



Getting Started with Database Accelerator (DBXL) v2.7

This Getting Started guide is an introductory tutorial that will show you how to quickly familiarize yourself with some of the features of DBXL v2.7 and to setup the Expense Report form sample in DBXL on your own. After completing this document you should be able to setup your own forms in DBXL and map to SQL. This document assumes that you have already installed DBXL v2.7 (Server or Client). If you have not installed DBXL 2.7 yet please see the DBXL v2.7 Installation Guide for more details.

WHAT NEXT?

After you complete this tutorial, visit the DBXL v2.7 Product page on Qdabra.com to find a list of [all available documents](#).

Qdabra offers DBXL training! The [videos are available here](#) and the [training modules are available here](#). All of the material covered in this Getting Started document (and more!) is included in the DBXL training.

EXPENSE REPORT SAMPLE

This tutorial uses the Expense Report sample form that is included in all InfoPath 2007 installations. For your convenience we have packaged the form with this Getting Started document, available for download on InfoPathDev.com. Although the Expense Report sample form was designed for InfoPath 2007 it will work with either InfoPath 2007 or InfoPath 2010. Instructions for both InfoPath 2010 and InfoPath 2007 are included below.

Note: DBXL v2.7 includes an Expense Report form that is installed and configured. The form is mapped to SQL and upon install the necessary tables were setup in your SQL server DB. You can use this form as-is and integrate with SharePoint right away and generate Excel or SSRS reports from the installed form. The sample document type is called **QdExpenseReport-Sample** (see Appendix for more information). Once you are finished with the tutorial below, you should end up with a solution similar to the QdExpenseReport-Sample included in DBXL.



CONTENTS

Task 1: Design the form	3
InfoPath 2010	3
Active Directory	3
Submit to DBXL	6
InfoPath 2007	8
Active Directory	9
Submit to DBXL	12
Task 2: Create a DBXL Document Type	15
Task 3: Create a SQL table to store data	17
Task 4: Configure SQL Mapping	17
Task 5: Modify permissions	21
Add another DBXL Administrator	21
Add a DBXL user	22
Task 6: Configure Event Viewer Notifications	22
Appendix	26
Support	27



TASK 1: DESIGN THE FORM

In the first task we will design the InfoPath form by adding connections to the DBXL web service.

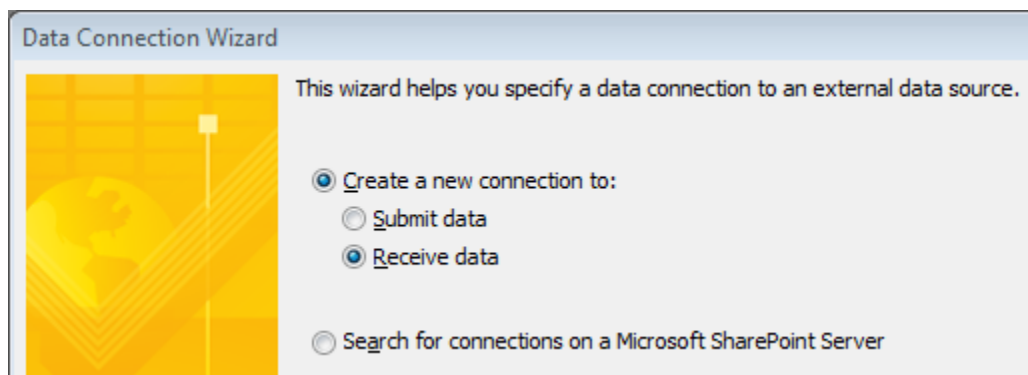
INFOPATH 2010

The following steps apply to users designing the form in InfoPath 2010.

ACTIVE DIRECTORY

Active Directory includes a wealth of information about your company's organization. This task will show you how to auto populate fields in your InfoPath form with the email address and full name of the currently logged on user.

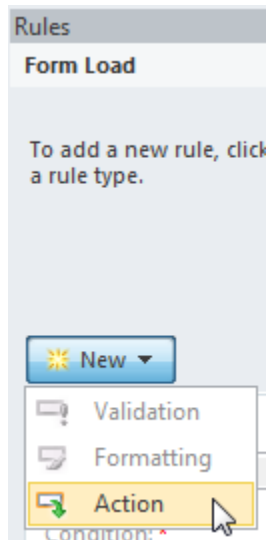
1. Open the Expense Report form template. To do this, right click on the Expense Report sample XSN on your local drive, and select **Design**.
2. Add the Active Directory Data Connection to your form.
 - a. In the InfoPath 2010 ribbon, select **Data** and then click on **Data Connections**.
 - b. Click **Add**.
 - c. Select the radio button to **Create a New Connection to** and **Receive Data**. Click **Next**.



- d. Select the radio button for **SOAP Web Service** and then click **Next**.
- e. Enter the URL of the service: **http://<yourservername>/QdabraWebService/ADUserInfo.asmx**. Click **Next**.
- f. Select the operation (Web Method) named **GetMyInfo**. Click **Next**.
- g. Click **Next**. There is no need to check the checkbox of **Store a copy of the data in the form template**.



- h. Uncheck the **Automatically retrieve data when form is opened** checkbox, click **Finish**, and then click **Close**.
3. Change the **Employee Email** and **Name** fields to auto-populate.
 - a. In the Data tab of InfoPath 2010, click **Form Load**.
 - b. Click **New** and select **Action**.



- c. Click the blue text under **Condition**.

Condition:
[None - Rule runs when form is ...](#)

- d. We will enter a condition because we only want to load this data once, when the form is first created. Select "**Select a field or group**" from the first dropdown and select the **emailAddress** under the **Employee** node. In the middle dropdown, select "**Is blank**". The condition should look as shown below. Click **OK**.

- e. Next to "**Run these actions**", click **Add** and select "**Query for data**".
- f. Select the **GetMyInfo** data connection and click **OK**.



Rule Details

Action:

Query for data

Data connection:

GetMyInfo

Add...

- g. Next to “Run these actions”, click **Add** and select “Set a field’s value”.
- h. Click the icon to the right of “Field” and select the **emailAddress** node under the **Employee** node.
- i. Click the icon to the right of “Value” and click **Insert field or Group**.
- j. Use the dropdown to switch to the **GetMyInfo** secondary data connection.
- k. Expand the **datafields** node, select the **Value** node, and click **Filter Data**.
- l. Click **Add**. We need to add a filter because the Active Directory web service returns multiple keys and their values. At this point, we want only the email.
- m. Select **Key** in the first dropdown and “is equal to” in the middle dropdown. In the third dropdown, select **Type text** and type in “mail” (no quotes).

Specify Filter Conditions

Display data that meets the following conditions:

Key is equal to *mail*

And >> Delete

OK Cancel

- n. Click OK until all dialogs are closed.
- o. Now we’ll set the employee name field. Next to “Run these actions”, click **Add** and select “Set a field’s value”.
- p. Click the icon to the right of “Field” and select the **name** field under the **Employee** node.
- q. Click the icon to the right of “Value” and click **Insert field or Group**.
- r. Use the dropdown to switch to the **GetMyInfo** secondary data connection.
- s. Expand the **dataFields** node, select the **Value** node, and click **Filter Data**.



- t. Click **Add**. We need to add a filter because the Active Directory web service returns multiple keys and their values. At this point, we want only the name.
- u. Select **Key** in the first dropdown and “**is equal to**” in the middle dropdown. In the third dropdown, select **Type text** and type in “**name**” (no quotes).
- v. Click **OK** until all dialogs are closed.
- w. Switch to the **Home** tab in the InfoPath 2010 ribbon and click **Preview**. You will see the current user’s data:



EXPENSE REPORT

Report Date: 2/27/2011 Expense Code:

Business Purpose:

Employee Information

Name: Ernesto Machado

Department:

E-mail Address: Ernesto.Machado@qdabra.com

- x. Click on **Close Preview**.

SUBMIT TO DBXL

1. Configure a Submit button to use the DBXL web service.
 - a. Switch to the **Data** tab in the InfoPath 2010 ribbon and click on **Submit Options**.
 - b. In the dropdown under “Send form data to a single destination”, select **Web Service**.
 - c. Click **Add**. This will load the Data Connection Wizard.
 - d. In the web service field, enter
<http://<yourservername>/qdabrawebservice/DbxlDocumentService.asmx> and click **Next**.
 - e. Select **SubmitDocument** and click **Next**.
2. Configure the parameters.



- a. Ignore the first parameter, called **tns:docTypeName**. It will default to the DocType of the form.
- b. Select the second parameter, named **tns:xml**.
 - i. Under **Parameter options** select the **Entire form (XML document, including processing instructions)** radio button.
 - ii. 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

- c. OPTIONAL: Select the **tns:name** parameter. You will notice that steps involved in subsections c, d and e has similar steps. This parameter, along with the Author and Description, are called the "DBXL tokens". These are displayed in the document list in the Documents tab in DAT. Setting a value for these are not required.
 - i. Select the radio button **Field or Group** in the **Parameter options** section.
 - ii. Click the Modify button next to the **Field or Group** text box. The **Select a Field or Group** dialog will appear.
 - iii. Select the *expenseReport/employee/name* field, click **OK** and check the **Submit data as a string** checkbox.



Parameters:

Parameter	Type	Element
tns:docTypeName	string	
tns:xml	string	/
tns:name	string	/my:expenseReport/my:e.

Parameter options

Submit the following data for the selected parameter:

☒ **Field or group:** /my:expenseReport/my:employee/my:name 

Include: Text and child elements only 

☐ **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.

- d. Select the **tns:author** parameter
 - i. Select the radio button **Field or Group** in the **Parameter options** section
 - ii. Click the Modify button next to the **Field or Group** text box. The **Select a Field or Group** dialog will appear.
 - iii. Select the *expenseReport/employee/emailAddress*, click **OK** and check the **Submit data as a string** checkbox.
 - e. Select the **tns:description** parameter.
 - i. Select the radio button **Field or Group** in the **Parameter options** section
 - ii. Click the Modify button next to the **Field or Group** text box. The **Select a Field or Group** dialog will appear.
 - iii. Select the *expenseReport/purpose*, click **OK** and check the **Submit data as a string** checkbox.
 - f. Click **Next**. Click **Finish**.
 - g. Back in the **Submit Options** dialog, enter **Submit** into the **Label** field.
 - h. Click **Advanced**. In the **After Submit** dropdown, select *Close the form*. Click **OK**.
3. Save the updated form by selecting **File > Save**. Then close the InfoPath 2010 designer.

INFOPATH 2007

These steps are for users designing the form in InfoPath 2007.



<http://www.qdabra.com>

Last updated on 1/5/2013 6:57 PM

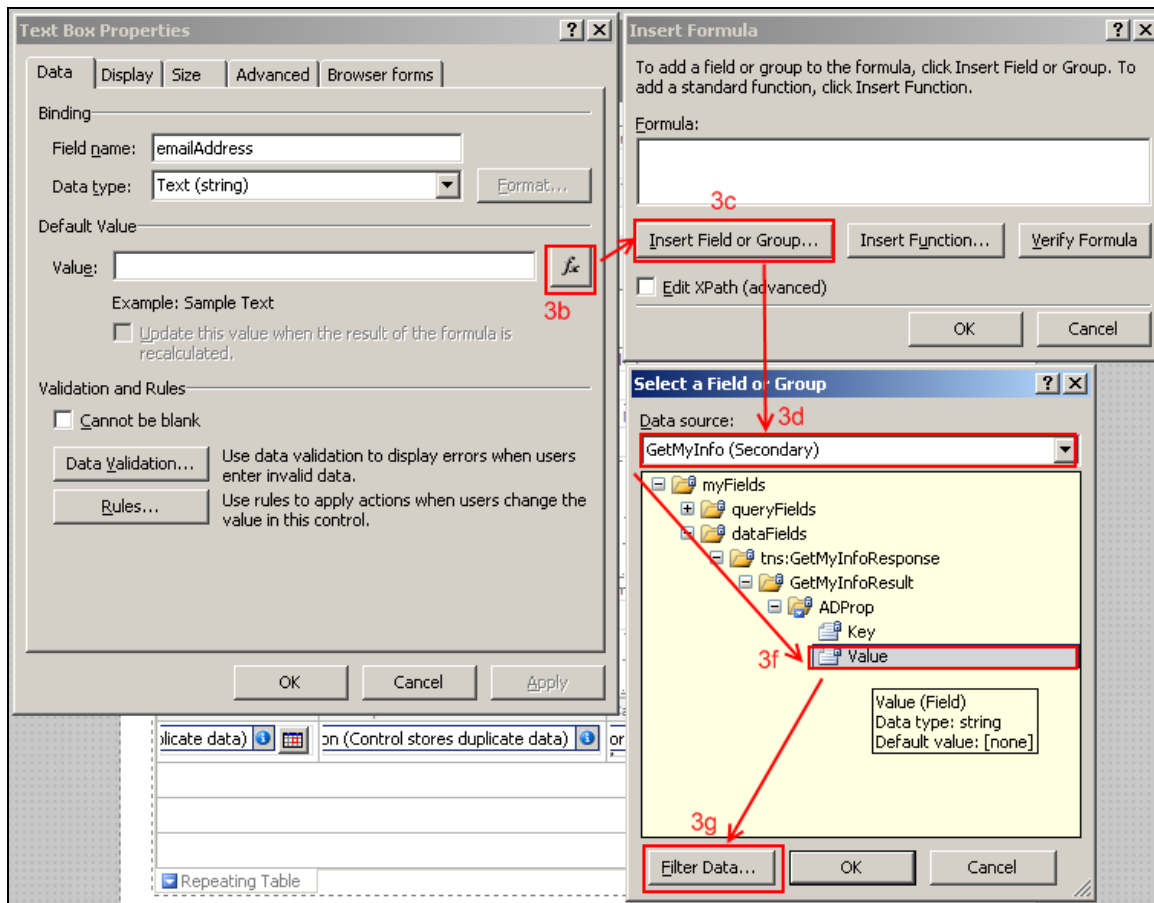
Copyright © 2006-2012 Autonomy Systems, LLC. All rights reserved.

ACTIVE DIRECTORY

Active Directory includes a wealth of information about your company's organization. This task will show you how to auto populate fields in your InfoPath form with the email address and full name of the currently logged on user.

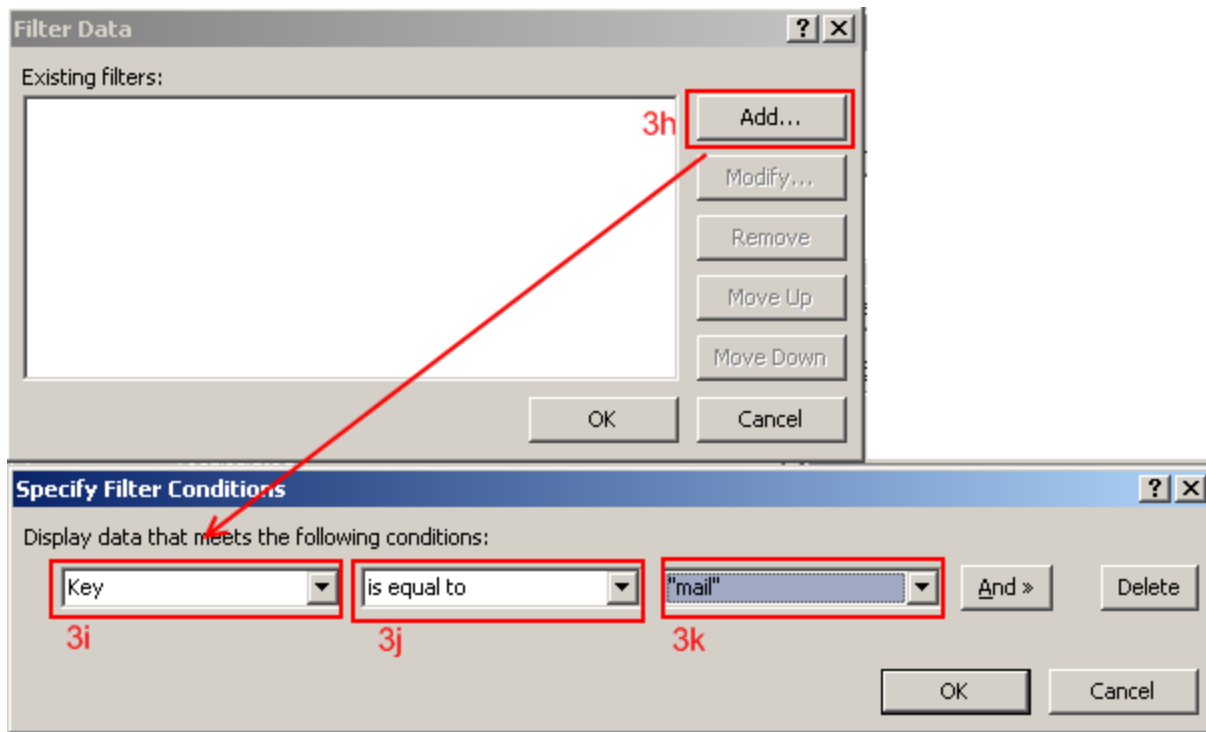
1. Open the Expense Report form template. To do this, right click on the Expense Report sample XSN on your local drive, and select **Design**.
2. Add the Active Directory Data Connection to your form.
 - a. Click **Tools** and select **Data Connections**.
 - b. Click **Add**.
 - c. Select the radio button to **Create a New Connection to** and **Receive Data**. Click **Next**.
 - d. Select the radio button for **Web Service**. Click **Next**.
 - e. Enter the URL of the service: `http://<yourservername>/QdabraWebService/ADUserInfo.asmx`. Click **Next**.
 - f. Select the operation (Web Method) named **GetMyInfo**. Click **Next**.
 - g. Click **Next**. There is no need to check the checkbox of **Store a copy of the data in the form template**.
 - h. Click **Finish**, making sure that the **Automatically retrieve data when form is opened** is checked, and then click **Close**.
3. Change the **Email Address** field to auto-populate.
 - a. Double click on the Employee **E-mail Address** field in the view.
 - b. In the **Default Value** section, click on the **fx** button to the right of the field.
 - c. Click **Insert Field or Group**.
 - d. Select the **GetMyInfo (Secondary)** data source from the dropdown
 - e. Expand `myFields/dataFields/tns:GetMyInfoResponse/GetMyInfoResult/ADProp`
 - f. Select the **Value** node.
 - g. Click **Filter Data**.





- h. On the **Filter Data** dialog, click **Add**.
- i. Select **Key** in the first drop down.
- j. Leave the second dropdown as **is equal to**.
- k. In the third dropdown, select the option **Type text**, then type **mail**. Notice that InfoPath adds double quotes after you type and tab out.





- l. Click **OK** five times to close the dialog boxes.
4. Change the Employee **Name** field to auto-populate.
 - a. Double click on the Employee **Name** field in the view.
 - b. In the **Default Value** section, click on the **fx** button to the right of the field.
 - c. Click **Insert Field or Group**.
 - d. Select the **GetMyInfo (Secondary)** data source from the dropdown.
 - e. Expand myFields/dataFields/tns:GetMyInfoResponse/GetMyInfoResult/ADProp
 - f. Select the **Value** node and click **Filter Data**.
 - g. On the **Filter Data** dialog, click **Add**.
 - h. Select **Key** in the first drop down.
 - i. Leave the second dropdown as **is equal to**.
 - j. In the third dropdown select **Type text**, and then type the text **cn**. Notice that InfoPath adds double quotes after you type and tab out.
 - k. Click **OK** five times to close the dialogs.



5. Verify the results.
 - a. Select **File > Save**, and click **Preview**. (Click **Yes** on the Microsoft Office InfoPath Security Notice)
 - b. Verify that the fields are auto-populating correctly.
 - c. Click on **Close Preview**.

SUBMIT TO DBXL

1. Configure a Submit button to use the DBXL web service.
 - a. Double click on the **Submit** button to display properties.
 - b. Click on **Submit Options**.
 - c. Select **Web Service** from the first dropdown.
 - d. Click the **Add** button.
 - e. Type in the location of the Qdabra DBXL Web Service in the format
`http://<yourservername>/QdabraWebService/DbxlDocumentServiceEx.asmx`.
 - f. Click **Next**. InfoPath will contact the server.
 - g. In the list of web method operations, choose the web method to use for submitting the form, which is called **SubmitDocument**.
 - h. Click **Next**.
2. Configure the parameters.
 - a. Ignore the first parameter, called **tns:docTypeName**. It will default to the DocType of the form.
 - b. Select the second parameter, named **tns:xml**.
 - i. Under **Parameter options** select the **Entire form (XML document, including processing instructions)** radio button.
 - ii. 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)

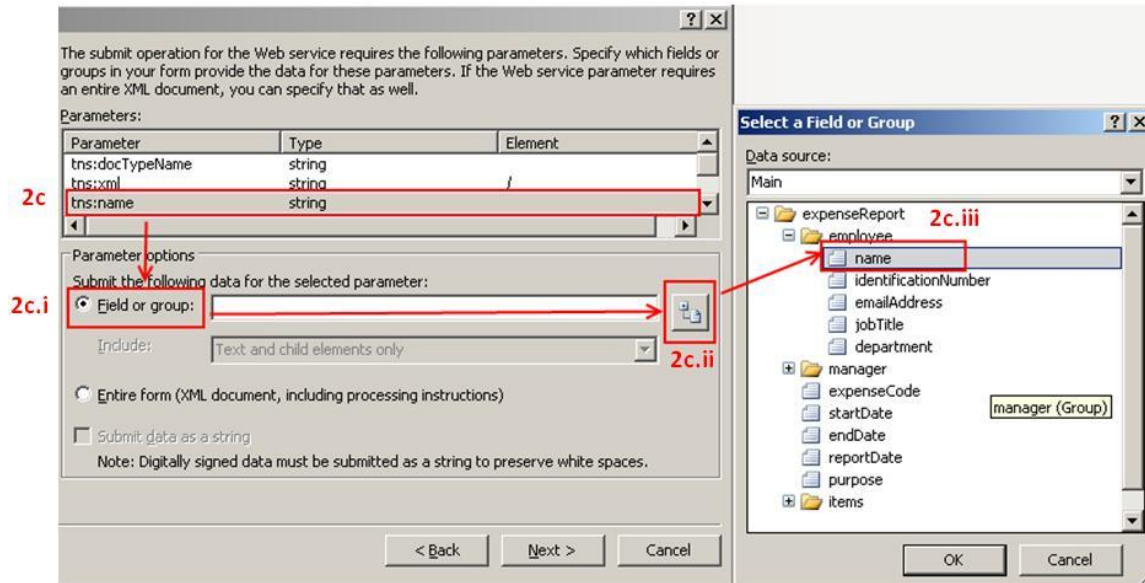
☒ Submit data as a string

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

< Back Next > Cancel

- c. Select the **tns:name** parameter. You will notice that steps involved in subsections c, d and e has similar steps. This parameter, along with the Author and Description, are called the “DBXL tokens”. These are displayed in the document list in the Documents tab in DAT. Setting a value for these is not required.
 - i. Select the radio button **Field or Group** in the **Parameter options** section.
 - ii. Click the Modify button next to the **Field or Group** text box. The **Select a Field or Group** dialog will appear.
 - iii. Select the *expenseReport/employee/name* field, click **OK** and check the **Submit data as a string** checkbox.





- d. Select the **tns:author** parameter
 - i. Select the radio button **Field or Group** in the **Parameter options** section
 - ii. Click the Modify button next to the **Field or Group** text box. The **Select a Field or Group** dialog will appear.
 - iii. Select the *expenseReport/employee/emailAddress*, click **OK** and check the **Submit data as a string** checkbox.
- e. Select the **tns:description** parameter.
 - i. Select the radio button **Field or Group** in the **Parameter options** section
 - ii. Click the Modify button next to the **Field or Group** text box. The **Select a Field or Group** dialog will appear.
 - iii. Select the *expenseReport/purpose*, click **OK** and check the **Submit data as a string** checkbox.
- f. Click **Next**. Click **Finish**.
3. Configure the form to close upon submit
 - a. Click **Advanced** in the **Submit Options** window.
 - b. In the **After Submit** dropdown, select *Close the form*.
 - c. Click **OK** twice.
4. Save the updated form.

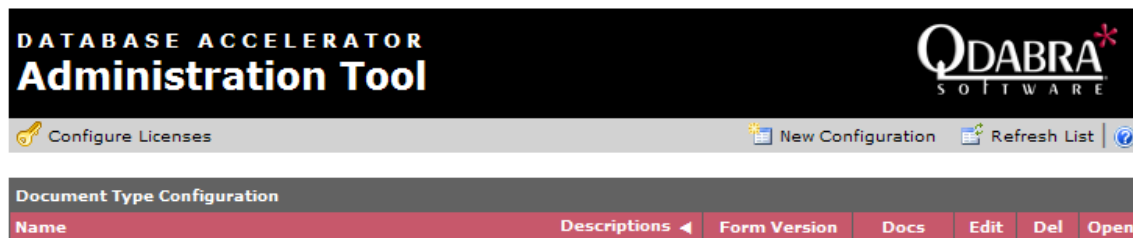


- a. Click **Save** on the **File** menu and save the form to a location on your computer. You will use this template in Task 2.
- b. **Close** the form.

Tasks 1 and 2 are covered as Module 1 in Qdabra's DBXL training, available by [clicking here](#). Active Directory is covered by Training module 4, [available here](#).

TASK 2: CREATE A DBXL DOCUMENT TYPE

In this section, we will use the DBXL Administration Tool (DAT). Access it through Internet Explorer by entering the URL <http://<<yourservername>>/QdabraWebService/default.htm>. Once the page opens, click the DBXL Administration Tool hyperlink. InfoPath should open a form similar to the one below:



1. Create a new configuration in DAT.
 - a. In DAT, click on **New Configuration**.
 - b. Type **ExpenseReport** into the **Name** field.
 - c. Under **InfoPath Form Template**, click on **Click here to attach a file**.
 - d. Select the Expense Report XSN and click **Insert**.
 - e. Click **Save**. DBXL will confirm the operation through a dialog box. Click **OK** to close it.
 - f. Click **Catalog**.



DATABASE ACCELERATOR Administration Tool

Document Type: Save (1e) Catalog (1f)

General Database Permissions Promoted Properties Documents Taxonomy Tree

Document Type Details

Name: 1b * Solution Group: [Dropdown] [Add]

Description: [Text Area]

InfoPath Form Template

XSN File: [Click here to attach a file] * 1c

Form Template Properties

Form Version: [Text Field]

Last Modified: [Text Field]

☐ Form template is signed

URL to Form Template: [Text Field]

Ref. ID

Format for Ref. ID	Base Reference #	XPath to Inject Ref. ID Into (Leave Blank to Disable)
	0	

Indexes

Name	XPath
☒ Insert index	

2. Verify that you can open the form.

- a. In the **ExpenseReport** row click the InfoPath icon under **Open**. The form will open and you will be able to fill it out.

ExpenseReport	1.0.0.60	Z	[Edit]	[Delete]	[InfoPath Icon (2a)]
---------------	----------	---	--------	----------	----------------------

3. Verify Submit works.

- a. Fill out the form including the **Business Purpose**. **Employee Name** and **Employee Email Address** fields should be auto-populated. When done, click the **Submit** button. The form will automatically close.
- b. Back in the DAT Catalog, click **Refresh List**. You'll see that the number in the Docs column has increased
- c. You can also click Edit for the Document Type, and then click the **Documents** tab. The new document is added to the Documents table.



Tasks 1 and 2 are covered as Module 1 in Qdabra's DBXL training, available by [clicking here](#).

TASK 3: CREATE A SQL TABLE TO STORE DATA

The following steps assume that you have SQL Server Management Studio (Express or full version)

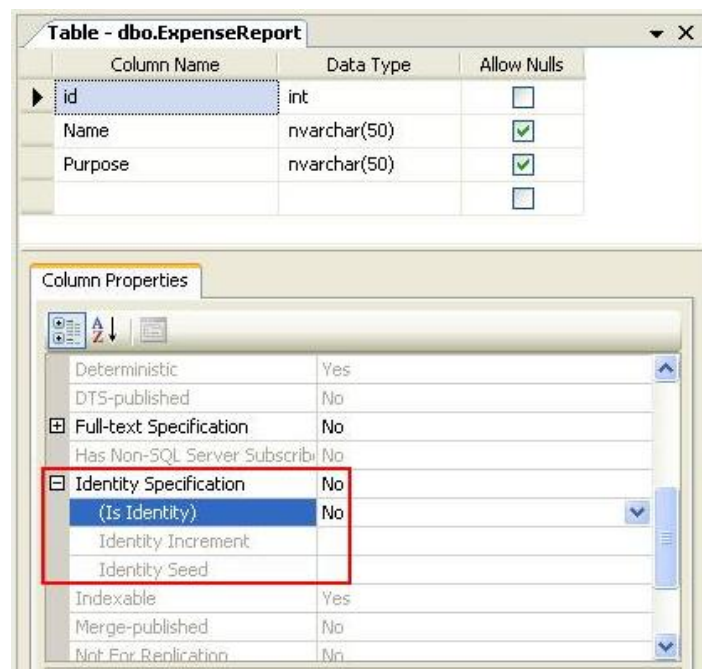
1. Design the database
 - a. Open SQL Server Management Studio and connect to the SQL instance you wish to shred into.
 - b. Create a database and a table to store your data. In this tutorial, we created a database named **DBXLTutorial** that contains the **ExpenseReport** table.
 - c. To prevent shredding errors when trying to store information greater than the defined column length, set the appropriate columns to allow the maximum textbox length in InfoPath. For this exercise, use the following data type information for the columns needed:

id: int

Name: nvarchar(50)

Purpose: nvarchar(50)

- d. Open the **Column Properties** dialog box for the **id** column, and scroll down until you find the **Identity Specification** section. Expand this section, and make sure **(Is Identity)** is marked as **No**.



- e. Make sure that the **Allow Nulls** option is checked for each of the columns, except id, and then save the table.

TASK 4: CONFIGURE SQL MAPPING



<http://www.qdabra.com>

Last updated on 1/5/2013 6:57 PM

Copyright © 2006-2012 Autonomy Systems, LLC. All rights reserved.

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

1. 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.

The screenshot shows the 'Database Configuration' window in the DAT application. The 'Database' tab is active. The 'Connection String' field is populated with 'Data Source=(LOCAL);Initial Catalog=DBXLTutorial;Integrated Security=True'. A 'Test Connection' button is located to the right of the connection string field and is highlighted with a red rectangular box. Below the connection string field, there is a section titled 'Submit Options' containing two checked checkboxes: 'Enable History' and 'Report Shredding Errors to the Form User'.

- d. Click **Save**, and then click **OK** in the confirmation dialog.
2. Configure the database mapping
 - a. Under **Database Map** section, click **Add Table**.
 - b. Click on the **Select Database Table** icon found next to the Table Name field. This will load the SQL schema in the Task Pane.
 - c. In the Database Tree, double-click on the table into which you wish to save data, which we have assumed is called **ExpenseReport**.
 - d. 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.
 - e. Double-click on the *my:expenseReport* node.
 - f. Click the **Add Column** link.
 - g. Click on the **Select Database Column** icon.
 - h. In the Database Tree, click the plus icon to expand the **ExpenseReport** table, and double-click on the **Name** column.
 - i. Click on the **Select Schema Node** icon and double click on the *my:employee/my:name* node.
 - j. Click the **Add Column** link.



- k. Click on the **Select Database Column** icon.
- l. In the Database Tree, expand the **ExpenseReport** table and double click on the **Purpose** column.
- m. Click on the **Select Schema Node** icon and double click on the *my:purpose* node.
- n. Click the **Add Column** link.
- o. Click on the **Select Database Column** icon.
- p. In the Database Tree, expand the **ExpenseReport** table and double-click on the **id** column.
- q. Click on the **Select Schema Node** icon and double click on the *DBXL::DocId* node.
- r. Check the **Key** checkbox.
- s. Click **Save** in the DAT header, and then click **OK** in the confirmation dialog.

Your completed database mapping should look like this:

The screenshot shows the QDABRA Administration Tool interface. The top header includes the title "DATABASE ACCELERATOR Administration Tool" and the QDABRA SOFTWARE logo. Below the header, the "Document Type" is set to "ExpenseReportTest". The interface has several tabs: "General", "Database", "Permissions", "SharePoint", "Documents", and "Taxonomy Tree". The "Database" tab is selected, showing the "Database Configuration" section with fields for "Connection String" (Data Source=(LOCAL);Initial Catalog=DBXLTutorial;Integrated Security=True) and "Submit Options" (Enable History and Report Shredding Errors to the Form User are checked). Below this is the "Database Map" section, which contains a table mapping database columns to schema nodes.

Table Name (3 Columns) ▼	Node Path		
ExpenseReport	/my:expenseReport		
Column Name	DB Type	Key	Node Path
id	Int32	☒	DBXL::DocId
Name	String	☐	my:employee/my:name
Purpose	String	☐	my:purpose

Below the table are links: "Add Column", "Add Sub Table", and "Add Table".

3. Test reshredding

- a. Click **Reshred All Documents** and click **Yes** on the warning dialog.



- b. Click **OK** in the confirmation dialog.
 - c. Use SQL Server Management Studio to verify that a document (added in Task 2) has now been mapped to your table.
4. Submit a document and verify the SQL table
- a. Click on the **General** tab.
 - b. Click **Open**.
 - c. Fill out the form including the Business Purpose. Employee Name and Employee Email Address fields should be auto-populated. When done, click the **Submit** button.
 - d. Click the **Documents** tab in DAT. The new document is added to the Documents column.
 - e. Use SQL Server Management Studio to verify that the new document has been mapped to your table.

You will notice that DBXL comes with a Document Type called QdExpenseReport-Sample. In the DAT catalog view, click **Edit** for this Document Type, then switch to the Database tab. You'll see that the QdExpenseReport-Sample comes with a sample database mapping (screenshot below). Compare this to the database mapping you have implemented above to further your understanding.



DATABASE ACCELERATOR Administration Tool



Document Type: QdExpenseReport-Sample

 Save
  Save As...
  Catalog
 

General
Database
Permissions
SharePoint
Documents
Taxonomy Tree

Database Configuration

Connection String



 Data Source=(LOCAL);Initial Catalog=QdabraUtilityv26_1109;Integrated Security=True

Submit Options

☒ Enable History
 ☒ Report Shredding Errors to the Form User

Database Map

 Reshed All Documents
  Refresh Database

Table Name (10 Columns) ◀	Node Path
QdExpenseReport	/my:expenseReport

Table Name (5 Columns) ◀	Node Path
QdExpenseReportEmployee	my:employee

Column Name	Key Type	Parent Column Name
ExpenseReportId	Column	DocId

☒ Insert Foreign Key
[Add Column](#) [Add Sub Table](#)

Table Name (9 Columns) ◀	Node Path
QdExpenseReportItem	my:items/my:item

Column Name	Key Type	Parent Column Name
ExpenseReportId	Column	DocId

☒ Insert Foreign Key

Task 4 is covered in a YouTube video demo, which can be accessed by [clicking here](#). It is also covered as part of Day 1 of Qdabra's DBXL Training, available by [clicking here](#). Finally, the Training module that corresponds to this Task can be found by [clicking here](#).

TASK 5: MODIFY PERMISSIONS

As the DBXL Administrator you can add permissions for other users. Here is how to edit these permissions:

1. In the Qdabrawebservice folder where DBXL was installed, find the web.config file.
2. Open the file (in a text editor, like Notepad) and look for the following: **ReaderGroupAlias**, **WriterGroupAlias**, **AdminGroupAlias** and **DbxlAdminAlias**.
3. Each of these Keys will be assigned a value (that is, a group).

ADD ANOTHER DBXL ADMINISTRATOR



<http://www.qdabra.com>

Last updated on 1/5/2013 6:57 PM

Copyright © 2006-2012 Autonomy Systems, LLC. All rights reserved.

During installation, DBXL creates a group called DBXLAdmins on the machine where DBXL was installed. That group, by default, is assigned to **AdminGroupAlias** and **DbxlAdminAlias**. To add a DBXL Administrator, simply add users to that group.

ADD A DBXL USER

ReaderGroupAlias and **WriterGroupAlias** are blank by default on new installations of DBXL.

1. On the machine where DBXL was installed, create a group called **DBXLUsers** and add users to this group.
2. Back in web.config, change **ReaderGroupAlias**, **WriterGroupAlias** to use **<machinename>\DBXLUsers**.
3. Save web.config.

TASK 6: CONFIGURE EVENT VIEWER NOTIFICATIONS

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

1. 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

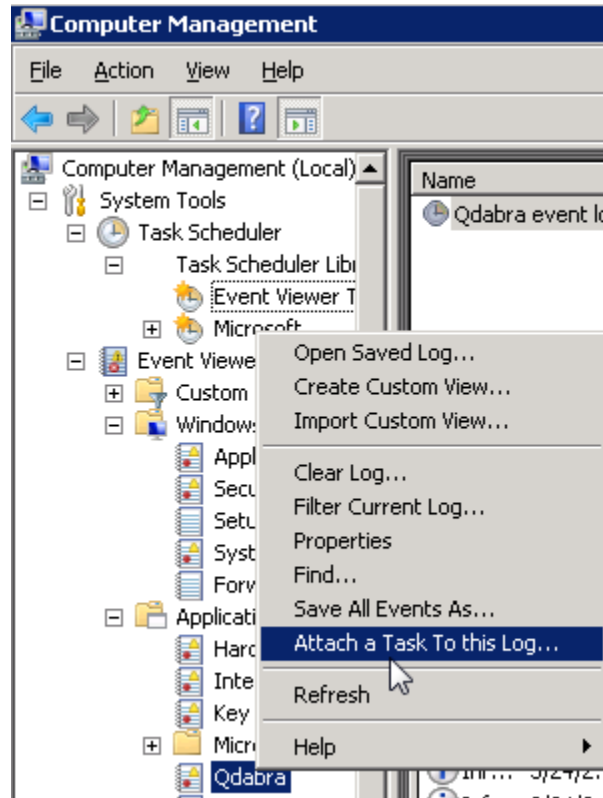
oMessage.Send

' 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 ""
```

2. Make any necessary modifications. You should update the **SMTP** server, the **from** email and the **to** email.
3. 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.

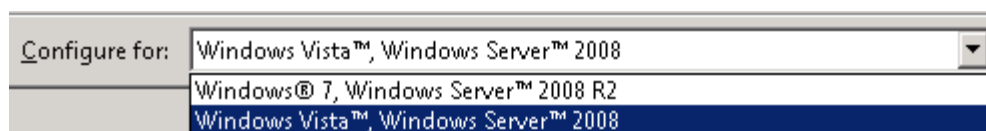


4. Open **Computer Management** and navigate to the Event Viewer.
5. Navigate to and select the Qdabra event log.
6. Right-click on the log and select **Attach Task to This Log**.

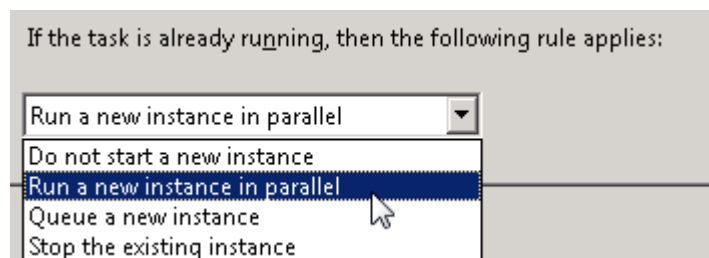


7. In the **Create Basic Task** wizard, type a name and description, then click **Next**.
8. In the **When a Specific Event is Logged** screen click **Next**.
9. In the **Action** window select **Start a program** and click **Next**.
10. Click **Browse**, select C:\Notifications\SendErrorEmail.vbs and click **Next**.
11. Check **Open the Properties dialog for this task when I click Finish**, and click **Finish**.
12. Under **Security options**, click **Change User or Group**.
13. Type in **SYSTEM**, click **Check Names** and click **OK**.
14. In the **Configure for** dropdown, select the correct option.



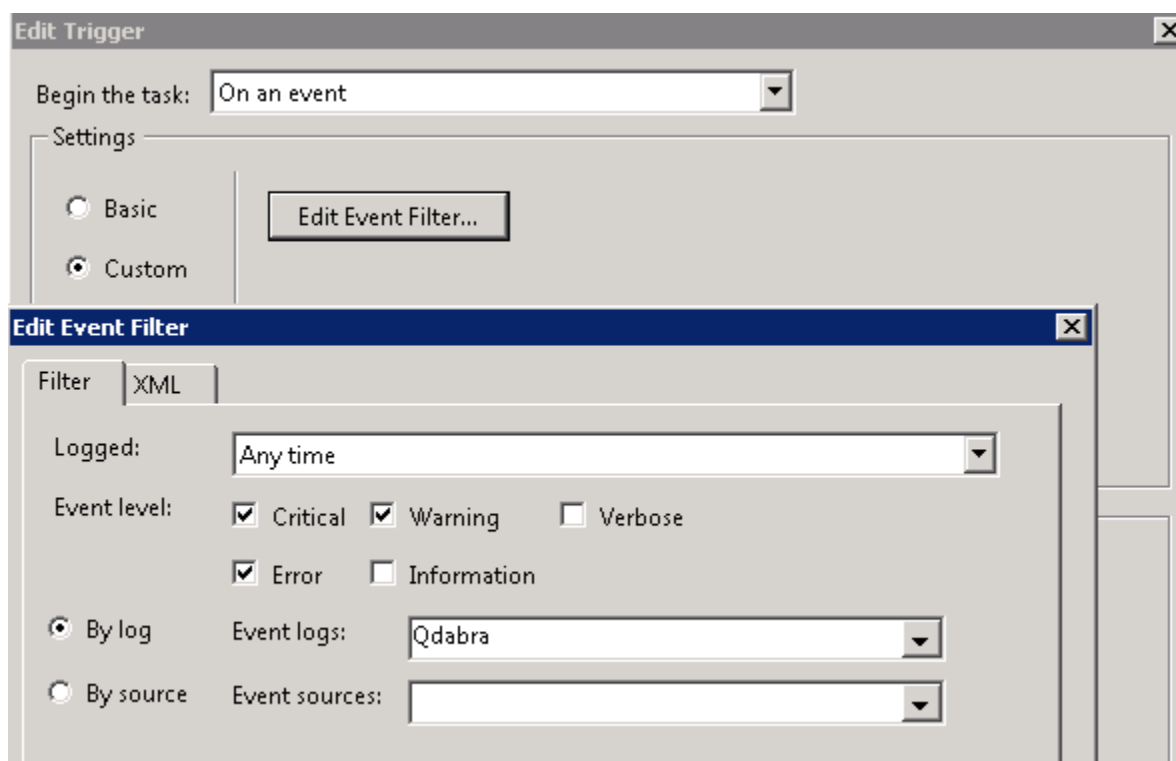


15. In the **Settings** tab, select



16. Click **OK**.

17. 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 specified in the vbs file will receive an email.



Qdabra Number of events: 1,242

Level	Date and Time	Source	Event ID	Task C...
Error	6/13/2012 2:53:50 PM	Database Accelerator	0	None
Error	6/13/2012 2:45:52 PM	Database Accelerator	0	None
Error	6/13/2012 2:39:36 PM	Database Accelerator	0	None
Information	6/13/2012 2:38:41 PM	Database Accelerator	0	None
Information	6/13/2012 2:38:38 PM	Database Accelerator	2	None
Information	6/13/2012 2:38:37 PM	Database Accelerator	8	None
Information	6/13/2012 2:38:36 PM	Database Accelerator	1	None
Information	6/12/2012 7:32:28 PM	Database Accelerator	4	None
Information	6/12/2012 6:42:18 PM	Database Accelerator	0	None

Event 0, Database Accelerator

General Details

An error occurred while setting up a SharePoint collection mapping with the administrative error details have been logged.

SharePoint URL:
<http://fail!>

Exception:
 System.Net.WebException: The remote name could not be resolved: '_vti_bin'
 at System.Net.HttpWebRequest.GetRequestStream(TransportContext& context)
 at System.Net.HttpWebRequest.GetRequestStream()

Log Name: Qdabra
 Source: Database Accelerator Logged: 6/13/2012 2:53:50 PM

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

Errors have been logged in Qdabra eventlog

ernesto.machado@qdabra.com

Sent: Wed 6/13/2012 6:54 PM

To: Ernesto Machado

Errors have been logged in the Qdabra eventlog

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.



<http://www.qdabra.com>

Last updated on 1/5/2013 6:57 PM

Copyright © 2006-2012 Autonomy Systems, LLC. All rights reserved.

APPENDIX

In this Appendix we provide additional information about the **QdExpenseReport-Sample** document type that is included in DBXL v2.7.

This document type includes:

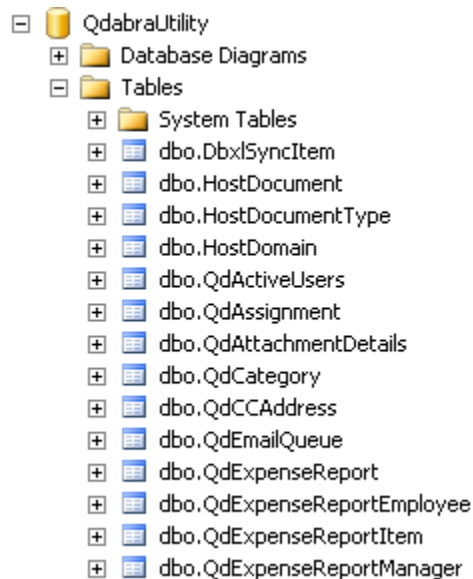
- A data connection that submits the form to the DBXL web service. This is described within Task 1 in this document.
- The document type itself, of course, which is described in Task 2.



Warning: You have expiring or expired License Packs.

Document Type Configuration						
Name	Descriptions ◀	Form Version	Docs	Edit	Del	Open
QdExpenseReport-Sample		1.0.0.4	0			

- SQL tables where the data from the Expense Report sample gets mapped. This is described in Task 3 above. The **QdExpenseReport** tables are located in the **QdabraUtility** database.



- Database (SQL) Mapping, which is described in Task 4.



<http://www.qdabra.com>

Last updated on 1/5/2013 6:57 PM

Copyright © 2006-2012 Autonomy Systems, LLC. All rights reserved.

DATABASE ACCELERATOR Administration Tool

Document Type: QdExpenseReport-Sample

Save
 Save As...
 Catalog

General
Database
Permissions
SharePoint
Documents
Taxonomy Tree

Database Configuration

Connection String

Test Connection

Data Source=(LOCAL);Initial Catalog=QdabraUtilityv26_1109;Integrated Security=True

Submit Options

☒ Enable History
 ☒ Report Shredding Errors to the Form User

Database Map

Reshed All Documents
 Refresh Database

Table Name (10 Columns) ▼	Node Path		
QdExpenseReport	/my:expenseReport		
Column Name	DB Type	Key	Node Path
CashAdvance	Decimal	☐	my:items/my:cashAdvance
DocId	Int64	☒	DBXL::DocId
EndDate	DateTime	☐	my:endDate
ExpenseCode	String	☐	my:expenseCode
Link	String	☐	DBXL::Link
Purpose	String	☐	my:purpose
Table Name (5 Columns) ◀	Node Path		
QdExpenseReportEmployee	my:employee		

You can use this database mapping as a reference when setting up your own. This Expense Report sample can also be extended to explore other DBXL features, such as SharePoint mapping or Active Directory.

SUPPORT

If you have questions about the information in this document, please contact us for assistance.

Licensed customers can contact us via Support@Qdabra.com.

Alternatively, please use the InfoPathDev.com Qdabra Product support forums to request help.



<http://www.qdabra.com>

Last updated on 1/5/2013 6:57 PM

Copyright © 2006-2012 Autonomy Systems, LLC. All rights reserved.