



How to retrieve the DocID for a submitted form

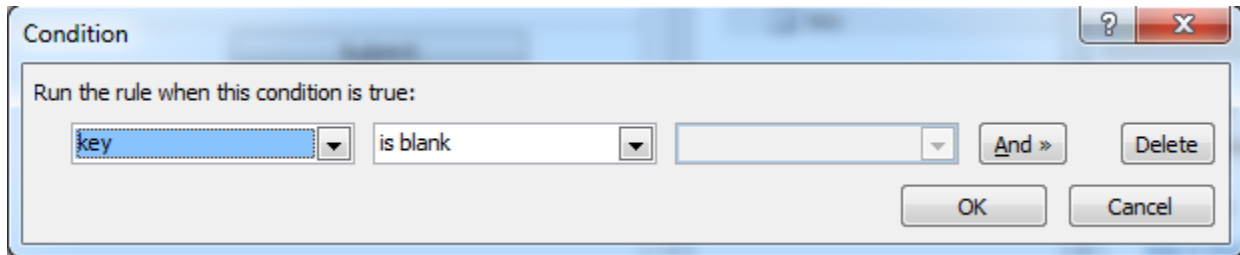
Sometimes it is useful to know the DocID of a document that was just submitted to DBXL. For example, you might want to send an email immediately after submission. So how can we retrieve the DocID of the document? Follow this tutorial to find out!

These steps assume that you have an InfoPath form template (XSN) configured as a Document Type that submits to DBXL. The document type must have at least one document for the purposes of these steps (it can be a test document and it can be deleted once you are done). For more information on how to achieve this pre-requisite, refer to the Getting Started document.

1. Open your XSN in design mode.
2. Go to **Data > Data Connections**.
3. Click **Add**. Follow the Data Connection Wizard to create a data connection that receives data from <http://<servername>/qdabrawebservice/ADUserInfo.aspx> using the method **GetMyInfo**. Be sure to uncheck the option to **"Automatically retrieve data when form is opened"**.
4. Add a new field and call it **key**.
5. Add one more node to the schema, and call it DocID.

A screenshot of the 'Add Field or Group' dialog box in Microsoft InfoPath. The dialog has a title bar with a question mark and a close button. Inside, there are several fields: 'Name:' with the text 'DocID', 'Type:' with a dropdown menu showing 'Field (element)', 'Data type:' with a dropdown menu showing 'Whole Number (integer)', 'Data namespace:' (empty), and 'Default value:' (empty). Below these fields is an 'Example:' label with the text '1234'. At the bottom, there are three checkboxes: 'Repeating' (unchecked), 'Cannot be blank (*)' (unchecked), and 'Non-empty occurrence required' (unchecked). At the very bottom are 'OK' and 'Cancel' buttons.

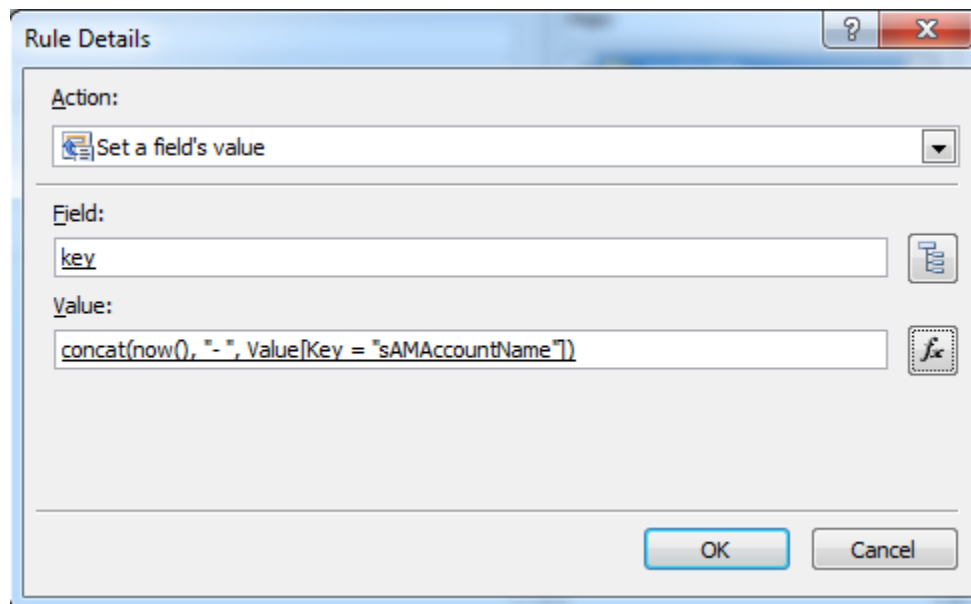
6. Click **Data > Form Load**.
7. Click **New > Action**. Rename the rule to "Set the value of key".
8. Add a condition such that this rule will only execute when key is blank.



9. Add an action that queries the GetMyInfo data connection that was created in step 3.
10. Add a second action that sets the value of the key to the following:

```
concat(xdDate:Now(), "- ",
xdXDocument:GetDOM("GetMyInfo")/dfs:myFields/dfs:dataFields/tns:GetMyInfoResponse/tns
:GetMyInfoResult/tns:ADProp/tns:Value[../tns:Key = "sAMAccountName"])
```

Note that you will need to click the fx button to enter the above formula. Your rule will look like this.



What is this rule doing? We are creating a unique key by concatenating the timestamp and the username. This creates a unique key because we can assume that one user won't be able to submit two documents at the same, exact moment.

For information on how to use Qdabra's Active Directory web service, refer to [this document](http://www.qdabra.com).

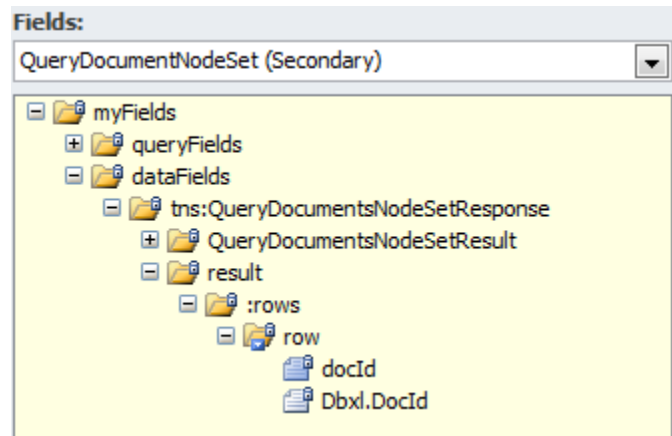
11. Next we will change the rules in the submit button. Set the button so that **Action** is set to **Rules** instead of **Submit**.



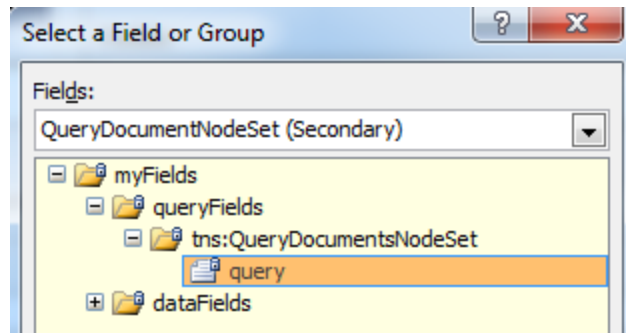
12. Go to **Data > Data Connections > Add**. Follow the Data Connection Wizard to add a data connection that receives data from <http://<servername>/qdabrawebservice/dbxldocumentservice.aspx>. This data connection will call the method QueryDocumentNodeSet and will not be set to “**Automatically retrieve data when form is opened**”. In the parameter screen, when asked for the value of the tns:query parameter, enter the following:

```
<query doctypeName="YOURDOCTYPENAME"><columns
repeatingPath="/my:myFields"><column name="DocId" type="token" /></columns></query>
```

You can confirm that this data connection successfully obtained a schema by checking the dataFields:



13. For the submit button, click **New > Action > Add > Submit Data**. In the dialog that appears, select the data connection that submits to DBXL.
14. Add a second action that sets the value of the query node:



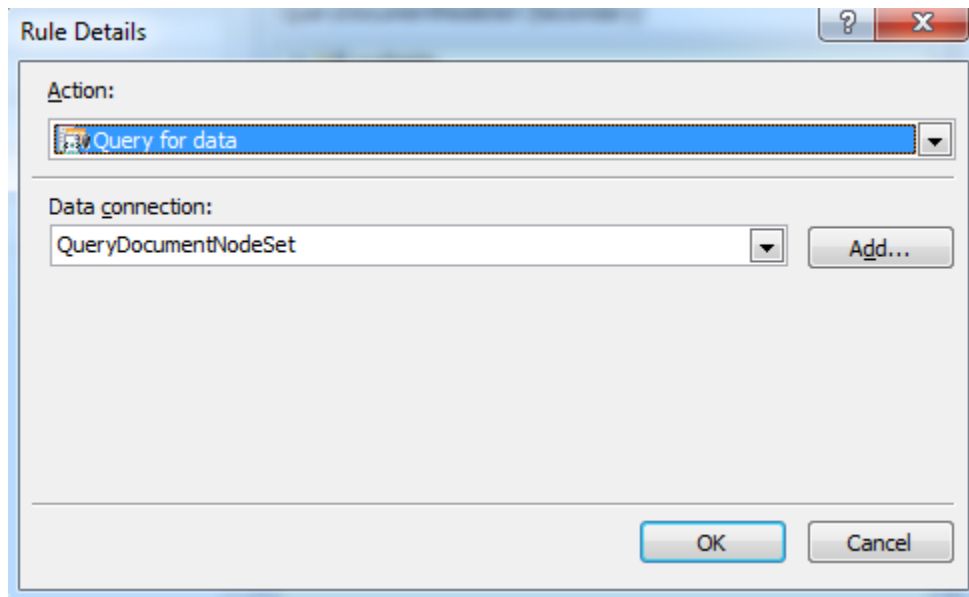
The value will be:

```
concat('<query doctypeName="YOURDOCTYPENAME"><columns repeatingPath="/my:myFields"><column
name="DocId" type="token" /></columns><filter><eq <column name="my:myFields/my:key"
/><value>',key,'</value></eq></filter></query>')
```

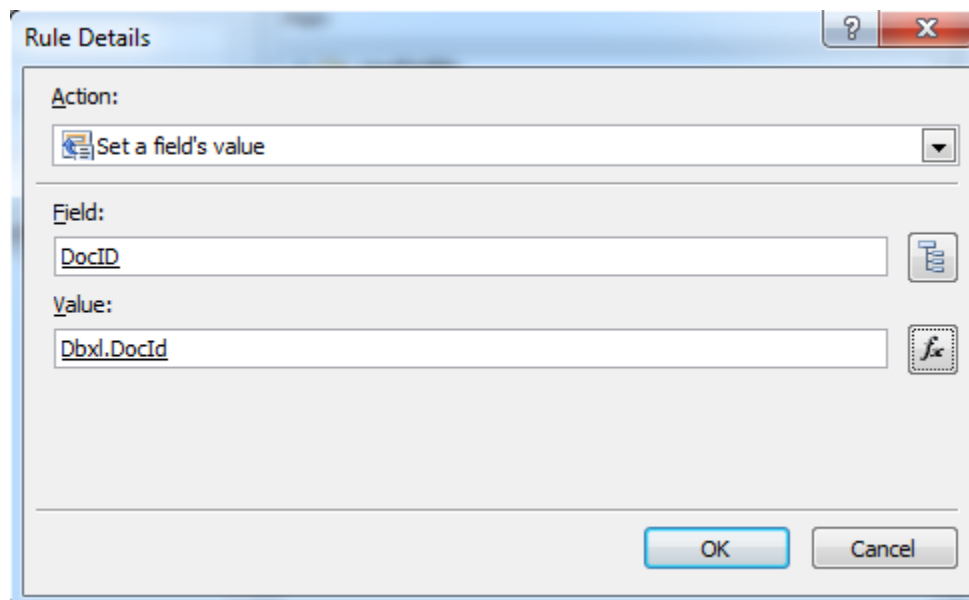
You will need to use the fx button when entering this value. In it, you'll see YOUR document type name and key, which is the field we added earlier in the steps.

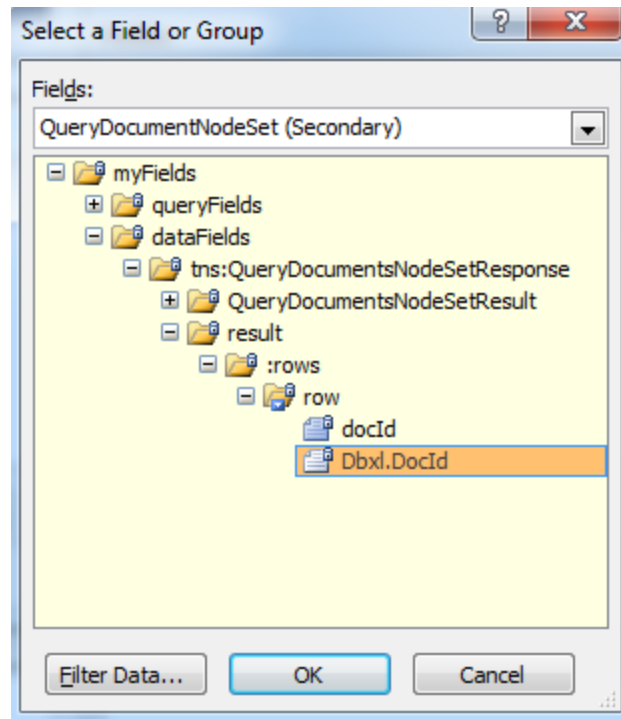


15. The third action will query data from the QueryDocumentNodeSet data connection.



16. The fourth action will set the value of my:DocID to the value returned by the data connection.





17. Save the XSN, re-attach it to your document type in DAT, and **Save**.

You can now test by submitting a new document and verifying that the DocID is returned to the form immediately after submitting.

This DocID can be used to generate a direct URL to your form, for example. The DocID, along with concat, can be used to set a field's value to the URL, like this:

```
concat("http://YOURSERVERNAME/qdabrawebservice/documents/",DocID,"/",DocID".xml")
```

Note that this method is only necessary to get the DocID for a newly submitted document. To obtain the DocID for a document that already exists in DBXL, you can use the processing instructions, as explained in [this document](#).

