



Getting Started with DBXL v3.2

This document will allow you to quickly get started with DBXL v3.2. After completing this document you will be able to setup your own forms in DBXL and connect them to SQL and SharePoint.

Starting with DBXL v2.8, Qdabra recommends using the **Event Receiver**. The InfoPath form template submits to a SharePoint form library, and the Event Receiver maps those forms to DBXL.

While there are additional scenarios that can be achieved with DBXL, this tutorial will focus on the Event Receiver scenario. Please refer to the [DBXL Deployment Guide](#) for information about other scenarios.

This tutorial uses the Expense Report sample form template that is included in all InfoPath 2007 installations. Although the Expense Report sample form was designed for InfoPath 2007 it will work with later versions of InfoPath.

Requirements:

- **DBXL v3.2**
- SharePoint 2010, 2013, or Office 365. This scenario is most important for SharePoint 2013 and Office 365. However, the tutorial in this document will focus on SharePoint 2013, and the steps should be similar if working on SharePoint 2010. The Event Receiver has not been tested in SharePoint 2003 or 2007. For Office 365 on Azure, kindly refer to the installation guides below:
 - [DBXL Installation](#)
 - [Event Receiver with DBXL](#)
- Microsoft InfoPath. This tutorial will use InfoPath 2013, but similar steps may be used in previous versions.
- Remote access to the server where DBXL is installed and to the SharePoint server.
- Access to SharePoint Central Administration.
- SQL Server Management Studio and access to the SQL instance.



CONTENTS

Make sure you're a DBXL Administrator	3
Add the site to Local Intranet	4
Set up the form and the Form Library	5
Set up the DBXL Document Type.....	7
Create A Secure Store Application ID	8
Permissions in SharePoint Foundation 2013	Error! Bookmark not defined.
Install the Event Receiver	10
Activate the Event Receiver on the site.....	13
Form Library Settings.....	14
Create a SQL table	16
Add Database mapping.....	17
Expand the Database mapping	19
Activate error notifications.....	21
What's next?.....	23
Appendix: Troubleshooting the Event Receiver	24



MAKE SURE YOU'RE A DBXL ADMINISTRATOR

To successfully complete the steps in this document, you must be a DBXL Administrator. The following steps will show you how to check whether you are an administrator, and add yourself if you are not.

1. Use Remote Desktop to connect to the machine where DBXL is installed.
2. Check the DBXL web.config file.
 - a. In the **Qdabrawebservice** folder, find the **web.config** file. The QdabraWebService folder is located, usually, in your SharePoint-80 site. For example:
C:\inetpub\wwwroot\wss\VirtualDirectories\80\QdabraWebService
 - b. Open the **web.config** file in a text editor, such as **Notepad**.
 - c. Search for the following: **AdminGroupAlias** and **DbxlAdminAlias**. You'll find a section that looks like this:

```
<add key="BrowserGroupAlias" value="NT AUTHORITY\Authenticated Users" />
<add key="ReaderGroupAlias" value="" />
<add key="WriterGroupAlias" value="" />
<add key="AdminGroupAlias" value="<machine name>\DbxlAdmins" />
<add key="DbxlAdminAlias" value="<machine name>\DbxlAdmins" />
```

Each of these keys is assigned a value (a group). During installation, DBXL creates a group called **DBXLAdmins** on the server. This group, by default, is assigned to both keys: **AdminGroupAlias** and **DbxlAdminAlias**.

3. To add a DBXL Administrator all you need to do is add users to the **DbxlAdmins** group.
 - a. Open **Computer Management**.
 - b. Under **System Tools > Local Users and Groups**, locate the **DbxlAdmins** group. Right-click to see its **Properties**.
 - c. Add yourself to this group.

Leave the Remote Desktop Connection open. We will use it later.



ADD THE SITE TO LOCAL INTRANET

Before we continue, we will add the server to Internet Explorer's Local Intranet zone.

4. In Internet Explorer, go to **Internet Options**, switch to the **Security** tab, select **Local intranet** and then click **Sites**.
5. In the **Local intranet** window, click **Advanced**.
6. Enter the site where DBXL is installed (e.g. <http://<servername>/>) and then click **Add**.

The site should now be listed amongst the local intranet websites.

7. Click **Close**, and then click **OK** twice to close all the dialogs.



SET UP THE FORM AND THE FORM LIBRARY

The basis for this scenario is that the form will submit directly to a form library. These steps will set this up.

8. Create a new form library. This can be accomplished by going to **Site Contents > add an app > Form Library**. Then provide a name for the form library and click **Create**.

9. Open the InfoPathExpenseReport.xsn form template in Design mode.
10. Go to **Data > Data Connections** and remove the **Main submit** data connection.
11. Click **Add > Submit Data** then click **Next**.
12. Select "To a document library on a SharePoint site" and click **Next**.
13. Enter the URL to the form library created earlier. For the **File name**, click **fx** to select the Expense Code node. Check "Allow overwrite if file exists".
14. Click **Next**, then click **Finish**, then click **Close**.
15. Click **File > Save** to save the form template.

Click **File > Publish > SharePoint server**. Walk through the InfoPath publishing wizard to publish the form to the form library created earlier.

16. In InfoPath, go to **File > Save As**. Save the form template to your local drive, making sure to select the correct version (depending on your environment).

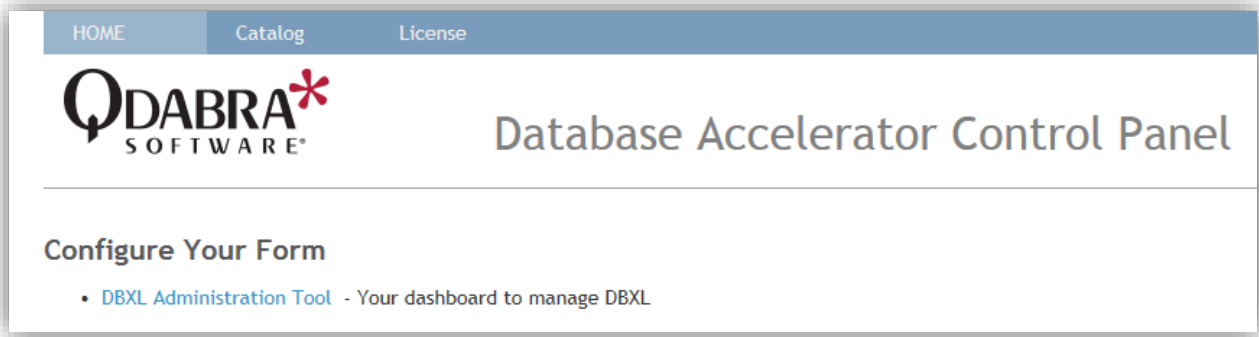


File name:	Expense Report.xsn
Save as type:	InfoPath 2010 Filler Form Template (*.xsn)
	InfoPath Web Browser Form Template (*.xsn)
	InfoPath Filler Form Template (*.xsn)
	InfoPath 2010 Web Browser Form Template (*.xsn)
	InfoPath 2010 Filler Form Template (*.xsn)
	InfoPath 2007 Web Browser Form Template (*.xsn)
	InfoPath 2007 Filler Form Template (*.xsn)
	InfoPath 2003 Form Template (*.xsn)



SET UP THE DBXL DOCUMENT TYPE

17. Using Internet Explorer, go to <http://<servername>/QdabraWebService/>.
18. Click on the first link, **DBXL Administration Tool**, to launch DAT.



19. Once DAT opens in InfoPath, click **New Configuration**.
20. In the **General** tab, enter a name under **Document Type Details > Name**, and attach the XSN form template.

Note the name you use in this step. You will use it later on.

21. Click **Save** and then click **OK** in the confirmation dialog.



<http://www.qdabra.com>

Last updated on 5/8/2017 1:01 PM

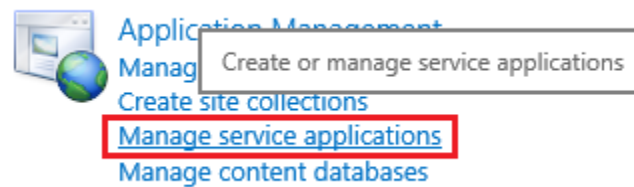
Copyright © 2006 - 2017 Autonomy Systems, LLC. All rights reserved.

CREATE A SECURE STORE APPLICATION ID

To create the Secure Store Application ID, the user must have access to SharePoint Central Administration.

Note: If you are using SharePoint Foundation 2013, the Secure Store Service is not available. Please skip to the next section.

22. Go to SharePoint Central Admin then click on **Manage Service Applications**.



23. Click on **Secure Store Service Application**.

24. If accessing for the first time, click on 'Generate New Key' on the ribbon and enter pass phrase as you like. If not accessing for the first time, click **Manage** in the ribbon.

25. Click **New** on the ribbon.

26. Enter the required values for the target application settings. Click **Next**.

Target Application Settings

The Secure Store Target Application ID is a unique identifier. You cannot change this property after you create the Target Application.

The display name is used for display purposes only.

The contact e-mail should be a valid e-mail address of the primary contact for this Target Application.

The Target Application type determines whether this application uses a group mapping or individual mapping. Ticketing indicates whether tickets are used for this Target Application. You cannot change this property after you create the Target Application.

The Target Application page URL can be used to set the values for the credential fields for the Target Application by individual users.

Target Application ID

Display Name

Contact E-mail

Target Application Type

 Group

Target Application Page URL

 Use default page

 Use custom page

 None

27. Use the default (Windows Username and Windows Password). Click **Next**.

28. Specify the Administrator for the target application. You can specify Members for the target application as well if you like to. Click **Ok**.



<http://www.qdabra.com>

Last updated on 5/8/2017 1:01 PM

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

Target Application Administrators

The list of users who have access to manage the Target Application settings. The farm administrator will have access by default.

Don Lambatin; Jim Cantwell; petera

Users who have Full Control or All Target Applications privileges can administer this Secure Store Target Application.

Members

The users and groups that are mapped to the credentials defined for this Target Application.

Everyone; All Users (windows)

29. Select the new AppID and click **Set Credentials**.

The screenshot shows the SharePoint ribbon with the 'EDIT' tab selected. The 'Credentials' group is active, and the 'Set' button is highlighted. A tooltip for the 'Set' button reads: 'Click here to set the Credential Field values for the selected Secure Store Target Application.' Below the ribbon, a list of target applications is visible, with 'DBXL' selected.

Group	Item	Selected
Central Administration	Target Application	<input type="checkbox"/>
	Cas	<input type="checkbox"/>
	DBXL	<input checked="" type="checkbox"/>
	DBXL0365	<input type="checkbox"/>
	DBXL2	<input type="checkbox"/>
	DBXL365	<input type="checkbox"/>

In the dialog, enter the credentials to be used by this Application ID. The credentials provided must have access to DBXL. If you wish to control user access, you can use SharePoint form library permissions to prevent users from accessing or submitting.



UNINSTALLING PREVIOUS EVENT RECEIVER

Note: this section is optional and only applies for users with prior installations.

To remove the WSP, please execute:

Uninstall-SPSolution Qdabra.Dbxl.SharePoint.wsp

Once this has finished, also execute:

Remove-SPSolution Qdabra.Dbxl.SharePoint.wsp



<http://www.qdabra.com>

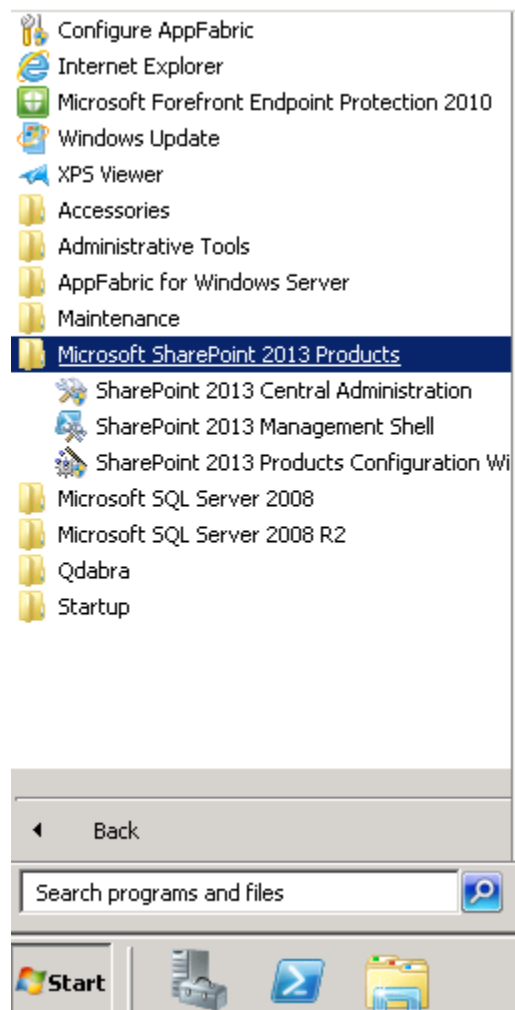
Last updated on 5/8/2017 1:01 PM

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

INSTALL THE EVENT RECEIVER

The following steps need to be executed on the SharePoint server and require the latest Event Receiver WSP file, which is located on the server where DBXL is installed, in the path: **<INSTALLATION PATH>\QdabraWebService\extras**. You can also download the Event Receiver from [here](#).

30. Connect to the SharePoint server via Remote Desktop Connection.
31. Copy the WSP file into a convenient location, such as C:\DBXL
32. Go to the **Start Menu > All Programs > Microsoft SharePoint 2013 Products**.
33. Right click on **SharePoint 2013 Management Shell** and select **Run As Administrator**.



34. Add the WSP file. This step requires that you provide the full path:



<http://www.qdabra.com>

Last updated on 5/8/2017 1:01 PM

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

Add-SPSolution C:\<path>\Qdabra.Dbxl.SharePoint.wsp

35. Deploy the WSP file.

**Install-SPSolution -Identity Qdabra.Dbxl.SharePoint.wsp -GACDeployment -CompatibilityLevel All -
WebApplication <SharePoint site URL>**



<http://www.qdabra.com>

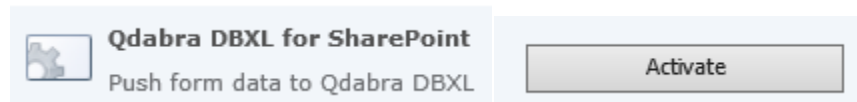
Last updated on 5/8/2017 1:01 PM

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

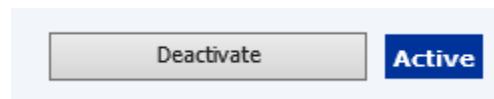
ACTIVATE THE EVENT RECEIVER ON THE SITE

These steps can be executed from a client machine, by a user who is a site collection administrator.

36. Use Internet Explorer to navigate to the SharePoint site where your form library is located, and access **Site Actions > Site Settings**.
37. Under **Site Collection Administration**, click on **Site collection features** or **Site features** (http://<servername> /<site>/_layouts/ManageFeatures.aspx).
38. Find “Qdabra DBXL for SharePoint” and click **Activate**.



Once activated, the available buttons will change to:



This step needs to be executed on every SharePoint site where the Event Receiver will be used.



FORM LIBRARY SETTINGS

Read / Contribute permissions to the form library are not enough to access the Form Library settings, and therefore not enough to set up this feature. **Design** is the minimum permission level required. **Full Control** would also be enough.

39. Go to **Form Library > Library Settings**. Under **General Settings**, find **DBXL Settings**.

40. Fill out the fields on this page. See the table below for details on each of the available settings.

DBXL Service	
DBXL URL	The URL to DBXL (QdabraWebService) on your server. Example: <i>http://sharepoint.domain.com/QdabraWebService</i>
DocumentType Name	The name of the Document Type in the DBXL Admin Tool. Example: <i>QdExpenseReport-Sample</i>
Conditional Submit <i>(optional)</i>	An expression that defines which documents will be submitted to DBXL. Examples: <i>/my:expenseReport/my:employee/my:name = "Ernesto"</i> <i>/my:expenseReport/my:reportDate = '2012-08-22'</i> <i>/my:expenseReport/my:items/my:total > 10</i> Note that date comparisons are not supported: <i>/my:expenseReport/my:reportDate > 2013-01-01</i> <i>substring(/my:expenseReport/my:reportDate) = "2013"</i>
Document Settings	
Name xpath <i>(optional)</i>	The xpath in the InfoPath Form Template (XSN) that will be used for the DBXL::Name Property in the DBXL Admin Tool. Note that this xpath is not validated by the webpage. Example: <i>/my:myFields/my:CustomerName</i>
Author xpath <i>(optional)</i>	The xpath in the InfoPath Form Template (XSN) that will be used for the DBXL::Author Property in the DBXL Admin Tool.
Description xpath <i>(optional)</i>	The xpath in the InfoPath Form Template (XSN) that will be used for the DBXL::Description Property in the DBXL Admin Tool.
Credentials	
Secure Store Application ID	Created in a previous task of this tutorial. <i>Please leave blank to use the SharePoint app pool account, especially when using SharePoint Foundation 2013. If you are a farm administrator, the app pool account will be shown here.</i>

Secure Store Application ID

Leave blank to use the SharePoint app pool account
(NT AUTHORITY\NETWORK SERVICE)



Notifications	
Email notifications to	Use the Person/Group Picker to select the user or users that will receive an email notification whenever there is an error when submitting data to DBXL, and whenever there is a Synchronize operation.
Enable DBXL Submit	
Enable DBXL submit	Enable (or disable) submitting to DBXL by checking (or unchecking) this checkbox.
Content Change Detection	
Force DBXL submit	The DBXL event receiver, by default, will ignore item update events where the XML content does not change. For example, when a workflow changes the workflow status column, SharePoint fires an update event. DBXL detects the content is unchanged since the last submit, and will ignore the update event. But in some cases, if the DBXL content was modified externally, you may want to force the XML content to be submitted to DBXL. Check this box to submit to DBXL even if it is determined that the XML content has not changed.
Buttons	
Synchronize	Click this button to “push” the latest SharePoint Form Library data to DBXL. Clicking Synchronize DOES NOT test the current settings and DOES NOT save any changes made to the settings. All Synchronize operations will send email to the users listed in the Notifications section.
Save & Test	Click this button to test and save the current settings.
Ok	Click this button to save and return to the Form Library Settings page. This button DOES NOT test the current settings.
Cancel	Click this button to cancel and return to the Form Library Settings page.

41. Click **Save & Test** to test your settings, or click **OK** to save the settings.

42. If you have existing forms in this form library, you can click **Synchronize** on the DBXL Settings page to “push” the data to DBXL.

That’s it! Now you can submit forms to the form library and the forms will show up in the Documents tab of your document type in the DBXL Admin Tool.

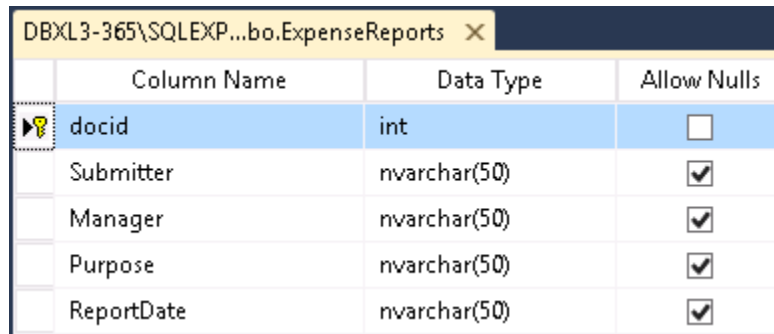



CREATE A SQL TABLE

The following steps require SQL Server Management Studio.

43. Design the database

- a. Use SQL Server Management Studio to connect to the SQL instance you wish to store the form's data.
- b. Create a database and a table to store your data. In this tutorial, we create a database named **SampleDB** that contains the **ExpenseReports** table.



	Column Name	Data Type	Allow Nulls
	docid	int	<input type="checkbox"/>
	Submitter	nvarchar(50)	<input checked="" type="checkbox"/>
	Manager	nvarchar(50)	<input checked="" type="checkbox"/>
	Purpose	nvarchar(50)	<input checked="" type="checkbox"/>
	ReportDate	nvarchar(50)	<input checked="" type="checkbox"/>

Notice that the **docid** column is the table's Primary Key.



ADD DATABASE MAPPING

Now that the table is ready, we can proceed to create the database mapping in the DBXL Administration Tool (DAT) to enable shredding.

44. Configure a database connection
 - a. In DAT, click on the **Database** tab.
 - b. Enter the Data Connection String appropriate for your SQL server. For example, *Integrated Security=SSPI;Data Source=(LOCAL)\sqlexpress;Initial Catalog=Your_Database*
 - c. Tab out of the **Connection String** field to make the **Test Connection** button to appear. Click **Test Connection**, and click **OK** when success is confirmed.
 - d. Click **Save**, and then click **OK** in the confirmation dialog.

45. Configure the table mapping.
 - a. Under **Database Map** section, click **Add Table**.
 - b. In the **Table Name** dropdown, select the name of the **ExpenseReports** table.
 - c. Click on the **Select Schema Node** icon found next to the **Node Path** field. This will load the form's XML schema in the Task Pane.
 - d. Double-click on the **my:expenseReport** node.

46. Add the columns to the mapping.
 - a. Click the **Add Column** link.
 - b. From the Column Name dropdown, select the **Manager** column in the SQL table.
 - c. Click on the **Select Schema Node** icon and double click on the **my:manager/my:managerName** node.

Repeat the steps above to add all columns. From the screenshot below, note that the docid column is mapped to the DBXL::DocID.



Database Configuration					
Connection String:					
Data Source=DBXL3-365\SQLEXPRESS;Initial Catalog=SampleDB;User ID=dbxadmin;Password=dbxadmin					Test Connection
Submit Options:			Database Actions:		
<input checked="" type="checkbox"/> Report Shredding Errors to the Form User			<input type="button" value="Reshred All Documents"/> <input type="button" value="Refresh Database"/>		
Database Mapping					
SQL Side	Template Sides				Del
Table Name(Columns)	Column Name	Type	Key	Node Path	
ExpenseReports	Add Sub Table				/my:expenseReport
	Manager	String	<input type="checkbox"/>	my:manager/my:managerName	
	Purpose	String	<input type="checkbox"/>	my:purpose	
	ReportDate	String	<input type="checkbox"/>	my:reportDate	
	Submitter	String	<input type="checkbox"/>	my:employee/my:name	
	docid	Int32	<input checked="" type="checkbox"/>	DBXL::DocId	
Add Column					

47. Click **Save** in the DAT header, and then click **OK** in the confirmation dialog.

If you have existing documents in the DBXL Document Type, click **Reshred All Documents**. This will push the document's data into SQL, based on the current Database Mapping. You can use SQL Server Management Studio to verify that the form data has now been mapped to the SQL table.

You can also submit a new document, then verify that the form data is mapped to SQL:

```

SQLQuery4.sql - DB...65\dbxladmin (64) X
/***** Script for SelectTopNRows command from SSMS *****/
SELECT TOP 1000 [docid]
, [Submitter]
, [Manager]
, [Purpose]
, [ReportDate]
FROM [SampleDB].[dbo].[ExpenseReports]

```

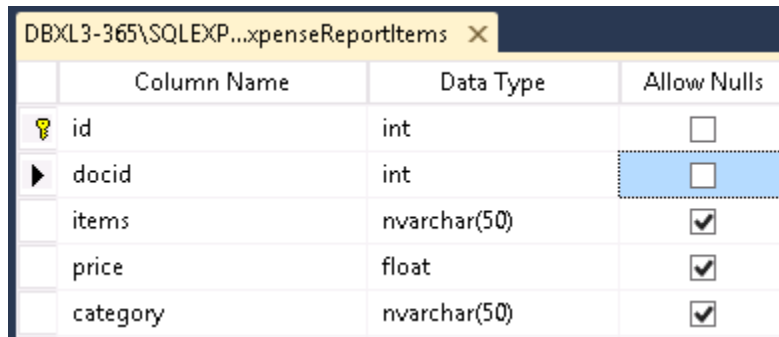
docid	Submitter	Manager	Purpose	ReportDate
1	28438	NULL	TEST SUBMIT	2017-05-01
2	28439	Don Stephen Lambatin	Business Trip	2017-05-01





EXPAND THE DATABASE MAPPING

Now we want to map the repeating data for **Items**, and to do that we need an additional SQL table.

48. In SQL Server Management Studio, create a new table and call it **ExpenseReportItems**.



	Column Name	Data Type	Allow Nulls
	id	int	<input type="checkbox"/>
	docid	int	<input type="checkbox"/>
	items	nvarchar(50)	<input checked="" type="checkbox"/>
	price	float	<input checked="" type="checkbox"/>
	category	nvarchar(50)	<input checked="" type="checkbox"/>

Why do we need the ExpenseReportId (**docid**) in the child table? Because this is what links our child data to its parent table.

49. Add an additional column called **id**. Set this column to be the Primary key, and also set it to auto-increment (Identity Specification = Yes).

50. Update the database mapping to include the new table and columns.

- Back in the DBXL Administration Tool, click **Edit** for the document type.
- Switch to the **Database** tab.
- Click on **Add Sub Table** and select the **ExpenseReportItems** table from the dropdown.
- Click the "Select Schema Node" icon under Node Path and select the my:items/my:item (which is the repeating group in the form's schema).
- Click **Insert Foreign Key**. From the **Column Name** dropdown, select the **docid** (this is from the Child table) and from the **Parent Column name** dropdown select also the **docid** (this is from the Parent table).
- Click **Add Column**, select **Category** from the dropdown, and select the **category** node for the **Node Path**. Repeat this step to add a mapping for each of the columns.

You might be left wondering why we did not map the **ExpenseReportItemID** column. It's because we set that be an auto-increment, primary key when we created the table. Therefore, there is no need to map a value from the InfoPath form.

51. Click **Save**, and then click **OK** in the confirmation dialog.



Database Mapping					
SQL Side			Template Sides		Del
Table Name(Columns)	Column Name	Type	Key	Node Path	
ExpenseReports	<input type="button" value="Add Sub Table"/>			/my:expenseReport	<input type="button" value="Add"/> <input type="button" value="Remove"/>
	Manager	String	<input type="checkbox"/>	my:manager/my:managerName	<input type="button" value="Add"/> <input type="button" value="Remove"/>
	Purpose	String	<input type="checkbox"/>	my:purpose	<input type="button" value="Add"/> <input type="button" value="Remove"/>
	ReportDate	String	<input type="checkbox"/>	my:reportDate	<input type="button" value="Add"/> <input type="button" value="Remove"/>
	Submitter	String	<input type="checkbox"/>	my:employee/my:name	<input type="button" value="Add"/> <input type="button" value="Remove"/>
	docid	Int32	<input checked="" type="checkbox"/>	DBXL::Docid	<input type="button" value="Add"/> <input type="button" value="Remove"/>
	<input type="button" value="Add Column"/>				
ExpenseReportItems	<input type="button" value="Add Sub Table"/>			/my:expenseReport/my:items/my:iten	<input type="button" value="Add"/> <input type="button" value="Remove"/>
	Foreign Key: docid	Column Name: docid	Parent Column Name: docid	Key Type: Column	<input type="button" value="Remove"/>
	<input type="button" value="Insert Foreign Key"/>				
	category	String	<input type="checkbox"/>	my:category	<input type="button" value="Add"/> <input type="button" value="Remove"/>
	items	String	<input type="checkbox"/>	my:description	<input type="button" value="Add"/> <input type="button" value="Remove"/>
	price	Double	<input type="checkbox"/>	my:amount	<input type="button" value="Add"/> <input type="button" value="Remove"/>
	<input type="button" value="Add Column"/>				

Now let's verify the mapping!

Submit a new document, making sure to fill out all the fields that have been mapped to SQL. Then use SQL Server Management Studio to verify that the form data was mapped correctly.

```

SQLQuery5.sql - DB...65\dbxladmin (63) X
/***** Script for SelectTopNRows command from SSMS *****/
SELECT TOP 1000 [id]
  , [docid]
  , [items]
  , [price]
  , [category]
FROM [SampleDB].[dbo].[ExpenseReportItems]

```

	id	docid	items	price	category
1	1	28438	NULL	0	NULL
2	2	28439	Plane ticket	1000	Transportation
3	3	28439	Starbucks	20	Meals



ACTIVATE ERROR NOTIFICATIONS

When errors occur in DBXL, information is recorded in the Event Viewer of the server where DBXL is installed. The steps in this task will allow the administrator to receive an email notification any time that an error is recorded in the Qdabra logs.

52. Create the SendErrorMessage.vbs file.
 - a. Open Notepad and paste the text below.

```

Set oMessage = CreateObject("CDO.Message")
oMessage.Subject = "Errors have been logged in Qdabra eventlog"
oMessage.From = "someone@yourcompany.com"
oMessage.To = "admin@yourcompany.com"
oMessage.TextBody = "Errors have been logged in the Qdabra eventlog"

' Send using a specific smtp host unless you have a predefined configuration
With oMessage.Configuration.Fields

    ' Send using Port = 2
    .Item("http://schemas.microsoft.com/cdo/configuration/sendusing") = 2

    ' Name or IP of Remote SMTP Server
    .Item("http://schemas.microsoft.com/cdo/configuration/smtpserver") =
"smtphost.yourcompany.com"

    ' Server port (typically 25)
    .Item("http://schemas.microsoft.com/cdo/configuration/smtpserverport") = 25

    .Update

End With

oMessage.Send

' Create an eventlog trigger that will execute this vbscript
'
' eventtriggers /Create /TR "Qdabra Error Notification" /TK C:\Notifications\SendErrorMessage.vbs /D "Send
an email for any DBXL errors" /L Qdabra /T ERROR /RU ""

```

- b. Make any necessary modifications. You should update the **SMTP server**, the **From email** and the **To email**.
 - c. Save the file, naming it SendErrorMessage.vbs (without a *.txt extension). Make sure to save the file SendErrorMessage.vbs to **C:\Notifications** on the server where DBXL is installed.



53. Configure the notifications

- d. Open **Computer Management** and navigate to the **Event Viewer**.
- e. Navigate to and select the **Qdabra** event log.
- f. Right-click on the log and select **Attach Task to This Log**.
- g. In the **Create Basic Task** wizard, type a name and description, then click **Next**.
- h. In the **When a Specific Event is Logged** screen click **Next**.
- i. In the **Action** window select **Start a program** and click **Next**.
- j. Click **Browse**, select **C:\Notifications\SendErrorEmail.vbs** and click **Next**.
- k. Check **Open the Properties dialog for this task when I click Finish**, and click **Finish**.
- l. Under **Security options**, click **Change User or Group**.
- m. Type in **SYSTEM**, click **Check Names** and click **OK**.
- n. In the **Configure for** dropdown, select the correct option.
- o. In the **Settings** tab, select
- p. Click **OK**.
- q. In the **Triggers** tab, add a custom trigger such that emails are only sent when a specific Event Level is recorded. In the screenshot below we have selected **Critical**, **Error** and **Warning**. These are the recommended settings.

Now, every time there is an error recorded in the event viewer, the email address specified in the vbs file will receive an email.

This serves as a notification telling the recipient to check the event viewer logs and address the error(s).

In some cases, these event notification emails are treated as junk mails, and therefore appear in the Junk/Spam folder. To remedy this, programs like Microsoft Outlook provides you options to 'whitelist' these messages.



WHAT'S NEXT?

- **SQL mapping:** If you need more information on the Database mapping between DBXL and SQL, a tutorial is available by [clicking here](#).
- **Training:** Qdabra offers DBXL training! Please [contact us](#) to find out about the next available online training session. The training materials from previous sessions are available [here](#). Through these modules you can learn about the Active Directory web service, the QueryDB web service and much more!
- **Community:** The InfoPathDev.com community has [Product support forums](#) where Qdabra's products can be discussed. This is a good place to contact other users of Qdabra's products. Qdabra's support staff also monitors this forum to identify issues and respond.
- **Support:** If you have questions about the information in this document, please contact us via Support@Qdabra.com for assistance. To learn more about Qdabra support, please visit [our website](#).



APPENDIX: TROUBLESHOOTING THE EVENT RECEIVER

When troubleshooting issues related to the Event Receiver, refer to the footer of the DBXL Settings page to learn which version of the WSP is installed. The footer should look like this: **Qdabra DBXL for SharePoint version 2.8.2013.909 (last updated 9/9/2013 2:34:17 PM)**. This information will be valuable when contacting Qdabra Support.

POWERSHELL EXECUTION

When executing PowerShell commands, you might encounter the following errors.

1. **“Admin SVC must be running in order to create deployment timer job.”**

To solve this, go to the Services section in Control Panel / System and Security / Administrative Tools and find the service called SharePoint Administration. Make sure this service has been started.

2. **“A deployment of retraction is already under way for the solution “Qdabra.DBXL.SharePoint.wsp”, and only one deployment or retraction at a time is supported”.**

To solve this, from the command line, execute the following:

```
C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\14\BIN>stsadm -o
execadmsvcjobs
```

If you see the following error: *“The administration service is running so all administration jobs will be run in the timer service”*, go to the Services section in Control Panel / System and Security / Administrative Tools and find the service called SharePoint Administration. Stop the service and then re-execute

```
C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\14\BIN>stsadm -o
execadmsvcjobs
```

THE OPERATION HAS TIMED OUT

We have seen the following error (sent via email by the Event Receiver) when the hard drive is low on disk space:

```
Web url: http://q-dev64-01
Library: ExpenseReport20130819
User: Ernesto Machado (Test) (i:0#w|autonomysystems\ernestom)
Item ID: 8
Item Url: ExpenseReport20130819/8.xml
Exception:
The operation has timed out
```



<http://www.qdabra.com>

Last updated on 5/8/2017 1:01 PM

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

To resolve, ensure the server where SharePoint is installed does not have disk space issues.

IS THE WSP SOLUTION DEPLOYED?

If you need to check whether the WSP has been already deployed, execute the following command in the PowerShell:

```
get-spsolution -Identity Qdabra.Dbxl.SharePoint.wsp
```

To get more details, execute:

```
get-spsolution -Identity Qdabra.Dbxl.SharePoint.wsp | SELECT *
```

OBJECT REFERENCE NOT SET TO AN INSTANCE OF AN OBJECT

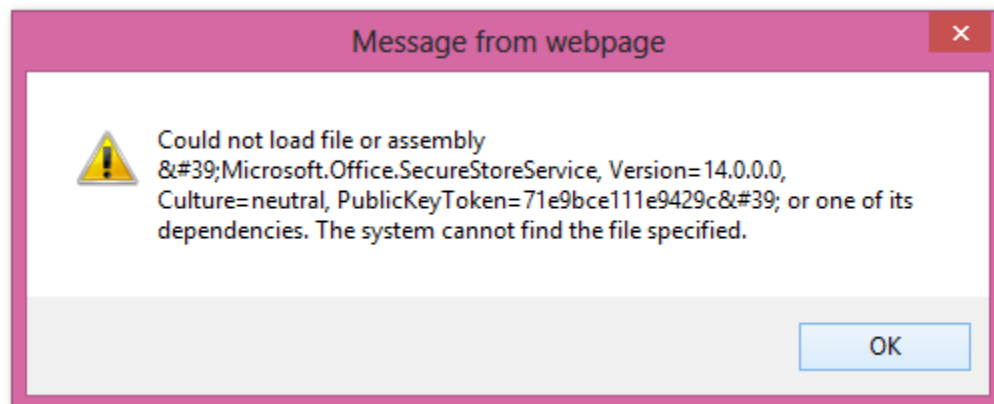
When configuring the Event Receiver, you may see a message that says: “Object reference not set to an instance of an object.” This error typically refers to one of the following errors in the DBXL Url:

- Incorrect server name, e.g. <http://<invalid>/qdabrawebservice/>
- Incorrect service name, e.g. <http://<correctservername>/<invalidservicename>/>

In SharePoint 2010, the above might also yield the message: “InvalidOperationException” and in SharePoint 2013 “NotSupportedException”.

SECURE STORE SERVICE REQUIRES SHAREPOINT STANDARD OR ENTERPRISE

In the DBXL Settings page, you might receive the following message:



Could not load file or assembly 'Microsoft.Office.SecureStoreService, Version=14.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c' or one of its dependencies. The system cannot find the file specified.



The SecureStoreService requires SharePoint Standard or Enterprise. This error might mean you are using SharePoint Foundation.

DELETING DATA

If you delete an xml document from the Documents tab in the DBXL Admin Tool, that document will NOT be deleted from the SharePoint Form Library.

However, if you delete the xml document from the SharePoint form library, the data will be deleted from DBXL and consequently from the SQL database (if a database mapping has been defined).

DBXL TOKENS

When no xpaths are defined for the DBXL tokens in the Settings page, the following will be used by default:

- **Name** is blank
- **Author** is domain\user
- **Description** is 6ad628ec-19a2-4e27-a0bf-dc15d4d3b325:1#;ExpenseReport_EventReceiver_20130821/1.xml

The Description is composed of the {SharePoint Unique ID} then the SharePoint ID and the site relative URL.

CONVERT DATA CONNECTIONS TO UDCX

Because in this scenario DBXL is on a server (or site) different than the SharePoint site, we must convert all QdabraWebService data connections to UDC and update them to use the Secure Store App ID. Here are steps to guide you:

1. In your SharePoint site, create a Data Connections library (if one does not already exist).
2. In InfoPath designer, go to **Data > Data Connections**.
3. Select the data connection and click "Convert to connection file".
4. Enter the URL for the new UDC file, such as *http://<servername>/Data%20Connections/ADWSGetMyInfo.udcx*, and click **OK**.
5. Using Internet Explorer, navigate to the **Data Connections** library.
6. Download a copy of the UDC file.
7. Open the UDC file locally in Notepad or a similar text editor.
8. Within the UDC file, change this line:

```
<!--udc:Authentication><udc:SSO AppId=" CredentialType=" /></udc:Authentication-->
```

to



<http://www.qdabra.com>

Last updated on 5/8/2017 1:01 PM

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

```
<udc:Authentication><udc:SSO AppId='<YOUR_APP_ID>' CredentialType='NTLM'  
></udc:Authentication>
```

9. Save and close the UDC file, then upload it to the Data Connections library, making sure to overwrite the existing file.

Remember to execute the steps above for all QdabraWebService data connections. The SSO App ID, as in the tutorial above, needs access to DBXL for these data connections to execute successfully.



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>



<http://www.qdabra.com>

Last updated on 5/8/2017 1:01 PM

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