



DBXL's Integrated Scenario

Qdabra has recommended the *Integrated* scenario to customers using DBXL v2.7 or earlier versions. We call it "Integrated" because DBXL is installed on the **SharePoint – 80** site (or the same site where the forms will be located) and the InfoPath form submits directly to the DBXL web service. Therefore, this scenario assumes DBXL was installed on the SharePoint site, typically SharePoint-80. This scenario is more common for DBXL users in MOSS 2007 and SharePoint 2010.

While there are additional scenarios that can be achieved with DBXL, this tutorial will focus on the Integrated 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 v2.8
- SharePoint 2010. This scenario is most important for SharePoint 2010. Therefore, the tutorial in this document will focus on SharePoint 2010. However, the steps should be similar if working on SharePoint 2007.
- 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	2
Add the site to Local Intranet	3
Submit to DBXL	3
Set up the DBXL Document Type	8
Add a SharePoint mapping	10
Test the solution!	11
Create a SQL table	12
Add Database mapping	12
Expand the Database mapping	13
Activate error notifications	15
What's next?	17

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="" />
```



<http://www.qdabra.com>

Last updated on 2/12/2014 2:19 PM

Copyright © 2013 Autonomy Systems, LLC. All rights reserved.

```
<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.

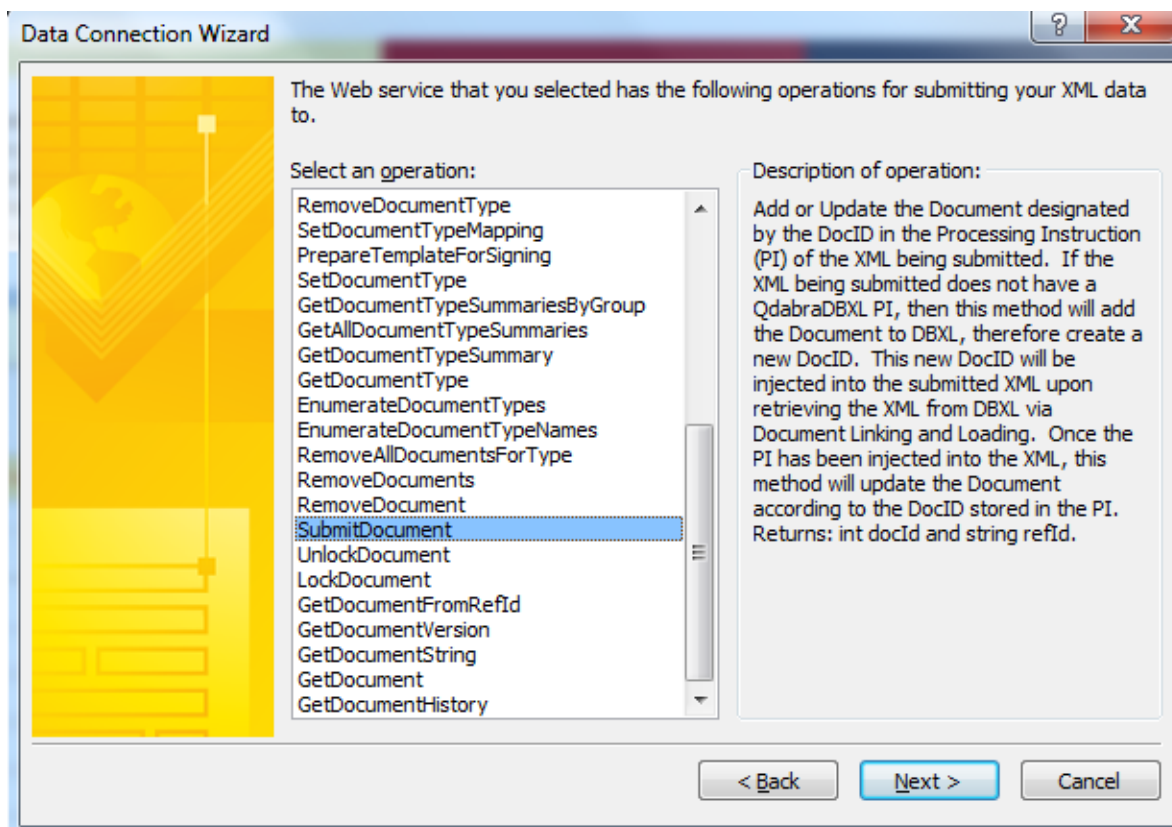
CONFIGURE THE FORM TO SUBMIT TO DBXL

In the Integrated scenario, the InfoPath form template needs to be configured to submit to the DBXL web service.

8. Open your InfoPath form template (XSN) in design mode.
9. Switch to the **Data** tab and click **Data Connections**.



10. You'll see a data connection called **Main Submit**. Since we want the form to submit to DBXL (instead of using this data connection), let's **Remove** this data connection.
11. Now click on **Add**. This will load the Data Connection Wizard.
12. Select **Submit data** and click **Next**, then select "**To a web service**" and click **Next**.
13. In the web service field, enter
http://<yourservename>/qdabrawebservice/DbxlDocumentService.asmx and click **Next**.
14. Select **SubmitDocument** and click **Next**.



Next, you'll configure the parameters used to submit to DBXL.

15. Ignore the first parameter, called **tns:docTypeName**. In this scenario, DBXL will be able to automatically detect this. There are other scenarios where this value will need to be defined.
16. Select the second parameter, named **tns:xml**. Under **Parameter options** select the **Entire form (XML document, including processing instructions)** radio button. Check the **Submit data as a string** checkbox.



The submit operation for the Web service requires the following parameters. Specify which fields or groups in your form provide the data for these parameters. If the Web service parameter requires an entire XML document, you can specify that as well.

Parameters:

Parameter	Type	Element
tns:docTypeName	string	
tns:xml	string	/
tns:name	string	

Parameter options

Submit the following data for the selected parameter:

☐ Field or group:

Include:

☒ Entire form (XML document, including processing instructions)

☒ Submit data as a string

Note: Digitally signed data must be submitted as a string to preserve white spaces.

< Back Next > Cancel

That's the only required parameter. The next three are optional parameters.

17. Select the **tns:name** parameter.

This parameter, along with the **Author** and **Description**, are called "DBXL tokens". These are displayed in the document list in the **Documents** tab in DAT.

18. Select the radio button **Field or Group** in the **Parameter options** section.

19. Click the Modify button next to the **Field or Group** text box. The **Select a Field or Group** dialog will appear.

20. Select the node **/my:expenseReport/my:expenseCode**, click **OK** and check the **Submit data as a string** checkbox.

21. Select the **tns:author** parameter.

22. Select the radio button **Field or Group** in the **Parameter options** section.

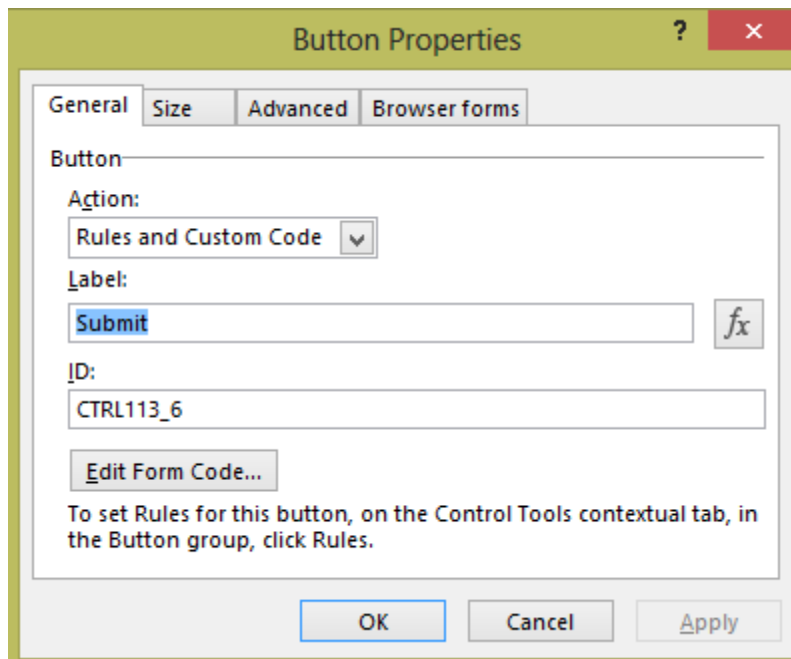


23. Click the **Modify** button next to the **Field or Group** text box. The **Select a Field or Group** dialog will appear.
24. Select the `my:name` field under `my:employee`, click **OK** and check the **Submit data as a string** checkbox.
25. Select the `tns:description` parameter.
26. Select the radio button **Field or Group** in the **Parameter options** section.
27. Click the **Modify** button next to the **Field or Group** text box. The **Select a Field or Group** dialog will appear.

Parameters:		
Parameter	Type	Element
tns:xml	string	/
tns:name	string	/my:expenseReport/my:expenseCode
tns:author	string	/my:expenseReport/my:employee/my:name
tns:description	string	/my:expenseReport/my:purpose

28. Click **Next**. Click **Finish**. Click **Close** to exit the **Data connections** dialog.⁴
29. Find the **Submit** button at the bottom of the form. Right click on it and select **Button Properties**.
30. In the **General** tab, change the **Action** dropdown to “Rules and Custom Code” and change the label to “Submit”. In the **Browser forms** tab, change the **Postback settings** to **Always**. Click **OK**.





Button Properties ? x

General Size Advanced Browser forms

Button

Action:
Rules and Custom Code ▼

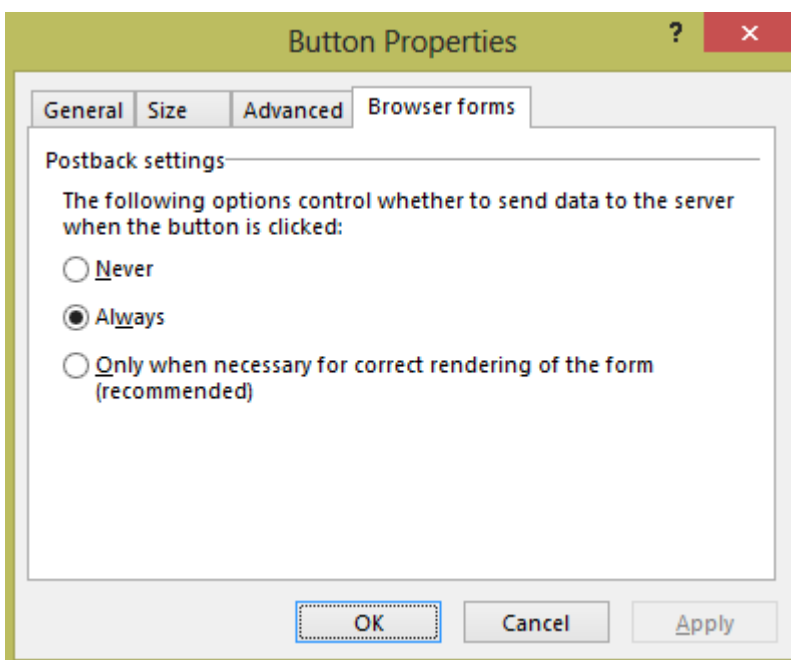
Label:
Submit fx

ID:
CTRL113_6

Edit Form Code...

To set Rules for this button, on the Control Tools contextual tab, in the Button group, click Rules.

OK Cancel Apply



Button Properties ? x

General Size Advanced Browser forms

Postback settings

The following options control whether to send data to the server when the button is clicked:

☐ Never

☒ Always

☐ Only when necessary for correct rendering of the form (recommended)

OK Cancel Apply

31. Select the button, then click on **Home > Manage Rules**.
32. In the rules taskpane, click **New > Action > Add > Submit data**.
33. There's only one data connection to select, so click OK.



The screenshot shows a 'Rule Details' dialog box. It has a title bar with a question mark and a close button. Inside, there are two main sections: 'Action:' and 'Data connection:'. The 'Action:' section has a dropdown menu currently showing 'Submit data'. The 'Data connection:' section has a dropdown menu showing 'Web Service Submit' and an 'Add...' button to its right. At the bottom right of the dialog are 'OK' and 'Cancel' buttons.

34. Select **Add > Close form** and click **OK**.

35. Save the updated form by selecting **File > Save**. When saving, save the form as an InfoPath 2010 Web Browser Form. Then close the InfoPath designer.

Your form is now configured to submit to the DBXL document service.

SET UP THE DBXL DOCUMENT TYPE

36. Using Internet Explorer, go to <http://<servername>/QdabraWebService/>.

37. Click on the first link, **DBXL Administration Tool**, to launch DAT.



<http://www.qdabra.com>

Last updated on 2/12/2014 2:19 PM

Copyright © 2013 Autonomy Systems, LLC. All rights reserved.



38. Once DAT opens in InfoPath, click **New Configuration**.

39. 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.

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



<http://www.qdabra.com>

Last updated on 2/12/2014 2:19 PM

Copyright © 2013 Autonomy Systems, LLC. All rights reserved.

ADD A SHAREPOINT MAPPING

Next we'll add a SharePoint mapping so that users can interact with this form via a SharePoint form library.

41. Switch to the **SharePoint** tab.
42. Click **Insert Mapping**.
43. In the **URL** field, enter the URL to a form library that does not exist.

Creating a SharePoint mapping requires that DBXL be installed on the same server as SharePoint. Therefore, your SharePoint form library will exist on the same server where DBXL is installed.

44. For **Method**, select **Object Model**.
45. Check the checkboxes for **Publish Library and XSN** and **Use Forms Services**.
46. Click **Save** and then click **OK** in the confirmation dialog.

The screenshot shows the QDABRA Administration Tool interface. The title bar reads "DATABASE ACCELERATOR Administration Tool" and "QDABRA SOFTWARE". The document type is "ExpenseReport". The interface has tabs for General, Database, Permissions, SharePoint, Documents, and Taxonomy Tree. The SharePoint tab is active, showing a section titled "SharePoint List or Library (i.e. Collection)". Below this is a text box explaining the integration process. A table with two columns, "URL" and "Method", contains one row with the URL "http://training-2010-b/DBXL_Training_Module_1" and the Method "Object Model". Below the table is an "Expression" field with a dropdown menu showing "Insert Expression". At the bottom, there are checkboxes for "Publish Library and XSN" (checked), "Use Forms Services" (checked), "Propagate Deletes" (unchecked), "Asynch" (unchecked), and "Content Type" (unchecked).



TEST THE SOLUTION!

47. Use Internet Explorer to navigate to the form library URL.
48. Click **Add Document**.
49. Fill out the form and submit.

The form library in Internet Explorer will refresh and you'll see the new document.

Type	Name	Modified	Modified By	Checked Out To	DBXL ID	DBXL Version	Author	Description	Title
	2012-03-18	3/18/2012 7:20 PM	AUTONOMYSYSTEMS\ErnestoM		6467	1	Michael Jones	This is a test	2012-03-18
Add document									

You can also check the Documents tab in DAT to verify that your new document is not stored in DBXL as well.

DATABASE ACCELERATOR
Administration Tool

Document Type: **ExpenseReport**
 Save Save As... Catalog

General | **Database** | **Permissions** | **SharePoint** | **Documents** | **Taxonomy Tree**

Export Documents

Import Documents

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: Ref. ID: Range:
 Previous Next Query Query All

If reshredding more than 100 documents, please use the DBXL Migration Tool Refresh

Documents Delete Delete All Reshred Reshred All

Doc. ID	Ref. ID	Version #	Name	Author	Description	New	
227	1	1	12345	Ernesto	this is a test	Open	☐

Powered by: **Qdabra® Database Accelerator**
Web Service URL: <http://TRAINING-2010-B/QdabraWebService/DbxlDocumentService.aspx>
DBXL Version: 2.8.2013.1202
Form Version: 2.8.0.6
User: ErnestoM

Copyright © 2008 Qdabra Software. All rights reserved.
Qdabra and the Qdabra logo are trademarks of Autonomy Systems LLC. www.qdabra.com

English ▼



<http://www.qdabra.com>

Last updated on 2/12/2014 2:19 PM

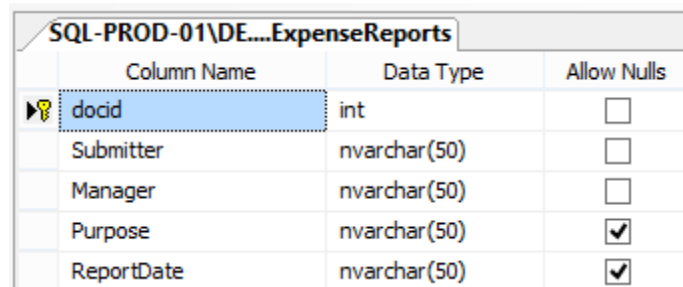
Copyright © 2013 Autonomy Systems, LLC. All rights reserved.

CREATE A SQL TABLE

The following steps require SQL Server Management Studio.

50. 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	☐
Submitter	nvarchar(50)	☐
Manager	nvarchar(50)	☐
Purpose	nvarchar(50)	☒
ReportDate	nvarchar(50)	☒

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.

51. 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.

52. 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.



53. 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.

Table Name (5 Columns) ▼	Node Path
ExpenseReports	/my:expenseReport

Column Name	Add Column	DB Type	Key	Node Path
Manager	▼	String	☐	my:manager/my:managerName
Purpose	▼	String	☐	my:purpose
ReportDate	▼	String	☐	my:reportDate
Submitter	▼	String	☐	my:employee/my:name
docid	▼	Int32	☒	DBXL::DocId

Qdabra® DBXL Admin

Schema Tree

- DBXL Mapping Tokens
 - DBXL::Link
 - DBXL::DocId
 - DBXL::DocType
 - DBXL::RefId
 - DBXL::VerNo
 - DBXL::CreatedBy
 - DBXL::DateCreated
 - DBXL::ModifiedBy
 - DBXL::DateModified
- my:expenseReport

54. 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.

EXPAND THE DATABASE MAPPING

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

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

SQL-PROD-01\DE...enseReportItems			
	Column Name	Data Type	Allow Nulls
🔑	id	int	☐
	docid	int	☐
	items	nvarchar(50)	☒
	price	float	☒
	category	nvarchar(50)	☒
			☐



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

56. Add an additional column called **id**. Set this column to be the Primary key, and also set it to auto-increment (Identity Specification = Yes).
57. Update the database mapping to include the new table and columns.
 - a. Back in the DBXL Administration Tool, click **Edit** for the document type.
 - b. Switch to the **Database** tab.
 - c. Click on **Add Sub Table** and select the **ExpenseReportItems** table from the dropdown.
 - d. 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).
 - e. 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).
 - f. 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.

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

Database Map Reshred All Documents Refresh Database

Table Name (5 Columns) ▼	Node Path			
ExpenseReports	/my:expenseReport			
Column Name	Add Column	DB Type	Key	Node Path
docid	▼	Int32 ▼	☒	DBXL::DocId
Manager	▼	String ▼	☐	my:manager/my:managerName
Purpose	▼	String ▼	☐	my:purpose
ReportDate	▼	String ▼	☐	my:reportDate
Submitter	▼	String ▼	☐	my:employee/my:name
Table Name (3 Columns) ▼	Node Path			
ExpenseReportItems	my:items/my:item			
Column Name	Key Type	Parent Column Name		
docid	Column	docid		
☒ Insert Foreign Key				
Column Name	Add Column	DB Type	Key	Node Path
category	▼	String ▼	☐	my:category
items	▼	String ▼	☐	my:description
price	▼	Double ▼	☐	my:amount



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.

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.

59. Create the SendErrorEmail.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\SendErrorEmail.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 `SendErrorEmail.vbs` (without a *.txt extension). Make sure to save the file `SendErrorEmail.vbs` to **C:\Notifications** on the server where DBXL is installed.

60. 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](#).

