

HOW TO USE REPORT BUILDER TO BUILD QUERIES AND RETRIEVE DATA FROM DBXL

PRODUCT: DBXL v2.5 or higher
LAST UPDATED: May 27, 2016

SCENARIO

Have you ever wanted to be able to create detailed reports from your XML data without going to the trouble of creating a database mapping? With DBXL v2.5 or higher and the Report Builder tool, now you can.

VALUE PROPOSITION

You could query your database via Query Builder, but that would mean having a database and mapping fields to a database in order to get the Query Builder working. However, you can easily perform a simple XML reporting in your form using DBXL's QueryDocumentsNodeSet and the Report Builder tool. Through this, you can:

- Extract data from your InfoPath forms even if you don't have a database mapping
- Build your queries using DBXL's Report Builder tool
- Export the results to Excel for pivoting and charting

REQUIREMENT

- DBXL v2.5 or higher

PUBLISH A SAMPLE INFOPATH FORM TO DBXL

1. Download the sample InfoPath form template ([CompanyContactInfo.xsn](#)) here – this is a simple InfoPath form that has a few fields and allows submit to DBXL via the *SubmitDocument* operation.
2. Open the form template in InfoPath Designer.
3. Go to **File > Save As** and save the XSN locally.
4. Launch the DBXL Administration Tool (DAT) from your Database Accelerator Control Panel (the local access URL would typically be in the format <http://localhost:8000/QdabraWebService/default.htm>).
5. Create a new document type configuration – give it a descriptive name and then attach your XSN. When done, click **Save**.



DATABASE ACCELERATOR Administration Tool

Document Type: CompanyContactInfo

 Save
 Save As...
 Catalog

General
Database
Permissions
SharePoint
Documents
Taxonomy Tree

Document Type Details

Name	Solution Group
CompanyContactInfo	▼ abi
Description	
Last Modified By	
AUTONOMYSYSTEMS\dmbatin	

InfoPath Form Template

XSN File	Form Template Properties
CompanyContactInfo.xsn Microsoft InfoPath Form Template 14.3 KB Open	Form Version: 1.0.0.19 Last Modified: 8/10/2015 4:29:01 PM ☐ Form template is signed ☐ URN-based solution
URL to Form Template	

6. Click on the **Open** link to open a new form using the template. Fill out and submit at least three to four documents.

Documents							Delete Delete All Reshred Reshred All	
Doc. ID	Ref. ID	Version #	Name	Author	Description	New		
436	1	1	Acme, Inc.			Open		
437	2	1	The Best Company			Open		
438	3	1	Widgets, Ltd.			Open		
439	4	1	Sprockets Corp.			Open		

BUILD YOUR QUERIES USING THE REPORT BUILDER TOOL

7. Download the Report Builder tool from your Database Accelerator Control Panel.

Forms

Qdabra Query Builder - see [Installing and Opening Query Builder](#) for assistance

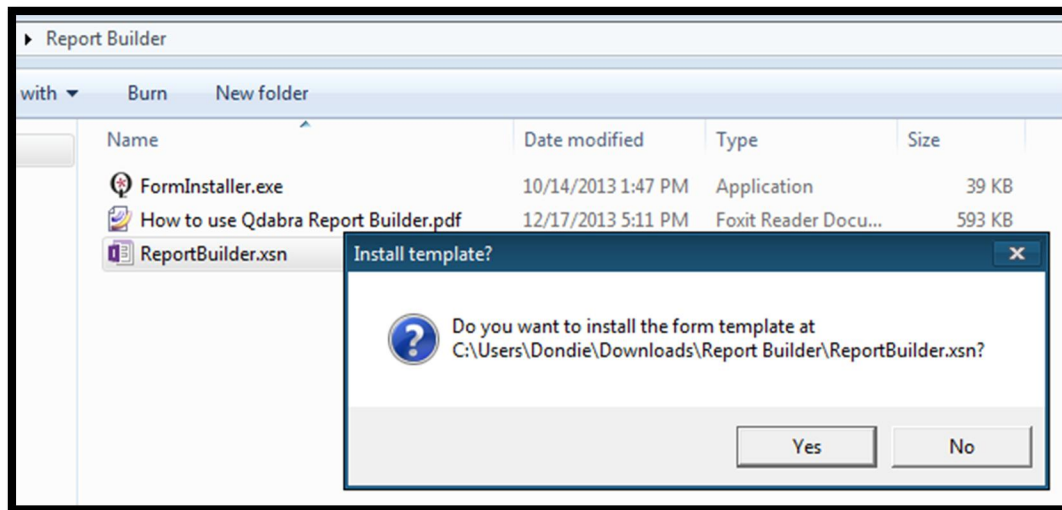
How to use the Standalone Dashboard

Qdabra Report Builder

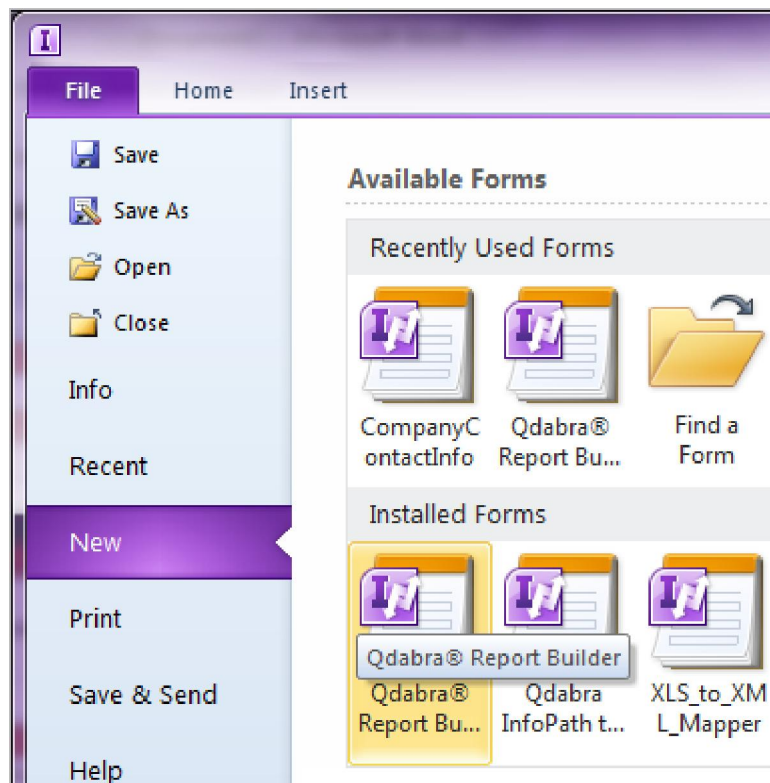
8. To install, first make sure that InfoPath is not running (close all active InfoPath windows).



9. Drag and drop the ReportBuilder.xsn onto FormInstaller.exe



10. Clicking "Yes" will install the tool, otherwise the tool will be uninstalled if you already have a previous installation of the Report Builder tool.
11. Launch the tool by double clicking the Report Builder.xsn (you may also open InfoPath, click New and select **Qdabra Report Builder** listed under the *Installed Forms* section).



12. In the *General* section of the Report Builder tool, enter the Web Service URL Prefix, e.g. <http://localhost:8000>.



13. Enter **QdabraWebService** as the Web Service Name and then click **Connect**.

Connection Information			
Query Target	Web Service URL Prefix (scheme://host-name:port)	Web Service Name	
☒ DBXL ☐ SharePoint	http://localhost:8000	/ QdabraWebService	Connect

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

Queries	
Document Type	
Create	
Add Query	

15. Select the document type you created in Step 5.

Document Information	
Document Type	
CompanyContactInfo	

16. Click on the *Schema* icon () – this will display the custom task pane at the right, double-click on the root node which in our sample XSN is **/my:myFields**.
17. 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.

Fields to Retrieve		
Base Path		
/my:myFields		
Field	Alias	
my:OriginalId	ID	
my:CompanyName	Company	
my:PhoneNumber	Phone	
Insert item		

Notice the *Query XML* section – here, you'll see the query string we'll be using to retrieve the data using DBXL web service *QueryDocumentsNodeSet* operation later on. Copy this string and paste it in Notepad for now.

Query XML	
XML for query	
<pre><query doctypeName="CompanyContactInfo"><columns repeatingPath="/my:myFields"><column name="my:OriginalId" alias="ID" /><column name="my:CompanyName" alias="Company" /><column name="my:PhoneNumber" alias="Phone" /></columns></query></pre>	Perform Query



18. 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 such:

Returned Query Data			
Documents			
Document Type: CompanyContactInfo			
docId	ID	Company	Phone
436	123	Acme, Inc.	090-1000-0000
437	456	The Best Company	123-444-5555
438	789	Widgets, Ltd.	212-555-1212
439	122	Sprockets Corp.	999-222-9922

USE QUERYDOCUMENTSNodeSET TO AUTO-POPULATE DROPDOWNS

We can now use the DBXL's web service method called *QueryDocumentsNodeSet* to retrieve data in a form from another form.

19. In InfoPath Designer, create a new blank form.
20. Go to **Data > Data Connections > Add** and add a data connection that retrieves data from a SOAP Web service.
21. The URL to the DBXL Document Service would be in the format:
<http://<DBXLServer>/QdabraWebService/DbxlDocumentService.asmx>, where <DBXLServer> is the machine where DBXL is installed (in this case, your local machine):

Web service details

Enter the location of the SOAP Web service definition:

Example: <http://www.contoso.com/Service.asmx?WSDL>

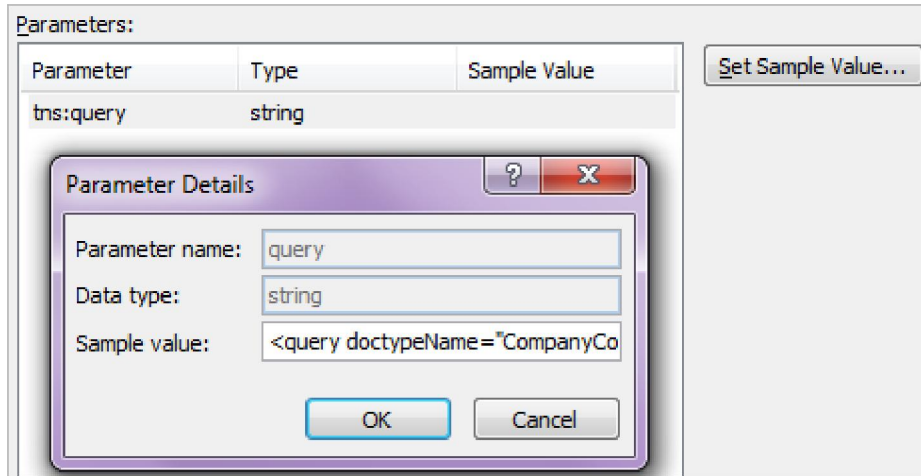
22. Select **QueryDocumentsNodeSet** from the list of operations.

Select an operation:

- GetAllDocumentTypeSummaries
- GetDocumentTypeSummary
- GetDocumentType
- EnumerateDocumentTypes
- EnumerateDocumentTypeNames
- RemoveAllDocumentsForType
- RemoveDocuments
- RemoveDocument
- SubmitDocument
- UnlockDocument
- LockDocument
- GetDocumentFromRefId
- GetDocumentVersion
- GetDocumentString
- GetDocument
- GetDocumentHistory
- QueryDocumentsNodeSet**
- QueryDocumentWithKey

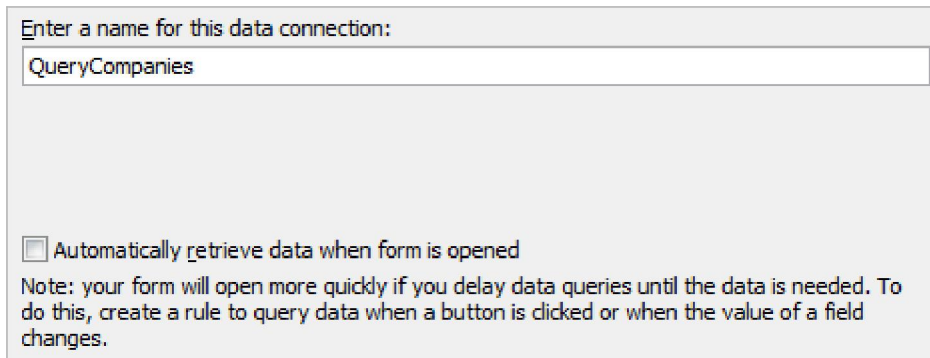


23. Set **Sample Value** of the *tns:query* parameter to the query string generated by Report Builder – remember that you’ve pasted this in Notepad earlier.



The screenshot shows the 'Parameters' dialog box. It contains a table with columns 'Parameter', 'Type', and 'Sample Value'. The first row shows 'tns:query' as a 'string'. To the right is a 'Set Sample Value...' button. Overlaid on this is the 'Parameter Details' sub-dialog box. It has three input fields: 'Parameter name' with 'query', 'Data type' with 'string', and 'Sample value' with '<query doctypeName="CompanyCo'. At the bottom are 'OK' and 'Cancel' buttons.

24. Click **Next** twice; give the data connection a descriptive name and then uncheck the box ‘Automatically retrieve data when form is opened’.



The screenshot shows a dialog box titled 'Enter a name for this data connection:'. It has a text input field containing 'QueryCompanies'. Below the field is an unchecked checkbox labeled 'Automatically retrieve data when form is opened'. A note at the bottom states: 'Note: your form will open more quickly if you delay data queries until the data is needed. To do this, create a rule to query data when a button is clicked or when the value of a field changes.'

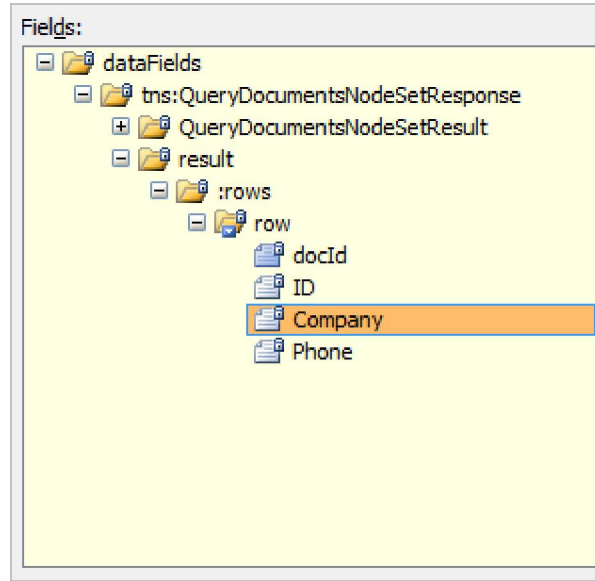
25. Click **Finish > Close** to exit out of the *Data Connections* wizard.
 26. In the *Fields* task pane, add a new field called **Company** – display this on the canvas in a dropdown list box control, like so:



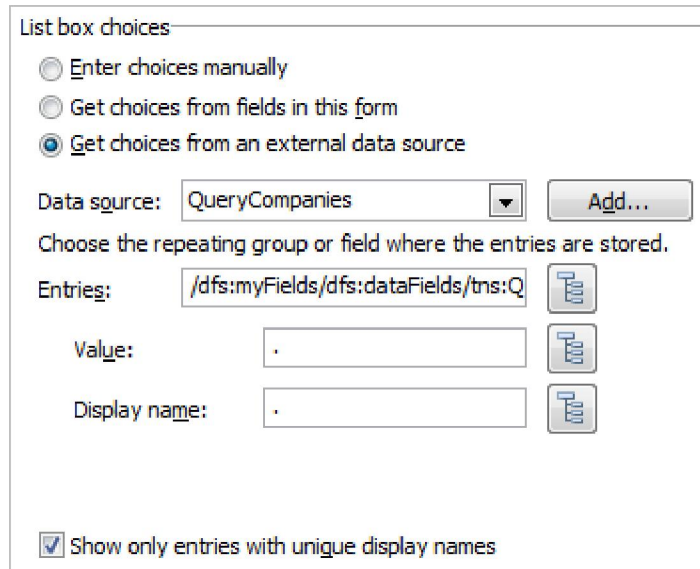
The screenshot shows a 'Sample Form' with a large title. Below the title is a label 'Company:' followed by a dropdown list box. The dropdown box has 'Select...' as the current selection and a small downward arrow on the right.

27. Right-click on the dropdown and select **Drop-Down List Box Properties**.
 28. Change the *List box choices* selection to **Get choices from an external data source**. The data source will default to the one you just added.
 29. Click on the icon next to *Entries* and then expand the nodes until you see the **Company** node – select that.





30. You might want to check the box 'Show only entries with unique display names' in case you have multiple entries with the same value.



List box choices

☐ Enter choices manually

☐ Get choices from fields in this form

☒ Get choices from an external data source

Data source: QueryCompanies Add...

Choose the repeating group or field where the entries are stored.

Entries: /dfs:myFields/dfs:dataFields/tns:Q

Value: .

Display name: .

☒ Show only entries with unique display names

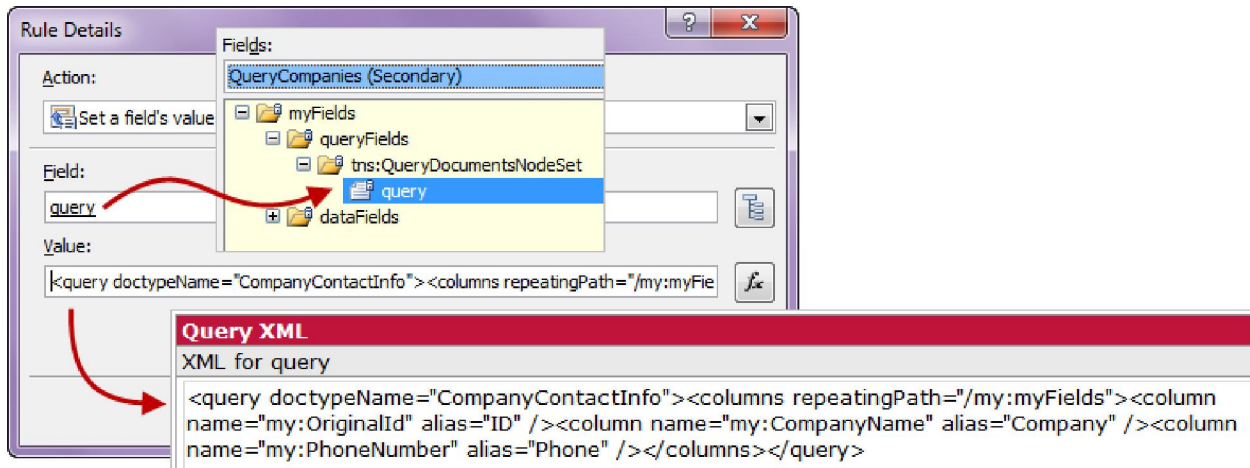
31. Next, we'll add form load rules that first sets the query string and then queries the data connection.
- From the InfoPath Designer menu, go to **Data > Form Load**



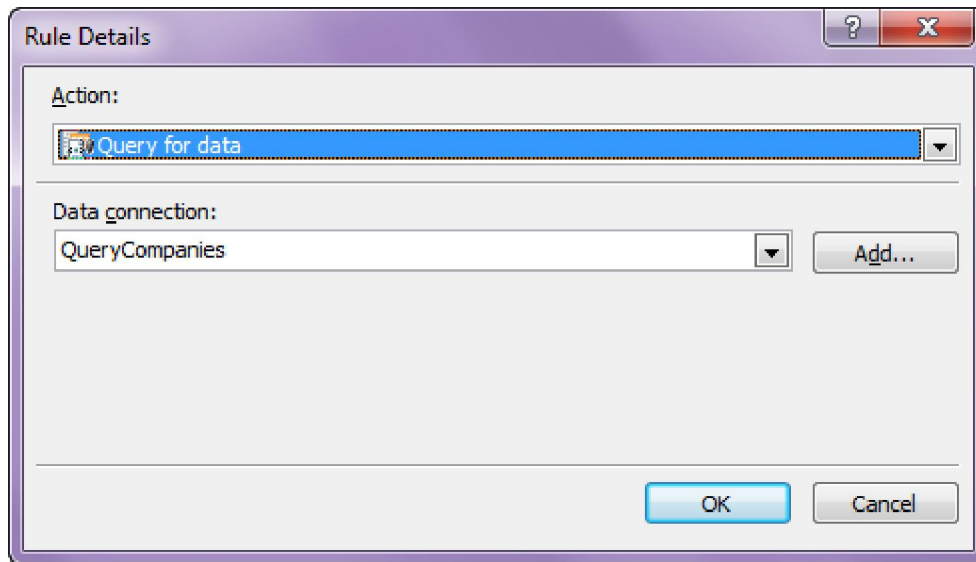
- In the *Rules* task pane that appears, click **New > Action > Add > Set a field's value**.



- c. Allow the action to set the **query** parameter of the *QueryCompanies* secondary data source, to the query string generated by the Report Builder (again, you pasted this in Notepad).



- d. Add another action in the same rule that queries for data in the *QueryCompanies* data connection.



32. Preview your form template – your dropdown should auto-populate with data from your DBXL doctype:



Sample Form

Company:

▼

Acme, Inc.
 The Best Company
 Widgets, Ltd.
 Sprockets Corp.

EXPORTING QUERIES TO EXCEL

In the event that you wish to export queried data to an Excel sheet, the Report Builder tool provides an option to do so by simply clicking the **Export All Queries to Excel** at the result page:

General

Query All

Returned Query Data

Documents

Specify Export Workbook

Export All Queries to Excel

Document Type: CompanyContactInfo Edit Query ▼

docid	ID	Company	Phone
2642	123	Acme, Inc.	090-100-0000
2643	456	The Best Company	123-444-5555
2644	789	Widgets, Ltd.	212-555-1212
2645	122	Sprockets Corp.	999-222-9922

By default, the Excel file is saved under the folder:

C:\Users\[User]\Documents\ReportBuilderOutput

This is a great feature especially if you have to create an outright quick report and work on charts in Excel.

SPECIFYING AN EXPORT WORKBOOK

To specify a workbook to update, the process will require having an existing Excel file to update. For now, let us use the workbook from the last section and save it in a different folder, in this case let's use the desktop location for this scenario.



Now, let's assume that the data in DBXL has been updated – and that there is a need to update the local workbook copy. To update this file, specify the path to the file location, select Sheet Name (Name to be overwritten), and the Query Name which is the name of the Document Type we have queried.

Returned Query Data			
Documents		Specify Export Workbook	Export All Queries to Excel
Workbook Location:			
C:\Users\Dondie\Desktop\ReportBuilderOutput.xlsx			
Sheet Name		Named Range	Query Name
CompanyContactInfo-1			CompanyContactInfo-1
☒ Insert item			
Document Type: CompanyContactInfo			Edit Query
docId	ID	Company	Phone
2642	123	Acme, Inc.	090-100-0000
2643	456	The Best Company	123-444-5555
2644	789	Widgets, Ltd.	212-555-1212
2645	122	Sprockets Corp.	999-222-9922
2655		Qdabra	999-888-7777

The Name Range on the other hand is the Table Name

After specifying the data above, click **Export All Queries to Excel** – this will automatically open up the updated workbook for you:

	A	B	C	D
1	ID	Company	Phone	docId
2	123	Acme, Inc.	090-100-0000	2642
3	456	The Best Company	123-444-5555	2643
4	789	Widgets, Ltd.	212-555-1212	2644
5	122	Sprockets Corp.	999-222-9922	2645
6		Qdabra	999-888-7777	2655

EDITING AND RELOADING A QUERY

In the event that your query needs to be modified, the **Edit Query** button will return you to the Build Query page and allow you to modify your strings and filters in case you need to modify the data returned by adding a filter, changing the sorting, or whatever have you.

Document Type: CompanyContactInfo				Edit Query
docId	ID	Company	Phone	
2642	123	Acme, Inc.	090-100-0000	
2643	456	The Best Company	123-444-5555	
2644	789	Widgets, Ltd.	212-555-1212	
2645	122	Sprockets Corp.	999-222-9922	
2655		Qdabra	999-888-7777	

Once done, simply click the Perform Query button to return to the Results page.

Query XML	
XML for query	
<pre><query doctypeName="CompanyContactInfo"><columns repeatingPath="/my:myFields"><column name="my:OriginalId" alias="ID" /><column name="my:CompanyName" alias="Company" /><column name="my:PhoneNumber" alias="Phone" /></columns><sort><column name="/my:myFields/my:CompanyName" order="ASC" /></sort></query></pre>	Perform Query



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](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
Website: <http://www.qdabra.com>
Community: <http://www.infopathdev.com>

