

qRules SubmitToSharePoint List User Guide

Product: Qdabra InfoPath Accelerator (qRules)

This user guide assumes that qRules has already been installed on your local machine. If not, please download qRules from [here](#) and refer to the installation guide at [this](#) link to get started. This guide is for use with versions 6.1 and later of qRules – if you are using an earlier version, please contact support@qdabra.com for appropriate documentation.

Alongside the qRules’ SubmitToSharePointList command is Qdabra’s InfoPath to **SharePoint List Tool**, which is included during the installation of qRules.

This tool helps resolve one of InfoPath’s and SharePoint’s limitations – publishing a form to a form library/document library with the ability to send repeating data to a custom list.

Many users experience this problem, and so Qdabra provides a quick and easy solution.

TABLE OF CONTENTS

What’s new in qRules SubmitToSharePointList?	2
The Mapping Tool	2
Installing the Mapping Tool	2
Walkthrough for SubmitToSharePointList	5
Working with Attachments	21
Working with Existing Data & Submitting a Single Mapping	33
Use SubmitToSharePointList with a Calendar	41
Submitting a Person/Group Picker to a SharePoint list	42
General Mapping Overview	51
Support	53
Contact Information	53

What's new in qRules SubmitToSharePointList?

Before we begin, we want to first assure you that the SubmitToSharePointList command still remains backward compatible. If you have a pre-existing form that uses SubmitToSharePointList, you can re-inject your form with a later qRules version, and the command will keep working.

In any case, the SubmitToSharePointList command, though backwards compatible, has been updated to incorporate hierarchical list functionality, allowing relationships between parent and child lists.

For more information, please visit [Qdabra InfoPath Accelerator \(qRules\)](#)

The Mapping Tool

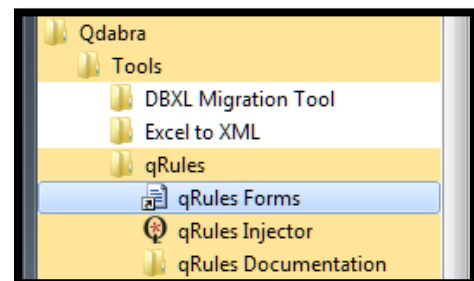
The Mapping Tool (also called the “Qdabra InfoPath to SharePoint List Tool”) allows users to map an InfoPath form’s XML data to a SharePoint list for qRules v4.2 and later versions – bulk import multiple XMLs in one step.

The latest version of the Mapping Tool was updated to accommodate the changes made to the SubmitToSharePointList command. The UI has been updated to show parent-child relationships, and a Parent List ID option has been added.

Installing the Mapping Tool

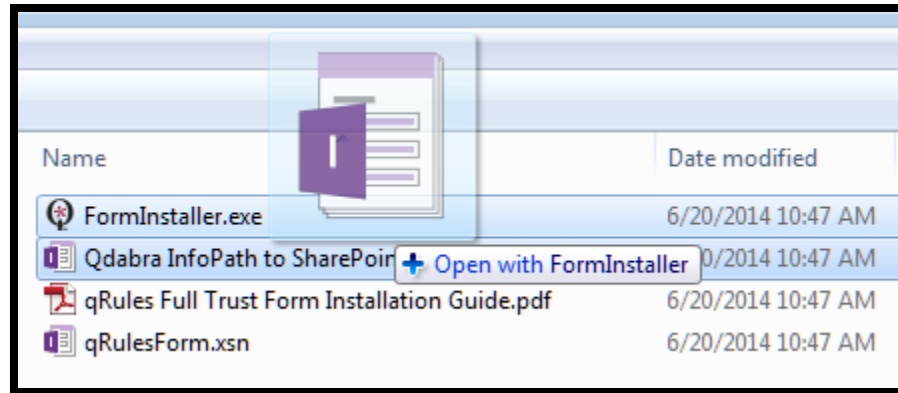
- At the Start menu, proceed to the folder where qRules is installed and select **qRules Forms**. This should open up the folder where the InfoPath to SharePoint List Tool is located.

Start Menu → Tools → qRules → qRules Forms

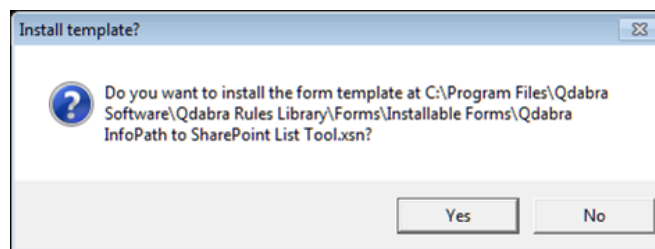


- At the folder location, drag the Qdabra InfoPath to SharePoint List Tool onto the FormInstaller.exe to install the tool.

- A

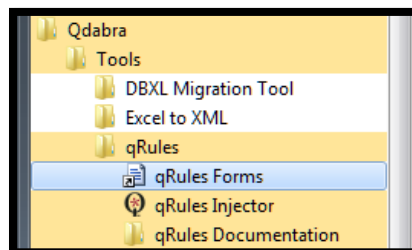


command prompt will appear with an option to install the SharePoint List Tool.

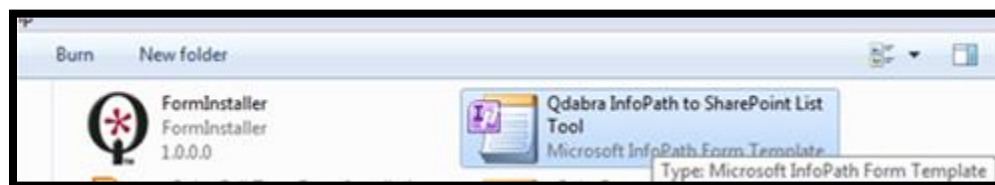


- Select **Yes** to install (close all InfoPath instances before clicking the button).
- A Success Notification will appear once the tool has been installed.

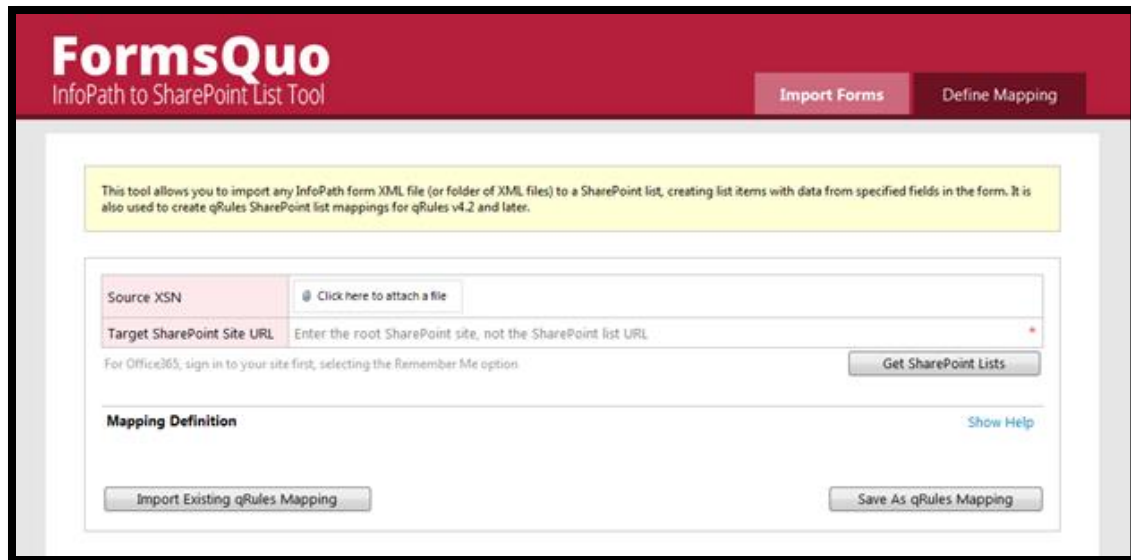
Once installed, navigate your way back to the qRules Forms folder:



Clicking on qRules Forms in the start menu opens a folder:

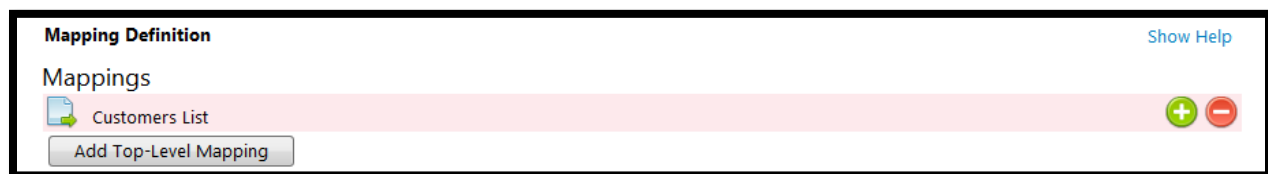


Click on the SharePoint List Tool template to open the Mapping Tool:

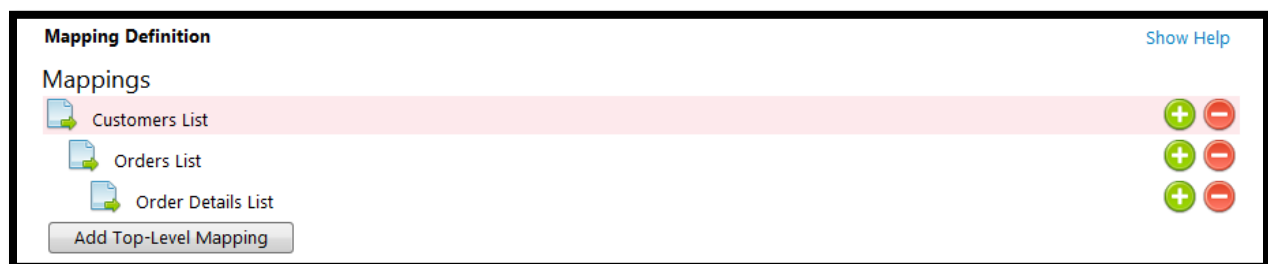


The new Mapping Tool, similar to the previous version, will still require the user to attach the XSN and provide the URL of the target SharePoint site in order to have the fields mapped suitably to the site.

Clicking the Get SharePoint Lists button will open up a section and display the new hierarchical list functionality.



This is an image of a mapping with a hierarchical structure:



In the mapping tool, you can indicate which field you would like to store the SharePoint list item id in – this replaces the /id parameter. You do not have to specify an id field, of course – there are use cases where you wish to only submit items from your form to the list, but not leave the form connected to the list item. If you have a field that contains attachments you wish to include with your list items, you can also specify those.

Form Field	Options(Hide)	SharePoint Column
my:myFields/my:Customer/my:CompName	☐ Id ☐ Rich Text ☐ Attachment	Company Name

Walkthrough for SubmitToSharePointList

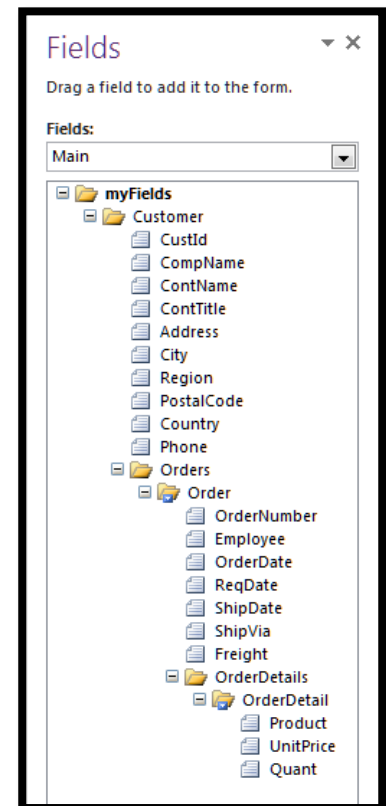
The Form

First, we'll create a simple form or download a pre-created form [here](#).

On the pre-created form you will notice a multiple layered nesting. This is where the new mapping tool becomes very useful in determining the hierarchical structure of the form.

Unlike older versions, qRules v6.1 and later no longer require ID field mappings for building up Parent-Child relationships between lists. The Mapping Tool now offers a new option for the Parent ID look up which will be dealt with later on.

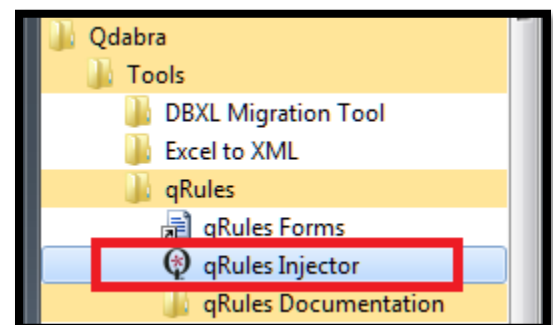
Here is the schema:



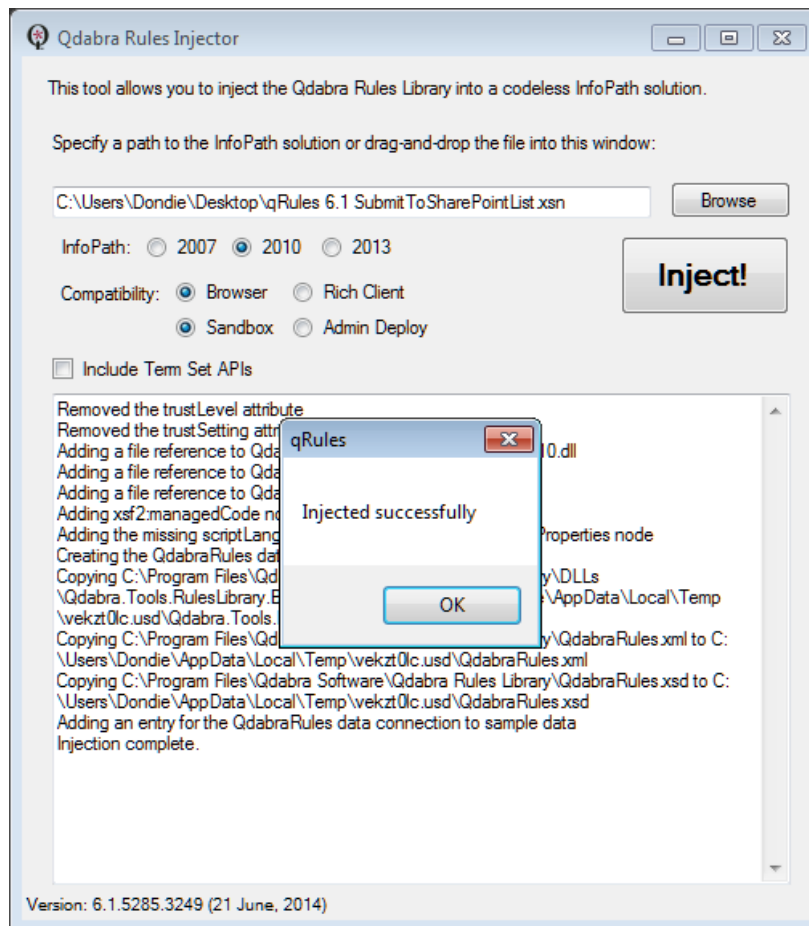
Before we define the mapping, let us first inject the form with qRules.

At the Start Menu, navigate to the qRules folder located under the Qdabra folder and run the qRules Injector.

Start Menu → Tools → qRules → qRules Injector



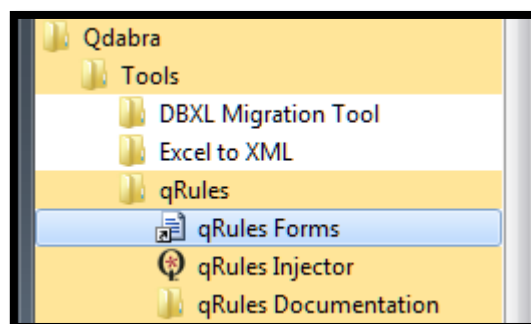
Browse for the desired form and click the **Inject** button. A Success Notification will appear once the form has been injected.



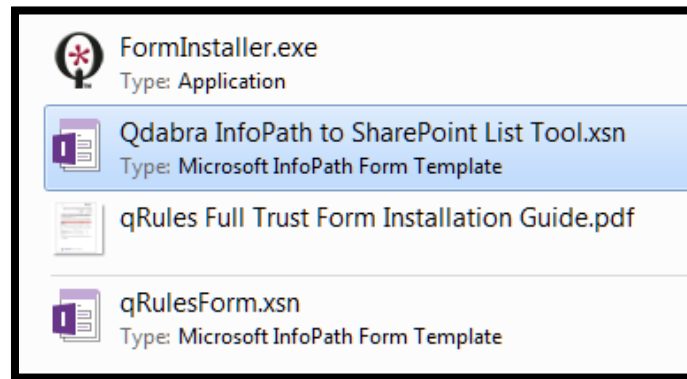
The Mapping

After the form has been injected with qRules, we'll now create our mapping file. If you do not have the InfoPath to SharePoint List Tool form yet that comes with qRules, you'll need to have it installed - you can find more information at the qRules user guide or refer to the previous section.

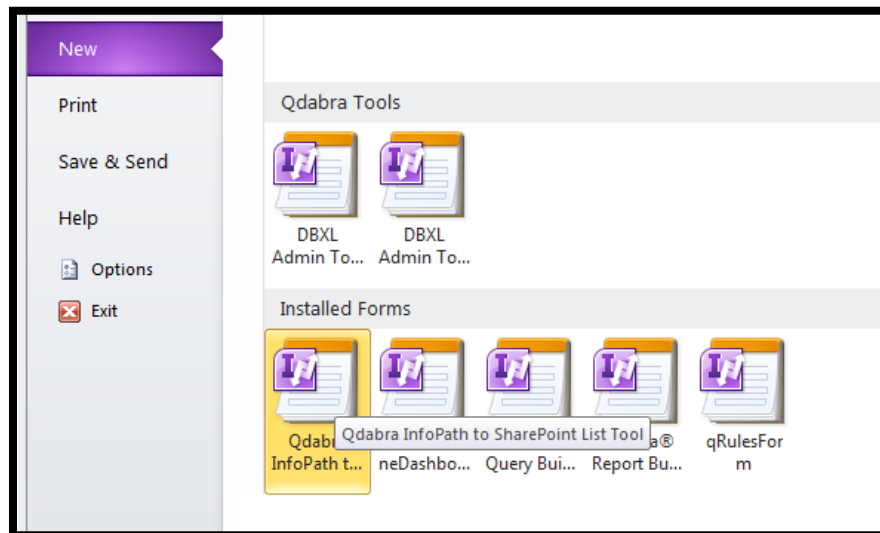
Navigate to the Qdabra qRules folder and select qRules Forms:



Clicking on qRules Forms in the start menu opens a folder:



The mapping tool is there, along with an installer for the form and instructions on how to install the form. Once the form has been installed, you can either access it directly from the folder, or by opening InfoPath Filler and opening the form from there:



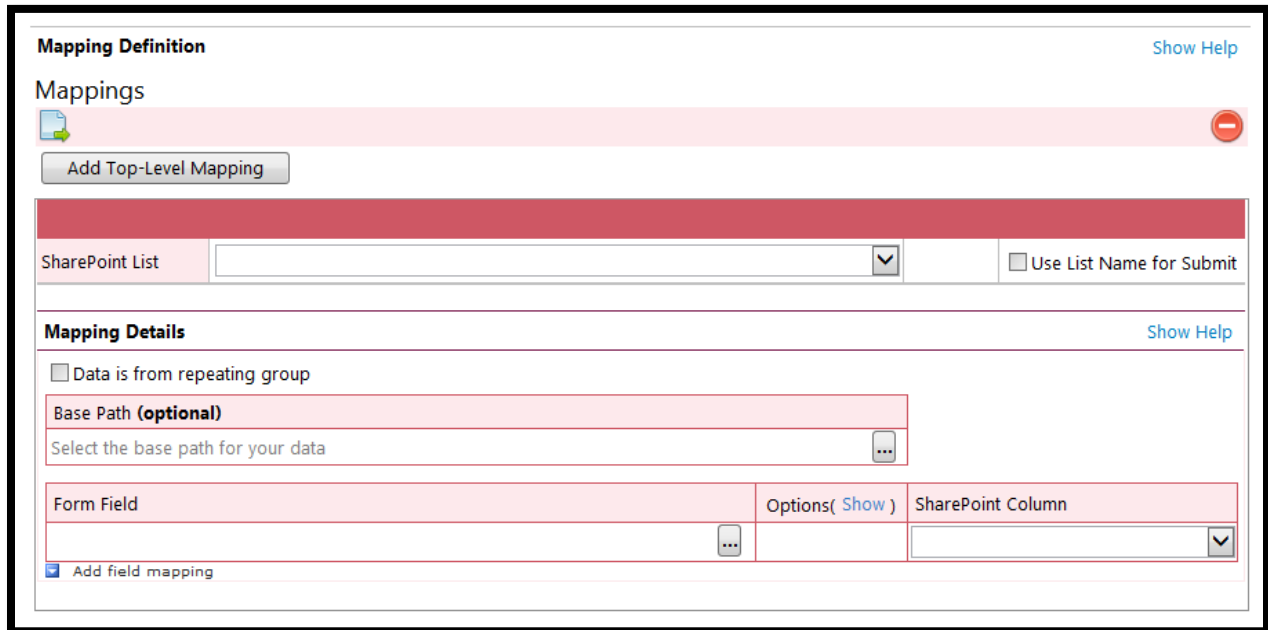
Under the *Define Mapping* tab, attach your XSN file and enter your SharePoint URL:

Source XSN	 qRules 6.1 SubmitToSharePointList.xsn Microsoft InfoPath Form Template 131 KB
Target SharePoint Site URL	http://au.formsboard.com/
For Office365, sign in to your site first, selecting the Remember Me option	
Get SharePoint Lists	
Mapping Definition Show Help	

Click the *Get SharePoint Lists* button to expand the *Mapping Definition* section of the form and allow you to pick the lists you wish to map to.

Our lists will be submitted to in the order they are mapped. In our form, the parent list is the Customers list, then the Orders list, and finally, the Order Details list. So, we will map our data in that order. Also, in our form, the Customer is not repeating, but we can have multiple orders and multiple order details.

Upon clicking the *Get SharePoint Lists* button, a section below will appear:



Mapping Definition [Show Help](#)

Mappings

[Add Top-Level Mapping](#)

SharePoint List		☐ Use List Name for Submit
-----------------	----------------------	---

Mapping Details [Show Help](#)

☐ Data is from repeating group

Base Path (optional)

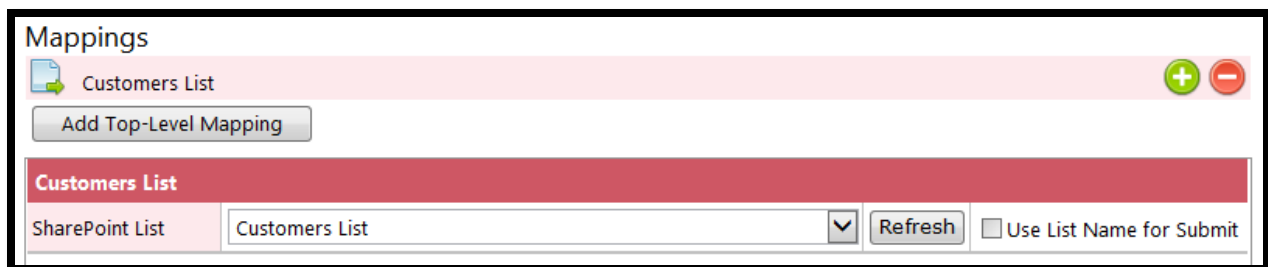
Select the base path for your data 

Form Field	Options(Show)	SharePoint Column

 [Add field mapping](#)

Notice the Add Top-Level Mapping and **Red –** button, this is one of the updates we have made for the Mapping Tool which allows users to create a hierarchy within the form and display the Parent-Child relationship between sections/repeating sections – we will further discuss this new update as we progress through the guide.

First, we'll map our customer information. Select the Customers list from the SharePoint List drop down control:



Mappings

 **Customers List**  

[Add Top-Level Mapping](#)

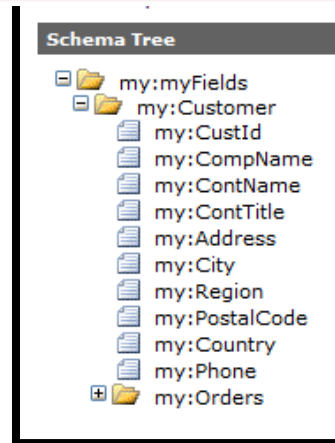
Customers List		
SharePoint List	Customers List	Refresh ☐ Use List Name for Submit

Click the “...” button next to the FormField:

Base Path (optional)	
Select the base path for your data	
Form Field	Options(Show)

☒ Add field mapping

This will display the form fields in the task pane:



Double click the **CustId** field and select the appropriate SharePoint column from the drop-down control.

Form Field	Options(Show)	SharePoint Column
my:myFields/my:Customer/my:CustId		Customer ID

Map the rest of the fields - these are all from the Customer group in the form template schema:

Form Field	Options(Show)	SharePoint Column
my:myFields/my:Customer/my:CustId		Customer ID
my:myFields/my:Customer/my:CompName		Company Name
my:myFields/my:Customer/my:ContName		Contact Name
my:myFields/my:Customer/my:ContTitle		Contact Title
my:myFields/my:Customer/my:Address		Address
my:myFields/my:Customer/my:City		City
my:myFields/my:Customer/my:Region		Region
my:myFields/my:Customer/my:PostalCode		Postal Code
my:myFields/my:Customer/my:Country		Country
my:myFields/my:Customer/my:Phone		Phone

☒ Add field mapping

We are now going to map the Orders group to the Orders list. In order to proceed, let us proceed back to the hierarchy section. Notice that a new button has appeared which will allow the user to add a child mapping.



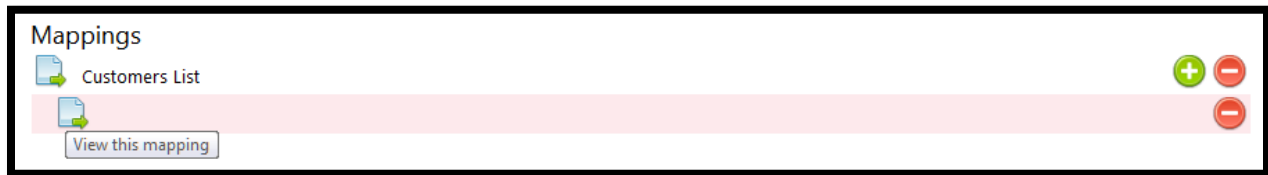
Mappings

Customers List

Add Top-Level Mapping

Add a child mapping

Click the **Green +** button to insert another SharePoint Mapping:



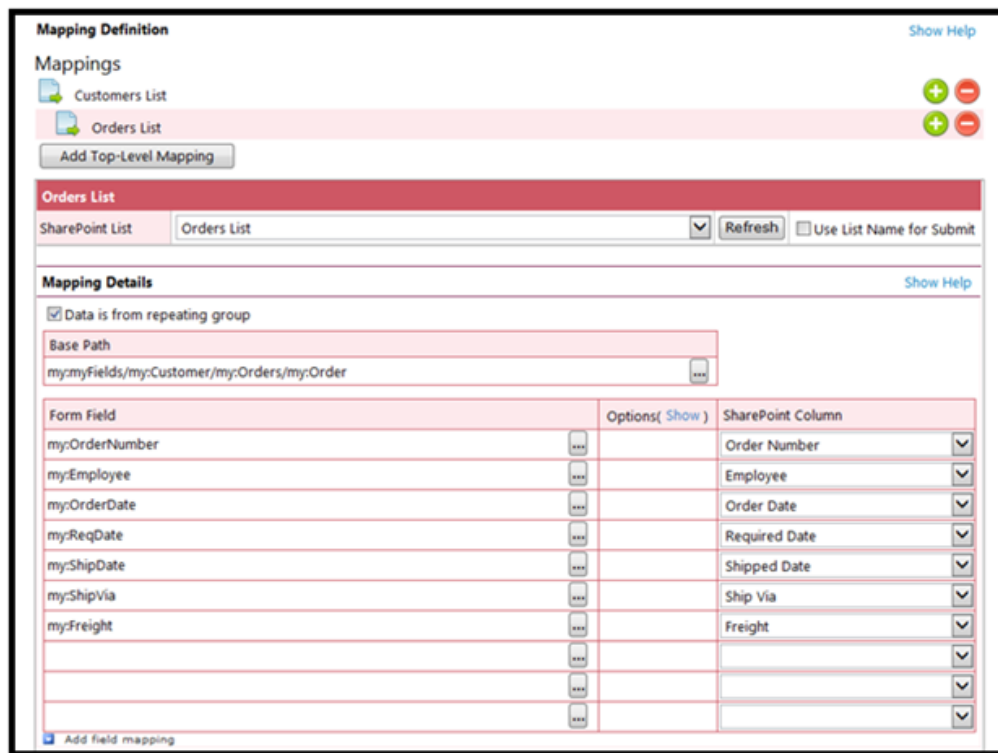
Mappings

Customers List

View this mapping

We want to map our lists in the order we want them to submit, as mentioned above, since qRules will process the mapping file sequentially (unless we specify the /mappingName parameter in the command). Because we want to submit our new Customer list item Id with our Order for the Customer lookup, we want the Customer mapping to submit first, so the field we've indicated as the Id is populated prior to submitting the Order.

Here's a partially completed Order mapping:



Mapping Definition

Show Help

Mappings

Customers List

Orders List

Add Top-Level Mapping

Orders List

SharePoint List: Orders List

Refresh

Use List Name for Submit

Mapping Details

Show Help

☒ Data is from repeating group

Base Path

my:myFields/my:Customer/my:Orders/my:Order

Form Field	Options(Show)	SharePoint Column
my:OrderNumber		Order Number
my:Employee		Employee
my:OrderDate		Order Date
my:ReqDate		Required Date
my:ShipDate		Shipped Date
my:ShipVia		Ship Via
my:Freight		Freight

Add field mapping

We have only filled out the Form Fields that can be selected using the field picker task pane at this point. This is a repeating group, so it needs to be indicated, and selected to the group in the **Repeating Group** field.

We need to manually enter the XPath for the remaining mappings – in order for it to be *relative* to the repeating group itself.

The repeating group is: *my:myFields/my:Customer/my:Orders/my:Order*

The other information we want is in the Customer group - we need to go up the tree twice: *../..*

That takes us from my:Order back up to my:Orders (../) and then again back up to my:Customer (../). When in doubt, let InfoPath figure it out - if we need a relative XPath and we are not sure what it would be, we can leverage InfoPath - here's a quick [guide](#).

Here we can see that the Customer address information will be the Shipping Address information. Here is how the mapping should look like after manually typing the relative paths to the Customer info:

../..my:ContName	...	Customer	▼
../..my:CompName	...	Ship Name	▼
../..my:Address	...	Ship Address	▼
../..my:City	...	Ship City	▼
../..my:Region	...	Ship Region	▼
../..my:PostalCode	...	Ship Postal Code	▼

 Add field mapping

Another new feature for the latest Mapping Tool is the “Parent List ID” option which we will incorporate into the child mapping.

At the SharePoint Child List, create a look up column:

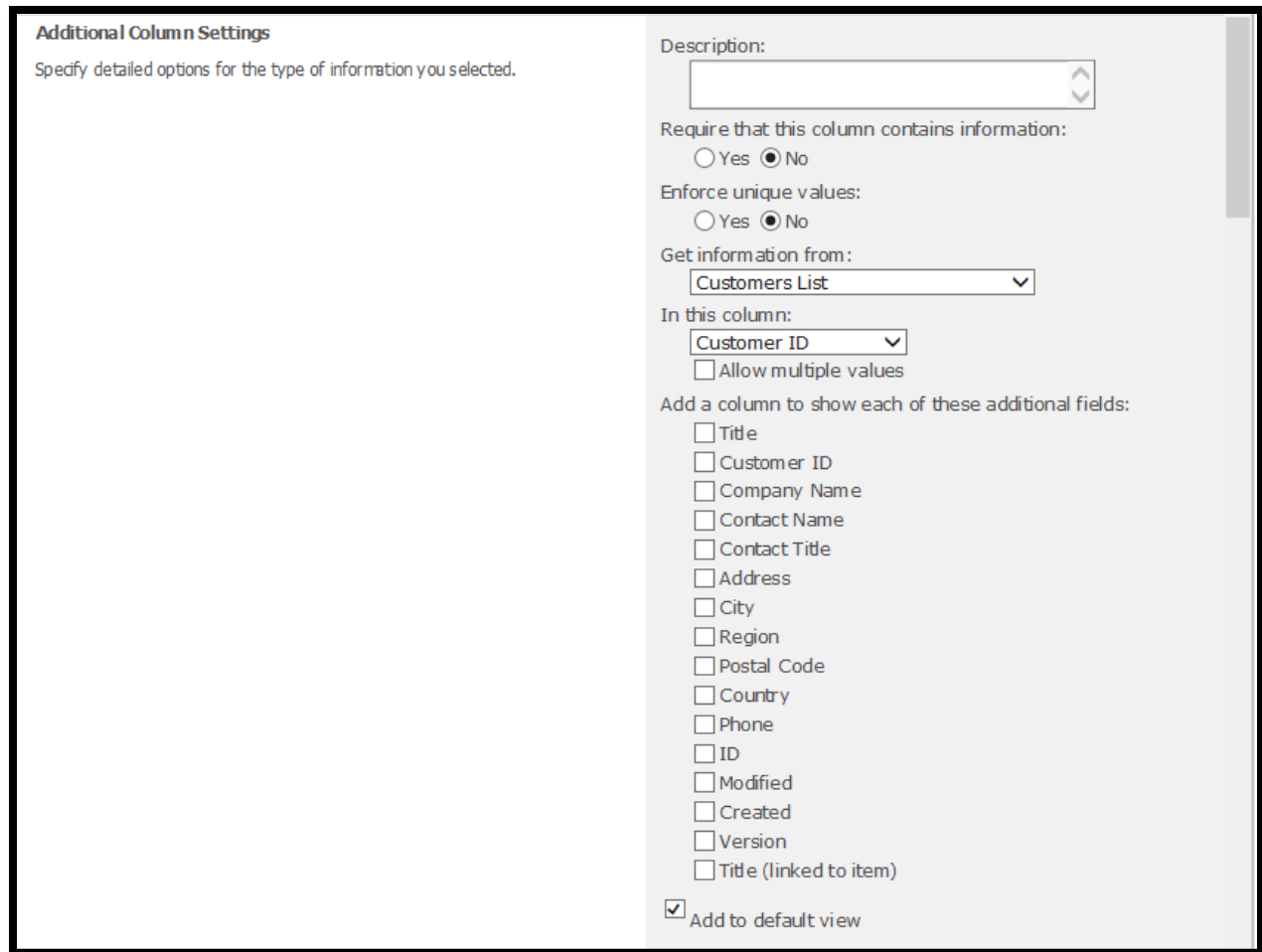
Column name:

The type of information in this column is:

☐ Single line of text
☐ Multiple lines of text
☐ Choice (menu to choose from)
☐ Number (1, 1.0, 100)
☐ Currency (\$, ¥, €)
☐ Date and Time
☒ Lookup (information already on this site)
☐ Yes/No (check box)
☐ Person or Group
☐ Hyperlink or Picture
☐ Calculated (calculation based on other columns)
☐ External Data
☐ Managed Metadata

At the Additional Column Settings, select the **Parent List** from the *Get information from* drop-down control.

Below the control, select a column from the *Parent List* which will represent the Look up column, in this case the *Customers List's* Customer **ID** column.



After creating the look up column, let us head back to the Mapping Tool and click the **Refresh** button for the child mapping. This should refresh the SharePoint column list.



Underneath the Ship Postal Code of the child mapping, insert another row which we will set to the value of the lookup column.

Form Field	Options (Show)	SharePoint Column
my:OrderNumber	...	Order Number
my:Employee	...	Employee
my:OrderDate	...	Order Date
my:ReqDate	...	Required Date
my:ShipDate	...	Shipped Date
my:ShipVia	...	Ship Via
my:Freight	...	Freight
.././my:ContName	...	Customer
.././my:CompName	...	Ship Name
.././my:Address	...	Ship Address
.././my:City	...	Ship City
.././my:Region	...	Ship Region
.././my:PostalCode	...	Ship Postal Code
	...	ParentId

[Add field mapping](#)

Click “Show” and tick the checkbox of the **Parent List Id** so that the mapping will determine that this is the look up column in SharePoint. This will serve as a reference upward to the parent list, which will be important for maintaining relationships between lists – hide the optional section afterwards.

.././my:PostalCode	☐ Parent List Id ☐ Id ☐ Rich Text ☐ Attachment ☐ Parent List Id	Ship Postal Code
	☐ Id ☐ Rich Text ☐ Attachment ☒ Parent List Id	ParentId

[Add field mapping](#)

Next, insert another SharePoint mapping for the template - this time, we'll be mapping to the Order Details list in the same hierarchical manner as to how the schema displays the fields.

Mapping Definition
[Show Help](#)

Mappings

- Customers List + -
- Orders List + -
- Order Details List + -

[Add Top-Level Mapping](#)

Order Details List

SharePoint List	Order Details List	Refresh	☐ Use List Name for Submit
-----------------	--------------------	-------------------------	---

Proceed with mapping the **Order Details** fields to the appropriate columns in SharePoint.

Order Details List

SharePoint List
Order Details List
Refresh
☐ Use List Name for Submit

Mapping Details
[Show Help](#)

☒ Data is from repeating group

Base Path
my:OrderDetails/my:OrderDetail

Form Field	Options(Show)	SharePoint Column
my:Product		Product
my:UnitPrice		Unit Price
my:Quant		Quantity
../my:OrderNumber		Order Number
	☒ Parent List Id	ParentId

☒ Add field mapping

Note that this is a repeating group, so it needs to be indicated, and selected to the group in the **Repeating Group** field.

This time, we're including a relative path to the OrderNumber field - while the list has an Order Number column, it is not a lookup. Thus, we want to submit the actual OrderNumber value, not the ID of the Order list item.

We can also add another look up column using the same column from the Parent. Since this is below the Orders List, the ParentId will refer to the items added to the Orders List, in this case a look up column to Employee of the Orders List in SharePoint.

Setting Conditions

When submitting with forms with multiple layers of nesting or Parent-Child relationships, we often encounter blank entry issues caused by blank rows pre-populated in the form – not having actual data in it.

A solution to prevent blank entries in SharePoint is by manually adding a condition to the Base Path for each sub-level or child mapping:

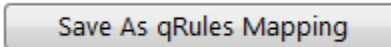
Base Path

my:myFields/my:Customer/my:Orders/my:Order[my:OrderNumber != '']

This condition will prevent the mapping from submitting data to the Order List if the Order Number field is blank.

Save the Mapping

Finally, click the **Save as qRules Mapping** button:

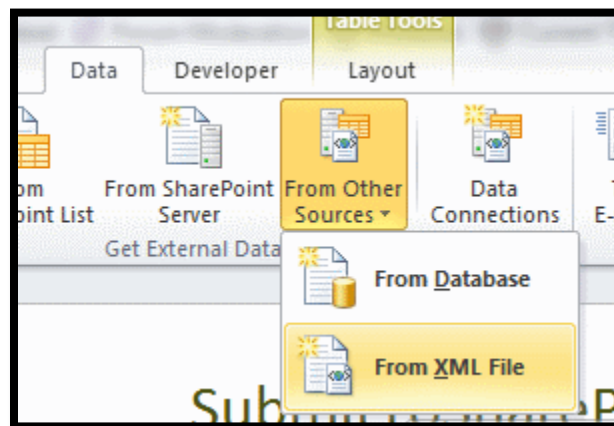


The saved mapping will default the name to mapping.xml – and we can just leave that as is. For reference, you may download a mapping copy [here](#) - keep in mind that even if you are using the same SharePoint template, your URL and list GUIDs will be different from those in the sample file.

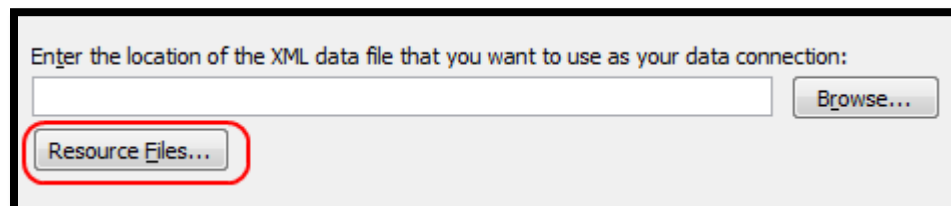
The Data Connections

Now we will add our mapping file, mapping data connection, and submit data connection to our form. There are various ways to add an XML resource file – but we can manually add the resource file instead of letting InfoPath do it, as InfoPath strips off the xml file extension when it helps you use the Browse option to select the file. If we add the resource file ourselves, it makes it simpler when we decide to change the mapping.

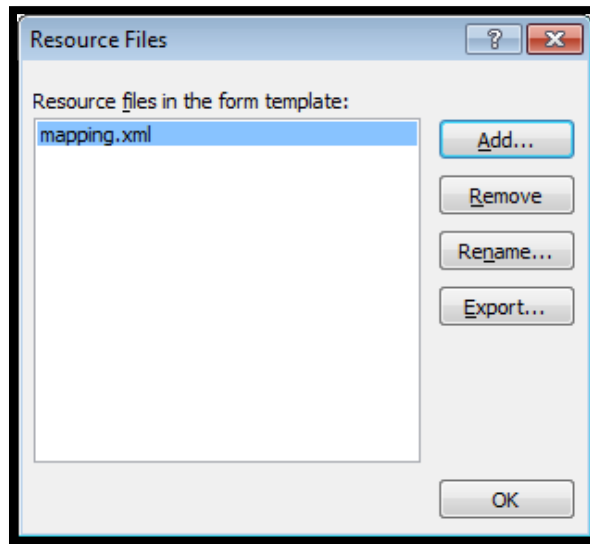
From InfoPath, select the **Data** tab on the Ribbon, then **From Other Sources** --> **From XML File**:



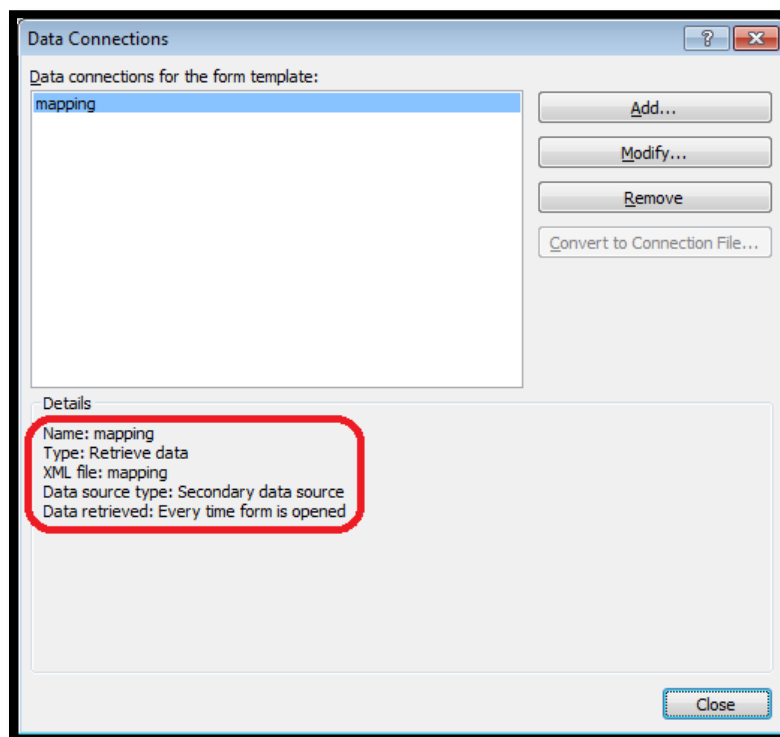
Click the **Resource Files** button:



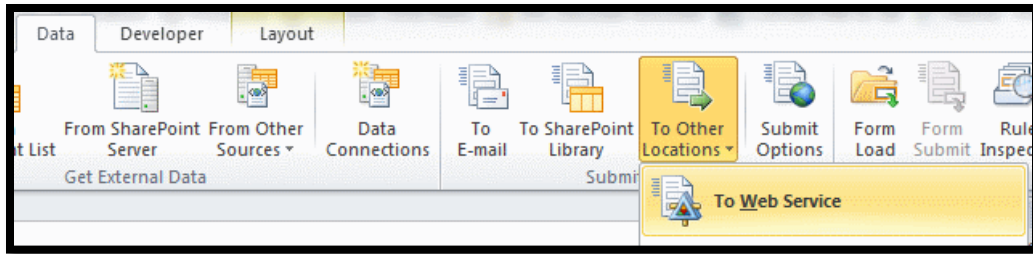
Click the **Add** button in the Resource Files dialog, and select the mapping.xml saved earlier:



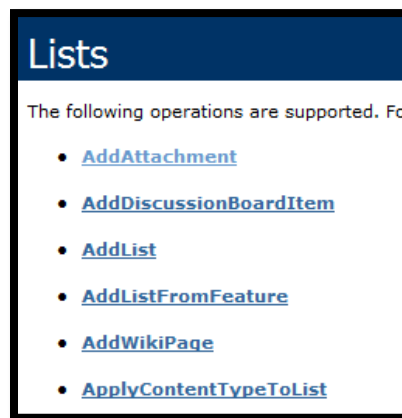
Click **OK**, then **Next** and **Finish** - the default name for your mapping data connection will be **mapping** - just leave that as is, and leave the checkbox to load data on open selected. In the data connections dialog, the details for your data connection should look like this:



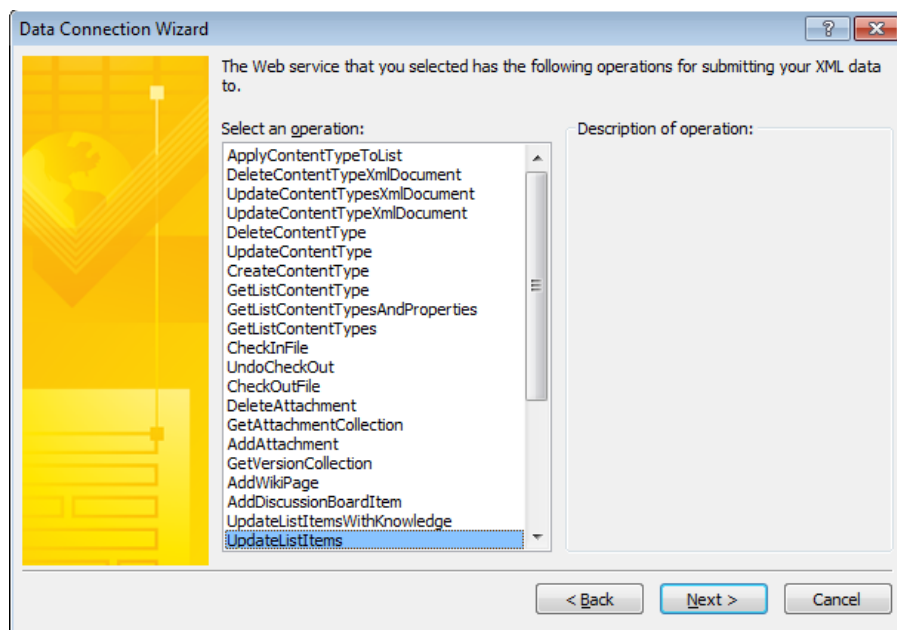
Next, we need to add a Submit data connection to the UpdateListItems method of the SharePoint Lists web service. On the **Data** tab of the Ribbon, select **To Other Locations** --> **To Web Service**:



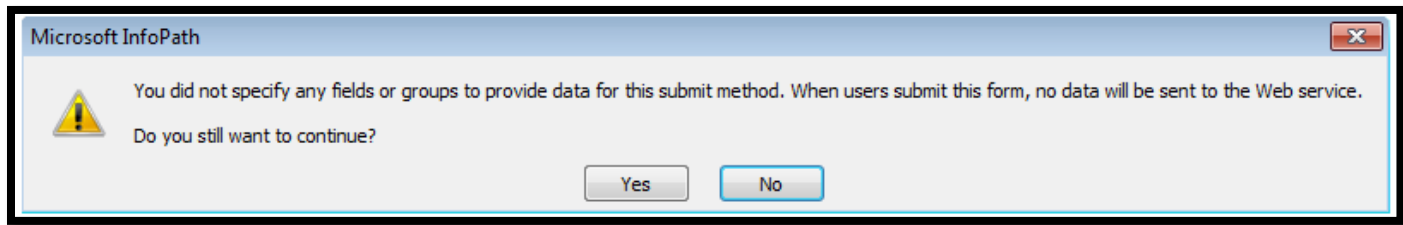
The URL for the Lists web service is: http://YourServer/_vti_bin/lists.asmx - replace "YourServer" with your actual server name. If in doubt, test your URL in a browser, making sure you get a page that looks like this:



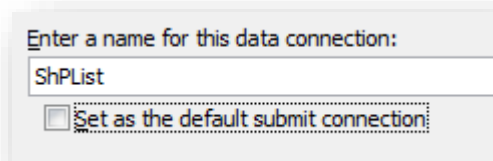
After entering the URL in the Data Connection Wizard, click **Next** and select the UpdateListItems method:



Click **Next** again and leave the parameters blank in the following dialog - just click **Next** and **Yes** on the warning:



qRules will populate the data for the submit parameters. In the final window, it is ideal to deselect the checkbox for using the connection as the submit default, and give our connection a simpler name:

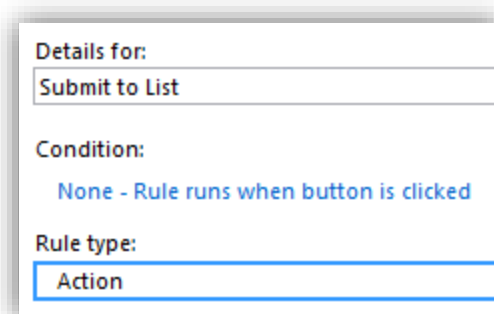


Make a note of the name you use - including the casing - as you will need it for your command.

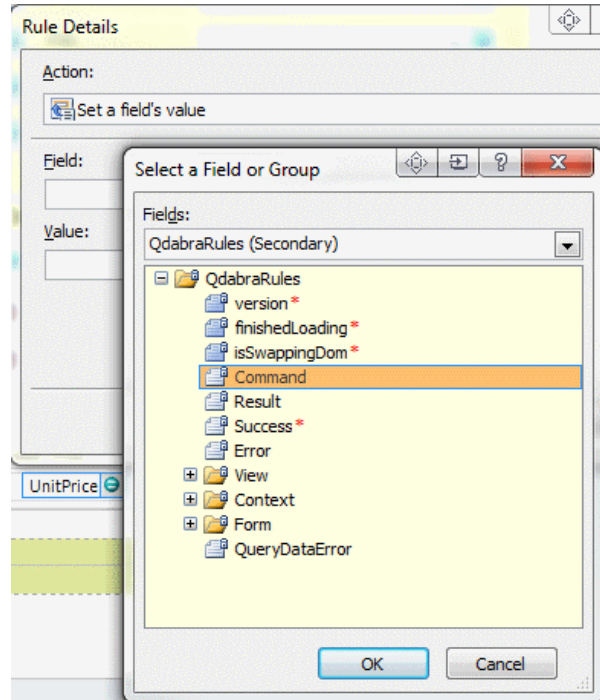
Add the qRules SubmitToSharePointList Command

We are now at a point where we can add our command. First, you'll need to inject your template with qRules. Save and close your form template, and from the **Start** menu, navigate to Qdabra / Tools and find the qRules folder. -Open the injector and inject your form. If you need more detail, qRules installs with a *Quick Start* guide that you'll find helpful.

After injecting, open your form in design mode, add a button to your form, and add an action rule to the button:



The Action should set the *Command* node in the *QdabraRules* data source:



The value we will set the **Command** node to is:

SubmitToSharePointList /submit=ShPList

Action:	 Set a field's value
Field:	<u>Command</u>
Value:	SubmitToSharePointList /submit=ShPList

Here we are using the name of the re-named UpdateListItems data connection. The value to the right of the "=" sign needs to be the exact name of your SharePoint Lists web service data connection - including the casing. So, if yours is named **MyFancySharePointSubmit** your command would be:

SubmitToSharePointList /submit=MyFancySharePointSubmit

Let's also assume that you followed along and left your mapping data connection named **mapping** - if it is named anything else, you'll need to include the /mapping parameter.

Example:

*SubmitToSharePointList /submit=MyFancySharePointSubmit
/mapping=MyMappingThatIRenamedToBeDifficultAboutThings*

Test Your Form!

First, from the QdabraRules data source, drag the **Error** node onto your form, underneath the submit button:



You can also drag the Success node onto the form for checking. Then, preview the form. Fill out some data:

qRules 6.1 - SubmitToSharePointList

Customer

Customer Id: ABPAI	Company Name: ABC Paint
Contact Name: John Smith	Contact Title: CEO
Address: 123 Main St.	City: Anywhere
Region: Northeast	Postal Code: 123456
Country: USA	Phone: 123-456-7890

Order

Order Number: 998877	Employee: Jane Doe
Order Date: 6/2/2014	Required Date: 6/24/2014
Ship Date: 6/9/2014	Ship Via: Speedy Express
Freight: \$123.45	

Product	Unit Price	Quantity	Total
Air Paint Sprayer	\$35.00	6	\$210.00
Oil Based Paint	\$27.98	10	\$279.80

☒ Insert item

☒ Insert Order

Error:
 Success: ☐

Click the Submit button.

The Error node will be blank and the Success box will be checked as true when the data has been submitted properly.

Here's how the list should look –

Customers:

Customer ID	Company Name	Contact Name	Contact Title	Address	City	Region	Postal Code	Country	Phone
ABPAI	ABC Paint	John Smith	CEO	123Main St.	Anywhere	Northeast	123456	USA	123-456-7890

Orders:

Order Number	Employee	Order Date	Required Date	Shipped Date	Ship Via	Freight	Customer	Ship Name	Ship Address	Ship City	Ship Region	Ship Postal Code
998877	Jane Doe	6/2/2014	6/24/2014	6/9/2014	Speedy Express	\$123.45	3	ABC Paint	123Main St.	Anywhere	Northeast	123456

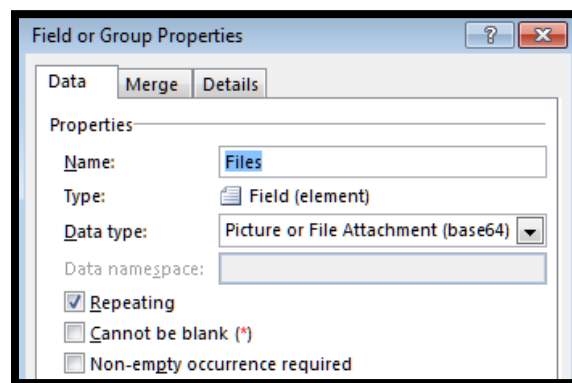
Order Details:

Product	Unit Price	Quantity	Order Number
Air Paint Sprayer	\$35.00	6	998877
Oil Based Paint	\$27.98	10	998877

Working with Attachments

Form Changes

First, we will need to add a field to our form data source for attachments. Let's add it to Customer – so we can attach a filled out credit application, or some other file related to the customer:

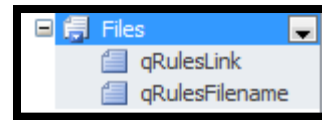


Let's set the field to be repeating, so the user will be able to add multiple files.

There are two options when adding attachments to the list item - we can keep the file in the form XML and **add** it to the list item in SharePoint **or** we can upload the file to the list item and **remove** it from the

XML. Since we don't want to keep the file in two places, let's take the second option – this will enable qRules to remove the file from the form after saving it to the list item.

To do that, all we have to do is add two attributes to the attachment node - qRulesLink and qRulesFilename:

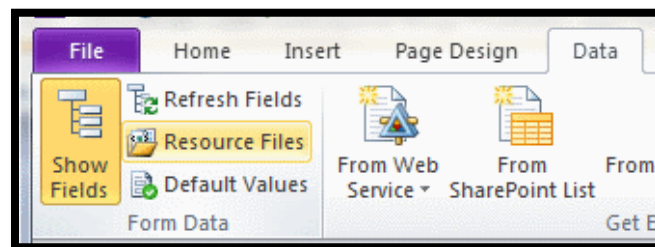


The presence of these attributes on the attachment node indicates to qRules that you want to have the file removed after it has been uploaded (like the functionality of the SaveToSharePoint command, only for a list instead of a library).

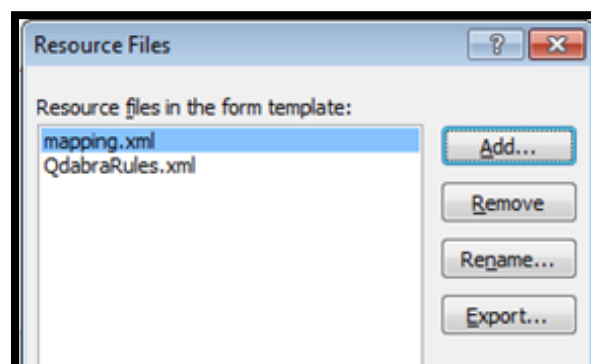
Mapping Changes

Next, we need to modify our list mapping file to indicate that we have an attachment field. Now, if you have worked with the SubmitToSharePointList command before, you may have had to re-do your mapping each time you wanted to make a change. We've improved that experience in qRules 5.1 or higher- we can now import our mapping into the mapping form.

So, first save your changes to your form template. Then, from the *Data* tab of the Ribbon, select *Resource Files*:

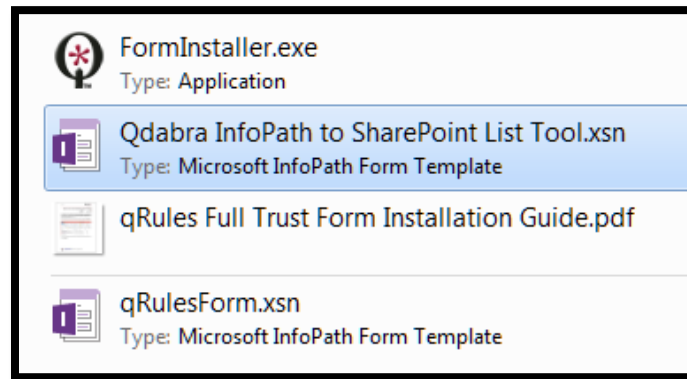


From the *Resource Files* dialog, click on your mapping.xml file:

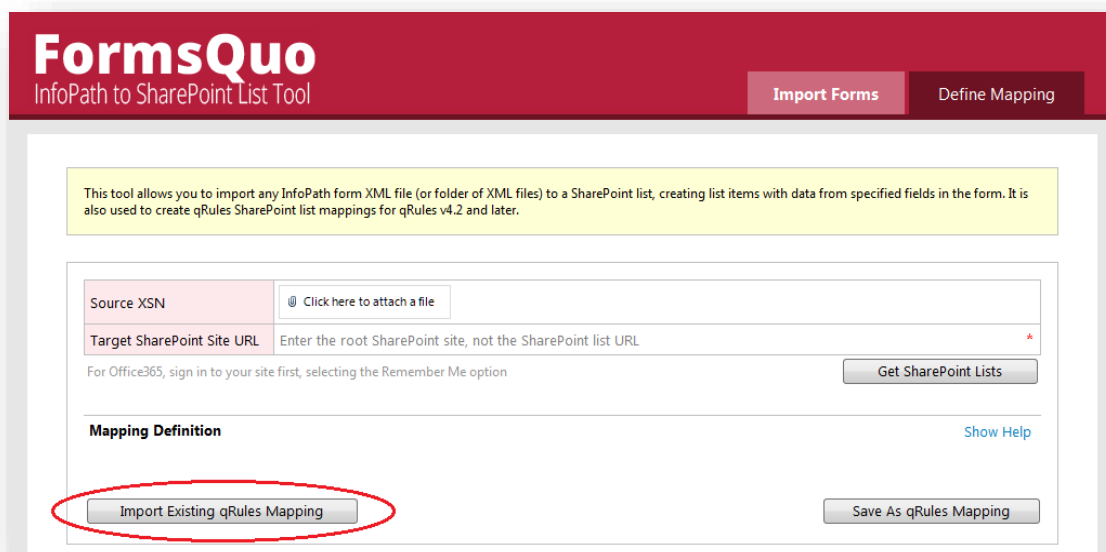


Click the *Export* button to save the file.

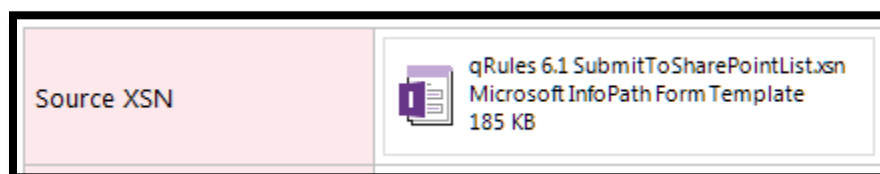
Once saved, open the mapping tool:



Select the *Define Mapping* tab. There is a button toward the bottom - *Import Existing qRules Mapping* - click that:



You'll get a warning about the form deleting existing mappings, which you can click *OK* on, and then select the mapping file you just exported from your form. Your existing mapping will be imported - all that remains is to attach your template to the *Source XSN* field - much better than remapping every time.



In the mapping to the *Customers* list, select *Add mapping*:

Customers List

SharePoint List: Customers List Refresh ☐ Use List Name for Submit

Mapping Details Show Help

☐ Data is from repeating group

Base Path (optional)
Select the base path for your data ...

Form Field	Options (Show)	SharePoint Column
my:myFields/my:Customer/my:CustId	...	Customer ID
my:myFields/my:Customer/my:CompName	...	Company Name
my:myFields/my:Customer/my:ContName	...	Contact Name
my:myFields/my:Customer/my:ContTitle	...	Contact Title
my:myFields/my:Customer/my:Address	...	Address
my:myFields/my:Customer/my:City	...	City
my:myFields/my:Customer/my:Region	...	Region
my:myFields/my:Customer/my:PostalCode	...	Postal Code
my:myFields/my:Customer/my:Country	...	Country
my:myFields/my:Customer/my:Phone	...	Phone

☒ Add field mapping

Select the new Files node that was added earlier for the *Form Field* and click the *Show* button in the *Options* column:

Options (Show)

Select the *Attachment* checkbox for the Files node mapping:

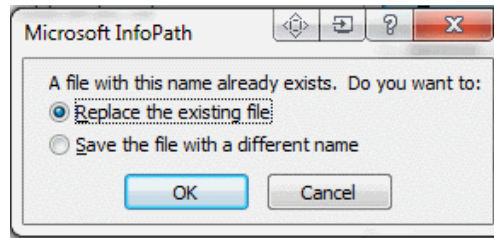
my:myFields/my:Customer/my:Files ...

☐ Id
☐ Rich Text
☒ Attachment

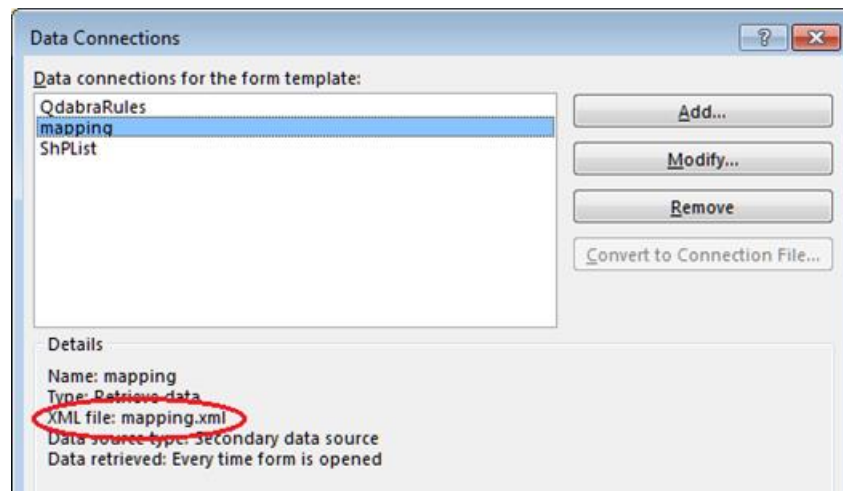
The attachment field is not mapped to a SharePoint column - it is used by qRules to add attachments to your list items if you include the /dsname parameter. You must indicate an ID field to also add attachments.

☒ Add field mapping

And click the *Save as qRules Mapping* button at the bottom of the form to save the mapping.xml. If you maintain the same name as the original file you exported, you can re-add the file as a resource file - no need to walk through the data connection wizard again. From the *Data* tab of the Ribbon, select *Resource Files* again to open the *Resource Files* dialog. Click *Add* and navigate to the mapping file you just saved. InfoPath will prompt you:

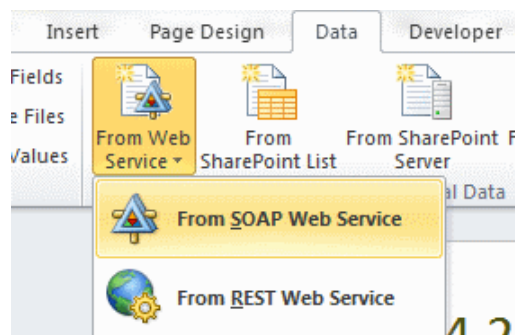


You want to replace the existing file, so leave the default selection and click *OK*. You shouldn't have to modify your mapping data connection - the schema of the XML file will not have changed at all, simply the data. As an aside, you can always verify what file is used by your data connection in the *Data Connection* dialog:

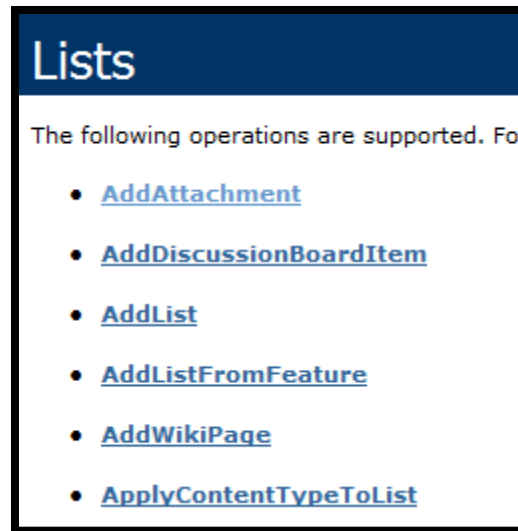


Add a Data Connection

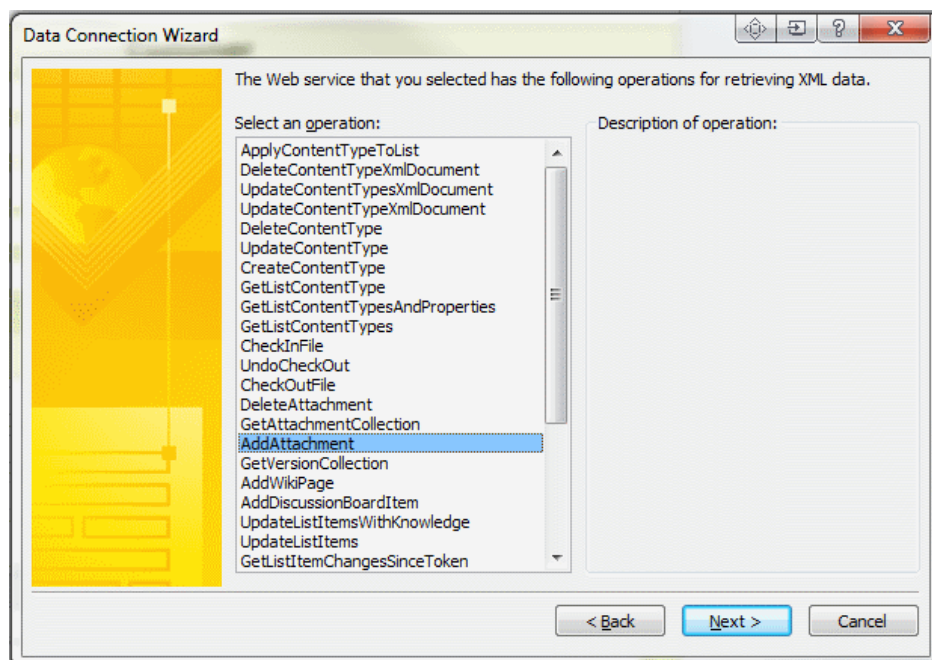
We need to add a Receive data connection to the AddAttachment method of the SharePoint Lists web service. On the *Data* tab of the Ribbon, select **From Web Service** --> **From SOAP Web Service**:



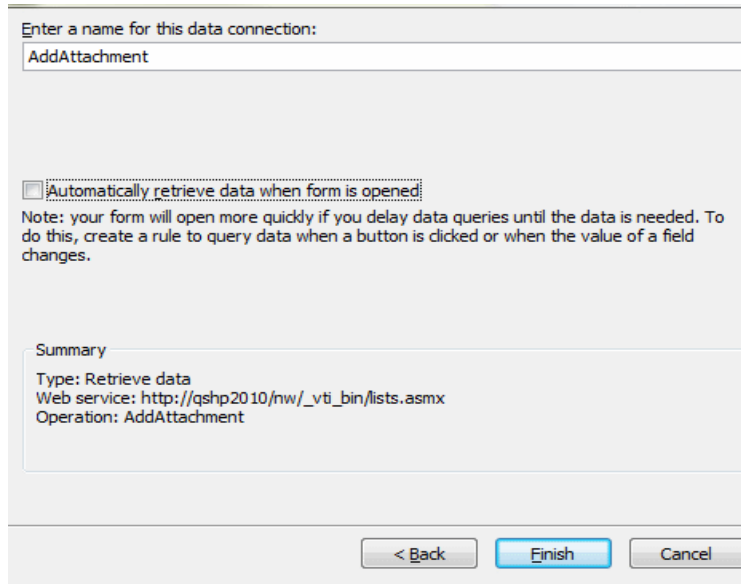
The URL for the Lists web service is: http://YourServer/_vti_bin/lists.aspx - replace "YourServer" with your actual server name. If in doubt, test your URL in a browser, making sure you get a page that looks like this:



After entering the URL in the Data Connection Wizard, click **Next** and select the AddAttachment method:



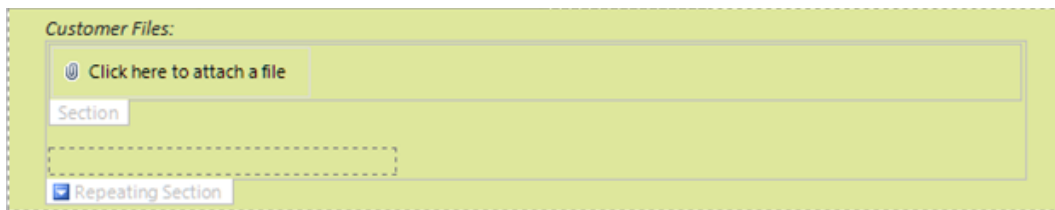
Click **Next** - no need to set any of the query parameters, so click **Next** again, then one more time - in the final screen of the dialog, deselect the "Automatically retrieve data when form is opened" checkbox and leave the default name for the data connection - you can name it something else if you like, but keep track of the name, because we will need it when we modify our command.



Add Controls to the Form

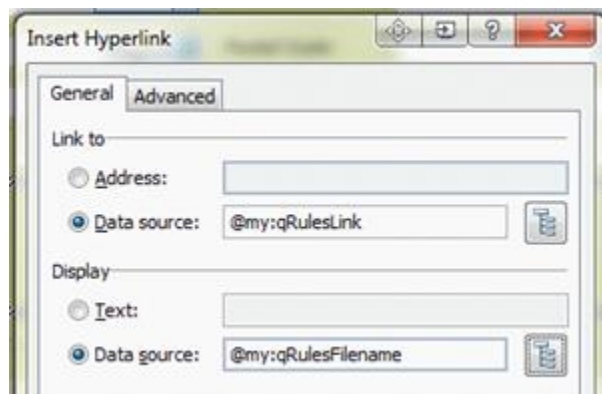
Add controls to your form to allow the selection of files. To do that, let's add a file picker and a hyperlink, with the file picker inside a section – note that the repeating section is bound to the Files field, and the section holding the file picker is bound to the qRulesLink attribute.

Here we have the repeating section and File Attachment control bound to the File field and the section bound to the qRulesLink attribute.



Set the section to hide if the qRulesLink attribute is not blank.

For the hyperlink, set qRulesLink attribute as the link, and the qRulesFilename attribute as the display:



Note that the hyperlink used here is the hyperlink from the Insert tab (**Insert → Hyperlink**), and not the hyperlink custom control under the Home tab.

Modify the Command

In the first walkthrough, we added a submit button to our form with the qRules command:
`SubmitToSharePointList /submit=ShPList`

Now, we need to modify this command to include our attachment files when we submit to the Customer list:
`SubmitToSharePointList /submit=ShPList /dsname=AddAttachment`

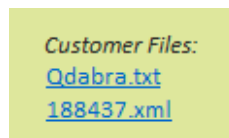
If you named your AddAttachment data connection something different, you need to modify the command above to use the name of *your* data connection:

`SubmitToSharePointList /submit=ShPList /dsname=MySuperSpecialAttachmentDataConnection`

Remember, case **cOunTs** - you need to use the exact same spelling and casing.

Test Your Form!

As always, be sure you have the QdabraRules Error node somewhere you can see it prior to testing. Preview your form, add a new customer, and add some files for the customer. Click the Submit button:



If everything is set up correctly, you'll now have links in your form instead of files – and in your Customer list item, the attached items will be displayed:



Add Delete Logic

This process will require an ID which will be mapped at the Mapping Tool in order to identify which entry gets updated at the SharePoint List.

Open the mapping tool, and select the Define Mapping tab. Attach your template to the *Source XSN* field and Click the Import Existing qRules Mapping button at the bottom to import the previous mapping file we've used.

FormsQuo
 InfoPath to SharePoint List Tool
 Import Forms
Define Mapping

This tool allows you to import any InfoPath form XML file (or folder of XML files) to a SharePoint list, creating list items with data from specified fields in the form. It is also used to create qRules SharePoint list mappings for qRules v4.2 and later.

Source XSN Click here to attach a file

Target SharePoint Site URL Enter the root SharePoint site, not the SharePoint list URL *

For Office365, sign in to your site first, selecting the Remember Me option

Get SharePoint Lists

Mapping Definition Show Help

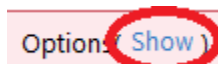
Import Existing qRules Mapping
Save As qRules Mapping

Here you will need to declare the ID field mapping by first inserting a new row to the mapping.

my:myFields/my:Customer/my:Region	...		Region
my:myFields/my:Customer/my:PostalCode	...		Postal Code
my:myFields/my:Customer/my:Country	...		Country
my:myFields/my:Customer/my:Phone	...		Phone
my:myFields/my:Customer/my:File	...	☒ Attachment	The attachment field is not mapped to a SharePoint column - it is used by qRules to add attachments to your list items if you include the /dsname parameter. You must indicate an ID field to also add attachments.
my:myFields/my:Customer/my:ID	...	☒ Id	The Id field is not mapped to a SharePoint column - it is used by qRules for populating the form XML with the SharePoint List Item Id upon successful submission

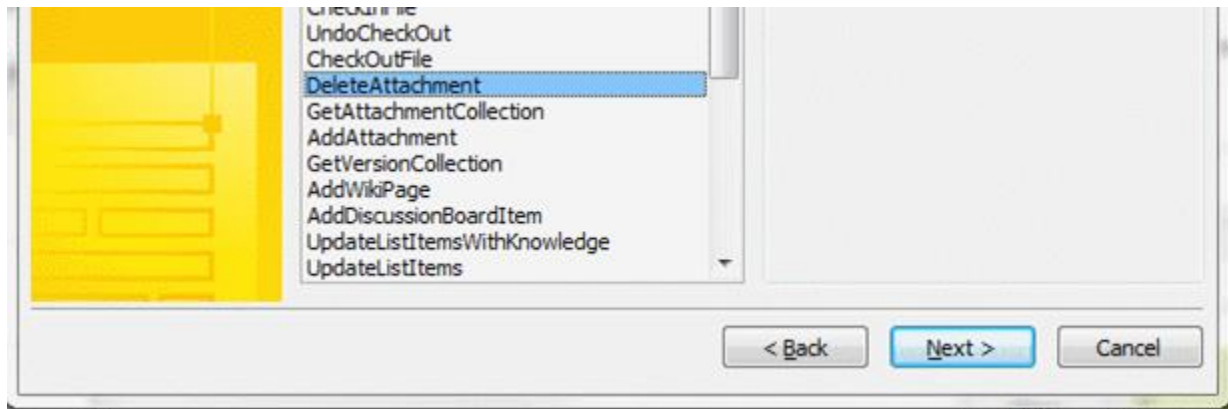
☒ Add field mapping

Select the **ID** from the schema for the Form field and declare this as an *Id* at the Options section by clicking the Show button and ticking the *Id* check box.



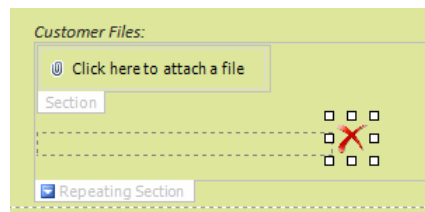
Once done, save the mapping.xml and update the original file. You can re-add the file as a resource file - no need to walk through the data connection wizard again. From the *Data* tab of the Ribbon, select *Resource Files* again to open the *Resource Files* dialog. Click *Add* and navigate to the mapping file you just saved. InfoPath will prompt you replace the existing file, so leave the default selection and click *OK*.

At the form, you will need to add another data connection to the DeleteAttachment method of the SharePoint Lists web service - exactly the same way as you added the AddAttachment method above:

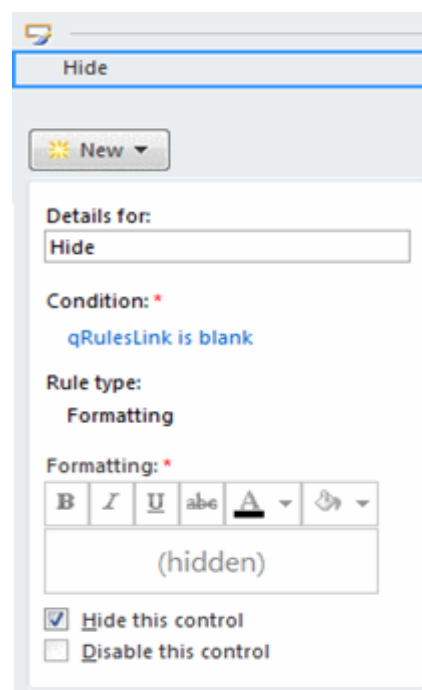


Don't bother filling out any of the query parameters, and deselect the option to automatically execute the data connection on load (just like we did for the AddAttachment data connection).

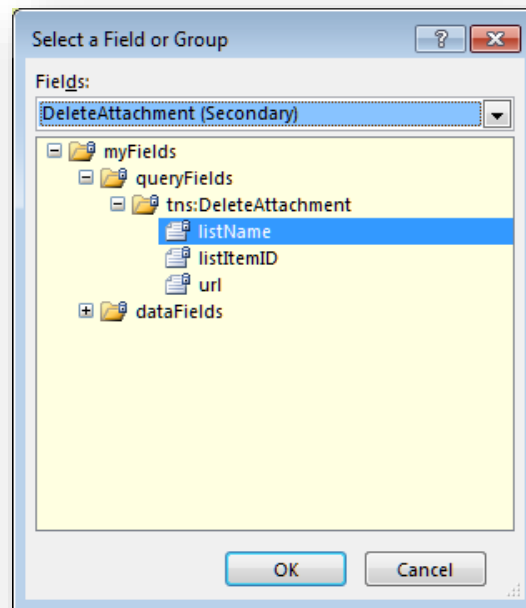
Add a button to your form next to your hyperlink control:



Add a conditional formatting rule to the button to hide it if the qRulesLink attribute is blank:



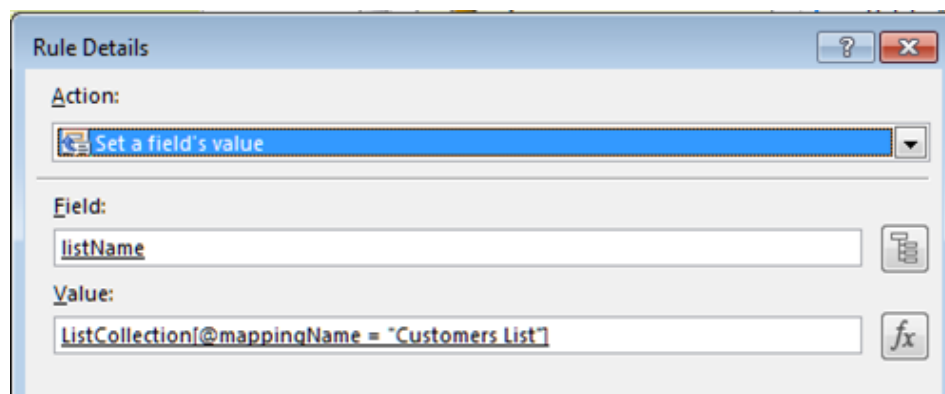
Add an action rule to the button. There are three fields we need to set in the DeleteAttachment data source queryFields:



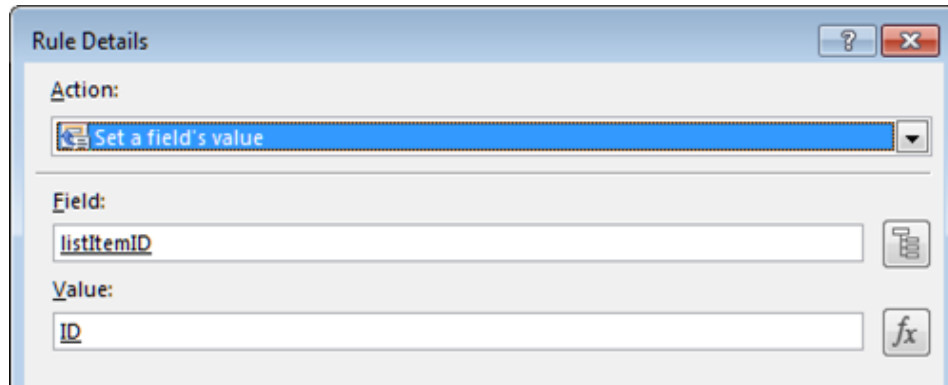
Here is a way to get the listName from the mapping file. MSDN [states](#) that the listName parameter can be the list GUID or name. We have the GUID in our mapping file - it is in the Customers mapping:

```
<Mappings mappingName="Customers">
  <ListCollection useName="false" ListName="Customers">{6E986BE0-F112-4F61-8931-C1891E239E74}</ListCollection>
</Mappings>
```

So, let's set that field to the ListCollection node on the mapping file where the mappingName is equal to Customers List:

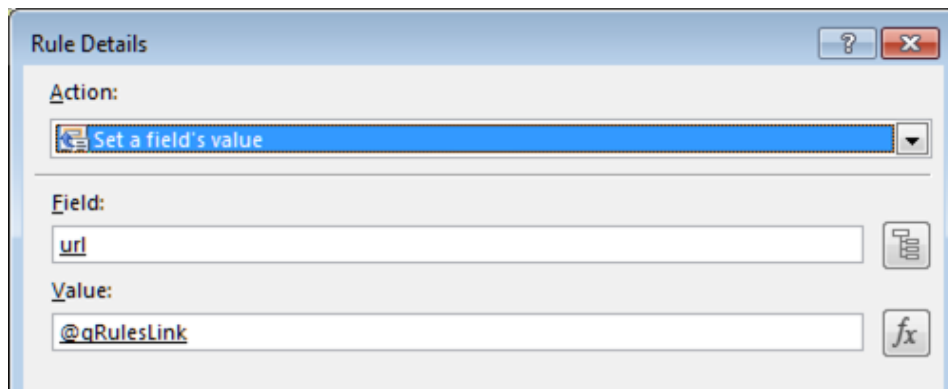


Next, let's set the listItemID parameter to the field are storing our SharePoint item ID in - in this case, that would be /my:myFields/my:Customer/my: ID:



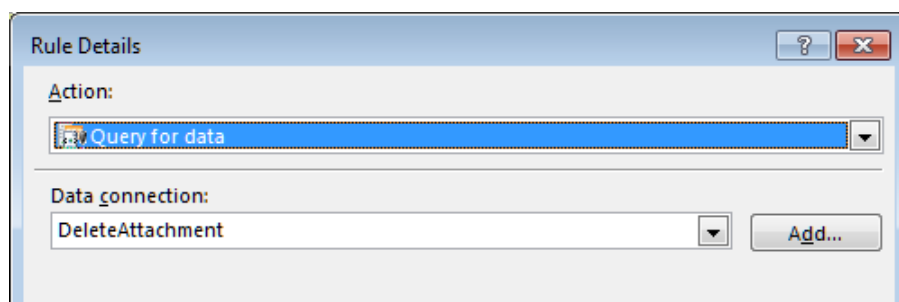
The 'Rule Details' dialog box shows the 'Action' dropdown set to 'Set a field's value'. The 'Field' text box contains 'listItemID' and the 'Value' text box contains 'ID'. There are icons for field selection and formula entry next to these boxes.

Let's set the url parameter to the qRulesLink attribute:



The 'Rule Details' dialog box shows the 'Action' dropdown set to 'Set a field's value'. The 'Field' text box contains 'url' and the 'Value' text box contains '@qRulesLink'. There are icons for field selection and formula entry next to these boxes.

We now need to execute the query:



The 'Rule Details' dialog box shows the 'Action' dropdown set to 'Query for data'. The 'Data connection' dropdown is set to 'DeleteAttachment' and there is an 'Add...' button next to it.

Unfortunately, the web service does not return a friendly error message if the action fails. Typically, however, if the query does not succeed, any rule actions after it will not run, so it is usually safe to assume that if rules keep running, the deletion has succeeded and we can clear the qRulesLink and qRulesFilename attributes.

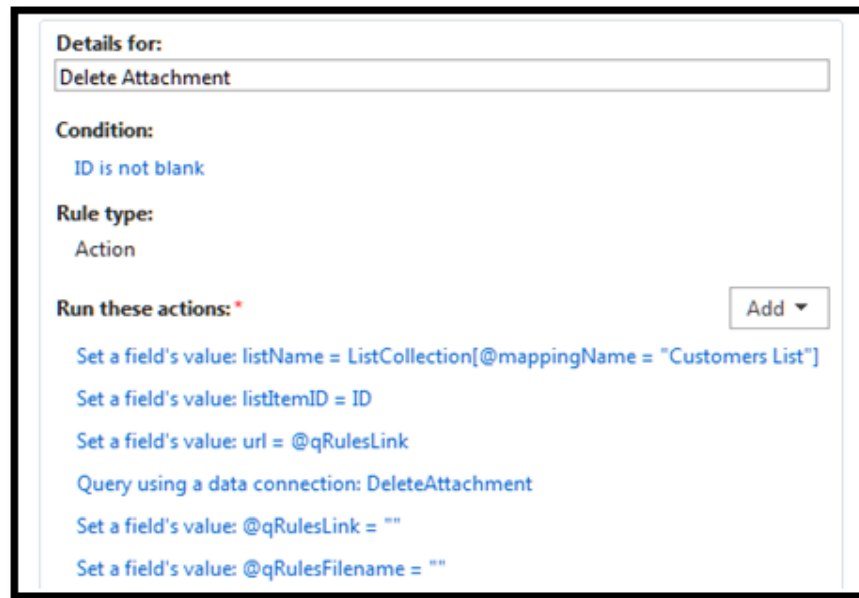
Two other options for error handling:

1. Use the qRules QueryData command to execute the DeleteAttachment query - it will handle any errors and return them to the QdabraRules *QueryDataError* node - you could then add some conditions around clearing the *qRulesLink* and *qRulesFilename* attributes
2. Or, after executing the data connection, you could query the GetAttachmentCollection method of the Lists web service - check the value of:

/dfs:myFields/dfs:dataFields/tns:GetAttachmentCollectionResponse/tns:GetAttachmentCollectionResult

And see if it contains the value of the *qRulesLink* attribute.

The final rule set may appear like this:



The screenshot shows the 'Details for: Delete Attachment' configuration window. It includes a 'Condition' section with 'ID is not blank', a 'Rule type' section with 'Action', and a 'Run these actions:' section with an 'Add' button. The actions listed are:

- Set a field's value: listName = ListCollection[@mappingName = "Customers List"]
- Set a field's value: listItemID = ID
- Set a field's value: url = @qRulesLink
- Query using a data connection: DeleteAttachment
- Set a field's value: @qRulesLink = ""
- Set a field's value: @qRulesFilename = ""

Here we have chosen to let InfoPath produce a query error. We've also added a condition to only run the rule if the SharePoint ID field has a value.

Now our form submits to three related SharePoint lists. It can include attachments, and the user can delete attachments in case needed.

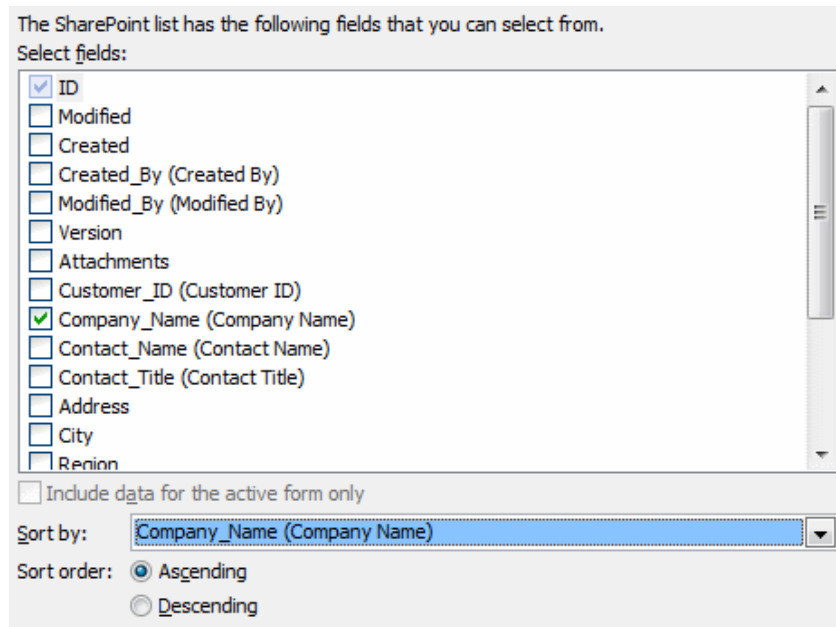
Working with Existing Data & Submitting a Single Mapping

Let us re-use the sample form and assume you have existing data in a SharePoint list, and you wish to query it, update it and resubmit it back into the same list. This section explains how qRules can help.

Data Connections

First, we need to add 2 data connections to our form - one that we'll use for a customer drop down, and the other that we'll use to return all the data for a given customer. The first one, which we'll call the GetCompanyName data connection, is to the Customer list, and returns these columns:

Let's sort by Company Name:



The SharePoint list has the following fields that you can select from.

Select fields:

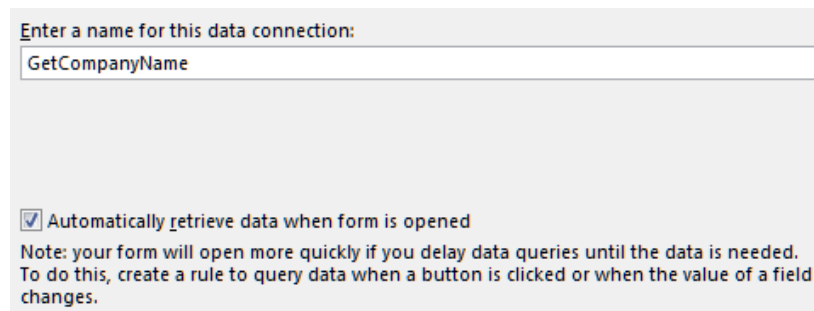
- ☒ ID
- ☐ Modified
- ☐ Created
- ☐ Created_By (Created By)
- ☐ Modified_By (Modified By)
- ☐ Version
- ☐ Attachments
- ☐ Customer_ID (Customer ID)
- ☒ Company_Name (Company Name)
- ☐ Contact_Name (Contact Name)
- ☐ Contact_Title (Contact Title)
- ☐ Address
- ☐ City
- ☐ Region

☐ Include data for the active form only

Sort by: **Company_Name (Company Name)**

Sort order: ☒ Ascending ☐ Descending

Return data on load for this connection only:



Enter a name for this data connection:

GetCompanyName

☒ Automatically retrieve data when form is opened

Note: your form will open more quickly if you delay data queries until the data is needed. To do this, create a rule to query data when a button is clicked or when the value of a field changes.

Now, create another data connection to the Customers list and select at least all of the columns that the form is mapped to submit data to:

The SharePoint list has the following fields that you can select from.

Select fields:

- ☒ Title
- ☒ Customer_ID (Customer ID)
- ☒ Company_Name (Company Name)
- ☒ Contact_Name (Contact Name)
- ☒ Contact_Title (Contact Title)
- ☒ Address
- ☒ City
- ☒ Region
- ☒ Postal_Code (Postal Code)
- ☒ Country
- ☒ Phone
- ☒ ID
- ☐ Modified
- ☐ Created

☐ Include data for the active form only

Sort by:

Sort order: ☒ Ascending ☐ Descending

This time, deselect the checkbox to query data on load by unchecking the checkbox at the last page of the data connection wizard.

Form Changes

In this form, we'll let the user choose an existing customer or just fill in the fields if they want to create a new customer. Add a drop down that is bound to the /my:myFields/my:Customer/my:ID field – the field previously added for deleting attachments.

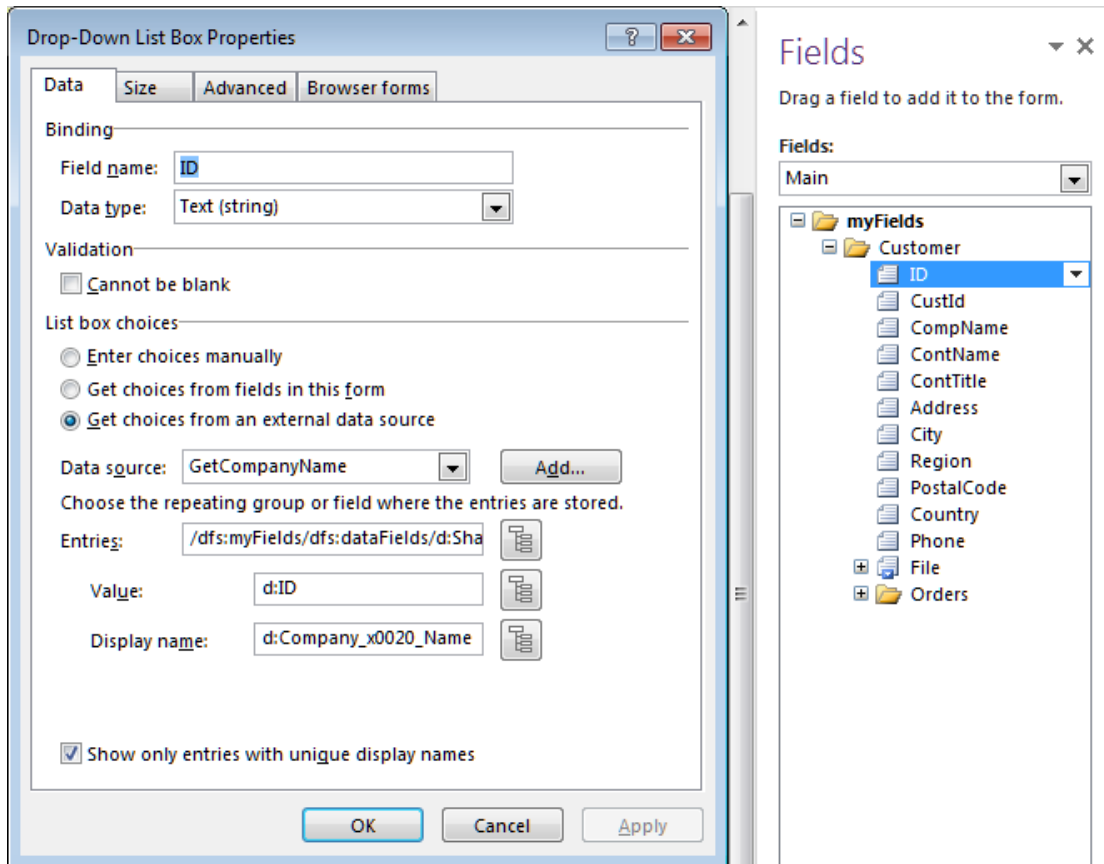
Although it is not required for every mapping, ID field mapping becomes necessary when we decide to update list items in SharePoint. This form field will hold the SharePoint List item id after a successful submit- we are also going to leverage this field to allow our users to open a new form and edit existing customers.

Customer

Select existing customer, or fill in fields below to create new:

▼

Set the drop down to use the GetCompanyName data source, with the id for the value and the Company Name for the display:

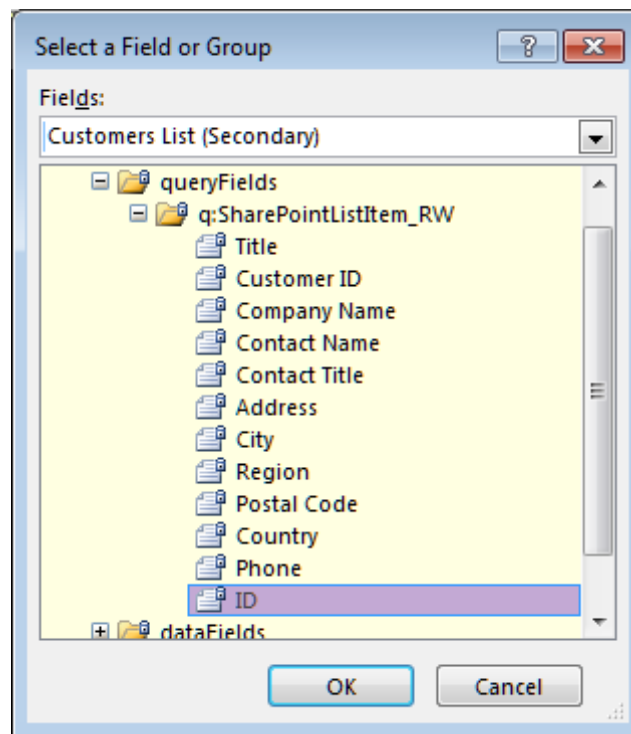


After the user selects the Customer, we'll need to query the Customers List data source for that customer's data and then populate the main data source nodes with those values - this can be done with ordinary InfoPath rules. While this could be done with rules set on the ID field the drop down is bound to, that would make things slightly more complex for us - we'd need to make sure these rules don't run when qRules is populating that field. So, for the purposes of this walkthrough, let us add a button and put these rules on the button. However, you have other options - like:

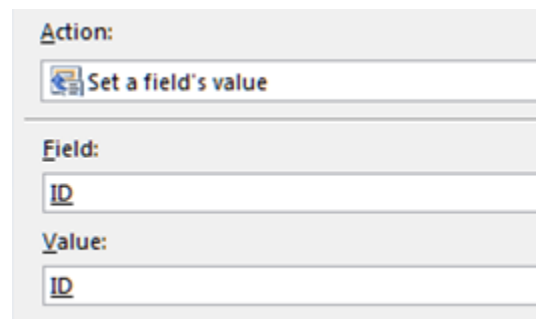
- Bind the drop down to another field, and execute the rules from that field (be sure you also set the form field for the id if you take this approach).
- Have a helper field in a 2ds that set when the form is about to execute the SubmitToSharePointList command, and base conditions off the value of that field to prevent execution of rules when qRules sets the id field for a new item.

For this form – let us proceed using a button.

The first rule action sets the Customers *query* id field:



To the value of the /my:myFields/my:Customer/my:ID field:

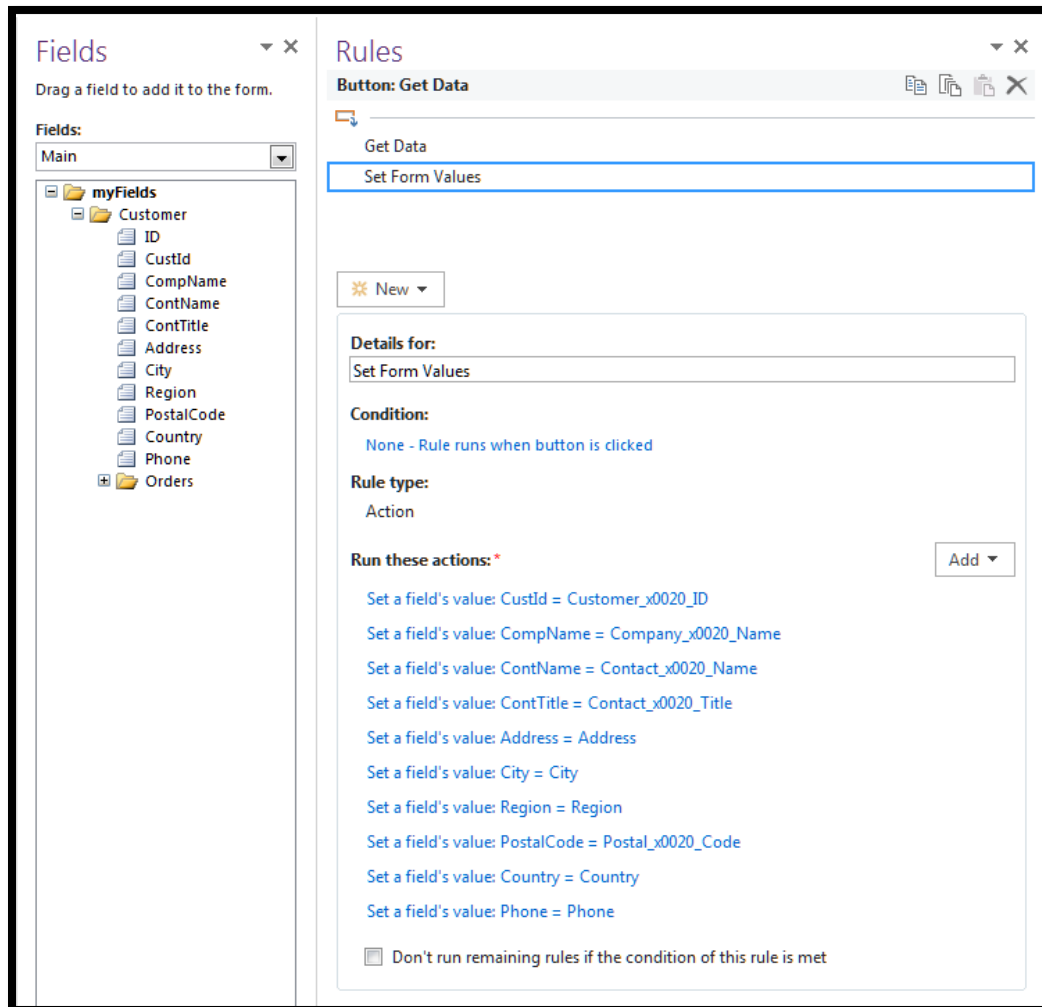


Then, let's execute the Customers query:



Now, since the user selected the id of the customer desired, we could have returned all the customer data and used filtered XPath to get the values we want - and if you are using SharePoint & InfoPath 2007, this would be the approach you would probably take (or use the qRules FilterOwssvr command to

get back just the single record you need). Since we will only have one record on the Customers data source, we don't have to use filtered XPath to populate the other form values.



In the rules above, Let's set the main data source Customer group nodes to values from the Customers list.

Preview

Time to check our work - preview the form, and select a customer from the drop down - does the entire customer's information get populated correctly?

Customer			
Select existing customer, or fill in fields below to create new:			
ABC Paint		Get Data	
Customer Id:	ABPAI	Company Name:	ABC Paint
Contact Name:	John Smith	Contact Title:	CEO
Address:	123 Main St.	City:	Anywhere
Region:	Northeast	Postal Code:	123456
Country:	USA	Phone:	123-456-7890

If not, check and make sure you are returning data to your Customers data source.

Submit a Single Mapping

Now that we have added the logic to pull an existing customer into the form, we can add a button to save just the user's Customer changes. The big trick here is making sure that in the mapping we have identified a field to use for the ID:

```
<Mappings mappingName="Customers List">
  <Properties>
    <Level>0</Level>
    <ParentMapping />
  </Properties>
  <ListCollection useName="false" ListName="Customers List">{8BE5B259-295B-4459-85A5-284F765C84C1}</ListCollection>
  <IsRepeating>false</IsRepeating>
  <RepeatingGroup />
  <Relationships>
    <RelatedList />
    <RelatedListId />
  </Relationships>
  <Mapping>
    <FormField IsRichText="false" IsId="true" IsAttachment="false" IsDate="false" IsParentId="false">my:myFields/my:Customer/my:ID</FormField>
    <SharePointColumn />
  </Mapping>
</Mappings>
```

And making sure we have in our form set that field to the SharePoint List item id that we wish to update - bind a drop down to the field (*/my:myFields/my:Customer/my:ID*) and then using a data connection to a target list for the values - so that the user will be picking a valid item id from the drop down.

Our original submit command was: *SubmitToSharePointList /submit=ShPList*

We then added to this command for adding attachments in the last walkthrough. We are not going to be adding attachments here, just editing the existing customer - so we only need to add the *mappingName* parameter to indicate which mapping we want to submit to:

SubmitToSharePointList /submit=ShPList /mappingName=Customers List

Let us update the data for this customer:

Customer

Select existing customer, or fill in fields below to create new:

▼

Customer Id:

Contact Name:

Address:

Region:

Country:

Company Name:

Contact Title:

City:

Postal Code:

Phone:

After clicking the button with the qRules command on it, we can see the region has been added to the customer:

Customer ID	Company Name	Contact Name	Contact Title	Address	City	Region	Postal Code	Country	Phone
ABPAI	ABC Painters	Scott Summers	COO	1110 Washington Drive	Somewhere	Southwest	654321	Canada	987-654-3210

If you would like to verify that only the Customer is submitted, add an Order and use the command that submits only the Customer data - you'll find the Customer is added / updated and the order is not.

You can also make this more interesting by allowing the user to copy in Orders or Order Details - using CopyTable or Insert - so those could be updated. ID field mapping is required for the child lists as well, you may want to reference the **ID** field for each table in the Data Source Task Pane. . You can learn a bit more about that [here](#).

And also, you'd probably be submitting the XML to a form library in a real form - so you would open the form to update the order.

So, now our form submits to three related SharePoint lists. It can include attachments. The user can delete attachments. [Here's](#) the final form, in case you want to take a look at any of the logic.

Use SubmitToSharePointList with a Calendar

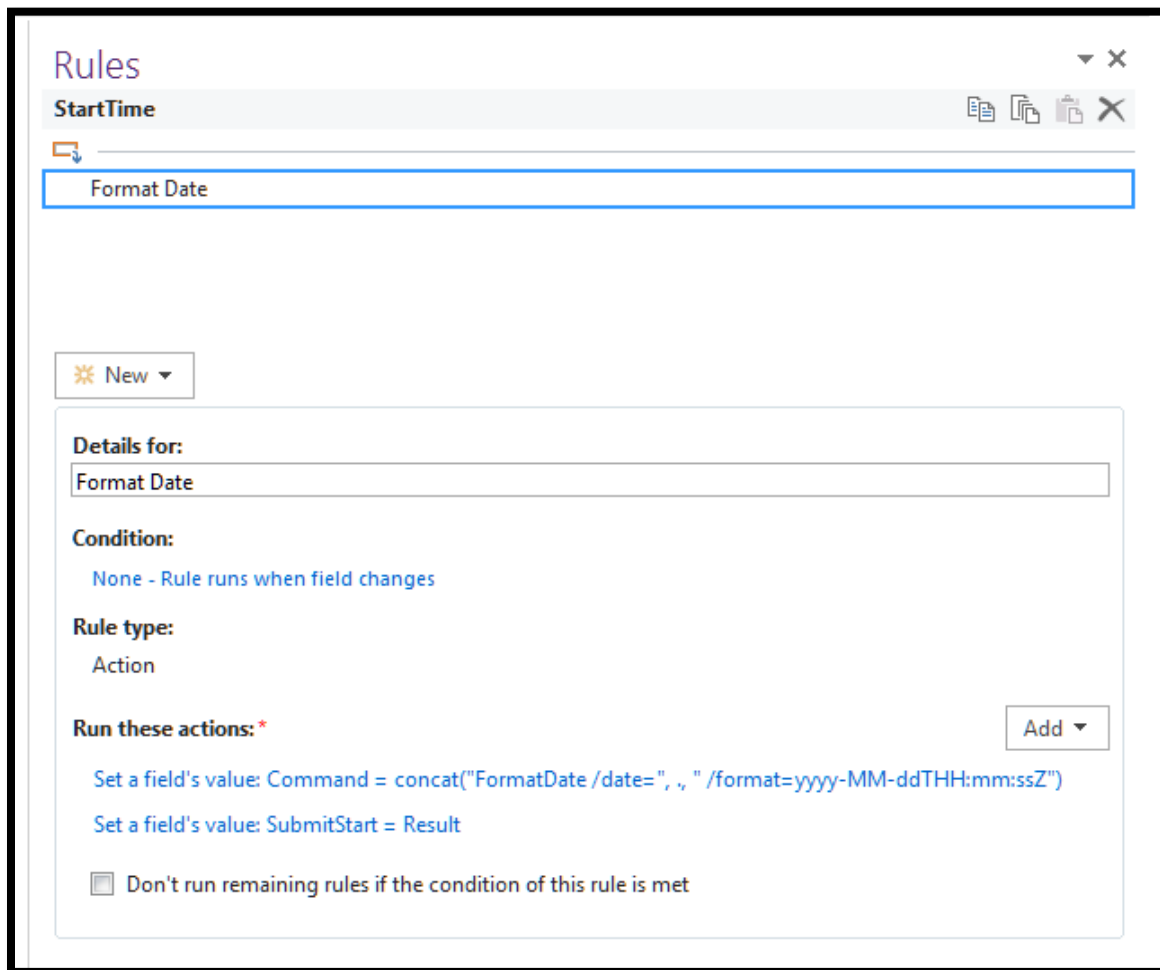
The qRules SubmitToSharePointList command is a great way to get information from your InfoPath form into a SharePoint List - however, there was a [thread](#) in the InfoPathDev forum where a qRules user was having problems submitting form data to a Calendar. The error was:

"Errors occurred submitting to list: Invalid date/time value. A date/time field contains invalid data. Please check the value and try again."

It turns out that the date time for the Calendar list required a different format:

yyyy-MM-ddTHH:mm:ssZ

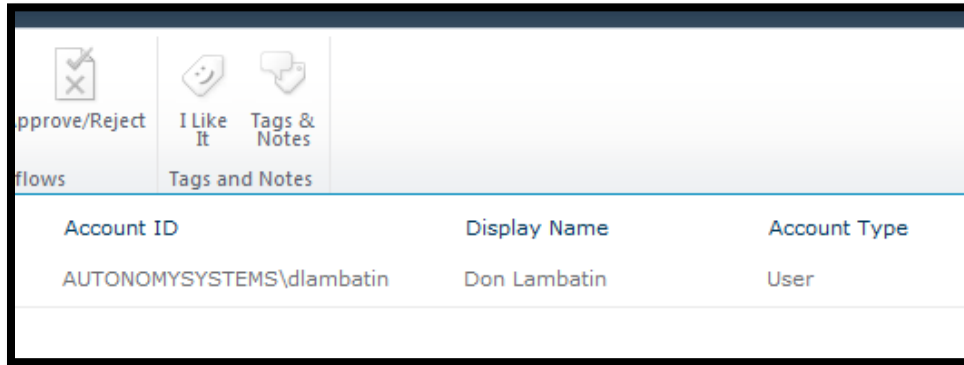
The qRules FormatDate command can return a date with a custom format. If you need to submit to a calendar, you may want to leave regular date time controls on your form for your users to select with, but have another field that is mapped to your SharePoint list. Set that field to the result of the FormatDate command; so that your Start and End date have the correct format prior to submitting.



The screenshot shows the 'Rules' editor window. At the top, the rule is named 'StartTime'. Below the name, there is a search bar and a 'Format Date' button. A 'New' button is visible on the left. The 'Details for:' section shows 'Format Date'. The 'Condition:' section is set to 'None - Rule runs when field changes'. The 'Rule type:' is set to 'Action'. The 'Run these actions:' section contains two actions: 'Set a field's value: Command = concat("FormatDate /date= ", , " /format=yyyy-MM-ddTHH:mm:ssZ")' and 'Set a field's value: SubmitStart = Result'. There is an 'Add' button next to the actions list. A checkbox at the bottom is labeled 'Don't run remaining rules if the condition of this rule is met'.

Submitting a Person/Group Picker to a SharePoint list

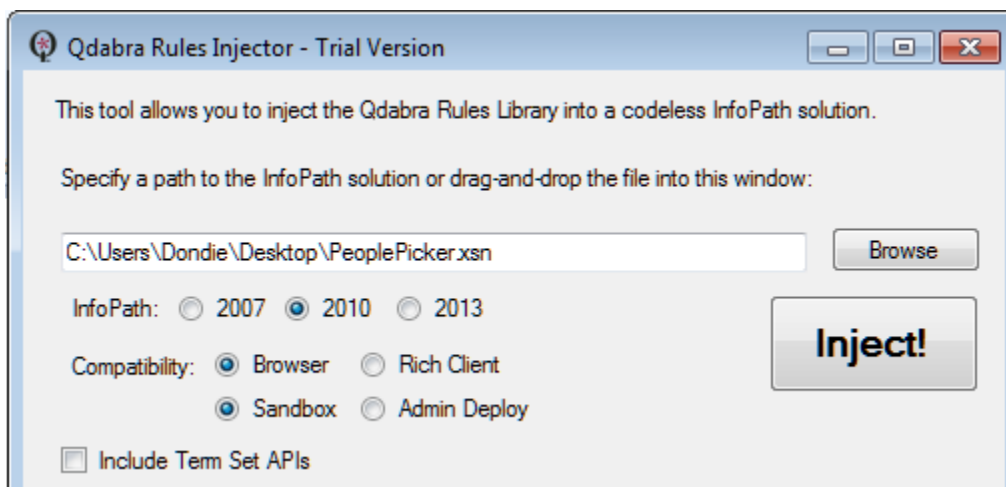
The Issue



When submitting data from a Person/Group Picker to a SharePoint list, the available fields are: **Display Name, Account ID, and Account Type**. To successfully submit to a Person or Group column type in a SharePoint list, we need to have the correct ID number for the user from the site – or convince SharePoint to look it up for us by using the following work around.

This guide uses SharePoint 2010 and assumes that the form is already pre-injected with qRules, if not kindly navigate to the qRules folder located under the Qdabra folder and run the qRules Injector.

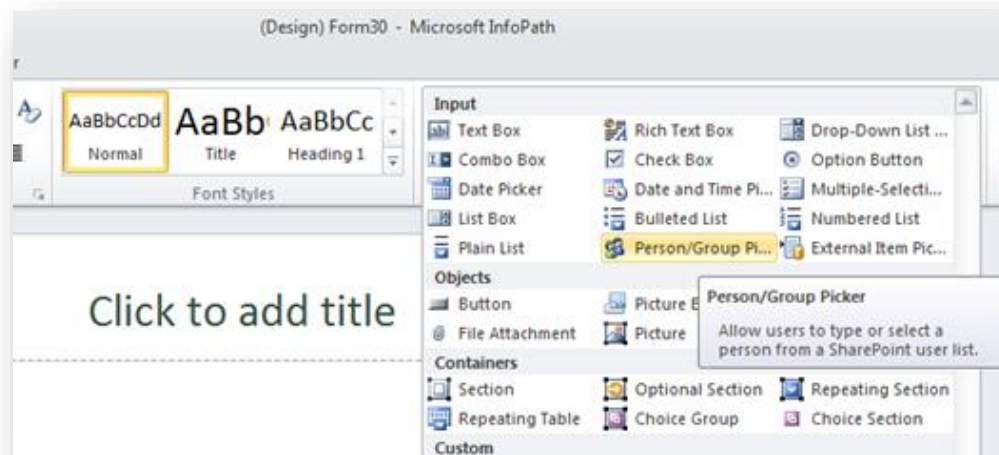
Start Menu → Tools → qRules → qRules Injector



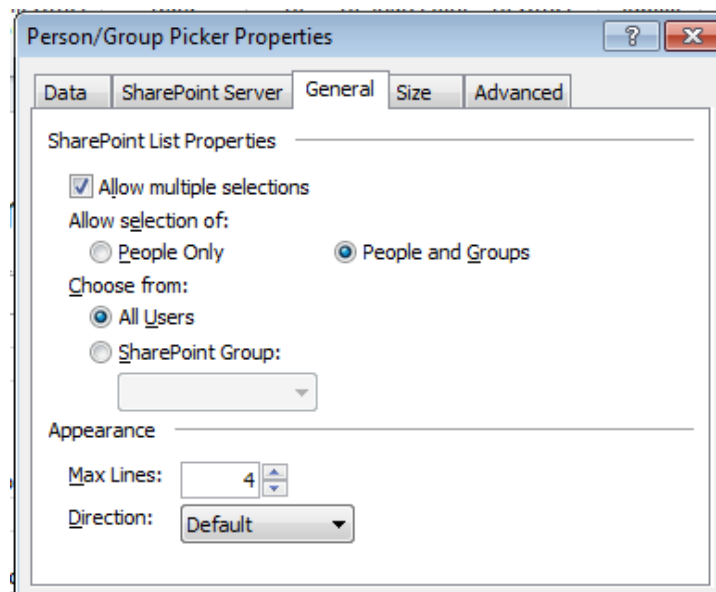
Browse for the form and inject it with qRules.

Adding a Person/Group Picker to a form

Let's start by adding a Person/Group Picker, or commonly called as People Picker, to our form. Navigate towards the Controls Gallery on the Home tab or the Controls task pane and click the Person/Group Picker control.

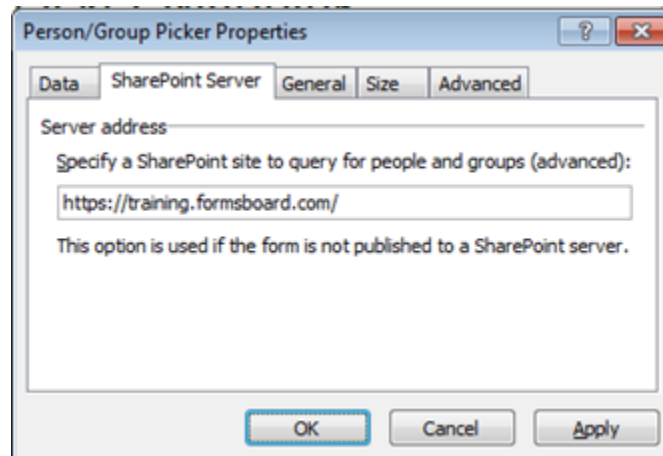


By default, a Person/Group Picker is configured to select a single contact and a selection of People Only. This can be modified at the General tab of the Person/Group Picker properties. Be sure to remember these properties as we will need to have the SharePoint columns synchronized to these settings.



Along the tabs is a SharePoint server property. This tab allows us to specify which SharePoint server to query our contacts from, unless the form has already been published to a SharePoint site, in which case that site will be used.

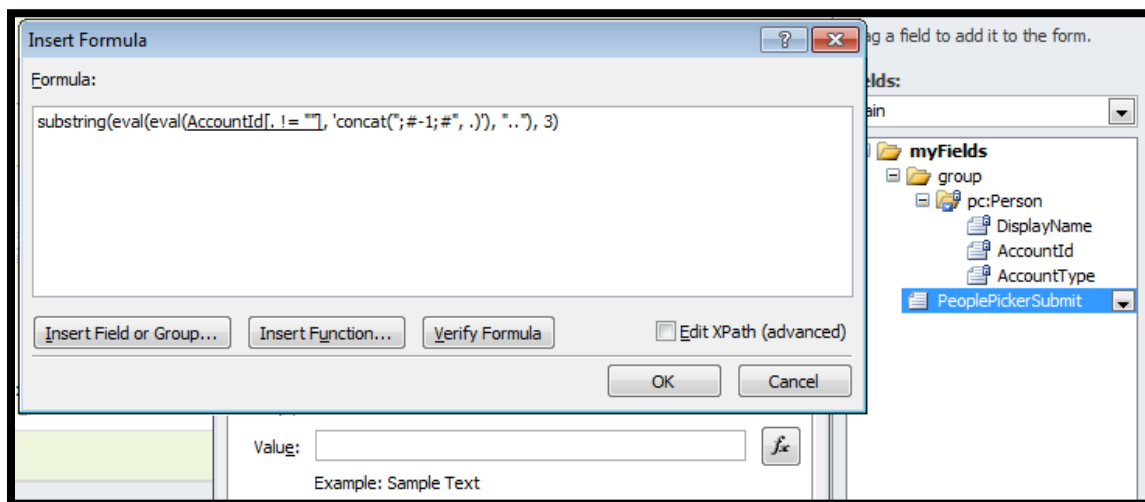
To specify which site at the SharePoint server tab, simply type in the URL of the site we wish to query.



Workaround

After inserting a People Picker control, let's add the solution.

First we must have a field added outside the People Picker control group. This field will be designated to manage the value of the AccountId field after querying, so let's call it **PeoplePickerSubmit**. Also, this field does not have to be visible on the form.



We then add a formula at the **fx** button which will allow us to submit the proper value to the SharePoint list.

Formula: `substring(xdMath:Eval(xdMath:Eval(AccountId[. != ''], 'concat(";#-1;#;', '.)'), '..'), 3)`

Notice that we are using a **double eval technique** and **substring** in order for us to have a specific part of a text string returned (AccountId). The **concat** function on the other hand the format and allows us to have multiple selections.

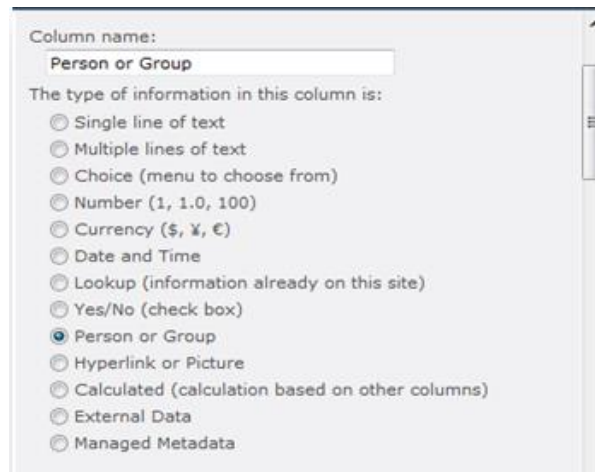
Be sure to select the AccountId field using the field or group selection button and add a filter with a condition that the AccountId is not blank.

In case you are using SharePoint 2013 or O365, you might want to check this webinar out and use a different Account Id format: [Migration Issue from SharePoint 2010 to 2013](#)

Mapping

The **PeoplePickerSubmit** field is now able to generate a value we can submit, and we may proceed on mapping this field onto our SharePoint list.

In creating a People Picker column at the SharePoint list, the column should be synchronized with the settings of the control.



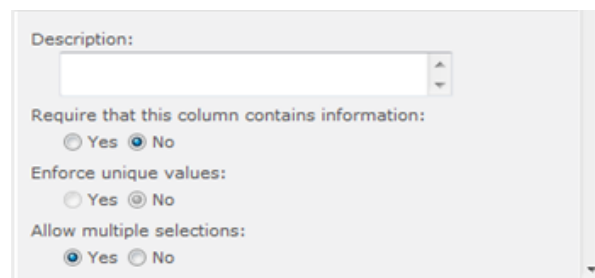
Column name:
Person or Group

The type of information in this column is:

- ☐ Single line of text
- ☐ Multiple lines of text
- ☐ Choice (menu to choose from)
- ☐ Number (1, 1.0, 100)
- ☐ Currency (\$, ¥, €)
- ☐ Date and Time
- ☐ Lookup (information already on this site)
- ☐ Yes/No (check box)
- ☒ Person or Group
- ☐ Hyperlink or Picture
- ☐ Calculated (calculation based on other columns)
- ☐ External Data
- ☐ Managed Metadata

The diagram above indicates that the column information is for a Person or Group.

And allow multiple selections:



Description:

Require that this column contains information:
☐ Yes ☒ No

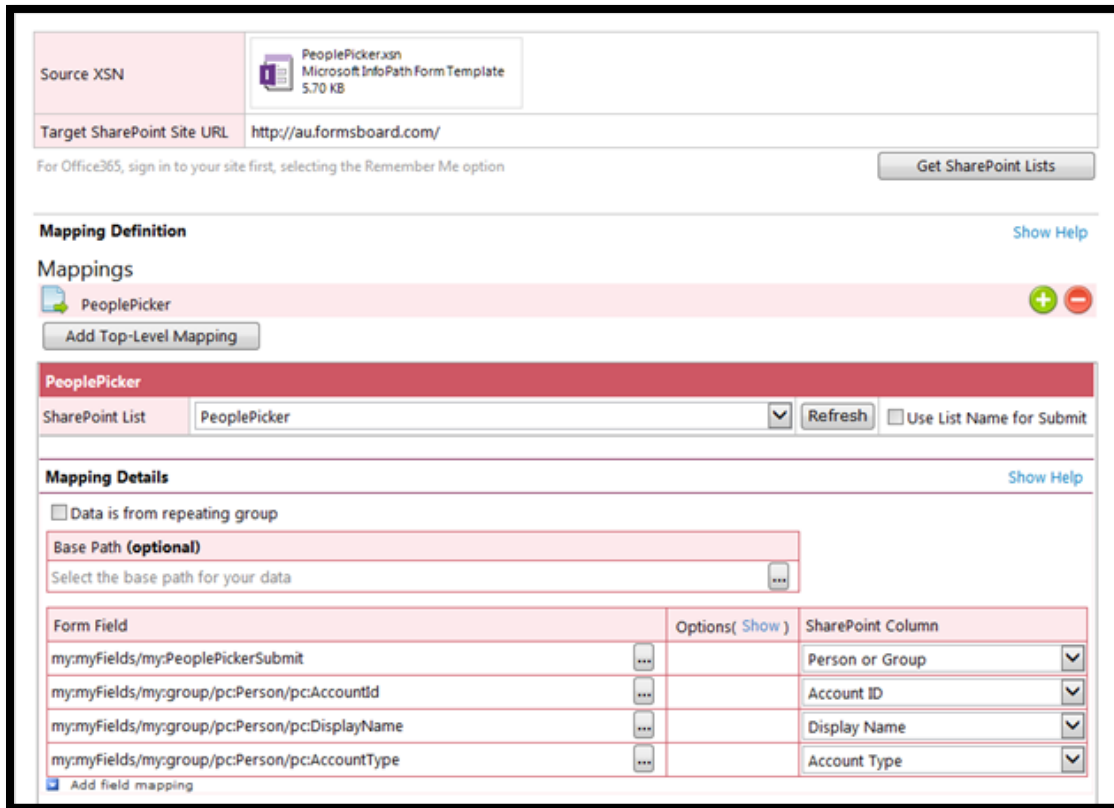
Enforce unique values:
☐ Yes ☒ No

Allow multiple selections:
☒ Yes ☐ No

After creating columns, we can proceed mapping the fields using the SharePoint List Tool.



Below is an example of a Person/Group Picker mapping which will later on show the difference between a direct mapping and the workaround we performed.



The screenshot shows the 'Mapping Definition' window in Microsoft InfoPath. The 'Source XSN' is 'PeoplePicker.xsn' (Microsoft InfoPath Form Template, 5.70 KB). The 'Target SharePoint Site URL' is 'http://au.formsboard.com/'. A button 'Get SharePoint Lists' is visible. Below, the 'Mappings' section shows a 'PeoplePicker' mapping. The 'Mapping Details' section includes a checkbox for 'Data is from repeating group' (unchecked), a 'Base Path (optional)' field, and a table for field mappings.

Form Field	Options(Show)	SharePoint Column
my:myFields/my:PeoplePickerSubmit	...	Person or Group
my:myFields/my:group/pc:Person/pc:AccountID	...	Account ID
my:myFields/my:group/pc:Person/pc:DisplayName	...	Display Name
my:myFields/my:group/pc:Person/pc:AccountType	...	Account Type

At the bottom, there is a checkbox for 'Add field mapping' which is checked.

Save... and add this mapping to the form as a Data Connection which receives data as an XML document.

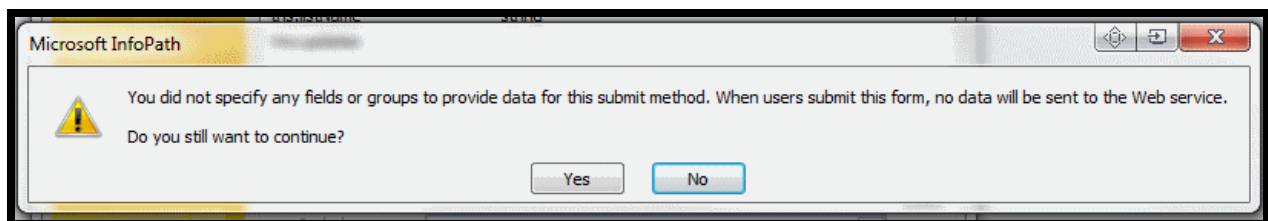
Submit Button

At Data Connections, add a connection which will allow us to submit data using the UpdateListItems operation of the SharePoint Lists web service. That web service URL should similar to this:

http://myserver/site/_vti_bin/lists.asmx?WSDL

With "[myserver/site/](#)" replaced with the actual SharePoint site URL.

We do not need to select any values for the parameters - InfoPath will display a warning, but we can safely ignore it:

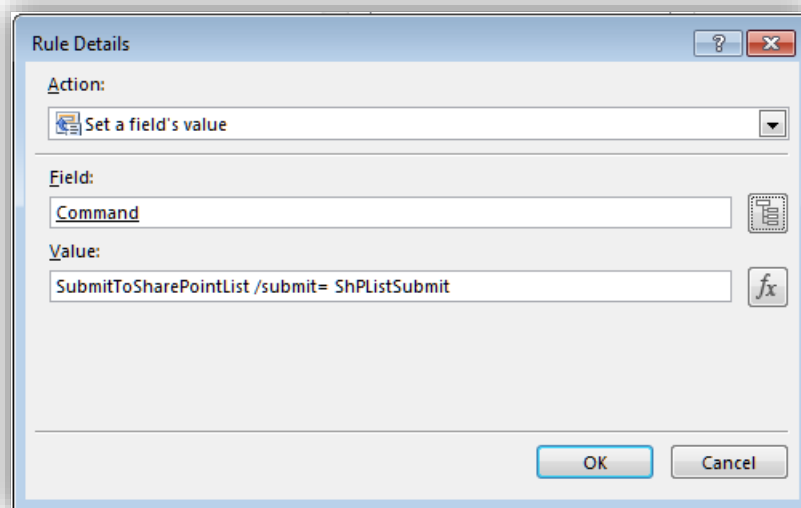


qRules will set the input parameters for you when you execute the command. Click "Yes" on this warning dialog and finish creating the connection using an identifiable name such as "ShPListSubmit".

Save the form.

Assuming that the form is already injected with qRules, if not, have it injected with qRules in order for us to submit to a list.

Re-open the form, click the submit button, and add a rule at the properties tab which will set a field's value when the button is clicked.



For Field, let's use **Command** of the QdabraRules Secondary source.

Value:

SubmitToSharePointList /submit= **ShPListSubmit**

Note:

If the mapping connection has been saved under a different name other than the default "mapping" name, we would need to declare this parameter, otherwise omit this procedure.

Ex. SubmitToSharePointList /submit= **ShPListSubmit** /mapping= **MapplnG1234**

Both parameters are **CaSe SENSiTivE**.

Test the Form!

People Picker Mapping

Don Lambatin

Display Name	Account Id	Account Type
Don Lambatin	AUTONOMYSYSTEMS\dmbatin	User

Insert item

Submit

People Picker Submit:

-1;#AUTONOMYSYSTEMS\dmbatin

Here we can see that the **PeoplePickerSubmit** has the value of “-1;#” and the account ID from the people picker. Clicking Submit will submit the data to the SharePoint list – and SharePoint will look up the correct ID for each user. Alternatively, you can click the “Address book” button to perform a search if you do not know the complete name of the user, and at the same time allows us to choose multiple contacts; as seen on the image below:

Person/Group Picker

jim

Display Name	Title	Department	E-Mail	Account Name
AUTONOMYSYSTEMS\jim...	Solutions Architect		Jim.Cantwell@Qdabra.c...	AUTONOMYSYS
Jim Cantwell (admin)				AUTONOMYSYS
Jim Cantwell (QDABRA)			jim.cantwell@qdabra.com	QDABRA\jimcar
Jim Cantwell				AUTONOMYSYS

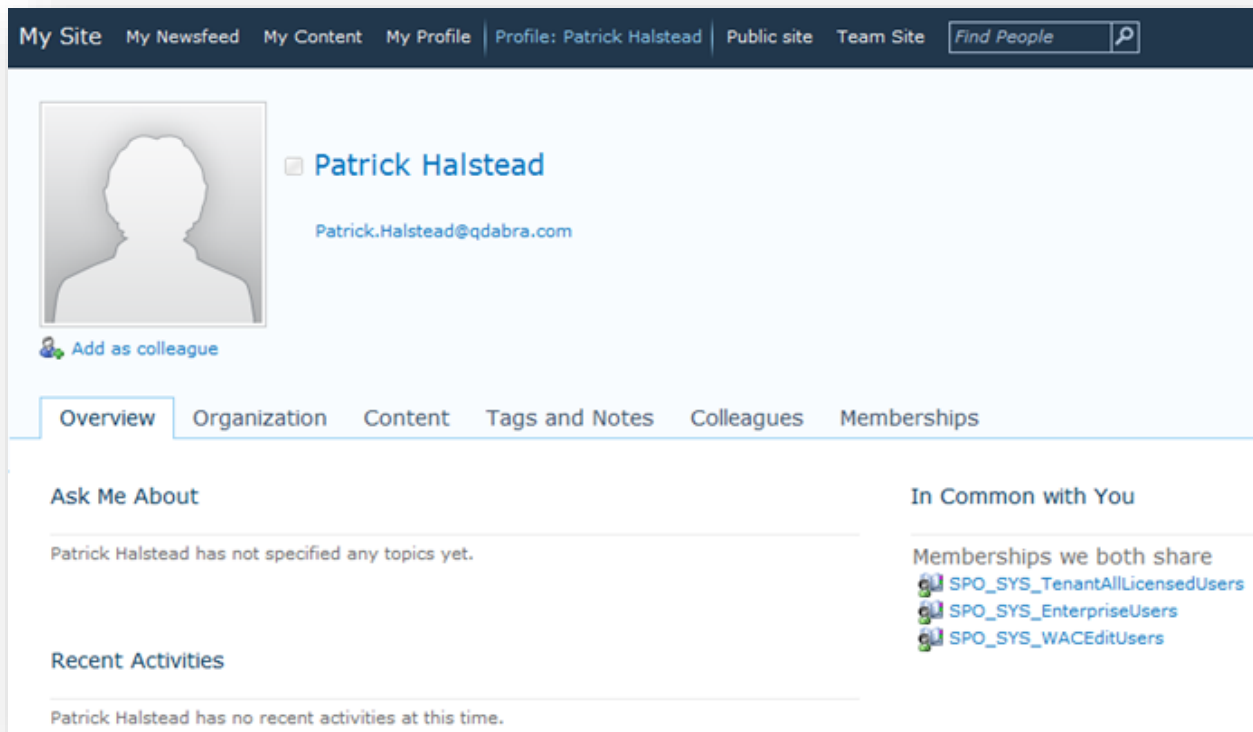
Add >>

Mel Test - Client: AUTONOMYSYSTEMS\MBalsamo; Jim Cantwell (QDABRA)

OK Cancel

Search this site...			
☐ Person or Group	Account ID	Display Name	Account Type
Mel Balsamo	i:0#.f membership mel.balsamo@qdabra.com	Mel Balsamo	User
Don Lambatin	i:0#.f membership don.lambatin@qdabra.com	Don Lambatin	User
Don Lambatin James Rishe Patrick Halstead Jim Cantwell	i:0#.f membership don.lambatin@qdabra.com	Don Lambatin	User

In comparison to the other fields in the screenshot above, it is evident that the **PeoplePickerSubmit** field allows sending multiple selections as long as it has been configured properly in the form and at the SharePoint List. Also, the data entry is a hyperlink which can re-direct us to the contact's profile.



My Site | My Newsfeed | My Content | My Profile | Profile: Patrick Halstead | Public site | Team Site | Find People

 **Patrick Halstead**
Patrick.Halstead@qdabra.com



Overview | Organization | Content | Tags and Notes | Colleagues | Memberships

Ask Me About
Patrick Halstead has not specified any topics yet.

Recent Activities
Patrick Halstead has no recent activities at this time.

In Common with You
Memberships we both share

-  SPO_SYS_TenantAllLicensedUsers
-  SPO_SYS_EnterpriseUsers
-  SPO_SYS_WACeditUsers

Aside from the above given functionalities of **qRules** with regards to submitting to a SharePoint library or list, there are still other various methods and techniques related to the `SubmitToSharePointList` command using **qRules**. One of which is refreshing your SharePoint List items. `SubmitToSharePointList`

performed with a **RefreshSharePointListItems** command will submit your updated data to a SharePoint list and refresh the previous listed data.

Another command also commonly used is **AttachToSharePointList**. As its name would suggest, the command deals with adding attachments to your InfoPath form and submits the file to a SharePoint List. If you have added attributes to your attachment field such as “qRulesLink” and “qRulesFilename”, the command will clear your attachment field after uploading the attachment and return the new link to the attachment and the file name to your attribute nodes.

[Here](#) is link for a guide regarding other qRules and SharePoint-related Commands, such as RefreshSharePointListItems and AttachToSharePointListItem.

General Mapping Overview

After using the SubmitToSharePointList command in qRules, it would be best to provide a brief overview on how the command works.

At its most basic, the command requires:

1. A submit data connection to the SharePoint Lists web service's UpdateListItems method.
2. A mapping file indicating the relationship between the InfoPath form's fields and the SharePoint list.
3. A data connection to the mapping file.

The mapping file is used by the command code to discern the following:

1. The URL for the SharePoint server for the list the data should submit to.
2. The GUID or name of the SharePoint list to submit the data to.
3. The form field to list column correspondences for the data.

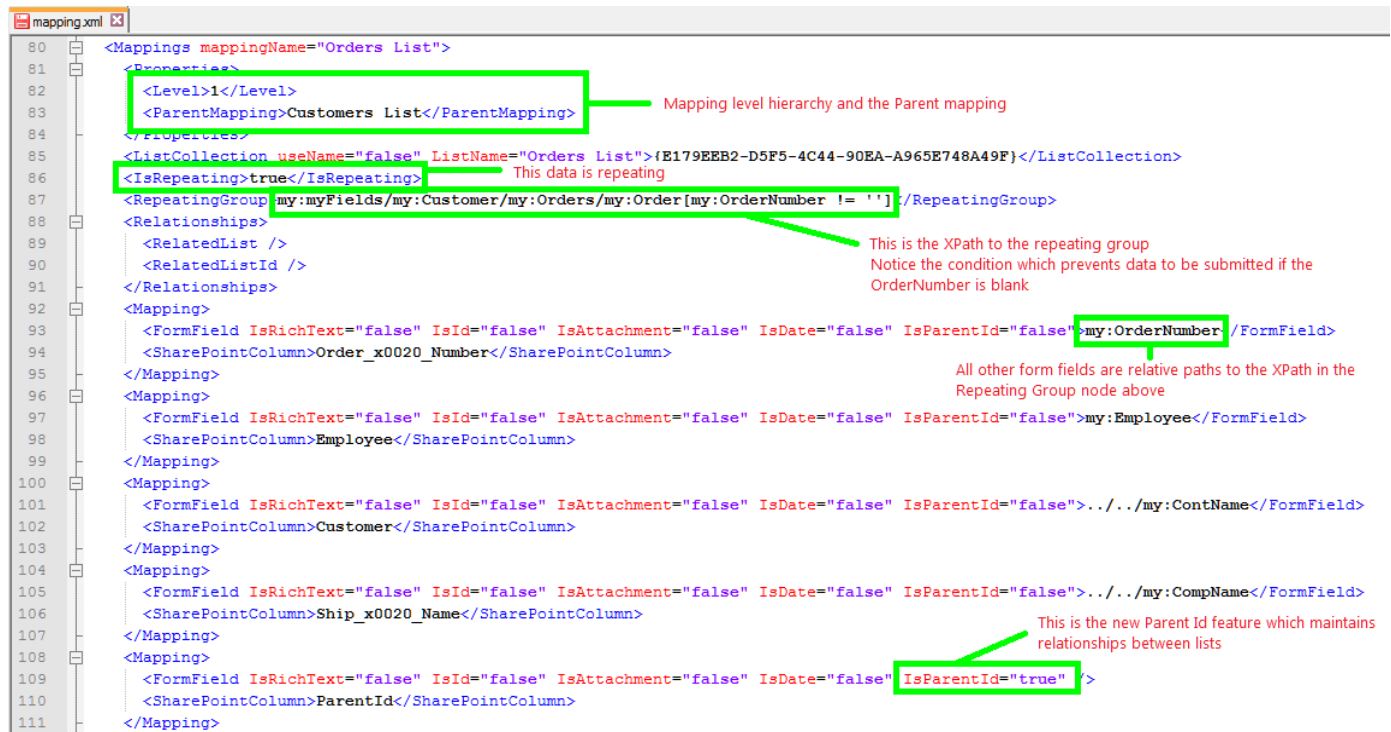
Here's what the mapping XML looks like - marked up to help you understand the data:



The screenshot shows an XML file named 'mapping.xml' with the following structure and annotations:

- Root Element:** `<?xml version="1.0" encoding="utf-16"?>` (Annotated: "Use the mapping name if your mapping has multiples and you only wish to submit one")
- myFields Element:** `<myFields schemaVersion="6.1">`
- Mappings Element:** `<Mappings mappingName="Customers List">` (Annotated: "This displays the hierarchy level of the mapping")
- Properties Element:** `<Properties>`
 - `<Level>0</Level>` (Annotated: "If useName is true, qRules will submit and query using the list name instead of the GUID")
 - `<ParentMapping />`
- ListCollection Element:** `<ListCollection useName="false" ListName="Customers List" 8BE5B259-295B-4459-85A5-284F765C84C1 />` (Annotated: "GUID for the list - you could change this in the form to a different GUID if you wanted, using Rules- say you want to re-use the form on a different site location...")
- IsRepeating Element:** `<IsRepeating>false</IsRepeating>` (Annotated: "Indicates if the mapping is from a repeating group")
- RepeatingGroup Element:** `<RepeatingGroup />`
- Relationships Element:** `<Relationships>`
 - `<RelatedList />` (Annotated: "More plans for the future... not currently in use")
 - `<RelatedListId />`
- Mapping Element:** `<Mapping>`
 - `<FormField IsRichText="false" IsId="false" IsAttachment="false" IsDate="false" IsParentId="false">my:myFields/my:Customer/my:CompName</FormField>` (Annotated: "This is a field to column mapping - the CompName field in the XML is mapped to the Company Name column the SharePoint List")
 - `<SharePointColumn>Company_x0020_Name</SharePointColumn>`
- Mapping Element:** `<Mapping>`
 - `<FormField IsRichText="false" IsId="true" IsAttachment="false" IsDate="false" IsParentId="false">my:myFields/my:Customer/my:ID</FormField>` (Annotated: "This field will be populated with the list item id on a successful submit")
 - `<SharePointColumn />`
- Mapping Element:** `<Mapping>`
 - `<FormField IsRichText="false" IsId="false" IsAttachment="true" IsDate="false" IsParentId="false">my:myFields/my:Customer/my:File</FormField>` (Annotated: "This field contains a list item attachment - like the id field above, it is not mapped to a list column. The /dsname parameter must be declared at the submit command to add the attachment")
 - `<SharePointColumn />`
- Batch Element:** `<Batch OnError="Continue">`
 - `<Method ID="1" Cmd="New">`
 - `<Field Name="Company_x0020_Name" />` (Annotated: "This is the skeleton for the CAML batch that will be created by qRules when the command executes")
 - `</Method>`
- Mappings Element:** `</Mappings>`
- Mappings Element:** `<Mappings mappingName="Orders List">` (Annotated: "Collapsed mappings node - we'll look at this in the next screenshot")
- Mapping Element:** `<Mapping>`
 - `<SharePointListURL>http://au.formsboard.com/</SharePointListURL>` (Annotated: "SharePoint server URL - the command will modify the submit data connection to use this URL when submitting using this mapping file. This is something else that could be changed using Rules prior to command execution")
- myFields Element:** `</myFields>`

In the screenshot below, the "Orders List" mapping has been expanded so you can see how the mapping changes when we are mapping repeating data:



```

80 <Mappings mappingName="Orders List">
81 <Properties>
82   <Level>1</Level>
83   <ParentMapping>Customers List</ParentMapping>
84 </Properties>
85 <ListCollection useName="false" ListName="Orders List">{E179EEB2-D5F5-4C44-90EA-A965E748A49F}</ListCollection>
86 <IsRepeating>true</IsRepeating>
87 <RepeatingGroup my:myFields/my:Customer/my:Orders/my:Order[my:OrderNumber != '']>/RepeatingGroup>
88 <Relationships>
89   <RelatedList />
90   <RelatedListId />
91 </Relationships>
92 <Mapping>
93   <FormField IsRichText="false" IsId="false" IsAttachment="false" IsDate="false" IsParentId="false">my:OrderNumber</FormField>
94   <SharePointColumn>Order_x0020_Number</SharePointColumn>
95 </Mapping>
96 <Mapping>
97   <FormField IsRichText="false" IsId="false" IsAttachment="false" IsDate="false" IsParentId="false">my:Employee</FormField>
98   <SharePointColumn>Employee</SharePointColumn>
99 </Mapping>
100 <Mapping>
101   <FormField IsRichText="false" IsId="false" IsAttachment="false" IsDate="false" IsParentId="false">../../my:ContName</FormField>
102   <SharePointColumn>Customer</SharePointColumn>
103 </Mapping>
104 <Mapping>
105   <FormField IsRichText="false" IsId="false" IsAttachment="false" IsDate="false" IsParentId="false">../../my:CompName</FormField>
106   <SharePointColumn>Ship_x0020_Name</SharePointColumn>
107 </Mapping>
108 <Mapping>
109   <FormField IsRichText="false" IsId="false" IsAttachment="false" IsDate="false" IsParentId="true">
110   <SharePointColumn>ParentId</SharePointColumn>
111 </Mapping>

```

Mapping level hierarchy and the Parent mapping

This data is repeating

This is the XPath to the repeating group
Notice the condition which prevents data to be submitted if the OrderNumber is blank

All other form fields are relative paths to the XPath in the Repeating Group node above

This is the new Parent Id feature which maintains relationships between lists

If there is an XPath in the RepeatingGroup node, all the other FormField values need to be relative to that XPath. You will see it starting at the my:Order node (using the example above).

Support

If you have questions about the information in this document, please contact Qdabra Software for assistance. The qRules product page on Qdabra.com will contain an updated list of all available documentation.

Licensed customers can contact us via [Submit a Support Request](#) or Support@Qdabra.com. You can also use the InfoPathDev.com [Qdabra Product support forums](#) to request help from the community.

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