Side-by-side scenario using DBXL and qRules

This document describes a side-by-side scenario, in which the InfoPath form submits separately to DBXL and to the SharePoint Form Library. This tutorial does not use a SharePoint mapping and it does not use the Event Receiver. 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
- qRules v5.2
- SharePoint (2010 or 2013)
- 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

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.

SERVER CHANGES

4. This scenario will require a domain account that can be used to connect SharePoint and DBXL. Please create such an account; for this tutorial, we're using an account called <domain>_dbxluser.

5. Search for the key for AnonymousUser. Set the value equal to the user you've created.

```
<add key="AnonymousUser" value="AUTONOMYSYSTEMS\_dbxluser" />
```

6. Near the end of web.config, find the following text:

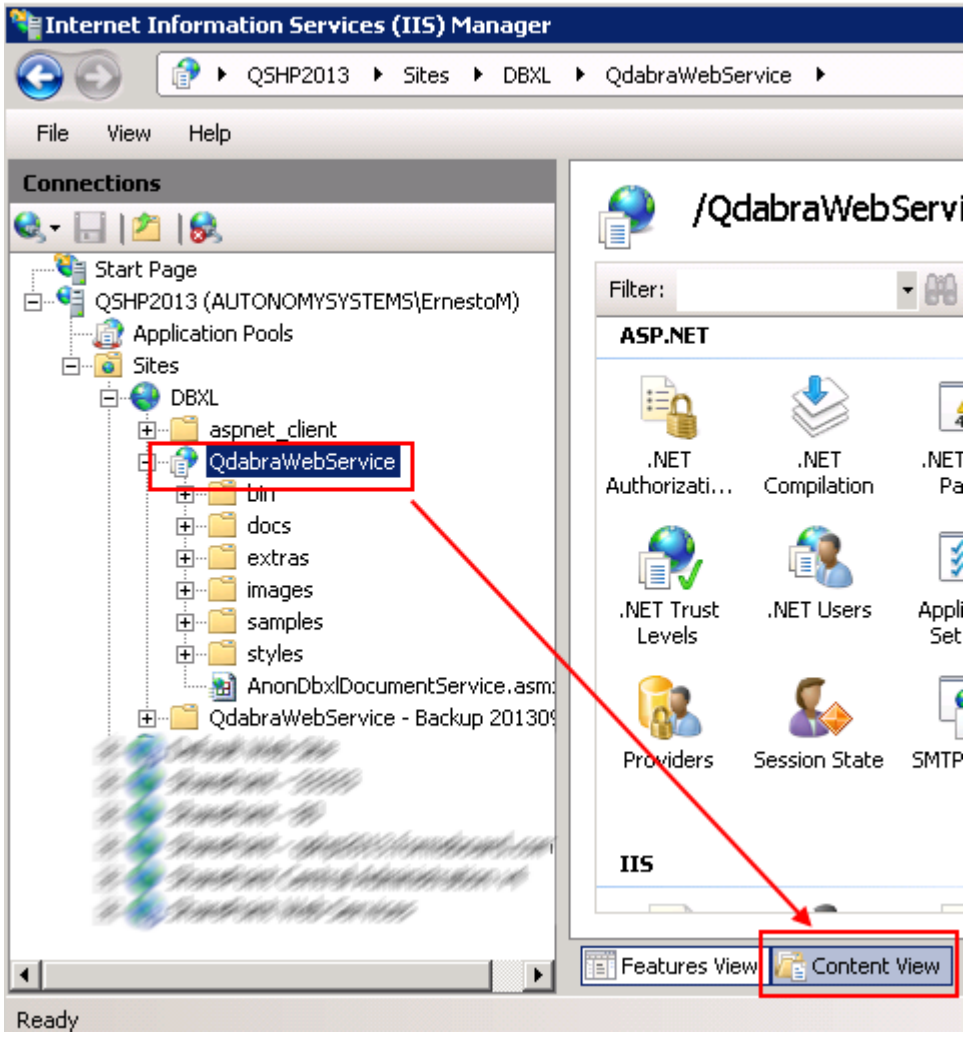
```
<!-- To allow anonymous access, uncomment out the location entries below and set the AnonymousUser account in the <appSettings> section -->
```

7. Uncomment the section below this text.

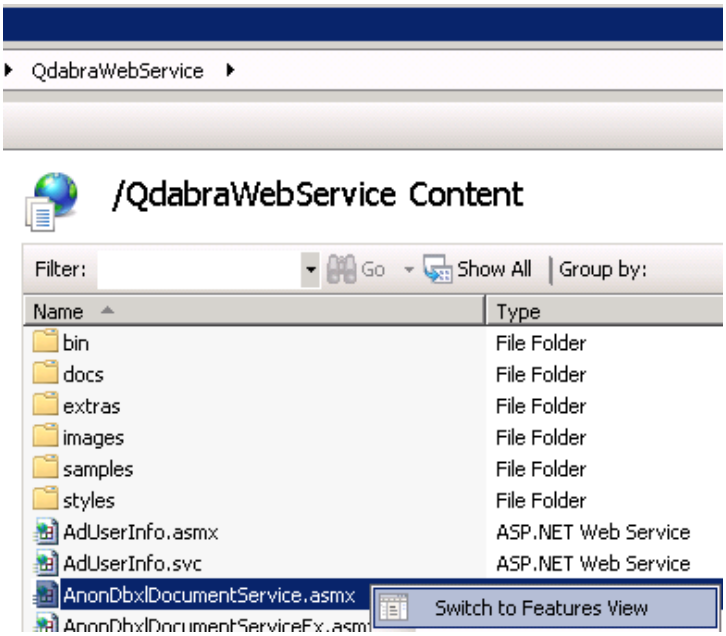
8. Save and close the web.config file.

9. Go to IIS Manager (still on the server where DBXL is installed).

10. Select the QdabraWebService site and then click **Content View** in the middle panel.



11. Right click on the AnonDbxlDocumentService.asmx and select **Switch to Features View**.



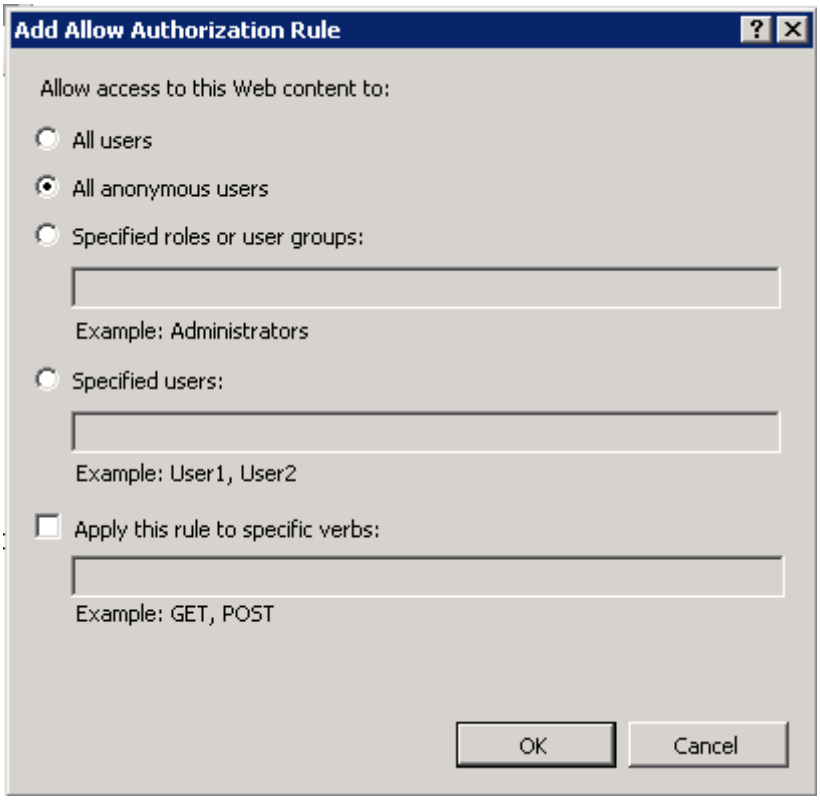
12. The AnonDbxlDocumentService.asmx is now selected on the panel on the left. In the middle panel, double click on **Authentication**.

13. Enable **Anonymous Authentication** and disable **Windows Authentication**.

14. Select AnonDbxlDocumentService.asmx on the panel on the left. In the middle panel, double click on **Authorization Rules**.

15. Remove any rules that exist already.

16. Add a rule that allows anonymous users.



These same steps would need to be repeated for any web service used in your form (for example, AnonQueryDB.asmx).

ADD THE SITE TO LOCAL INTRANET

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

17. In Internet Explorer, go to **Internet Options**, switch to the **Security** tab, select **Local intranet** and then click **Sites**.

18. In the **Local intranet** window, click **Advanced**.

19. 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.

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

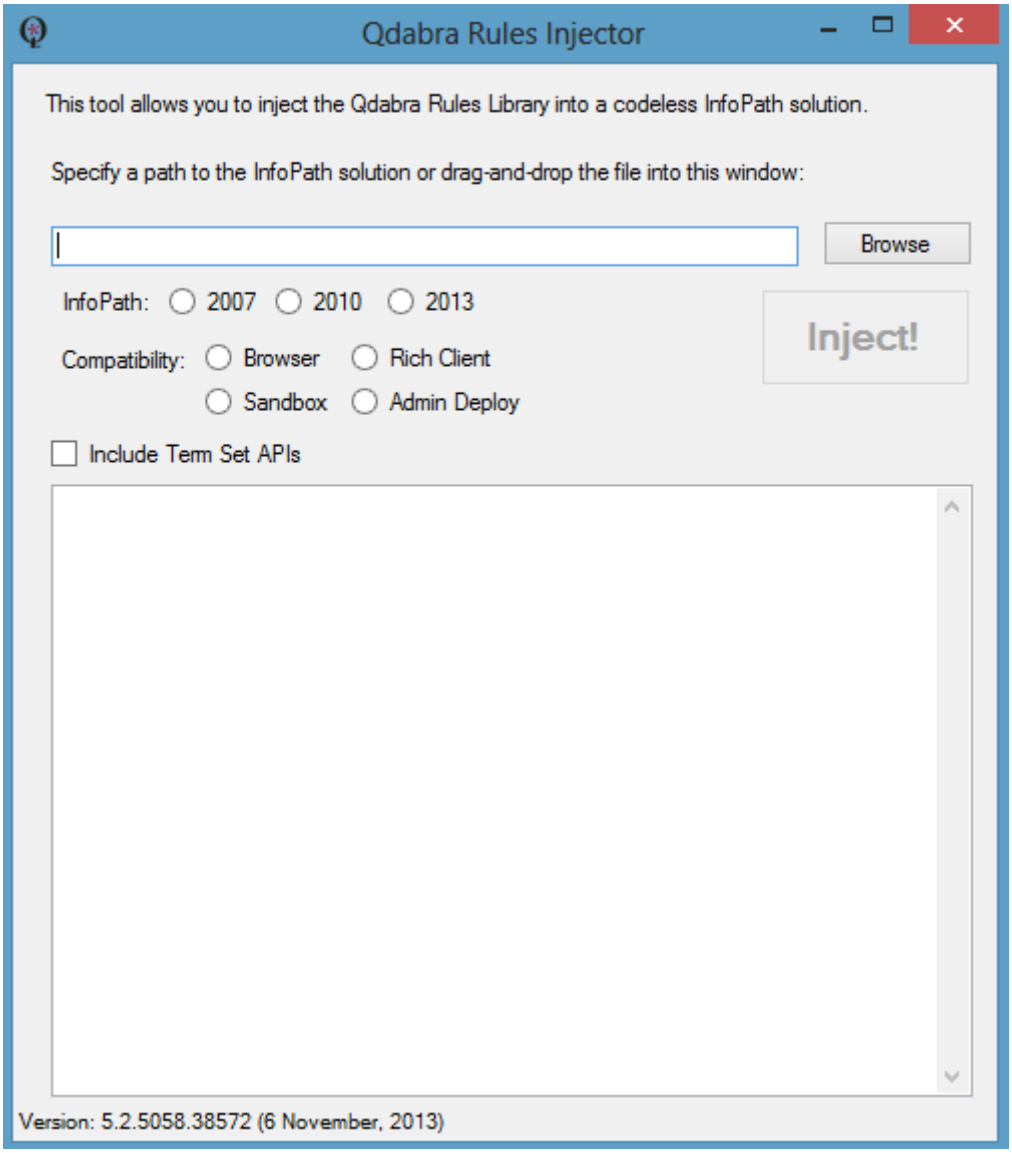
INJECT THE FORM WITH QRULES

The following steps assume qRules has already been downloaded and installed. A free trial of qRules can be obtained from Qdabra.com

21. Create a backup of your XSN, if desired. This is necessary because injection overwrites your XSN.

22. Close the form you wish to inject. This is necessary because injection will fail on an open form.

23. Launch the Qdabra Rules Injector tool by clicking on **Start** > **Programs** > **Qdabra** > **Tools** > **qRules** > **qRules Injector**.



24. Specify the XSN file you wish to inject. The injector will try to automatically detect the InfoPath version. However, you can change this in the injector if necessary.

25. Choose the type of form you are injecting qRules into and click the **Inject** button.

26. Close the injector.

SETUP THE SUBMITDOCUMENT DATA CONNECTION

27. Open your InfoPath form template (XSN) in design mode.

28. Switch to the **Data** tab and click **Data Connections**.

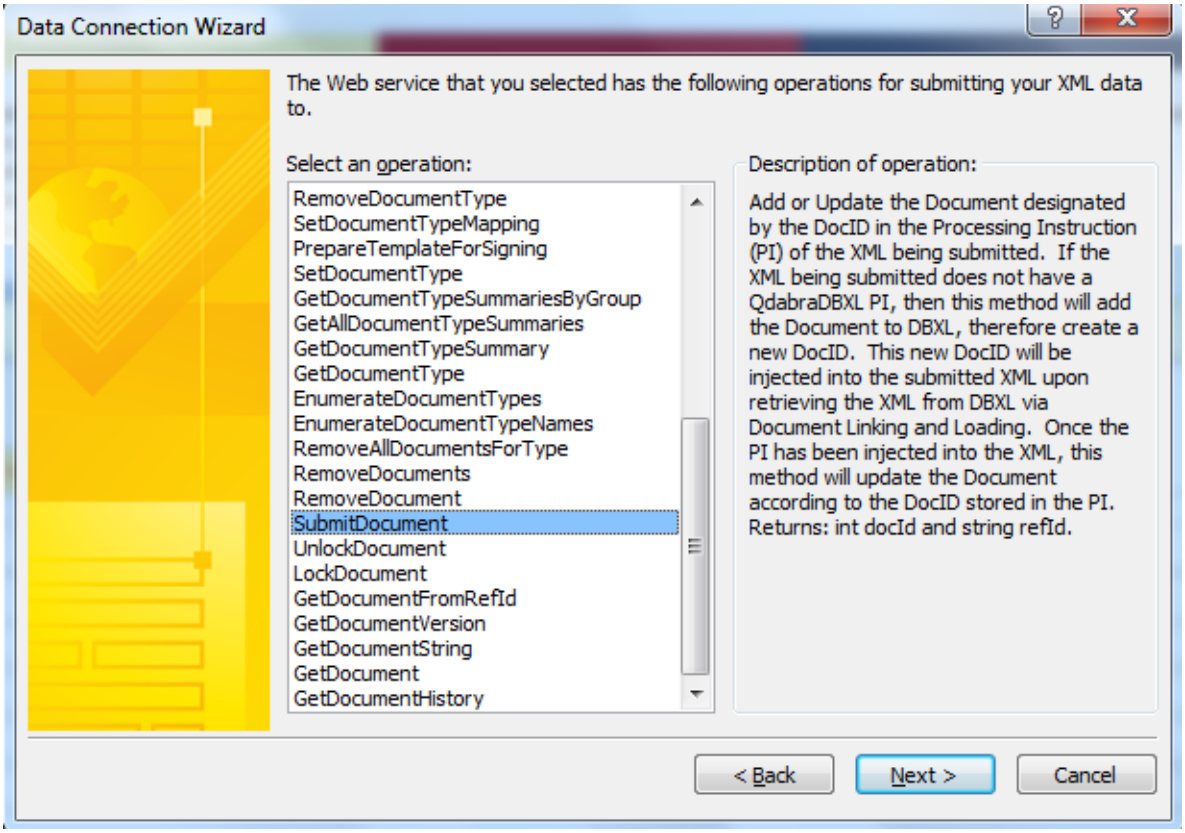
29. You'll see a data connection called ***Main Submit***. Since we want the form to submit to DBXL (instead of using this data connection), let's **Remove** this data connection.

30. Now click on **Add**. This will load the Data Connection Wizard.

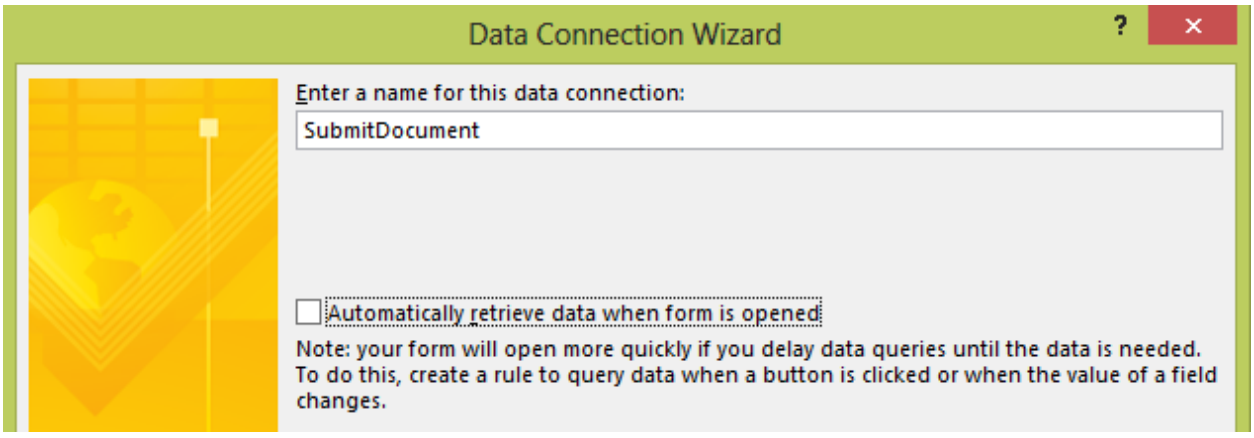
31. Select **Receive data** and click **Next**, then select "**To a web service**" and click **Next**.

32. In the web service field, enter http://<yourservername>/qdabrawebservice/AnonDbxlDocumentService.asmx and click **Next**.

33. Select **SubmitDocument** and click **Next**.



You do not need to set any values for this data connection. You may name it whatever you like – but you will need to know the name for use in the qRules commands. This document will call this data connection "SubmitDocument". Also, make sure to uncheck the "Automatically retrieve data when form is opened" checkbox on the last page of the wizard.



CREATE THE SHAREPOINT FORM LIBRARY AND THE SUBMIT DATA CONNECTION

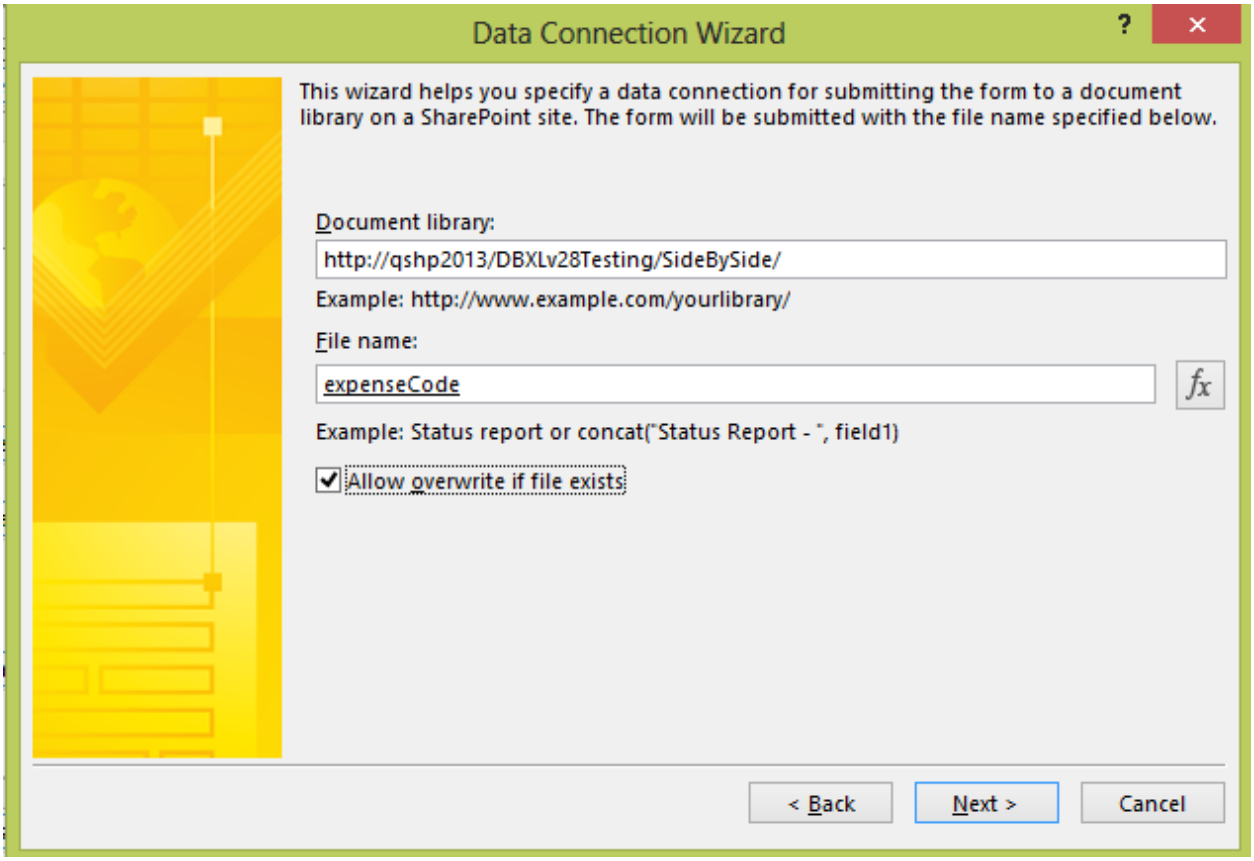
34. Create a new form library. This can be accomplished by going to **Site Contents** > **add an app** > **Form Library**. Then provide a name for the form library and click **Create**.

35. Back in the InfoPath form, go to **Data** > **Data Connections**.

36. Click **Add** > **Submit Data** then click **Next**.

37. Select "To a document library on a SharePoint site" and click **Next**.

38. 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".

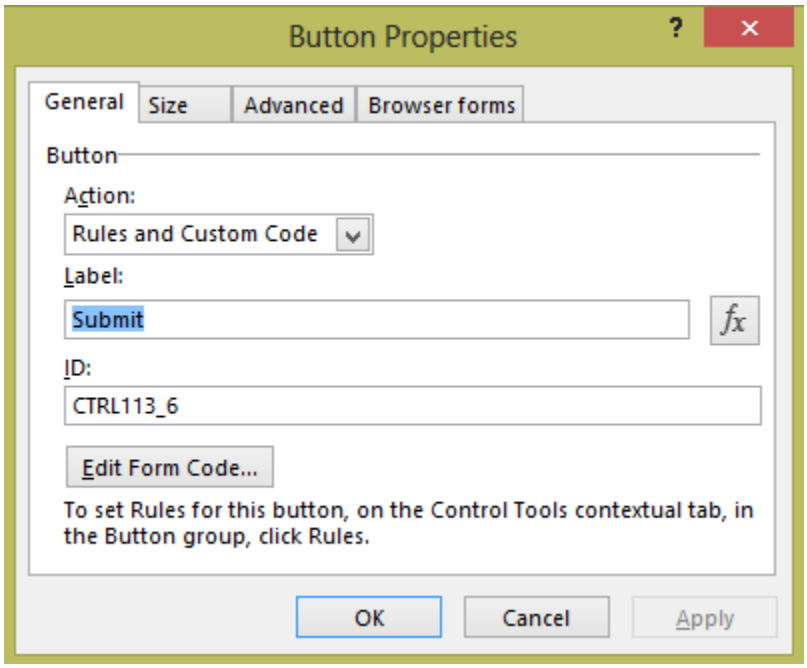


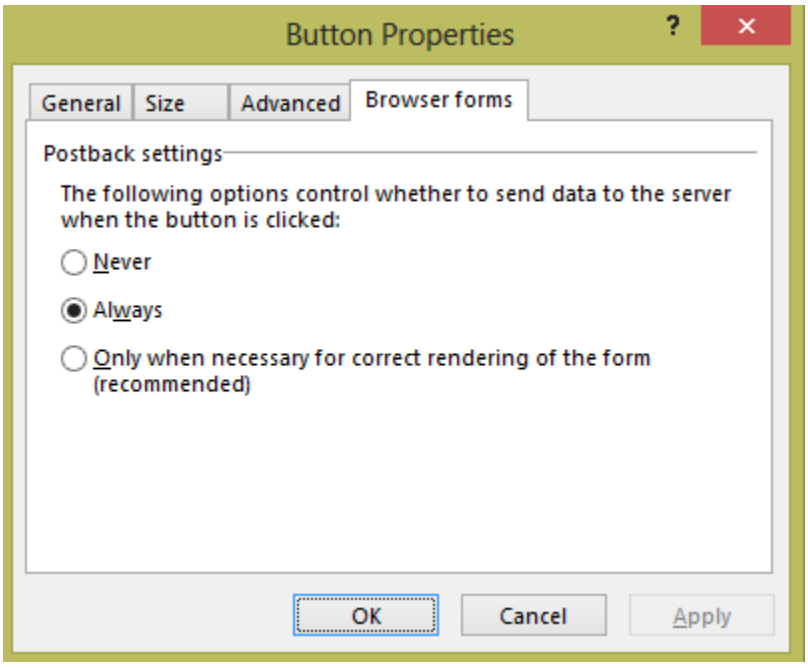
39. Click **Next**, then click **Finish**, then click **Close**.

SET UP THE SUBMIT BUTTON ON THE FORM

40. Find the **Submit** button at the bottom of the form. Right click on it and select **Button Properties**.

41. 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**.





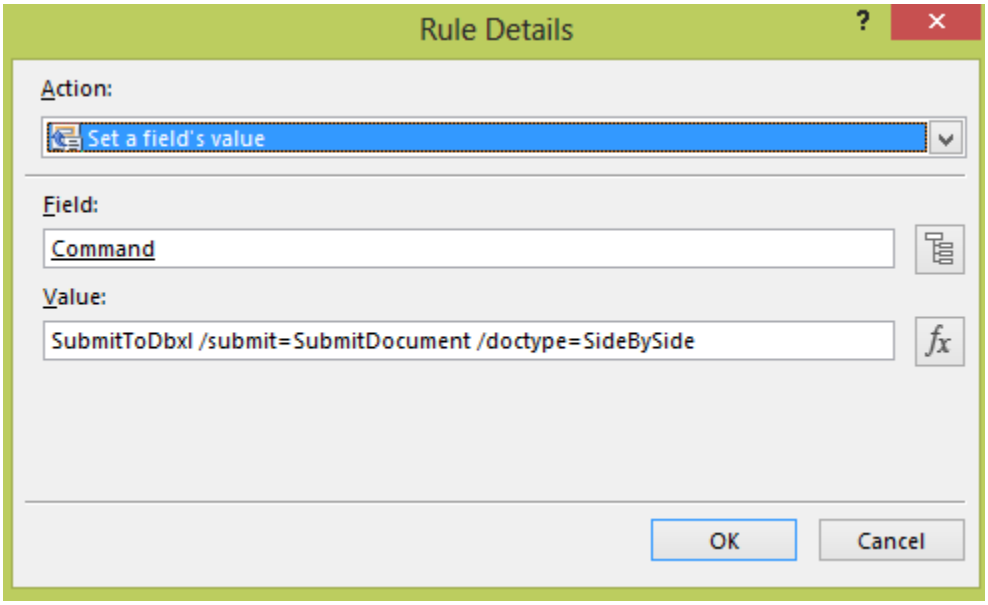
42. Select the button, then click on **Home** > **Manage Rules**.

43. Select **New** > **Action** > **Add** > **Set a field's value**.

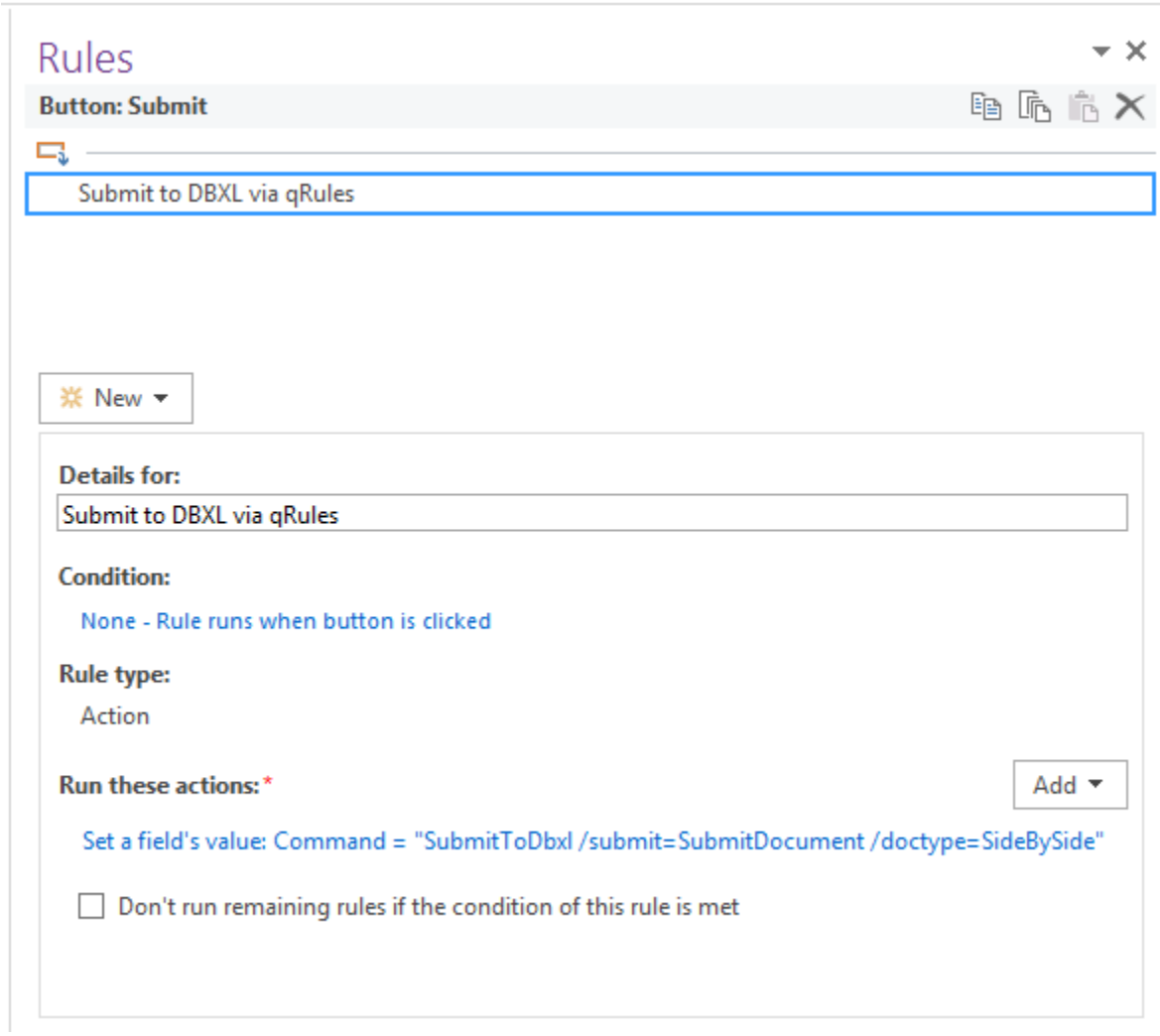
44. For the **Field**, select the Command node in the QdabraRules secondary data source.

45. For the **Value**, enter:

SubmitToDbxl /submit=SubmitDocument /doctype=SideBySide



46. Call this rule Submit to DBXL via qRules.



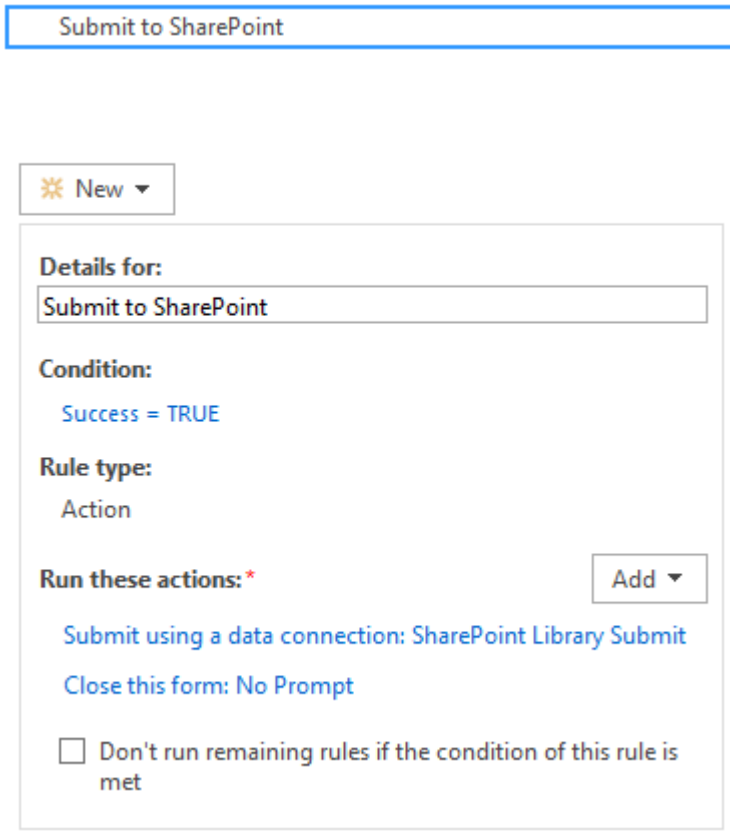
47. Click **New** > **Action** > **Add** > **Submit Data**.

48. Select the **SharePoint Form Library Submit** data connection and click **OK**.

49. Click **Add** > **Close the form** and then click OK.

50. Call this rule “Submit to SharePoint”.

51. Give this rule a condition. We want to check that the DBXL Submit was succesful. To do so, check that the Success node in QdabraRules data connection is equal to TRUE.



52. Save the updated form by selecting **File** > **Save**.

CONVERT SUBMITDOCUMENT TO UDC

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.

53. In your SharePoint site, create a Data Connections library (if one does not already exist).

54. In InfoPath designer, go to **Data** > **Data Connections**.

55. Select the SubmitDocument data connection and click “Convert to connection file”.

56. Enter the URL for the new UDC file in this format:
    http://<servername>/Data%20Connections/<connectionName>.udcx, and click **OK**.

PUBLISH THE FORM TO THE SHAREPOINT FORM LIBRARY

57. Click **File** > **Publish** > **SharePoint server**.

58. Walk through the InfoPath publishing wizard to publish the form to the form library created earlier.

SET UP THE DBXL DOCUMENT TYPE

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

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

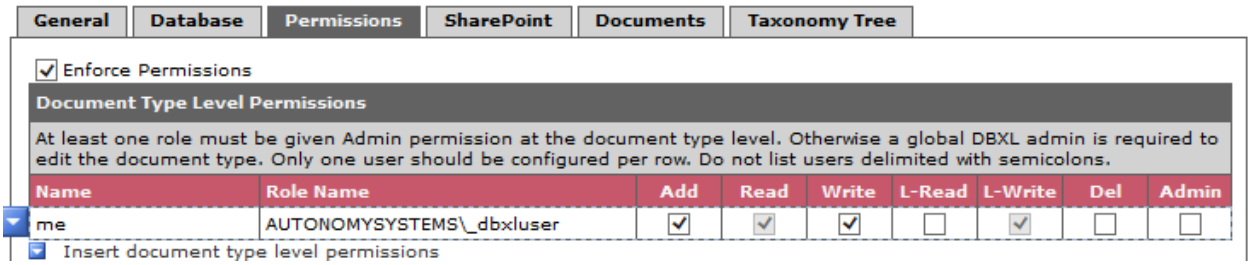
61. Once DAT opens in InfoPath, click **New Configuration**.

62. In the **General** tab, enter a name under **Document Type Details** > **Name**, and attach the XSN form template. You must use the same name used in the qRules SubmitToDbxl command.

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

64. Switch to the **Permissions** tab.

65. Check the checkbox for "Enforce Permissions", and give the DbxlUser Document Type permissions.

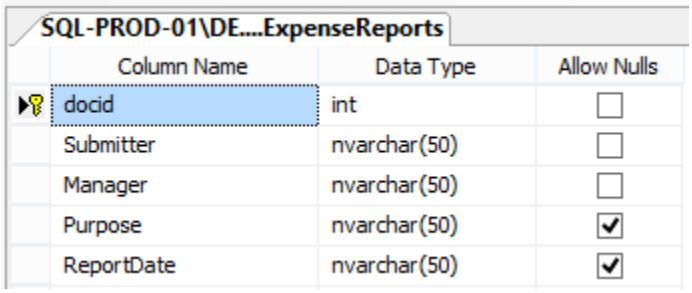


66. Save the document type.

CREATE A SQL TABLE

The following steps require SQL Server Management Studio.

67. 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.



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.

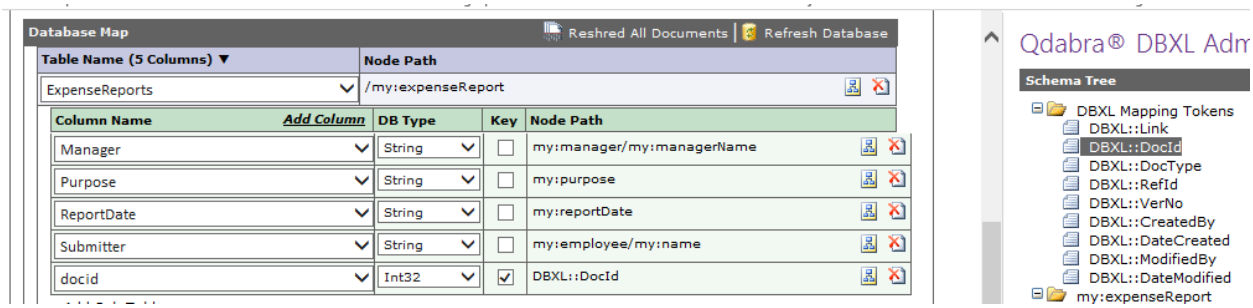
68. Configure a database connection
    a. In DAT, click on the **Database** tab.
    b. Enter the Data Connection String appropriate for your SQL server. For example, *Integrated Security=SSPI;Data Source=(LOCAL)\sqlexpress;Initial Catalog=Your_Database*

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.

69. 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.

70. 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.



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

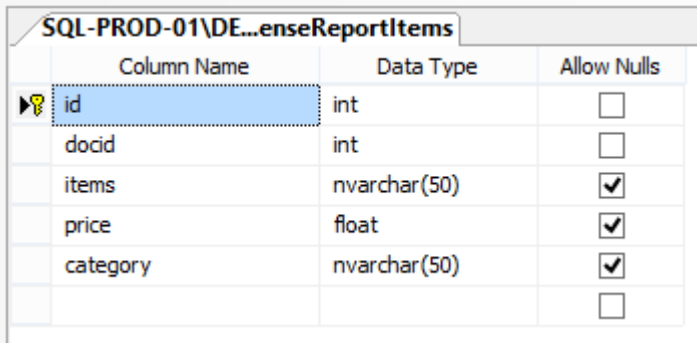
If you have existing documents in the DBXL Document Type, click **Reshred All Documents**. This will push the document's data into SQL, based on the current Database Mapping. You can use SQL Server Management Studio to verify that the form data has now been mapped to the SQL table.

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.

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



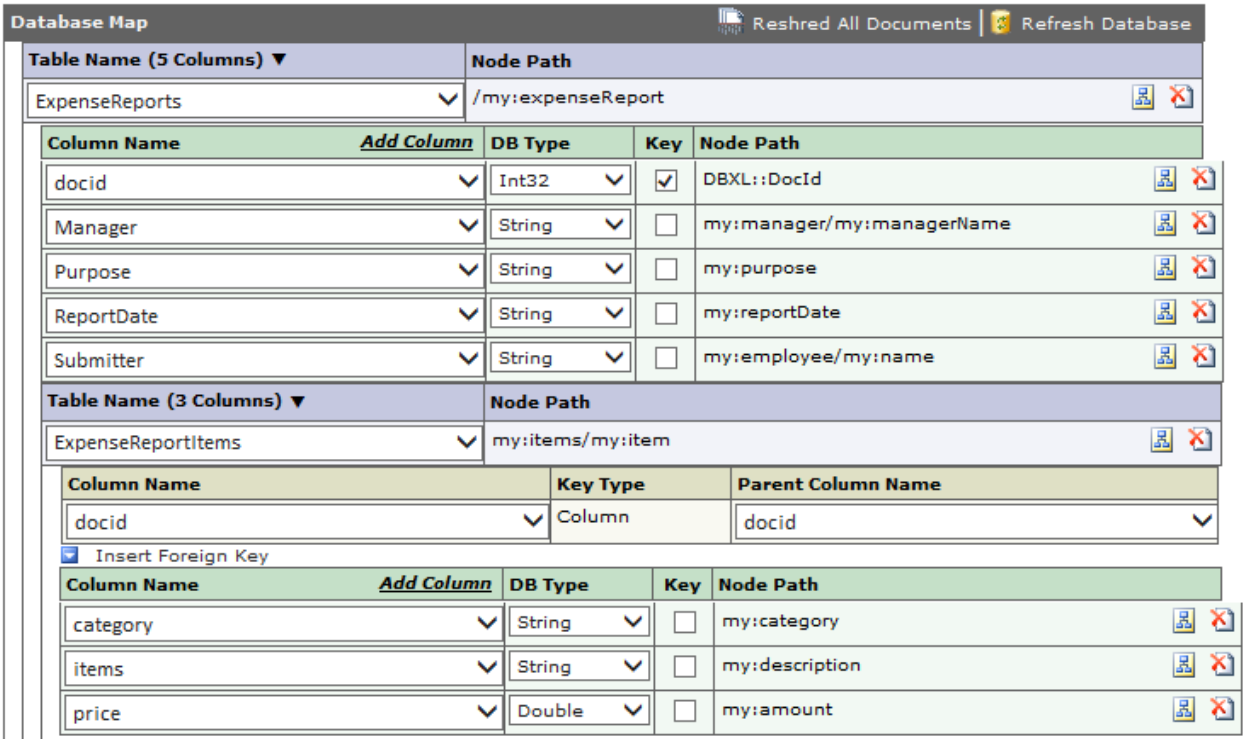
Why do we need the ExpenseReportId (**docid**) in the child table? Because this is what links our child data to its parent table.

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

74. 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.

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



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.

76. 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 smtphost 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.

77. 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.