

# QRULES AND SHAREPOINT-RELATED COMMANDS

**PRODUCT:** qRules v6.1 or higher  
**LAST UPDATED:** September 14, 2015

## SCENARIO

Have you ever wanted to be able to add InfoPath form data to SharePoint list? Sure, promoted properties in your form library are fine – for one layer deep data. Real world data often has one to many relationships, however (think Customers and Orders, or Orders and Products), and SharePoint and InfoPath like to handle that by mashing all your nicely organized information into semi-colon delimited lists. Not so pretty, not so useful.

Perhaps you also have attachments in your form that you'd like added to existing list items, speeding up your form by keeping it sleek and having the attachments out of the XML where they are easier to find. InfoPath and SharePoint won't do this for you, but qRules can.

What about updating your list items in your InfoPath form if they were changed after the form was submitted and items were submitted to your list? Wouldn't that be great? You could have the best of both worlds.

## VALUE PROPOSITION

You could write code to submit your list items. You could even add a secondary data connection to your form to help you generate CAML and submit repeating items one at a time. Of course, that means you have to write and maintain code, or ever worse, rely on your users to push the "Submit" button on each item. They'd never forget to do that, right?

With qRules, you can easily add the ability to submit information from your form to a list or to two lists. You can map the fields you want to submit. You can use additional commands to add any attachments to the list items. And, if you want, qRules will even refresh the list items in the form with the latest from the list. qRules consists of modular commands that you can leverage together to create powerful forms with all the features that custom code provides and none of the maintenance that custom code carries along with it.

## REQUIREMENTS

- qRules v6.1 or higher
- Access to the SharePoint List you wish to submit items to
- InfoPath To SharePoint List tool for mapping (this tool is provided with qRules)



## HOW TO USE THE SUBMITTOSHAREPOINTLIST COMMAND

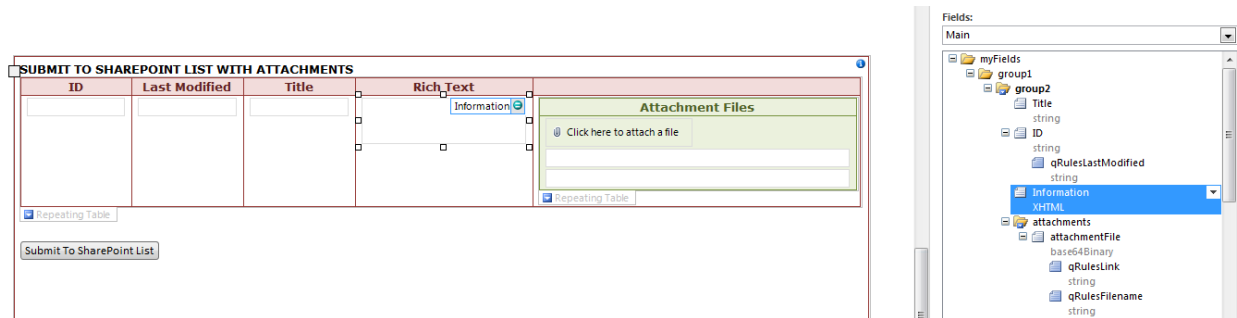
The **SubmitToSharePointList** command features the ability to submit data from an InfoPath form to the SharePoint list it has been mapped to. It can be used to submit rich text to SharePoint lists, and even supports submitting attachments (single or repeating).

For qRules v6.1 or higher, the /id parameter no longer needs to be specified in the command. We can now map the Id in the mapping tool instead and specify the node as an Id, allowing the form to update the list item when re-submitted, and allowing the form to be updated from the list item when the **RefreshSharePointListItems** command is executed.

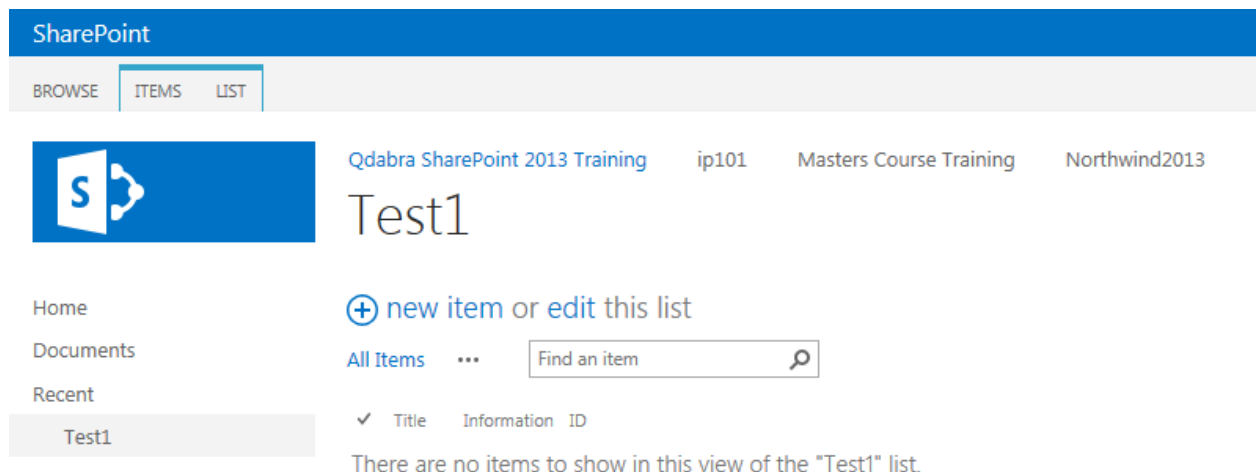
This command requires the use of Qdabra's **InfoPath to SharePoint List Tool** to create the XML file that will be used for mapping the fields in the InfoPath form to the columns in the SharePoint list. Refer to this [user guide](#) for in-depth instructions on how to use the tool, paying special attention to instructions on Mapping, as the Import functionality of that tool is not used in qRules.

### Create a mapping file and add to template

As an example, we will use a simple InfoPath form template (injected with qRules v6.1 or higher) that has a repeating table with the fields: id, last modified, title, information, and attachment:

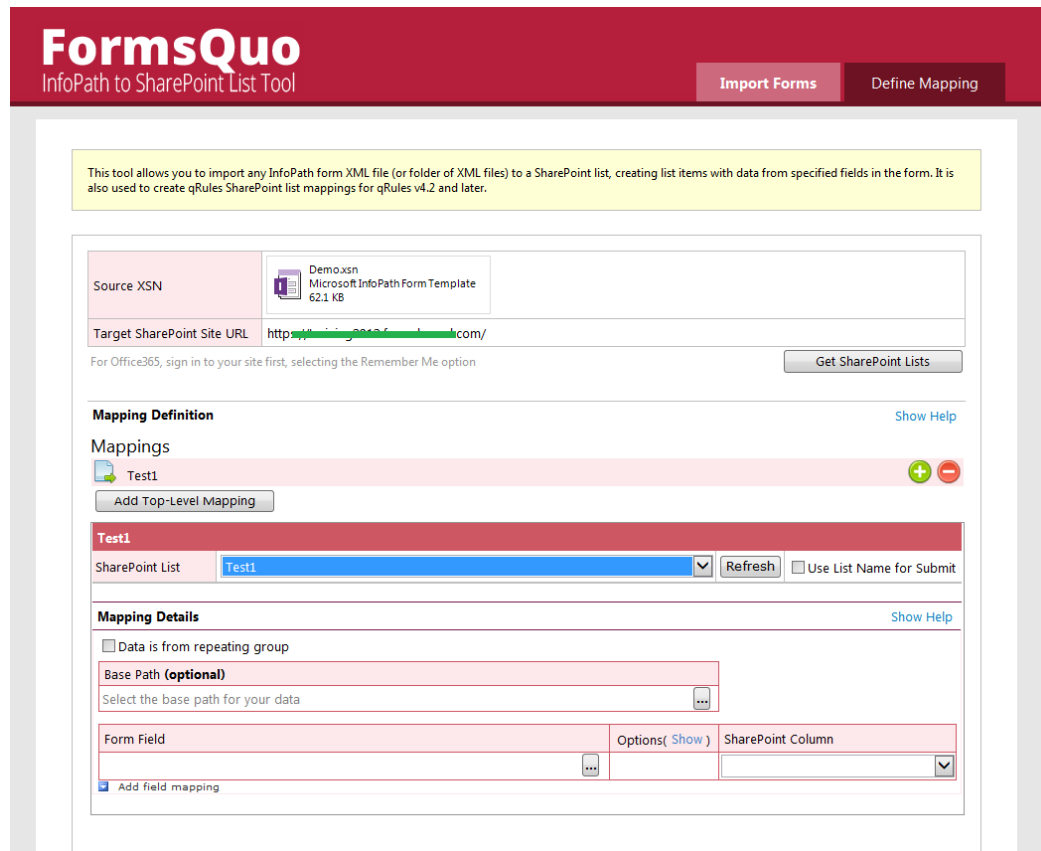


We will then map these fields onto our SharePoint list which also has three columns:

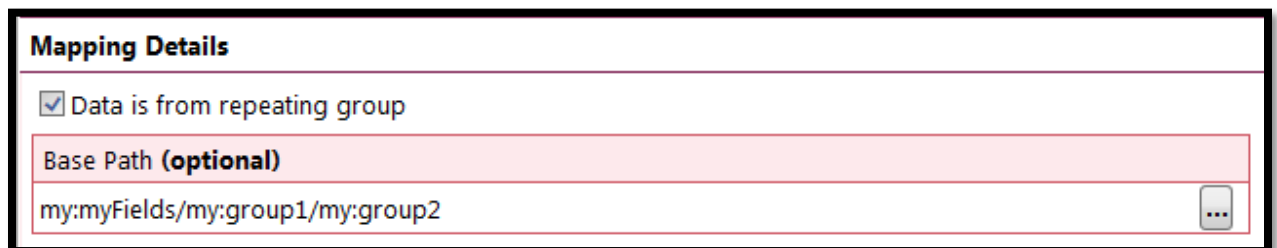



For SharePoint 2013, the ID column has been added via List Settings > Views > All items.

1. Launch **InfoPath to SharePoint List Tool** and go to the **Define Mapping** tab:
2. In the **Source XSN** field, attach the InfoPath form template (XSN).
3. Enter the root SharePoint site in the **Target SharePoint Site URL** field, e.g. <http://SharePointSite>
4. Click on the **Get SharePoint Lists** button.
5. In the **SharePoint Lists** dropdown, select the list you wish to submit items to, in this sample, "Test1" (yeah, we are crazy creative like that with our names).



6. Since we're using a repeating group in our form, we will have to check the box "Data is from repeating group" in the **Mapping Definition** section.
7. Click on the  icon to display the Schema Tree taskpane and select the repeating group for your data, which in this case is *my:myFields/my:group1/my:group2*




8. Click on the  icon in each **Form Field** row to select the fields from the Schema Tree taskpane, and then select the corresponding **SharePoint Column** to map the fields to. You can also indicate if any of the fields you wish to submit should be treated as Rich Text.

| Form Field     | Options( <a href="#">Hide</a> )                                                                                     | SharePoint Column |
|----------------|---------------------------------------------------------------------------------------------------------------------|-------------------|
| my:Title       | <input type="checkbox"/> Id<br><input type="checkbox"/> Rich Text<br><input type="checkbox"/> Attachment            | Title             |
| my:Information | <input type="checkbox"/> Id<br><input checked="" type="checkbox"/> Rich Text<br><input type="checkbox"/> Attachment | Information       |

[Add field mapping](#)

- Map the Id in the mapping tool instead and specify the node as an Id (see second paragraph of section [How to use the SubmitToSharePointList command](#)).
- (Optional) For attachments, select the intended field for attachment and tick the Attachment checkbox for the Files node mapping.

| Mapping Details                                                  |                                                                                                                     | <a href="#">Show Help</a>                                                                                                                                                                                           |
|------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <input checked="" type="checkbox"/> Data is from repeating group |                                                                                                                     |                                                                                                                                                                                                                     |
| <b>Base Path (optional)</b><br>my:myFields/my:group1/my:group2   |                                                                                                                     |                                                                                                                                                                                                                     |
| Form Field                                                       | Options( <a href="#">Hide</a> )                                                                                     | SharePoint Column                                                                                                                                                                                                   |
| my:Title                                                         | <input type="checkbox"/> Id<br><input type="checkbox"/> Rich Text<br><input type="checkbox"/> Attachment            | Title                                                                                                                                                                                                               |
| my:Information                                                   | <input type="checkbox"/> Id<br><input checked="" type="checkbox"/> Rich Text<br><input type="checkbox"/> Attachment | Information                                                                                                                                                                                                         |
| my:ID                                                            | <input checked="" type="checkbox"/> Id<br><input type="checkbox"/> Rich Text<br><input type="checkbox"/> Attachment | 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                                                    |
| my:attachments/my:attachmentFile                                 | <input type="checkbox"/> Id<br><input type="checkbox"/> Rich Text<br><input checked="" type="checkbox"/> 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](#)

- When done mapping, click on the **Save As qRules Mapping** button.
- Save the mapping as an XML file in your local drive. In this example, we will leave the default name as mapping.xml, although we can use any filename.
- Once done, close the tool and open the form template in InfoPath Design mode.
- Under the **Data** tab, click Resource Files and add the xml file as a resource.
- Go to **Data Connections** and add a receive data connection.
- Select XML document as the source of your data and click **Next**.
- Click the Resource Files button and select the mapping.xml file that's been saved in Step 10 and click **Next**.



16. Leave the default to include the data as a resource file in the form template and allow automatic retrieve on form open. For simplicity, we will call this data connection “**mapping**” – so that there will be no need to declare the mapping parameter at the SubmitToSharePointList command (the mapping parameter only needs to be declared if the data connection is named differently, i.e. “MySpecialMappingFile”, you would need to add the parameter **/mapping= MySpecialMappingFile**)

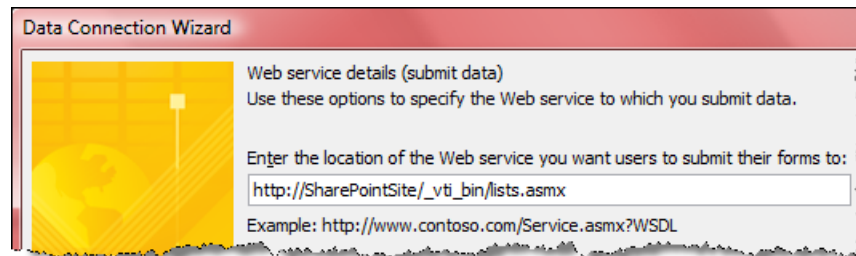
Notes:

When creating or updating the mapping data connection –

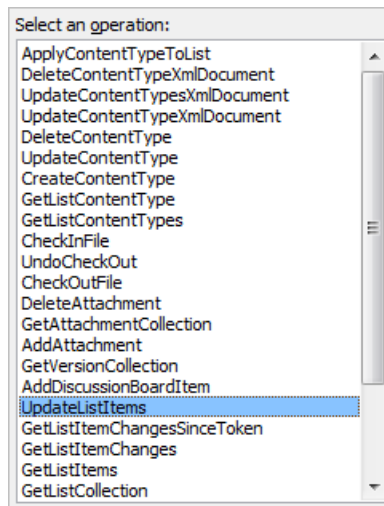
- If you don’t add the xml file to the resources first, IP removes the file extension – which is annoying.
- If you update the file by walking through the data connection again, IP keeps adding more and more copies of the file to your resource files.

### *Add a submit connection to SharePoint Lists Web Service UpdateListItems*

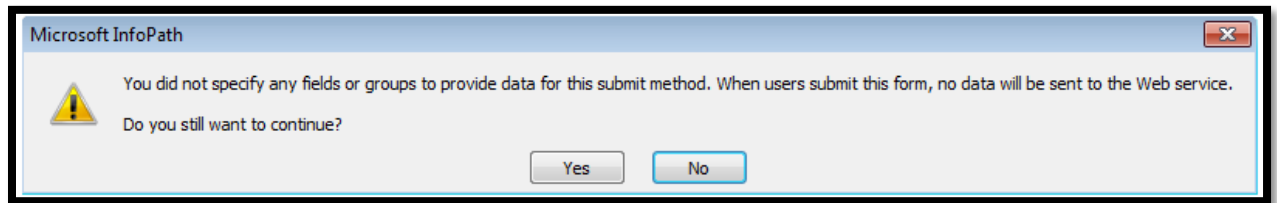
17. Add another data connection that submits data to a Web Service since SharePoint uses a web service for the list methods. The URL should be: [http://SharePointSite/\\_vti\\_bin/lists.aspx](http://SharePointSite/_vti_bin/lists.aspx) where <SharePointSite> is the root site for the SharePoint list.



18. In the list of operations, select **UpdateListItems**.



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

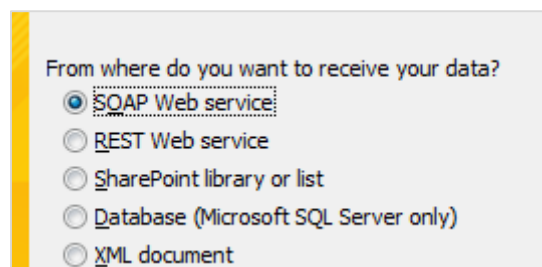


20. qRules will populate the data for the submit parameters. In the final window, it is ideal to deselect the checkbox for using the connection as the submit default, and give our connection a simple name such as **ShPSubmit** – although, like the mapping connection, you may use whatever name you like. Just be certain to use the same name in the qRules command, including the same case (i.e., if the data connection is “SharePointListSubmit” you won’t want to use “sHarEPoiNTLISTSuBMit” for the /submit parameter).

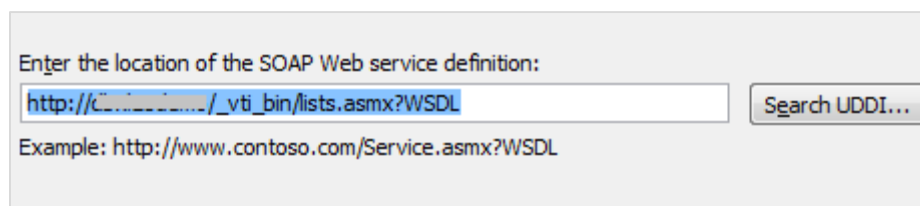
### *Add a receive connection to SharePoint Lists Web Service AddAttachment*

This section assumes that the mapping file already has an attachment field specified or has been updated to include the node intended for file attachments. See **Step 8.a** for reference – replace the xml in the resource files and walk through the **mapping** data connection wizard.

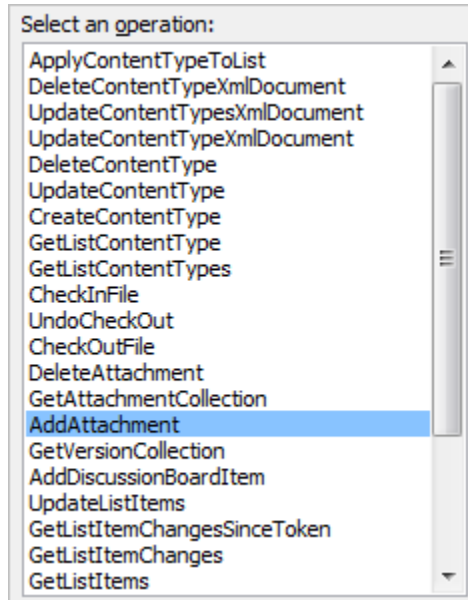
21. Add an **AddAttachment** data connection (if you intend to submit attachments with your list items):
- Select **Data > Data Connections > Add...**
  - In the first screen of the *Data Connection* wizard, select a new connection to receive data and click **Next**
  - Select the option to receive data from a SOAP Web Service and click **Next**:



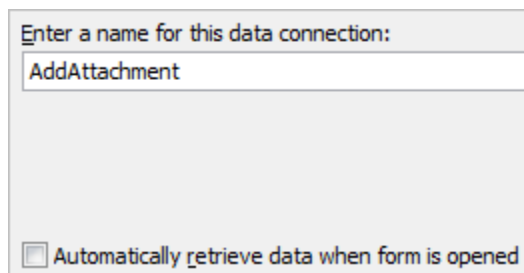
- Enter the web service url: [http://server/\\_vti\\_bin/lists.aspx?WSDL](http://server/_vti_bin/lists.aspx?WSDL), and click **Next**:



- e. Select **AddAttachment** operation, and click **Next**:



- f. Click Next - no need to set any of the query parameters, so click **Next** again.
- g. 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.



- h. Click **Finish**

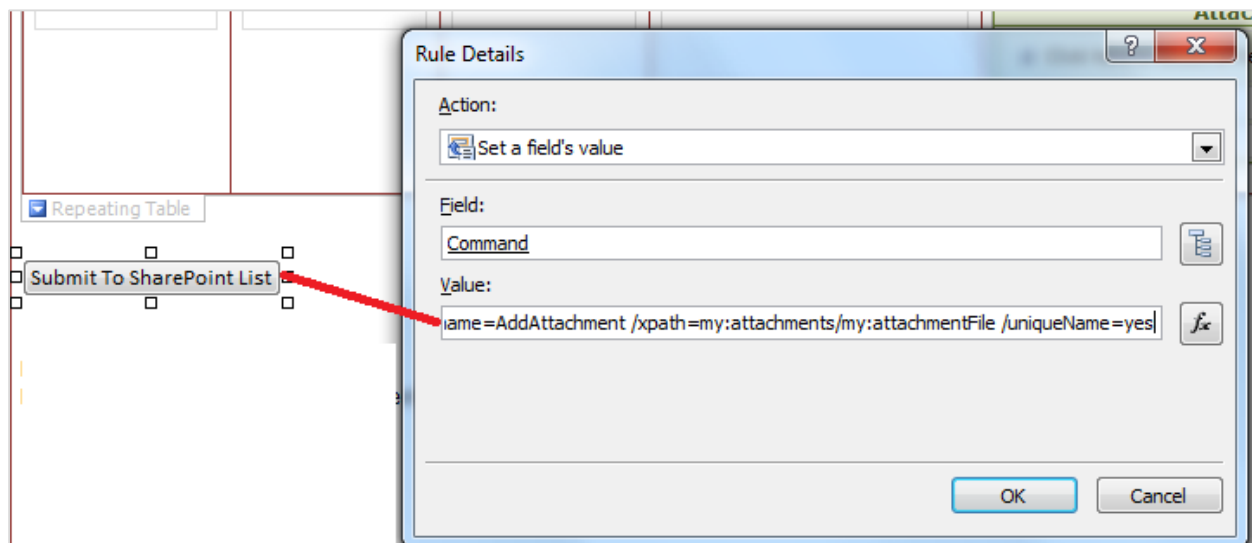
---

### Add the SubmitToSharePointListCommand and Test

22. Add a button to your form, and add a rule to it (in the sample form, this button has the label **Submit To SharePoint List**)
- Set the value of the QdabraRules **Command** field to **SubmitToSharePointList /submit=ShPSubmit /dsname=AddAttachment /xpath=my:attachments/my:attachmentFile /uniqueName=yes**
  - If an Xpath is not found, the attachment container in SharePoint will be blank.



- The portion of the command that deals with adding attachments follows the logic laid out in the section of this document on the AttachToSharePointList command. If you have added attributes to your attachment field called “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.
- Breaking the command out:
  - /submit = The name of the Submit connection you set up to your SharePoint List in Step 15
  - /mapping = The name of the Receive connection you set up to your mapping.xml file in Step 14 (**NOTE:** there is no need to declare this parameter if the Mapping Receive connection is named **mapping**, if it is named differently, i.e. “MySpecialMappingFile”, you would need to add the parameter **/mapping= MySpecialMappingFile** at the SubmitToSharePointList command)
  - /dsname = (optional) The Receive data connection you set up in Step 17 for adding attachments from your form to your submitted list items.
  - /xpath = (optional) The XPath to the field that has the attachments you want to submit with your list items. If you include this, the /dsname parameter is required.
  - /uniqueName = (optional) yes|no Defaults to no – this parameter will prepend the file name you are saving with date and time information to create a unique name. This will prevent accidental attachment overwrites, but may not be desirable if a new attachment is submitted with the list item every time the items are updated from the form.



23. Finally, in **Preview** mode, try submitting new list items by filling out the **Title** and the **Value** fields and clicking **Submit**. You should see that the item gets submitted to the SharePoint list it has been mapped to and returns the list item ID back in your form.

| ID | Last Modified       | Title     | Rich Text                                                 | Attachment Files                                                                                                                                                                                                                                                                |
|----|---------------------|-----------|-----------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2  | 2015-09-06 17:39:44 | 11-Lab 4b | Creating an Infopath web part to infopath web part mashup | <p>Click here to attach a file</p> <p><a href="https://training2013.formsboard.com/Lists/Test1/Attachments/2/2015090708394790954_11-Lab 4b.docx">https://training2013.formsboard.com/Lists/Test1/Attachments/2/2015090708394790954_11-Lab 4b.docx</a></p> <p>11-Lab 4b.docx</p> |

☒ Insert item

☐ Insert item

Submit To SharePoint List

SharePoint List:

|           |                              |           |                                                           |    |
|-----------|------------------------------|-----------|-----------------------------------------------------------|----|
| Home      | + new item or edit this list |           |                                                           |    |
| Documents | All Items                    | ...       | Find an item                                              |    |
| Recent    |                              |           |                                                           |    |
| Test1     | ✓                            | Title     | Information                                               | ID |
|           |                              | 11-Lab 4b | Creating an Infopath web part to infopath web part mashup | 2  |

To update the existing items in the SharePoint list from the InfoPath form, modify your values and re-submit.

## HOW TO USE THE REFRESHSHAREPOINTLISTITEMS COMMAND AND TEST

The **RefreshSharePointListItems** command can be used to refresh the SharePoint list items. This command has two modes: report or update. When used with the **SubmitToSharePointList** command, your form will be able to submit to a SharePoint list and be updated *from* the list.

This command also is what triggers qRules to keep track of deletes in the form. If you run Refresh, we create an in-memory list of the items currently populated with an ID (after Refresh removes anything that had been deleted from SharePoint) and then if you delete rows in your form that had previously been submitted, we run a delete on the next Submit command.

There are two possible data source options for this command. One is a data connection using owssvr.dll. The other uses a SharePoint List connection. Data connections are only used in preview or InfoPath Filler, or if you have set useWebServices to "true" in the QdabraRules data source. In a published browser form the SharePoint object model is used to return list item information to the form.



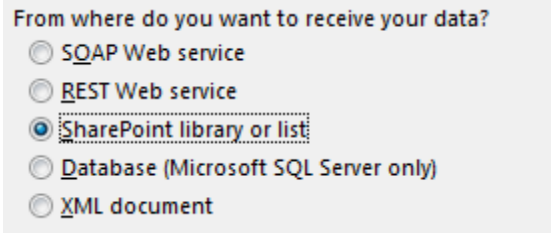
Both options are provided here for completeness:

---

### *Using a SharePoint List Data Connection to Refresh List Items*

#### ADD A RECEIVE CONNECTION TO A SHAREPOINT LIST

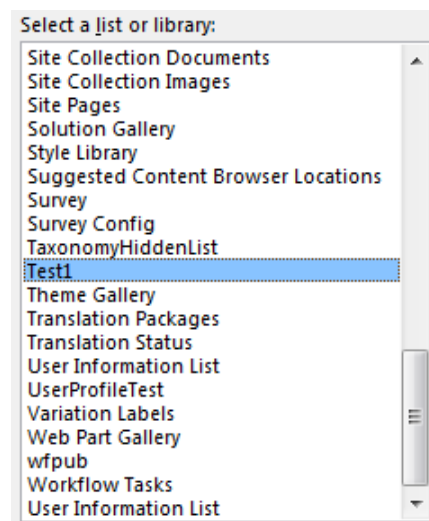
1. Add a new data connection to your form. This connection is a **Receive** data connection from a **SharePoint List**:



From where do you want to receive your data?

- ☐ SOAP Web service
- ☐ REST Web service
- ☒ SharePoint library or list
- ☐ Database (Microsoft SQL Server only)
- ☐ XML document

2. Select the list used in the previous section:



Select a list or library:

- Site Collection Documents
- Site Collection Images
- Site Pages
- Solution Gallery
- Style Library
- Suggested Content Browser Locations
- Survey
- Survey Config
- TaxonomyHiddenList
- Test1**
- Theme Gallery
- Translation Packages
- Translation Status
- User Information List
- UserProfileTest
- Variation Labels
- Web Part Gallery
- wfpub
- Workflow Tasks
- User Information List

3. Tick checkboxes for items to be retrieved:



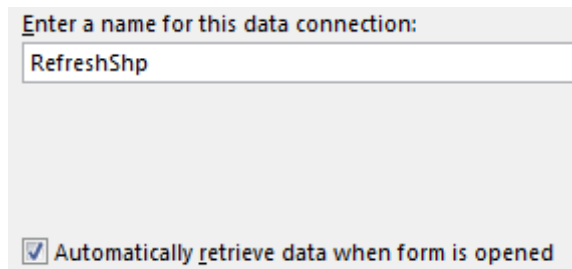
The SharePoint list has the following fields that you can select from.

Select fields:

- ☒ Title
- ☒ Information
- ☒ ID
- ☒ Modified
- ☐ Created
- ☐ Created\_By (Created By)
- ☐ Modified\_By (Modified By)
- ☐ Version
- ☐ Attachments

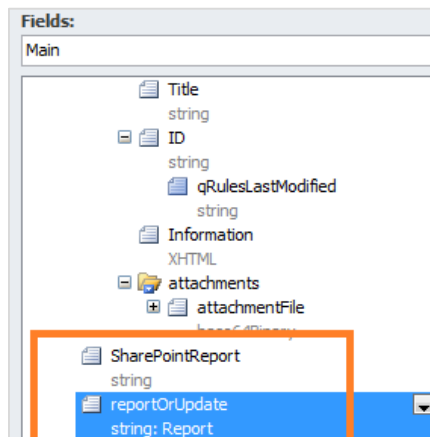


4. Click **Next** twice.
5. Name the data connection, and leave the checkbox ticked for loading the data on open:

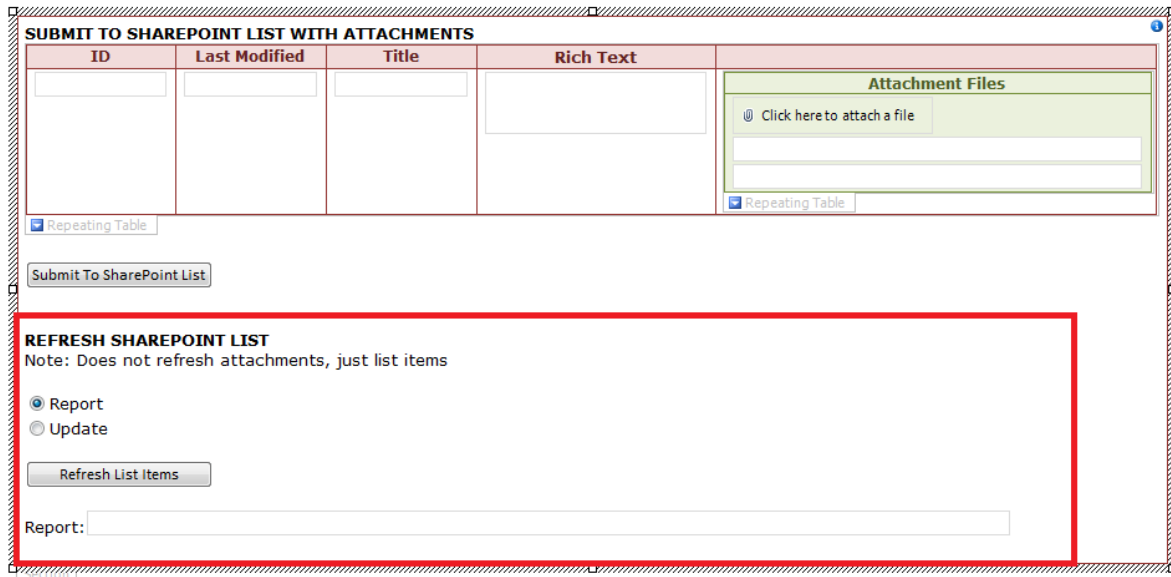


#### ADD THE REFRESHSHAREPOINTLISTITEMS COMMAND AND TEST

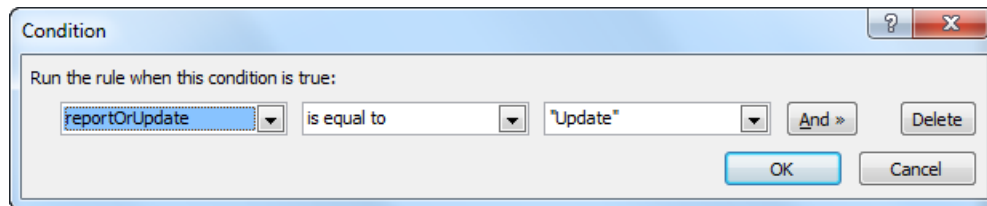
6. RefreshSharePointListItems requires an attribute called qRulesLastModified to be added to whatever field you are using for the id node in your SubmitToSharePoint list command (see step 20 in “How to use the SubmitToSharePointList command”). Right click on the field and add a string attribute called qRulesLastModified. When items are submitted to a SharePoint list via the SubmitToSharePointList command, this attribute will be populated with the date and time of the submit. When the RefreshSharePointListItems command is run, that date and time will be compared to the Modified field from the SharePoint list to determine whether the item should be updated.
7. For testing the different modes, add two new fields to your main data source as shown below:



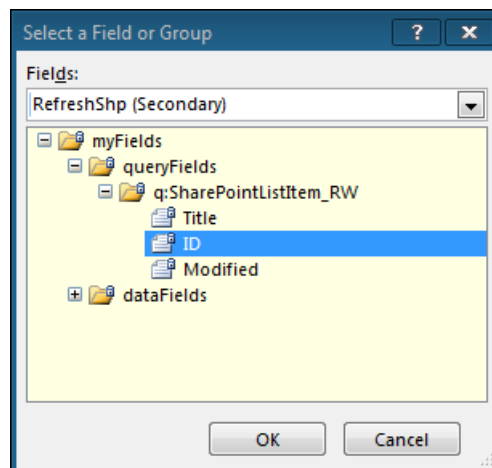

8. Add a text box bound to **SharePointReport**, and 2 option buttons bound to **reportOrUpdate**:



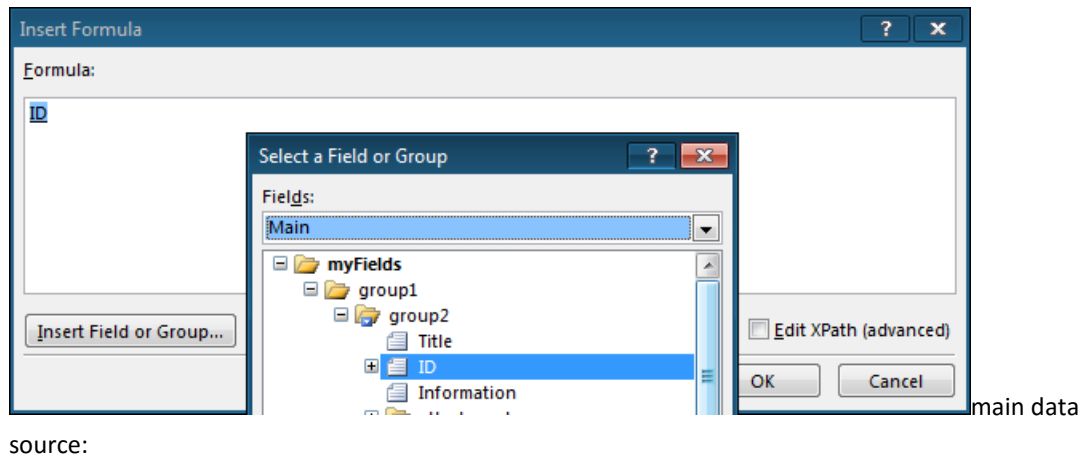
9. Label one option *Report* and the other *Update*.  
 10. Set the values on your option buttons to match the labels.  
 11. Create a rule on your **RefreshListItems** button. Set a condition to run this rule if **reportOrUpdate** is equal to **Update**:



12. For the first action of the rule, set the target field to the ID node of the **RefreshShp** data connection's **queryFields**:




13. Next, click the **fx** button > Insert Field or Group > select the ID field from the

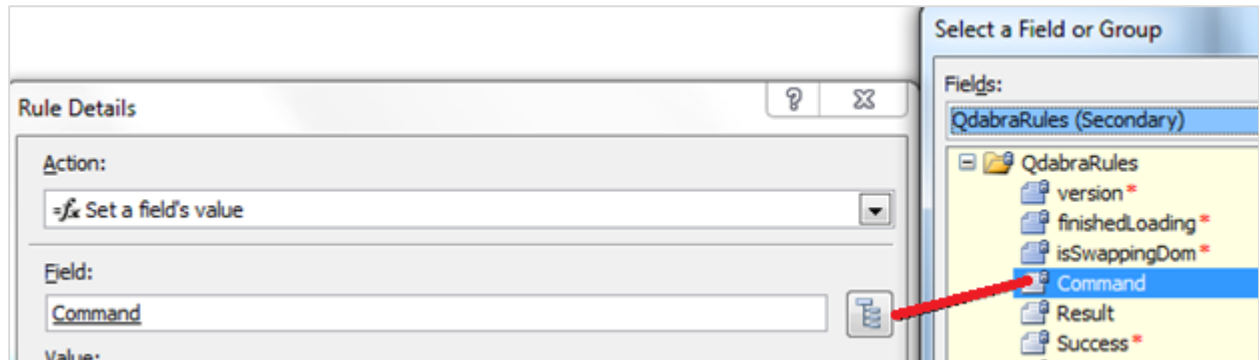


14. The action of the rule should set the **Command** field from the **QdabraRules** data source to the command: **RefreshSharePointListItems /dsname=RefreshShp /mode=Update /xpathsrc=/dfs:myFields/dfs:dataFields/d:SharePointListItem\_RW /mappingName=Test1**

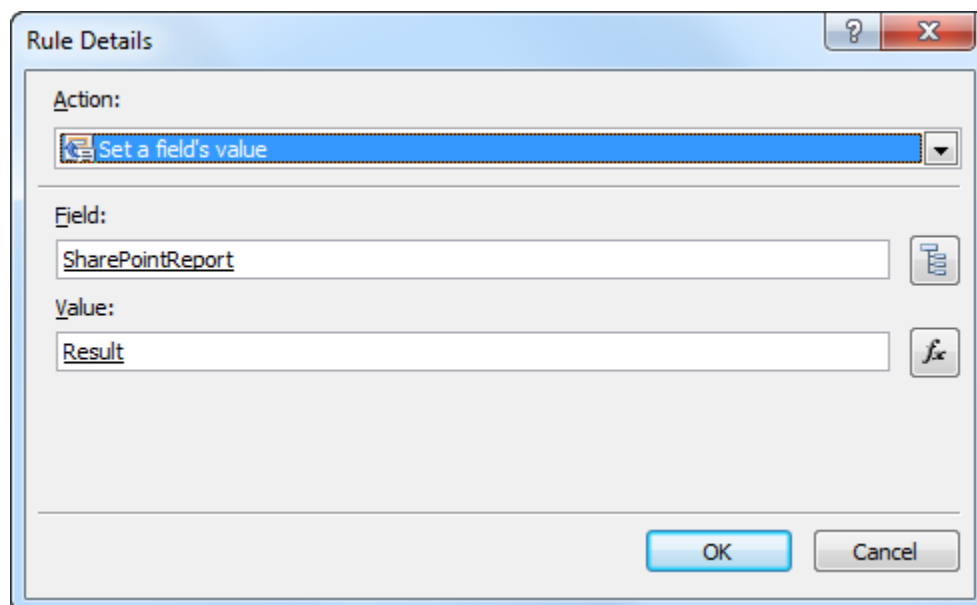
Breaking the command out:

- i. /dsname = The **RefreshShp** data connection we just set up
- ii. /mapping = The same mapping data connection used in the SubmitToSharePointList command (there is no need to declare this parameter if the Mapping Receive connection is named **mapping** the same mapping data connection used in the SubmitToSharePointList command)
- iii. /mode = (optional) Update|Report Indicates whether we want to update our list items rather than simply report on which items are different. Report is the default.
- iv. /xpathsrc = The path to the repeating node in your SharePoint list where you have your list items you want to use for refreshing your form. The form will need to query for data prior to executing the Refresh command if using this option, as the command will not execute the query. We left the data connection to automatically retrieve data in step 5.
- v. /mappingName = The mapping to refresh. This option is required if your mapping contains several lists, otherwise the form will not be able to identify which data to update.

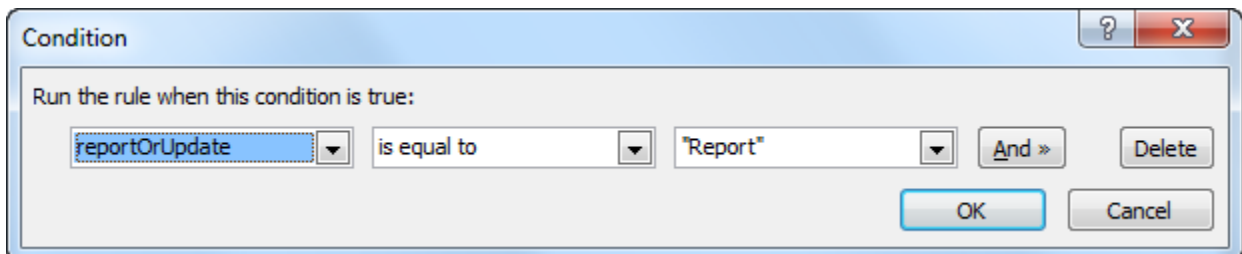




15. Create another action on the same rule to populate the **SharePointReport** field in the main data source with the value from the **Qdabra Rules** data source **Result** field:



16. Create another rule on the **Refresh List Items** button. Set a condition to run this rule if **reportOrUpdate** is equal to **Report**:



17. Again, set the target field to the ID node of the **RefreshShp** data connection's **queryFields** to the value of the ID field from the main data source.
18. The next action of the rule should set the **Command** field from the **QdabraRules** data source to the command:



**RefreshSharePointListItems /dsname=RefreshShp /mode=Report  
/xpathsrc=/dfs:myFields/dfs:dataFields/d:SharePointListItem\_RW /mappingName=Test1.**

Since the default for /mode is Report, you may also choose to not include that parameter.

19. Create another action on the same rule to populate the **SharePointReport** field in the main data source with the value from the **Qdabra Rules** data source **Result** field.
20. Save and preview the form.
21. Enter the id for an existing SharePoint list item id in the **ID** field, and select **Update** radio button, and select **Refresh List Items** button (Note: Attachments are not refreshed). The mapped fields should populate for your list item. Experiment with the command by modifying the list item in SharePoint, selecting the "Report" option and running the command again.

**SUBMIT TO SHAREPOINT LIST WITH ATTACHMENTS**

| ID | Last Modified       | Title          | Rich Text                         | Attachment Files                                   |
|----|---------------------|----------------|-----------------------------------|----------------------------------------------------|
| 3  | 2015-09-14 07:04:01 | Lab 1 - Dashbo | Create a dashboard using InfoPath | <div>Click here to attach a file</div> <div></div> |

☒ Insert item

Submit To SharePoint List

**REFRESH SHAREPOINT LIST**  
 Note: Does not refresh attachments, just list items
 

☐ Report  
☒ Update

Refresh List Items

Report: Report for mapping: Test1. Item 3 does not match.

### Using an Owssvr.dll Data Connection to Refresh List Items

ADD A RECEIVE CONNECTION TO THE LIST USING OWSSVR.DLL

1. Add a new data connection to your form. This connection is a receive data connection, and the source is an XML file:

This wizard helps you specify a data connectio

☒ Create a new connection to:  
☐ Submit data  
☒ Receive data

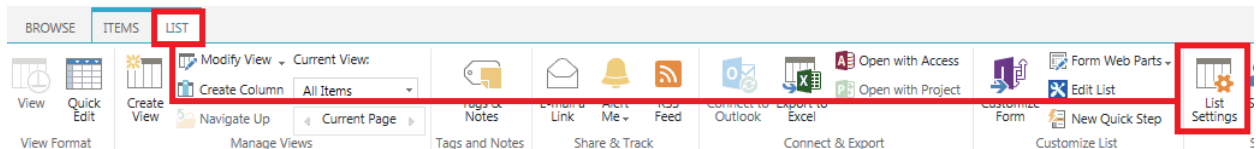


From where do you want to receive your data?

☐ SOAP Web service  
☐ SharePoint library or list  
☐ Database (Microsoft SQL Server or Microsoft Access only)  
☒ XML document

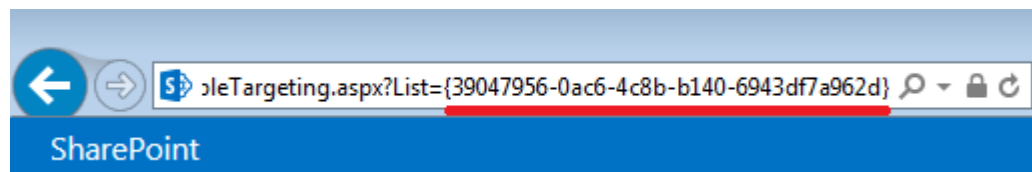
2. The location is the fun bit. To obtain the GUID for our list:

i. Open your SharePoint list in a browser, and proceed to **List tab > List Settings**:



ii. Under *General Settings*, click **Audience targeting Settings**.

iii. Everything past the equal sign in the URL is the list GUID, URL encoded:



We need that for the location of our XML file. An owssvr.dll link is in the following format:

<http://<serverName>/vti/bin/owssvr.dll?Cmd=Display&List={39047956-0ac6-4c8b-b140-6943df7a962d}&XMLDATA=TRUE&NOREDIREC=TRUE>

You need to change <serverName> to the name of your server. Change the GUID in the curly braces to the GUID to your list. If you want to test the link in the browser to see if XML is returned from your list, you can.

**For SharePoint 2007**, you may need to remove &NOREDIREC=TRUE for the XML to render.

Specify location of our XML file:

XML data file details

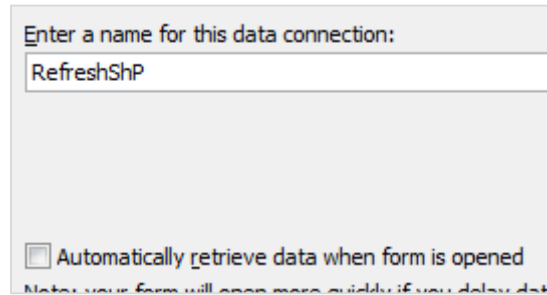
Enter the location of the XML data file that you want to use as your data connection:

3. Click **Next** and select the option to **Access the data from the specified location**:

☐ Include the data as a resource file in the form template or template part  
☒ Access the data from the specified location  
☐ Store a copy of the data for offline use

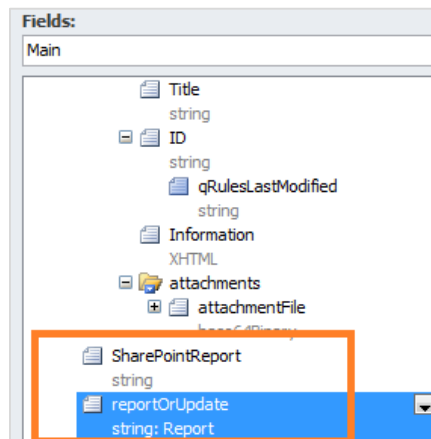


4. Name the data connection, and deselect the checkbox for loading the data on open:



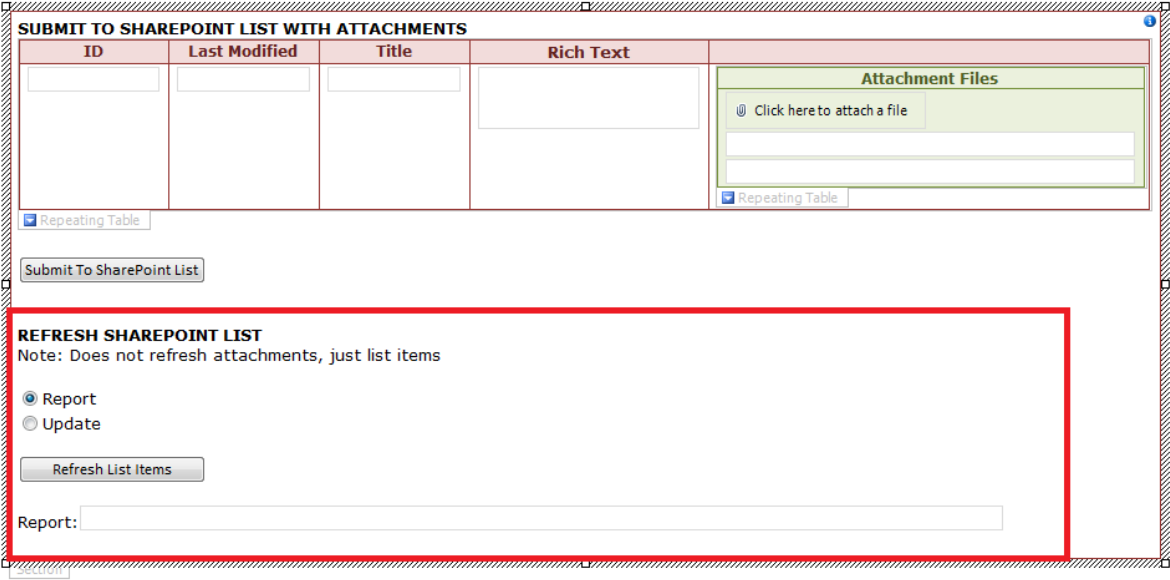
#### ADD THE REFRESHSHAREPOINTLISTITEMS COMMAND AND TEST

5. RefreshSharePointListItems requires an attribute called qRulesLastModified to be added to whatever field you are using for the id node in your SubmitToSharePoint list command (see step 20 in “How to use the SubmitToSharePointList command”). Right click on the field and add a string attribute called qRulesLastModified. When items are submitted to a SharePoint list via the SubmitToSharePointList command, this attribute will be populated with the date and time of the submit. When the RefreshSharePointListItems command is run, that date and time will be compared to the Modified field from the SharePoint list to determine whether the item should be updated.
6. For testing the different modes, add two new fields to your main data source as shown below:



7. Add a text box bound to **SharePointReport**, and 2 option buttons bound to **reportOrUpdate**:





**SUBMIT TO SHAREPOINT LIST WITH ATTACHMENTS**

| ID | Last Modified | Title | Rich Text | Attachment Files                               |
|----|---------------|-------|-----------|------------------------------------------------|
|    |               |       |           | Click here to attach a file<br>Repeating Table |

Submit To SharePoint List

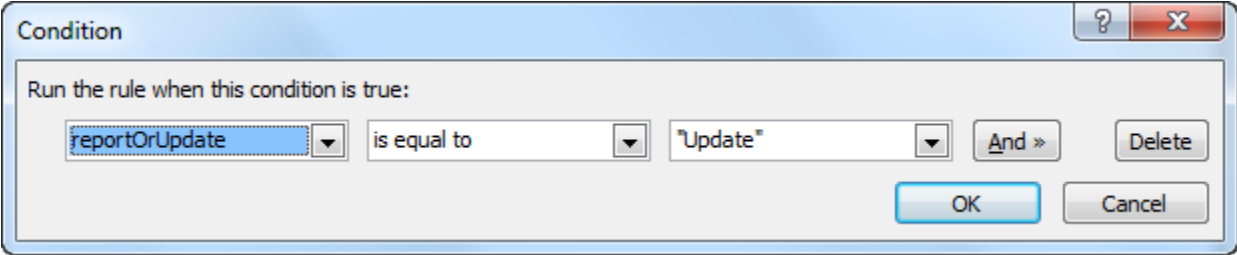
**REFRESH SHAREPOINT LIST**  
 Note: Does not refresh attachments, just list items

☒ Report  
☐ Update

Refresh List Items

Report:

8. Label one option *Report* and the other *Update*.
9. Set the values on your option buttons to match the labels.
10. Create a rule on your **RefreshListItems** button. Set a condition to run this rule if **reportOrUpdate** is equal to **Update**:



Condition

Run the rule when this condition is true:

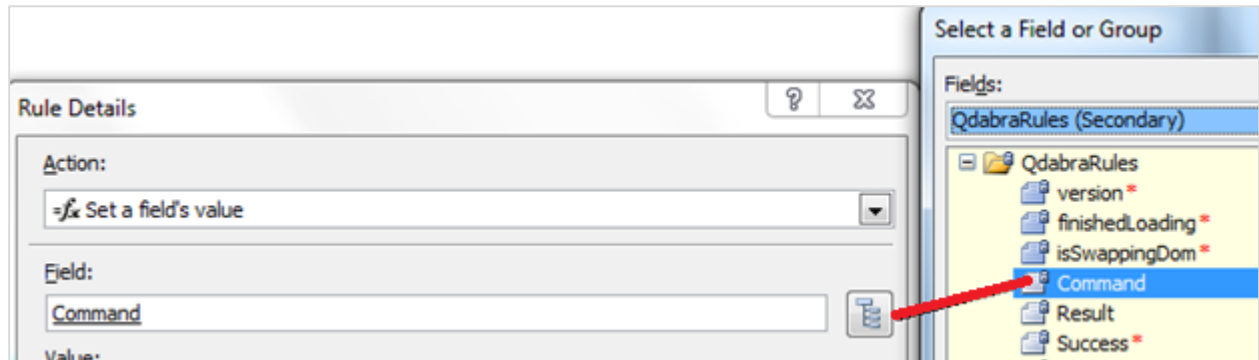
reportOrUpdate is equal to "Update"

And >> Delete

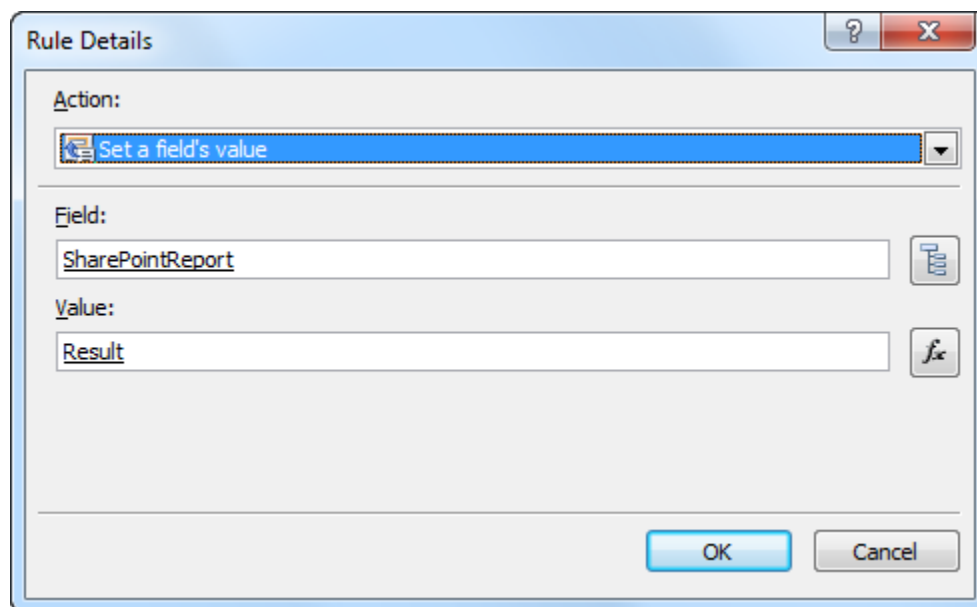
OK Cancel

11. The action of the rule should set the **Command** field from the **QdabraRules** data source to the command: **RefreshSharePointListItems /dsname=RefreshShp /mode=Update**.  
 Breaking the command out:
  - i. /dsname = The **RefreshShp** data connection we just set up
  - ii. /mapping = The same mapping data connection used in the SubmitToSharePointList command (there is no need to declare this parameter if the Mapping Receive connection is named **mapping** the same mapping data connection used in the SubmitToSharePointList command)
  - iii. /mode = OPTIONAL Update|Report Indicates whether we want to update our list items rather than simply report on which items are different. Report is the default

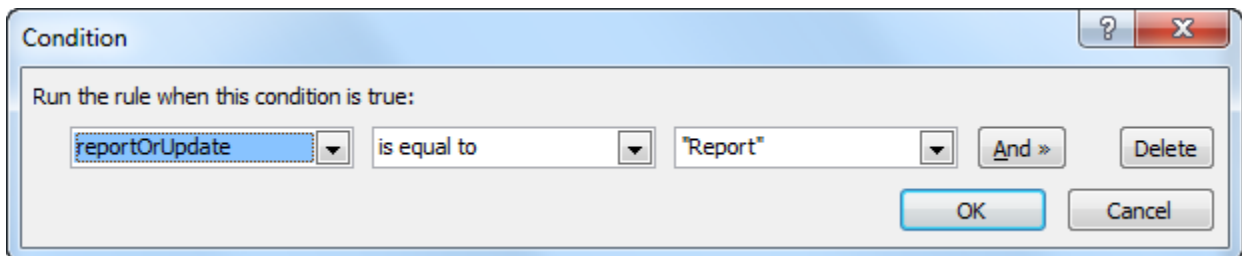




12. Create another action on the same rule to populate the **SharePointReport** field in the main data source with the value from the **Qdabra Rules** data source **Result** field:



13. Create another rule on the **Refresh List Items** button. Set a condition to run this rule if **reportOrUpdate** is equal to **Report**:



14. The action of the rule should set the **Command** field from the **QdabraRules** data source to the command: **RefreshSharePointListItems /dsname=RefreshShp /mode=Report**. /dsname is the data connection we just set up, /mapping is not declared since the mapping data connection is named mapping —(the same mapping data connection used in the SubmitToSharePointList command) and



- /mode indicates we want to report on our list items rather than update. Since the default for /mode is Report, you may also choose to not include that parameter.
15. Create another action on the same rule to populate the **SharePointReport** field in the main data source with the value from the **Qdabra Rules** data source **Result** field.
  16. Save and preview the form.
  17. Submit a new entry using the **Submit To SharePoint List** button and save the xml file (under File tab, click Save As to save the xml).
  18. Open the xml (either by clicking the xml file or specifying the sample at the Form Option's Preview category – browse for the saved xml).
  19. Once the xml is opened, delete the Title, Information, qRulesLink and qRulesFilename. Select the **Update** radio button, and select **Refresh List Items** button (Note: Attachments are not refreshed). The mapped fields should populate for your list item. Experiment with the command by modifying the list item in SharePoint, selecting the "Report" option and running the command again.

| SUBMIT TO SHAREPOINT LIST WITH ATTACHMENTS |                     |                 |                                   |                                                    |
|--------------------------------------------|---------------------|-----------------|-----------------------------------|----------------------------------------------------|
| ID                                         | Last Modified       | Title           | Rich Text                         | Attachment Files                                   |
| 3                                          | 2015-09-14 07:04:01 | Lab 1 - Dashboz | Create a dashboard using InfoPath | <div>Click here to attach a file</div> <div></div> |

☒ Insert item

☐ Insert item

**REFRESH SHAREPOINT LIST**  
Note: Does not refresh attachments, just list items

☐ Report  
☒ Update

Report: Report for mapping: Test1. Item 3 does not match.

How to use the AttachToSharePointList command

## HOW TO USE THE ATTACHTOSHAREPOINTLIST COMMAND

The **AttachToSharePointList** command is used to add attachment(s) to existing SharePoint list items. This command is also leveraged behind the scenes when certain parameter are included with a SubmitToSharePointList command, as noted in the section in this document on that command.

In order to use this command, you must know the list item Id of the item you want to save your attachment to. For example, you might have a form that has a drop down control with a list of job candidates. The data source for the drop down might be a list in SharePoint. Your drop down could use the job candidate name for the display, and the



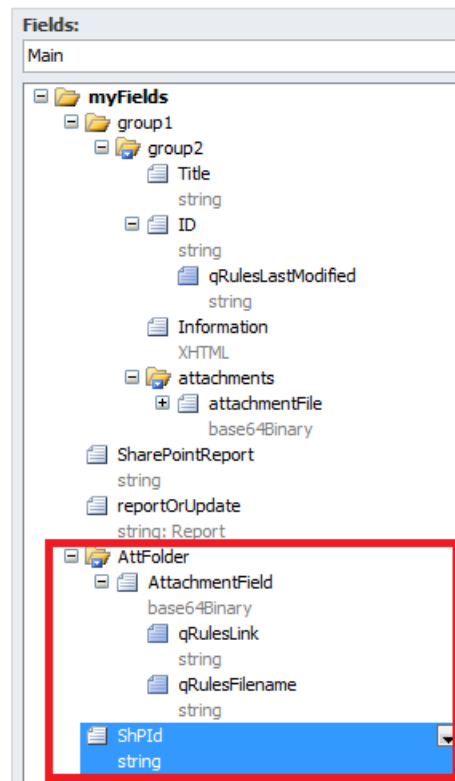
SharePoint list item ID for the value. Users would select a job candidate, attach a file to the form (a resume, for example) and click a button to add that attachment to the list item for that candidate.

### **Add Receive Connection to SharePoint Lists Web Service AddAttachment**

The data source must be a receive data connection to the Lists Web Service for your SharePoint site, and set to use the **AddAttachment** method. Please see [this](#) section for step-by-step instructions for adding this connection. Remember to use the correct GUID for the list you wish to submit your attachments to (this will be discussed later on as we progress).

### **Add Fields to your Form for Attachments**

1. Add a **ShPid** field and **AttFolder** group to the main data source. The **AttFolder** group should look similar to the one below:

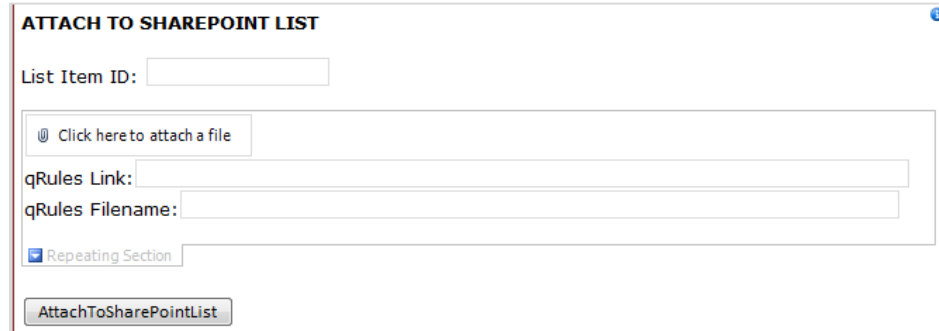


2. Of particular note are the attributes "qRulesLink" and "qRulesFilename". If these attributes exist for the attachment node, after the attachment is saved to the list item, the attachment node will be cleared, and the qRulesLink and qRulesFilename fields will be populated with the link and filename, respectively, for the new list item attachment.



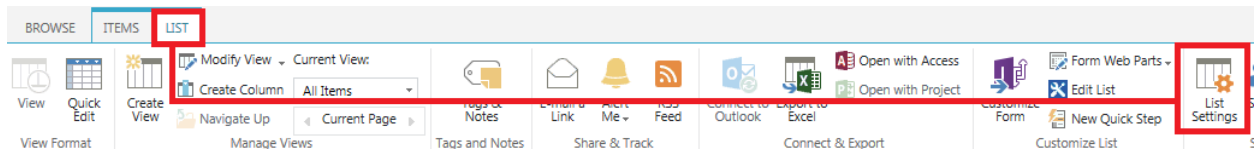
This means you can cut down on the size of your final XML by effectively removing files from your form and submitting them to SharePoint. With a little creative conditional formatting, your user will have the link available for easy file access, and the file attachment field hidden.

3. Drag and drop the newly created fields onto the canvas, and add a button for the qRules command:

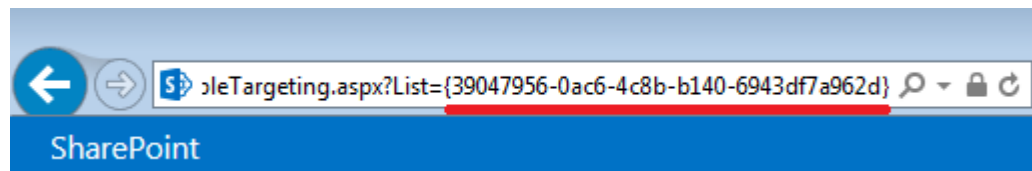


### Add the AttachToSharePointList Command and Test

4. Obtain the GUID for our list:
  - a. Open your SharePoint list in a browser, and proceed to **List tab > List Settings**:

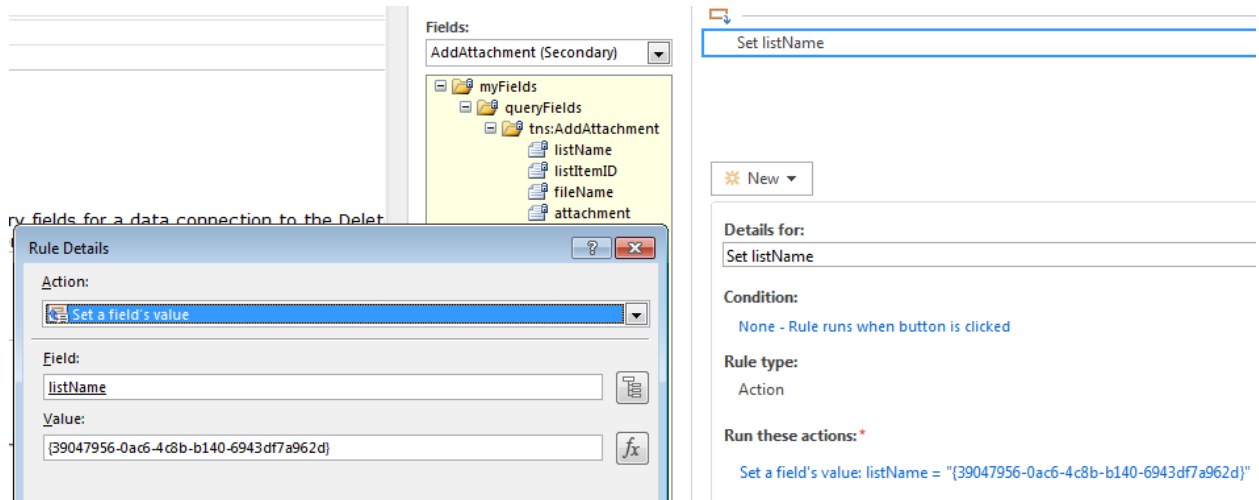


- b. Under *General Settings*, click **Audience targeting Settings**.
  - c. Everything past the equal sign in the URL is the list GUID, URL encoded:

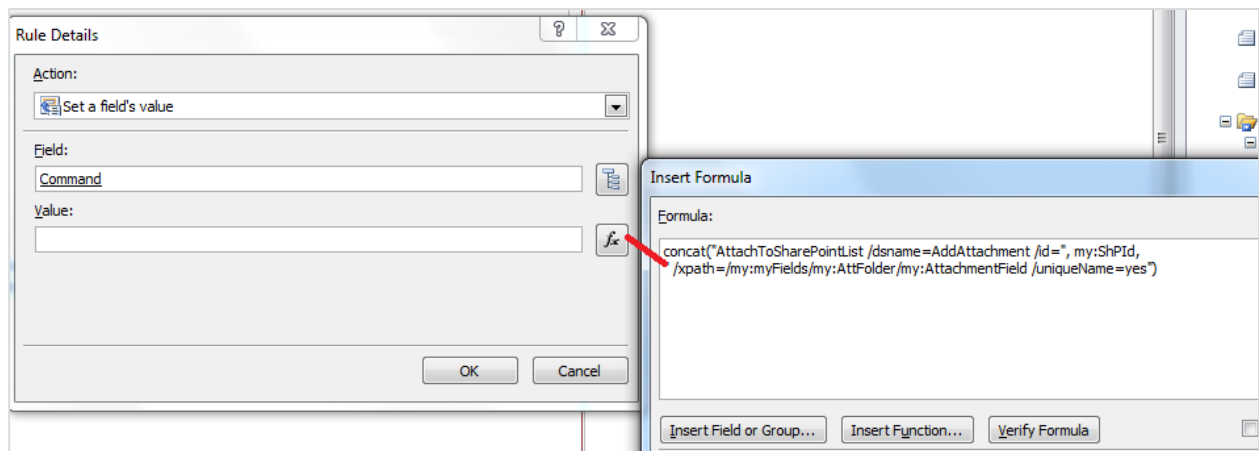


5. Add a rule to set the ListName of the AddAttachment data connection to the target GUID:





6. Add another rule to the **AttachToSharePointList** button to set the **Command** field in the **QdabraRules** data source.
7. Set the **Value** to `concat("AttachToSharePointList /dsname=AddAttachment /id=", ShPIId, " /xpath=/my:myFields/my:AttFolder/my:AttachmentField /uniqueName=yes")`. Since we are using a "concat" function, we need to use the **Insert Formula** dialog:



Breaking the command out:

- /dsname = The data connection to **AddAttachment** that we just created
- /id = The SharePoint List item id of the item we want to attach to
- /xpath = The full XPath to the field for our attachments. May be repeating.
- /uniqueName = (optional) yes|no Defaults to no – this parameter will prepend the file name you are saving with date and time information to create a unique name. This will prevent accidental attachment overwrites, but may not be desirable if a new attachment is added to the SharePoint List item multiple times.



8. Click **OK**, and save and preview the form.
9. Enter a list id in the **List Item ID** field (this tutorial uses **2**), and attach an attachment, and click the **AttachToSharePointList** button:

**ATTACH TO SHAREPOINT LIST**

List Item ID:

 Click here to attach a file

qRules Link:

qRules Filename:

☒ Insert item

**Attach to List Item**

10. Open the list from a SharePoint site, and you will see that your file has been added to the list item specified:

|                                                                                                                                              |                                                                                                                                                              |
|----------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Title                                                                                                                                        | 11-Lab 4bA                                                                                                                                                   |
| Information                                                                                                                                  | Creating an Infopath web part to infopath web part mashup                                                                                                    |
| Attachments                                                                                                                                  | <a href="#">2015090708394790954_11-Lab 4b.docx</a><br><a href="#">2015090711133935253_11-Lab 3b.docx</a><br><a href="#">2015090711154022644_Courses.xlsx</a> |
| Created at 9/6/2015 5:39 PM by <input type="checkbox"/> dlambatin<br>Last modified at 9/6/2015 8:15 PM by <input type="checkbox"/> dlambatin |                                                                                                                                                              |
| <input type="button" value="Close"/>                                                                                                         |                                                                                                                                                              |

## HOW TO USE THE SHAREPOINT DELETEATTACHMENT OPERATION

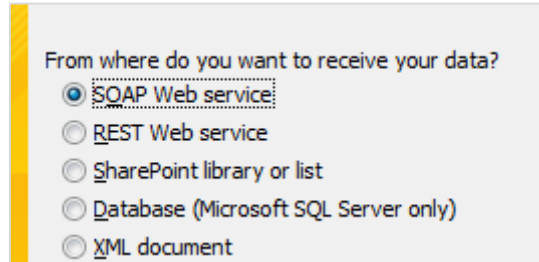
The **DeleteAttachment** operation deletes an attachment from a SharePoint list item. Now that we can add attachments to list items, there may be times we want to delete them. For example, perhaps your user wants to delete a file from a form, and the file has been added to a list item and replaced with a link... Your form could provide the user with a control (like a button or a checkbox) that will delete the linked attachment from the original SharePoint list item.

---

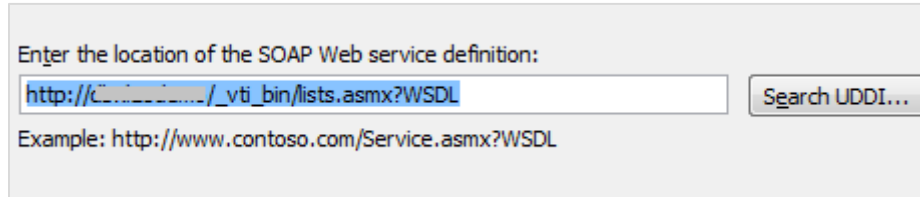
*Add a Receive Connection to SharePoint Lists Web Service DeleteAttachment*



1. Navigate to **Data > Data Connections > Add...**
2. In the first screen of the *Data Connection* wizard, select a new connection to receive data and click **Next**.
3. Select the option to receive data from a SOAP Web Service and click **Next**:



4. Enter the web service URL for your SharePoint server: [http://server/\\_vti\\_bin/lists.aspx?WSDL](http://server/_vti_bin/lists.aspx?WSDL), and click **Next**:



Select the **DeleteAttachment** operation, and click **Next**:

Select an operation:

- ApplyContentTypeToList
- DeleteContentTypeXmlDocument
- UpdateContentTypesXmlDocument
- UpdateContentTypeXmlDocument
- DeleteContentType
- UpdateContentType
- CreateContentType
- GetListContentType
- GetListContentTypes
- CheckInFile
- UndoCheckOut
- CheckOutFile
- DeleteAttachment**
- GetAttachmentCollection
- AddAttachment
- GetVersionCollection
- AddDiscussionBoardItem
- UpdateListItems
- GetListItemChangesSinceToken
- GetListItemChanges
- GetListItems
- GetListCollection

- a. Enter the listName (either the actual name or the GUID), and click **Next** twice:

Parameters:

| Parameter      | Type   | Value                |
|----------------|--------|----------------------|
| tns:listName   | string | 856F041E-33E8-46E... |
| tns:listItemID | string |                      |
| tns:url        | string |                      |

- b. Uncheck **Automatically retrieve data when form is opened** checkbox, and click **Finish**:

Enter a name for this data connection:

DeleteAttachment

☐ Automatically retrieve data when form is opened



### Add Controls and Test

5. Drag and drop the fields from **DeleteAttachment** data source onto the canvas and add a button for the web service query

**NON-~~q~~RULES BASED**

To delete an attachment, simply set the query fields for a data connection to the Delete Attachment SharePoint Web Service and query:

List Name:

List Item ID:

Url:

Section

6. Add a rule on the **Delete Attachment** button to **Query for data** and select the data connection we just created:

**Rule Details**

Action:

Data connection:

7. Click **OK**, save, and preview the form



8. To add an attachment to delete, enter a SharePoint list id (this tutorial uses **2**), and attach a new file to the attachment field, and click **AttachToSharePointList** button from the previous section:

### ATTACH TO SHAREPOINT LIST

List Item ID:

 Click here to attach a file

qRules Link:

qRules Filename:

☒ Insert item

**Attach to List Item**

9. Open the SharePoint list item to verify the newly attached file in the previous step:

|                                                                                                                                              |                                                                                                                                                              |
|----------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Title                                                                                                                                        | 11-Lab 4bA                                                                                                                                                   |
| Information                                                                                                                                  | Creating an Infopath web part to infopath web part mashup                                                                                                    |
| Attachments                                                                                                                                  | <a href="#">2015090708394790954_11-Lab 4b.docx</a><br><a href="#">2015090711133935253_11-Lab 3b.docx</a><br><a href="#">2015090711154022644_Courses.xlsx</a> |
| Created at 9/6/2015 5:39 PM by <input type="checkbox"/> dlambatin<br>Last modified at 9/6/2015 8:15 PM by <input type="checkbox"/> dlambatin |                                                                                                                                                              |
| <input type="button" value="Close"/>                                                                                                         |                                                                                                                                                              |

10. Now back in the form, enter the **List Item ID** (this tutorial uses **2**, and enter the attachment URL, and click **DeleteAttachment** button:

{39047956-0ac6-4

List Item ID:

Url:

**Delete Attachment**



11. Go back to the SharePoint list, and verify the attachment specified has been deleted:

|                                                                         |                                                                                                          |
|-------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|
| Title                                                                   | 11-Lab 4bA                                                                                               |
| Information                                                             | Creating an Infopath web part to infopath web part mashup                                                |
| Attachments                                                             | <a href="#">2015090708394790954_11-Lab 4b.docx</a><br><a href="#">2015090711133935253_11-Lab 3b.docx</a> |
| Created at 9/6/2015 5:39 PM by <input type="checkbox"/> dlambatin       |                                                                                                          |
| Last modified at 9/6/2015 9:24 PM by <input type="checkbox"/> dlambatin |                                                                                                          |
| <div>Close</div>                                                        |                                                                                                          |



## SUPPORT

If you have questions about the information in this document, please contact Qdabra Software for assistance.

Licensed customers can contact us via [Support@Qdabra.com](mailto:Support@Qdabra.com).

You can also use the [InfoPathDev.com](http://www.infopathdev.com) [Qdabra Product support forums](http://www.infopathdev.com) 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](mailto:Support@Qdabra.com)  
**Website:** <http://www.qdabra.com>  
**Community:** <http://www.infopathdev.com>



<http://www.qdabra.com>

Copyright © 2006-2015 Autonomy Systems, LLC. All rights reserved.