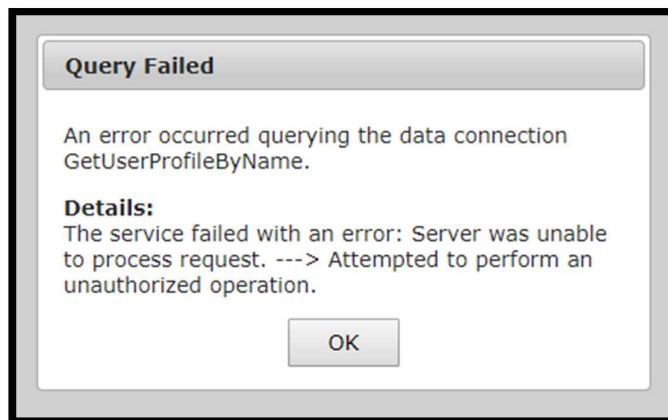


HOW TO FORMSVIEWER-ENABLE YOUR FORMS

REVERT UDCX DATA CONNECTIONS BACK TO REGULAR DATA CONNECTIONS

Most of the forms have data connections that use the GetUserProfileByName SharePoint Web Service method to retrieve the user's information. These data connections were, at some point, converted to UDCX files from the previous environment, and it does not work in FormsViewer on SharePoint O365.

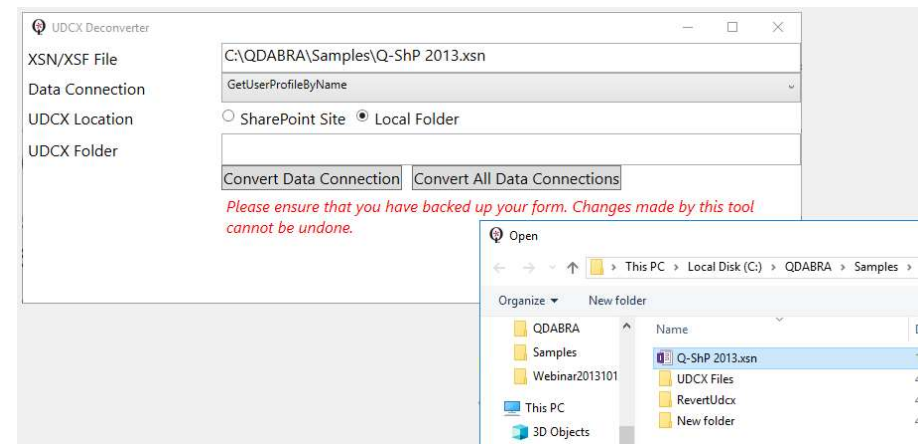


Download the Revert UDCX Tool and Guide at the link below, then follow these steps to revert connections back to regular data connections.

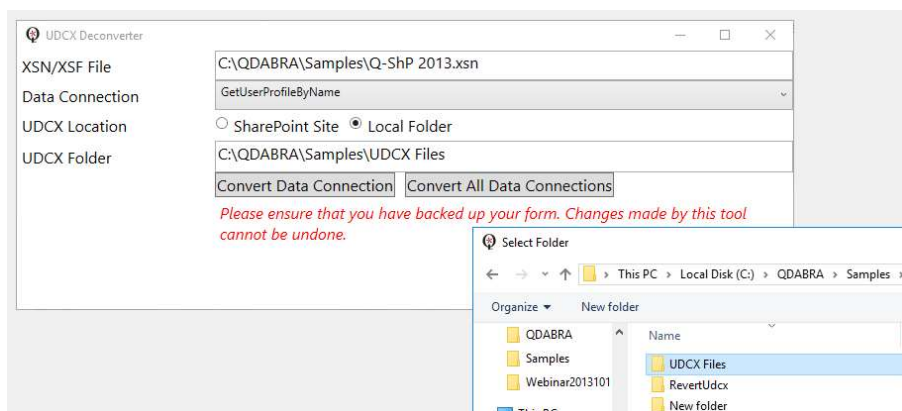
<https://www.infopathdev.com/files/folders/revertudcx/default.aspx>

IMPORTANT: Please ensure that you have backed up your form. Changes made by this tool cannot be undone.

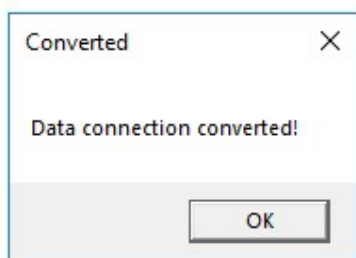
1. Download the UDCX file that you wish to deconvert, from your SharePoint site and save it to your local folder in your machine.
2. Extract the contents of the tool: RevertUdcx.zip file into your local machine.
3. Double-click on the RevertUdcx.exe file to launch the UDCX Deconverter tool – accept security prompts, if any.
4. Enter the location of the XSN that you wish to deconvert the data connection of – you may also click inside the XSN/XSF File textbox to browse through your folders.
5. Select the UDCX Data Connection to be converted



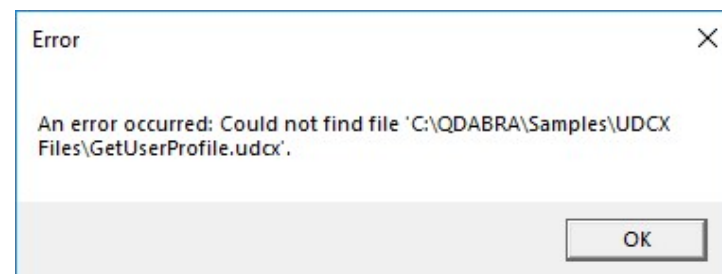
6. Select the folder where the UDCX data connection file you wish to deconvert is saved, or manually enter the path to the folder:



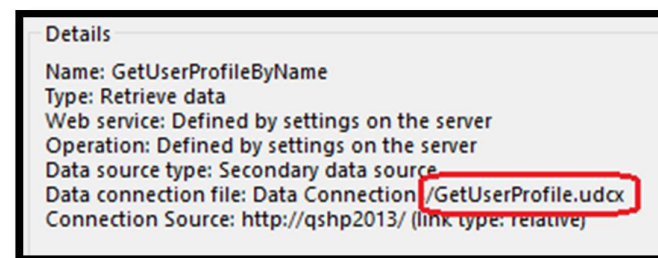
You should receive a confirmation prompt if the deconversion has been successful. Verify by opening your XSN in InfoPath Designer – the data connection should now appear like a regular data connection (non-UDCX).



If you get an error such as the error below:



Make sure that the saved UDCX's file name matches the file name declared at the data connections when the file was created:



For more info, please refer to the [Revert UDCX Tool User Guide](#).

MAKE DATA CONNECTIONS DYNAMIC

For easier maintenance when working in different sites/subsites, we recommend making the form data connections dynamic, i.e. the form detects the site where the it is being opened from, and then updates the data connections based on that location. To make it even easier to implement across your forms, we have created an InfoPath Template Part (XTP) for you.

Follow these steps and guidelines to add the XTP and implement the technique in your forms.

1. Locate the XTP named DynamicDCs.xtp2 that we have included in this package.
2. Add the XTP in your InfoPath Custom Controls. Go to this link for detailed steps: <https://support.office.com/en-ie/article/insert-a-template-part-fd60bf03-74e3-43ea-b35b-99bbe8cf2d79#bm1>.
3. Add the XTP into your XSN. Go to this link for detailed steps: <https://support.office.com/en-ie/article/insert-a-template-part-fd60bf03-74e3-43ea-b35b-99bbe8cf2d79#bm2>
4. Follow the steps provided within the XTP.
5. On step 4 of the XTP Usage steps, after you've exported the resource file named **XtpLogic_DynamicDCs.xml** into your machine, open it in a text editor such as Notepad.

6. Update the attributes – library, list and data connection names, accordingly based on your form's data connections – see comments inline:

```
<Environment>
  <HostUrl />

  <!-- Update libraryName with the name of the library where
  attachments will be uploaded -->
  <Attachments libraryName="" libraryUrl="" />

  <!-- Update dataConnectionName with the name of your Submit to
  SharePoint Library data connection. Update libraryName with the name
  of the library where your form submits -->
  <SubmitLibrary dataConnectionName="" libraryName="" libraryUrl="" />

  <!-- Insert separate lines for your form lists data connections with
  the corresponding data connection names and list names -->
  <Lists>
    <List dataConnectionName="" listName="" triggerFixup="" />
    <List dataConnectionName="" listName="" triggerFixup="" />
    <!-- Add additional lines for all your form lists data
    connections -->
  </Lists>
</Environment>
```

For example:

```
<Environment>
  <HostUrl />

  <!-- Update libraryName with the name of the library where attachments will be uploaded
  -->
  <Attachments libraryName="Images and Files" libraryUrl="" />

  <!-- Update dataConnectionName with the name of your Submit to SharePoint Library data
  connection. Update libraryName with the name of the library where your form submits -->
  <SubmitLibrary dataConnectionName="SharePoint Library Submit" libraryName="Sample
  Library" libraryUrl="" />

  <!-- Insert separate lines for your form lists data connections with the corresponding
  data connection names and list names -->
  <Lists>
    <List dataConnectionName="Connection Name 1" listName="List Name 1" triggerFixup="" />
    <List dataConnectionName="Connection Name 2" listName="List Name 2" triggerFixup="" />
    <!-- Add additional lines for all your form lists data connections -->
  </Lists>
</Environment>
```

7. If your form is configured to submit data to a list via the qRules SubmitToSharePointList command, you would have a mapping file. If so, also update the listName attribute within the <Mapping> tags of the XtpLogic_DynamicDCs.xml file:

```
<Mapping>
<!-- Update listName with the name of the list specified in your
mapping file for submitting data back to the list -->
  <tempMappingListName listName="" />
  <tempMappingListUrl />
</Mapping>
```

You will find the exact list name in this portion of your mapping.xml file:

```
<?xml version="1.0" encoding="utf-16"?>
<myFields schemaVersion="5.2">
  <Mappings mappingName="PO Status Update">
    <ListCollection useName="true" ListName="PO Status Update">
      { 9544CD98-AA2D-4306-A813-D78146044E4F } </ListCollection>
```

Also ensure that the **useName** attribute in your mapping file equals "true".

8. Save your changes to the XtpLogic_DynamicDCs.xml file and then re-add the updated file into your XSN's resource files, overwriting the previous one.

When done, you may delete the XTP section as well as the DynamicDCs group that got added in your form's Main data source.

Additional Steps:

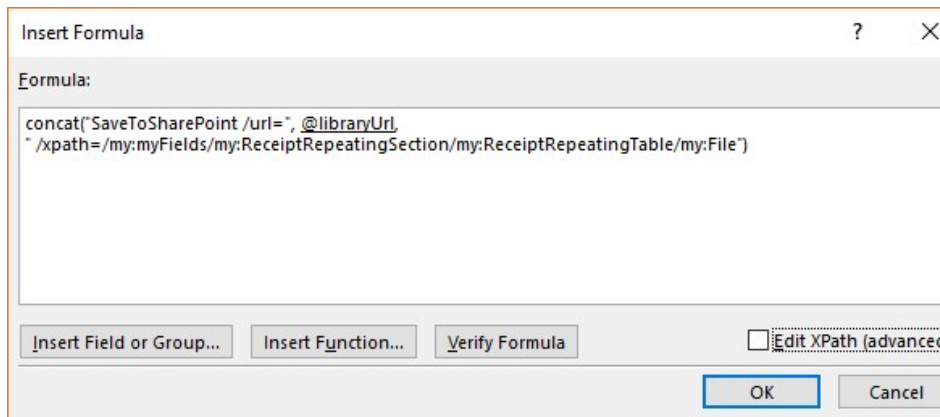
You will have to manually update some of your form's rules that were dependent on the non-dynamic versions of your data connections.

1. Inspect your form for any rule that queries the GetUserProfileByName data connection – these will have to be replaced with a rule that sets the QdabraRules/Command node to the value:
GetUserProfileByName
2. Inspect your form for any rule that saves attachments to a SharePoint library. These will have to be replaced with a dynamic rule to set the /url parameter of the SaveToSharePoint qRules Command to point to the XtpLogic_DynamicDCs data source > /XtpLogic/Values/Environment/Attachments/@libraryUrl field. For example, instead of a static value:

[SaveToSharePoint /url=https://my.qdabra.sharepoint.com/Requests/PO-CR/CreditCardPurchases/ImagesAndFiles](https://my.qdabra.sharepoint.com/Requests/PO-CR/CreditCardPurchases/ImagesAndFiles)
[/xpath=/my:myFields/my:ReceiptRepeatingSection/my:ReceiptRepeatingTable/my:File](#)

It will be replaced with a dynamic value of:

```
concat("SaveToSharePoint /url=",  
xdXDocument:GetDOM("XtpLogic_DynamicDCs")/XtpLogic/Values/Environmen  
t/Attachments/@libraryUrl, "  
/xpath=/my:myFields/my:ReceiptRepeatingSection/my:ReceiptRepeatingTable/  
my:File")
```



3. There might be times when the lists you've imported into your O365 site does not match the ones added in your form's data connections/sources. There could be missing or renamed columns. You will know that this is the case if you see an error when the form loads in FormsViewer. Please contact us for troubleshooting these kinds of issues.

UPLOAD TEMPLATES TO FORMSVIEWER

It is a lot easier and quicker to publish XSNs into your FormsViewer app as opposed to traditional way of going through the InfoPath Designer's Publishing Wizard.

Simply go to your FormsViewer app and click on **Manage Templates** > Upload a New Template > designate a unique and descriptive Form Name, browse through your XSN and then click Upload. You may also categorize your XSN into a group for better organization. When done, click Upload and you should be able to open the form in FormsViewer.

Download the package in this link:

https://www.infopathdev.com/files/folders/fv_how_to/entry118954.aspx for more detailed steps in Getting Started with FormsViewer.

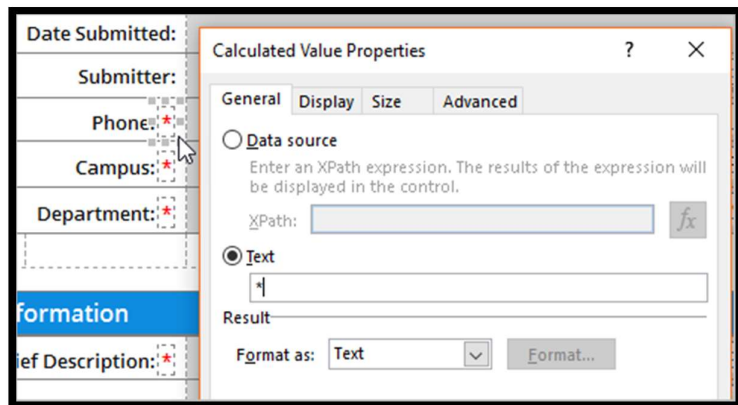
CONFIGURE LIBRARY AND LIST LINKS TO OPEN IN FORMSVIEWER

References:

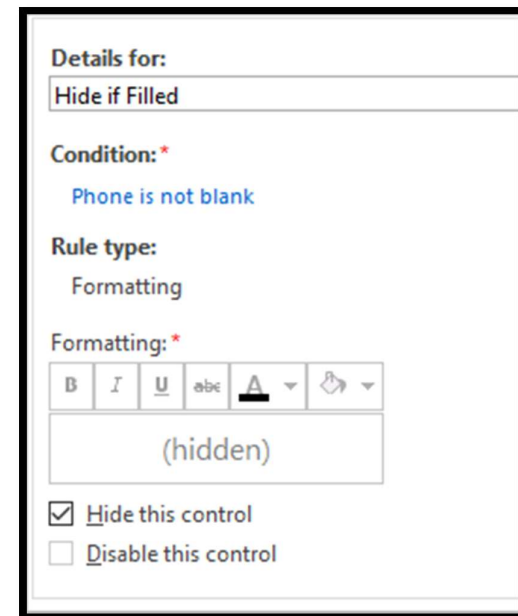
- [FVLinks - FormsViewer links for SharePoint Classic UI](#)
- [FormsViewer Extensions - FormsViewer links for SharePoint Modern UI](#)

HANDLING REQUIRED FIELDS

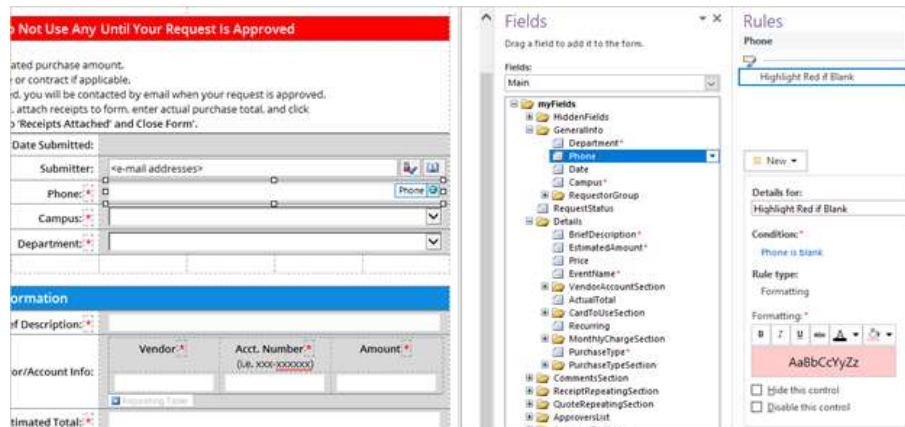
While the data validation feature in FormsViewer is still a work in progress, the workaround is to add conditional formatting rules. Let's say we have a Phone field that is required – to bring this to users' attention, we can either add a red asterisk in a calculated value control, like so:



This will indicate that the field cannot be blank; and then add a conditional formatting rule to hide it if the field has been filled out:

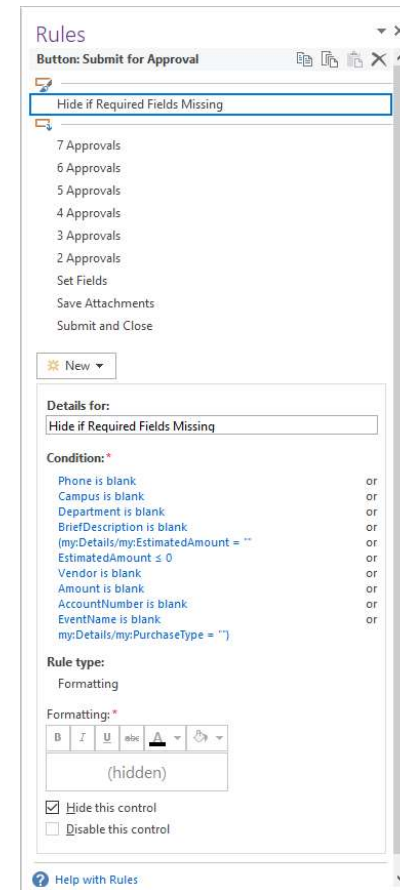


In addition, we can also highlight the control itself, a textbox for example, if it is blank:



The screenshot shows a form with a red header bar that says "Not Use Any Until Your Request is Approved". Below the header, there are several fields: "Date Submitted:", "Submitter:", "Phone:", "Campus:", and "Department:". The "Phone" field is highlighted in red. To the right of the form, there is a "Rules" panel. The "Rules" panel shows a rule named "Highlight Red if Blank" with a condition "Phone is blank".

We also need to make sure that any Submit button cannot be clicked unless all the required fields have been filled out. To do so, we'll have to specify all of the required fields in the conditional formatting rule – to disable or hide the Submit button if at least one of the conditions is met, i.e. one of the required fields is blank:



The screenshot shows the "Rules" panel in the QDABRA software. The rule is named "Hide if Required Fields Missing". The condition is a list of fields that are blank, separated by "or":
 Phone is blank
 Campus is blank
 Department is blank
 BriefDescription is blank
 (my:Details/my:EstimatedAmount = ""
 EstimatedAmount ≤ 0
 Vendor is blank
 Amount is blank
 AccountNumber is blank
 EventName is blank
 my:Details/my:PurchaseType = ""
 The "Rule type" is set to "Formatting". The "Formatting" section shows a list of formatting options: Bold, Italic, Underline, Color, and Background Color. The "Background Color" option is selected, and the color is set to red. The "Rule type" is set to "Formatting". The "Formatting" section shows a list of formatting options: Bold, Italic, Underline, Color, and Background Color. The "Background Color" option is selected, and the color is set to red. The "Rule type" is set to "Formatting". The "Formatting" section shows a list of formatting options: Bold, Italic, Underline, Color, and Background Color. The "Background Color" option is selected, and the color is set to red.

This one formatting rule can be copied and pasted in the Submit buttons in all of the views, adding and/or removing any expression that is not needed in the current view based on the form's logic.

IMPORTING LISTS/LIBRARIES FROM ONE SHAREPOINT SITE TO ANOTHER

Although this process is not entirely in the scope of FormsViewer-enabling your form, we want to make sure that your converted FormsViewer forms are fully working in your new site. Here are some tips in importing your site objects (lists and libraries in particular) from your old site to the new one, using site templates (STPs).

Reference: <https://support.office.com/en-gb/article/manage-list-templates-c3884ad1-bc49-44b8-b3d6-3bc6a01eb393#ID0EACAAA>

Follow the steps in the sections:

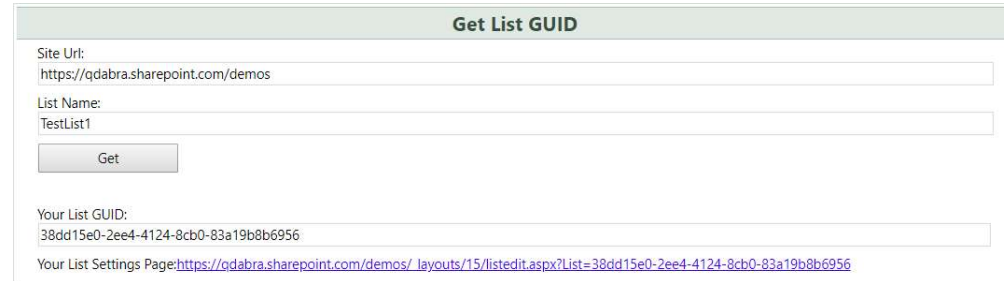
- Create a list template (this will done from the old site)
- Download a template to the file system (download to your local machine)
- Upload a template to the list template gallery (upload to your new site)

You'll then add a new app by searching for the template that you uploaded to your new site.

In cases where you don't see the list/library contents on the view, it could be that the default view configured from the old site has been corrupted or is not compatible in your new environment. If that's the case, the easiest way is to change the default view.

However, you won't be able to get access to the list settings page because the ribbon is hidden – so we built a simple form for you to get to that page. Follow the steps below to get around the issue.

1. Locate the XSN named GetListGuid that is included in this package – upload the XSN to your FormsViewer site (where you plan to import your list/library).
2. Open the form and fill out the fields, then click on the Get button.



3. You should see a generated link – this is the link to your List Settings page.
4. Click on the link and scroll down to the very bottom of the page where you see the Views section.
5. Select a different view that is not the default view.
6. In the Name section, check the box to make your selected view the default one:



7. Click OK – you should now see the contents of your new list/library.

From here you might need to either modify the previous default view after inspecting what could have caused the rendering issue; or just try to create a new view from scratch, ensuring compatibility this time since you're creating it on the new site.

Hope that helps!

SUPPORT

If you have encountered any issues with this tool, please contact support@qdabra.com.

For any other questions not covered in this document, please contact Qdabra Software.

CONTACT INFORMATION

Qdabra Software
218 Main Street,
Suite 731,
Kirkland, WA 98033

Phone: 877.544.2389
Email: Support@Qdabra.com
Website: <http://www.qdabra.com>
Community: <http://www.infopathdev.com>