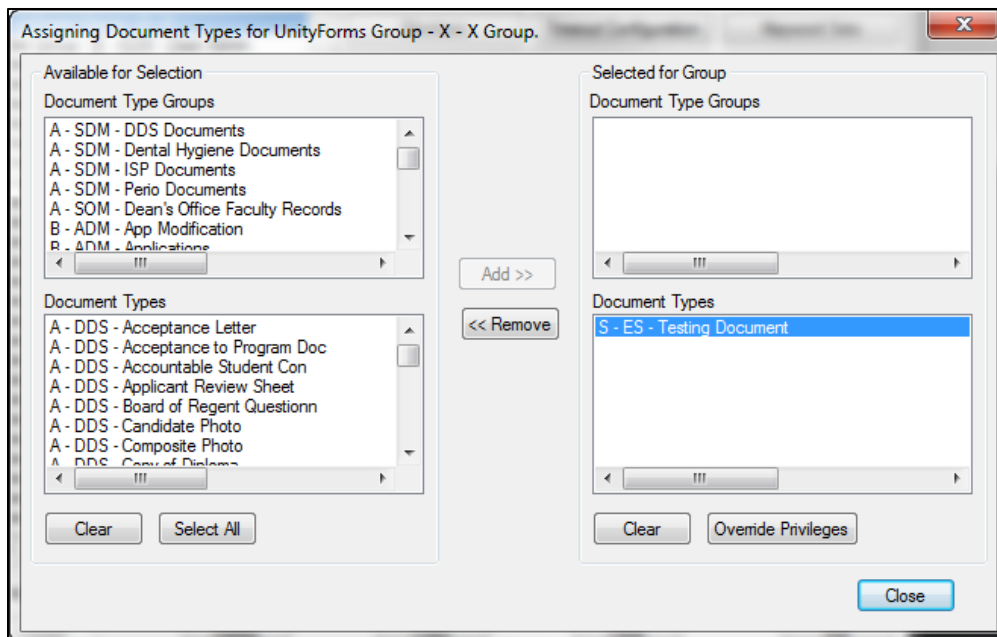
1. Log in to the OnBase Configuration Client
2. Select **Users** then **User Groups/Rights**.



3. Find the *UnityForms Group - X - X* group and select **Document Types**.



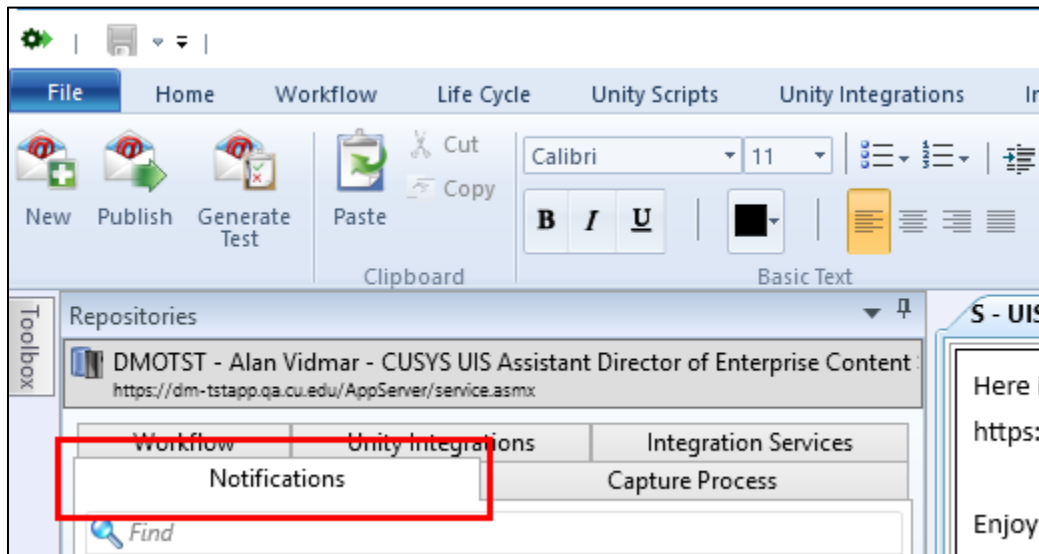
4. Add the appropriate document type to the group by finding it in the list of available document types and moving it to the list of selected document types using the **Add** button or double clicking.



## OnBase Studio - Create the Workflow Notification

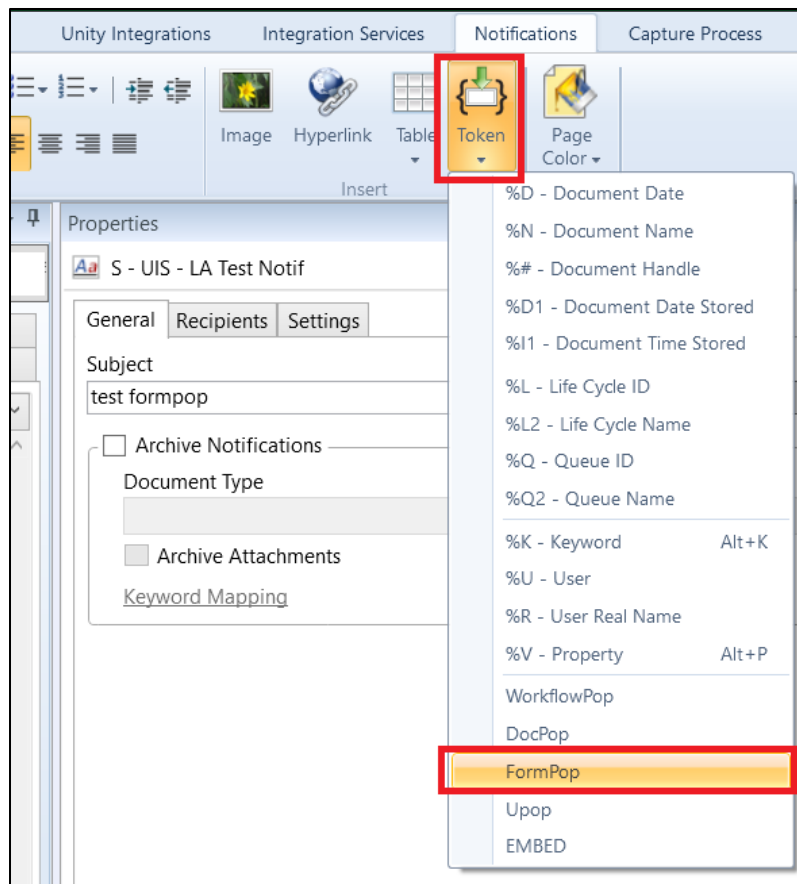
1. Log into OnBase Studio. You will need to be in the *Functional - Developer - Workflow* group.

2. Click on the **Notifications** tab within the desired repository.

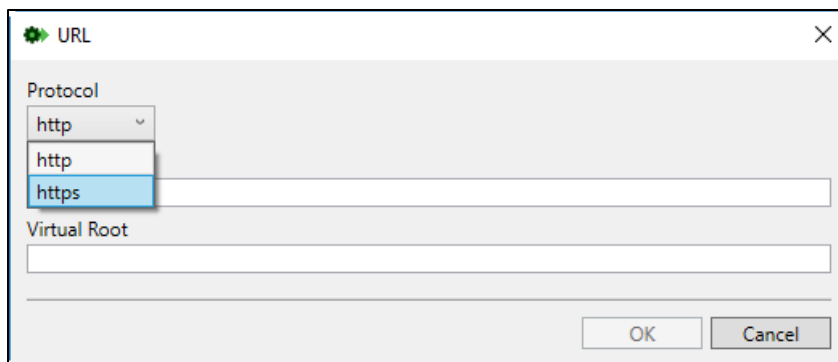


3. Either create a new notification or navigate to the desired notification if it already exists and check it out. Refer to the Workflow MRG for more information if needed.
4. Click in body text of your notification to place the cursor where you want the FormPop URL to be.
5. Click on the **Token** button on the Notification tab of the menu and select **FormPop**.

Note: In EP3, the new “FormPop” option did not include all necessary portions of the URL so it did not work! With 22.1, this option now works correctly and should be used instead of the DocPop option.



6. You will be presented with a URL dialog that you can fill in with the OnBase environment information. Always make sure that the **Protocol** is changed to https.



7. These are the values that should be used depending on the OnBase environment.

*Make sure your entries do not include any leading or trailing spaces! Studio will not trim them out and your URL will not work if it includes any spaces.*

***For public form modification (non-OnBase users):***

Anyone with the URL can view and make changes to the form.

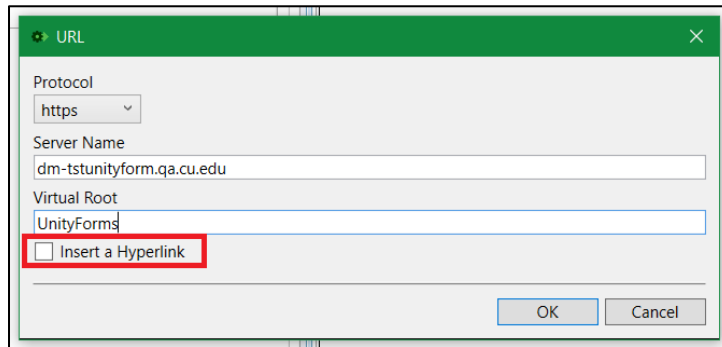
	SERVER	VIRTUAL ROOT
DEV	dm-devunityform.dev.cu.edu	UnityForms
TST	dm-tstunityform.qa.cu.edu	UnityForms
STG	dm-stgunityform.qa.cu.edu	UnityForms
PRD	dm-unityform.prod.cu.edu	UnityForms

***For users authenticating to their own OnBase accounts:***

Users will need to authenticate to OnBase using SSO with their campus credentials. Any users using the URL to access the form will need to have access to the document's document type.

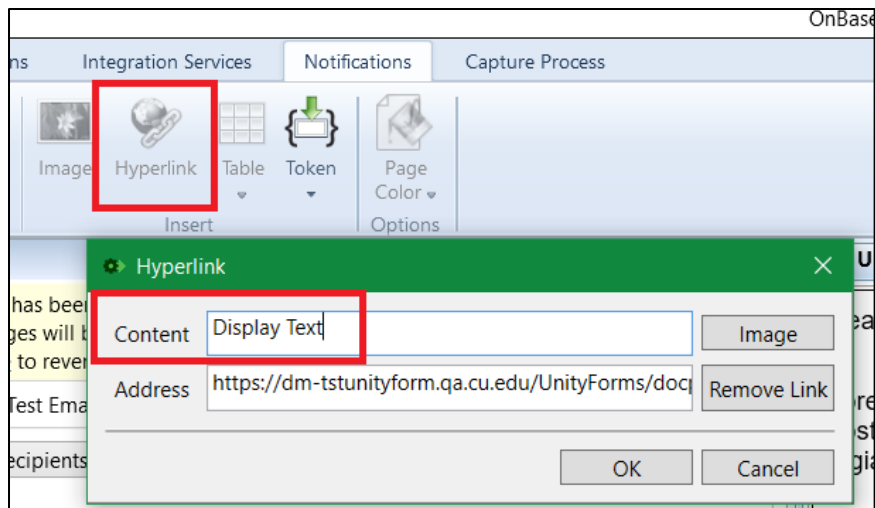
	SERVER	VIRTUAL ROOT
DEV	dm-dev.dev.cu.edu	AppNet
TST	dm-tst.qa.cu.edu	AppNet
STG	dm-stg.qa.cu.edu	AppNet
PRD	dm.prod.cu.edu	AppNet

- If you would like to display the formpop URL as a hyperlink, with “friendly” display text, check the **Insert Hyperlink** box.

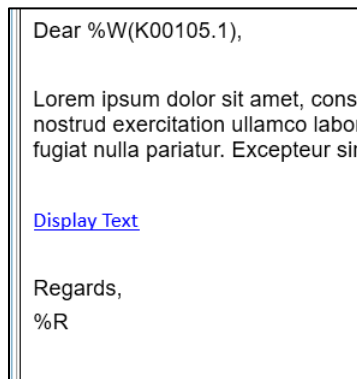


Once the formpop URL is inserted in the template as a hyperlink, select it and then use the **Hyperlink** button on the menu to modify the display text. Enter the desired display text in the **Content** field.



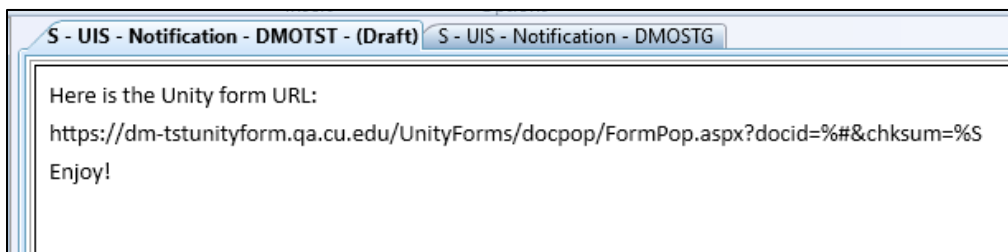


The template will be updated to display the text entered.



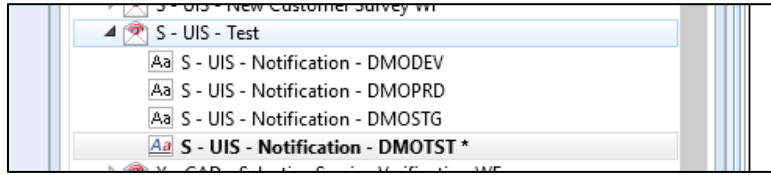
9. Click on **OK** to have the URL placed into your notification window.

10. This is an example of what the final FormPop notification URL looks like for DMOTST (if not displayed as a hyperlink). Make sure not to modify the URL in the notification template.



11. When creating a notification using a FormPop integration, you need to hardcode the server URL for the OnBase environment. Because of this, it's recommended that you create separate notification templates for each OnBase environment and send the correct one based upon the current environment variable.

Otherwise, you will need to update or recreate the URL in your notification template(s) after migrating the notification to a different environment.



See the [OnBase Guide - Unity Script - How to use the Read Environment Variable Script](#) guide for more information on how to setup and determine the OnBase environment.



If you are using the script, your steps should look like:


<p><b>Description</b></p>	
<p><b>Action: Run Unity Script</b> Get the environment variable</p>	

<p><b>Action: Set Property Value</b> Set a property value to reference the correct notification template</p>		
<p><b>Action: Send Notification</b> Send the notification, with the template determined based on the property value</p>		

**-OR-**

<p><b>Description</b></p>	<p>2 Option 2</p> <ul style="list-style-type: none"> <li>Get Environment Variable</li> <li>is environment DMOPRD? <ul style="list-style-type: none"> <li>On True <ul style="list-style-type: none"> <li>Send template DMOPRD</li> </ul> </li> <li>On False</li> </ul> </li> <li>is environment DMOSTG? <ul style="list-style-type: none"> <li>On True <ul style="list-style-type: none"> <li>Send template DMOSTG</li> </ul> </li> <li>On False</li> </ul> </li> <li>is environment DMOTST? <ul style="list-style-type: none"> <li>On True <ul style="list-style-type: none"> <li>Send template DMOTST</li> </ul> </li> <li>On False</li> </ul> </li> <li>is environment DMODEV? <ul style="list-style-type: none"> <li>On True <ul style="list-style-type: none"> <li>Send template DMODEV</li> </ul> </li> <li>On False</li> </ul> </li> </ul>	
<p><b>Action: Run Unity Script</b> Get the environment variable</p>	<p>Action Type Run Unity Script</p> <p>General Documentation</p> <p>Target Current Item</p> <p>Script GEN - Read Environment Variable</p> <p><input checked="" type="checkbox"/> Refresh item after script has executed</p>	
<p><b>Rule: Check Property Value</b> Check if DMOPRD</p>	<p>Rule Type Check Property Value</p> <p>General Documentation</p> <p>Property Name EnvironmentPropertyName</p> <p>Operator Type =</p> <p><input type="checkbox"/> Case insensitive</p> <p>Compare To All values in the property will</p> <p>Keyword Type &lt;None&gt;</p> <p>Constant value DMOPRD</p>	
<p><b>On True: Send Notification</b> Send PRD version of template</p>	<p>Action Type Send Notification</p> <p>General Tracking Documentation</p> <p>Attachment None</p> <p>Notification S - UIS - FormPop - DMOPRD</p> <p><input type="checkbox"/> Get Notification from Property</p> <p>Do Not Show Global Notifications <input checked="" type="checkbox"/></p>	

<p><b>Rule: Check Property Value</b> Check if DMOSTG</p>	<p>Rule Type Check Property Value</p> <p>General Documentation</p> <p>Property Name EnvironmentPropertyName</p> <p>Operator Type =</p> <p><input type="checkbox"/> Case insensitive</p> <p>Compare To   All values in the property will be compared         </p> <p>Keyword Type &lt;None&gt;</p> <p>Constant value DMOSTG</p>	
<p><b>On True: Send Notification</b> Send STG version of template</p>	<p>Action Type Send Notification</p> <p>General Tracking Documentation</p> <p>Attachment None</p> <p>Notification S - UIS - FormPop - DMOSTG</p> <p>Do Not Show Global Notifications <input checked="" type="checkbox"/></p>	
<p><b>Rule: Check Property Value</b> Check if DMOTST</p>	<p>Rule Type Check Property Value</p> <p>General Documentation</p> <p>Property Name EnvironmentPropertyName</p> <p>Operator Type =</p> <p><input type="checkbox"/> Case insensitive</p> <p>Compare To   All values in the property will be compared         </p> <p>Keyword Type &lt;None&gt;</p> <p>Constant value DMOTST</p>	
<p><b>On True: Send Notification</b> Send TST version of template</p>	<p>Action Type Send Notification</p> <p>General Tracking Documentation</p> <p>Attachment None</p> <p>Notification S - UIS - FormPop - DMOTST</p> <p>Do Not Show Global Notifications <input checked="" type="checkbox"/></p>	

<p><b>Rule: Check Property Value</b> Check if DMODEV</p>	<p>Rule Type Check Property Value</p> <p>General Documentation</p> <p>Property Name EnvironmentPropertyName</p> <p>Operator Type =</p> <p><input type="checkbox"/> Case insensitive</p> <p>Compare To   All values in the property will be compared to the constant value.</p> <p>Keyword Type &lt;None&gt;</p> <p>Constant value DMODEV</p>
<p><b>On True: Send Notification</b> Send DEV version of template</p>	<p>Action Type Send Notification</p> <p>General Tracking Documentation</p> <p>Attachment None</p> <p>Notification S - UIS - FormPop - DMODEV</p>

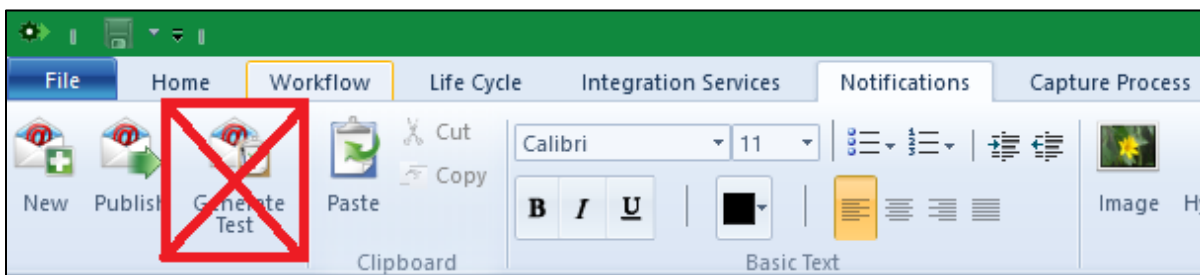
## Testing

You will need to thoroughly test the Unity FormPop to make sure it can be processed correctly, the correct portions are visible at each step, the correct portions are editable at each step, etc.

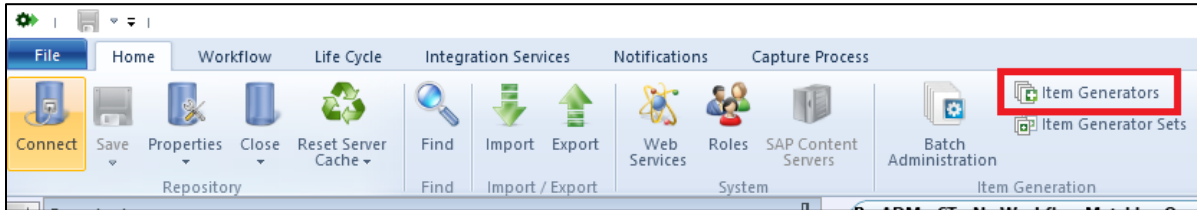
In non-production, you will need to be on a campus network or using a campus VPN to view Unity forms outside the OnBase clients.

Refer to the [handbook section on workflow notifications](#) for more information about testing workflow notifications.

You won't be able to use the **Generate Test** button on the Notification tab in Studio as the docid and checksum portions of the URL won't generate unless real content is used to populate them when sending the notification.



Instead, use the **Item Generators** on the Home tab in Studio to create test content or manually submit forms that will enter the workflow process. Refer to the Studio MRG for more information on using Item Generators.



## Securing FormPop Content After Usage

Once a user has the FormPop URL to the form, they may use that link at a later time when you do not want them to access the information or make further changes.

There are a couple of ways you can secure the content from being viewed or accessed later on.

1. Move the document from the original document type to another document type that is not assigned to the *UnityForms Group - X – X* group.
2. Add [Custom Actions](#) to the Unity form to prevent viewing or making modifications to the form after the form has been processed (based on user group, form values, and or workflow queues).

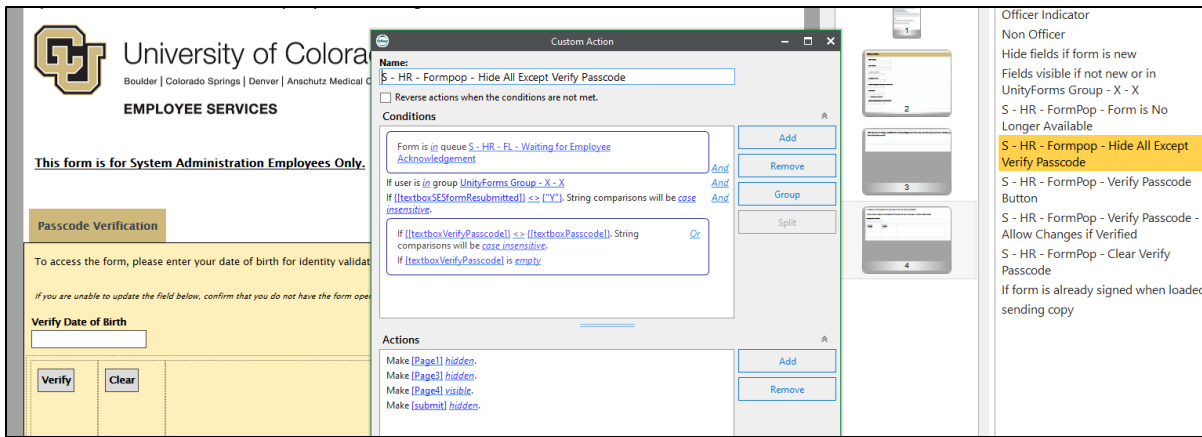
### ***Custom Action Recommendations to Control Visibility/Editing***

Since every process will have different requirements, you'll need to determine which of these methods are appropriate or where adjustments are needed. The intention is that these examples will give you some ideas for using custom actions to facilitate the best possible process using formpop. It is important to test all potential scenarios before going live with your solution, as any changes made will only apply to future forms, not existing forms.

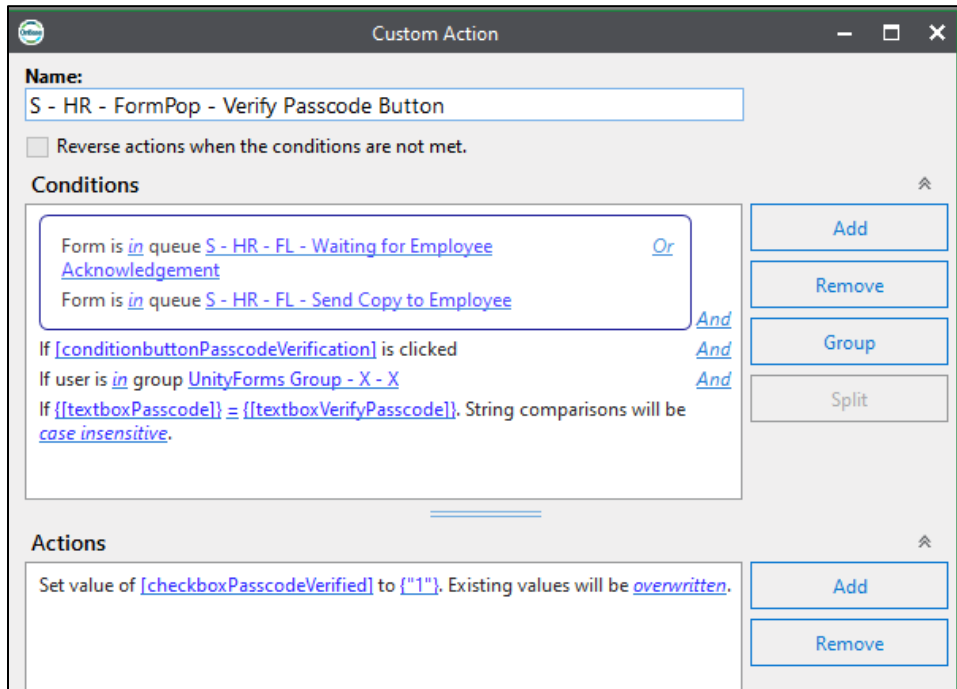
### **Verifying Identity**

You may wish to verify a user's identity prior to allowing them to see personal information that's been filled into the form (in case others are able to access the link). For example, you can require a passcode that is sent to the user or for them to enter their information.

In this case, users are shown a page with only a verification field where they are asked to enter their DOB to proceed.



Once they enter the passcode/DOB and click the verify button, a hidden field's value is set to be used in other custom actions.



Once the criteria are met, users can see page 1 of the form where they need to review the form and provide additional information.



**Custom Action**

**Name:**  
S - HR - FormPop - Verify Passcode - Allow Changes if Verified

Reverse actions when the conditions are not met.

**Conditions**

Form is *in* queue [S - HR - FL - Waiting for Employee Acknowledgement](#) *And*

If `{{checkboxPasscodeVerified}}` = `{\"1\"}`. String comparisons will be *case insensitive*. *And*

If user is *in* group [UnityForms Group - X - X](#) *Or*

If user is *in* group [Workflow - S - ES - System Administration](#)

**Actions**

Make `[Page1]` *visible*.

Make `[Page3]` *hidden*.

Make `[Page4]` *hidden*.

Make `[submit]` *visible*.

## Form Submission

Since this method is re-submitting changes to a form that already exists, the user is not routed to a webpage after clicking the Submit button like they are when submitting a form shared to a URL.

However, when the form is submitted, it is reloaded. In order to prevent further changes to the form, you can use a custom action to check for the desired conditions “on load” and show the user a message that the form is no longer available when those conditions are met. For example:

**Custom Action**

**Name:**  
If form is already signed when loaded

Reverse actions when the conditions are not met.

**Conditions**

If user is *in* group [UnityForms Group - X - X](#) *And*

If form is loading *And*

If `[signature32]` is *signed*

**Actions**

Make `[Page1]` *hidden*.

Make `[Page2]` *hidden*.

Make `[Page3]` *visible*.

Make `[Page4]` *hidden*.

Make `[submit]` *hidden*.

## Form is No Longer Available

Once the form is no longer in the queue where external users are being asked to provide information, if the link is used again the user will receive a message that the form is no longer available.

The screenshot shows a configuration interface for a form titled "S - HR - FormPop - Form is No Longer Available". The interface is divided into several sections:

- Header:** University of Colorado logo and "EMPLOYEE SERVICES" text.
- Message:** "This form is for System Administration Employees Only." and "Form is not available at this time." (highlighted in yellow).
- Form Content:** "This form is no longer available for acknowledgement. You may now close your browser window, your form has been saved." (highlighted in yellow).
- Conditions:** A list of conditions with "Add" and "Remove" buttons. The conditions are:
  - If user is in group [UnityForms Group - X - X](#) *And*
  - If form is [not new](#) *And*
  - Form is [not in](#) queue [S - HR - FL - Waiting for Employee Acknowledgement](#)
- Actions:** A list of actions with "Add" and "Remove" buttons. The actions are:
  - Make [\[Page1\] hidden](#).
  - Make [\[Page2\] hidden](#).
  - Make [\[Page3\] visible](#).
  - Make [\[paragraphFormNoLongerAvailable\] visible](#).
  - Make [\[submit\] hidden](#).
- Properties:** A list of properties on the right side, including "Officer Indicator", "Non Officer", "Hide fields if form is new", "Fields visible if not new or in UnityForms Group - X - X", and "S - HR - FormPop - Form is No Longer Available" (highlighted in yellow).

## FormPop and Document Locks

Once any document is being viewed or interacted with in OnBase, that document is locked by the user/session doing so. This means that if you are viewing the document in the Unity or web client at the same time you or someone else attempts to view the document using the formpop URL, the form will be read-only when viewed using the URL (since it is already locked). More information about locks is available in the Unity client MRG.

To help prevent confusion for users who are trying to modify a form that has been locked, you can use these options to display a warning message in that situation.

For example, viewing a form through formpop when it's already locked and therefore read-only:

**Unity script test**

This form is read-only because it is being viewed by another person or in another location.

**Boring old form stuff**

Student ID  
SADSFASDF

email address  
asdfsdf

email type  
asdfsdfasdfsdf

Campus Code  
CUSYS

**Document Lock Details**

Document Handle 41599681	formLockResult True	formLockUsername ARMA000017	formLockRealname Larissa Armand - CUSYS Or
formLockTime 04/21/2022 04:31:47 PM	formLockUsernum 460		

vs. in the Unity client (where the lock originated and the form can be edited):

S - UIS - ICS Testing Document - 4/21/2022 - SADSFASDF - - -

**Unity script test**

**Boring old form stuff**

Student ID  
SADSFASDF

email address  
asdfsdf

email type  
asdfsdfasdfsdf

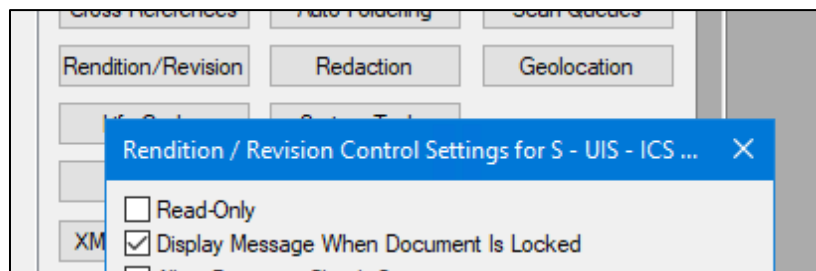
Campus Code  
CUSYS

**Document Lock Details**

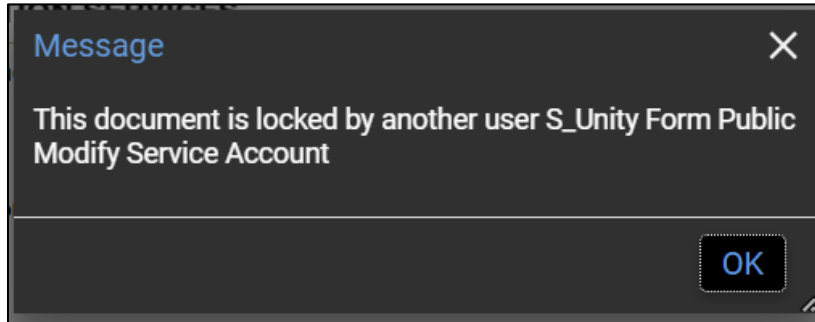
Document Handle 41599681	formLockResult False
-----------------------------	-------------------------

### Document Type Setting

Starting with version 22.1, the revision setting to warn users when viewing a locked document is respected for Unity forms. Go to the document type in Configuration, then choose Rendition/Revision and check the box for “Display Message When Document is Locked.”



When viewing a locked form, a message will appear including the real name of the user with the lock.



### **Script Option**

You may prefer to use this script instead, for further customization of the message to users.

### **Configuring Form Fields**

Certain field IDs must be used for the script to work. For input, you will need:

<b>Value/Use</b>	<b>Field ID</b>	<b>Used as</b>
<p>This should be set to the form’s document handle. The easiest way to do this is to add the “Document Handle” property field to the form. The value will then be populated automatically when the form is saved.</p> <p>You will need to update the field ID to match the expected input for the script.</p>	formDocumentHandle	Input, required

For outputs, you will need at least one of the following:

<b>Value/Use</b>	<b>Field ID</b>	<b>Used as</b>
<p>Date and time for the existing lock</p> <p>Data type of field must be Date &amp; Time</p>	formLockTime	Output, optional
Usernum/ID for the existing lock	formLockUsernum	Output, optional
Username (operID) for the existing lock	formLockUsername	Output, optional
Real name for the existing lock	formLockRealname	Output, optional



Will return either True (if a previous lock exists) or False (if no previous lock exists).	formLockResult	Output, optional
--	----------------	------------------

### Troubleshooting Form Scripts

If you'd like to include error messages, you can add the following output fields (these are optional):

Value	Field ID
General Error Message (all scripts)	error_message
Error Messages for this script only	error_message_GENOnBaseUnityFormLockWarning

### Custom Action to Execute Script

A custom action is required to execute a Unity script. Create a custom action with the desired conditions, then add an action to execute the **GEN - OnBase - Unity Form Lock Warning** script.

At a minimum, you should have a condition that the form is **not new** and that the form is **loading**. This script will not work on forms that have not been submitted and saved as documents (since a document handle/ID will not exist) and should be run as the form loads.

Required conditions:

- Form is loading
- Form is not new

You may wish to add other conditions such as:

- User is in the *UnityForms Group - X - X* user group

Required actions:

- Execute script GEN - OnBase - Unity Form Lock Warning

## Custom Action to Handle Script Result

The script will fill values for any of the output fields that are configured on the form template. You will need to create one or more custom actions to achieve the desired result.

For example, if you want to display a message, it will need to be made visible.

<b>Name:</b>	show message if locked
<b>Description:</b>	
<b>Conditions</b>	<p>If <code>{formLockResult} = ["True"]</code>. String comparisons will be <a href="#">case insensitive</a>.</p> <p><input checked="" type="checkbox"/> Reverse actions when the conditions are not met.</p>
<b>Actions</b>	<p>Make <code>{paragraphlocked}</code> <a href="#">visible</a>.</p>

Suggested condition:

- Expression is true - formLockResult = "True"

You may wish to add other conditions such as:

- User is in the *UnityForms Group - X - X* user group

Suggested action:

- Change Visibility - Make paragraphLocked visible

You could also show the information for the user who has the lock on the form if that would be helpful. However, if the public submission account is being used, showing the information for that account may cause more confusion than it prevents, so you may want to include that as a condition in the custom action to only show that information if it indicates a specific user that could be contacted.



Custom Action

**Name:**  
show username if not public account

**Description:**

**Conditions**

If `{{formLockResult}}` = `"True"`. String comparisons will be *case insensitive*. And

If `{{formLockUsername}}` <> `"S_UNITYFORMS_POP"`. String comparisons will be *case insensitive*.

Reverse actions when the conditions are not met.

---

**Actions**

Make `[formLockRealname]` *visible*.

Suggested condition:

- Expression is true - formLockResult = "True"
- Expression is true - formLockUsername <> "S\_UNITYFORMS\_POP"

You may wish to add other conditions such as:

- User is in the *UnityForms Group - X - X* user group

Suggested action:

- Change Visibility - Make formLockRealname visible

Using a custom action to check for the public account, I see the name of the user with the lock if it is not the public account:



## Unity script test

This form is read-only because it is being viewed by another person or in another location.

User viewing form

### Boring old form stuff

Student ID

email address

email type

Campus Code

### Document Lock Details

Document Handle	formLockResult	
<input type="text" value="41599681"/>	<input type="text" value="True"/>	
formLockTime	formLockUsernum	formLockUsername
<input type="text" value="04/22/2022 10:04:33 A"/>	<input type="text" value="----"/>	<input type="text" value="-----"/>

But not when it is the public account:

## Unity script test

This form is read-only because it is being viewed by another person or in another location.

### Boring old form stuff

Student ID

email address

email type

Campus Code

### Document Lock Details

Document Handle	formLockResult	
<input type="text" value="41599681"/>	<input type="text" value="True"/>	
formLockTime	formLockUsernum	formLockUsername
<input type="text" value="04/22/2022 10:06:00 AM"/>	<input type="text" value="----"/>	<input type="text" value="S_UNITYFORMS_POP"/>

