



Working with a qRules solution that uses DBXL and MOSS 2007

With DBXL, you can quickly integrate your solution with SharePoint 2007 and IPFS in order to create a browser-based solution. However, the challenge is that InfoPath Forms Server can only open documents from SharePoint document libraries. Connecting an existing form template with DBXL for use on IPFS can be done by using code to access the DBXL web services; but now that there's qRules, all this can be done without the need of writing code.

Since browser forms allow users to pass in custom parameters via the URL, we will pass in the DBXL document ID (DocID) using the qRules command called *GetInputParameter*. If passed in, we use a data connection to DBXL to retrieve the document then replace the main DOM with the retrieved XML using the new qRules command called *SwapDomWithDocument*.

This document details the steps on how to configure your qRules-based solution that can submit documents to DBXL and load them in IPFS. This requires qRules v2.1 (or later), MOSS 2007, and DBXL v2.3 (or later). For creating SharePoint data views, SharePoint Designer is required.

The document outlines the following:

- [Create the data connections](#)
- [Configure the qRules commands](#)
- [Configure the submit command](#)
- [IPFS-enable your form template](#)
- [Publish your form as an admin-approved form template](#)
- [Create an IPFS-enabled SharePoint library](#)
- [Create a document type in DBXL](#)
- [Test the solution](#)
- [Create a data view to access your DBXL forms](#)



CREATE THE DATA CONNECTIONS

Our solution will use two web service data adapters, one called *GetDocument* which will retrieve document data from DBXL, the other called *SubmitDocument* which will post a modified document back to DBXL.

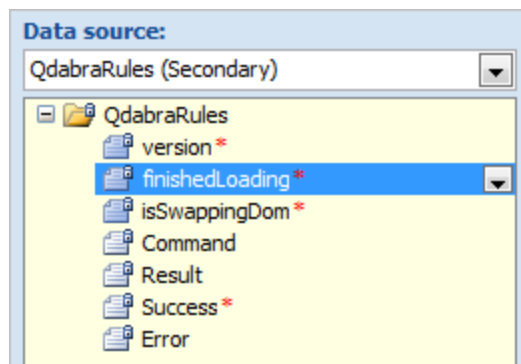
1. Design your qRules-injected form template in InfoPath.
2. Add the *GetDocument* data connection:
 - a. Go to **Tools > Data Connections** and click **Add**.
 - b. Select **Create a new connection to: Receive data** and click **Next**.
 - c. Select **Web Service** in the next screen and click **Next**.
 - d. For the Web Service URL, enter the location of the Qdabra DBXL Document web service. In default DBXL installations, the format would be:
<http://<YourServerName>/QdabraWebService/DbxlDocumentService.asmx>. Click **Next**.
 - e. Select **GetDocument** from the list of operations and click **Next**.
 - f. Double-click on the **tns:docId** parameter and enter the number '0' (zero), then click **Next**.
 - g. No value is needed for next **tns:docId** parameter so leave it blank and click **Next** twice.
 - h. Give your data connection a name of your choosing. For this document, we will use *GetDocument*. We will use this name later when we configure our qRules command.
 - i. Uncheck the box 'Automatically retrieve data when the form is opened' and click **Finish**.
3. Add the *SubmitDocument* data connection:
 - a. Back in the Data Connections dialog box, click **Add**.
 - b. Select **Create a new connection to: Receive data** and click **Next**.
 - c. Select **Web Service** in the next screen and click **Next**.
 - d. For the Web Service URL, enter the location of the Qdabra DBXL Document web service which is the same as what we used in the previous step
(<http://<YourServerName>/QdabraWebService/DbxlDocumentService.asmx>), and click **Next**.
 - e. Select **SubmitDocument** from the list of operations and click **Next** three times, leaving the rest of the defaults.
 - f. Give your data connection a name of your choosing. For this document, we will use *SubmitDocument*. We will use this name later when we configure our qRules command.
 - g. Uncheck the box 'Automatically retrieve data when the form is opened' and click **Finish**.
 - h. Click **Close** to exit out of the Data Connections dialog.



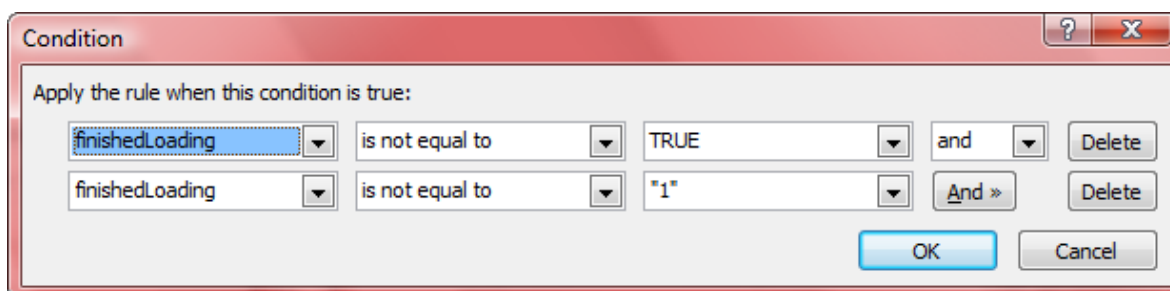
CONFIGURE THE QRULES COMMANDS

Now that we've added our data connections, we need to add rules to the *finishedLoading* attribute of the QdabraRules (Secondary) data source. This will trigger getting the input parameters on load and use those parameters with the *SwapDomWithDocument* command.

1. In the Data source taskpane, switch to the **QdabraRules (Secondary)** data source.



2. Right-click on the *finishedLoading* node and select **Properties**.
3. In the field properties window, switch to the 'Rules and Merge' tab and click **Add**.
4. The first rule will simply stop the succeeding rules to run if the conditions are met. Click on the **Set Condition** button and add two conditions as shown:



5. Click **OK**.
6. Check the box 'Stop processing rules when this rule finishes'. No action is needed for this rule. It should look similar to this:



Rule

Name: Stop if not true

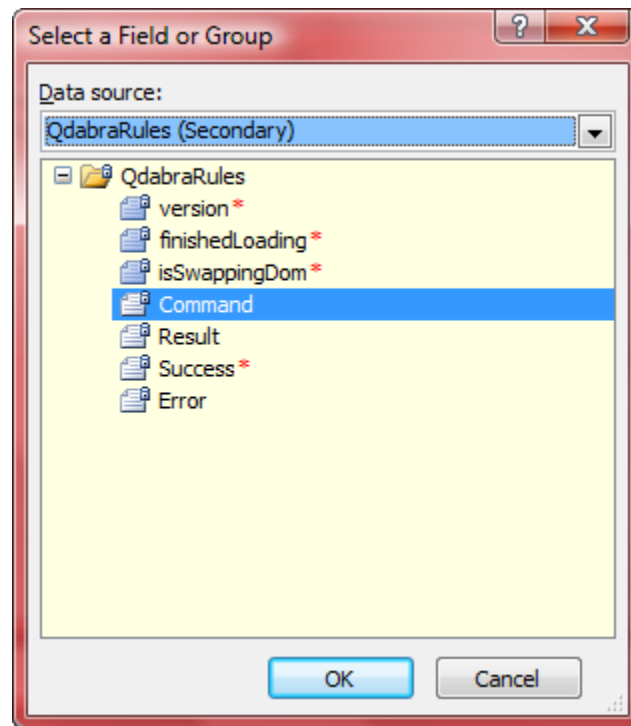
Condition: finishedLoading != TRUE and finishedLoading != "1" Set Condition...

Actions (run in the order shown):

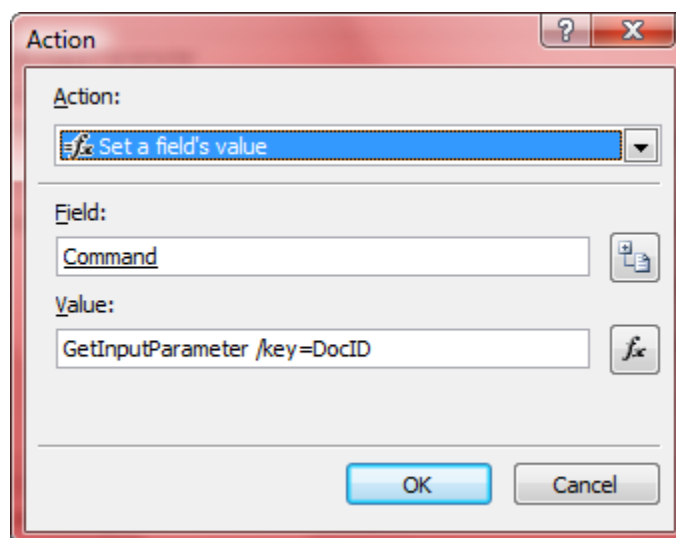
☒ Stop processing rules when this rule finishes

OK Cancel

7. Give your rule a name of your choosing and click **OK**.
8. Click **Add** to add another rule that will call the qRules *GetInputParameter* command and will use 'DocID' as the key.
9. Click **Add Action** and leave the first dropdown with the action "Set a field's value".
10. Click on the schema icon next to **Field** and select the **Command** node from the **QdabraRules (Secondary)** data source.



11. Click **OK**.
12. In the **Value** box, enter the command string: *GetInputParameter /key=DocID*. (Note that *DocID* is case-sensitive. It should match the specified parameter in the URL that we will set up in the later part of this document.) Your rule action should be similar to this:



13. Click **OK** twice and then click **Add** to add the third rule.



14. The third rule will stop the succeeding rule to run if the specified condition is met. Click on the **Set Condition** button and add a condition as shown:

15. Click **OK**.
16. Check the box 'Stop processing rules when this rule finishes'.

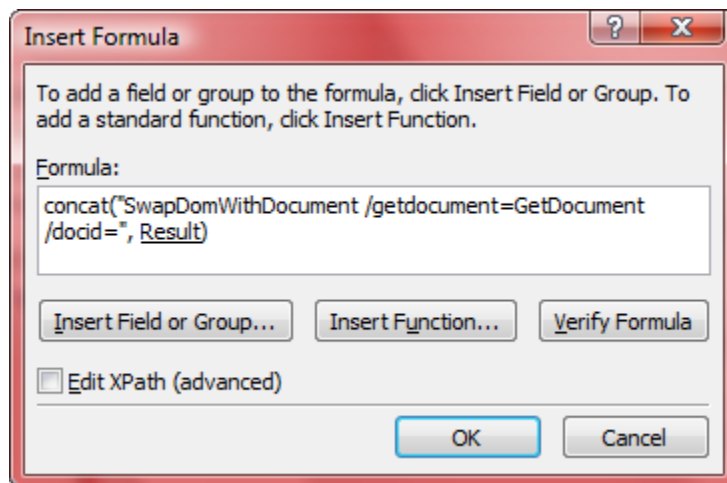
Again, no action is needed for this rule. It should look similar to this:

17. Click **OK**, then click **Add** to add the last rule that will call the qRules *SwapDomWithDocument* command and will use the resulting 'DocID' returned by the *GetDocument* data connection.
18. Click **Add Action** and leave the first dropdown with the action "Set a field's value".
19. For the **Field**, select the **Command** node from the **QdabraRules (Secondary)** data source.
20. For the **Value**, click on the **fx** button and enter the formula:

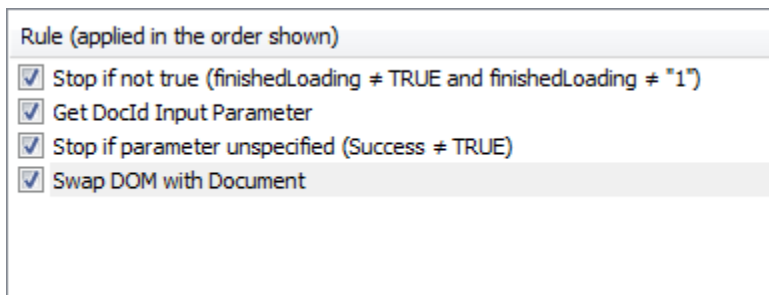
concat("SwapDomWithDocument /getdocument=GetDocument /docid=", Result)



where the **/getdocument** parameter is the name of your *GetDocument* data connection, and **/docid** is the **Result** node of the **QdabraRules (Secondary)** data source. Your formula should look like this:



21. Click **OK** three times. We now have four rules in the QdabraRules *finishedLoading* node:



22. Click **OK** to close the Field Properties window.



CONFIGURE THE SUBMIT COMMAND

We will use the qRules command called *SubmitToDbxl* via rules to enable our forms to submit to DBXL. For this, we will add a button that will trigger the submit rule. (For more detailed, step-by-step instructions, see the tutorial [Use qRules to submit to DBXL](#).)

The *SubmitToDbxl* qRules command allows specifying three values to use as metadata for each of the forms that are submitted to DBXL. These values are:

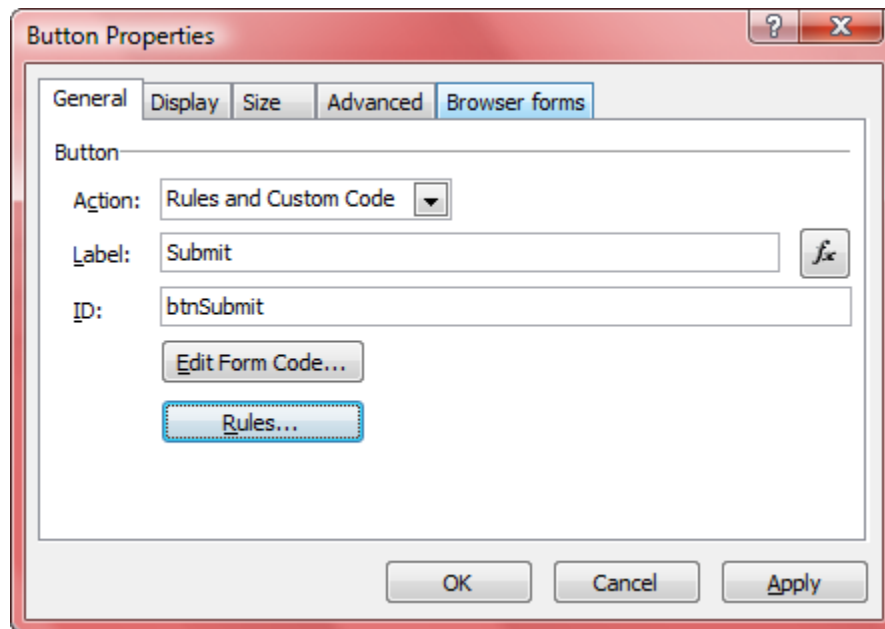
- **Name** – A title for each form
- **Author** – The name of the person who created the form, or who last modified it, depending on how you would like to use this value.
- **Description** – A description for each form

In order to make these values dynamic, we will use the fields in our form corresponding to each of these metadata values.

1. Add three text fields in your form, as shown:



2. Drag these fields on your form canvas.
3. Next, we will add our submit button. In the *Controls* taskpane, click on the **Button** control to add it onto your view.
4. Double-click on the button on your canvas to see its properties.
5. Label your button 'Submit' and give it an ID of your choosing, e.g. *btnSubmit*.



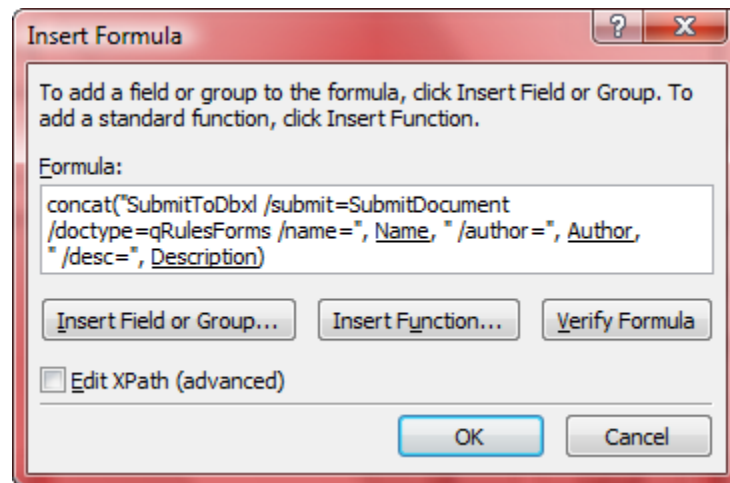
6. Click on **Rules**, and then **Add**.
7. Give your rule a name of your choosing and click **Add Action**.
8. Leave the first dropdown with the action "Set a field's value".
9. For the **Field**, select the **Command** node from the **QdabraRules (Secondary)** data source.
10. For the **Value**, click on the **fx** button and enter the formula:

**concat("SubmitToDbxl /submit=SubmitDocument /doctype=qRulesForms /name=", Name, "
/author=", Author, " /desc=", Description)**

where we are concatenating the values in the following parameters:

- **/submit** – name of the *SubmitDocument* data connection
- **/doctype** – specifying a value in this parameter is the same as 'hardcoding' the document type name. Since we want to submit forms via IPFS, we will have to declare the doctype in our qRules command. You may give it any name of your choosing, but remember that we will use this when we create our document type in DBXL later. For this document, we will use the name '*qRulesForms*'.
- **/name, /author, /desc** – the underlines indicate that these are the field that you created above. If you click the **Edit Xpath (advanced)** checkbox, you should see the full XPaths of the fields.

Your formula should look like this:



11. Click **OK** to close all the dialogs.
12. Our sample form will simply look like shown:

Name	
Author	
Description	
Submit	



IPFS-ENABLE YOUR FORM TEMPLATE

1. Go to **Tools > Form Options > Compatibility**, and enable browser-compatibility:

Browser compatibility

☒ Design a form template that can be opened in a browser or InfoPath

☐ Hide errors for code that uses InfoPath-only features

Enter the URL of a server that is running InfoPath Forms Services and can be used to verify compatibility:

Example: <http://www.contoso.com/>

Backward compatibility

☐ Show report on compatibility with InfoPath 2003

2. Make sure that the Submit button (since it makes use the qRules command) is set to ALWAYS send data to the server. To do this, double-click on the Submit button to show its properties, and go to the **Browser forms** tab, then select **Always**.

General Display 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)

3. Save your form template to your network or local hard drive. For the purposes of this document, we will name our form template '*qRulesForms*'.



PUBLISH YOUR FORM AS AN ADMIN-APPROVED TEMPLATE

Since qRules injects code into your forms, you will have to publish your XSN as an admin-approved form template. This allows you to manage the form in one place and use it in many places.

1. In InfoPath's menu, go to **File > Publish**.
2. Select the option to publish the form template 'To a SharePoint Server with or without InfoPath Forms Services' and click **Next**.

Where do you want to publish the form template?

☒ To a SharePoint server with or without InfoPath Forms Services

☐ To a list of e-mail recipients

☐ To a network location

☐ As an installable form template (requires Microsoft Visual Studio)

3. Enter the address of your SharePoint or IPFS site and click **Next**.
4. The next page will allow you to publish your form as an **Administrator Approved form template**. Make sure that option is selected and click **Next**.

☒ Enable this form to be filled out by using a browser

What do you want to create or modify?

☐ Document Library

Publish this form template as a template in a document library. A document library stores forms based on this form template. Users can open and fill out forms in the library. You can specify which fields in the template appear as columns in the library.

☐ Site Content Type (advanced)

A site content type allows this form template to be used in multiple libraries and sites. You can specify which fields in the template appear as columns in the library.

☒ Administrator-approved form template (advanced)

Prepare this form template for an administrator approval.

5. Browse to the local or network location where you wish to save your published form template to, click **Save** and then click **Next**.
6. Add some columns to be promoted to the SharePoint library and click **Next**, and then click **Publish**.

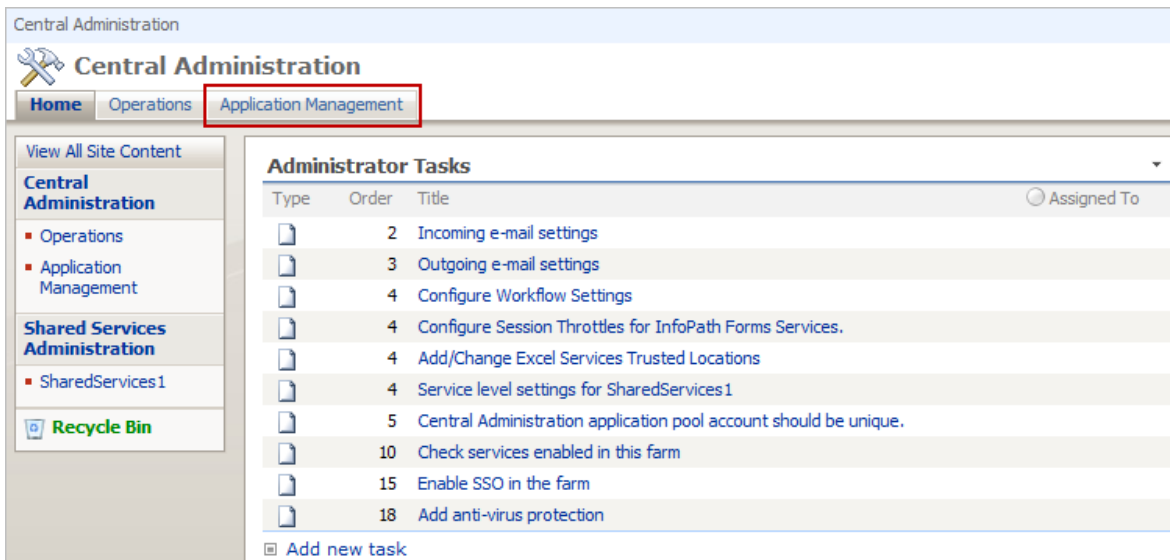
You will see a message that your form was published successfully and you are to give your SharePoint Administrator the information listed on the screen.

7. Click **Close** and close **InfoPath**.

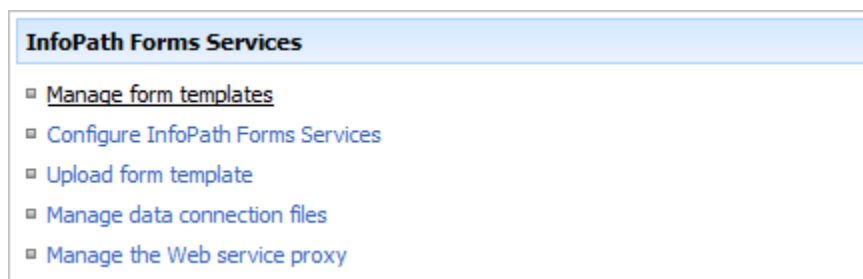
You need to be a SharePoint Administrator with access to SharePoint Central Administration to complete the next steps.



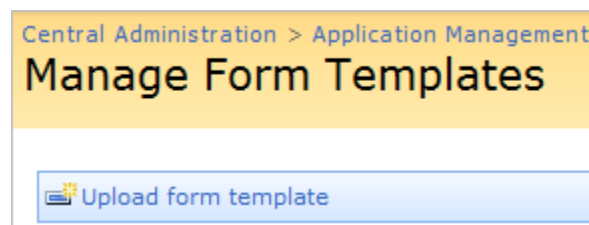
8. Navigate to the Central Administration web site.
9. Click on the **Application Management** tab.



10. Under InfoPath Forms Services, click **Manage form templates**.



11. In the 'Manage Form Templates' page, click **Upload form template**.



12. In the 'Upload Form Template' page, browse to the location where you saved your published form template and click **Save**.



Central Administration > Application Management > Manage Form Templates > Upload Form Template

Upload Form Template

Upload Form Template
Browse to the form template you intend to add to the server.

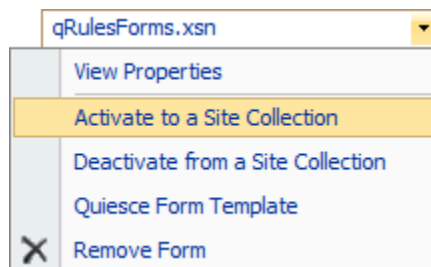
File Name:

Before uploading, you can check this form template for detailed errors, warnings, and other information.

Upgrade
Set the behavior if the server contains a previous version of this form template with the same Form ID.

☒ Upgrade the form template if it already exists
New sessions will start using the upgraded version of the form template. Forms already open will continue to use the current version of the form template.
How would you like to handle existing browser-based form filling sessions?
☒ Allow existing browser-based form filling sessions to complete using the current version of the form template.
☐ Terminate existing browser-based form filling sessions. Any data in those sessions will be lost.
If you wish to wait until all sessions of the form template have completed before upgrading, navigate to Manage Form Templates, select the form template, and choose Quiesce form.

13. Accept the other defaults and click **Upload**. You should see a message indicating a successful upload. Click **OK**.
14. In the 'Manage Form Templates' page, click on the down arrow next to the template that you just uploaded and select **Activate to a site collection**.

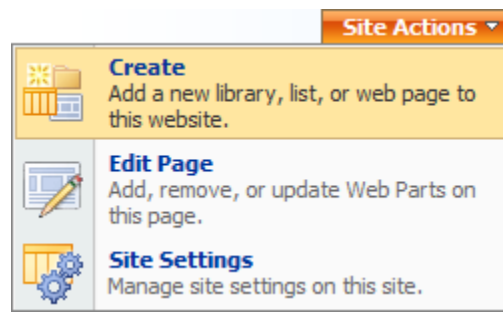


15. If the Activation Location Site Collection shows an incorrect site, click on the down arrow next to **Site collection** and select **Change Site Collection**. (Note: SharePoint almost always defaults to the *wrong* site collection the first time a form template is activated).
16. In the pop-up window, click the **Web Application** selector menu and choose **Change Web Application**.
17. Click **SharePoint - 80**, and then click the **/** to activate to the entire site collection. Click **OK** twice.
18. Close Central Administration.

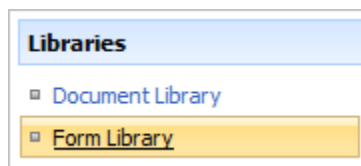


CREATE AN IPFS-ENABLED SHAREPOINT LIBRARY

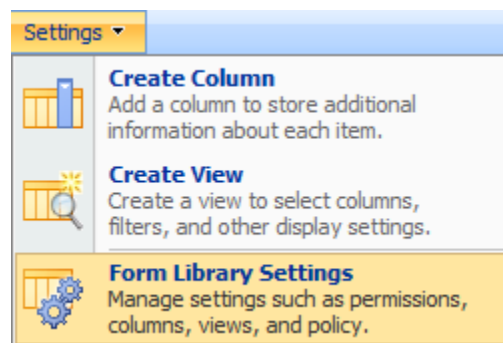
1. Navigate to the Site Collection in Internet Explorer.
2. Click **Site Actions** > **Create**



4. Click **Document Library** or **Form Library** to create new library.



5. Continue to the page to create your new library. Give it a name of your choosing. For the purposes of this document, we will name our library 'qRulesForms'. Select the options as desired, then click **Create**.
6. Go to the library created and click **Settings** > **Form Library Settings**.



7. Click **Advanced settings** under *General Settings*.



General Settings

- Title, description and navigation
- Versioning settings
- Advanced settings**
- Audience targeting settings

8. In the Advanced Settings Page, select **Yes** under **Allow management of content types**, click **Display as a Web page** to open forms in the browser, then click **OK**.

Allow management of content types?
☒ Yes ☐ No

Template URL:

Opening browser-enabled documents
☐ Open in the client application
☒ Display as a Web page

9. Back on the Library Settings Page, you should now see a section called **Content Types**. Choose the link **Add from existing site content types**.

Content Types

This document library is configured to allow multiple content types. Use content types to specify the information you want to display about an item, in addition to its policies, workflows, or other behavior. The following content types are currently available in this library:

Content Type	Visible on New Button	Default Content Type
Form	✓	✓

Add from existing site content types

Change new button order and default content type

10. Find your administrator-deployed template name in the **Available Site Content Types** list box and click **Add**, then click **OK**.
11. Back in Settings page, click **Change new button order and default content type** under Content Type section.



Content Type	Visible on New Button
Form	✓
qRulesForms	✓
▪ Add from existing site content types ▪ Change new button order and default content type	

12. Uncheck the Visible check box for Content Type 'Form' or 'Document' and click **OK**.

Now when users click the **New** button in the document library, your Administrator-Enable form will launch.



CREATE A DOCUMENT TYPE IN DBXL

1. In DBXL, click on **New Configuration**.

DATABASE ACCELERATOR
Administration Tool

Configure Licenses **New Configuration** Refresh List

Document Type Configuration

Name Descriptions Form Version Docs Edit Del Open

System Document Types

Powered by: Qdabra® Database Accelerator Copyright © 2008 Qdabra Software. All rights reserved. Qdabra and the Qdabra logo are trademarks of Autonomy Systems LLC. www.qdabra.com English

2. In the *General* tab, *Document Type* details section, enter the **Document Type Name**, making sure that it matches the same name that was hardcoded into the form. In this case, we used the name 'qRulesForms'.

Document Type Details

Name

qRulesForms

3. Click on the file attachment control in the **InfoPath Form Template** section and attach the admin-approved form template that you have published and saved locally (i.e. not the one in SharePoint).

InfoPath Form Template

XSN File

qRulesForms.xsn
Microsoft Office InfoPath Form Template
47.8 KB

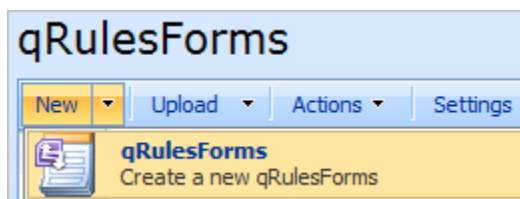
Open

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



TEST THE SOLUTION

1. Test creating and updating a new document from the browser.
 - a. In the DBXL Administration Tool (DAT), click the **Documents** tab and observe that there are no documents listed.
 - b. In your IPFS library, click on the down arrow next to **New** and select the admin-approved form template that you have published.



- c. The form will open in the browser. Fill out the fields and click on the **Submit** button. (Do not close the form.)

- d. In DAT, click **Refresh** and verify that the submitted document appears.

Documents						Delete		Delete All		Reshred		Reshred All	
Doc. ID	Ref. ID	Version #	Name	Author	Description	New							
253	11	1	Form1	Mel Balsamo	This is my first form.	Open							

- e. Back in the form in IPFS, modify the description and click **Submit**.



Close | Print View

Name: Form1

Author: Mel Balsamo

Description: I am editing my first form.

Submit

Close | Print View

- f. In DAT, click **Refresh** and verify that the document is now showing the modified value in the *Description* column. Notice that the DocID column has not changed because the form is now connected to the previously submitted document.

Documents							
Delete Delete All Reshred Reshred All							
Doc. ID	Ref. ID	Version #	Name	Author	Description	New	
253	11	2	Form1	Mel Balsamo	I am editing my first form.	Open	

2. Test opening and updating an existing document from the browser.

- a. To open an existing document, specify the DocID parameter in the URL.

You can identify a document ID by looking in DAT under the DocID column. The URL to open this document would then look like:

[http://<YourServerName>/ layouts/formserver.aspx?XsnLocation=/FormServerTemplates/<FormTemplatePublished>.xsn&OpenIn=Browser&DocID=<DocumentID>](http://<YourServerName>/layouts/formserver.aspx?XsnLocation=/FormServerTemplates/<FormTemplatePublished>.xsn&OpenIn=Browser&DocID=<DocumentID>).

- b. In a new web browser window, navigate to this URL, replacing <YourServerName> with yours, changing <FormTemplatePublished> with the name of your admin-approved template which in this case is 'qRulesForms', and setting the <DocumentID> parameter to the one for your document. In this case, the document we have submitted in Step 1 has a DocID of 253. Your URL should then look like:
<http://mySharePointSite/ layouts/formserver.aspx?XsnLocation=/FormServerTemplates/qRulesForms.xsn&OpenIn=Browser&DocID=253>.
 - c. Verify that your form opens in the browser without errors.
 - d. Modify the description and click **Submit**.
 - e. In DAT, click **Refresh** and verify that the document shows the changed value in the *Description* column.

3. Test submitting documents via DBXL and opening them in the browser.



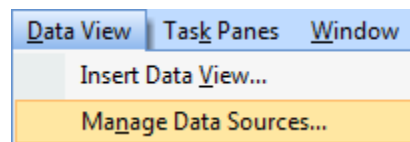
- a. In DAT, go to the *General* tab and click on the **Download** icon  to open the form in InfoPath.
- b. Fill out the form fields and click **Submit**.
- c. Go to the **Documents** tab and verify that the submitted document appears.
- d. Open the document in the browser by passing in the DocID parameter in the URL. Use the same URL as before, replacing DocID with that of the newly-submitted document.
- e. Verify that your form opens in the browser without errors.
- f. Continue submitting new documents to have at least 3 in preparation for the next task.



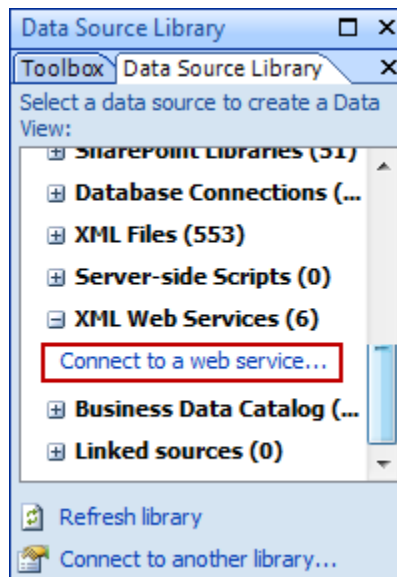
CREATE A DATA VIEW TO ACCESS YOUR DBXL FORMS

To make the process of opening and submitting documents easier, it will be best to add a data view to the SharePoint site that would list all the documents in DBXL via the *GetListItems* method. This requires the installation of SharePoint Designer.

1. In the SharePoint Designer menu, go to **File > Open Site**.
2. Enter the URL for your SharePoint site and click **Open**.
3. In the 'Folder List' pane, navigate to the SharePoint library that you created in the previous steps, which in this case is '*qRulesForms*'.
4. Expand **qRulesForms > Forms**, and double-click on **AllItems.aspx** to open it in the designer.
5. In the menu, go to **Data View > Manage Data Sources**.



6. Under **XML Web Services**, click **Connect to a web service**.



7. Enter the **Service Description Location** (URL) to the DBXLDocumentService, usually located at <http://<servername>/QdabraWebService/DbxlDocumentService.aspx>, and click **Connect**.



General Source Login

Service description location:
 Browse...

Example: http://example.com?WSDL

Establish a connection with the specified XML Web Service.

8. In the Operation dropdown, select **GetListItems**.

Operation: GetDocumentHistory
GetListItems
GetMyListItems
QueryDocuments
QueryDocumentsHistory
Search
SearchEx

Parameters

Name	* docId

You should be able to see the following:

Data Source Properties

General Source Login

Service description location:
 Browse...

Example: http://example.com?WSDL

Select which data command to configure: Select

Select Connection Info

Service: DbxlDocumentService

Name:

Port: DbxlDocumentServiceSoap

Address: http://dbxl23demo/qdabrawebservice/dbxldocumentservice.asmx

Operation: GetListItems

Parameters (* required):

Name	Value	Element	Type	Runtim...
listName			s:string	☐
viewName			s:string	☐
query				☐
viewFields				☐
rowLimit			s:string	☐
queryOptions				☐



9. Select **listName** and click **Modify**, then enter the name of the Document Type you wish to query. Remember that we have already created a document type and have named it '*qRulesForms*'.

A dialog box titled "Parameter" with a red border. It contains the following fields and controls:

- Name:** A text box containing "listName".
- Value:** A dropdown menu showing "qRulesForms".
- Element:** "1 of 1"
- Maximum Allowed:** "1"
- ☐ The value of this parameter can be set via a Web Part connection
- Buttons:** "OK" and "Cancel".

10. Click **OK**.

GetListItems limits the results returned in your SharePoint list to the first 100 documents. In order to increase this, set the rowLimit to more than 100 rows.

A screenshot of the "Parameter" dialog box overlaid on a larger window. The dialog box is titled "Parameter" with a blue border. It contains the following fields and controls:

- Name:** A text box containing "rowLimit".
- Value:** A dropdown menu showing "1000".
- Element:** "1 of 1"
- Maximum Allowed:** "1"
- ☐ The value of this parameter can be set via a Web Part connection
- Buttons:** "OK" and "Cancel".

A red arrow points from the "rowLimit" entry in the "Parameters (* required):" list of the background window to the "Name" field of the dialog box.

11. Go to the **General** tab and enter a name for the connection.

A screenshot of the "General" tab of a configuration window. It has three tabs: "General", "Source", and "Login". The "General" tab is selected. It contains a text box labeled "Name:" with the value "Get qRulesForms Items".

12. Go to the **Login** tab and select **Use Windows authentication**, then click **OK**.



General Source Login

Login

☐ Don't attempt to authenticate

☐ Save this username and password in the data connection

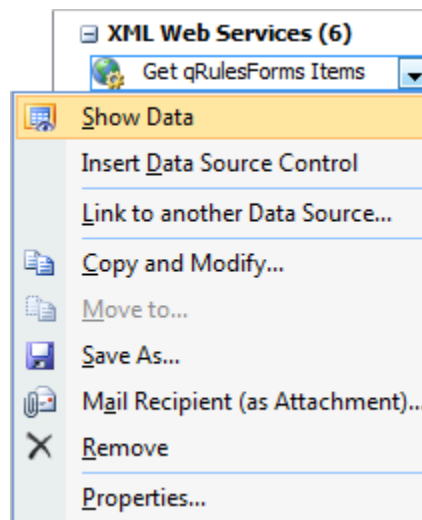
User name:

Password:

☒ Use Windows authentication

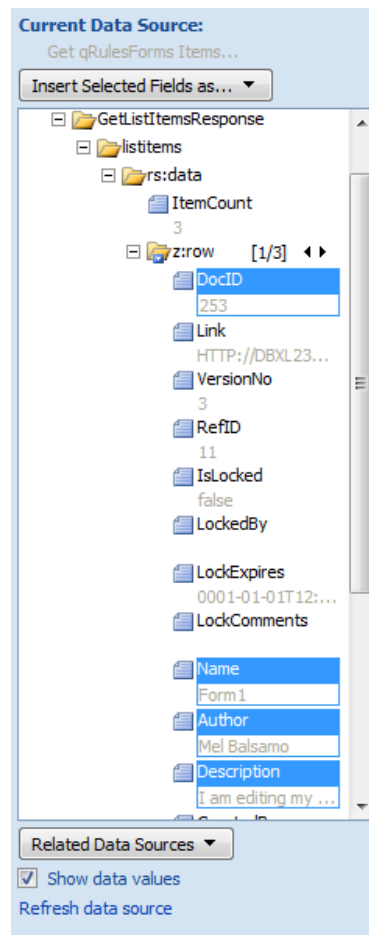
☐ Use Single Sign-On authentication

13. The new web service connection will show up under **XML Web Services** in the Data Source Library. Click the down arrow for the connection you just created and click on **Show Data**.

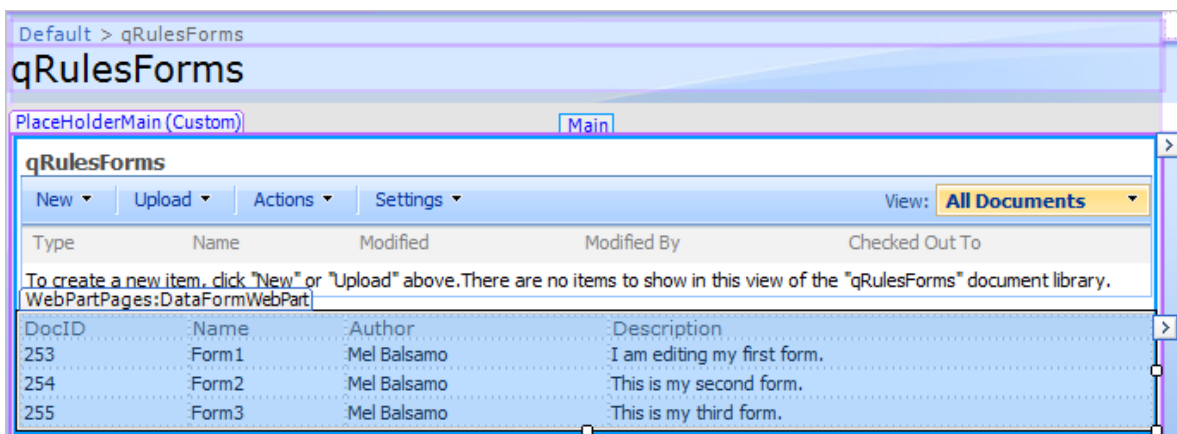


The Data Source Details tab will show the columns retrieved from the Document Type. For the purposes of this document, we will do the following:

14. Hold down the **Ctrl** key and use your mouse to click on the **DocID**, **Name**, **Author** and **Description**.



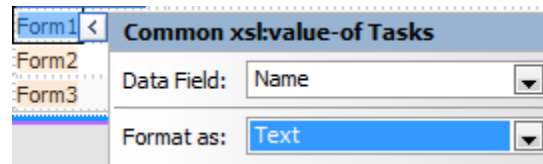
15. Drag and drop these columns onto a location on the page where you wish to display this data. Your canvas should now look like this:



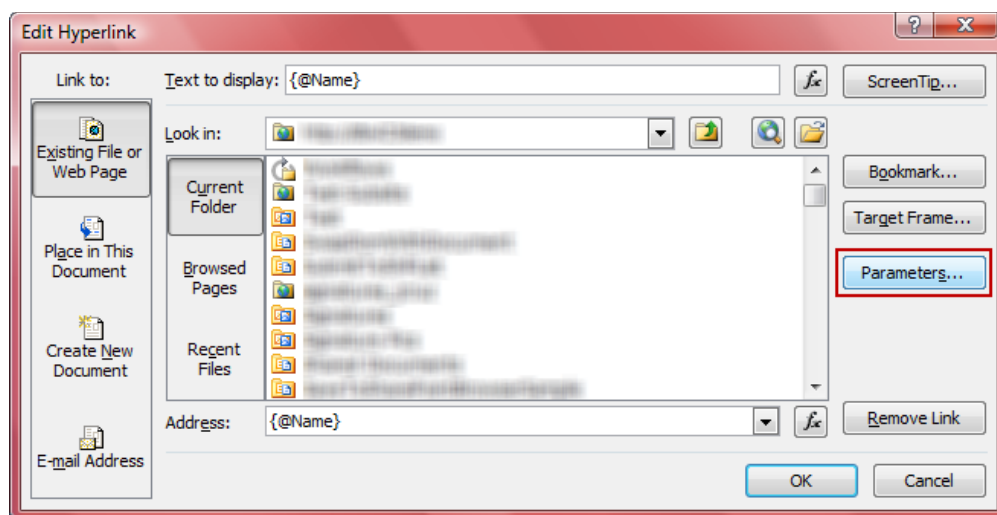
16. Add a hyperlink to open the documents in the browser:



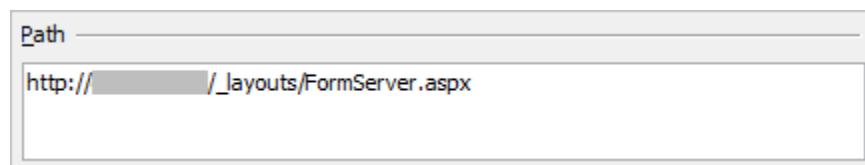
- a. Click on  next to any entry in the column you wish to display form links in. A menu will appear:



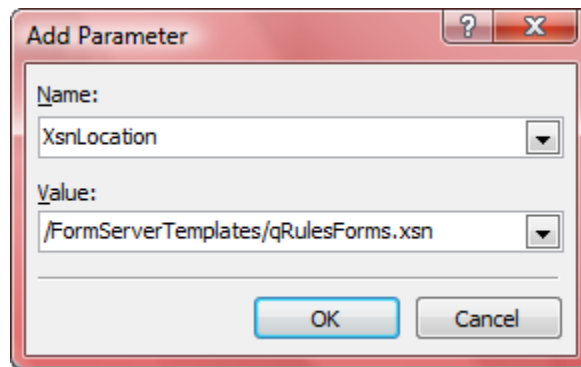
- b. Select **Hyperlink** in the 'Format as' dropdown. Click **Yes** when prompted.
- c. In the 'Edit Hyperlink' dialog, click on the **Parameters** button.



- d. Remove the text in the Path field and enter http://<YourServerName>/_layouts/FormServer.aspx, replacing <YourServerName> with yours.



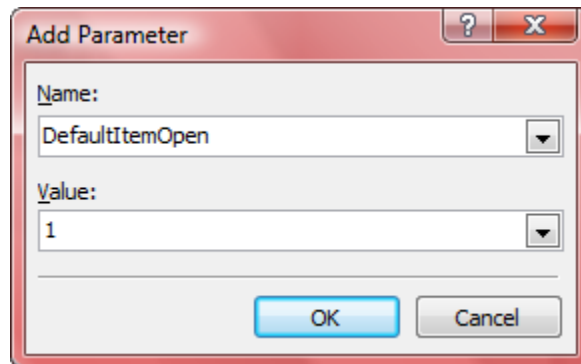
- e. Click on the **Add** button to start adding your query string parameters:
- In the **Name** field, enter **XsnLocation**.
 - In the **Value** field, enter the relative path to the published form template, e.g. </FormServerTemplates/qRulesForms.xsn>, then click **OK**.



The 'Add Parameter' dialog box has a title bar with a question mark and a close button. It contains two labeled text boxes: 'Name:' and 'Value:'. The 'Name' box contains the text 'XsnLocation' and has a dropdown arrow on its right. The 'Value' box contains the text '/FormServerTemplates/qRulesForms.xsn' and also has a dropdown arrow on its right. At the bottom of the dialog are two buttons: 'OK' and 'Cancel'.

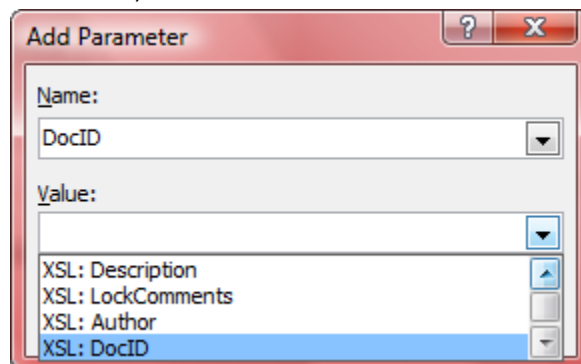
Continue to add the following parameters:

iii. DefaultItemOpen = 1



The 'Add Parameter' dialog box is shown with 'Name:' set to 'DefaultItemOpen' and 'Value:' set to '1'. Both fields have dropdown arrows on their right. The 'OK' and 'Cancel' buttons are at the bottom.

iv. DocID = {@DocID} (Select **XSL: DocID** from the Value dropdown and remove the @ sign in the **Name**)



The 'Add Parameter' dialog box is shown with 'Name:' set to 'DocID'. The 'Value:' dropdown menu is open, showing a list of XSL paths: 'XSL: Description', 'XSL: LockComments', 'XSL: Author', and 'XSL: DocID'. The 'XSL: DocID' option is currently selected and highlighted in blue. The 'OK' and 'Cancel' buttons are at the bottom.

v. OpenIn = Browser



The 'Add Parameter' dialog box has a title bar with a question mark and a close button. It contains two dropdown menus: 'Name' with 'OpenIn' selected and 'Value' with 'Browser' selected. At the bottom are 'OK' and 'Cancel' buttons.

- vi. Source = (relative path where the user is taken when they close the document), e.g. /qRulesForms/Forms/AllItems.aspx.

The 'Modify Parameter' dialog box has a title bar with a question mark and a close button. It contains two dropdown menus: 'Name' with 'Source' selected and 'Value' with '/qRulesForms/Forms/AllItems.aspx' selected. At the bottom are 'OK' and 'Cancel' buttons.

You should now have the following Query strings:

Query string	
Name	Value
XsnLocation	/FormServerTemplates/qRulesForms.xsn
DefaultItemOpen	1
DocID	{@DocID}
OpenIn	Browser
Source	/qRulesForms/Forms/AllItems.aspx

- f. Click **OK**.
- g. In the 'Edit Hyperlink' dialog, enter {@Name} as the text to display.

A text input field with the label 'Text to display:' and the value '{@Name}'.

- h. Verify that the link generated in the Address field is correct. The address should be in the format:
- [http://<YourServerName>/ layouts/formserver.aspx?XsnLocation=/FormServerTemplates/](http://<YourServerName>/layouts/formserver.aspx?XsnLocation=/FormServerTemplates/)



[<FormTemplatePublished>.xsn&DefaultItemOpen=1&DocID={@DocID}&OpenIn=Browser&Source=/<YourSharePointLibrary>/Forms/AllItems.aspx](#)

- i. Click **OK** to close the **Edit Hyperlink** dialog.
17. In the menu, click **File > Save** to save the changes to your SharePoint library.
18. In Internet Explorer, navigate to your SharePoint library. You will now see your new data view added onto your SharePoint library.

qRulesForms				
New	Upload	Actions	Settings	View: All Documents
Type	Name	Modified	Modified By	Checked Out To
To create a new item, click "New" or "Upload" above. There are no items to show in this view of the "qRulesForms" document library.				
DocID	Name	Author	Description	
253	Form1	Mel Balsamo	I am editing my first form.	
254	Form2	Mel Balsamo	This is my second form.	
255	Form3	Mel Balsamo	This is my third form.	

Test opening new forms in the browser by clicking on the **New** link, and submitting using the **Submit** button. Your form will be submitted to DBXL and get displayed on your SharePoint data view. Test opening existing forms and verify that they open in the browser, and that you can modify and submit any changes.