

BULK EDITING DASHBOARD ON O365

A business intelligence dashboard is a data visualization tool that displays the current status of metrics and key performance indicators for a business, department, or specific process at a quick glance.

Rather than a SharePoint library or list view, using InfoPath as a dashboard gives us more control or customizability over the design, layout, and filters.

Dashboard Requirements

This guide assumes we are working on a SharePoint Online environment with Office 365 and requires the following elements:

- [gRules](#)
- InfoPath
- [FormsViewer](#)

The Bulk Edit Dashboard also works for SharePoint 2016 or lower, where Code-Based Sandbox is enabled – FormsViewer is not required and can be optional.

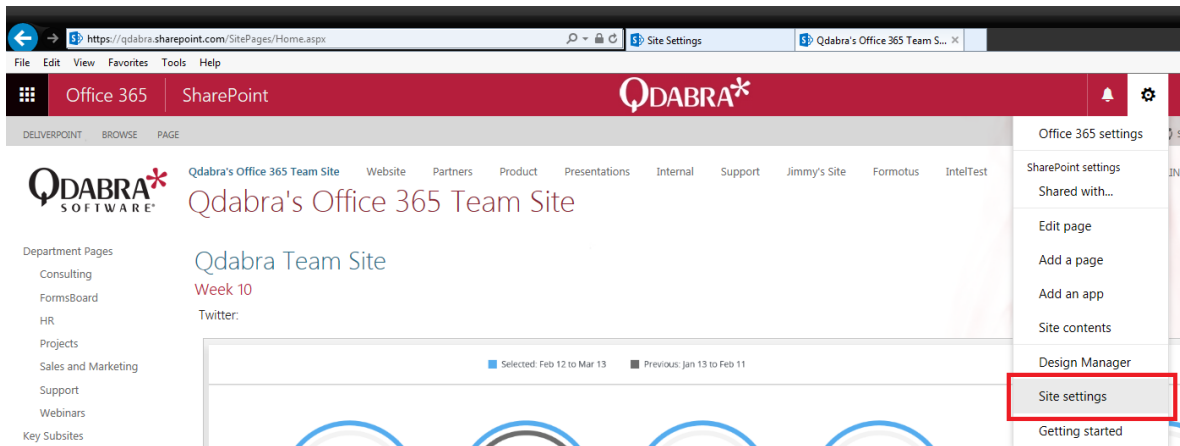
The Target Library (Optional)

This section is optional if you already have a target library which you intend to query for your Bulk Editing Dashboard tool, but for instructional purposes, a re-usable Form Template Library (STP) file is provided for Office 365.

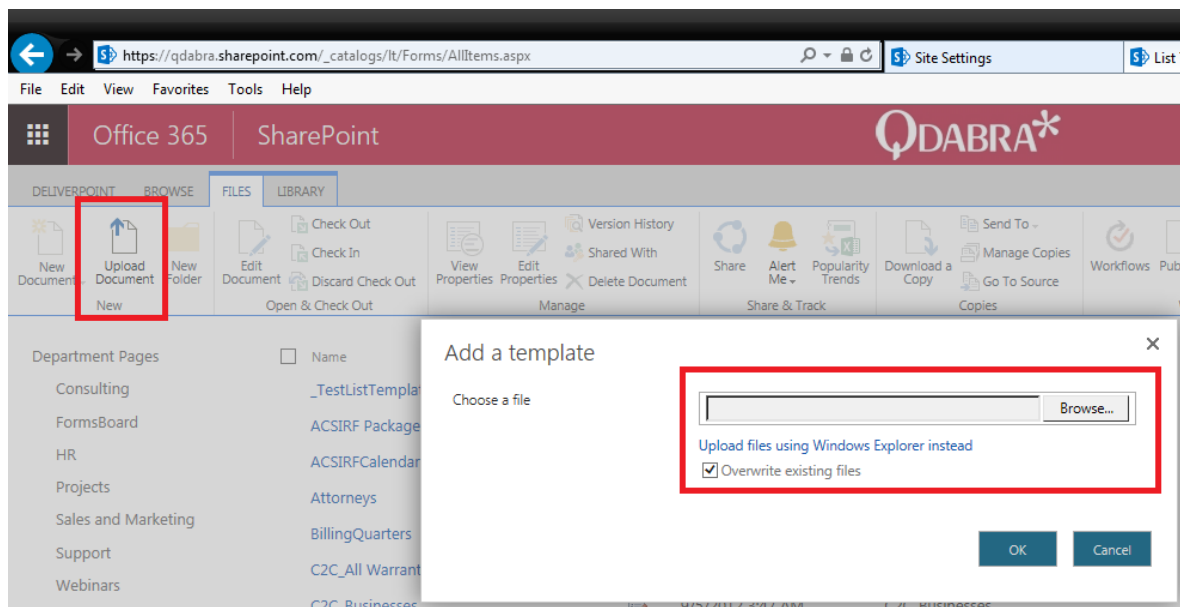
Deploy the site template (STP) file.

- a. Download the STP file [here](#).
- b. Go to the Site Settings (make sure you are on the top level site settings)

InfoPath Dashboards



- c. Under Web Designer Galleries, click List templates.
- d. Upload the **Invoice_Request_Library** STP file.
 - i. Click Upload Document under Files tab
 - ii. Browse for downloaded STP file and click OK



Refer to the link below for a more detailed steps on the web on how to upload an STP template

<https://support.office.com/en-us/article/Manage-list-templates-c3884ad1-bc49-44b8-b3d6-3bc6a01eb393>

Create the Library using the STP file

- a. Proceed to the target site/subsite.

InfoPath Dashboards

- b. Go to Site Contents
- c. Add an app.
- d. Search for the **Invoice_Request_Library** STP file.
- e. Click the App to create the library.
- f. Enter a library name such as **Invoice_Request_Library**

When done, proceed to the library and check out the contents:

Invoice_Request_Library

New

Upload

Sync

Share

More

All Documents

My Documents

...

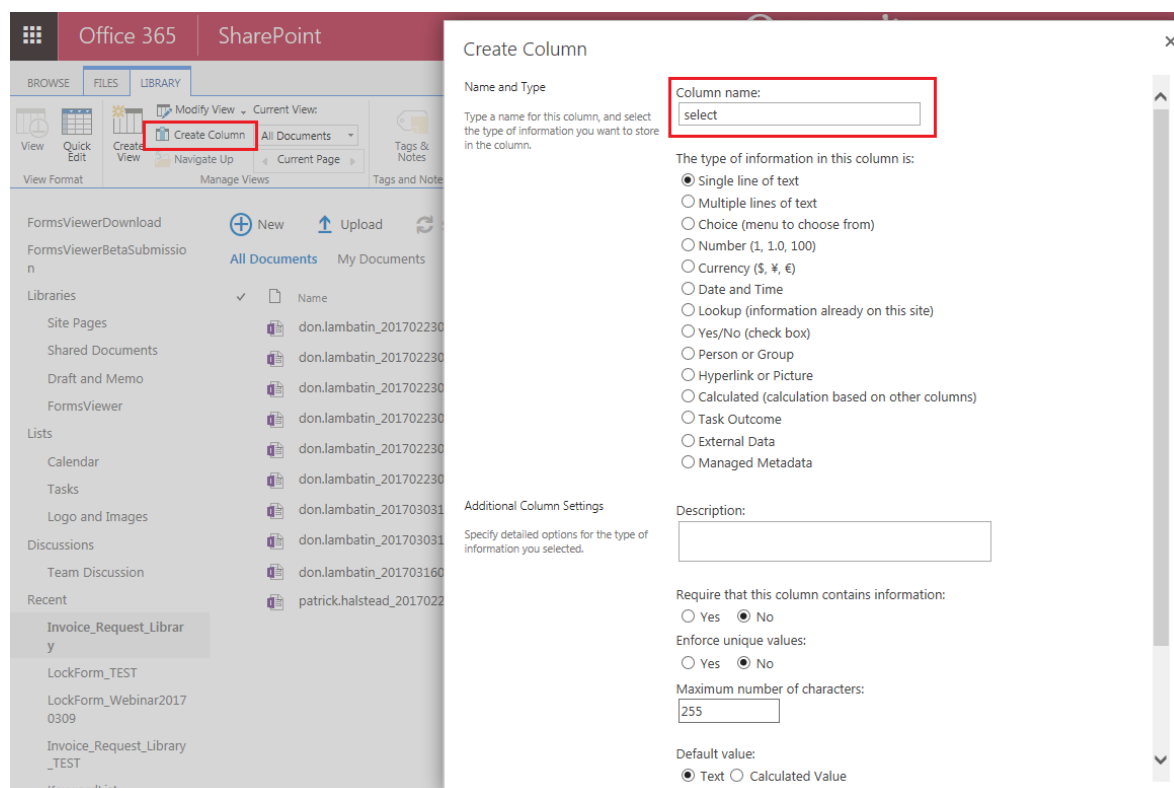
Find a file

✓		Name	Modified	Modified By	Checked Out To	Total	Status	Invoice	Company	ID
		don.lambatin_20170223045617	...	March 5	☐ Don Lambatin	2,304	Not Started	Prepayment	Microsoft	14
		don.lambatin_20170223045717	...	March 5	☐ Don Lambatin	217	Not Started	Expense Report Invoice	UPS	15
		don.lambatin_20170223045749	...	March 5	☐ Don Lambatin	45	Not Started	Expense Report Invoice	McDonalds	16
		don.lambatin_20170223045939	...	March 5	☐ Don Lambatin	800	Not Started	PO Default Invoice	Nike	17
		don.lambatin_20170223050017	...	March 5	☐ Don Lambatin	80	Not Started	Chargeback	Starbucks	18
		don.lambatin_20170223050522	...	March 5	☐ Don Lambatin	15	Not Started	Credit Memo	Reebok	19
		don.lambatin_20170303133418	...	March 5	☐ Don Lambatin	313	Not Started	Credit Memo	QDABRA	22
		don.lambatin_20170303142035	...	March 5	☐ Don Lambatin	314	Not Started	Credit Memo	1111	23
		patrick.halstead_20170224090641	...	March 5	☐ Don Lambatin	123,123	Not Started	PO Default Invoice	Qdabra	20

We will be performing a bulk edit operation on the items above at the end of this guide.

- g. At the **Library** tab, create a new column called **select**.
- h. Leave the rest of the configuration as is and click **OK**.

InfoPath Dashboards



- i. Repeat the same procedure for creating two more columns called **submit** and **success**.

When done, you may choose to submit new items by clicking on the New button.

The 3 newly added columns will be used later on for the Dashboard's Bulk Edit logic.

FormsViewer

This guide assumes that you already have [FormsViewer](#) installed on your SharePoint O365 Environment. Kindly skip this section if you already have the FormsViewer app installed.

If you plan to create this Bulk Edit Dashboard on SharePoint 2016 or lower without FormsViewer, be sure to have Sandbox enabled in order for qRules to run.

To install FormsViewer, download the App along with the guide from the link provided below:

http://www.infopathdev.com/files/folders/formsviewer_v11/entry104513.aspx

Creating the Dashboard

In order to fully understand the fundamental elements of the Bulk Edit Dashboard, you will be creating the dashboard from scratch.

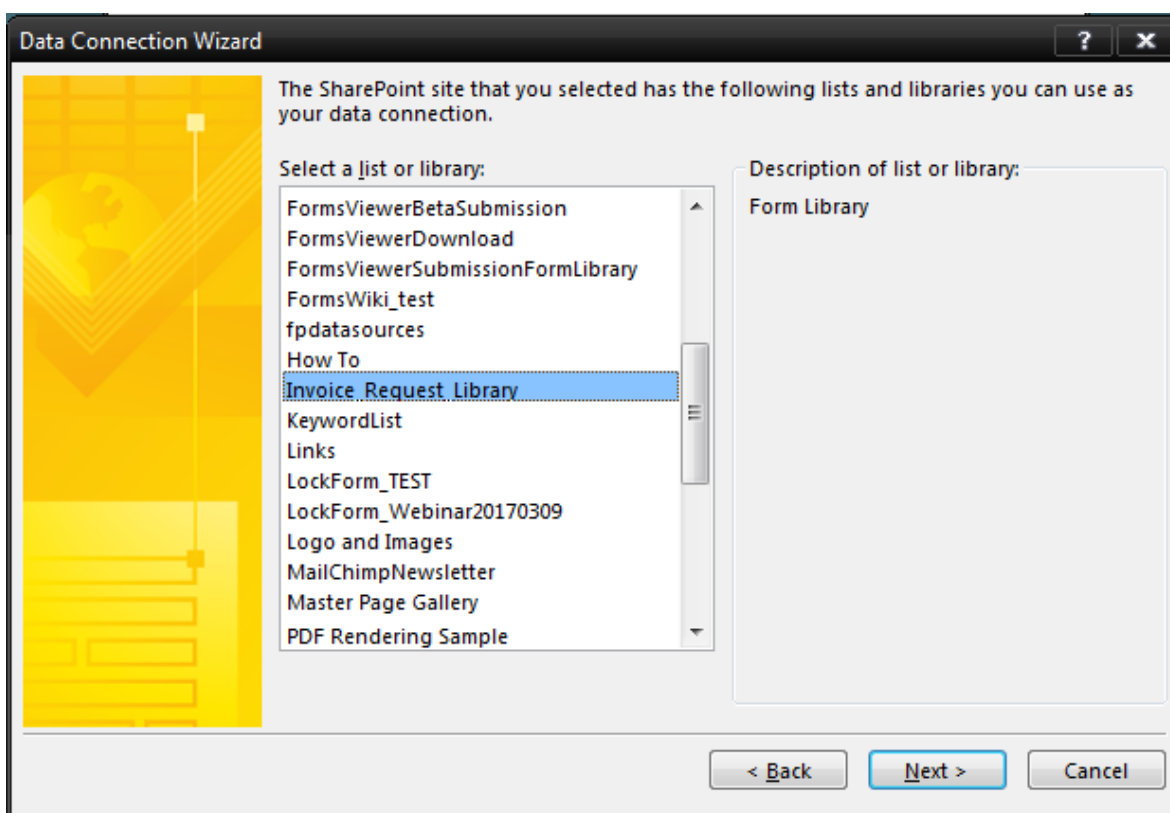
InfoPath Dashboards

Before we proceed, here's a list of data connections needed:

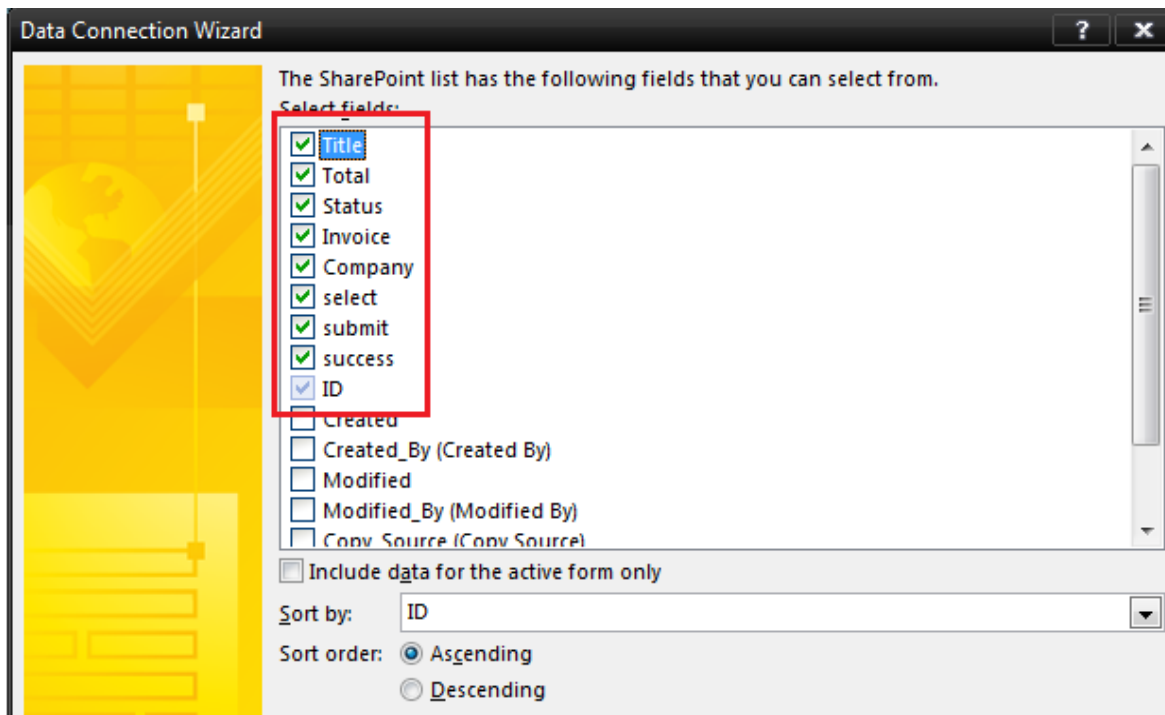
- Retrieve library data connection to query target library – for displaying items intended for bulk editing
- Retrieve XML document data connection – for retrieving the XML document to be updated
- Submit to Library – for submitting retrieved document along with modified content/s

Steps:

1. Open a Blank InfoPath Designer form.
2. Under the Data Tab, click Data Connections and **Add** a data connection which **receives** data from the target **SharePoint library**, in this case the **Invoice_Reques_Library** created earlier.



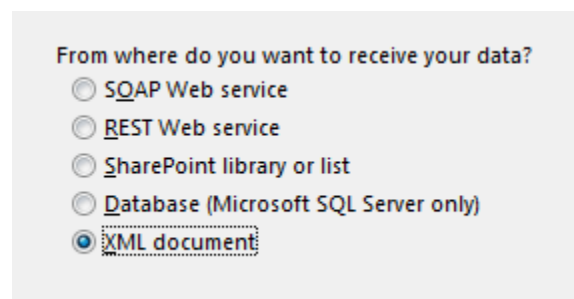
3. At the next page, select the following columns from the SharePoint library:



4. Click **Next** twice until you reach the last page of the data connection wizard.
5. At the last page, **untick** the *Automatically retrieve data* checkbox and click **Finish** to exit the wizard.

Disabling the *Automatically retrieve* checkbox is optional but highly recommended since it enables you to filter out unwanted data on your on load query by specifying a filter at the data connection's query fields before triggering the query action – As far as performance is concerned, we would not want to return all the data when the form loads.

6. Add another data connection which **receives** data from an **XML document**.

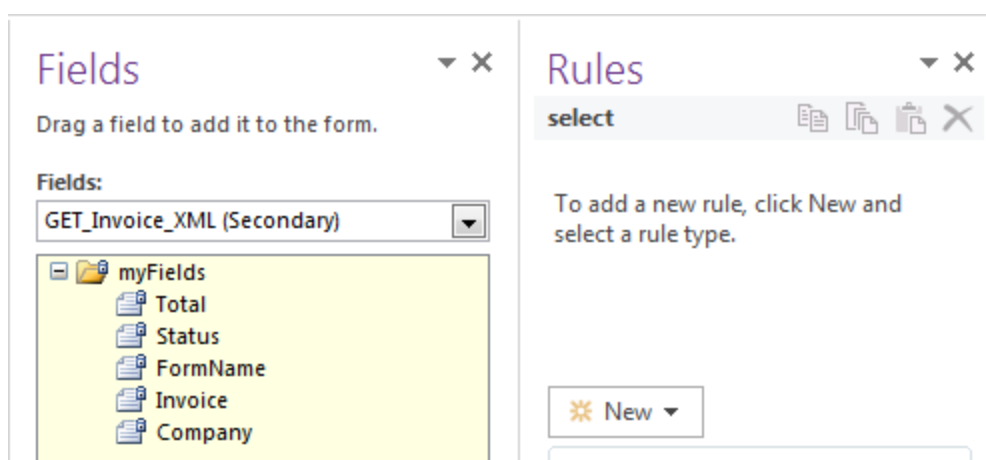


7. Click Next.
8. Enter the **URL to the library**.
9. After the URL, add a slash and enter an existing **File Name (XML)** from the library and add the prefix .xml.

https://qdabra.sharepoint.com/Website/Invoice_Request_Library/don.lambatin_20170303142035.xml

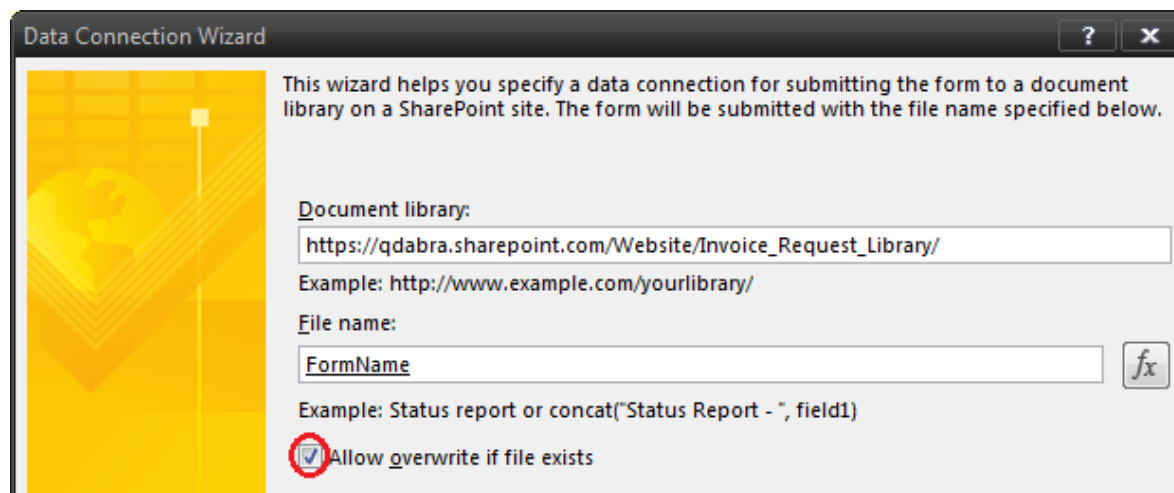
Open up the link in the browser to verify if you have the correct link to your form. Be sure you got the File Name correct otherwise an error will appear if you do not have a correct link and it will not properly open up the form in the browser.

10. Click **Next**.
11. At the next page, select *Access to the data from the specified location*.
12. Click **Next**.
13. Provide a meaningful data connection name such as **GET_Invoice_XML**.
14. Click finish.



Notice that you now have a new secondary data connection source with fields matching the columns from the target library.

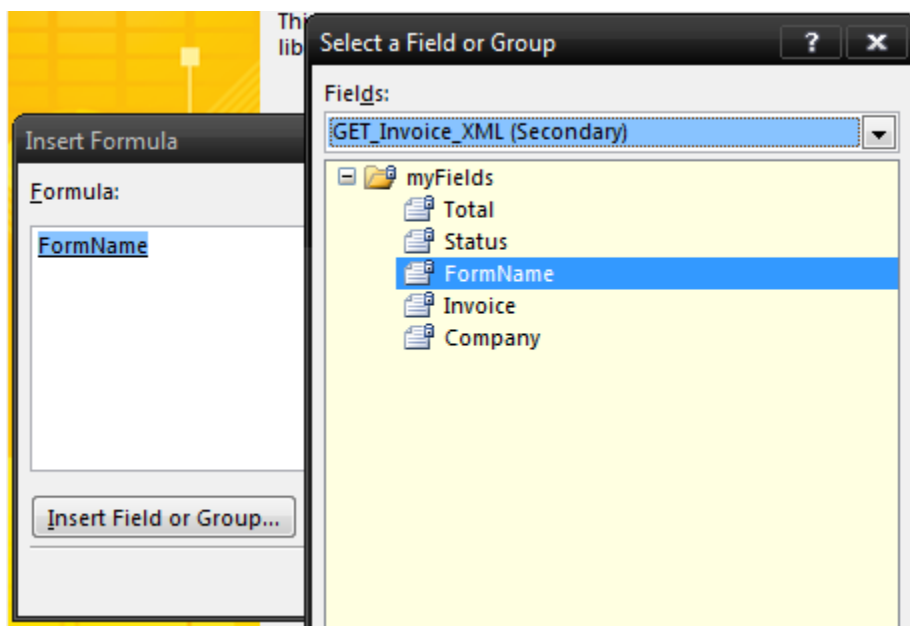
15. **Add a submit** data connection that submits data to the target SharePoint Library:



InfoPath Dashboards

Be sure to specify the correct link to the library.

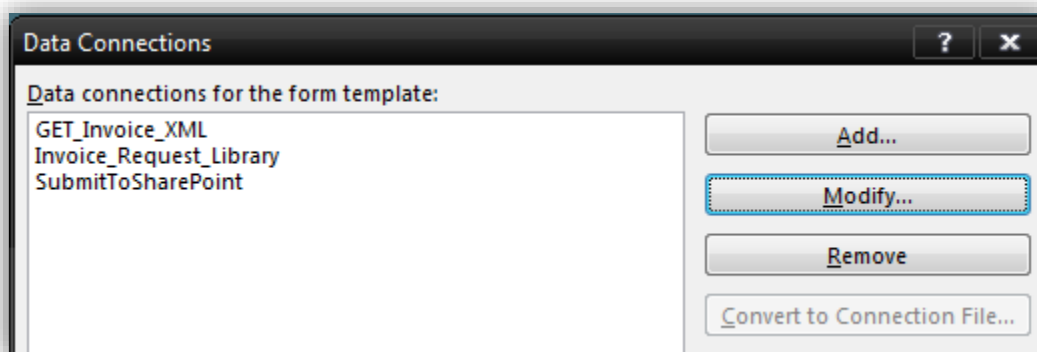
16. Tick the Allow overwrite checkbox – **IMPORTANT!**
17. For the File Name, click the **fx** button.
18. Click **Insert Field or Group**.
19. Select the FormName field from the GET XML data connection:



Note that this field should match the target library's XSN template's selected File Name field declared at the Submit to SharePoint Library data connection wizard.

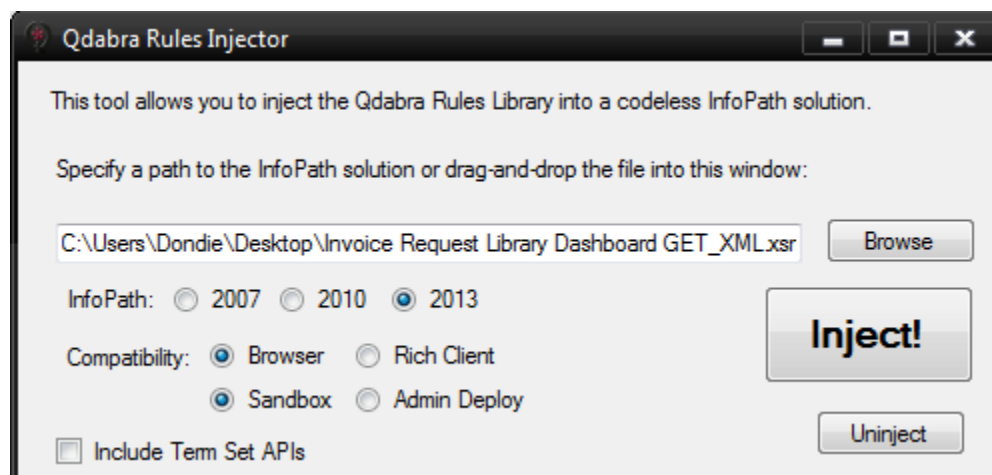
20. Click **Next**.
21. Leave the name as is and click **Finish**.

Your list of data connections should look similar to this:

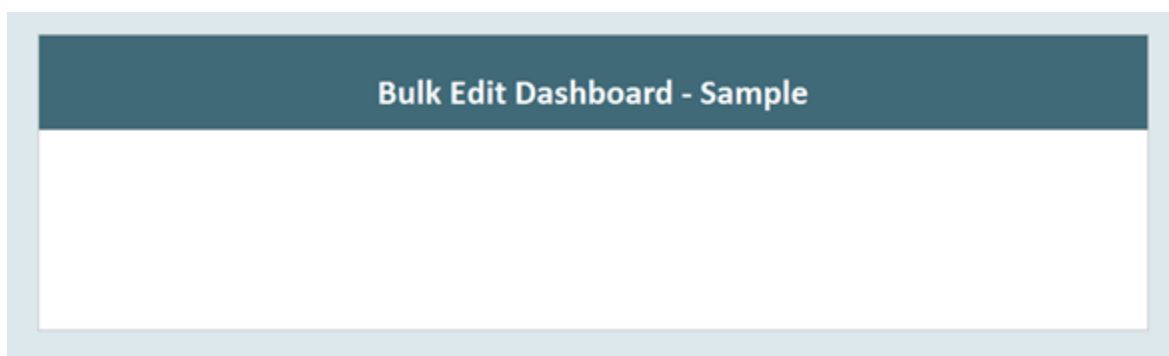


InfoPath Dashboards

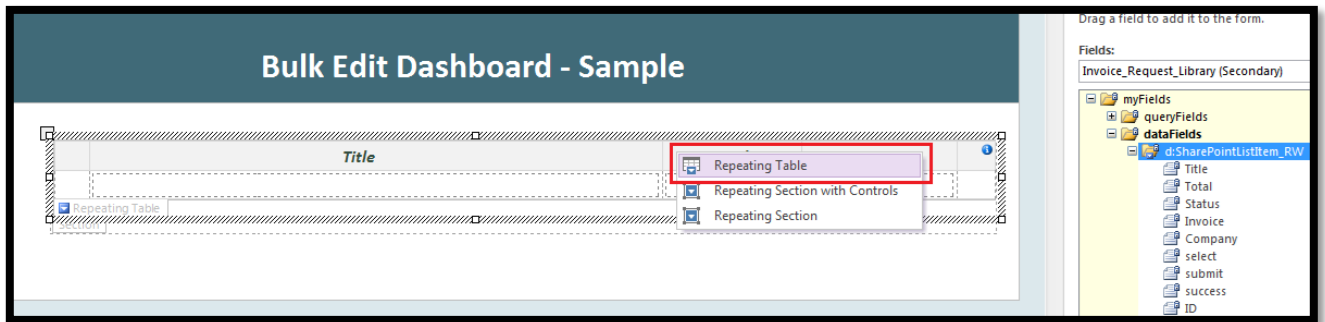
22. Save the form and inject with qRules.
 - a. Go to **File** tab > **Save As** > select desired save location.
 - b. **Exit** form.
 - c. Launch the Qdabra Rules Injector tool by clicking on **Start > Programs > Qdabra > Tools > qRules > qRules Injector**.
 - d. Specify the template by browsing for the Dashboard XSN saved.
 - e. Click **Inject!**



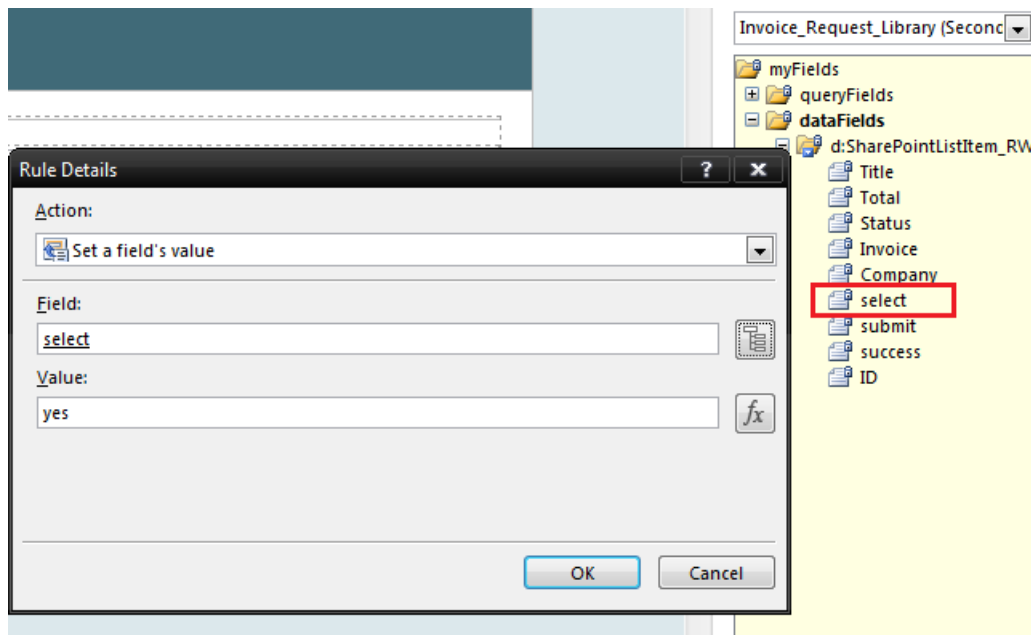
23. Re-open the injected form (**right click** on the XSN and select **design**).
24. Design your layout – Header and Footer



25. At the **Invoice_Request_Library** secondary data connection, layout your repeating table onto the canvas. – Right click on the repeating section, drag it onto the canvas, and select Repeating Table



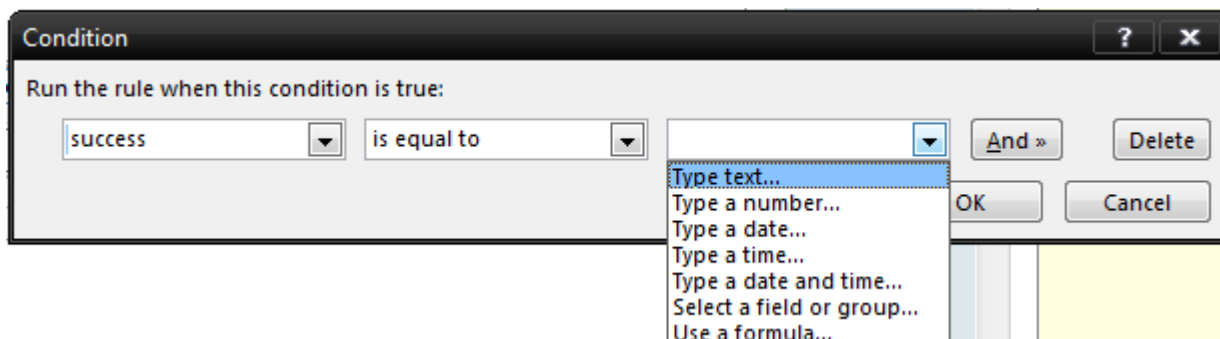
26. Select columns you would like to display.
27. Replace all text boxes with a Calculated Field Box (Right click on the field > Change Control > select Calculated Value).
28. At the left most section of the table, insert a narrow column for the **select check box** field.
29. Change the **select** text field control to a checkbox (right click > properties > change control > select check box).
30. At the header of the **select check box** column, add a button.
31. Under the **Home** tab click on **Manage Rules**.
32. **Add** a rule that sets the value of the **select** field to the value **yes**. Name this rule **Select All**.



33. At the left most section of the table, insert a column for the **success** field.
34. **Click** on the **success** field and change this control to a **calculated field**.
35. Still at the **success** field, **Add** two formatting rules:
 - a. Set the name of the first formatting rule to **Green if success**
 - b. Add a condition where **success** is equal to the word "success"

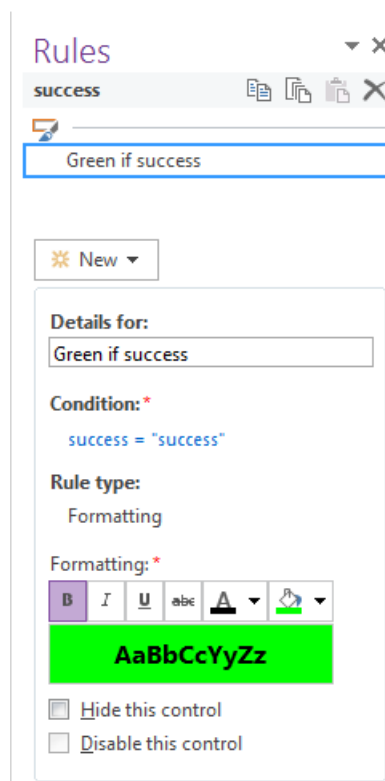
InfoPath Dashboards

- For the first and second dropdown, leave the two selections as is
- For the third dropdown, select Type text and manually enter the work “**success**” without the double quotes.



- At the Formatting section, change the shading to **Green** and set the text font to **Bold**.

Your formatting rule should look similar to this:



- c. Perform the same steps for the second formatting rule where:

InfoPath Dashboards

- Rule name: **Red if failed**
- Condition: **success is equal to "false"**
- Set shading to **Red**, text font to **White** and set it to **Bold**.

The screenshot shows the 'Details for:' section with the rule name 'Red if failed'. The 'Condition:' section shows 'success = "failed"'. The 'Rule type:' is set to 'Formatting'. The 'Formatting:' section shows a text box with the text 'AaBbCcYyZz' in bold white font on a red background. Below the text box are two checkboxes: 'Hide this control' and 'Disable this control'.

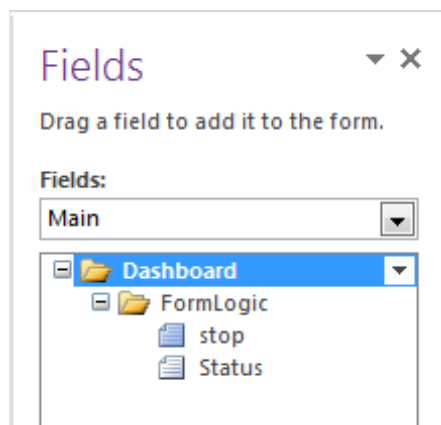
This field will later on display the result of the *SubmitToSharePoint* **qRules** command

36. At the Main Data Source, change the section's name **myFields** to a more meaningful name such as **Dashboard**.
37. **Add** a new section named **FormLogic**.
38. Inside this section, add the fields below and follow the corresponding data type:

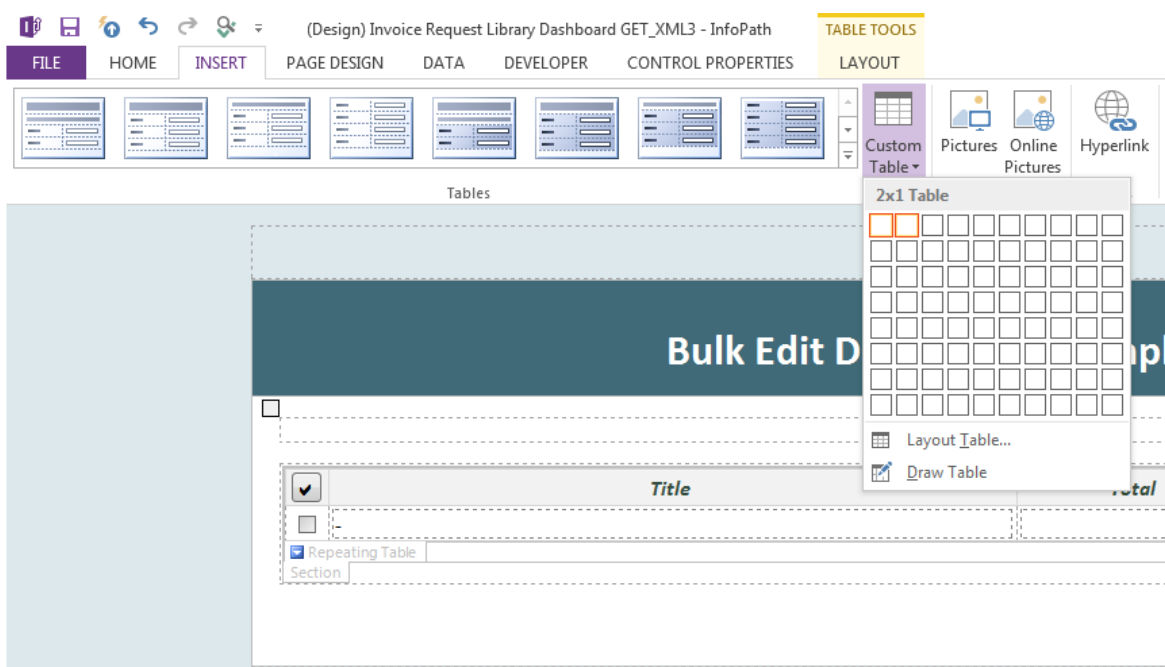
Field Name	Data Type
stop	Field attribute
Status	Field element

Your schema should look similar to this:

InfoPath Dashboards



39. Insert a 2 x 1 Custom Table right above the repeating table – this section will hold the buttons and dropdown control for the bulk edit Status selection.



40. Inside the right section of the custom table, display Status field from the FormLogic section as a dropdown.
41. **Right click** on the dropdown control and manually enter the following choices:

List box choices

☒ Enter choices manually
☐ Get choices from fields in this form
☐ Get choices from an external data source

Value	Display Name	Default
Select...	Select...	Yes
Completed	Completed	
Closed	Closed	
In Progress	In Progress	
On Hold	On Hold	
Not Started	Not Started	

Add...
Modify...
Remove
Move Up
Move Down
Set Default

42. Beside the dropdown control, add a button and rename the button label to **Update**.
43. Add an action rule named **Update**.
 - a. On this rule, **Add** an action that sets a field's value.
 - b. Set the target field to the **Command** node of the **Invoice_Request_Library** data connection.

Rule Details

Action: Set a field's value

Field: Command

Value:

OK Cancel

Select a Field or Group

Fields:

QdabraRules (Secondary)

QdabraRules

- version*
- finishedLoading*
- isSwappingDom*
- expires*
- Command
- Result
- Success
- Error
- View
- Context
- Form
- QueryDataError
- DnsLookup
- ExecutionHistory

OK Cancel

- c. Set this field to the value:

SetValue /xpath=/dfs:myFields/dfs:dataFields/d:SharePointListItem_RW/d:submit[../d:select = "yes"]
/value=1 /dsname=Leave_Requests

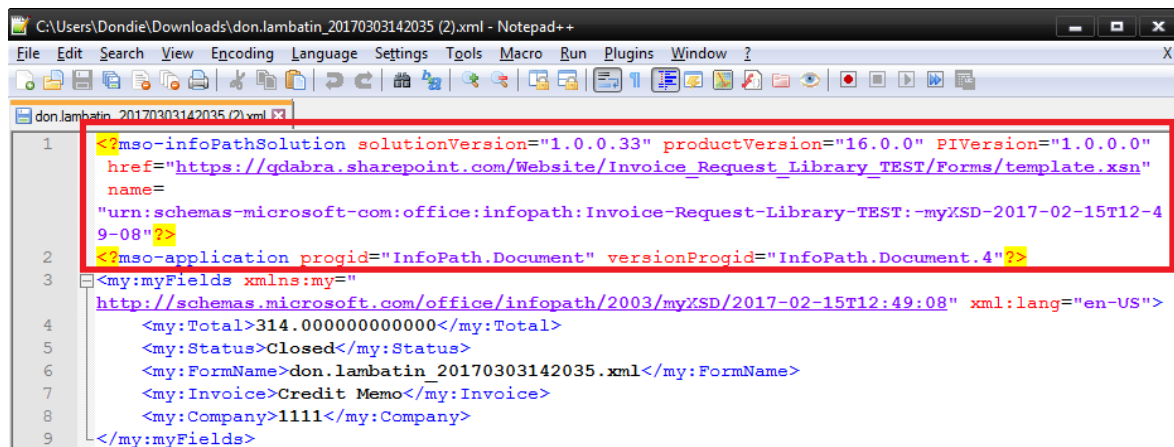
This action is going to set the submit node to the value "1" if the select node is equal to "yes", and trigger a set of actions for updating the xml.

InfoPath Dashboards

44. **Add** a new rule to set the **Status** field of the **FormLogic** section to a blank value – we would like to reset the dropdown field after the bulk edit operation has succeeded.
45. **Add** a *formatting rule* that disable the **Update** button if the value of the **Status** field is blank – this will prevent users from clicking the Update button if there is no new Status selected.
46. At the Left section of the Custom table, add a new button and re-name the button label to **Refresh**.
47. At the **Refresh** button, add an action rule that queries the **Invoice_Request_Library** data connection.

We are now down to the last major component of this guide which will use the following qRules commands:

- a. ChangeConnectionUrl
 - This command will update the XML location specified at the **Get_Invoice_XML** data connection created earlier, to the corresponding **FileName** selected from the displayed items at the **Invoice_Request_Library** repeating table.
- b. InsertPi
 - This command is used to manipulate the InfoPath form's Processing Instructions
 - Since we are only getting the XML content from the **Get_Invoice_XML** data connection, we will insert Processing Instructions that will conform to the target Library's requirements in order for the forms to open up properly in the browser.



```

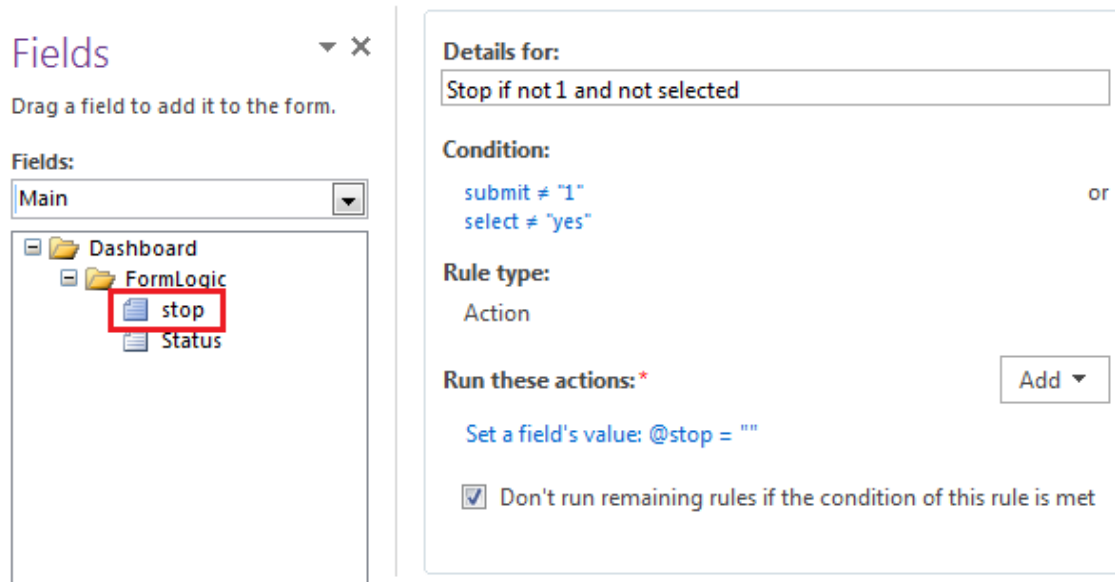
1  <?mso-infoPathSolution solutionVersion="1.0.0.33" productVersion="16.0.0" PIVersion="1.0.0.0"
   href="https://qdabra.sharepoint.com/Website/Invoice_Request_Library_TEST/Forms/template.xsn"
   name=
   "urn:schemas-microsoft-com:office:infopath:Invoice-Request-Library-TEST:-myXSD-2017-02-15T12-4
   9-08"?>
2  <?mso-application progid="InfoPath.Document" versionProgid="InfoPath.Document.4"?>
3  <my:myFields xmlns:my="
   http://schemas.microsoft.com/office/infopath/2003/myXSD/2017-02-15T12:49:08" xml:lang="en-US">
4     <my:Total>314.000000000000</my:Total>
5     <my:Status>Closed</my:Status>
6     <my:FormName>don.lambatin_20170303142035.xml</my:FormName>
7     <my:Invoice>Credit Memo</my:Invoice>
8     <my:Company>1111</my:Company>
9  </my:myFields>
  
```

- c. SubmitToSharePoint
 - This is a bit self-explanatory – this command will be used for submitting the each row separately as XMLs to the target **Invoice_Request_Library**.
48. At the submit node of the **Invoice_Request_Library**, add the following rules:
- a. Stop if not 1 and not selected
 - Rule name: **Stop if not 1**
 - Condition: **submit is not equal to "1"**
 - Click the condition link and leave the first dropdown as is
 - Change the second dropdown to **not equals to**

InfoPath Dashboards

- Select **Type text** for the third dropdown and manually enter the value **1**
- Click the **And** button beside the third dropdown to add a new condition below, and replace this with an **OR** condition.
- Set the first dropdown to the **select** node.
- Change the second dropdown to **not equals to**
- Select **Type text** for the third dropdown and manually enter the value **yes**
- Action target field: **stop** attribute field of the FormLogic section located at the main data source.
- Value: **none** (blank)
- Tick the checkbox to stop all other actions if the condition is met.

Your rule should look similar to this:



The screenshot shows the 'Fields' task pane on the left and the 'Details for:' pane on the right.

Fields:

- Drag a field to add it to the form.
- Fields: Main
- Dashboard
 - FormLogic
 - stop** (highlighted with a red box)
 - Status

Details for:

Stop if not 1 and not selected

Condition:

submit ≠ "1" or select ≠ "yes"

Rule type:

Action

Run these actions: *

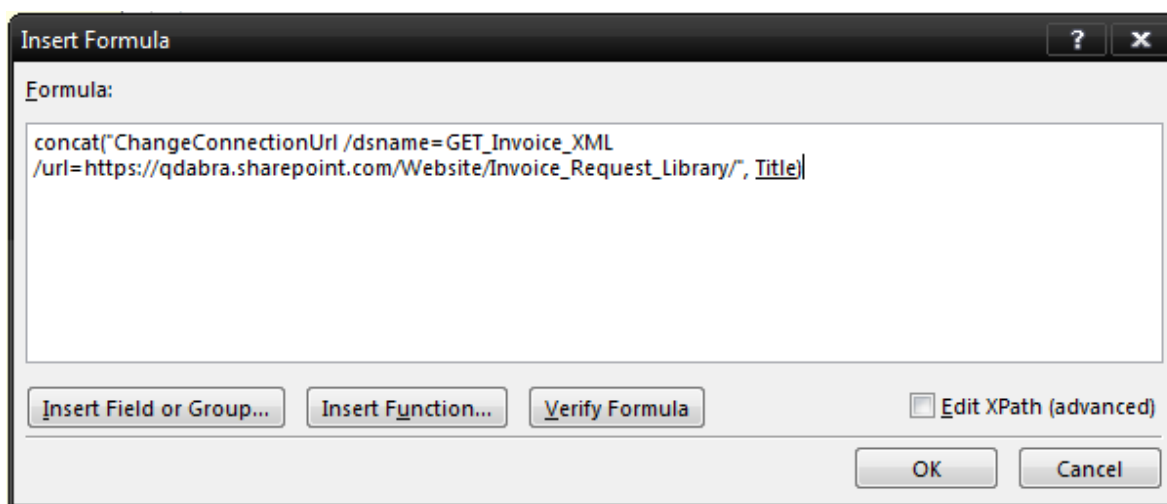
Set a field's value: @stop = ""

☒ Don't run remaining rules if the condition of this rule is met

The objective of this condition is to prevent other rules from running if the submit action is not triggered or if the current row is not selected.

InfoPath Dashboards

- b. ChangeConnectionUrl and query XML
 - Rule name: **ChangeConnectionUrl and query XML**
 - Condition: **none** (blank)
 - Action target field: **Command** node of **qRules** located at the **QdabraRules** data source.
 - Value: click the **fx** button to insert the formula:



`concat("ChangeConnectionUrl /dsname=GET_Invoice_XML
/url=https://qdabra.sharepoint.com/Website/Invoice_Request_Library/", Title)`

In this formula we are calling the qRules command **ChangeConnectionUrl** which will update the XML location specified for the GET XML query.

The parameter **dsname** identifies which data connection will be updated. For this guide we have used the GET XML data connection named **GET_Invoice_XML**.

The parameter url should point to the location of the XML in SharePoint. Since the target library is the Invoice_Request_Library, we will use the link to that library and concatenate that url to the Title field which contains the File Name (or FormName) that was specified in step 19.

Concatenating the syntax to the **Title** field allows us to filter out the query data connection dynamically – each row will have a different URL.

The result of this formula should be similar to the URL specified in step 9.

- Add a data connection to query the Get_Invoice_XML data connection

InfoPath Dashboards

Your rule should look similar to this:

Details for:

ChangeConnectionUrl and query XML

Condition:

None - Rule runs when field changes

Rule type:

Action

Run these actions: * Add ▾

Set a field's value: Command = concat("ChangeConnectionUrl /dsname=GET_Invoice_X..")

Query using a data connection: GET_Invoice_XML

☐ Don't run remaining rules if the condition of this rule is met

c. Update current Status

- Rule name: **Update current Status**
- Condition: **none**
- Action target field: **Status** field of the **Get_Invoice_XML** data connection
- Value: **Status** field of the FormLogic section from the Main data source

Your rule should look similar to this:

Fields

Drag a field to add it to the form.

Fields:

GET_Invoice_XML (Secondary)

myFields

Total

Status

FormName

Invoice

Company

Rules

submit

Details for:

Update current Status

Condition:

None - Rule runs when field changes

Rule type:

Action

Run these actions: * Add ▾

Set a field's value: Status = Status

☐ Don't run remaining rules if the condition of this rule is met

Fields:

Main

Dashboard

FormLogic

stop

Status

Before proceeding to the next rule, we must first understand what the parts of the Processing Instructions are.

InfoPath Dashboards

Download a sample XML from the **Invoice_Request_Library** and open the file using a text editor such as notepad. Here's how the first Processing instruction should look like:

```
<?mso-infoPathSolution solutionVersion="1.0.0.33" productVersion="16.0.0"
PIVersion="1.0.0.0"
href="https://qdabra.sharepoint.com/Website/Invoice_Request_Library/Forms/
template.xsn" name="urn:schemas-microsoft-com:office:infopath:Invoice-
Request-Library-TEST:-myXSD-2017-02-15T12-49-08"?>
```

In the sample above, we have the Processing Instruction name located at the beginning of the string: **mso-infoPathSolution**. The succeeding **data** are the contents for this particular Processing Instruction.

Let's build the InsertPi command.

```
InsertPi /name=mso-infoPathSolution /data=solutionVersion="1.0.0.33" productVersion="16.0.0"
PIVersion="1.0.0.0"
href="https://qdabra.sharepoint.com/Website/Invoice_Request_Library/Forms/template.xsn"
name="urn:schemas-microsoft-com:office:infopath:Invoice-Request-Library:-myXSD-2017-02-15T12-49-08"
/dsname=GET_Invoice_XML /update=true
```

Here we have declared the name of the **Processing Instruction** along with the **data** content. We have also declared the parameter **dsname** so that qRules identifies where we would like to add the processing instructions to, and that we are **updating** the data that is already existing.

For the first action for this command, we will

d. Insert PI

- Rule name: **Insert PI**
- Condition: **none**
- First Action:
 - Target field: **Command** node of **qRules**
 - Value:

```
InsertPi /name=mso-infoPathSolution /data=solutionVersion="1.0.0.33" productVersion="16.0.0"
PIVersion="1.0.0.0"
href="https://qdabra.sharepoint.com/Website/Invoice_Request_Library/Forms/template.xsn"
name="urn:schemas-microsoft-com:office:infopath:Invoice-Request-Library:-myXSD-2017-02-15T12-49-08"
/dsname=GET_Invoice_XML /update=true
```

Note that this is a hardcoded value and not a formula

- Second Action:
 - Target field: **Command** node of **qRules**
 - Value:

```
InsertPi /name=mso-application /data=progid="InfoPath.Document" versionProgid="InfoPath.Document.4"
/dsname=GET_Invoice_XML /update=true
```

InfoPath Dashboards

e. Submit

- Rule name: **Submit**
- Condition: **none**
- Action target field: **Command** node of **qRules**
- Value:

[SubmitToSharePoint /dssubmit=SharePoint Library Submit /dsname=GET_Invoice_XML](#)

In this command, we are using the **SharePoint Library Submit** data connection for the submit operation, and we declared that the **GET_Invoice_XML** is the data source that which will be submitted.

f. Set Success

- Rule name: **Set Success**
- Condition:
 - Click the condition link and set the first dropdown to the **Result** node of **qRules**
 - Leave the second dropdown to **equals to**
 - Select **TRUE** for the third dropdown
- First Action
 - Target field: **success** node of the current data connection
 - Value: Type in the word "**success**" without the double quotes
- Second Action
 - Target field: **Status** node of the current data connection
 - Value: **Status** field of the FormLogic section from the Main data source

Your action should look similar to this:

Details for:

Set Success

Condition:

Success = TRUE

Rule type:

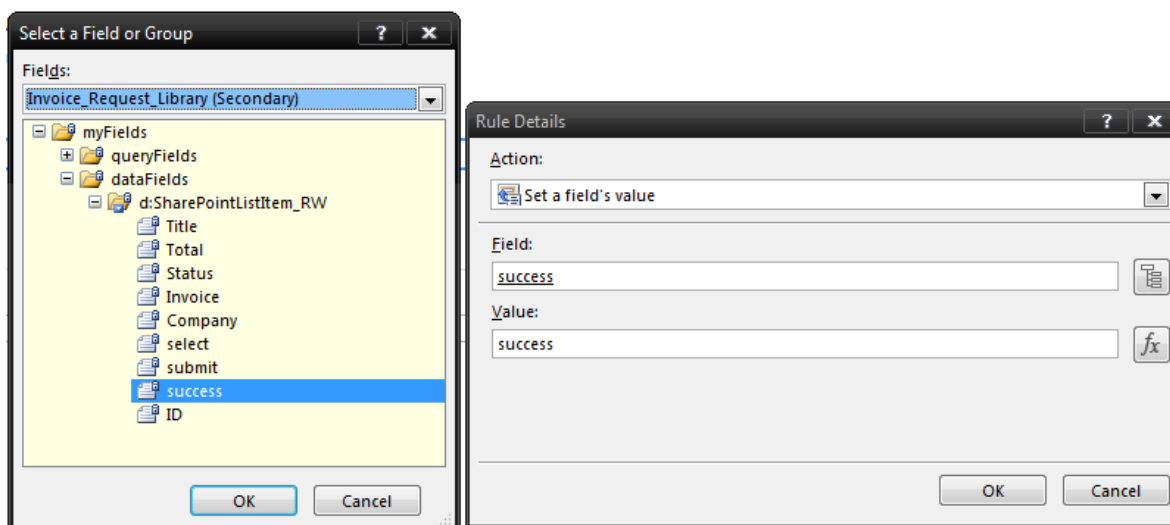
Action

Run these actions: *

Set a field's value: success = "success"
Set a field's value: Status = Status

☐ Don't run remaining rules if the condition of this rule is met

InfoPath Dashboards

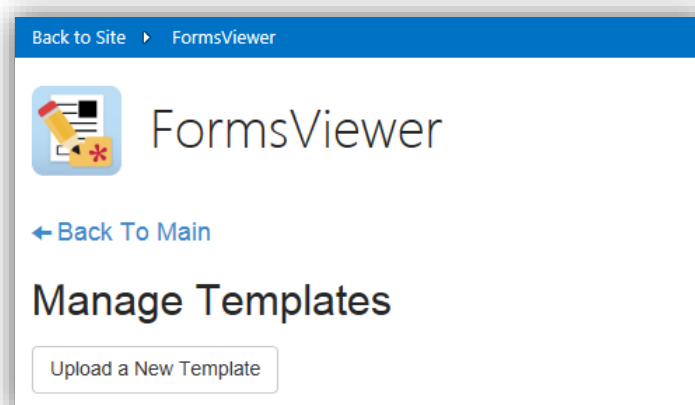


g. Set Failed

- Rule name: **Set Failed**
- Condition:
 - Click the condition link and set the first dropdown to the **Result** node of **qRules**
 - Leave the second dropdown to **equals to**
 - Select **FALSE** for the third dropdown
- Action target field: **success** node of the current data connection
- Value: Type in the word **"failed"** without the double quotes

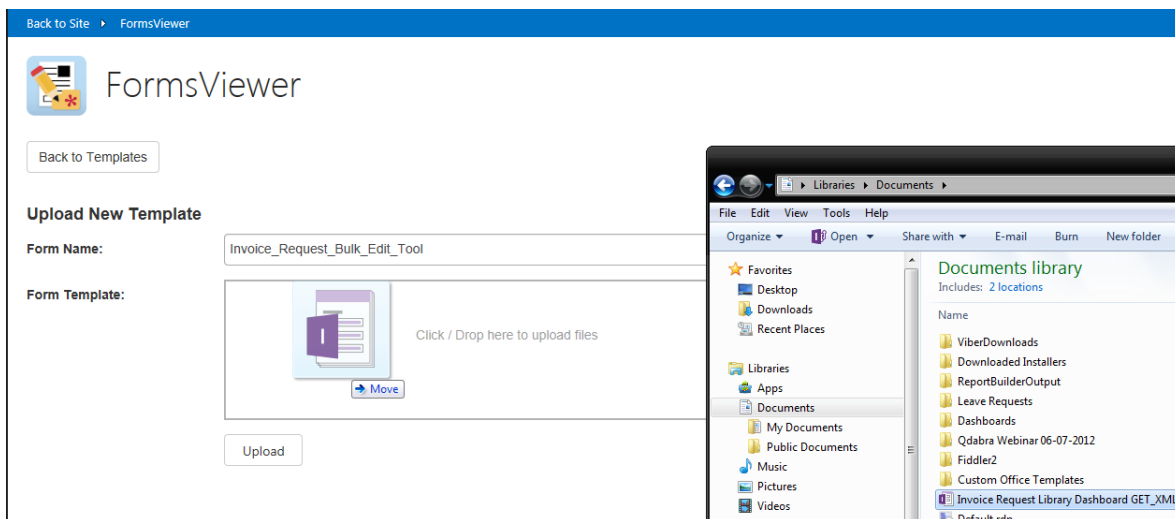
When done, upload your form to FormsViewer.

At the FormsViewer site, click on Upload a New Template.



InfoPath Dashboards

Provide a meaningful name, drag the XSN template, and click on the Upload button.



When done, click **Open Form** to view and test your Dashboard:

Bulk Edit Dashboard - Sample				
Refresh		Select...	Update	
☒	Title	Total	Status	
☐	Prepayment - Microsoft	\$2304.00	Completed	
☒	PO Default Invoice - Nike	\$800.00	Not Started	success
☐	PO Default Invoice - Qdabra	\$123123.00	Completed	
☐	Credit Memo - test	\$1608.00	Completed	
☒	Expense Report Invoice - UPS	\$217.00	Not Started	success
☒	Expense Report Invoice - McDonalds	\$45.00	Not Started	success
☐	Chargeback - Starbucks	\$80.00	In Progress	
☐	Credit Memo - Reebok	\$15.00	In Progress	
☒	Credit Memo - QDABRA	\$313.00	Not Started	success
☐	Credit Memo - 1111	\$314.00	In Progress	