



How to prevent a user from submitting duplicate data to DBXL

Imagine that your InfoPath form, which submits to DBXL, needs to make sure that there are no duplicate records. Using the web method QueryDocumentNodeSet, we can quickly check whether DBXL already contains a matching record.

For the purposes of this tutorial, we will implement the solution on a very simple form, though you can extend this as desired.

CREATE BASIC FORM

1. Launch the InfoPath designer and create a new, blank form.
2. Using Notepad, create a file called FormOptions.xml, whose contents are just:

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

3. In **Data > Data Connections**, click **Add**.
4. 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.

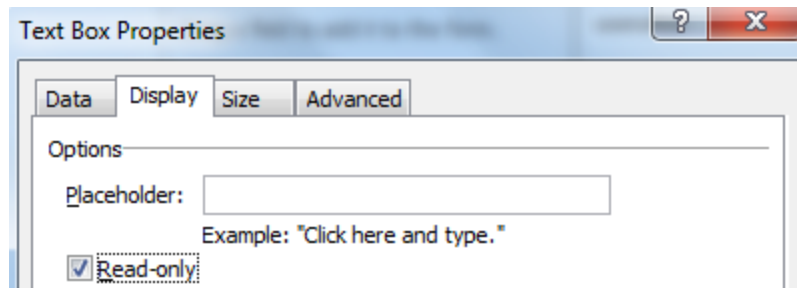
We will need these nodes later on.

5. Add one field, called **username**.

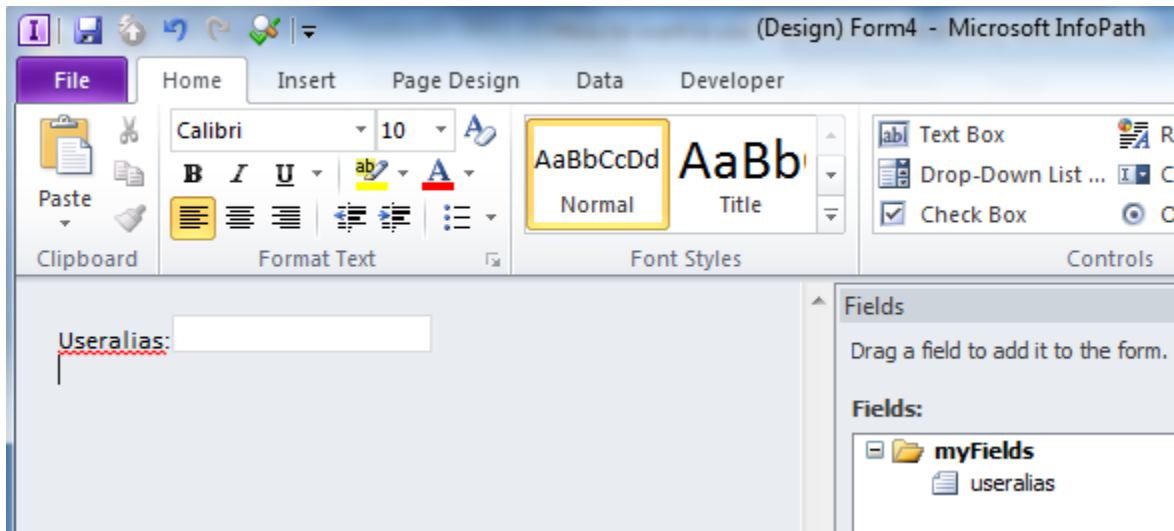
This scenario requires the presence of a unique key, so we will use the current username.

6. Drag and drop the field into the canvas. You should probably make it read only, so that users don't change its value and break the uniqueness.





So far, your form looks like this:



7. Go to **Data > Data Connections** and add a data connection to the ADUserInfo web service, to retrieve data from **GetMyInfo**. This data connection does not need to run on load.

For additional information or detailed steps, please refer to the [DBXL Getting Started document](#).

8. In **Data > Form Load**, add a rule that first queries **GetMyInfo** and then sets the value of the username field, but add a condition such that it only executes if the field is blank.
9. Finally, add one more action, which will set the **IsNew** node in the **FormOptions** secondary data source, to 1.



Details for:
This is a new form

Condition:
useralias is blank

Rule type:
Action

Run these actions: *

Add ▾

Query using a data connection: GetMyInfo

Set a field's value: useralias = Value[Key = "sAMAccountName"]

Set a field's value: IsNew = "1"

For more information on using the Active Directory, refer to [this document](#).

10. Add a Submit-type data connection, which submits data to the DBXL Document Service. For details and step by step instructions, please refer to the [DBXL Getting Started document](#).
11. Add a button that uses that data connection to submit.

Button Properties

General Size Advanced

Button

Action:
Submit ▾

Label:
Submit

Submit Options...

Use the Submit action to submit data to only one data source. To submit data to multiple data sources by using multiple buttons, create a rule for each button instead.

OK Cancel Apply

Submit Options

Configure options for submitting this form

☒ Allow users to submit this form

☒ Send form data to a single destination

Web service ▾

Choose a data connection for submit:
Main submit ▾ Add... Manage...

☐ Perform custom action using Rules

☐ Perform custom action using Code Edit Code...

☒ Show the Submit button in both the Ribbon and the Info tab in the InfoPath Filler.

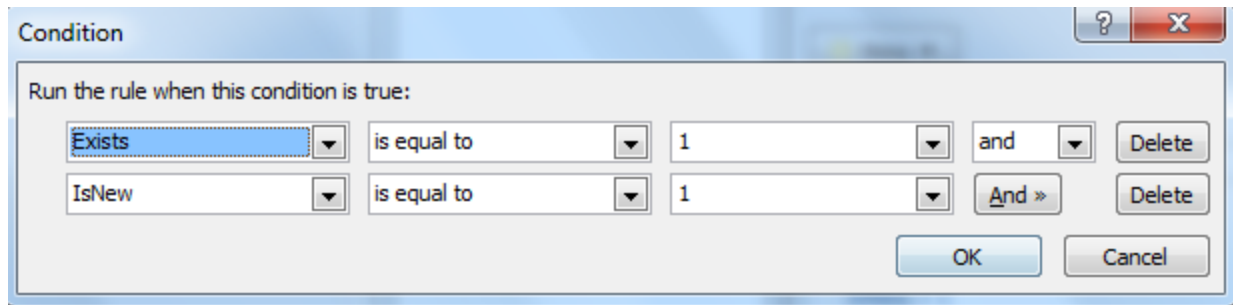
Label: Submit

Advanced >> OK Cancel

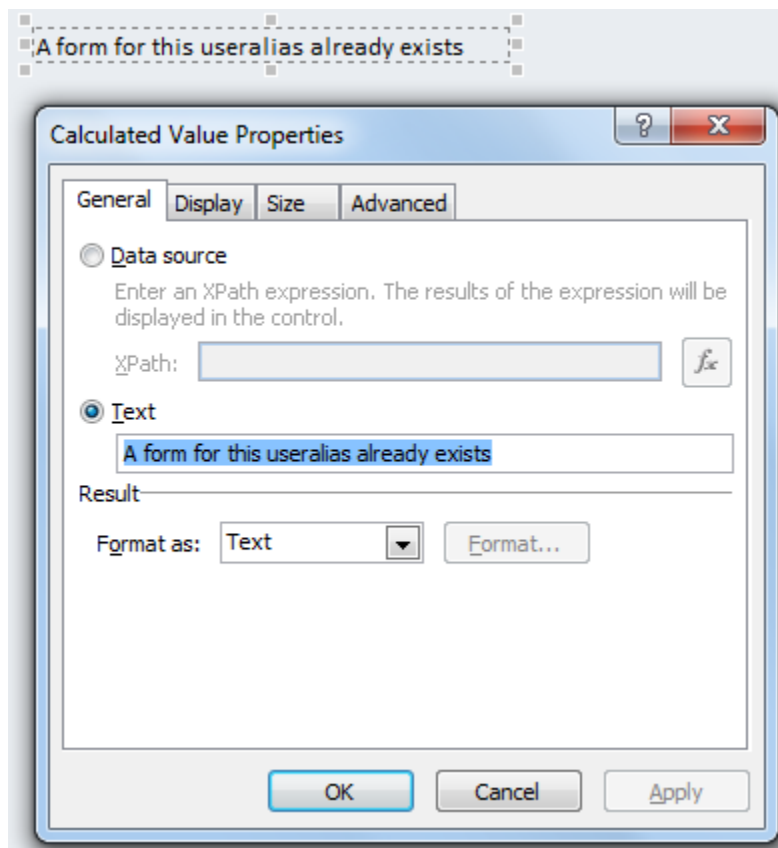
SET UP CONDITIONAL FORMATTING

12. Select the **Submit** button, and then click **Home > Manage Rules**.
13. Select **New > Formatting**. 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 **Hide this control**.





14. Add a Calculated value (Expression box) with the text: "A form for this useralias already exists".



15. Add a Hyperlink control.

Now, for both the Expression box and the hyperlink control:

16. Select **New > Formatting**. 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**.



Rules

Calculated Value

Hide if does not exist

New

Details for:
Hide if does not exist

Condition: *
Exists ≠ 1

Rule type:
Formatting

Formatting: *
 B I U abc A ▼ ▼
 (hidden)

☒ Hide this control
☐ Disable this control

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

Condition

Run the rule when this condition is true:

IsNew is equal to 0

And » Delete

OK Cancel

USE REPORT BUILDER TO CREATE THE QUERY

18. [Download and install Report Builder](http://www.qdabra.com).

In order to use Report Builder, our form must exist already in DAT.



<http://www.qdabra.com>

Last updated on 3/6/2012 11:12 AM

Copyright © 2006-2011 Autonomy Systems, LLC. All rights reserved.

19. Launch the DBXL Administration Tool (DAT) and create a new document type for the XSN saved in the section above. Call it "DuplicateTest".

For additional information or detailed steps on creating a new document type in DAT, please refer to the [DBXL Getting Started document](#).

The screenshot shows the QDABRA Administration Tool interface. The title bar reads 'DATABASE ACCELERATOR Administration Tool' and 'QDABRA SOFTWARE'. The 'Document Type: DuplicateTest' is selected. The 'General' tab is active, showing 'Document Type Details' with fields for 'Name' (DuplicateTest) and 'Solution Group' (abi). Below this is the 'InfoPath Form Template' section, which includes an 'XSN File' (SampleForm.xsn, 6.80 KB) and 'Form Template Properties' (Form Version, Last Modified, and checkboxes for 'Form template is signed' and 'URN-based solution').

20. In order to use create the query, we must have at least one document. Open the form and submit one document.
21. Once installed, launch the Report Builder by opening the InfoPath 2010 Filler and selecting Report Builder from the Installed Forms section.
22. Enter the Web Service URL Prefix, the Web Service Name, and then click **Connect**.

The screenshot shows the 'Connection Information' section of the QDABRA Administration Tool. It features a table with columns for 'Query Target', 'Web Service URL Prefix (scheme://host-name:port)', 'Web Service Name', and a 'Connect' button. The 'Query Target' is set to 'DBXL', the 'Web Service URL Prefix' is 'http://demo2', and the 'Web Service Name' is 'QdabraWebService'.

Query Target	Web Service URL Prefix (scheme://host-name:port)	Web Service Name	
☒ DBXL ☐ SharePoint	http://demo2	/ QdabraWebService	Connect

23. Under **Queries**, click **Add query**, then click **Create**.
24. Under **Query Information**, for the **Document Type** dropdown, select **DuplicateTest**. (i.e. the name of the document type in DBXL)



Query Information
Document Type
DuplicateTest

25. Under **Fields to Retrieve**, click the selector icon ().
26. In the taskpane, double click the base node, **my:myFields**.
27. Click **Insert item**.
28. Click the selector icon () under **Field**.
29. In the taskpane, double click the base node, **my:useralias**.
30. Under **Alias**, enter UserAlias. This is just a descriptive name for the retrieved field.

Fields to Retrieve	
Base Path	
/my:myFields	
Field	Alias
my:useralias	UserAlias
Insert item	

31. Under **Filter**, change the dropdown for **Type** to **Simple Comparison**.
32. Click the selector icon () and double click the useralias field in the taskpane.
33. In the middle dropdown, select the equal sign.
34. In the right most field, type in PLACEHOLDER.

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

35. Copy the XML for query.

QUERYDOCUMENTNODESET

Return to the form in design mode.

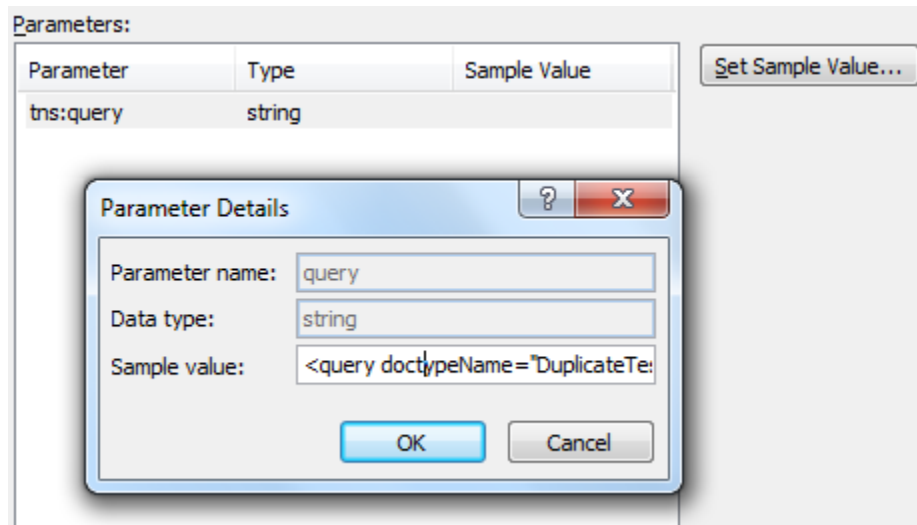
36. Go to **Data > Data connections** and click **Add**.
37. In the first screen of the Data Connection Wizard, select **Create a new data connection to Receive data**. Click **Next**.
38. Select **SOAP Web Service** and click **Next**.
39. Enter <http://<servername>/QdabraWebService/DBXLDocumentService.asmx>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

Now we need to use the query xml and we also need to remember the document that was submitted. In place of the text PLACEHOLDER enter the alias of the user. In my case, it is:

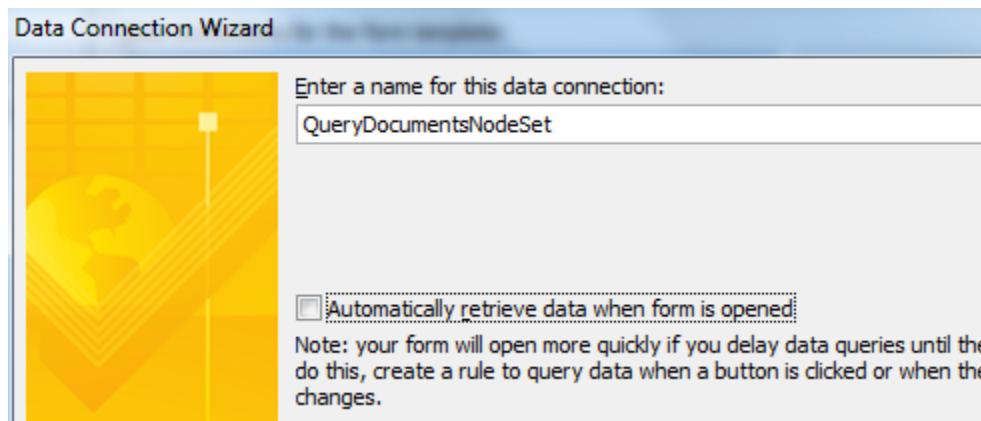
```
<query doctypeName="DuplicateTest"><columns repeatingPath="/my:myFields"><column name="my:useralias"
alias="UserAlias" /></columns><filter><eq><column name="/my:myFields/my:useralias"
/><value>ErnestoM</value></eq></filter></query>
```



41. Enter the query, click OK and click **Next** twice.



42. Uncheck the option to retrieve the data on load, and click **Finish**.



43. Click **Close** on the Data Connections screen.

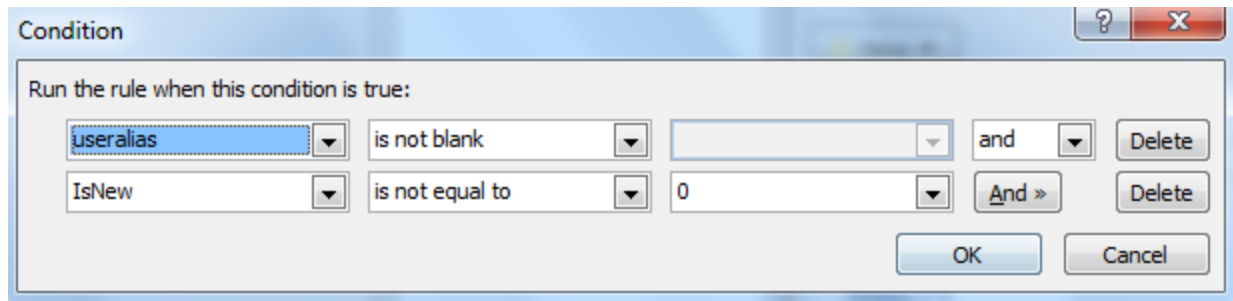
Now that we have the data connection, let's use it on load.

44. Return to **Data > Form Load**.

45. Click **New > Action**.

46. Add a condition such that this rule will execute when **useralias** 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.





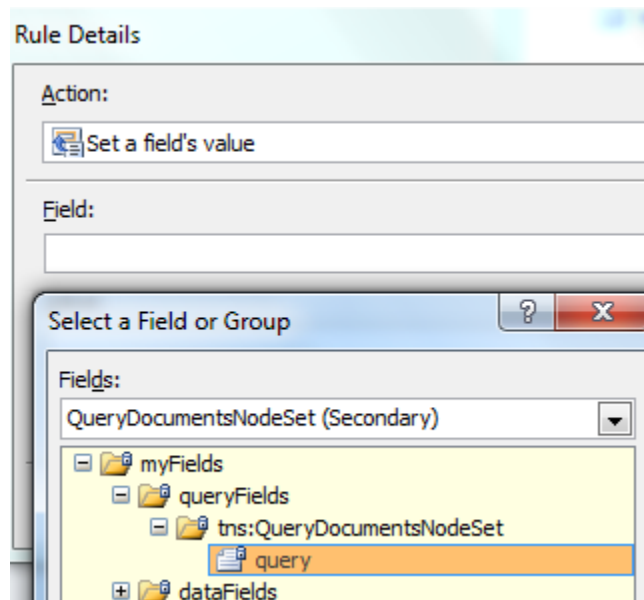
Condition

Run the rule when this condition is true:

useralias	is not blank		and	Delete
IsNew	is not equal to	0	And »	Delete

OK Cancel

47. Click on **Add > Set a field's value**.
48. For **Field**, select the query node from the **QueryDocumentNodeSet** data connection.



Rule Details

Action:

Set a field's value

Field:

Select a Field or Group

Fields:

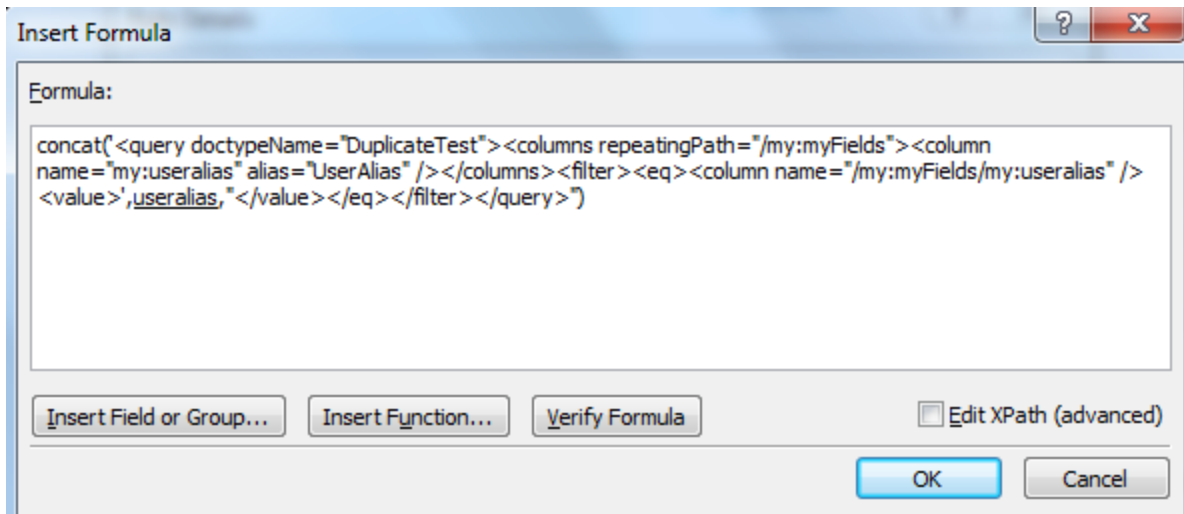
QueryDocumentsNodeSet (Secondary)

- myFields
 - queryFields
 - tns:QueryDocumentsNodeSet
 - query
- dataFields

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

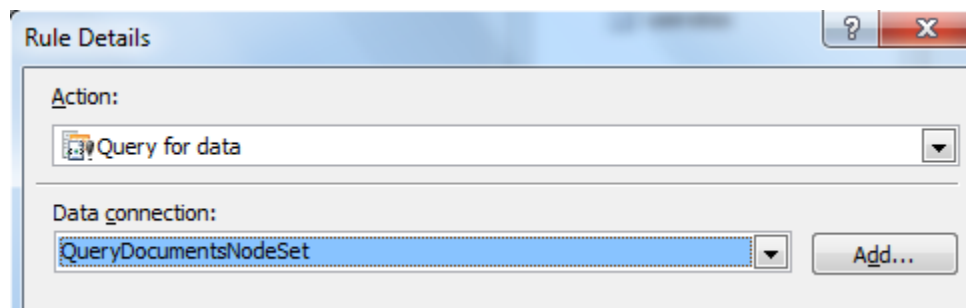
```
concat('<query doctypeName="DuplicateTest"><columns repeatingPath="/my:myFields"><column
name="my:useralias" alias="UserAlias" /></columns><filter><eq><column name="/my:myFields/my:useralias"
/><value>', my:useralias, "</value></eq></filter></query>')
```



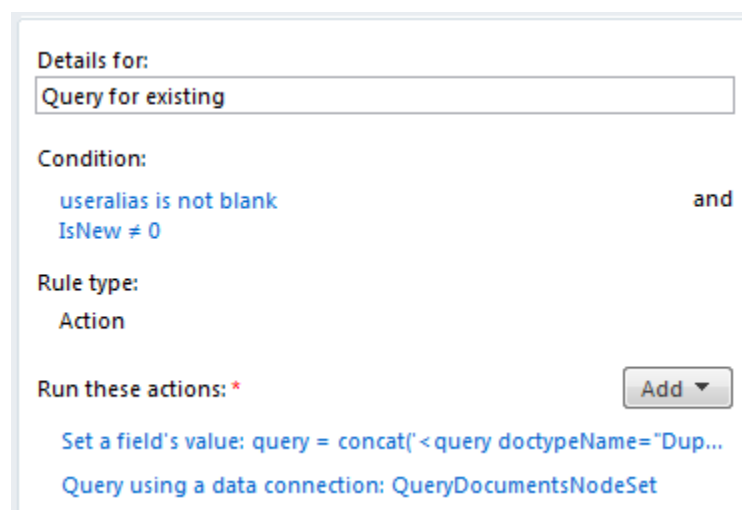


Where did this come from? This is the original query, generated by Report Builder, but we're using concat to include the useralias field instead of the text PLACEHOLDER.

50. Click **Add > Query for data**, and select the QueryDocumentNodeSet data connection.

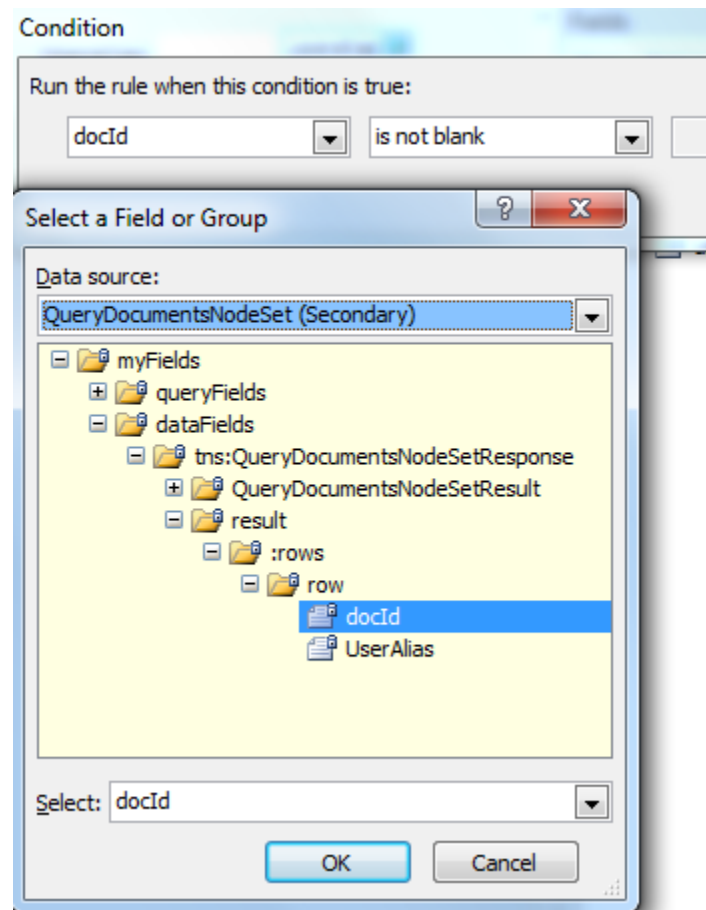


Your Form Load rules now look like this:

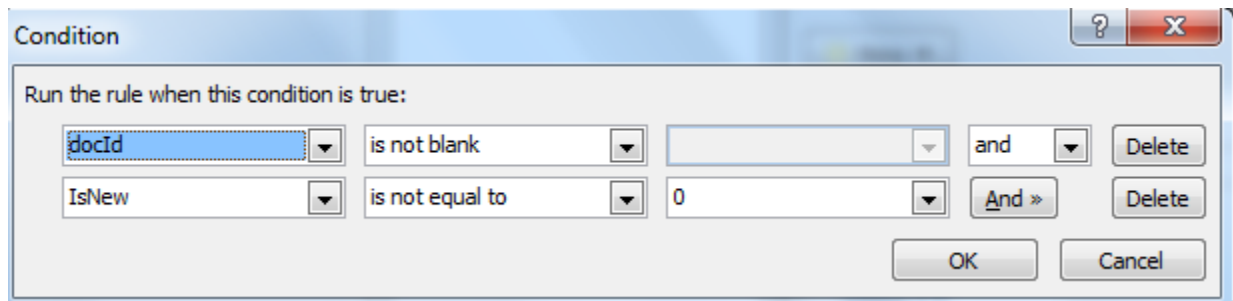


Finally, we need to set that flag in the secondary data source called FormOptions.

51. Click **New > Action**.
52. Add a condition for docId (from the QueryDocumentNodeSet results) is not blank.

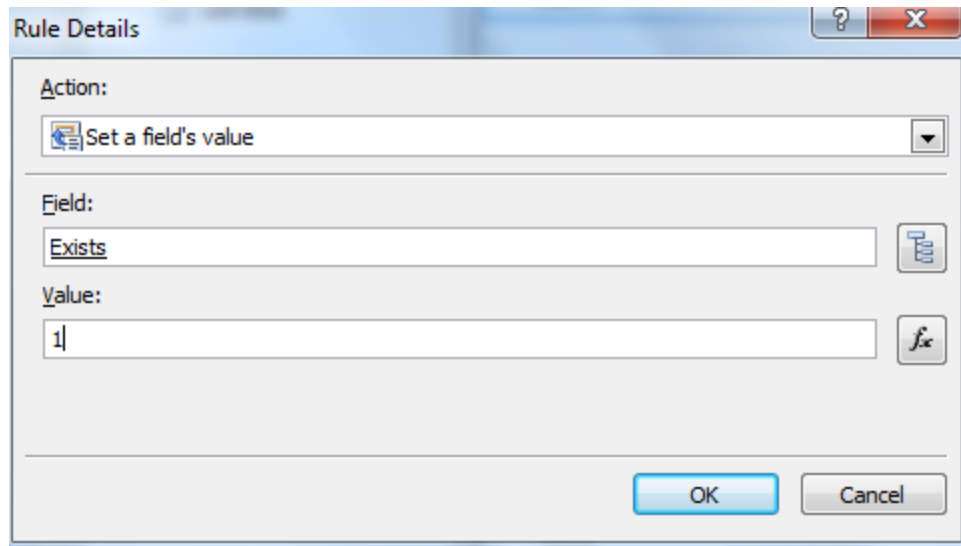


53. Again, we don't need to check this if we're opening an existing form, so add a condition for that as well:



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





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

But we need to set the value of the URL!

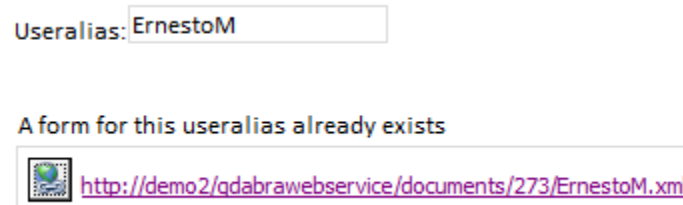
57. Click on **Add > Set a field's value**.
58. For **Field**, select the URL (hyperlink) field.
59. For **Value**, we're going to use concat to come up with the hyperlink needed to open the existing document.

```
concat("http://<servername>/qdabrawebservice/documents/",@docId,"/",useralias,".xml")
```

The @docid is from the results of the QueryDocumentNodeSet data connection. The useralias is from our main data source.

PUBLISH AND TEST

60. 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 and the link.



61. Click the link to test that it opens the existing document correctly. If all your conditions have been set up correctly, then the message and link seen above will not be seen in the existing form:



<http://www.qdabra.com>

Last updated on 3/6/2012 11:12 AM

Copyright © 2006-2011 Autonomy Systems, LLC. All rights reserved.

User alias:

Our sample form is simple, but you could have an entire section hide / show using the same conditions used on the Submit button. This way your users can modify the existing data but cannot create a new form for this same key (which, in this example, is the AD username).

SUPPORT

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

Licensed customers can contact us via Support@Qdabra.com. Alternatively, please use the InfoPathDev.com [Qdabra Product support forums](#) to request help.



<http://www.qdabra.com>

Last updated on 3/6/2012 11:12 AM

Copyright © 2006-2011 Autonomy Systems, LLC. All rights reserved.