

HOW TO USE THE DBXL SIDE-BY-SIDE XTP

PRODUCT: DBXL v2.4 or newer
LAST UPDATED: March 3, 2010

SCENARIO

In DBXL v2.3 we can accomplish “Side-by-Side” by writing code in the form. However, for those that can’t add code to their form, there is a new feature in DBXL 2.4 (or later) called “Side-by-Side”. Side-by-Side refers to the architecture. Forms using this architecture can be codeless and stand alone. They stand alone because they have no DBXL DocID reference in them – they can live in SharePoint or on someone’s local disk. Codeless means that no code exists in the form – this requires adding a schema node and some form logic changes.

VALUE PROPOSITION

To simplify the form changes, Qdabra created the DBXL Side-By-Side template part. Users can add the XTP in InfoPath, and then just drag and drop to the bottom of the form. The XTP verifies that a form is not out-of-date (more recent form exists), checks if a form is locked for editing by another user (thus preventing simultaneous edits and/or submits) and retrieves DBXL information about the form.

This document details the instructions on how to add and use the DBXL Side-By-Side template part in your DBXL solutions.

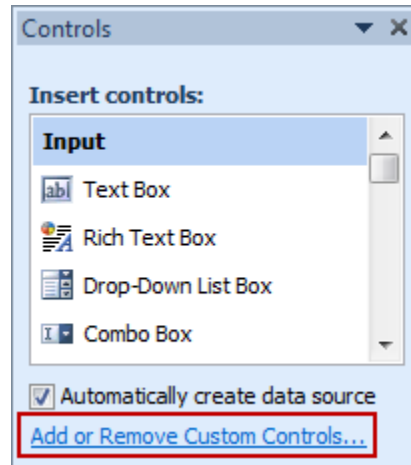
REQUIREMENTS

- DBXL v2.4 or higher
- InfoPath 2007 with SP2 or InfoPath 2010 with SP2
- SharePoint

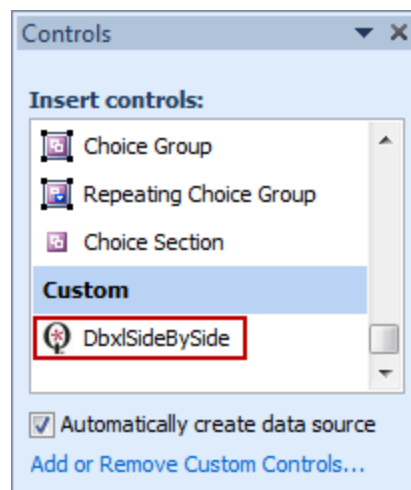


ADD THE TEMPLATE PART IN YOUR INFOPATH FORM

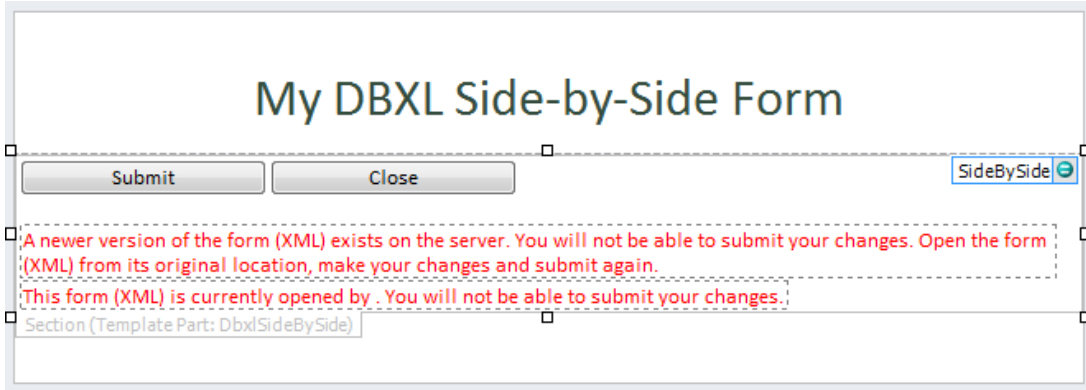
1. Launch InfoPath Designer and design a new blank form template.
2. In the *Controls* pane, click **Add or Remove Custom Controls...**
*Tip: In InfoPath 2010, you can display the Controls pane using your keyboard by first pressing **Alt + H** (to go to the Home tab) and then press **O**.*



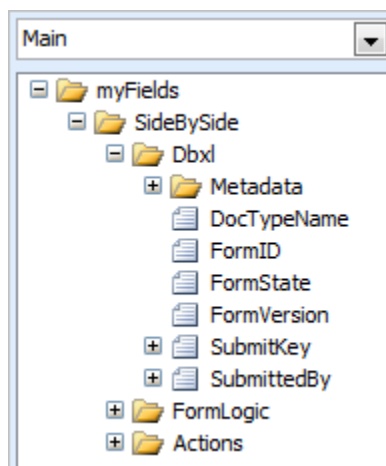
3. In the Custom Controls dialog box, click **Add > Template Part**, and browse to the location of your *Qdabra DBXL Side-by-Side* template part.
4. Verify that the XTP has been added to your Custom Controls.



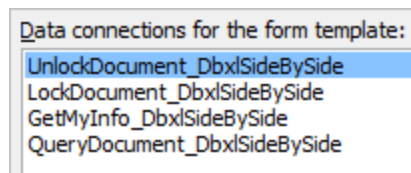
5. Drag the *DbxlSideBySide* XTP on your form canvas.
 - a. This will add a section on your form, similar to this:



- b. Your main data source will look similar to this:



- c. You will have the following DBXL Web Service data connections:



These temporarily point to [HTTP://localhost/qdabrawebservice](http://localhost/qdabrawebservice) which will automatically be updated to your DBXL server URL when you attach the XSN to your *DBXL Administration Tool* (DAT) later.

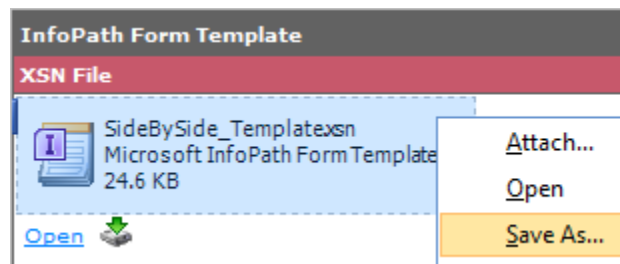
6. Save your form template (XSN), and then close when done.



CREATE A DOCUMENT TYPE IN DBXL

7. Launch the DBXL Administration Tool (DAT):
 - a. Open Internet Explorer and enter the URL `http://<servername>/qdabrawebservice/` replacing `<servername>` with your own.
 - b. Click on the DBXL Administration Tool link.

Tip: If you are using InfoPath 2010 and are getting an error message that states: "Forms that require the domain permission level contain features that access information on a network". – follow the workaround in this [blog post](#).
8. In the *Catalog* view of DAT, click **New Configuration**.
9. Enter a name of your choosing, e.g. *SideBySide_Template* (Note that spaces and punctuation marks are not allowed in document type names.) We will use this name when we modify our XSN later.
10. In the *InfoPath Form Template* section, click inside the XSN File attachment control, browse to the location where you saved your XSN, and then click **Insert**.
11. When done, click on the **Save** button to save your new doctype configuration, and then click **OK** in the confirmation dialog.
12. Obtain the saved XSN by right-clicking on the attachment and clicking **Save As...**



13. You may then overwrite your existing XSN because this version already has data connections pointing to your own DBXL server instance.



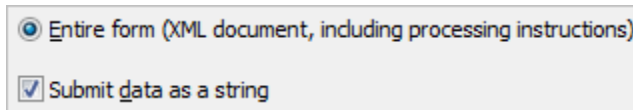
MODIFY YOUR FORM TEMPLATE (XSN)

Since template parts (XTPs) do not support some features such as adding submit data connections, on-load rules, etc.; we will perform these steps in the form template (XSN) itself. Open your form template in InfoPath Designer and follow the steps below to start with the modifications.

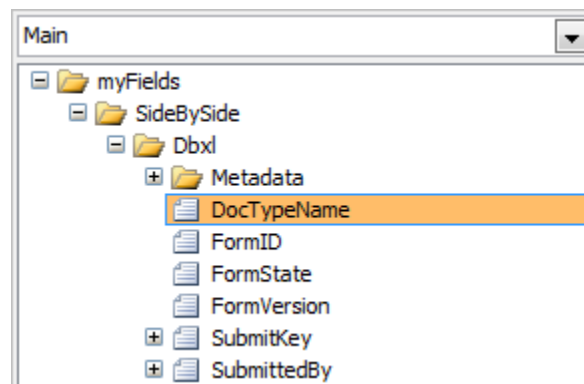
ADD THE SUBMIT DATA CONNECTIONS

14. Go to **Tools > Data Connections > Add**, and follow these guidelines to add a DBXL submit data connection:

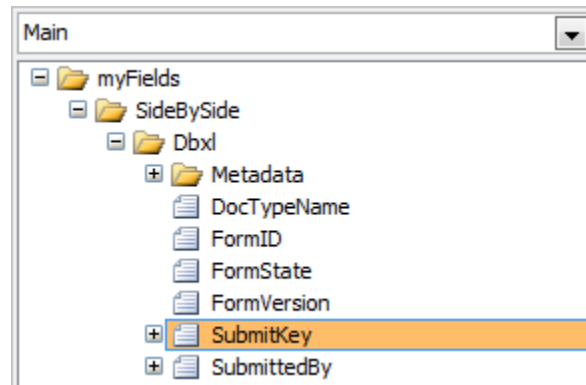
- a. Create a new connection that submits data to a Web service.
- b. Enter the DBXL Document web service URL in the format:
`http://<servername>/qdabrawebservice/dbxldocumentservice.asmx`, replacing **<servername>** with your own.
- c. Select the **SubmitDocumentWithKey** operation in the list.
- d. Set the elements/values of your parameters:
 - i. **tns:xml** – Entire form, Submit data as a string (enable these in the parameter options)



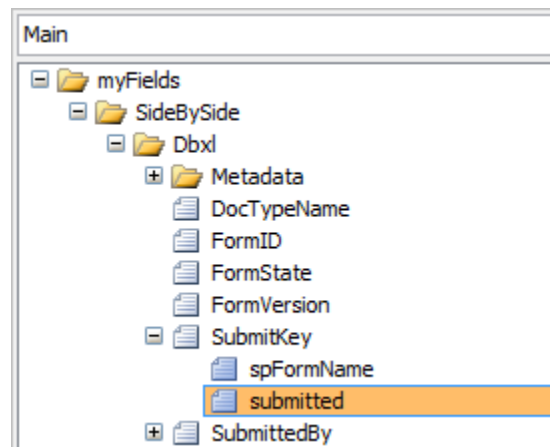
- ii. **tns:documentType** – /my:myFields/my:SideBySide/my:Dbxl/**my:DocTypeName**



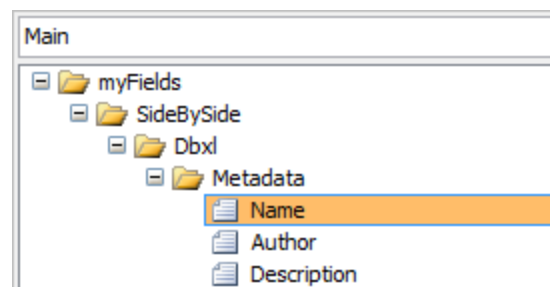
- iii. **tns:key** – /my:myFields/my:SideBySide/my:Dbxl/**my:SubmitKey**



- iv. **tns:clientDateTime** –
/my:myFields/my:SideBySide/my:Dbxl/my:SubmitKey/**@my:submitted**



- v. **tns:name** – /my:myFields/my:SideBySide/my:Dbxl/my:Metadata/**my:Name**



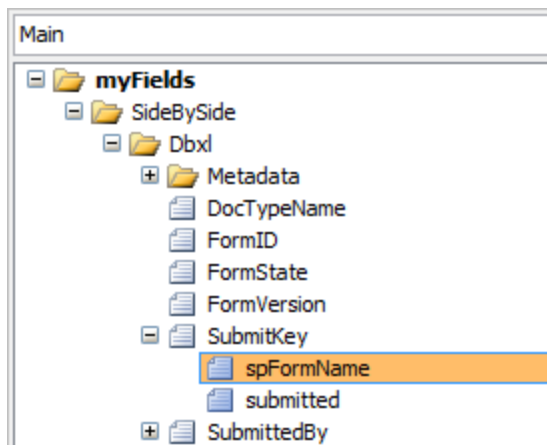
- vi. **tns:author** – /my:myFields/my:SideBySide/my:Dbxl/my:Metadata/**my:Author**
vii. **tns:description** –
/my:myFields/my:SideBySide/my:Dbxl/my:Metadata/**my:Description**



Your Parameters screen should look like this:

Parameter	Type	Element
tns:xml	string	/
tns:documentType	string	/my:myFields/my:SideBySide/my:Dbxl/my:DocTypeName
tns:key	string	/my:myFields/my:SideBySide/my:Dbxl/my:SubmitKey
tns:clientDateTime	string	/my:myFields/my:SideBySide/my:Dbxl/my:SubmitKey/@my:submitted
tns:name	string	/my:myFields/my:SideBySide/my:Dbxl/my:Metadata/my:Name
tns:author	string	/my:myFields/my:SideBySide/my:Dbxl/my:Metadata/my:Author
tns:description	string	/my:myFields/my:SideBySide/my:Dbxl/my:Metadata/my:Description

- e. Give your DBXL submit data connection any name of your choosing.
15. Still in the *Data Connections* window, click **Add** again, and follow these guidelines to add a SharePoint submit data connection (Note: This assumes that you already have a SharePoint form library where you wish to submit your XMLs to.):
 - a. Create a new connection that submits data to a form library on a SharePoint site.
 - b. Enter the SharePoint document URL where you wish to submit your forms to, e.g *http://<servername>/SideBySide_Forms*, replacing *<servername>* with your own.
 - c. Point the File name to *my:SideBySide/my:Dbxl/my:SubmitKey/@my:spFormName*.



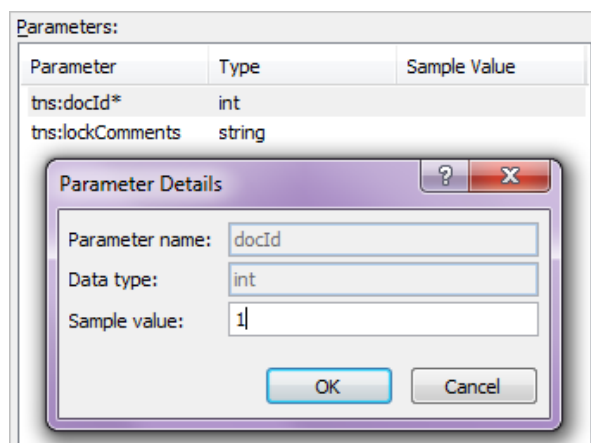
- d. Check the box "Allow overwrite if file exists".
- e. Give your SharePoint submit data connection any name of your choosing.



CLEAN-UP THE DATA CONNECTION NAMES (OPTIONAL)

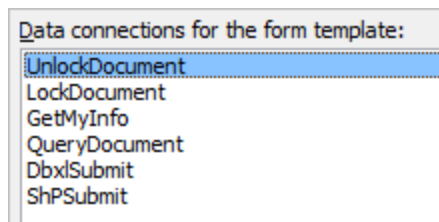
Notice that the data connections included in the XTP have suffixes in their names. Although **not required**, you may remove these suffixes **as desired**. To do so, follow these guidelines:

- You can modify the **GetMyInfo_DbxlSideBySide** and the **QueryDocument_DbxlSideBySide** data connections by simply clicking *Modify* and proceeding to the next screens of the wizard, leaving all the defaults as they are (including blank parameters). Remove the suffix in the name on the last screen, just before clicking *Finish*.
- Since the data connections **Lock_DbxlSideBySide** and **Unlock_DbxlSideBySide** both use web service methods that require sample values to query with, set the required docId parameter to 1:
 - In the first screen where you see the list of parameters, click on the *Set Sample Value...* button corresponding to the **tns:docId** parameter, and then enter the number 1.



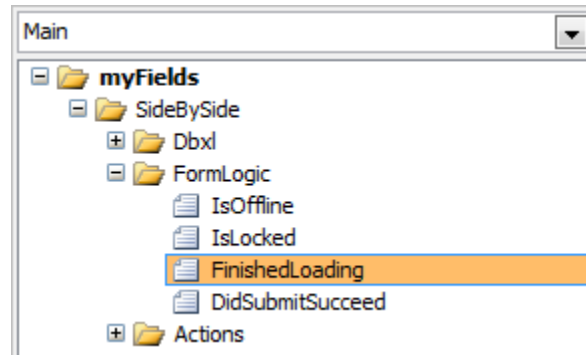
- In InfoPath 2007, when you click *Next*, notice another screen that looks similar to the above image. No need to set a sample value this time, i.e. leave it blank. Click *Next*, remove the suffix in the name and click *Finish*.
- In InfoPath 2010, when you click *Next*, notice a check box that states: "Include change information when submitting data" – uncheck this box, click *Next*, remove the suffix in the name and click *Finish*.

Your data connections list may look similar to this:

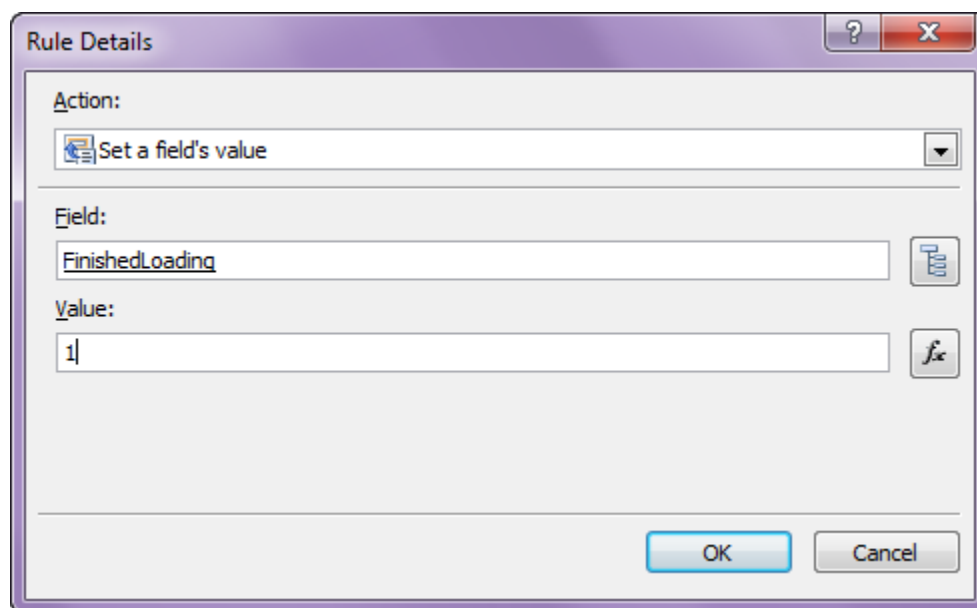


MODIFY THE FORM LOGIC

16. Add a rule that triggers on form open:
 - a. Go to **Tools > Form Options**, and then under the *Open and Save* category, click **Rules**.
*Tip: In IP2010, you can access the **Form Load** button under the Data tab > Rules section.*
 - b. Add a new form-load rule with the action that sets the **FinishedLoading** field:



to the value of "1" (without the quotes, enter **1** directly inside the *Value* box i.e. no need to click on the **fx** button).



- c. Give your rule a descriptive name such as "Trigger Finished Loading Rules".



Your form-load rule may look similar to this:

Details for:
Trigger Finished Loading Rules

Condition:
None - Rule runs when form is opened

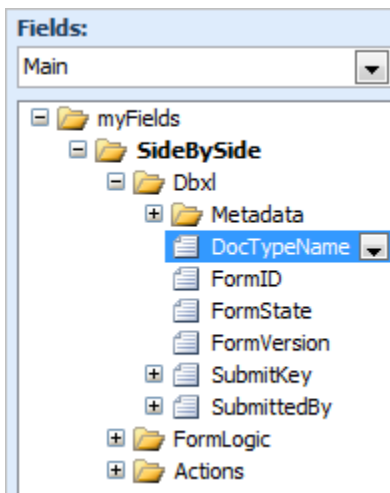
Rule type:
Action

Run these actions: * Add ▼

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

☐ Don't run remaining rules if the condition of this rule is met

17. Specify your document type name (this is the name of the doctype that you have configured earlier).
 - a. In your *Data Source* or *Fields* taskpane, locate the **DocTypeName** field:



- b. Right-click on the field and select **Properties**.
 - c. Enter your DBXL document type name in the *Value* box:

Default Value

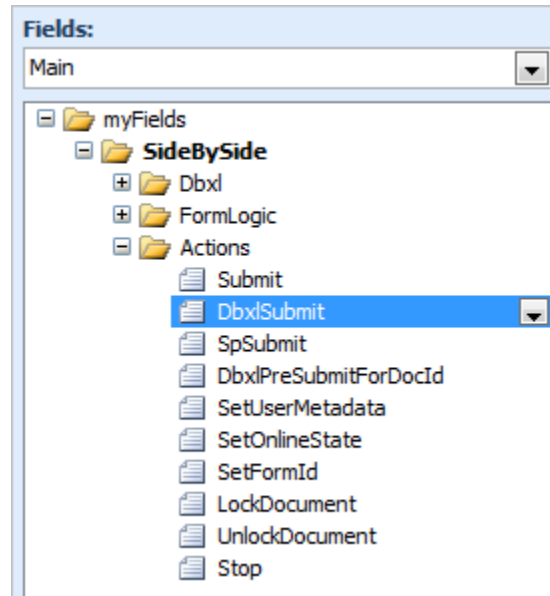
Value: 

Example: Sample Text

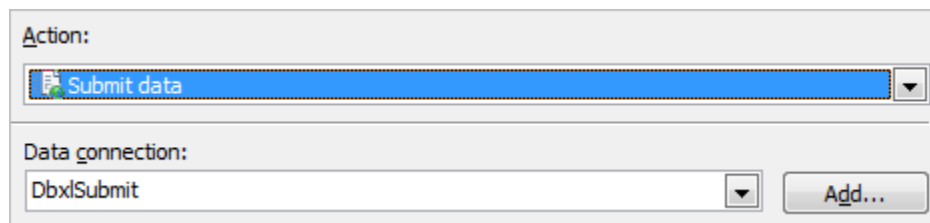
☐ Refresh value when formula is recalculated



18. Add a new 'DBXL Submit' rule that uses the DBXL Submit data connection you've created earlier.
 - a. In your *Data Source (Fields in IP2010)* taskpane, locate the **DbxlSubmit** field:



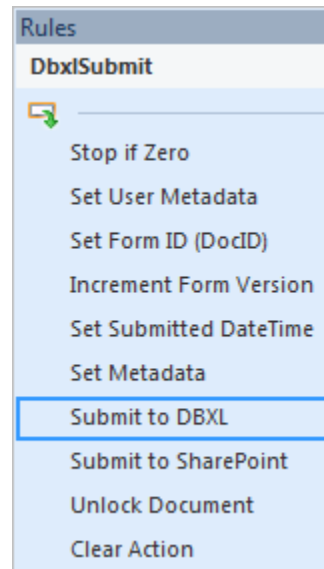
- b. Right-click on the field and select **Properties**, and then go to the *Rules and Merge* tab.
*Tip: In IP2010, you can add/edit rules by clicking on the **Manage Rules** button under the Home tab > Rules section.*
 - c. Right after the 'Set Metadata' rule, add a new rule with an action that submits data to your DBXL Submit data connection.



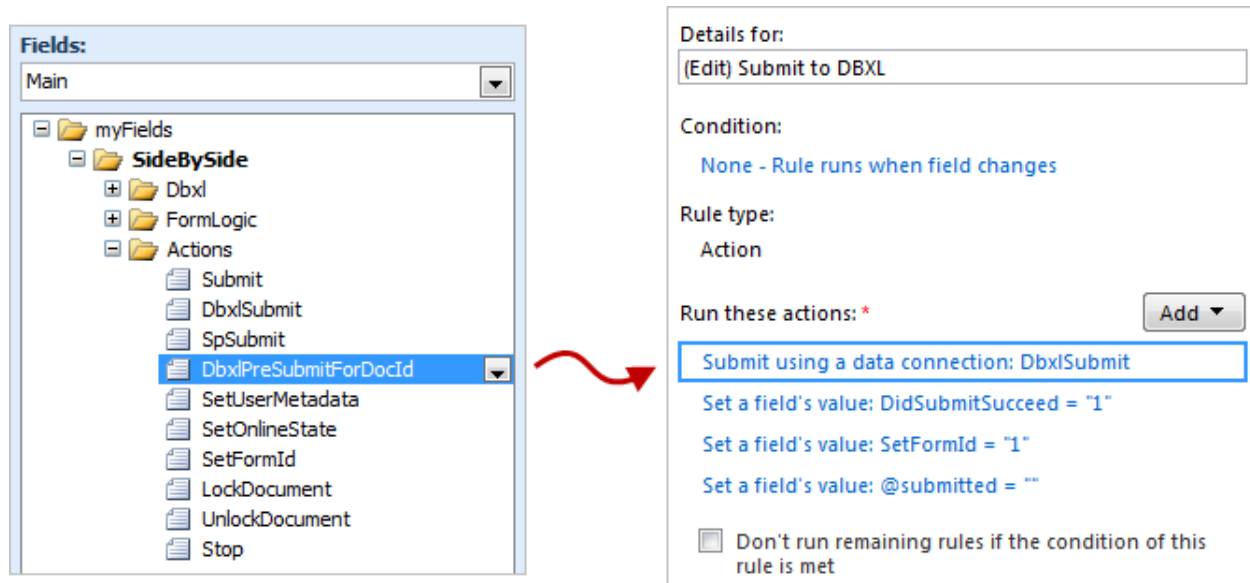
- d. Give your rule a descriptive name such as "Submit to DBXL".



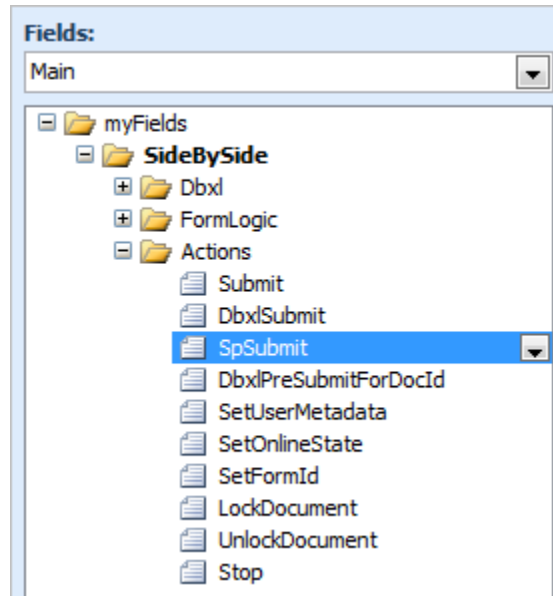
You should have the following set of rules in the DbxlSubmit field:



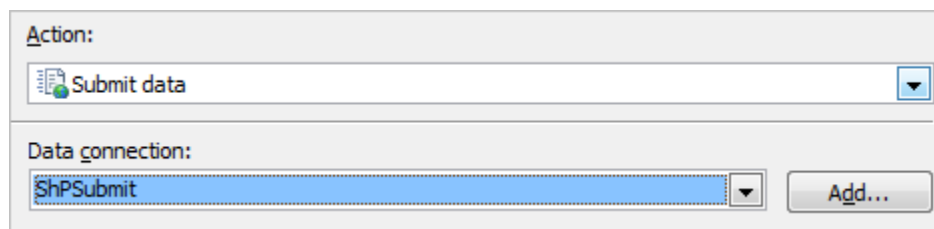
19. Repeat *Step 18* to edit the DBXL Submit rule in the **DbxlPreSubmiForDocId** field, this time adding DBXL submit as the first action in the list:



20. Add a 'SharePoint Submit' rule that uses the ShP Submit data connection you've created earlier.
- In your *Data Source (Fields in IP2010)* taskpane, locate the **SpSubmit** field:

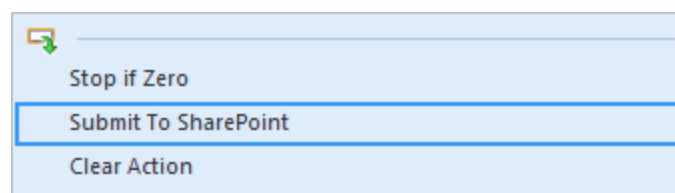


- Right-click and select **Properties > Rules and Merge** (*skip this step if you are on IP2010*).
- Add a new rule with an action that submits data to your SharePoint Submit data connection.



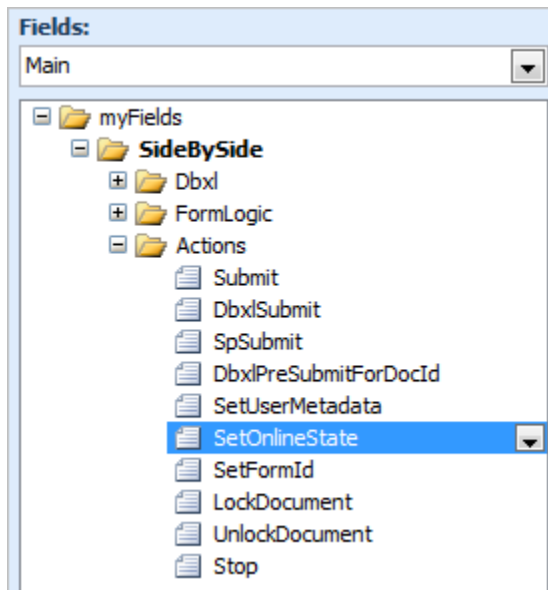
- Give your rule a descriptive name such as "Submit to SharePoint".

Your **SpSubmit** field rules should be in the following order (*Submit To SharePoint* being the second in the list):



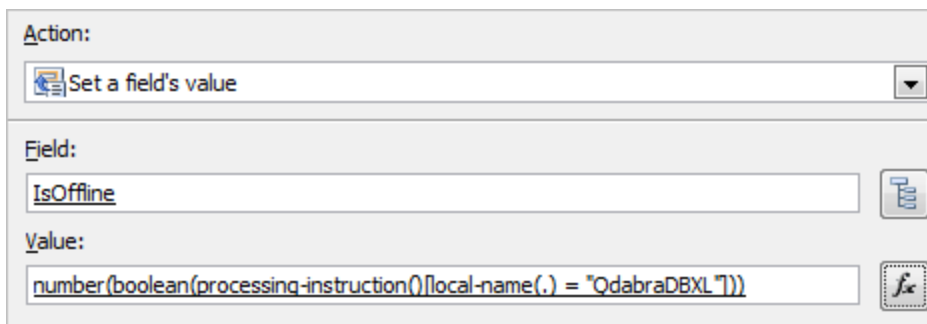
21. Edit the rules in the **SetOnlineState** field:

- a. In your *Data Source* (*Fields* in IP2010) taskpane, locate the **SetOnlineState** field:



- b. Right-click and select **Properties > Rules and Merge** (*skip this step if you are on IP2010*).
- c. Modify the rule called (Edit) Set IsOffline; select the only action and click **Modify** again.
- d. Copy the entire text in the *Value* box (also including here in case you've accidentally deleted it):
`number(boolean(xdXDocument:get-DOM()/processing-instruction())[local-name(.) = "QdabraDBXL"])]`
- e. Click on the **fx** button, paste the entire text in the *Formula* box and click **OK**.

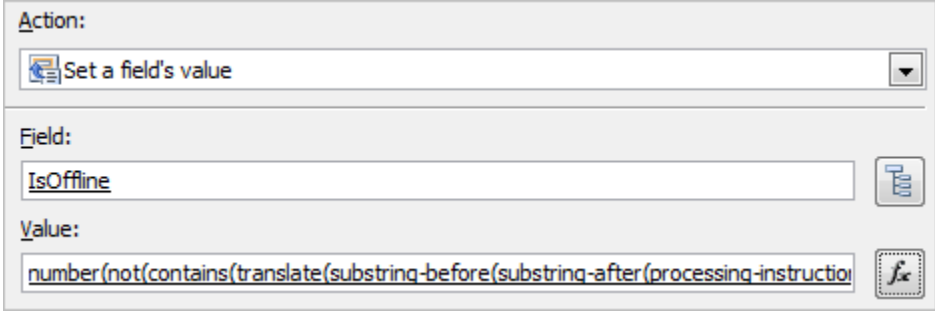
Your action dialog for this rule should be similar to this:




- f. Still in the **SetOnlineState** field rules, repeat *Steps c-e*, this time modifying the rule named (Edit) Double-Check using HREF value. The formula to use is:

number(not(contains(translate(substring-before(substring-after(xdXDocument:get-DOM()/processing-instruction()[local-name(.) = "mso-infoPathSolution", 'href="'), ""), "ABCDEFGHIJKLMNOPQRSTUVWXYZ", "abcdefghijklmnopqrstuvwxyz"), "/forms/template.xsn"))

Your action dialog for this rule should be similar to this:



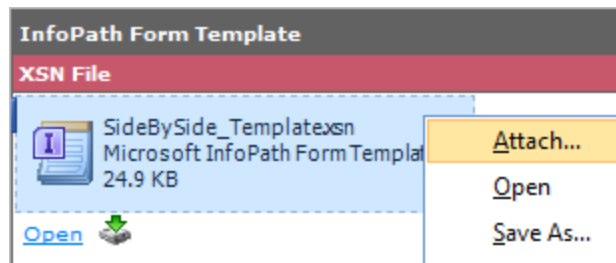
The screenshot shows a dialog box with three sections: 'Action:', 'Field:', and 'Value:'. The 'Action:' section has a dropdown menu with 'Set a field's value' selected. The 'Field:' section has a text box containing 'IsOffline'. The 'Value:' section has a text box containing the formula: `number(not(contains(translate(substring-before(substring-after(processing-instruction[local-name(.) = 'mso-infoPathSolution', 'href='], ''), 'ABCDEFGHIJKLMNOPQRSTUVWXYZ', 'abcdefghijklmnopqrstuvwxyz'), '/forms/template.xsn')))`. There are icons for field selection and formula insertion next to the respective text boxes.

Save your changes. You're done modifying your Side-by-Side form template.



PUBLISH YOUR FORM TEMPLATE

22. Go to **File > Publish** and publish your form template to your SharePoint library by following these guidelines:
 - a. Select the option to publish 'To a SharePoint server with or without InfoPath forms Services'. (In IP2010, click on the **SharePoint Server** button.)
 - b. Enter your SharePoint server URL.
 - c. Select 'Document Library' (Form Library in IP2010).
 - d. Select the option 'Update the form template in an existing document/form library'.
 - e. From the list, select the library where you've set your forms to submit to, e.g. *SideBySide_Forms*.
 - f. Promote the fields which you wish to display as columns in your SharePoint library. For the purposes of this tutorial, we will add the fields
Form ID (/my:myFields/my:SideBySide/my:Dbxl/my:FormID) and
Form Version (/my:myFields/my:SideBySide/my:Dbxl/my:FormVersion).
 - g. Finish the *Publishing Wizard* and close when done.
23. Re-publish the XSN to DBXL by updating the document type configured earlier.
 - a. Open the DBXL Administration Tool (DAT).
 - b. If you're in the Catalog view, click on the **Edit** icon (📝) for your doctype. Otherwise, proceed to the next step.
 - c. Right-click on the XSN File attachment and select **Attach...**



- d. Browse to your latest form template (XSN) and click **Insert**.
- e. Click **Save** in DAT.



TEST YOUR SOLUTION

You may test by simulating the following scenarios:

- Submit
 - Verify that you are able to submit new forms (XMLs) from SharePoint. XMLs get submitted to both SharePoint and DBXL.
 - **Note that XML forms submitted from DBXL are not mapped to SharePoint. All new forms and modifications must happen after opening the form from SharePoint. Otherwise, SharePoint and DBXL will fall out of sync.**
- Version check
 - Open an existing XML from SharePoint, save it locally, and then close the form.
 - Re-open the same XML from SharePoint, make changes and submit.
 - Open the XML saved locally – you should get a warning message stating there is a newer version of the form; Submit button is disabled.
- Locking
 - Open an existing XML from SharePoint.
 - Have another user open the same form – this user should see a warning stating the form is locked by another user; Submit button is disabled.

IMPORTANT NOTES

- The following DBXL Permissions should be added for users accessing this Document Type:
 - NT AUTHORITY\Authenticated Users should be granted ADD, READ and WRITE access.

General	Database	Permissions	SharePoint	Documents	Taxonomy Tree			
☒ Enforce Permissions								
Document Type Level Permissions At least one role must be given Admin permission at the document type level. Otherwise a global DBXL admin is required to edit the document type. Only one user should be configured per row. Do not list users delimited with semicolons.								
Name	Role Name	Add	Read	Write	L-Read	L-Write	Del	Admin
Everyone	NT AUTHORITY\Authenticated Users	☒	☒	☒	☐	☒	☐	☐
☒ Insert document type level permissions								
Document Level Permissions								
Name	Membership XPath	Read	Write	L-Read	L-Write	Del		
☒ Insert document level permissions								

- All users should use the **Close** button included in the XTP. This will unlock the XML and make it available for editing by other users.



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.

You can also use the InfoPathDev.com [Qdabra Product support forums](http://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
Website:	http://www.qdabra.com
Community:	http://www.infopathdev.com

