USING QUERYDOCUMENTNODESET TO AVOID DUPLICATE DATA

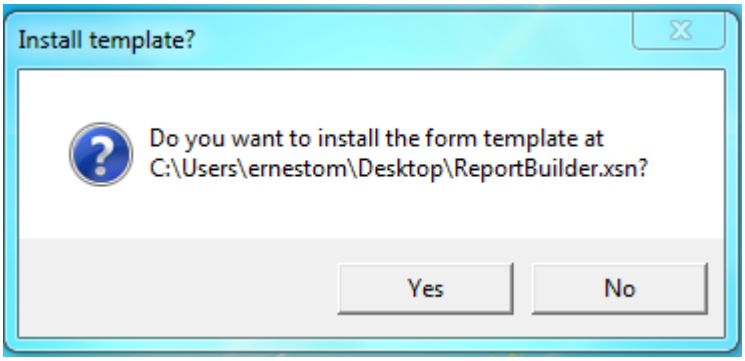
Though we have seen how DBLX can map data to a SQL database, Qdabra's Report Builder will allow you to build Excel reports that extract data from the xml files and not from SQL.

CREATE AN EXCEL REPORT USING REPORT BUILDER

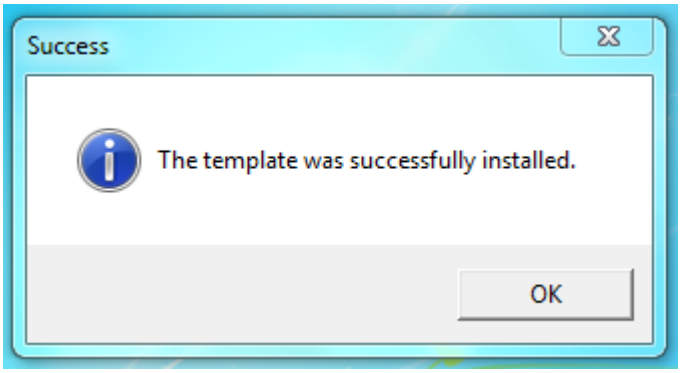
1. First make sure that InfoPath is not running (close all active InfoPath windows).

Note that sometimes, even after closing InfoPath, a process will remain active in the background. You can check this via the Task Manager.

2. Drag the ReportBuilder XSN onto the FormsInstaller icon. Click Yes when prompted.

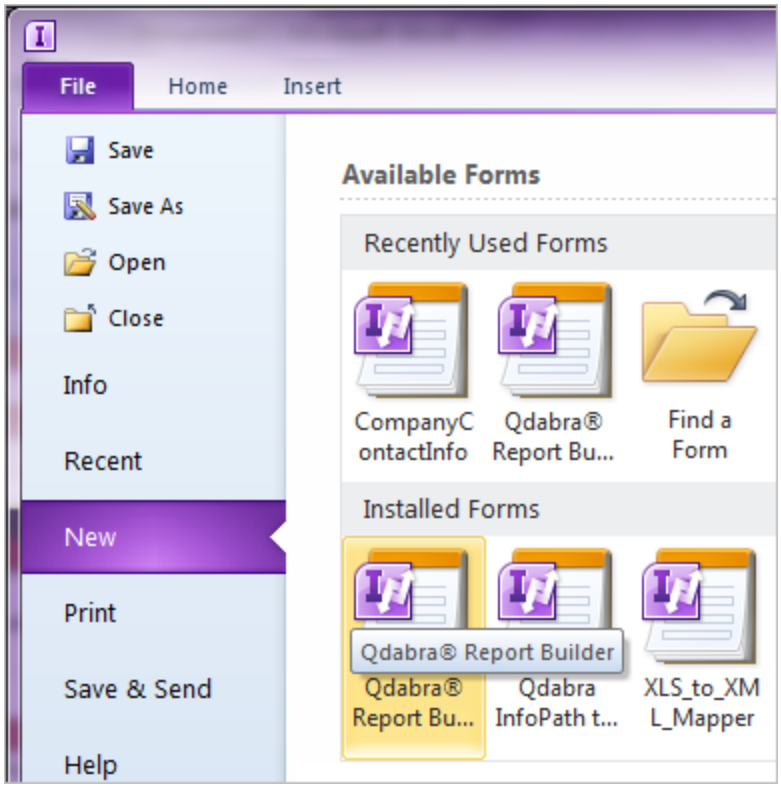


3. FormsInstaller will install the ReportBuilder form and form via a dialog.

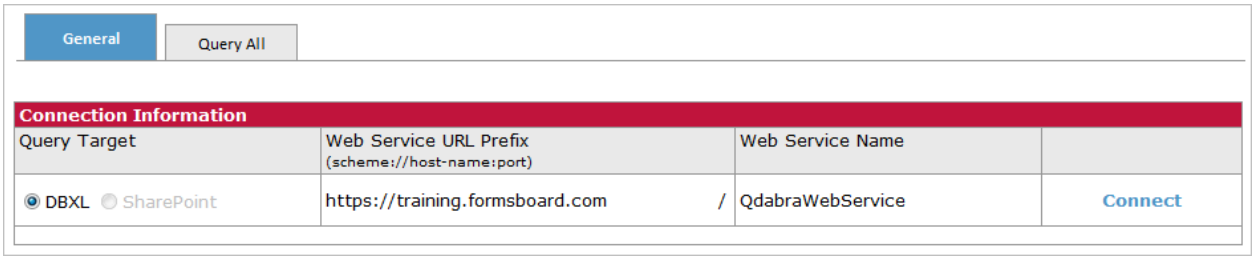


To launch the tool,

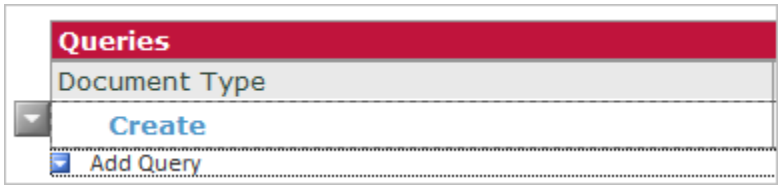
4. Open InfoPath Filler.
5. Click **New.**
6. Select **Qdabra Report Builder** listed under the **Installed Forms** section.



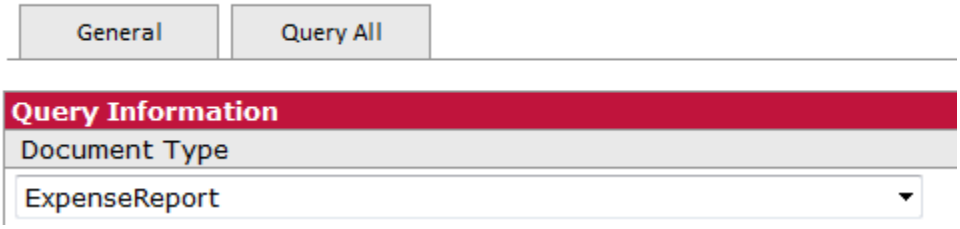
7. In the **General** section of the Report Builder tool, enter the Web Service URL Prefix, e.g. https://training.formsboard.com.
8. Enter **QdabraWebService** as the Web Service Name and then click **Connect**.



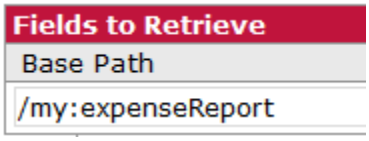
9. Start building your query by clicking **Add Query** > **Create** in the *Queries* table.



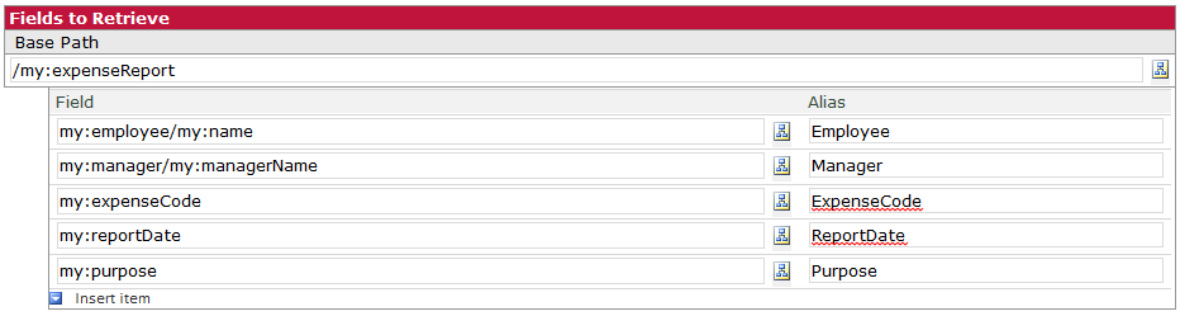
10. Select the Expense Report document type from the dropdown.



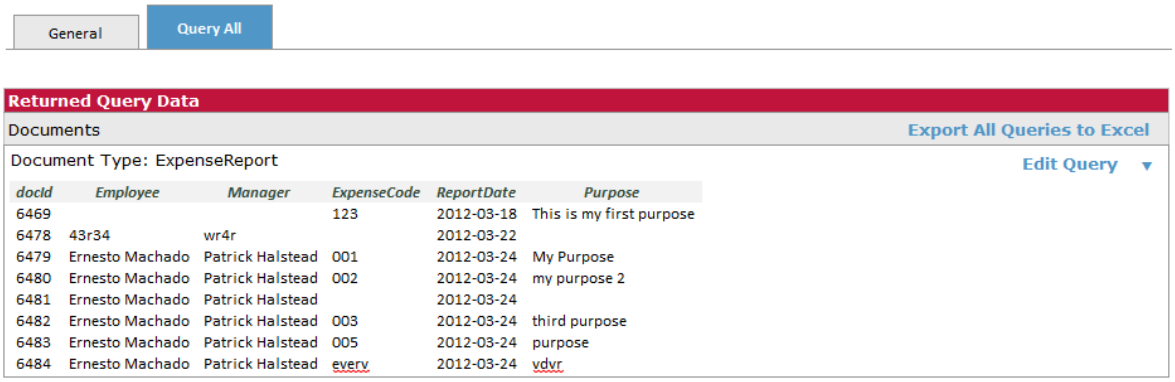
11. Under **Fields to Retrieve**, click on the Schema icon ( ). This will display the custom task pane at the right.
12. Double-click on the root node which in our sample XSN.



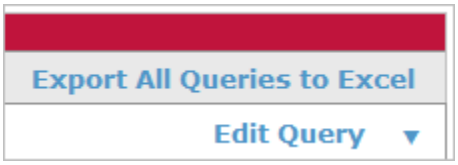
13. Under **Field**, click **Insert Item** and start adding the fields you wish to retrieve. Give each field an alias which will be the label/title in the Excel report for that particular field/column. Note that we can't use spaces in aliases.



14. Click **Perform Query** – you'll be taken to the *Query All* tab, click on the left arrow key (◄) to expand and see the query result, like this:



You can click on the **Export All Queries to Excel** should you wish to use the data in Excel and do reporting.



You have now retrieved your XML data into an Excel report!

	A	B	C	D	E	F
1	**docId**	**Employee**	**Manager**	**ExpenseCode**	**ReportDate**	**Purpose**
2	6469			123	2012-03-18	This is my first purpose
3	6478	43r34	wr4r		2012-03-22	
4	6479	Ernesto Machado	Patrick Halstead	001	2012-03-24	My Purpose
5	6480	Ernesto Machado	Patrick Halstead	002	2012-03-24	my purpose 2
6	6481	Ernesto Machado	Patrick Halstead		2012-03-24	
7	6482	Ernesto Machado	Patrick Halstead	003	2012-03-24	third purpose
8	6483	Ernesto Machado	Patrick Halstead	005	2012-03-24	purpose
9	6484	Ernesto Machado	Patrick Halstead	everv	2012-03-24	vdvr

AVOID DUPLICATES

Think back to the previous lab, when we called QueryDocumentNodeSet and passed in a query value of:

> *<query doctypeName="YOUR DOCUMENT TYPE"><columns repeatingPath="/my:expenseReport"><column name="my:employee/my:name" alias="Employee" /></columns></query>*

This query (and others) can be generated using Qdabra's Report Builder. That is, you can use QueryDocumentNodeSet to query data from xml documents belonging to your document type. In this section we will add rules to our form to call QueryDocumentNodeSet and avoid duplicates.

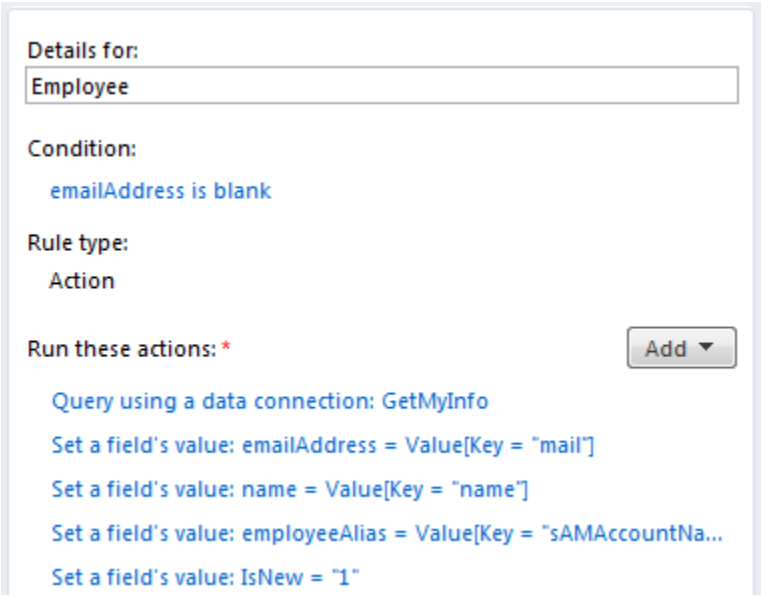
15. Using Notepad, create a file called FormOptions.xml, whose contents are just:

```
<FormOptions>
        <IsNew>0</IsNew>
        <Exists>0</Exists>
</FormOptions>
```

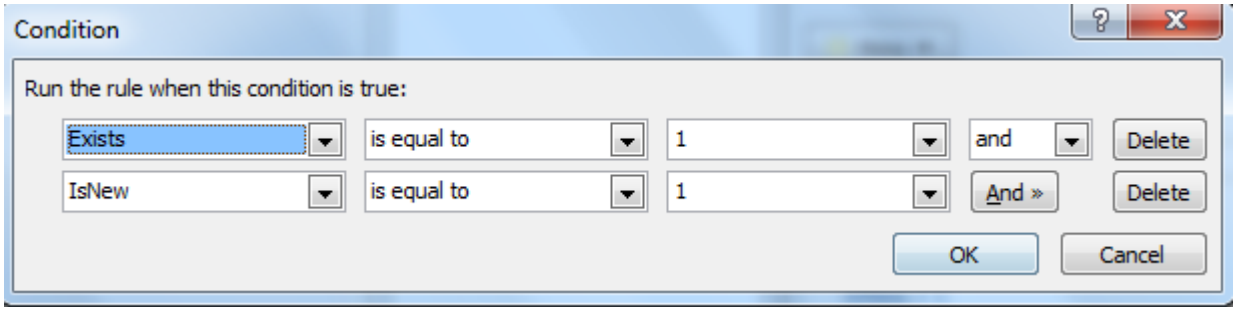
16. Open the Expense Report form in Design mode.
17. In **Data** > **Data Connections**, click **Add**.
18. Create a new data connection that Receives data from an XML document. Select the FormOptions.xml file you created in the previous step, and finish the Data Connection Wizard, making sure that the connection executes on load.

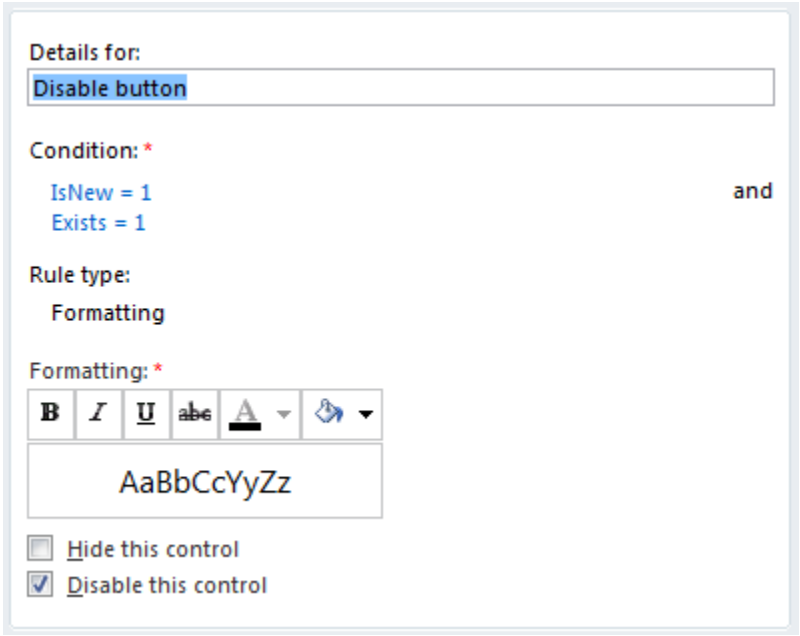
We will need these nodes later on.

19. In **Data** > **Form Load**, find the Employee rule. Add an action that sets the **IsNew** node in the **FormOptions** secondary data source, to 1.

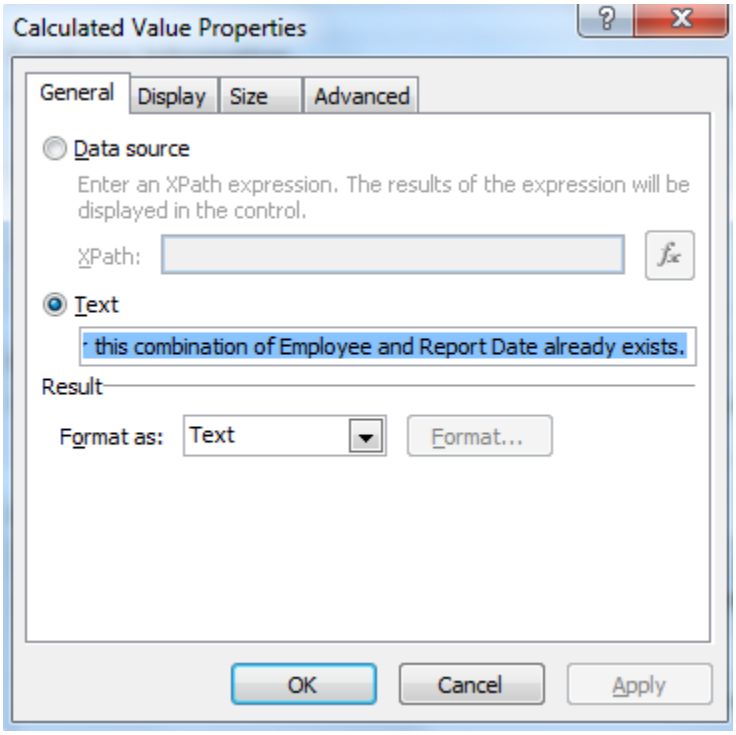


20. Select the **Submit** button.
21. Select **New** > **Formatting**.
22. For the condition, you'll want to check the **Exists** and **IsNew** fields in the **FormOptions** secondary data source. If they are equal to 1, we want to **Disable this control**.



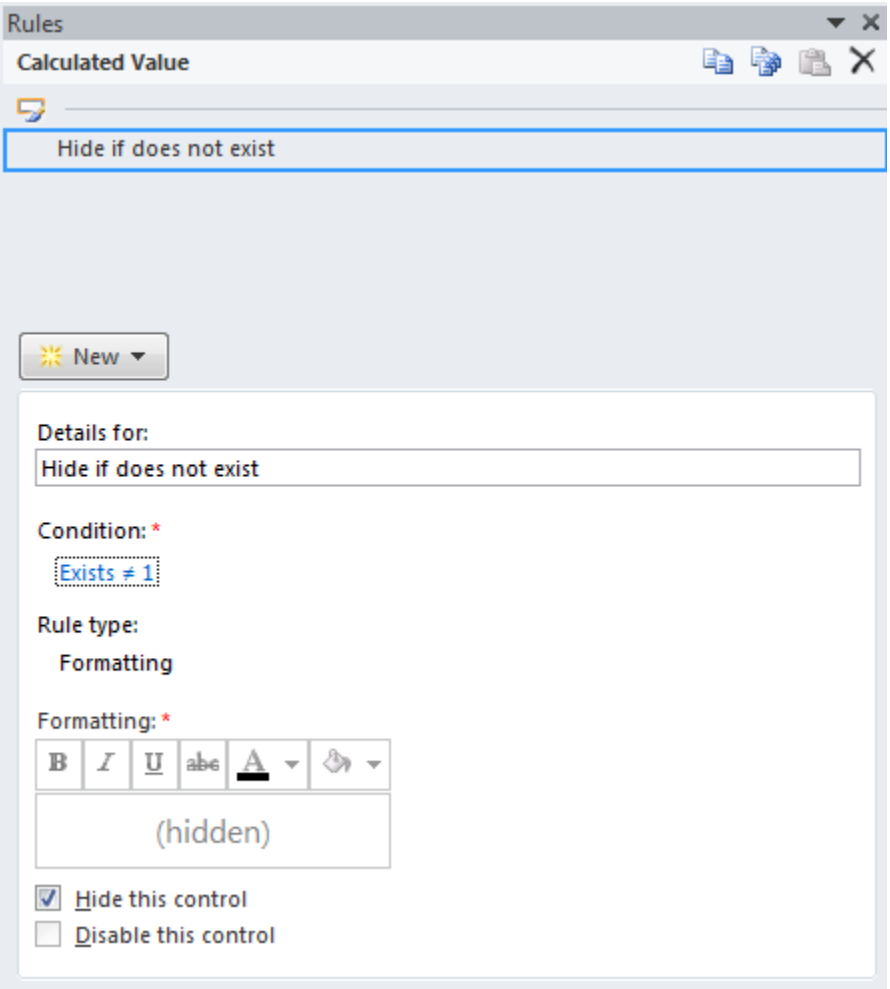


23. Add a Calculated value (Expression box) with the text: "A form for this combination of Employee and Report Date already exists."

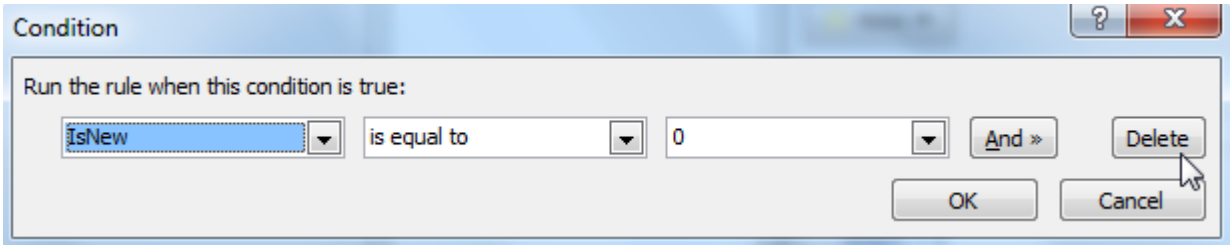


Now, for the Expression box:

24. Select **New** > **Formatting**.
25. For the condition, you'll want to check the **Exists** field in the **FormOptions** secondary data source. If it is not equal to 1, we want to **Hide this control**.



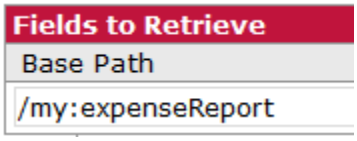
26. Add a second formatting, which will hide the Expression box and the link if the user has opened an existing record.



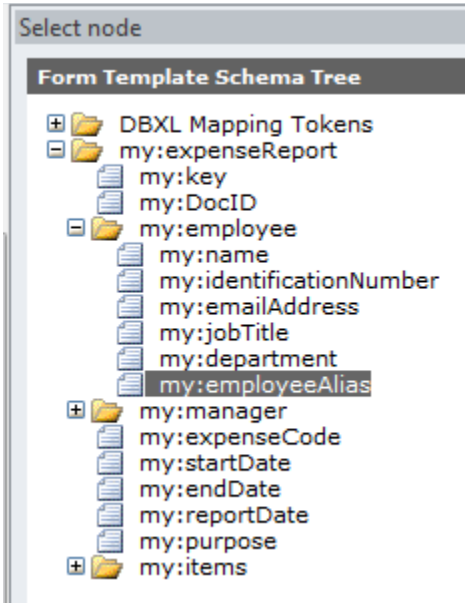
Before we proceed, we need to generate the queryXML via Report Builder.

27. Launch the Report Builder by opening the InfoPath 2010 Filler and selecting Report Builder from the Installed Forms section.

28. Enter the Web Service URL Prefix, the Web Service Name, and then click **Connect**.
29. Under **Queries**, click **Add query**, then click **Create**.
30. Select the ExpenseReport document type from the dropdown.
31. Under **Fields to Retrieve**, click the selector icon ( ).
32. In the taskpane, double click the base node.



33. Click **Insert item**.
34. Click the selector icon ( ) under **Field**.
35. In the taskpane, double click the node for the Employees alias.



36. Under **Alias**, enter EmployeeAlias. This is just a descriptive name for the retrieved field.
37. Under **Filter**, change the dropdown for **Type** to **Complex Filter**.
38. Click **Insert item** twice.

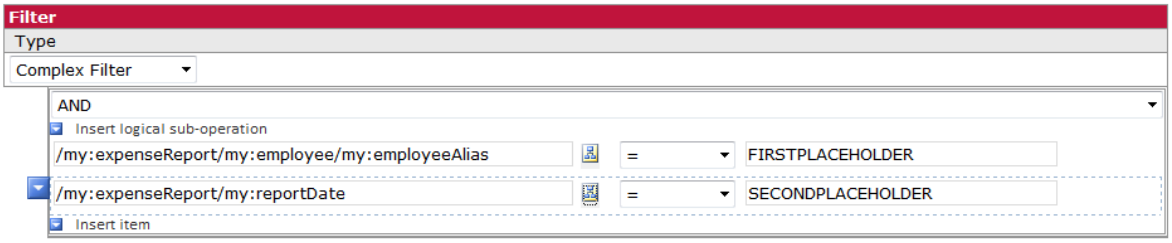
This inserts two rows for filtering. Let's set the first one.

39. Click the selector icon ( ) and double click the Employee alias field in the taskpane.
40. In the middle dropdown, select the equal sign.
41. In the right most field, type in FIRSTPLACEHOLDER.

Now for the second filter.

42. Click the selector icon ( ) and double click the reportDate field in the taskpane.
43. In the middle dropdown, select the equal sign.
44. In the right most field, type in SECONDPLACEHOLDER.

The filter should look like this:



You'll see that a query has been generated under QueryXML.

45. Copy the XML for query.

<query doctypeName="ExpenseReport"><columns repeatingPath="/my:expenseReport"><column name="my:employee/my:employeeAlias" alias="EmployeeAlias" /></columns><filter><and><eq><column name="/my:expenseReport/my:employee/my:employeeAlias" /><value>FIRSTPLACEHOLDER</value></eq><eq><column name="/my:expenseReport/my:reportDate" /><value>SECONDPLACEHOLDER</value></eq></and></filter></query>

Return to the form in design mode.

46. Go to **Data** > **Data connections** and click **Add**.
47. In the first screen of the Data Connection Wizard, select **Create a new data connection** to **Receive data**. Click **Next**.
48. Select **SOAP Web Service** and click **Next**.
49. Enter http://<servername>/QdabraWebService/DBXLDocumentService.asmx and click **Next**.
50. Select **QueryDocumentNodeSet** and click **Next**.

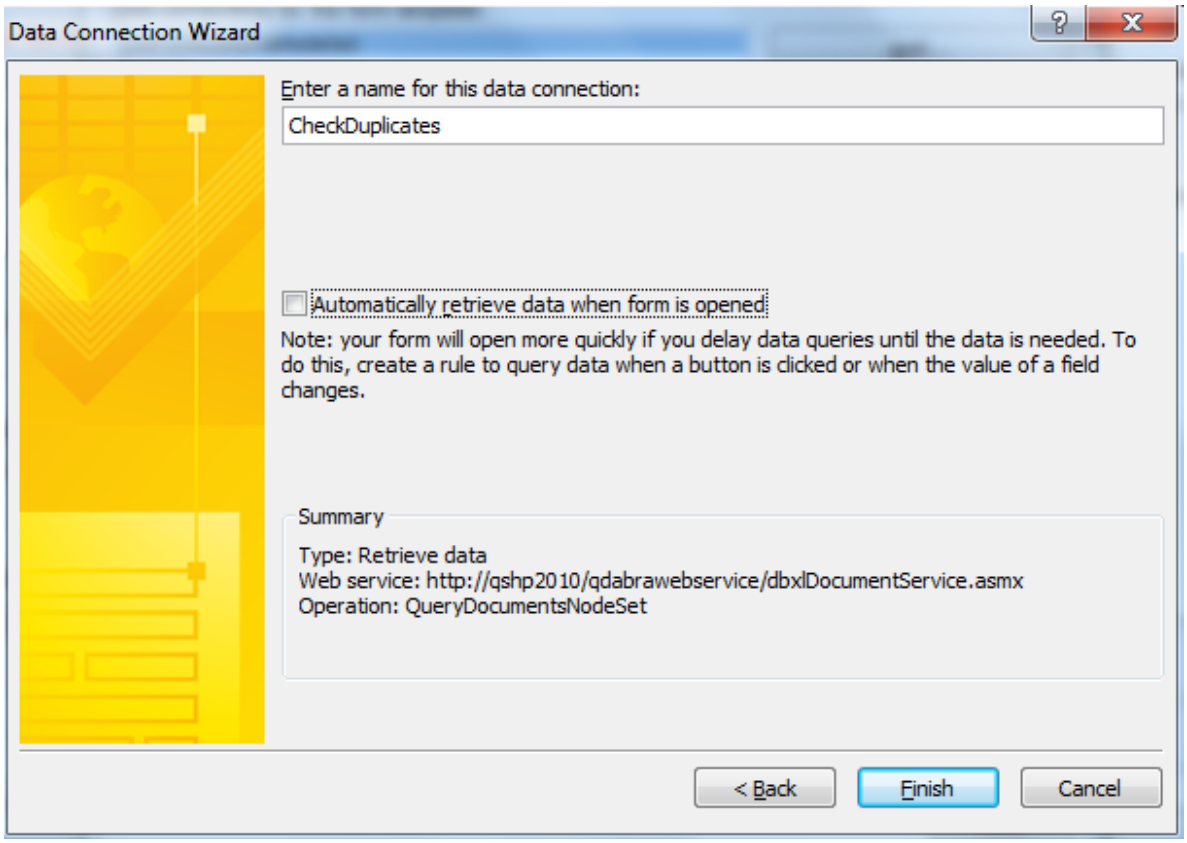
Now we need to use the query xml and we also need to remember the values from any document that was submitted, so that we can replace the placeholder text, for example:

<query doctypeName="ExpenseReport"><columns repeatingPath="/my:expenseReport"><column name="my:employee/my:employeeAlias" alias="EmployeeAlias" /></columns><filter><and><eq><column name="/my:expenseReport/my:employee/my:employeeAlias"

/><value>ErnestoM</value></eq><eq><column name="/my:expenseReport/my:reportDate"
/><value>2012-03-24</value></eq></and></filter></query>

(note that the alias is case sensitive)

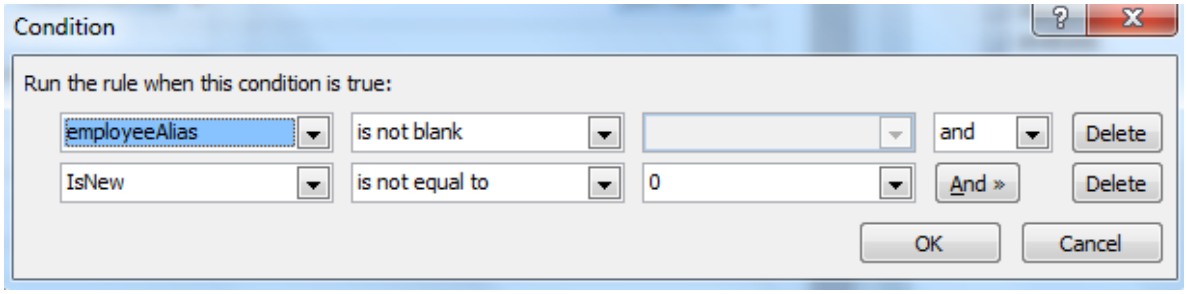
51. Enter the query, click OK and click **Next** twice.
52. Uncheck the option to retrieve the data on load, and click **Finish**.



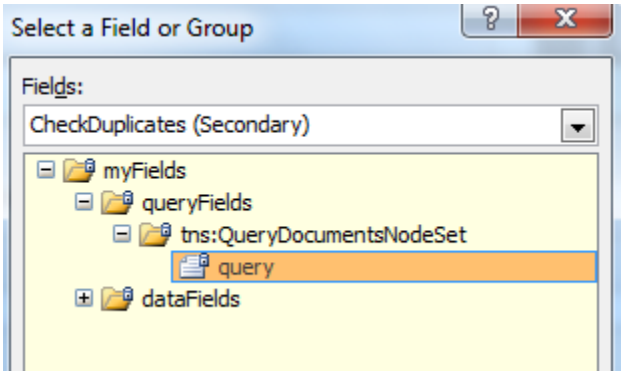
53. Call this data connection “CheckDuplicates”. Click **Close** on the Data Connections screen.

Now that we have the data connection, let’s use it on load.

54. Return to **Data** > **Form Load**.
55. Click **New** > **Action**. Call this “Check duplicates”.
56. Add a condition such that this rule will execute when **employeeAlias** is not blank. If we are opening an existing record, then we don’t need to check this, so add a condition for that as well.



57. Click on **Add** > **Set a field's value**.
58. For **Field**, select the query node from the **CheckDuplicates** data connection.



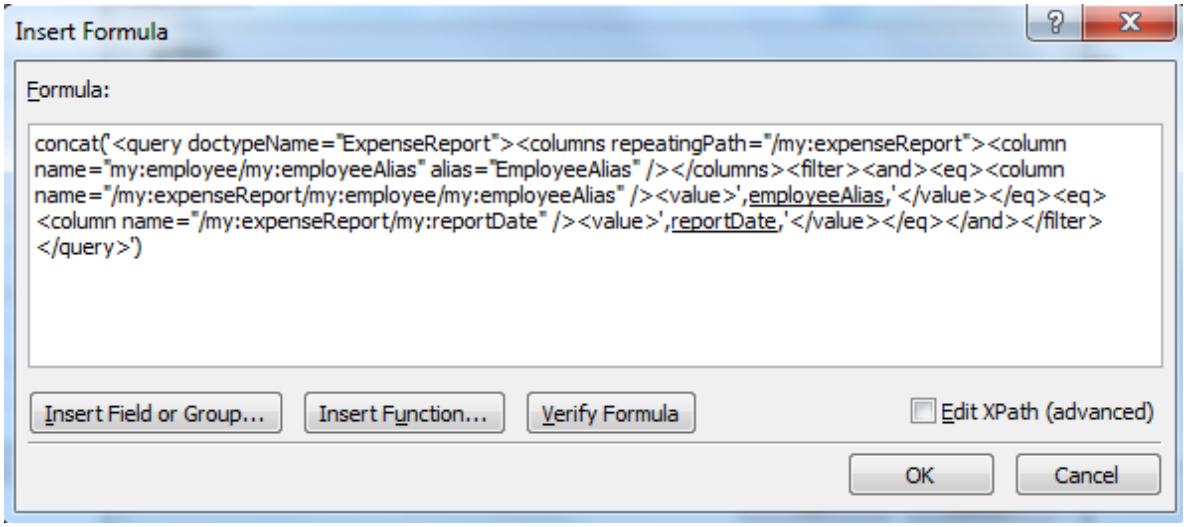
59. For **Value**, click **fx,** and enter:

concat('<query doctypeName="ExpenseReport"><columns repeatingPath="/my:expenseReport"><column name="my:employee/my:employeeAlias" alias="EmployeeAlias" /></columns><filter><and><eq><column name="/my:expenseReport/my:employee/my:employeeAlias" /><value>',,'</value></eq><eq><column name="/my:expenseReport/my:reportDate" /><value>',,'</value></eq></and></filter></query> ')

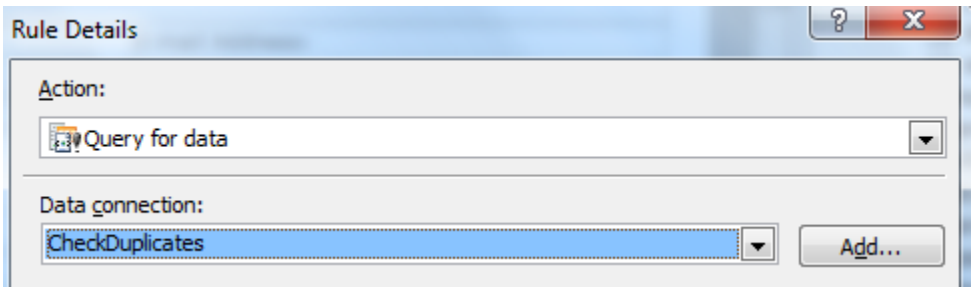
Where did this come from? This is the original query, generated by Report Builder, but we're using concat so that we can use the values in the form.

Notice the two locations where there are double commas.

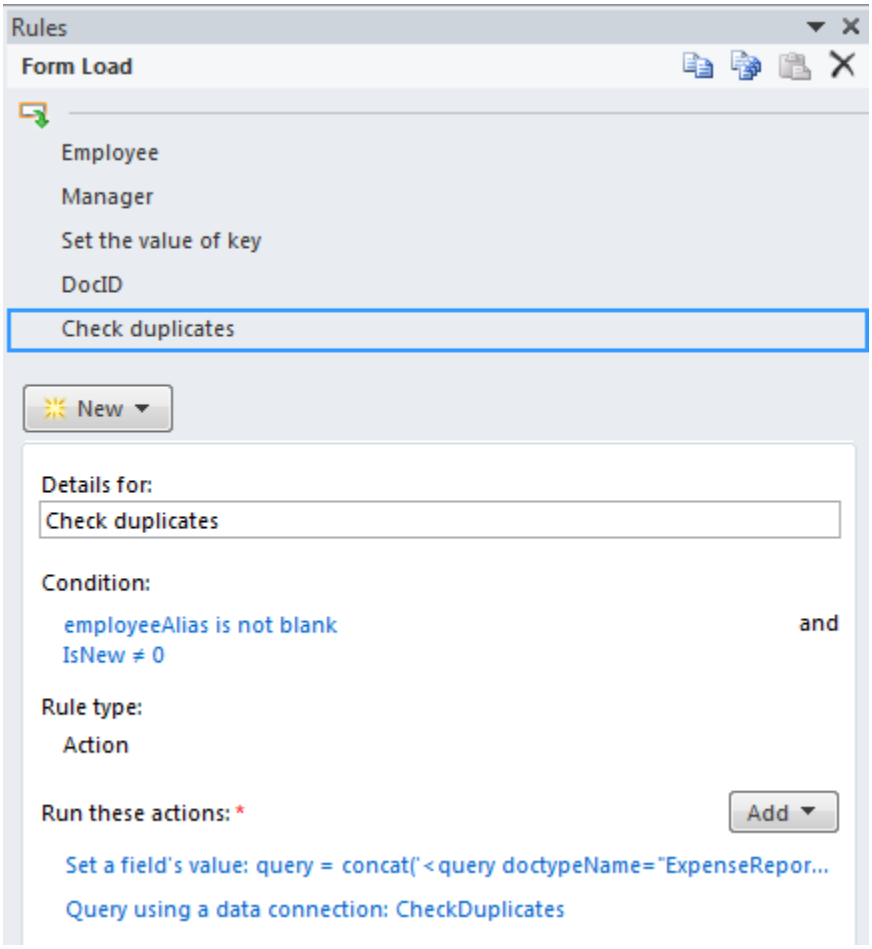
60. Place your cursor in between the commas and click **Insert field or group**, so that you can enter the employeeAlias and the ReportDate.



61. Click OK.
62. Click **Add** > **Query for data**, and select the QueryDocumentNodeSet data connection.

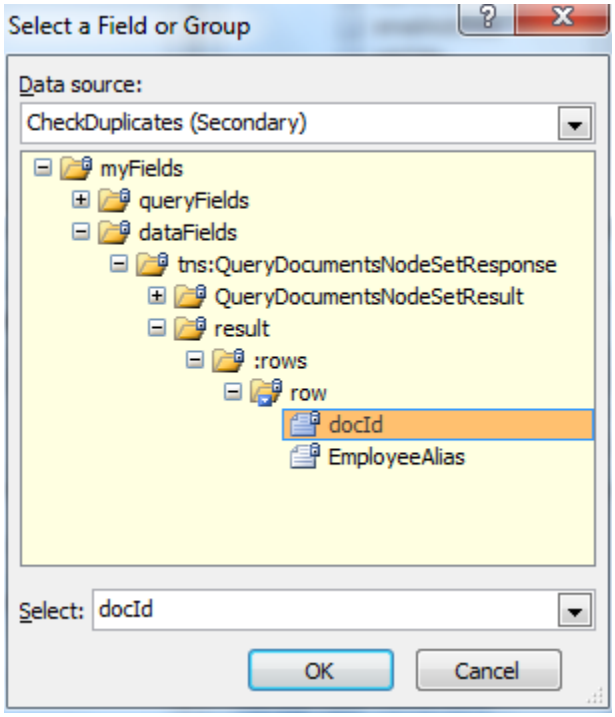


Your new Form Load rule now look like this:

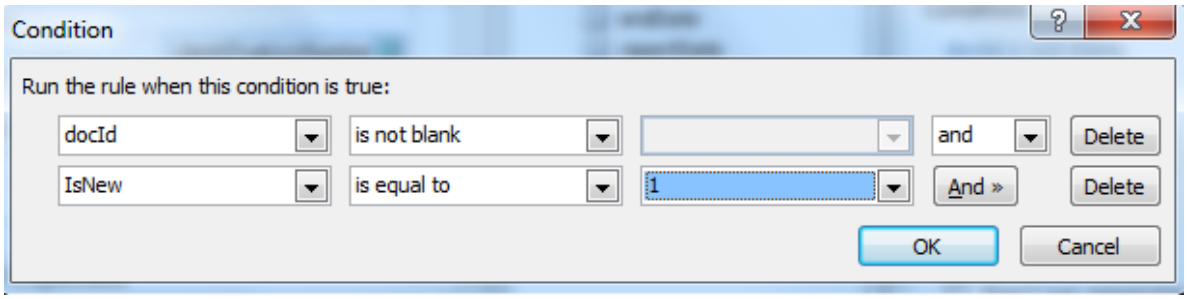


Finally, we need to set that flag in the secondary data source called FormOptions.

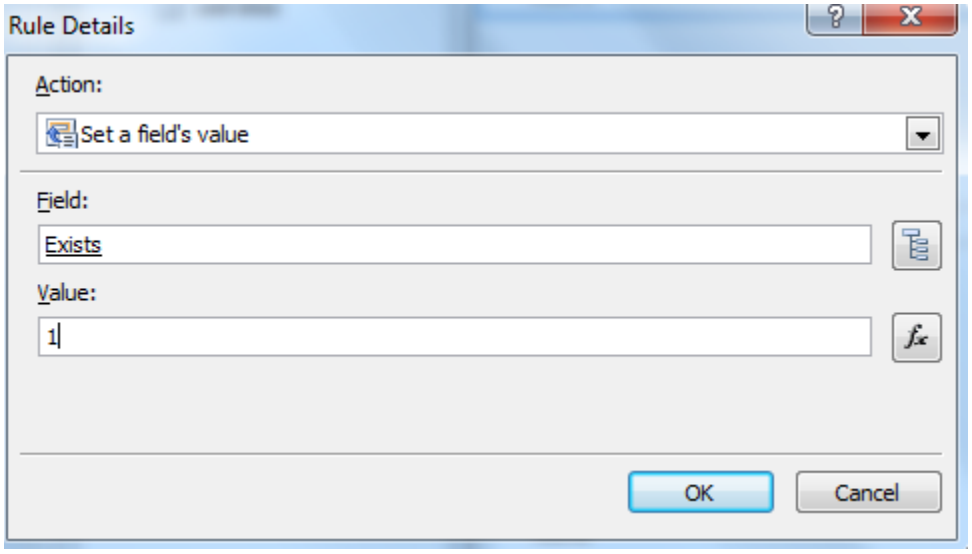
63. Click **New** > **Action**. Call this rule "Set flag".
64. Add a condition for docId (from the CheckDuplicates results) is not blank.



65. Again, we don't need to check this if we're opening an existing form, so add a condition for that as well.

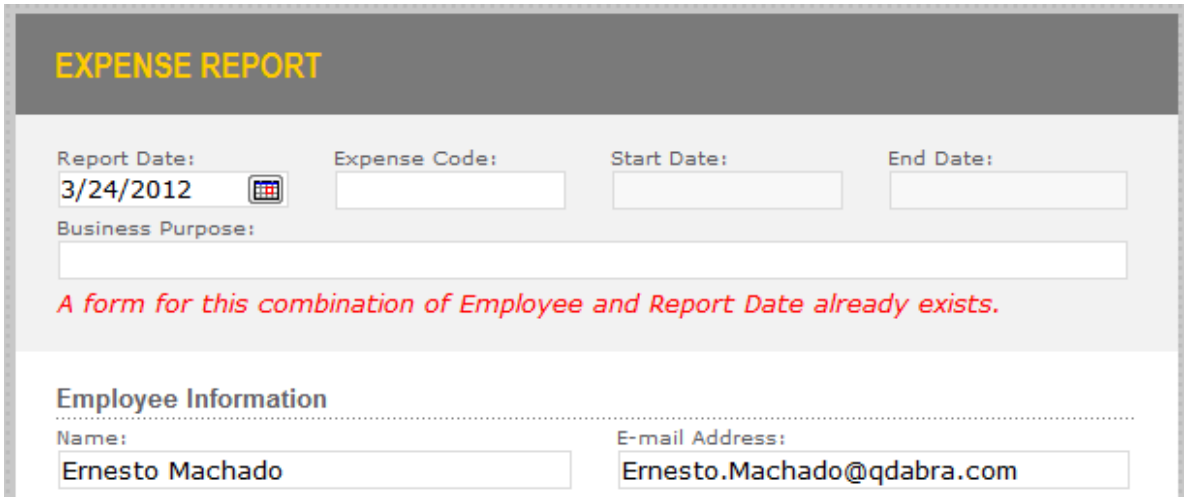


66. Click on **Add** > **Set a field's value**.
67. For the **Field**, select the **Exists** node from the **FormOptions** secondary data source.
68. For the **Value**, enter 1.

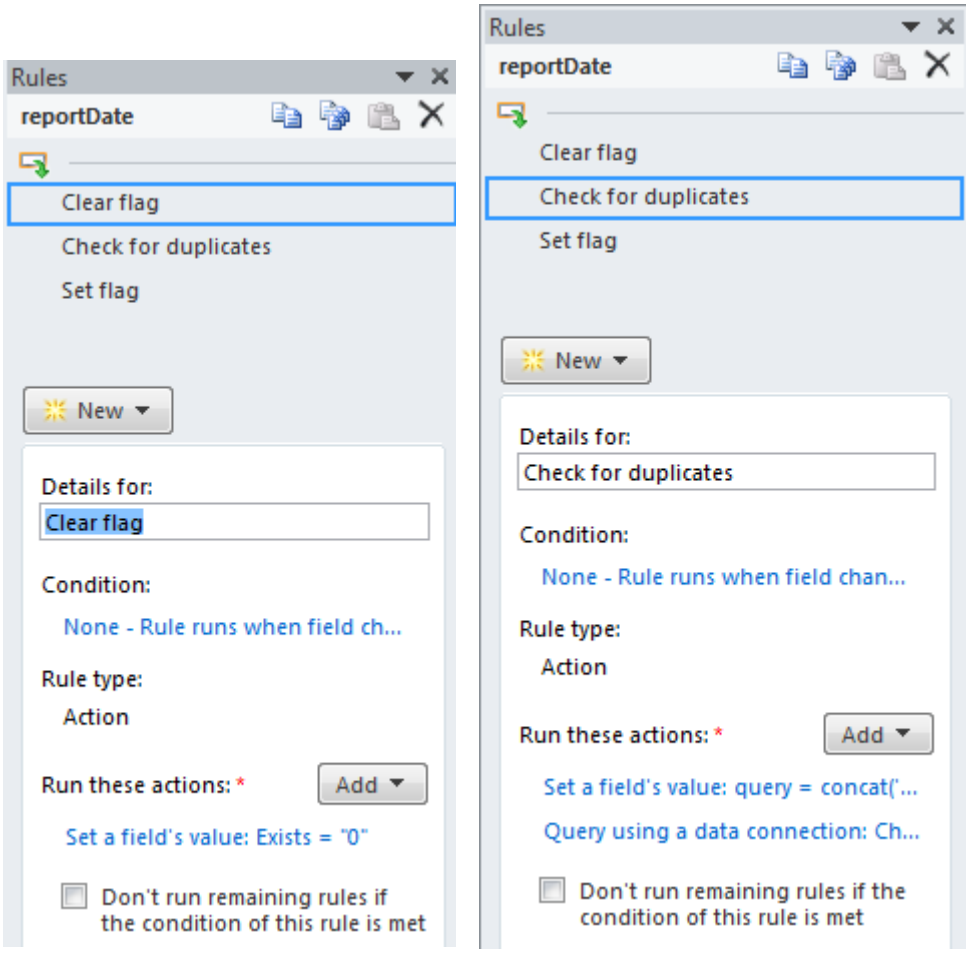


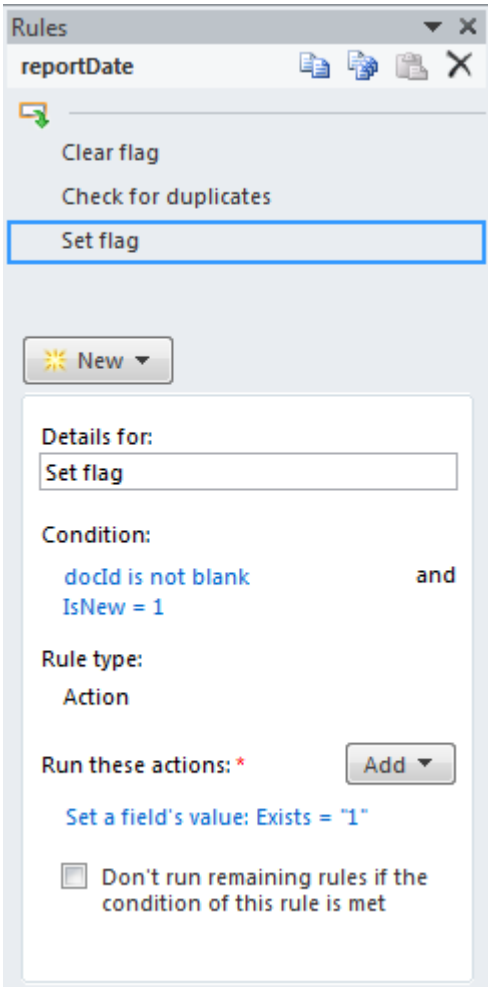
In other words, if a result was found, then a document for this user on this report date already exists!

69. Preview the form. Because you already submitted a document to DBXL, the form will detect this and hide the submit button, and display the message.



Extra credit: Add rules to the Report Date control, such that CheckDuplicates is called should the user change the ReportDate. You will need to rules, as seen below:


Rules
reportDate
Clear flag
Check for duplicates
Set flag
New
Details for:
Clear flag
Condition:
None - Rule runs when field ch...
Rule type:
Action
Run these actions: *
Add
Set a field's value: Exists = "0"
Don't run remaining rules if the condition of this rule is met
Rules
reportDate
Clear flag
Check for duplicates
Set flag
New
Details for:
Check for duplicates
Condition:
None - Rule runs when field chan...
Rule type:
Action
Run these actions: *
Add
Set a field's value: query = concat('...
Query using a data connection: Ch...
Don't run remaining rules if the condition of this rule is met


Rules
reportDate
Clear flag
Check for duplicates
Set flag
New
Details for:
Set flag
Condition:
docId is not blank
and
IsNew = 1
Rule type:
Action
Run these actions: *
Add
Set a field's value: Exists = "1"
Don't run remaining rules if the condition of this rule is met