University of Colorado
Boulder | Colorado Springs | Denver | Anschutz Medical Campus

OnBase by Hyland

OnBase Guide – Workflow – FormPop Notifications

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

Complexity Level: Departmental Workflow Developers, Departmental Administrative Users

1/6/2025

Table of Contents

Background

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

FormPop allows users to view and edit existing Unity Forms using a simplified web 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.

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

FormPop does not allow users to create/submit new forms, only to modify existing forms. To allow submission of Unity forms using a URL, 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.

Your form and additional workflow process should already be designed and built, this guide only covers the configuration aspects directly related to using FormPop.

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.

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.

Environment Usage

Different server address values are used for each of our OnBase environments. The FormPop URL created and inserted into a notification will only work for the environment referenced in the URL.

This can pose an issue for testing, migrating, and refreshing amongst different environments so takes some additional consideration and configuration. However with a little extra work up front, it is possible to have configuration that will work in any environment with no additional maintenance/adjustments between environments.

One option (previously recommended) is to have a separate template per environment and use rules/actions to send the template that corresponds to the current environment. The downside to this approach is having to create and maintain 4x as many templates and when testing in non-production, you aren't testing the template that will actually be used which can lead to issues.

Our current recommendation (outlined in this guide) is to substitute the server address value in the FormPop URL with a property value, so that the value can be set dynamically.

Either approach will require using a script to determine the current OnBase environment. See the OnBase Guide - Unity Script - How to use the Read Environment Variable Script guide for more information on how to use this script to determine the OnBase environment.

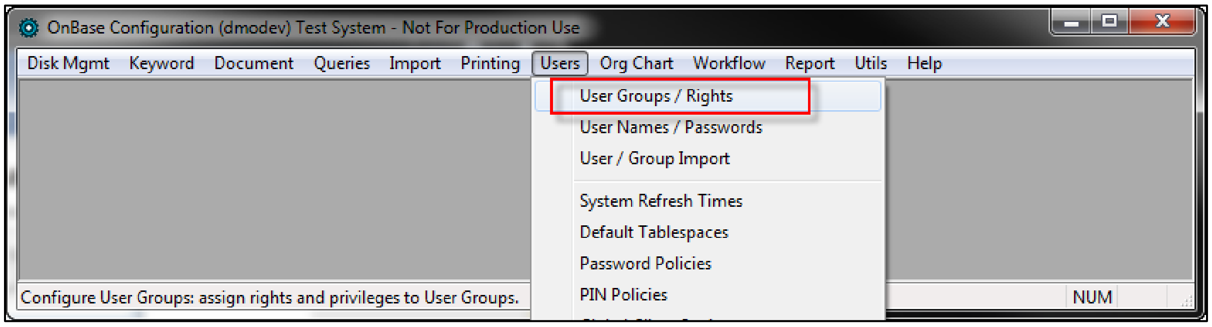
OnBase Configuration Client – User Group Access

Add the document type that contains the Unity Forms 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 specific group used for public submission/modification.

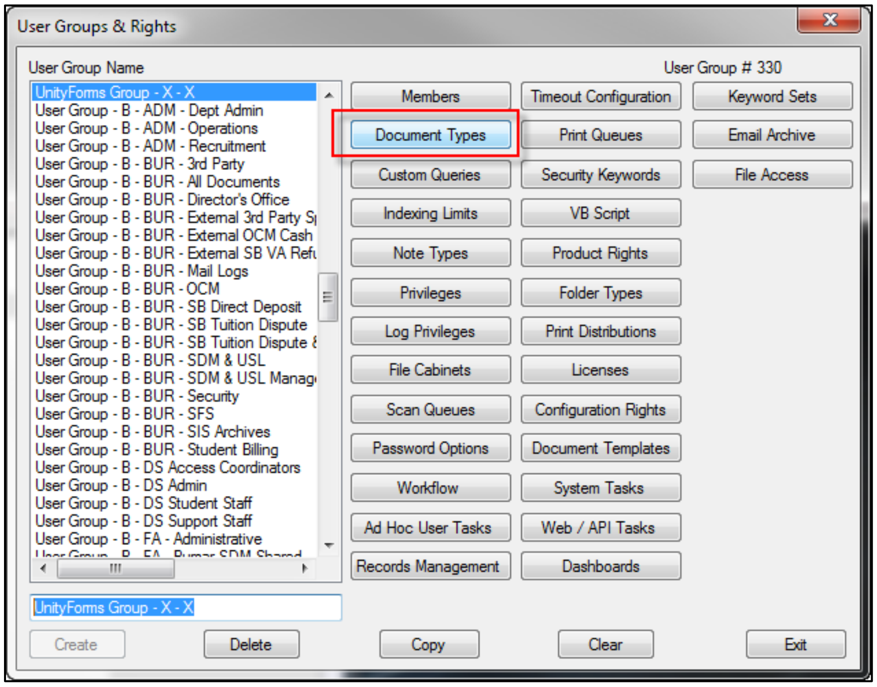
1. Log in to the OnBase Configuration Client

2. Select **Users** then **User Groups/Rights**.

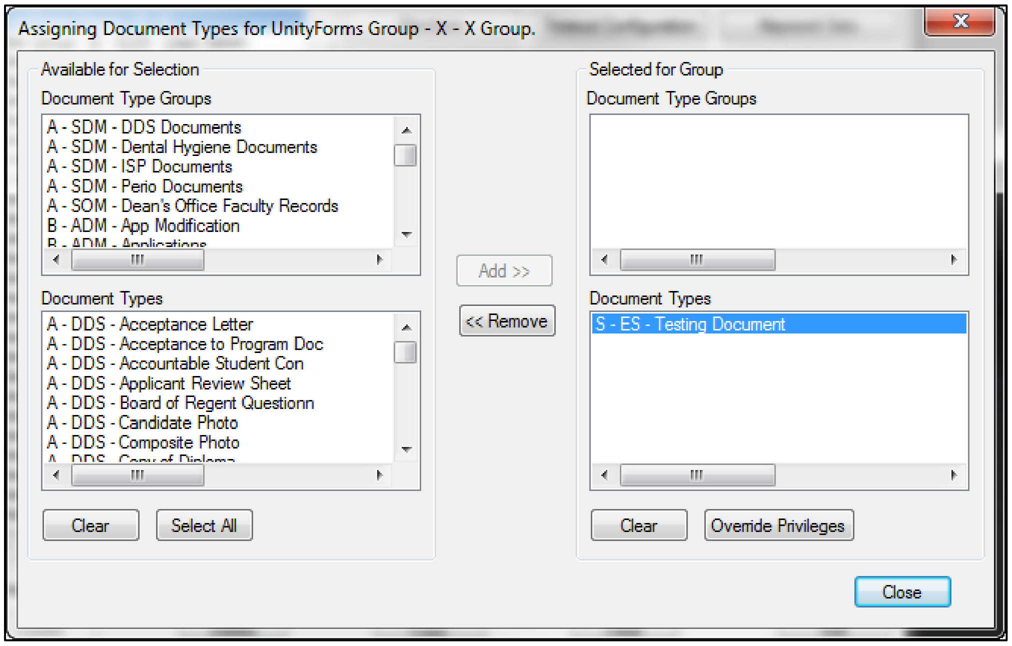

OnBase Configuration (dmodev) Test System - Not For Production Use
Disk Mgmt Keyword Document Queries Import Printing Users Org Chart Workflow Report Utils Help
User Groups / Rights
User Names / Passwords
User / Group Import
System Refresh Times
Default Tablespaces
Password Policies
PIN Policies
NUM

Configure User Groups: assign rights and privileges to User Groups.

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


User Groups & Rights
User Group Name
User Group # 330
UnityForms Group - X - X
User Group - B - ADM - Dept Admin
User Group - B - ADM - Operations
User Group - B - ADM - Recruitment
User Group - B - BUR - 3rd Party
User Group - B - BUR - All Documents
User Group - B - BUR - Director's Office
User Group - B - BUR - External 3rd Party S
User Group - B - BUR - External OCM Cash
User Group - B - BUR - External SB VA Refu
User Group - B - BUR - Mail Logs
User Group - B - BUR - OCM
User Group - B - BUR - SB Direct Deposit
User Group - B - BUR - SB Tuition Dispute
User Group - B - BUR - SB Tuition Dispute
User Group - B - BUR - SDM & USL
User Group - B - BUR - SDM & USL Manag
User Group - B - BUR - Security
User Group - B - BUR - SFS
User Group - B - BUR - SIS Archives
User Group - B - BUR - Student Billing
User Group - B - DS Access Coordinators
User Group - B - DS Admin
User Group - B - DS Student Staff
User Group - B - DS Support Staff
User Group - B - FA - Administrative
Members
Timeout Configuration
Keyword Sets
Document Types
Print Queues
Email Archive
Custom Queries
Security Keywords
File Access
Indexing Limits
VB Script
Note Types
Product Rights
Privileges
Folder Types
Log Privileges
Print Distributions
File Cabinets
Licenses
Scan Queues
Configuration Rights
Password Options
Document Templates
Workflow
System Tasks
Ad Hoc User Tasks
Web / API Tasks
Records Management
Dashboards
UnityForms Group - X - X
Create
Delete
Copy
Clear
Exit

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.


Assigning Document Types for UnityForms Group - X - X Group.

Available for Selection

Document Type Groups

A - SDM - DDS Documents
A - SDM - Dental Hygiene Documents
A - SDM - ISP Documents
A - SDM - Perio Documents
A - SOM - Dean's Office Faculty Records
B - ADM - App Modification
B - ADM - Applications

Document Types

A - DDS - Acceptance Letter
A - DDS - Acceptance to Program Doc
A - DDS - Accountable Student Con
A - DDS - Applicant Review Sheet
A - DDS - Board of Regent Questionn
A - DDS - Candidate Photo
A - DDS - Composite Photo
A - DDS - Copy of Diploma

Selected for Group

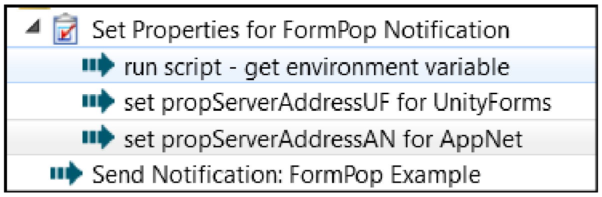
Document Type Groups

Document Types

S - ES - Testing Document

OnBase Studio - Workflow Configuration

You'll need to ensure the environment property value is set prior to sending any notification using FormPop. Since there are a couple steps, it's recommended you use a task list for consistency and ease of maintenance.


Set Properties for FormPop Notification
run script - get environment variable
set propServerAddressUF for UnityForms
set propServerAddressAN for AppNet
Send Notification: FormPop Example

1. Get the value of EnvironmentPropertyName using the script.
    a. Action Type: Run Unity Script
    b. Script name: GEN – OnBase – Read Environment Variable

2. Set the server address value based on the EnvironmentPropertyName value.
    a. Action Type: Set property to expression
    b. This is where you'll use the values from the section above, depending on whether users receiving the notification will log in to OnBase. You may need to use both authenticated and unauthenticated FormPop within the same process, so it can help to set both property values with different names so you can use the correct one for any notification.
    c. Expression for unauthenticated users (ex. Students, parents), in this example property is called propServerAddressUF:

```
iif(%VEnvironmentPropertyName = "DMOPRD";"dm-unityform.prod.cu.edu";
```

University of Colorado
Boulder | Colorado Springs | Denver | Anschutz Medical Campus

```
iif(%VEnvironmentPropertyName = "DMOSTG";"dm-stgunityform.qa.cu.edu";
iif(%VEnvironmentPropertyName = "DMOTST";"dm-tstunityform.qa.cu.edu";
iif(%VEnvironmentPropertyName = "DMODEV";"dm-devunityform.dev.cu.edu";
""))))
```

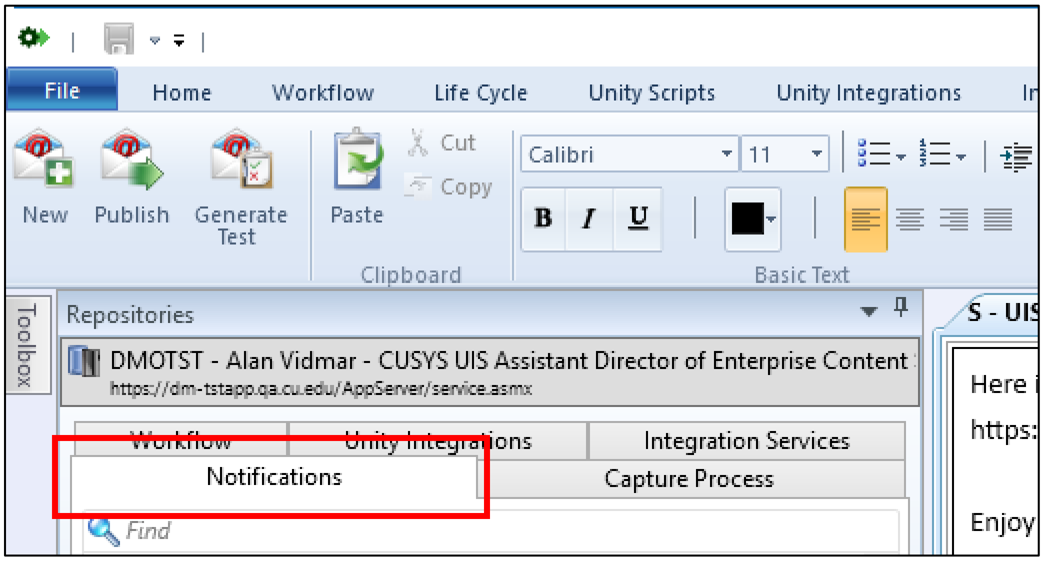
   d. Expression for authenticated users (ex. Faculty and staff with OnBase access), in this example property is called propServerAddressAN:

```
iif(%VEnvironmentPropertyName = "DMOPRD";"dm.prod.cu.edu";
iif(%VEnvironmentPropertyName = "DMOSTG";"dm-stg.qa.cu.edu";
iif(%VEnvironmentPropertyName = "DMOTST";"dm-tst.qa.cu.edu";
iif(%VEnvironmentPropertyName = "DMODEV";"dm-dev.dev.cu.edu";
""))))
```

3. Send the notification.
   a. Action Type: Send Notification
   b. The template configuration below needs to be completed, with the template published to be available for selection.
   c. Make sure you are using the appropriate property bag to have the property value for the server address (and any other property values) filled into the template.

OnBase Studio – Notification Template Configuration

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

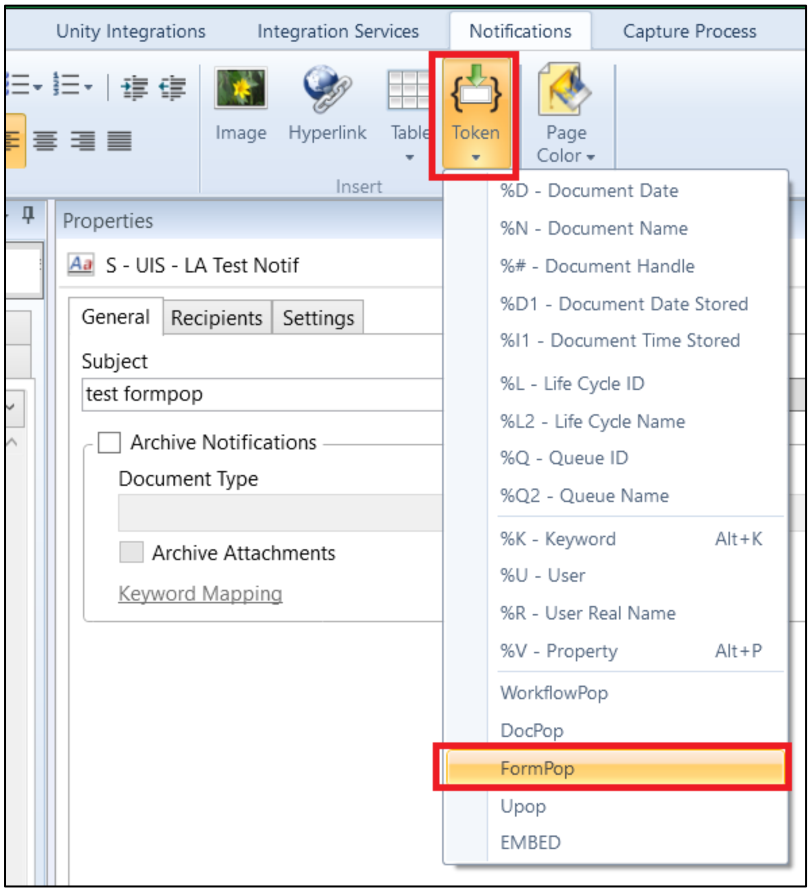


2. Either create a new notification or navigate to and check out the desired notification if it already exists. Refer to the Workflow MRG for more information if needed.

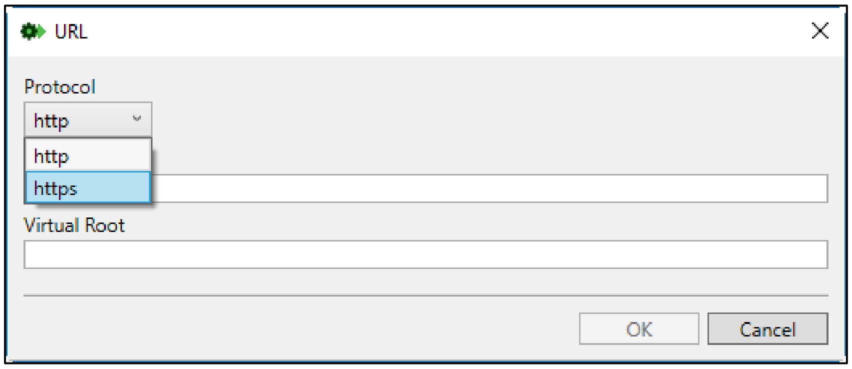
3. Click in body text of your notification to place the cursor where you want the FormPop URL to be.

4. 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.



5. 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.



6. These are the values that should be used depending on the OnBase environment. **See section on Environment Usage for more information before completing configuration with hard-coded values.**

   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

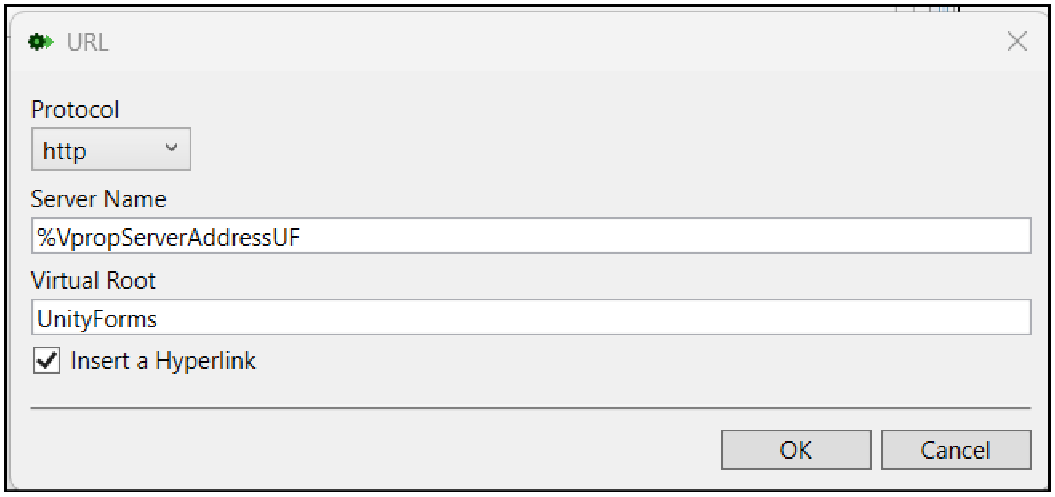
   For users authenticating to their own OnBase accounts:

   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

   Instead of hardcoding these values, this is where we suggest using the property value for the server name/address.

   The corresponding virtual root value can be entered depending on whether users will be authenticating (AppNet) or not (UnityForms).

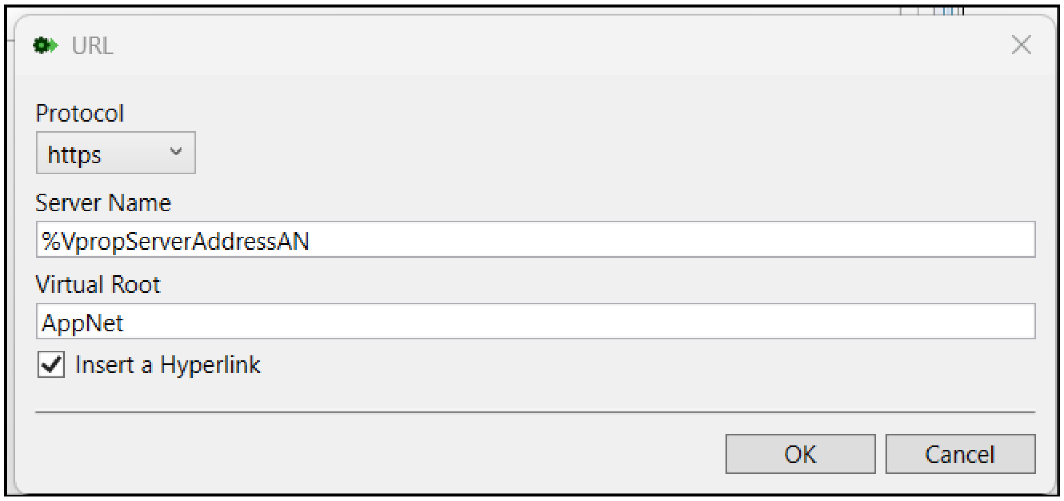

URL

Protocol: http

Server Name: %VpropServerAddressUF

Virtual Root: UnityForms

✓ Insert a Hyperlink


URL

Protocol: https

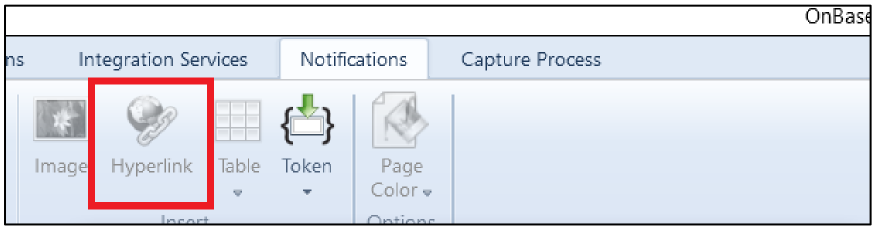
Server Name: %VpropServerAddressAN

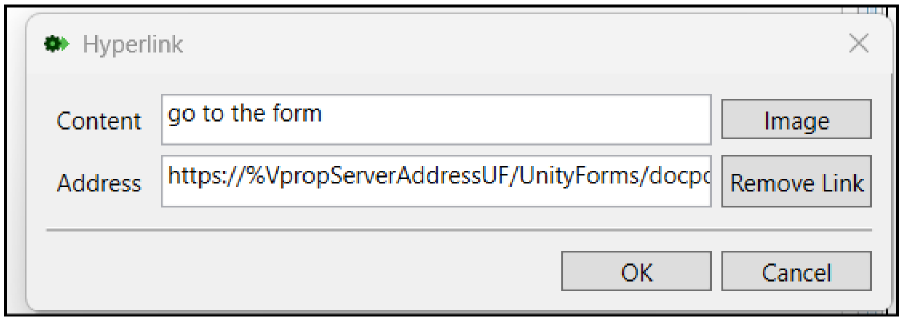
Virtual Root: AppNet

✓ Insert a Hyperlink

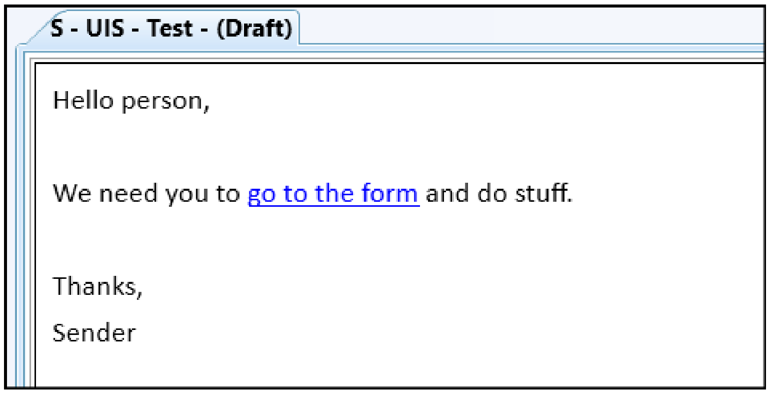
7. If you would like to display the FormPop URL as a hyperlink, with "friendly" display text, check the **Insert Hyperlink** box.

   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.



Hello person,

We need you to go to the form and do stuff.

Thanks,
Sender

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

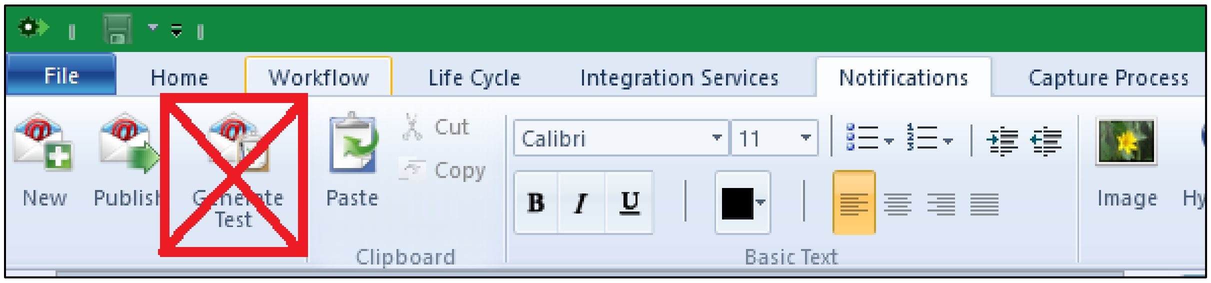
Testing

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

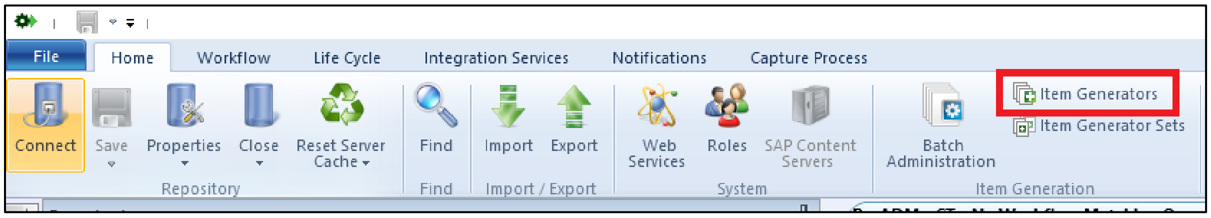
In non-production, you will need to be on a campus network or using a campus VPN to use any OnBase functionality, including accessing forms using a URL.

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).

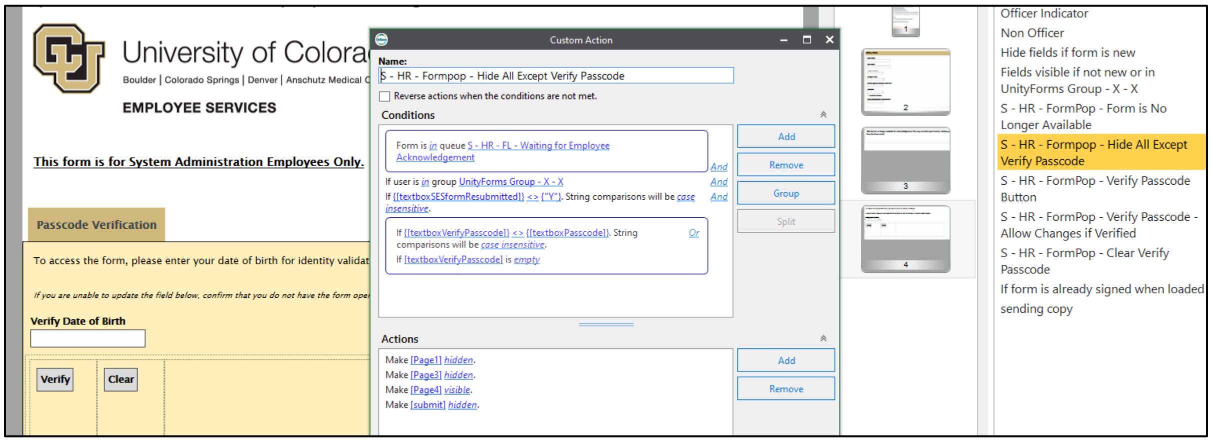
Custom Action Recommendations to Control Visibility/Editing

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.


University of Colora
Boulder | Colorado Springs | Denver | Anschutz Medical C

EMPLOYEE SERVICES

This form is for System Administration Employees Only.

Passcode Verification

To access the form, please enter your date of birth for identity validat

If you are unable to update the field below, confirm that you do not have the form ope

Verify Date of Birth

Verify | Clear

Custom Action

Name:
S - HR - Formpop - Hide All Except Verify Passcode

Reverse actions when the conditions are not met.

Conditions

Form is *in* queue S - HR - FL - Waiting for Employee Acknowledgement — *And*
If user is *in* group UnityForms Group - X - X — *And*
If {[textboxSESformResubmitted]} <> {"Y"}. String comparisons will be *case insensitive*. — *And*

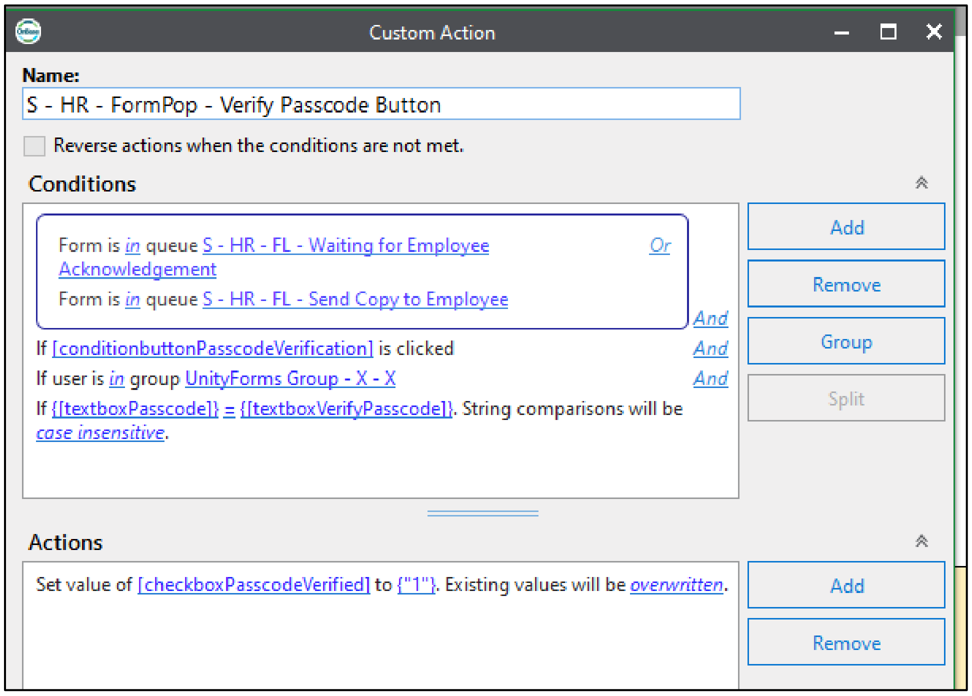
If {[textboxVerifyPasscode]} <> {[textboxPasscode]}. String comparisons will be *case insensitive*. — *Or*
If [textboxVerifyPasscode] is *empty*

Actions

Make [Page1] *hidden*.
Make [Page3] *hidden*.
Make [Page4] *visible*.
Make [submit] *hidden*.

Officer Indicator
Non Officer
Hide fields if form is new
Fields visible if not new or in UnityForms Group - X - X
S - HR - FormPop - Form is No Longer Available
S - HR - Formpop - Hide All Except Verify Passcode
S - HR - FormPop - Verify Passcode Button
S - HR - FormPop - Verify Passcode - Allow Changes if Verified
S - HR - FormPop - Clear Verify Passcode
If form is already signed when loaded
sending copy

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.


Custom Action

Name:
S - HR - FormPop - Verify Passcode Button

Reverse actions when the conditions are not met.

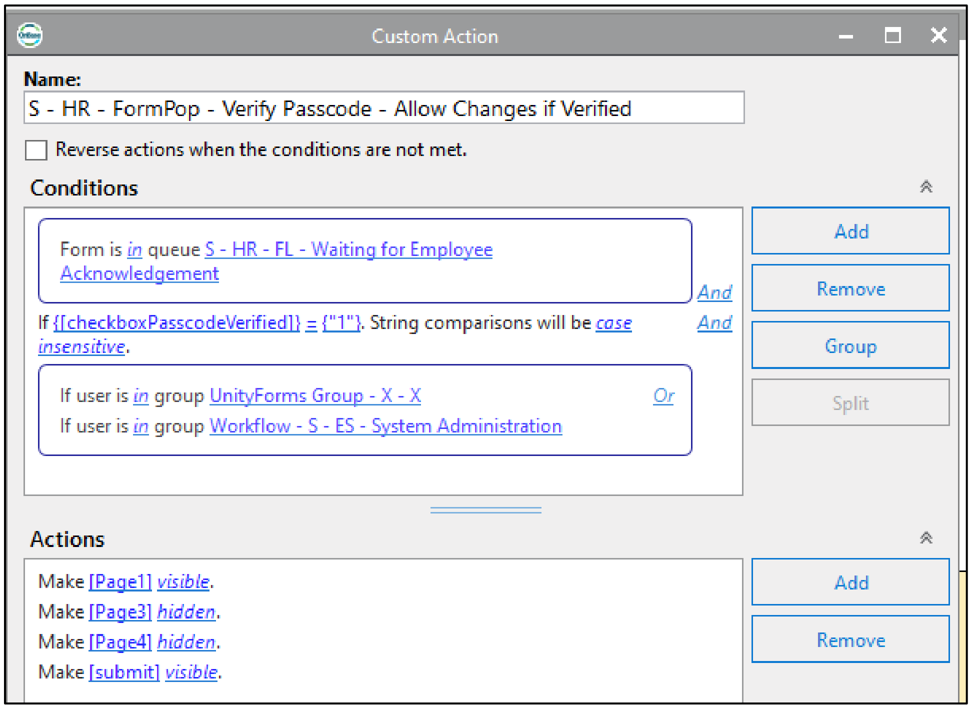
Conditions

Form is *in* queue S - HR - FL - Waiting for Employee Acknowledgement — *Or*
Form is *in* queue S - HR - FL - Send Copy to Employee — *And*
If [conditionbuttonPasscodeVerification] is clicked — *And*
If user is *in* group UnityForms Group - X - X — *And*
If {[textboxPasscode]} = {[textboxVerifyPasscode]}. String comparisons will be *case insensitive*.

Actions

Set value of [checkboxPasscodeVerified] to {"1"}. Existing values will be *overwritten*.

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

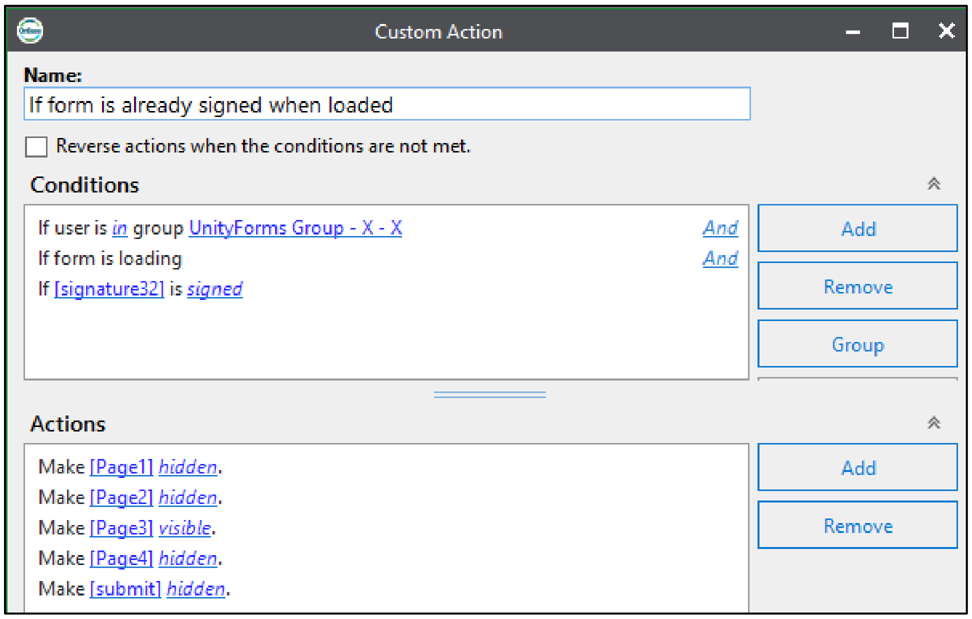
Actions

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 redirected to a webpage after clicking the Submit button like they are when submitting a form shared using 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

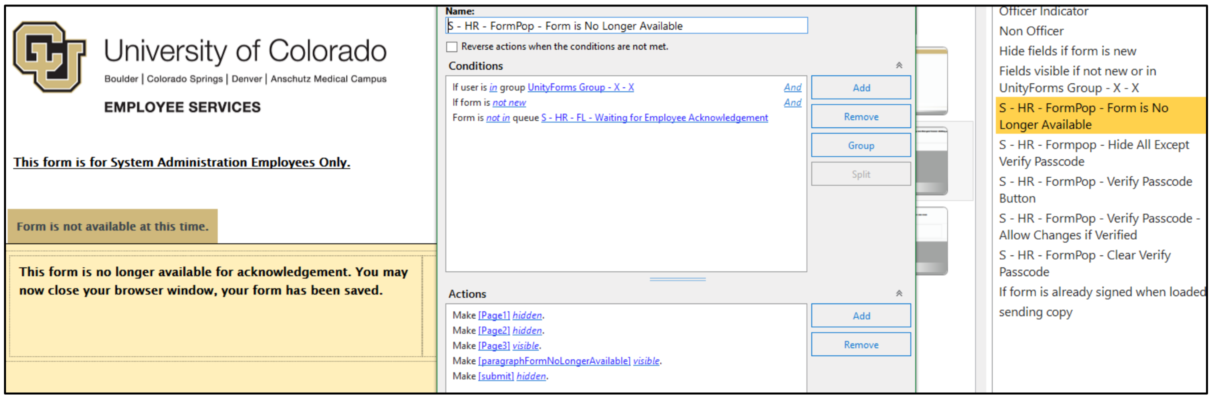
If user is in group UnityForms Group - X - X
And
If form is loading
And
If [signature32] is signed

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.



University of Colorado
Boulder | Colorado Springs | Denver | Anschutz Medical Campus
EMPLOYEE SERVICES

This form is for System Administration Employees Only.

Form is not available at this time.

This form is no longer available for acknowledgement. You may now close your browser window, your form has been saved.

Name: S - HR - FormPop - Form is No Longer Available

Reverse actions when the conditions are not met.

Conditions

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

Make [Page1] *hidden*.
Make [Page2] *hidden*.
Make [Page3] *visible*.
Make [paragraphFormNoLongerAvailable] *visible*.
Make [submit] *hidden*.

Officer Indicator
Non Officer
Hide fields if form is new
Fields visible if not new or in UnityForms Group - X - X
S - HR - FormPop - Form is No Longer Available
S - HR - Formpop - Hide All Except Verify Passcode
S - HR - FormPop - Verify Passcode Button
S - HR - FormPop - Verify Passcode - Allow Changes if Verified
S - HR - FormPop - Clear Verify Passcode
If form is already signed when loaded
sending copy

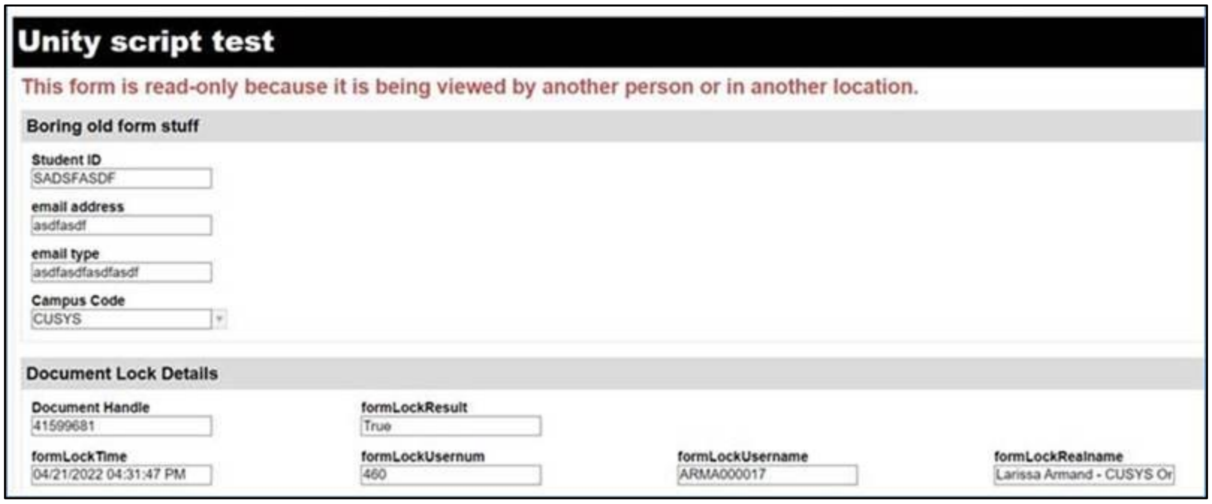
FormPop and Document Locks

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.

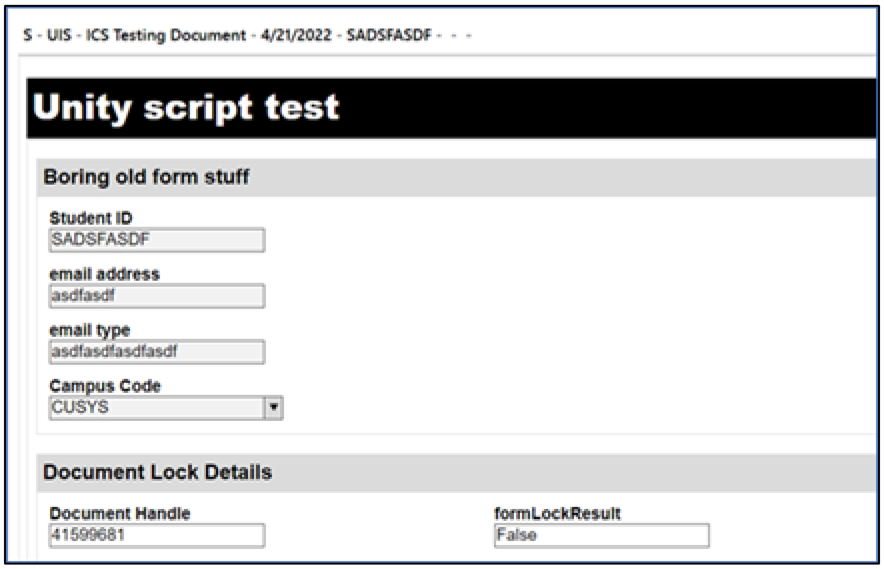
Boring old form stuff

Student ID	SADSFASDF
email address	asdfasdf
email type	asdfasdfasdfasdf
Campus Code	CUSYS

Document Lock Details

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

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

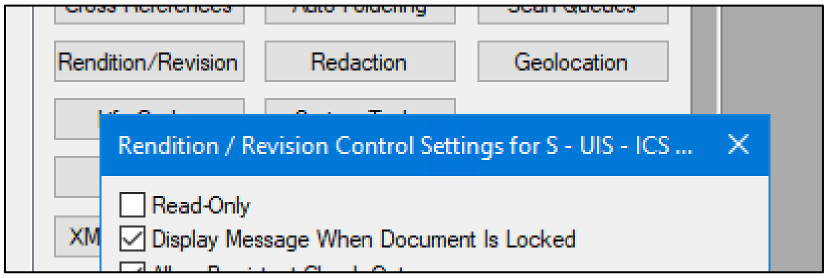
Student ID	SADSFASDF
email address	asdfasdf
email type	asdfasdfasdfasdf
Campus Code	CUSYS

Document Lock Details

Document Handle	41599681
formLockResult	False

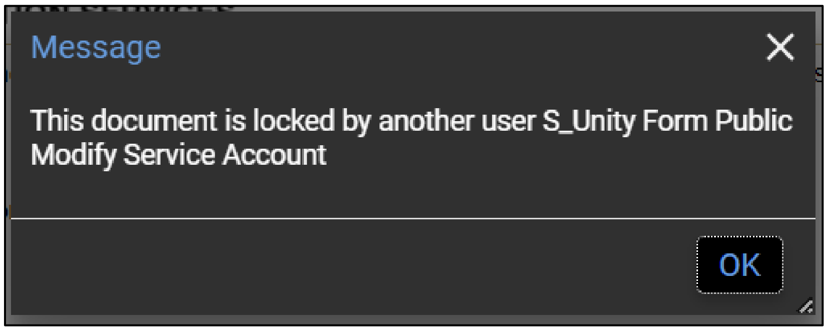
Document Type Setting

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.”



☐ Read-Only
☑ Display Message When Document Is Locked

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



This document is locked by another user S_Unity Form Public Modify Service Account

Script Option

You may prefer to use this script instead, for further customization of the message to users, especially users who will not be familiar with OnBase terminology.

Configuring Form Fields

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

Value/Use	**Field ID**	**Used as**
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. You will need to update the field ID to match the expected input for the script.	formDocumentHandle	Input, required

For outputs, you will need <u>at least one</u> of the following:

Value/Use	**Field ID**	**Used as**
Date and time for the existing lock Data type of field must be Date & Time	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

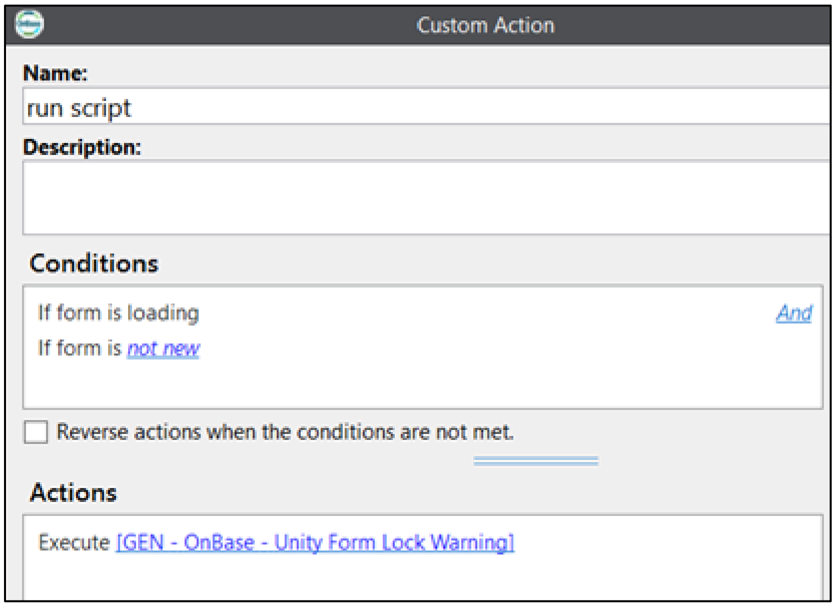
If you'd like to include error messages, you can add the following output fields (these are optional):

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.



Custom Action

Name:
run script

Description:

Conditions

If form is loading
If form is not new

And

☐ Reverse actions when the conditions are not met.

Actions

Execute [GEN - OnBase - Unity Form Lock Warning]

Required conditions:

- Form is loading
- Form is not new

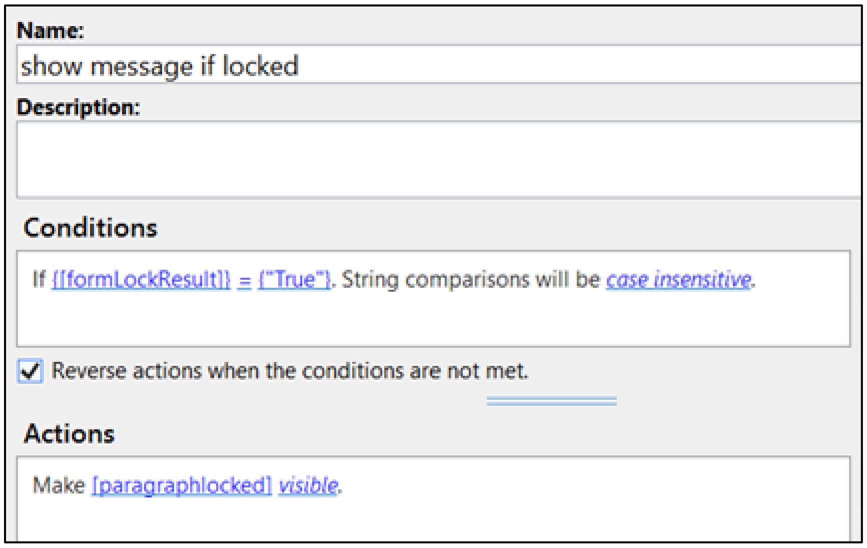
You may wish to add other conditions such as:
- User is in the *UnityForms Group - X - X* user group

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.


Name:
show message if locked
Description:

Conditions
If {[formLockResult]} = {"True"}. String comparisons will be *case insensitive*.

☑ Reverse actions when the conditions are not met.

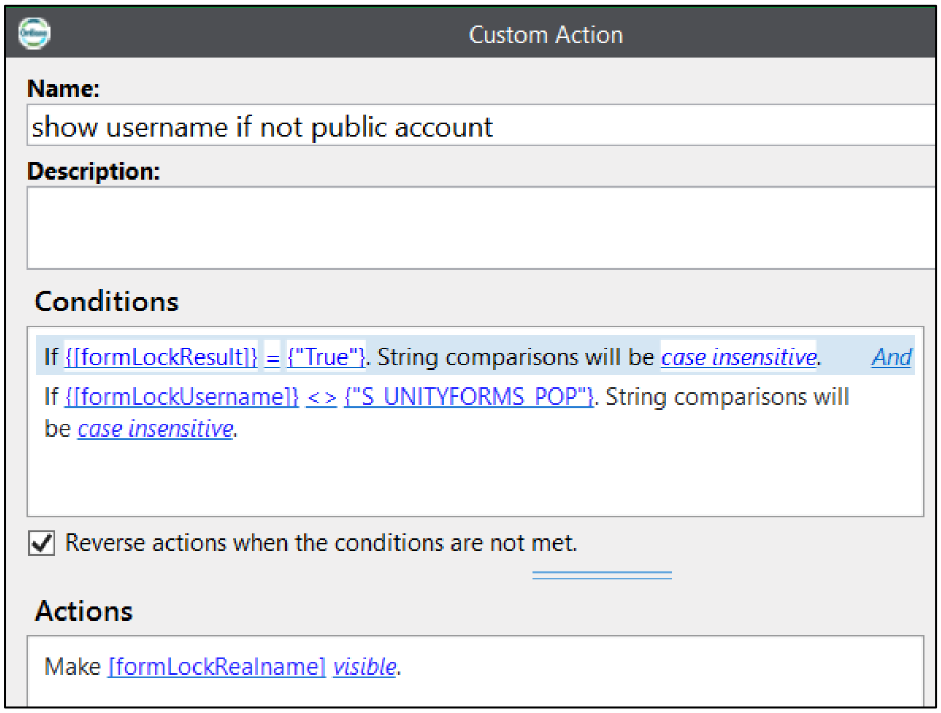
Actions
Make [paragraphlocked] *visible*.

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”

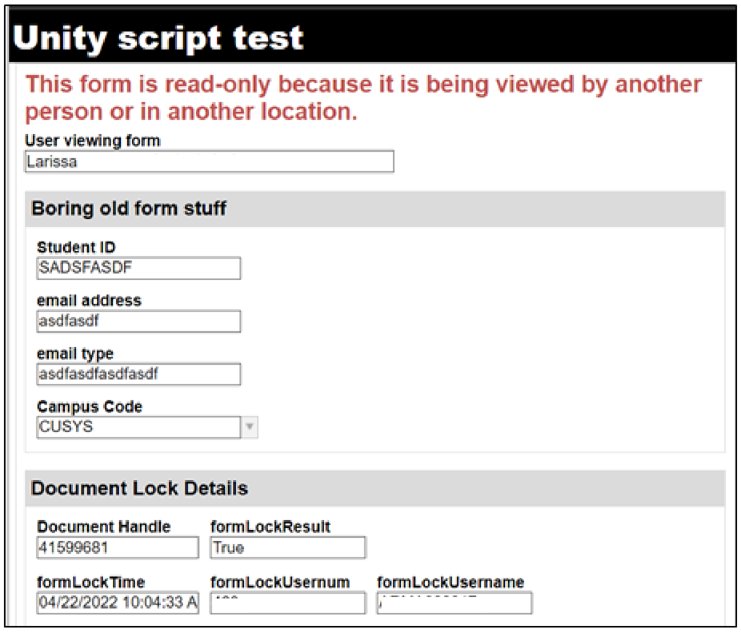
You may wish to add other conditions such as:

- 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
Larissa

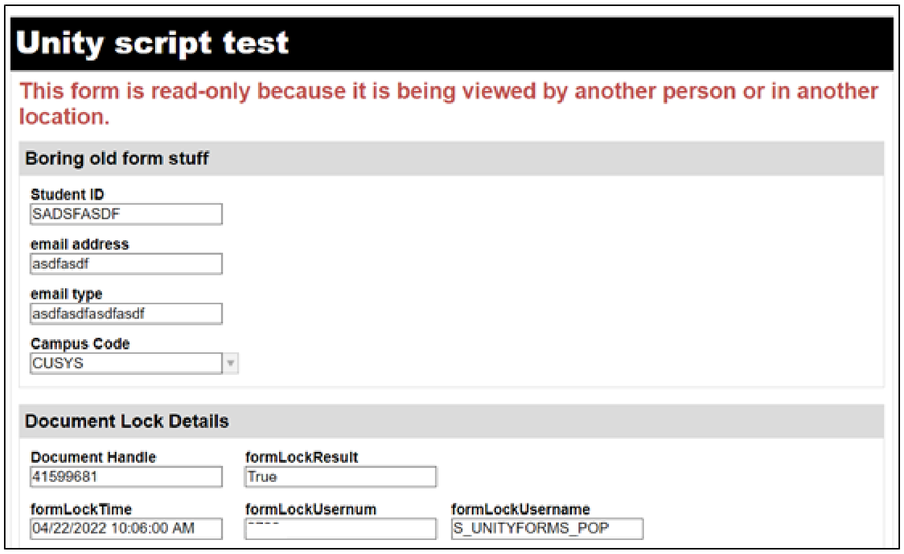
Boring old form stuff

Student ID	SADSFASDF
email address	asdfasdf
email type	asdfasdfasdfasdf
Campus Code	CUSYS

Document Lock Details

Document Handle	formLockResult	
41599681	True	
formLockTime	**formLockUsernum**	**formLockUsername**
04/22/2022 10:04:33 A	[redacted]	[redacted]

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	SADSFASDF
email address	asdfasdf
email type	asdfasdfasdfasdf
Campus Code	CUSYS

Document Lock Details

Document Handle	formLockResult	
41599681	True	
formLockTime	**formLockUsernum**	**formLockUsername**
04/22/2022 10:06:00 AM	[redacted]	S_UNITYFORMS_POP