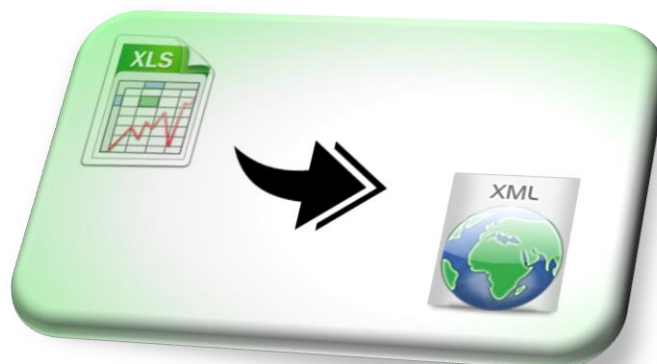




Qdabra DBXL v3.1 Excel to XML Import User Guide



CONTENTS

Overview.....	2
DOCTYPE Configuration	3
Upload your form to DBXL.....	3
The Excel File	4
XLS to XML Import Configuration	6
General Settings	7
XLS to XML Mapping.....	9
Execution History.....	13
Re-running Import	13
Error Logs.....	14
Support.....	15



OVERVIEW

DBXL v3.1's Excel to XML Import feature provides the ability to convert row data from an Excel sheet (XLS file) to an InfoPath form (XML document). The feature uses a mapping skeleton that you specify to convert the Excel data into InfoPath forms.

It can be used on a regular basis to interface data into an InfoPath solution. Many legacy systems support exporting data to Excel format. You can save these Excel files, create a configuration, and set up a task to process the Excel spreadsheets. Since this feature has the ability to check for duplicates, you will have the option to only convert new data rows into XML documents.

Although there is already an existing tool called [Qdabra Excel to InfoPath Migration Tool](#), this new DBXL component eliminates the need for installation of the tool since it already comes as a feature for DBXL v3.1.

You may use this guide as reference on how to configure the Excel spreadsheet mapping for your InfoPath form or follow the whole guide's step by step process by downloading the sample form and spreadsheet from this link:

[DBXL v3.1 XLS to XML sample files](#)

On the sample form provided, notice that there are 3 main fields at the top (**OriginalId**, **CompanyName**, and **PhoneNumber**) along with two repeating sections below. The first three fields will be treated as our Parent data, and the two repeating sections as Child data. We will be creating a Parent-Child type of relationship later on as we move forward.

Another great thing about this feature is that it can support an unlimited amount of nested child repeating sections – if you look a little closer, you will notice that the E-mailAddress field is a repeating node:



<http://www.qdabra.com>

Last updated on 4/29/2017 1:06 AM

Copyright © 2006-2017 Autonomy Systems, LLC. All rights reserved.

DOCTYPE CONFIGURATION

This section assumes that you have DBXL v3.1 installed on your server

UPLOAD YOUR FORM TO DBXL

In this section, you will be uploading your InfoPath form to DBXL as a document type.

1. Open Internet Explorer and access the URL:

<http://<ServerName>/QdabraWebService/default.htm>

2. Once the page opens, click the **DBXL Administration Tool** hyperlink to manage your DBXL instance.
3. Click the **New Configuration** link at the upper right portion of your screen.
4. Click the Browse button and proceed to the folder location where your InfoPath template is located.
5. Click the **Save** button to create the document type.

The screenshot displays the 'Administration Tool' interface for 'DocType: CompanyContactInfo'. The top navigation bar includes links for Home, Catalog, DocType, Tools, and License. A 'Save' button is visible in the top right corner. The interface is divided into several sections:

- Document Type Details:** Contains fields for 'Name' (CompanyContactInfo), 'Solution Group' (a dropdown menu), and 'Description' (a text area). A 'Delete This DocType' link is also present.
- InfoPath Form Template:** Includes an 'XSN File' section with a file explorer showing 'C:\Program Files (x86)' and a 'Browse...' button. Below this are fields for 'Form Version', 'Last Modified', and 'Last Modified By'. To the right, under 'Form Template Properties', there are checkboxes for 'Form template is signed', 'URN-based solution', and 'Enable Document History' (which is checked).
- Ref. ID:** Contains three input fields: 'Format for Ref. ID', 'Base Reference #' (containing '0'), and 'XPath to Inject Ref. ID Into (Leave Blank to Disable)'.
- Indexes:** Includes a table with columns 'Name', 'XPath', and 'Additional Options'. There is a checkbox for 'Enable Xml Indexes (requires extra storage spaces)' and an 'Insert Index' link.
- Form Template Versions:** A section at the bottom with a 'Show Versions' button.

The footer contains the Qdabra logo, copyright information (© 2016 Qdabra Software), a language dropdown set to 'English', and the web service URL: <https://dbxl365.formsboard.com/QdabraWebService/DbxlDocumentService.aspx>. It also shows the DBXL Version (3.1.2017.0104) and the user (NA).



<http://www.qdabra.com>

Last updated on 4/29/2017 1:06 AM

Copyright © 2006-2017 Autonomy Systems, LLC. All rights reserved.

THE EXCEL FILE

For an InfoPath form with a simple xml schema structure – meaning no repeating sections or parent and child relationship data, a single spreadsheet page is enough for conversion.

Here's a sample Excel sheet exported from a SharePoint Library:

1	A	B	C	D	E	F	G	H	I
1	Attachments	ID	Name	Business Unit	Date Assigned	SOURCE	Mfg Location	Description	Notes
2	FALSE	8313	2016-09-12T10_36_44.xml	CONSUMER		CHANGE	BUY/SELL	Instruction sheet update. BOM Item Description Update. Drawing Update.	
3	FALSE	7748	2016-06-06T15_38_27.xml	CONSUMER		CHANGE	SOUTH BEND	Changing all gong brushes to 12 pack	
4	FALSE	8718	2016-11-14T16_49_04.xml	CONSUMER		NEW PRODUCT	SOUTH BEND	Test	
5	FALSE	6168	2015-08-27T12_15_07.xml	CONSUMER		NEW PRODUCT	BUY/SELL		
6	FALSE	5581	2015-05-28T10_11_00.xml	CONSUMER	5/28/2015	NEW PRODUCT	BUY/SELL		
7	FALSE	8942	2017-01-05T09_00_24.xml	ELECTRICAL		CHANGE	REYNOSA ELECTRICAL	VEIT6 - Connector line does not have a "premium look" suitable for retail shelves	Remove clamshell and associated packaging. Update pack drawings with n
8	FALSE	8936	2017-01-03T09_15_01.xml	ELECTRICAL		CHANGE	REYNOSA ELECTRICAL	CHRIS2 - Connector line does not have a "premium look" suitable for retail shelves	Remove clamshell and associated packaging. Update pack drawings with n
9	TRUE	8932	2016-12-18T17_15_29.xml	ELECTRICAL		CHANGE	REYNOSA ELECTRICAL	Correct installation instructions to eliminate customer confusion	should be corrected to read -
10	FALSE	8926	2016-12-20T15_12_06.xml	ELECTRICAL		CHANGE	REYNOSA ELECTRICAL	STEVE5 - Connector line does not have a "premium look" suitable for retail shelves	Remove clamshell and associated packaging. Update pack drawings with n
11	FALSE	8925	2016-12-20T14_35_37.xml	ELECTRICAL		CHANGE	REYNOSA ELECTRICAL	CHRIS2 - Connector line does not have a "premium look" suitable for retail shelves	Remove clamshell and associated packaging. Update pack drawings with n
12	FALSE	8924	2016-12-20T14_26_27.xml	ELECTRICAL		CHANGE	REYNOSA ELECTRICAL	UPDATE DRAWINGS TO NEW CARTON AND NEW 4-WAY WIRE SPECIFICATIONS	UPDATE PACK DRAWINGS AND ADD NEW PRODUCT CARTONS, LABEL, AND S
13	FALSE	8923	2016-12-20T14_13_49.xml	ELECTRICAL		CHANGE	REYNOSA ELECTRICAL	UPDATE DRAWINGS TO NEW CARTON AND NEW 4-WAY WIRE SPECIFICATIONS	CARONT SEALING. UPDATE TOP LEVEL CONNECTOR ASSEMBLY CALLING OUT
14	FALSE	8922	2016-12-20T13_36_48.xml	ELECTRICAL		CHANGE	REYNOSA ELECTRICAL	STEVE4 - Connector line does not have a "premium look" suitable for retail shelves	Remove clamshell and associated packaging. Update pack drawings with n
15	FALSE	8921	2016-12-20T11_17_07.xml	ELECTRICAL		CHANGE	REYNOSA ELECTRICAL	New Generic Split Label is required for new T-One packaging.	Release Generic Split Label 17499-275.
16	FALSE	8912	2016-12-19T10_06_06.xml	ELECTRICAL		CHANGE	REYNOSA ELECTRICAL	STEVE3 - Connector line does not have a "premium look" suitable for retail shelves	
17	FALSE	8911	2016-12-19T10_38_20.xml	ELECTRICAL		CHANGE	REYNOSA ELECTRICAL	New wires needed for new converter boards, 342 and 411	Release 12 AWG wire in black w/ red stripe and the inverse, red with b
18	FALSE	8909	2016-12-19T08_01_02.xml	ELECTRICAL	12/19/2016	CHANGE	REYNOSA ELECTRICAL	UPDATE DRAWINGS TO NEW CARTON AND NEW 4-WAY WIRE SPECIFICATIONS	
19	FALSE	8908	2016-12-19T07_40_12.xml	ELECTRICAL	12/19/2016	CHANGE	REYNOSA ELECTRICAL	UPDATE DRAWINGS TO NEW CARTON AND NEW 4-WAY WIRE SPECIFICATIONS	
20	FALSE	8907	2016-12-18T16_06_02.xml	ELECTRICAL		CHANGE	REYNOSA ELECTRICAL	Garriss2 - T-One line does not have a "premium look" suitable for retail shelves.	
21	FALSE	8906	2016-12-18T15_40_27.xml	ELECTRICAL		CHANGE	REYNOSA ELECTRICAL	VEIT5 - T-One line does not have a "premium look" suitable for retail shelves.	
22	FALSE	8905	2016-12-18T15_16_59.xml	ELECTRICAL		CHANGE	REYNOSA ELECTRICAL	STEVE2 - Connector line does not have a "premium look" suitable for retail shelves	
23	FALSE	8904	2016-12-18T14_54_06.xml	ELECTRICAL		CHANGE	REYNOSA ELECTRICAL	STEVE1 - Connector line does not have a "premium look" suitable for retail shelves.	
24	FALSE	8902	2016-12-18T14_43_34.xml	ELECTRICAL		CHANGE	REYNOSA ELECTRICAL	VEIT4 - Connector line does not have a "premium look" suitable for retail shelves.	
25	FALSE	8901	2016-12-18T14_36_58.xml	ELECTRICAL		CHANGE	REYNOSA ELECTRICAL	VEIT3 - Connector line does not have a "premium look" suitable for retail shelves.	
26	FALSE	8900	2016-12-18T14_29_22.xml	ELECTRICAL		CHANGE	REYNOSA ELECTRICAL	VEIT2 - Connector line does not have a "premium look" suitable for retail shelves.	
27	FALSE	8899	2016-12-18T13_48_02.xml	ELECTRICAL		CHANGE	REYNOSA ELECTRICAL	VEIT1 - T-One line does not have a "premium look" suitable for retail shelves.	
28	FALSE	8893	2016-12-14T15_54_03.xml	ELECTRICAL		CHANGE	REYNOSA ELECTRICAL	Need more new 2-bond wire assemblies of black wire with brown and yellow stripes for 201	
29	FALSE	8892	2016-12-14T15_09_48.xml	ELECTRICAL		CHANGE	REYNOSA ELECTRICAL	Need more new 2-bond wire assemblies of black wire with white and green stripes for 2016	
30	FALSE	8891	2016-12-14T15_09_48.xml	ELECTRICAL		CHANGE	REYNOSA ELECTRICAL	Need more new single green wire assemblies of black wire for 2016 Repack Project.	

The data from the spreadsheet above can be easily mapped in a straightforward manner where rows represent each XML to be generated, and each column header as the schema field element.

Forms with complex schema structures on the other hand will require two or more spreadsheet pages with **key columns** to specify or indicate relationships between multiple spreadsheets (for nested Parent - Child repeating tables).

On sample XLS file provided, the MainValues page's XLS to XML mapping is straightforward and self-explanatory:

1	A	B	C	D	E	F	G
1	ID	Company Name	PhoneNumber				
2	123	Acme, Inc.	090-100-0000				
3	456	The Best Company	123-444-5555				
4	789	Widgets, Ltd.	212-555-1212				
5	122	Sprockets Corp.	999-222-9922				
6							

mvFields

- OriginalId
- CompanyName
- PhoneNumber
- Contacts
 - Contact
 - Name
 - PhoneNumber
 - E-mailAddress
- ProductsOwned
 - Product
 - Name

MainValues | ContactDetails | E-mail Addresses | ProductsOwned



<http://www.qdabra.com>

Last updated on 4/29/2017 1:06 AM

Copyright © 2006-2017 Autonomy Systems, LLC. All rights reserved.

The ID (OriginalId) here is used as an index that should be unique in order to identify if there are duplicate entries – to make it unique, you may use your Library ID if you are exporting data from SharePoint.

The ProductsOwned sheet will require a key column that associates the data from the main page:

	A	B	C	D	E	F	G	H
1	CompanyID	ProductName						
2	123	Ultra Widget						
3	456	Mega Sprocket						
4	123	Power Zapper						
5								
6								

READY

MainValues ContactDetails E-mail Addresses **ProductsOwned**

myFields

- OriginalId
- CompanyName
- PhoneNumber
- Contacts
 - Contact
 - Name
 - PhoneNumber
 - E-mailAddress
- ProductsOwned
 - Product
 - Name

Similarly, the ContactDetails sheet uses the key column for the parent ID, and another ID to associate the E-mail Addresses sheet:

	A	B	C	D	E
1	CompanyID	ID	Name	PhoneNumber	
2	123	12	John Doe	12345	
3	456	13	Jane Doe	67890	
4	123	14	John Wayne	23456	
5	789	15	Jane Wayne	78901	
6					

READY

MainValues **ContactDetails** E-mail Addresses ProductsOwned

myFields

- OriginalId
- CompanyName
- PhoneNumber
- Contacts
 - Contact
 - Name
 - PhoneNumber
 - E-mailAddress
- ProductsOwned
 - Product
 - Name

The E-mail Addresses on the other hand uses the key column (see ID column from above image) to associate itself with the ContactDetails sheet:

	A	B	C	D	E	F	G	H	I
1	PersonID	E-mail Address							
2	12	john@doe.com							
3	12	johndoe@mail.com							
4	13	jane@doe.com							
5	14	johnwayne@yahoo.net							
6									

READY

MainValues ContactDetails **E-mail Addresses** ProductsOwned

myFields

- OriginalId
- CompanyName
- PhoneNumber
- Contacts
 - Contact
 - Name
 - PhoneNumber
 - E-mailAddress
- ProductsOwned
 - Product
 - Name



XLS TO XML IMPORT CONFIGURATION

Now that we have the two major components for the conversion all set, let's proceed with the XLS to XML mapping configuring.

One requirement for the mapping is the Key Info which will be treated as the index.

The screenshot shows the 'Administration Tool' interface for 'DocType: CompanyContactInfo'. The top navigation bar includes 'Home', 'Catalog', 'DocType', 'Tools', and 'License'. The main content area has tabs for 'General', 'Database', 'Permissions', 'Documents', 'Statistics', 'PDF', 'XML', and 'Import'. The 'General' tab is active, showing 'Document Type Details' with fields for 'Name' (CompanyContactInfo), 'Solution Group', and 'Description' (Testing DBXL Migration Tool). Below this is the 'InfoPath Form Template' section, which includes 'XSN File' details (Download, Upload New, Form Version: 1.0.0.12, Last Modified: 1/05/2017 5:06:13, Last Modified By: DBXL3-365\dbxladmin) and 'Form Template Properties' (URL: https://dbxl365.formsboard.com/QdabraWebService/Forms/ID/331/template.xsn, checkboxes for 'Form template is signed', 'URN-based solution', and 'Enable Document History'). The 'Ref. ID' section has fields for 'Format for Ref. ID', 'Base Reference #', and 'XPath to Inject Ref. ID Into'. The 'Indexes' section is highlighted with a green border and contains a table with columns 'Name', 'XPath', and 'Additional Options'. The 'Form Template Versions' section is at the bottom with a 'Show Versions' checkbox. The footer includes the Qdabra DBXL logo, copyright information (© 2016 Qdabra Software), and a language dropdown set to 'English'.

6. At the **General** settings page, click **Insert Index** for the *Indexes* section.
7. Create a meaningful name for the Index Name such as **DBXLID**.

This screenshot shows the 'Indexes' section of the 'Administration Tool' interface. It features a table with columns 'Name', 'XPath', and 'Additional Options'. The 'Name' field contains 'DBXLID', the 'XPath' field contains 'my:myFields/my:OriginalId', and the 'Additional Options' field has a checkbox for 'Enable Xml Indexes (requires extra storage spaces)'. Below the table is an 'Insert Index' link. The 'Form Template Versions' section is at the bottom with a 'Show Versions' checkbox. The top right corner of the section has an 'Update' button. The footer includes the Qdabra DBXL logo, copyright information (© 2016 Qdabra Software), and a language dropdown set to 'English'.



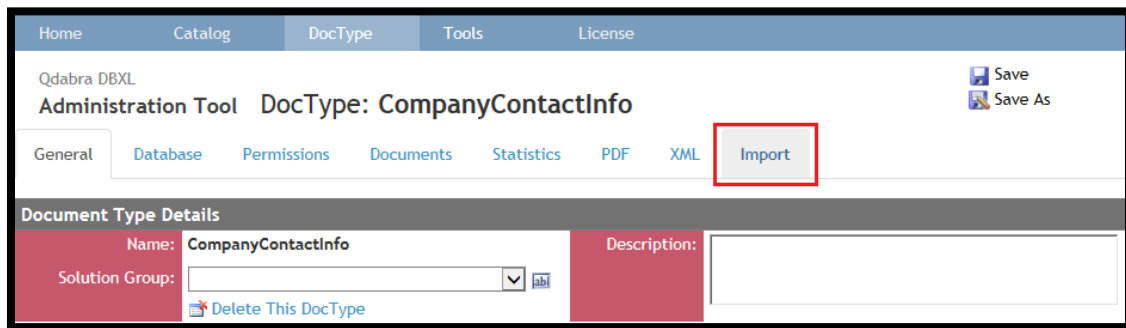
<http://www.qdabra.com>

Last updated on 4/29/2017 1:06 AM

Copyright © 2006-2017 Autonomy Systems, LLC. All rights reserved.

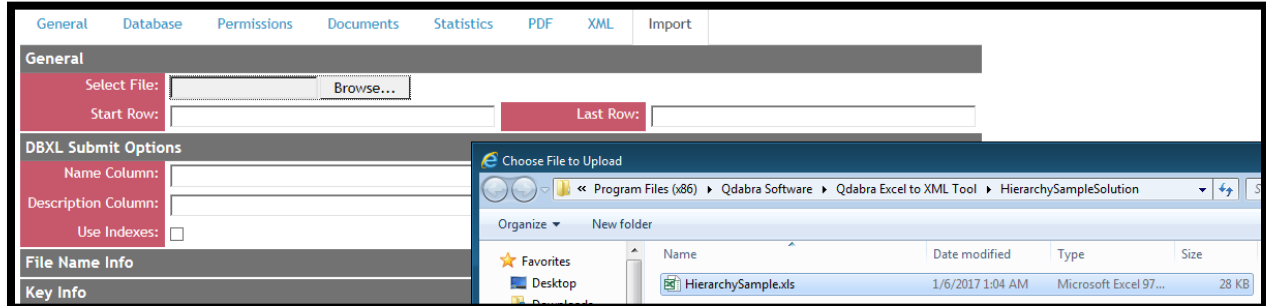
Since this is the unique index number, it makes sense to use the xpath of the OriginalId field.

8. Open the **CompanyContactInfo** xsn template in design mode (right click > select design).
9. Right click **OriginalId** and click properties
10. Select **Copy XPath**.
11. Paste the copied value onto the XPath field and click **Save**.
12. After saving your document type, a set of new tabs will be displayed. Click the **Import** button tab to open up the XLS to XML Import page.

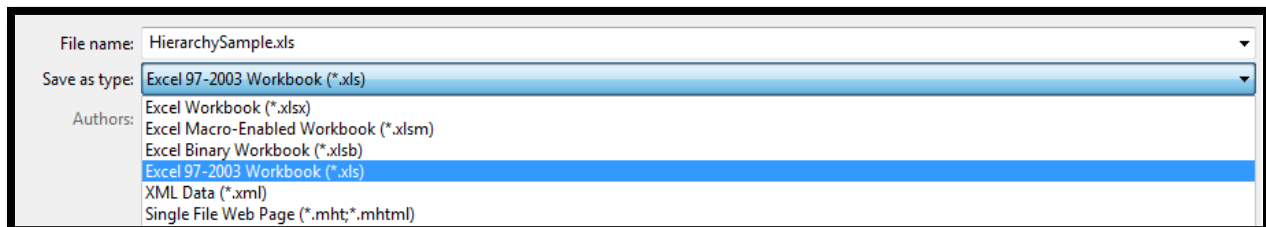


GENERAL SETTINGS

13. At the **General** settings section, browse for the Excel spreadsheet file and upload.



Note that your spreadsheet must be in XLS format (not XLSX).



14. **StartingRow:** The row of the Excel file where processing should start. In the case of repeating data, this only applies to the main spreadsheet specified in the top-level mapping.

Since the header is on the first row, it makes sense to set the **Start Row** to the value 2.

15. **LastRow:** This optional field can be used to specify a stopping row if needed. Leave the default value as '0' if you don't wish to specify the last row – this would generate XMLs for all the rows in the Excel sheet.
16. **Use Indexes:** Set this to true to use doctype indexes (configured in DAT) to identify forms that have already been submitted to DBXL. If this is false, this tool will use a slower method that doesn't require indexes to be configured.
17. To allow overwriting existing documents in DBXL, you must set **Allow Document Overwrite** to **true**, and provide an xpath filter. This xpath is used to identify documents that have already been submitted.

Sample xpath:

```
/my:myFields[my:OriginalId = '{{0}}']
```

18. **Name, Author, and Description** are optional but very useful – the data here is displayed on the *Documents* tab page to help identify submitted documents in a quick glance.
19. **(Optional): Doc Filter XPath** is required if **Use Indexes** is false. It is an XPath to identify documents that have already been submitted, and produce no match for those that haven't.

Here's how your configuration may look like at this point:

General	
Select File:	abra Software\Qdabra Browse...
Start Row:	2
Last Row:	0
DBXL Submit Options	
Name Column:	Company Name
Description Column:	
Use Indexes:	☒
Author Column:	ID
Doc Filter XPath:	
Allow Overwrite:	☒



20. For the Key Info section, specify the **Index Name** (the **DBXLID** which was declared earlier), along with the associated ID from the spreadsheet.

- If *Use Indexes* is true, *Index Name* should indicate the name of the index (*DBXLID*), and *Column Name* should indicate the name of a column in the main spreadsheet that corresponds to the field stored in that index.
- If *Use Indexes* is false, *Column Name* should indicate a column in the main spreadsheet whose will be used in the <Doc Filter XPath> format string. *Index Name* can be blank in this case (see step 17).

Key Info	
Index Name	Column Name
DBXLID	ID

The indexes used by the Excel to XML tool (above) must match the indexes set up in the DocType's General Settings page (below):

Indexes	
Name:	XPath:
DBXLID	my:myFields/my:OriginalId
☐ Enable Xml Indexes (requires extra storage spaces)	
Insert Index	
Form Template Versions	

XLS TO XML MAPPING

In this section each Excel spreadsheet column is mapped to an XML field.

21. **Parent Sheet:** This section specifies the mapping to the root node in the target XML and the parent sheet in the Excel file.
- Root:** The XPath of a node in the target XML, relative to which all other XPaths will be evaluated.
 - Sheet Name:** Should indicate the main (top level) spreadsheet of the Excel file.

Mapping	
Sheet Name:	Root Name:
MainValues	my:myFields



- c. **Fields:** The list of fields from the main spreadsheet that should be inserted into the generated XML forms.
- i. **Column Name** – the name of a column in the main spreadsheet
 - **Node XPath** – the path to the corresponding XML field (relative to the *Root* node specified above).
 - **Data Type** – data type of the corresponding field in the form.
 - **Nullable** – uncheck this box if the corresponding field 'cannot be blank'.
 - **Special** – used for fields that use InfoPath functions such as *today()*.

Fields					Add Field
Column Name	Node XPath		Data Type	Nullable	Special Value
ID	my:OriginalId		string	☒	☐
Company Name	my:CompanyName		string	☒	☐
PhoneNumber	my:PhoneNumber		string	☒	☐

22. **Child Sheet:** Indicates mappings to child spreadsheets in the target Excel file, to be mapped to the parent spreadsheet and into the XML forms. This section is optional.

- a. **ParentKeys** - This section should contain one or more nodes.
- i. **Child Field** – the name of a column in this child spreadsheet, that has values that correspond to values in a column of the parent spreadsheet (indicated with the *Parent Field* node)
 - ii. **Parent Field** – the name of the column in the parent spreadsheet that defines the relationship of this child sheet to the parent sheet.

Child Sheet		Add Parent Keys
Child Field	Parent Field	
CompanyID	ID	

Follow the same guidelines as above when adding the Mapping info for the child sheets.

Mapping		Add Child Mapping
Sheet Name:	ProductsOwned	Root Name: my:myFields/my:ProductsOwned/my:Produc 
Fields		
Column Name	Node XPath	
ProductName	my:Name	
		string ☒ ☐



Repeat same pattern as above when adding child sheets. Child mappings can be nested inside other child mappings to handle parent -> child -> grandchild [-> great grandchild, etc] relationships.

The screenshot displays two overlapping 'Child Sheet' configuration windows in the Qdabra application. Each window has a header bar with the sheet name and a close button. Below the header, there are sections for 'Parent Keys', 'Child Field', 'Mapping', and 'Fields'.

Top Window (ContactDetails):

- Parent Keys:** 'Add Parent Keys' button.
- Child Field:** 'CompanyID' (Child Field) mapped to 'ID' (Parent Field).
- Mapping:** 'Sheet Name: ContactDetails', 'Root Name: my:myFields/my:Contacts/my:Contact'.
- Fields:**

Column Name	Node XPath	Data Type	Nullable	Special Value
Name	my:Name	string	☒	☐
PhoneNumber	my:PhoneNumber	string	☒	☐

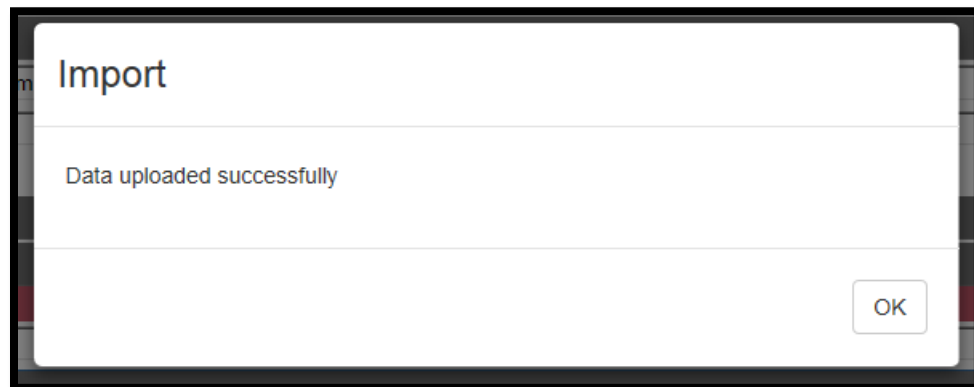
Bottom Window (E-mail Addresses):

- Parent Keys:** 'Add Parent Keys' button.
- Child Field:** 'PersonID' (Child Field) mapped to 'ID' (Parent Field).
- Mapping:** 'Sheet Name: E-mail Addresses', 'Root Name: Contacts/my:Contact/my:E-mailAddress x'.
- Fields:**

Column Name	Node XPath	Data Type	Nullable	Special Value
E-mail Address	.	string	☒	☐

When done, click the **Submit** button to generate the XML files.

You should receive a confirmation message that the conversion was successful.



Proceed to the Documents page by clicking the Document tab and verify that you have your XMLs generated:

General
Database
Permissions
Documents
Statistics
PDF
XML
Import

Query Documents

To limit the number of documents shown, enter a starting Doc. ID and the number of documents to show, and then click query. You can then jump forward or backward by the number of shown documents, or click Query All to show all documents. To search for a Document with a specific Ref. ID, type the ID into the field, and then click Query.

Starting Doc. ID:
Range:
Ref. ID:

Query
Query All
Previous
Next

Actions on Documents

Select documents for Reshred and Delete.
If reshredding more than 100 documents, please use the DBXL Migration Tool.

Reshred
Delete
Reshred All
Delete All
Refresh
New Doc

Documents

Total Documents: 4

	Doc. ID	Ref. ID	Version #	Name	Author	Description	Open	History
☐	26294		1	Acme, Inc.	123		Open	History
☐	26295		1	The Best Company	456		Open	History
☐	26296		1	Widgets, Ltd.	789		Open	History
☐	26297		1	Sprockets Corp.	122		Open	History

Unlike the [Qdabra Excel to XML Migration Tool](#), this feature currently only offers one output location which is directly to DBXL.

If you wish to export the XML files to a local folder or directly upload to SharePoint, kindly download the tool from the link provided above or use the [DBXL Migration Tool \(Document Deployer\) v2.8](#) for downloading XMLs from DBXL.



<http://www.qdabra.com>

Last updated on 4/29/2017 1:06 AM

Copyright © 2006-2017 Autonomy Systems, LLC. All rights reserved.

EXECUTION HISTORY

Errors and XLS sheet file versions are logged into the Execution history.

General Database Permissions Documents Statistics Email XML Import Query Builder

Use this tab to create documents for this docType using XLS file.

Import Data Save Mapping Show Mappings

Execution History				
Details	Date	Start Time	End Time	Imported By
Details	4/27/2017	4/27/2017 8:10:55 PM	4/27/2017 8:10:56 PM	DBXL3-365\dbxladmin
Details	4/27/2017	4/27/2017 8:13:20 PM	4/27/2017 8:13:20 PM	DBXL3-365\dbxladmin
Details	4/27/2017	4/27/2017 8:13:45 PM	4/27/2017 8:13:46 PM	DBXL3-365\dbxladmin
Details	4/27/2017	4/27/2017 8:14:52 PM	4/27/2017 8:14:52 PM	DBXL3-365\dbxladmin
Details	4/27/2017	4/27/2017 8:20:18 PM	4/27/2017 8:20:18 PM	DBXL3-365\dbxladmin
Details	4/27/2017	4/27/2017 8:22:06 PM	4/27/2017 8:22:06 PM	DBXL3-365\dbxladmin
Details	4/27/2017	4/27/2017 8:24:32 PM	4/27/2017 8:24:33 PM	DBXL3-365\dbxladmin
Details	4/27/2017	4/27/2017 8:28:53 PM	4/27/2017 8:28:54 PM	DBXL3-365\dbxladmin
Details	4/27/2017	4/27/2017 11:16:46 PM	4/27/2017 11:16:47 PM	DBXL3-365\dbxladmin
Details	4/27/2017	4/27/2017 11:38:31 PM	4/27/2017 11:38:31 PM	DBXL3-365\dbxladmin

Previous 1 2 Next

RE-RUNNING IMPORT

In the event that a user needs to update the XLS sheet but has no access to it, versions of the imported XLS file are saved and can be downloaded from the Execution History by clicking on the [Details](#) link of the desired version.

This should open up a new view – click [Download](#) to extract the XLS file:

Home	Catalog	DocType	Tools	License
Back to DocType				
Execution Details				
File	Download			
Date	4/27/2017			
Start Time	4/27/2017 11:41:23 PM			
End Time	4/27/2017 11:41:24 PM			
Invalid Count	0			
Existing Count	0			
Inserted Count	4			
Imported By	DBXL3-365\dbxladmin			

You should now be able to update the XLS version and re-run Import.



<http://www.qdabra.com>

Last updated on 4/29/2017 1:06 AM

Copyright © 2006-2017 Autonomy Systems, LLC. All rights reserved.

ERROR LOGS

Still on the same page, errors are being logged to help identify mapping configuration inaccuracies.

Execution Errors
Description
An unhandled exception was thrown: No node with the relative path 'my:myFields/my:ProductsOwned/my:Product' could be found to perform the mapping for the ProductsOwned sheet
An unhandled exception was thrown: No node with the relative path 'my:myFields/my:ProductsOwned/my:Product' could be found to perform the mapping for the ProductsOwned sheet
An unhandled exception was thrown: No node with the relative path 'my:myFields/my:ProductsOwned/my:Product' could be found to perform the mapping for the ProductsOwned sheet
An unhandled exception was thrown: No node with the relative path 'my:myFields/my:ProductsOwned/my:Product' could be found to perform the mapping for the ProductsOwned sheet



SUPPORT

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>



<http://www.qdabra.com>

Last updated on 4/29/2017 1:06 AM

Copyright © 2006-2017 Autonomy Systems, LLC. All rights reserved.