



QDABRA DATABASE ACCELERATOR V2.5

Create a SharePoint 2010 Data View that uses GetListItems to access DBXL forms

SharePoint Designer allows you to receive data from DBXL's web methods, and one method is particularly interesting when creating a SharePoint Data View. The DBXLDocumentService web service contains GetListItems, which retrieves promoted properties in the form and displays them as columns in SharePoint. GetListItems has an added bonus: it will respect DBXL permissions.

This document contains two approaches:

- The first approach will discuss forms that will open in the client (InfoPath).
- The second approach will deal with browser forms (IPFS) in SharePoint 2010. This second approach requires the use of Qdabra qRules.

To successfully complete this tutorial, you MUST have the following:

- DBXL v2.4 or later
- SharePoint 2010
- SharePoint Designer 2010
- qRules v2.4 or later (for browser forms only)

CONTENTS

Client Forms.....	2
Browser forms	8

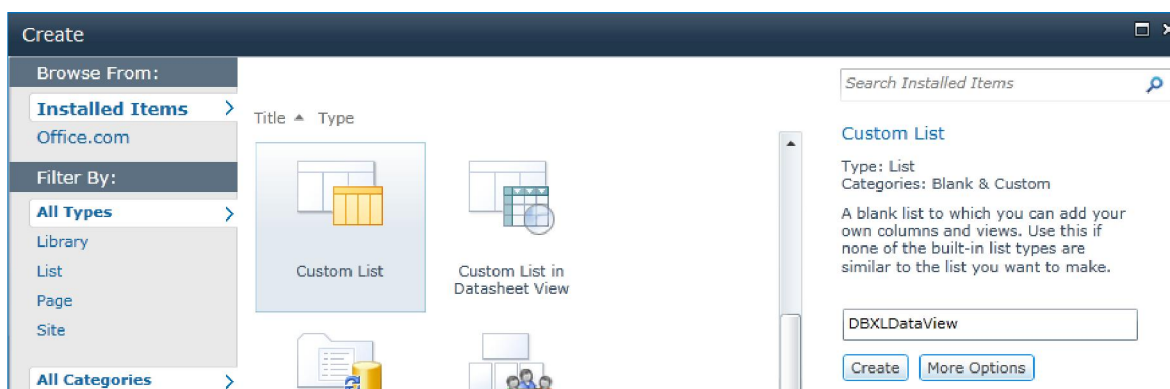


CLIENT FORMS

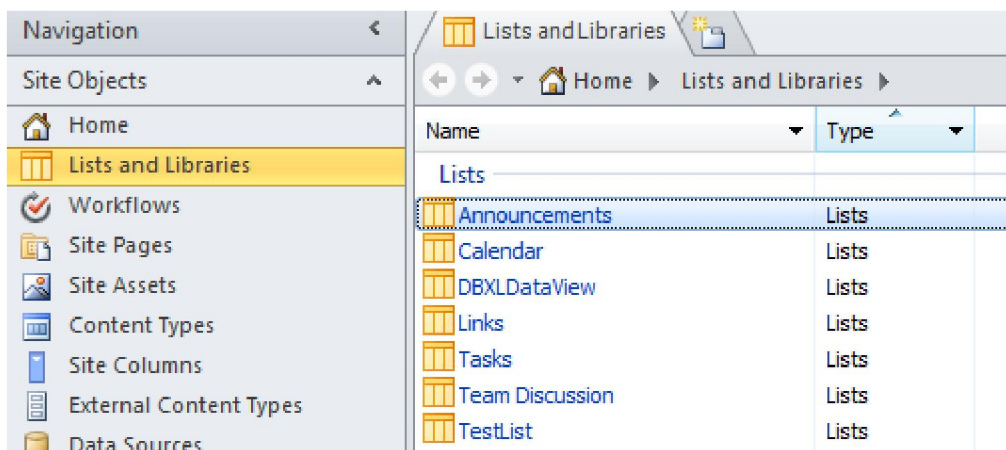
This section assumes that you already have a DBXL Document Type with at least two documents submitted. For information on this, please see the **Getting Started** document included with DBXL. Besides the details covered in the Getting Started document, there are no additional requirements when setting up the XSN or the Document Type. (Additional steps are only necessary for browser-compatible forms, with details later in this document.)

CREATE A NEW SHAREPOINT LIST

1. Navigate to the site where you wish to add this data view. Click on **All Site Content** and then click on **Create**.
2. Select **Custom List**, enter a name for the list and click **Create**.



3. Launch SharePoint Designer 2010.
4. Go to **File > Open Site**. Enter the URL to your site and click **Open**. The site will load and might take a few seconds depending on the size of your site.
5. In the Navigation frame, select **List and Libraries**. This will load the list of available lists and libraries.

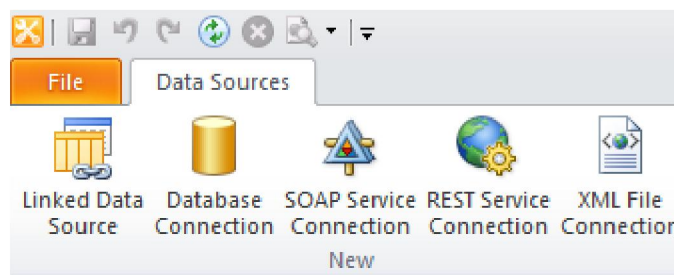




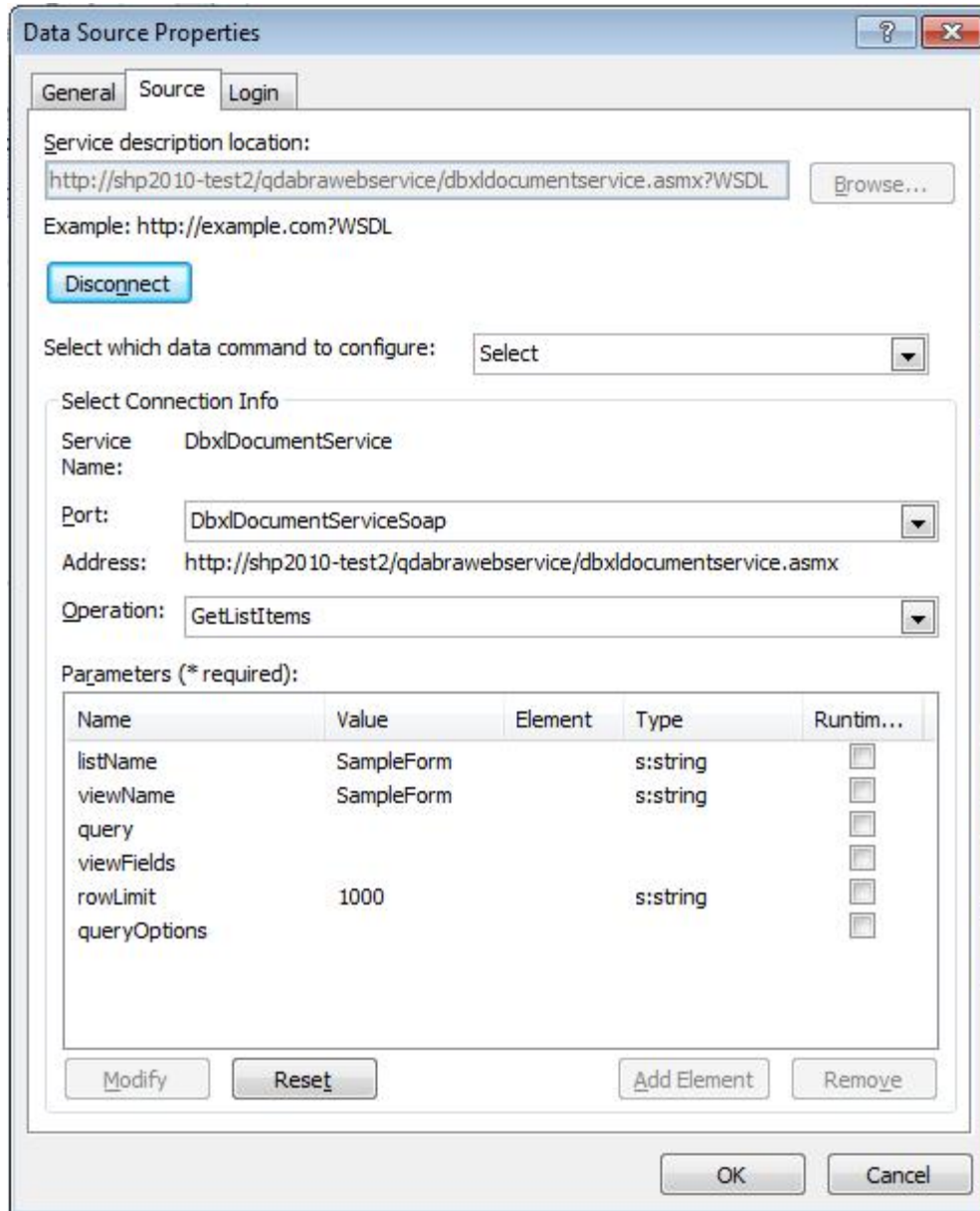
6. Click on the list we previously created to open it.
7. Under **Views**, click on **All Items** to open this default view in the editor.

Name	Type	Default
All Items	HTML	Yes

8. In the **Navigation** pane, select **Data Sources**.
9. In the ribbon, under **Data Sources**, select **SOAP Service Connection**.



10. In the **General** tab, enter a name for your data source, such as **DBXLGetListItems**.
11. In the **Source** tab, enter the web service URL: ***http://<your-server-name>/qdabrawebservice/dbxldocumentservice.asmx***
12. Click on **Connect Now**.
13. From the **Operation** dropdown select **GetListItems**.
14. Double click on the **listName** and **viewName** parameters, and enter the name of the DBXL Document Type you wish to query. Click **OK**.
15. **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.

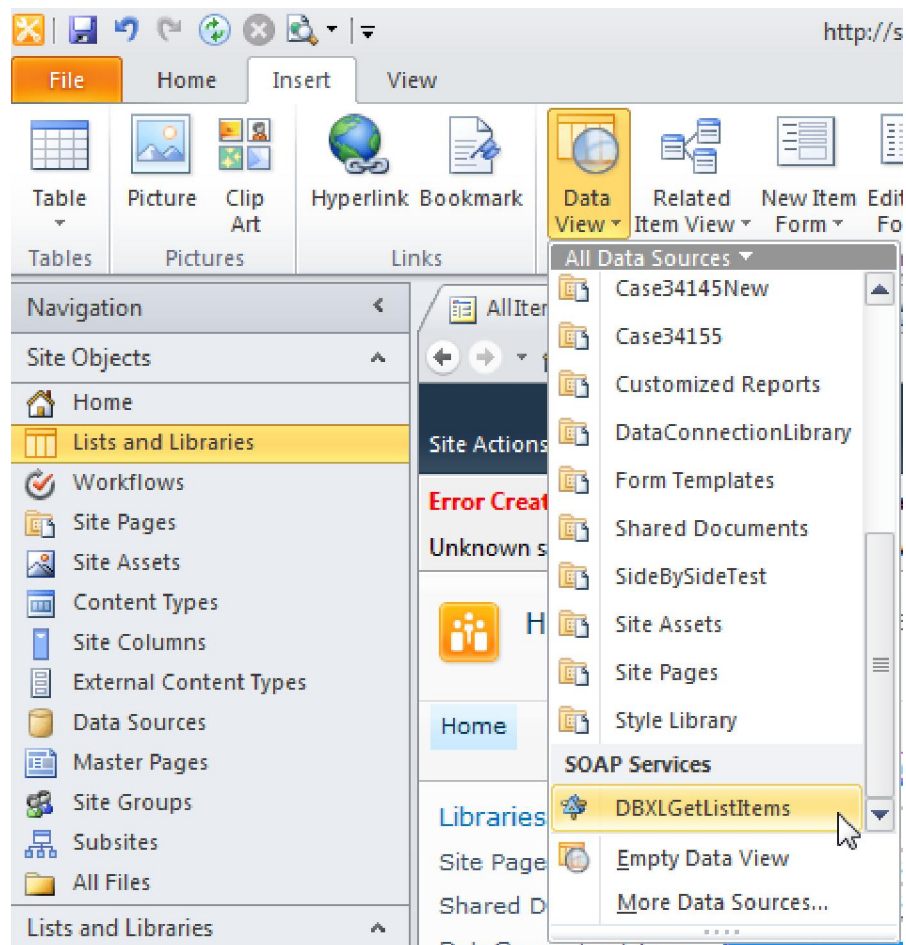


The 'Data Source Properties' dialog box is shown with the 'Source' tab selected. It contains the following fields and controls:

- Service description location:** A text box containing 'http://shp2010-test2/qdabrawebservice/dbxldocumentservice.asmx?WSDL' and a 'Browse...' button.
- Example:** A text box containing 'http://example.com?WSDL'.
- Disconnect:** A button.
- Select which data command to configure:** A dropdown menu with 'Select' chosen.
- Select Connection Info:** A section containing:
 - Service Name:** 'DbxlDocumentService'.
 - Port:** A dropdown menu with 'DbxlDocumentServiceSoap' selected.
 - Address:** 'http://shp2010-test2/qdabrawebservice/dbxldocumentservice.asmx'.
 - Operation:** A dropdown menu with 'GetListItem' selected.
- Parameters (* required):** A table with 5 columns: Name, Value, Element, Type, and Runtime... (with checkboxes).

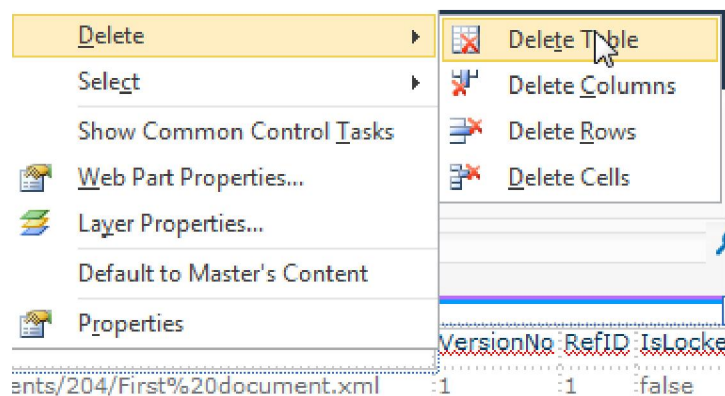
Name	Value	Element	Type	Runtime...
listName	SampleForm		s:string	☐
viewName	SampleForm		s:string	☐
query				☐
viewFields				☐
rowLimit	1000		s:string	☐
queryOptions				☐
- Buttons:** 'Modify', 'Reset', 'Add Element', 'Remove', 'OK', and 'Cancel'.

16. Return to the AllItems.aspx edit view. SharePoint adds some generic layout tables and controls to this view, which you can remove or modify as desired.
17. Place the cursor on the location where you wish to enter the data view.
18. From the ribbon option **Insert**, select **Data View** and then select the **GetListItem** data source we created above. If it doesn't display among the options, you might need to select **More Data Sources**. If the **Data View** menu item is not enabled, your cursor is not in a location where the data can be displayed properly.



You now have a basic Data View, but it might not display as desired. The next steps will show you how to modify it. You'll notice that SharePoint designer selected a few columns to display. However, we might want to display a different set of columns.

19. Right click anywhere on the table displaying the data view, and select **Delete > Delete Table**.

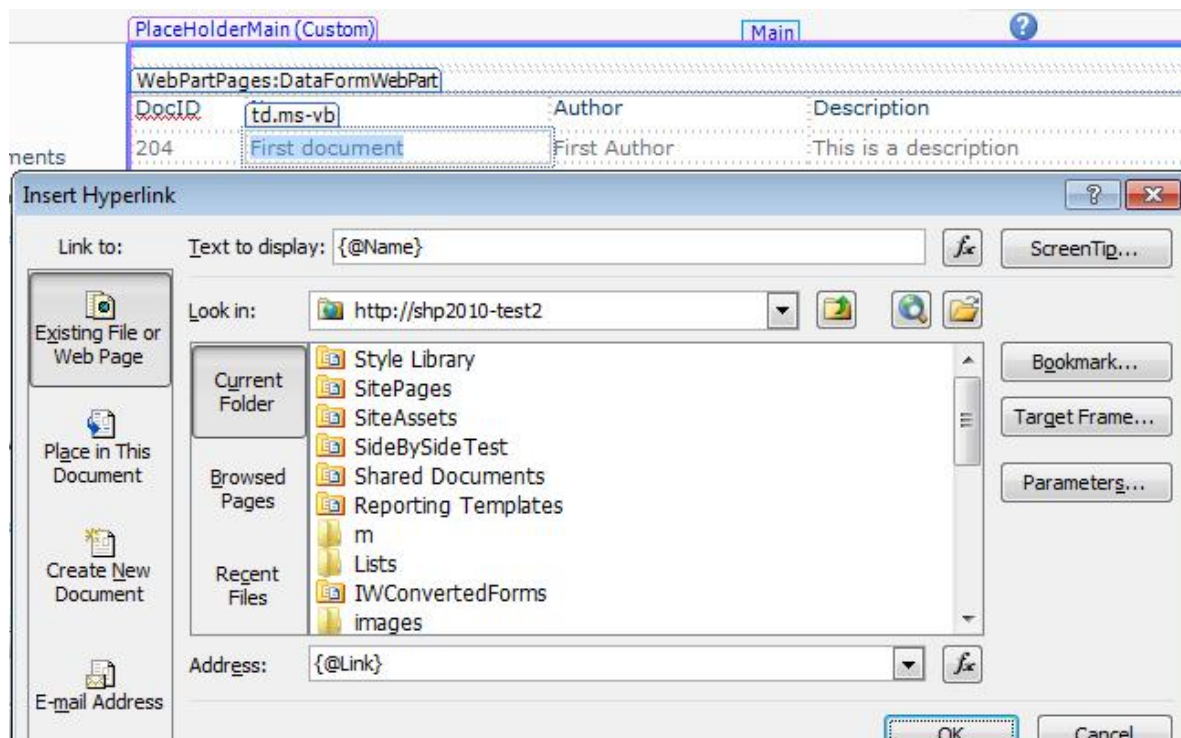




20. You'll see that the Data Source Details are displayed in the right hand frame. Select the columns you wish to display by using Control+click. Once you have selected the desired columns, drag and drop into the location in the view where you wish to display the data view table.

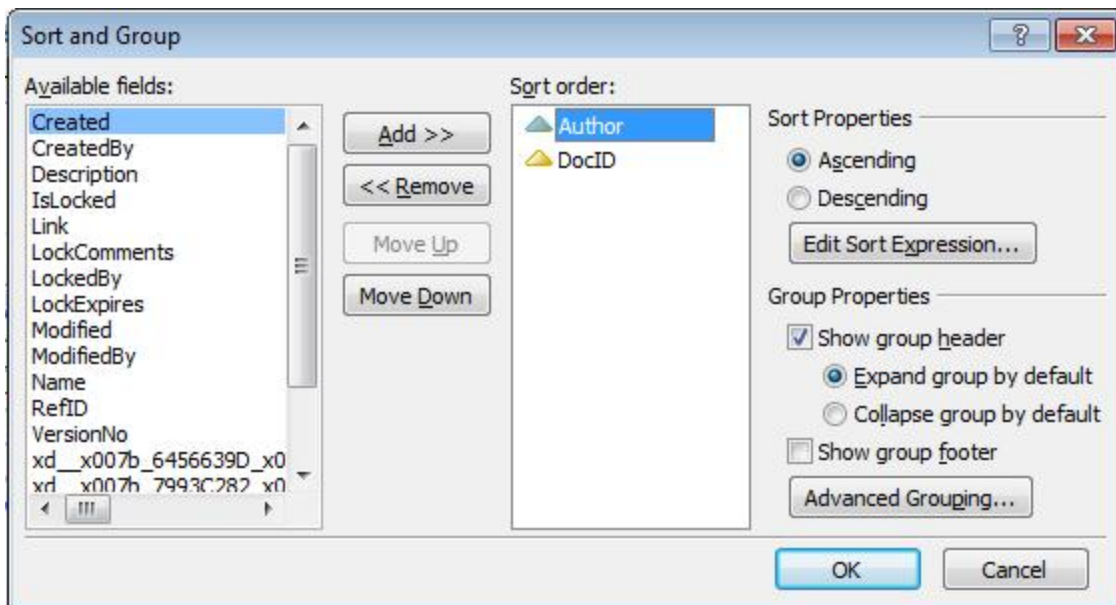
You can also modify the available columns by going to **Options > Add/Remove columns** in the ribbon.

21. Add a hyperlink to the DBXL location of the xml form by selecting the column item you wish to use as a link, and then select **Insert > Hyperlink** from the ribbon.
22. In the dialog, use the **fx** button to add the **{@Link}** property as the **Address**. Click **OK**.

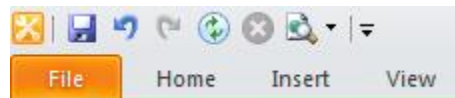


DATA VIEW SORTING AND GROUPING (OPTIONAL)

23. Place the cursor on the table and select **Options > Sort and Group**.
24. Select **DocID** and click **Add >>**. Repeat for **Author**.
25. Click **Author** in **Sort order**, click **Move Up**, and then click **Show group header**. Click **OK**.



26. Click the **Save** icon to save the changes to your SharePoint List.



27. In Internet Explorer, open the URL to your SharePoint List.

Should your document type contain a large amount of documents, sorting and grouping may not be enough. You can use paging to improve performance. You can add paging by going to **Options > Paging**.



BROWSER FORMS

This section will guide you through the usage of Qdabra's qRules in order to create a SharePoint 2010 data view which will display your DBXL documents.

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 DATA CONNECTIONS

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.

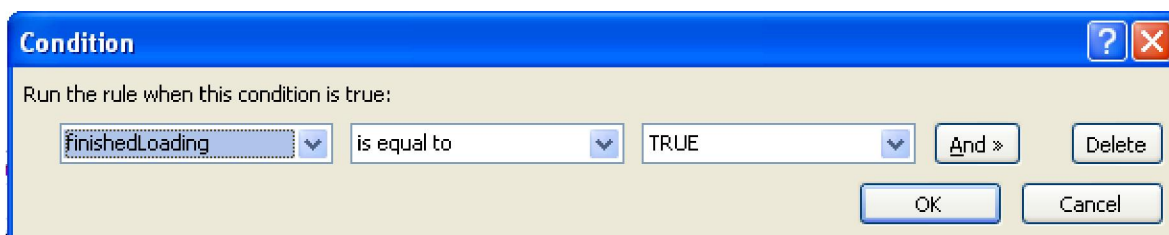
1. In the Data source task pane, switch to the **QdabraRules (Secondary)** data source.



2. Right-click on the *finishedLoading* node and select **Rules**.
3. Click **Add** to add a rule that will call the qRules *GetInputParameter* command and will use 'DocID' as the key.



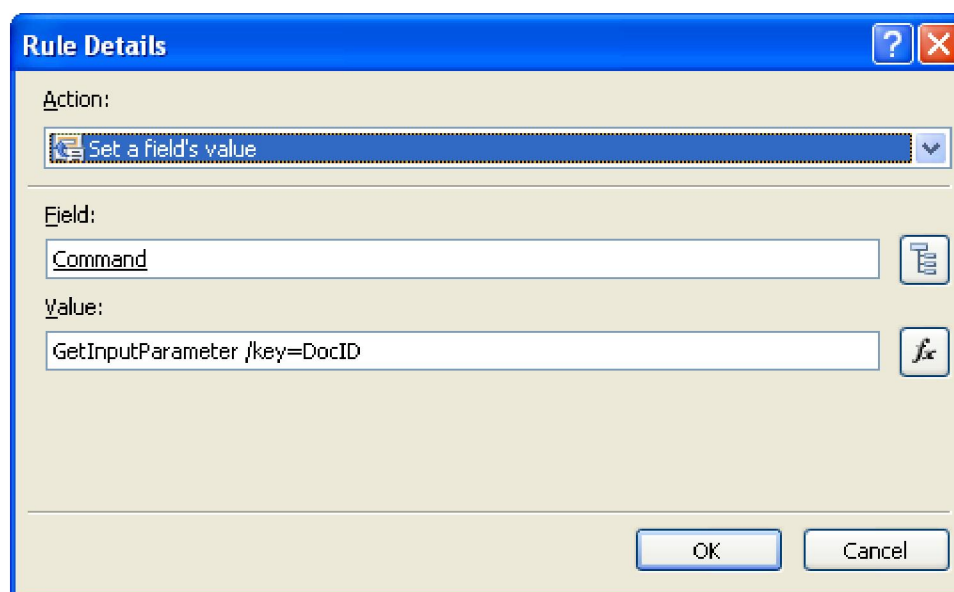
4. Give your rule a name of your choosing.
5. Add the condition:



The 'Condition' dialog box has a blue title bar with a question mark and a close button. The main area is light beige and contains the text 'Run the rule when this condition is true:'. Below this text is a row of controls: a dropdown menu with 'finishedLoading' selected, followed by 'is equal to' in a dropdown, then 'TRUE' in a text box, and a dropdown arrow. To the right of these are 'And >' and 'Delete' buttons. At the bottom right are 'OK' and 'Cancel' buttons.

Where:

- finishedLoading – QdabraRules data source
 - To enter TRUE as value, select **Use a formula** and enter `string(true())`
6. Click **Add** for action, and select **Set a field's Value**.
 7. Click on the schema icon next to **Field** and select the **Command** node from the **QdabraRules (Secondary)** data source.
 8. 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:

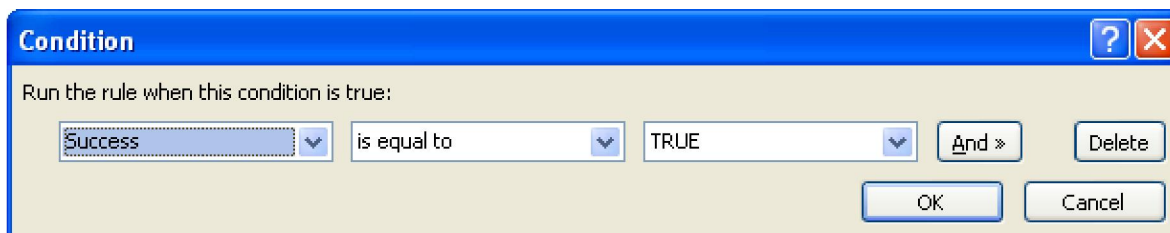


The 'Rule Details' dialog box has a blue title bar with a question mark and a close button. The main area is light beige and contains three sections: 'Action:' with a dropdown menu showing 'Set a field's value'; 'Field:' with a text box containing 'Command' and a schema icon button; and 'Value:' with a text box containing 'GetInputParameter /key=DocID' and a formula icon button. At the bottom right are 'OK' and 'Cancel' buttons.

9. Click **OK**.
10. Click **Add** to add another rule that will call the `qRules SwapDomWithDocument` command and will use the resulting 'DocID' returned by the `GetDocument` data connection.



11. Add the condition:



The 'Condition' dialog box has a blue title bar with a question mark and a close button. The main area is light beige and contains the text 'Run the rule when this condition is true:'. Below this, there is a sequence of three dropdown menus: the first contains 'Success', the second contains 'is equal to', and the third contains 'TRUE'. To the right of these is an 'And »' button and a 'Delete' button. At the bottom right are 'OK' and 'Cancel' buttons.

Where:

- Success – QdabraRules data source
- To enter TRUE as value, select **Use a formula** and enter `string(true())`

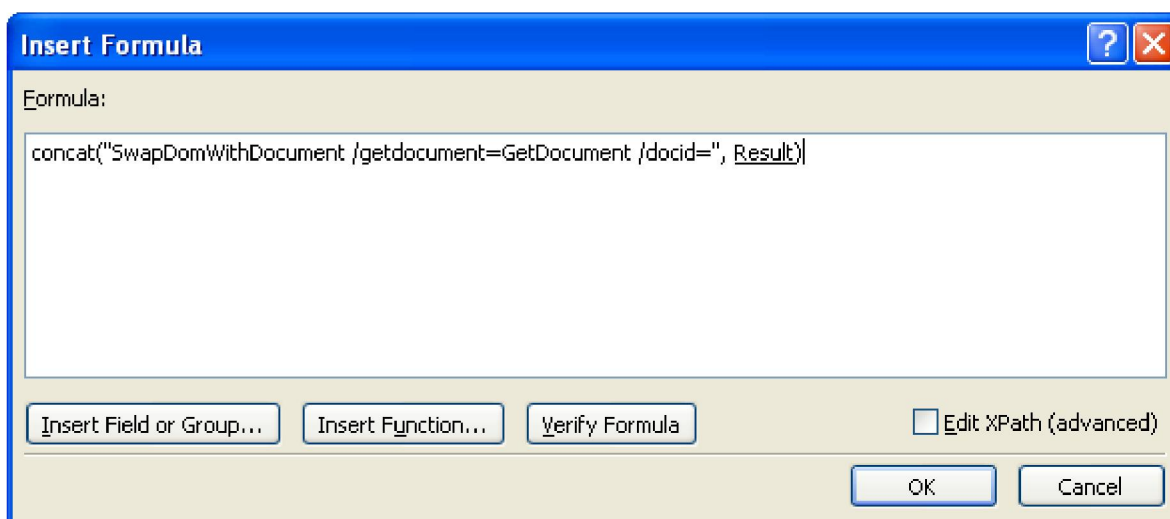
12. Click **Add** and select the action **Set a field's value**.

13. For the **Field**, select the **Command** node from the **QdabraRules (Secondary)** data source.

14. For the **Value**, click on the **fx** button and enter the formula:

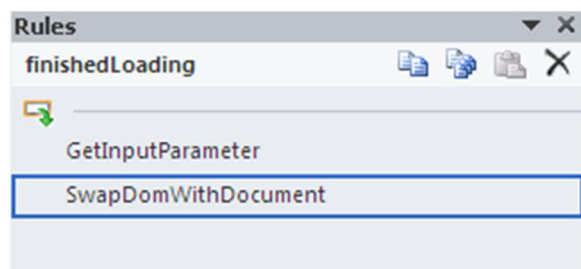
concat("SwapDomWithDocument /getdocument=GetDocument /docid=", Result)

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:



The 'Insert Formula' dialog box has a blue title bar with a question mark and a close button. The main area is light beige and contains the text 'Formula:'. Below this is a large text area containing the formula: `concat("SwapDomWithDocument /getdocument=GetDocument /docid=", Result)`. At the bottom, there are three buttons: 'Insert Field or Group...', 'Insert Function...', and 'Verify Formula'. To the right of these is a checkbox labeled 'Edit XPath (advanced)'. At the bottom right are 'OK' and 'Cancel' buttons.

15. Click **OK** to close all the dialog boxes. We now have two rules in the QdabraRules *finishedLoading* node:



16. Close the Rules pane for the finishedLoading field.

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* task pane, 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 click **New > Action**.



7. Give your rule a name of your choosing and click **Add > Set a field's value**.
8. For the **Field**, select the **Command** node from the **QdabraRules (Secondary)** data source.
9. 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 XPath of the fields.

Your formula should look like this:

Insert Formula

Formula:

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

Insert Field or Group... Insert Function... Verify Formula

☐ Edit XPath (advanced)

OK Cancel

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



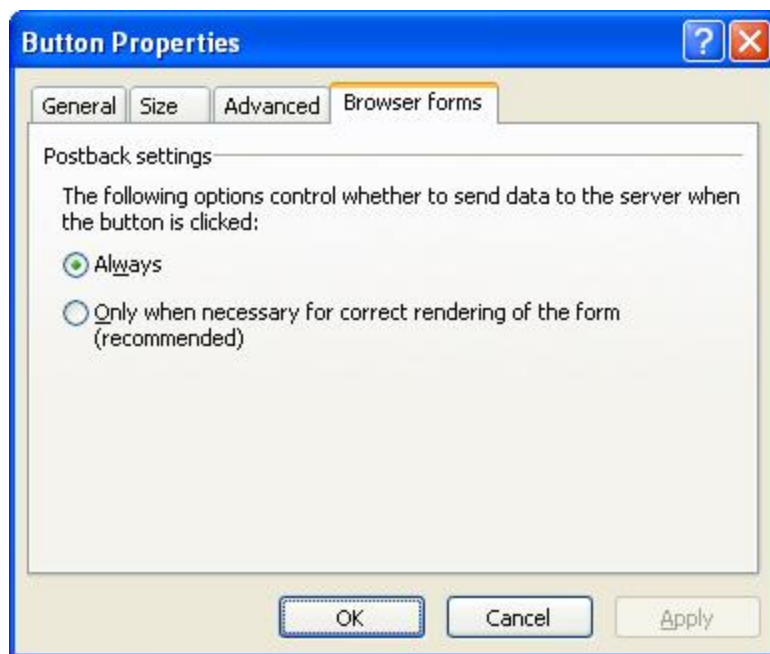
Name	
Author	
Description	
Submit	

BROWSER-ENABLE YOUR FORM TEMPLATE

1. Go to **Form Options > Compatibility**, and select “Web Browser Form” as form type. Enter the link to the SharePoint InfoPath Forms Services. Click **OK**.

The screenshot shows the 'Form Options' dialog box with the 'Compatibility' tab selected. On the left, a list of categories includes 'Web Browser', 'Filler Features', 'Offline', 'E-Mail Attachments', 'Property Promotion', 'Digital Signatures', 'Security and Trust', 'Preview', 'Programming', 'Compatibility' (highlighted), 'Versioning', and 'Advanced'. The main area shows 'Form type' set to 'Web Browser Form'. Under 'Server validation', there is a text box with the URL 'https://shp.formsboard.com/_vti_bin/FormsServices.asmx' and an example 'http://www.contoso.com/'. A checkbox labeled 'Allow code to use features that work only when filling forms with InfoPath Filler' is present and unchecked.

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, click on Control Properties in the menu and go to the **Browser forms** tab, then select **Always**.

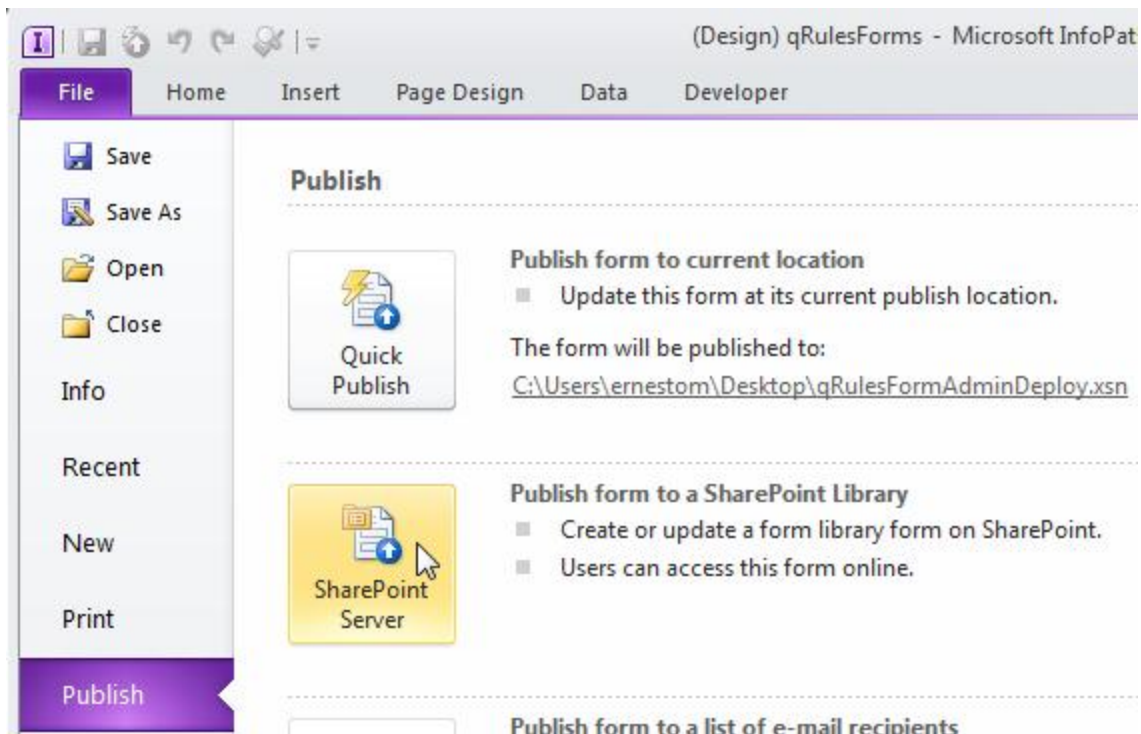


3. Save your form template to your network or local hard drive. For the purposes of this document, we will name our form template '*qRulesForms*'.
4. When you save your form, make sure to select "InfoPath Web Browser Form Template".

Save as type: InfoPath Web Browser Form Template

ADMIN-DEPLOY THE FORM

1. With the InfoPath form open in design mode, go to **File > Publish > SharePoint Server**.



2. If you are asked to save the form template, save the XSN to your local drive.
3. In the Publishing Wizard, enter the URL to the SharePoint 2010 site and click **Next**.
4. Select **Administrator-approved form template** and make sure to check “**Enable this form to be filled out by using a browser**”. Click **Next**.
5. Specify a location for the XSN and click **Next** twice.
6. Click **Publish** to finish the wizard.
7. Using Internet Explorer, open the SharePoint 2010 Central Admin site.
8. Go to **General Application Settings > Manage Form Templates > Upload Form Template**.
9. Browse to select the form template.



Upload Form Template Browse to the form template you intend to add to the server.	File Name: C:\Users\lernerstom\Desktop\qRulesForms.xsn Browse... Before uploading, you can check this form template for detailed errors, warnings, and other information. Verify
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.

10. Click **Upload**, and then click **OK** in the confirmation screen.
11. Find the template in the list, expand the menu and select **Activate to a Site Collection**.



12. Make sure that the correct **Web Application** is selected. If not, select the appropriate **Web Application**. Then make sure to select the correct **Site Collection**. Click **OK**.

CREATE A DBXL DOCUMENT TYPE

Open DAT and create a DBXL Document Type using the same XSN from step 9 above.

Remember, from page 13, that the name of the Document Type must match the qRules command you configured in the form.

For details on DAT and creating a Document Type, please refer to the Getting Started document.

You do not need to add a SharePoint mapping, but you may add Database mapping if desired.

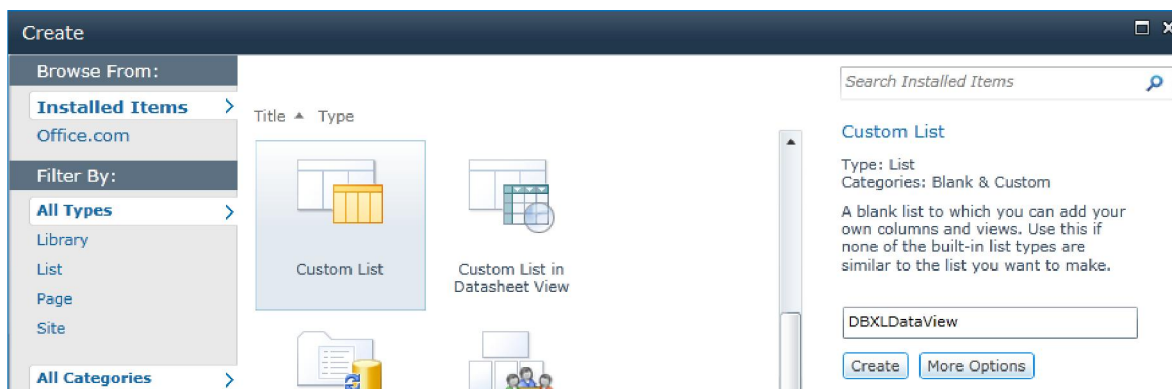
You should submit at least two documents to this new Document Type. These are required in order to complete the following section.



CREATE A SHAREPOINT 2010 DATA VIEW

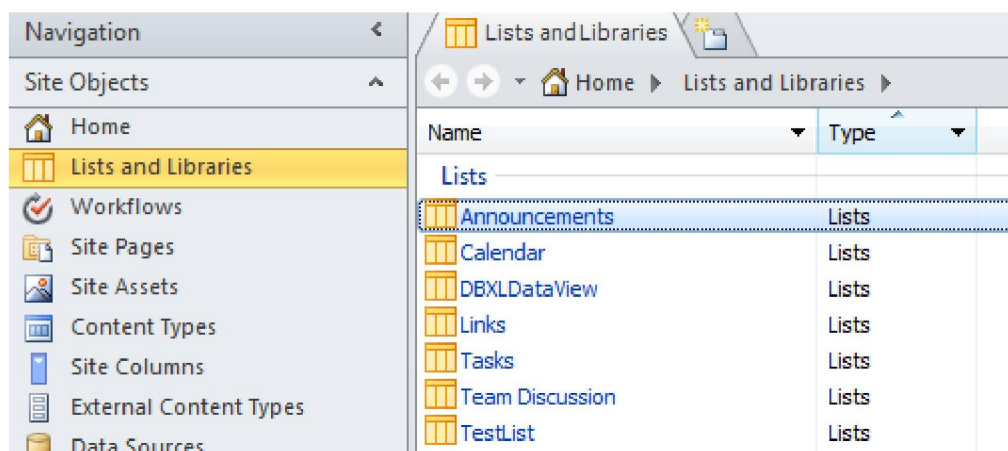
A. Create a new SharePoint List.

1. Navigate to the site where you wish to add this data view. Click on **All Site Content** and then click on **Create**.
2. Select **Custom List**, enter a name for the list and click **Create**.



B. Modify the list using SharePoint Designer.

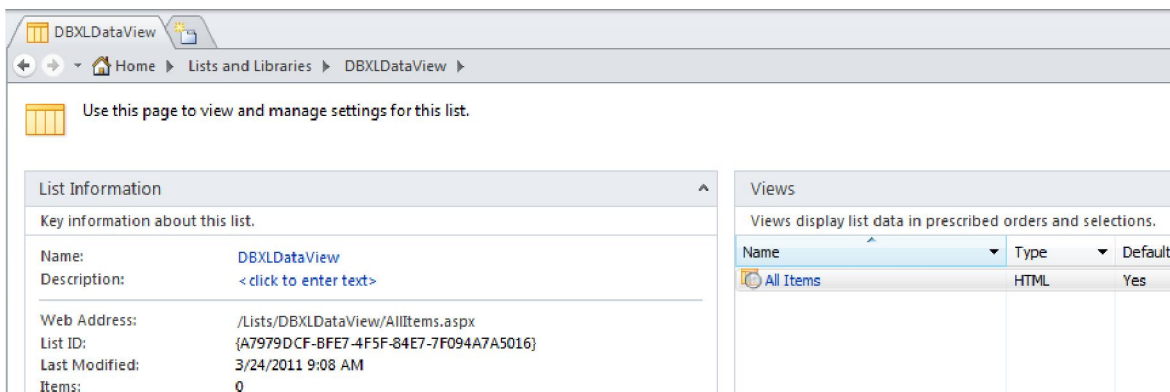
1. Launch SharePoint Designer 2010.
2. Go to **File > Open Site**. Enter the URL to your site and click **Open**. The site will load and might take a few seconds depending on the size of your site.
3. In the Navigation frame, select **List and Libraries**. This will load the list of available lists and libraries.



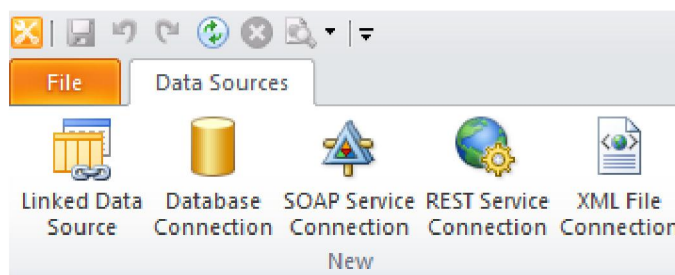
4. Click on the list we previously created to open it.



- Under **Views**, click on **All Items** to open this default view in the editor.



- In the **Navigation** pane, select **Data Sources**.
- In the ribbon, under **Data Sources**, select **SOAP Service Connection**.



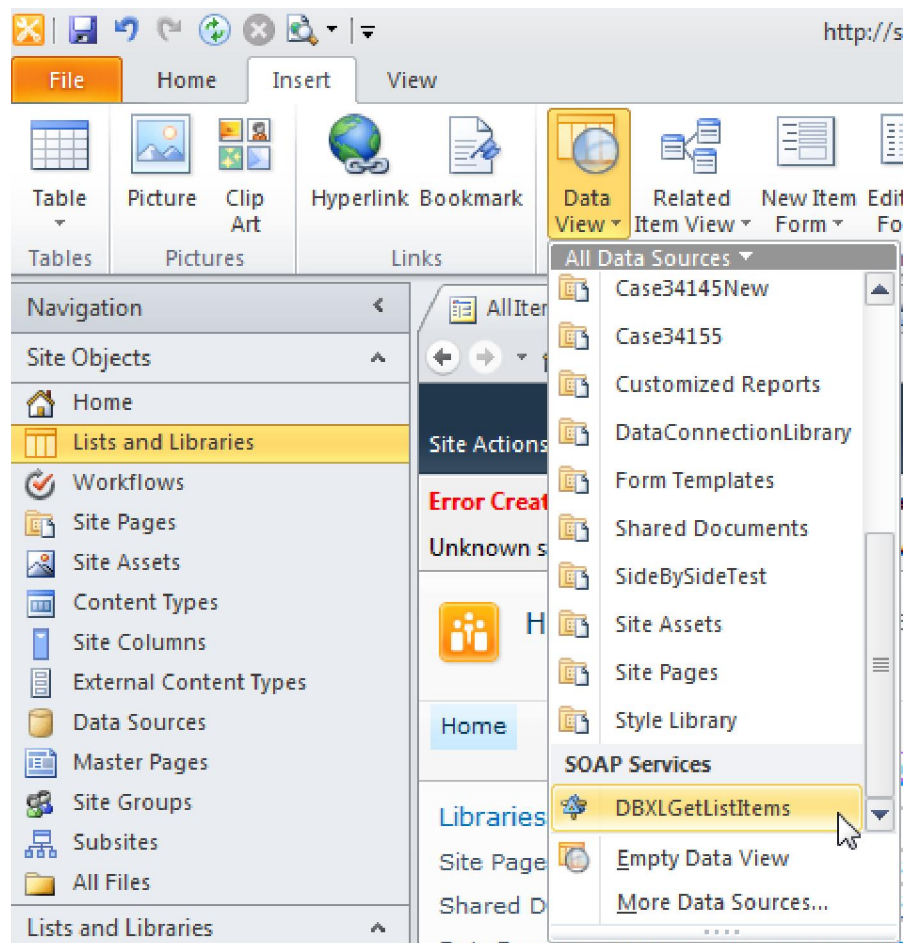
- In the **General** tab, enter a name for your data source, such as **DBXLGetListItems**.
- In the Source tab, enter the web service URL: <http://<your-server-name>/qdabrawebservice/dbxldocumentservice.asmx>
- Click on **Connect Now**.
- From the **Operation** dropdown select **GetListItems**.
- Double click on the **listName** and **viewName** parameters, and enter the name of the DBXL Document Type you wish to query (the same name used on page 17 when creating the document type). 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.



The screenshot shows the 'Data Source Properties' dialog box with the 'Source' tab selected. The 'Service description location' is set to 'http://shp2010-test2/qdabrawebservice/dbxldocumentservice.asmx?WSDL'. Below this is an example URL and a 'Disconnect' button. A dropdown menu for 'Select which data command to configure:' is set to 'Select'. The 'Select Connection Info' section shows 'Service Name: DbxlDocumentService', 'Port: DbxlDocumentServiceSoap', 'Address: http://shp2010-test2/qdabrawebservice/dbxldocumentservice.asmx', and 'Operation: GetListItems'. A table lists parameters for the 'GetListItems' operation, including 'listName', 'viewName', 'query', 'viewFields', 'rowLimit', and 'queryOptions'. At the bottom are buttons for 'Modify', 'Reset', 'Add Element', 'Remove', 'OK', and 'Cancel'.

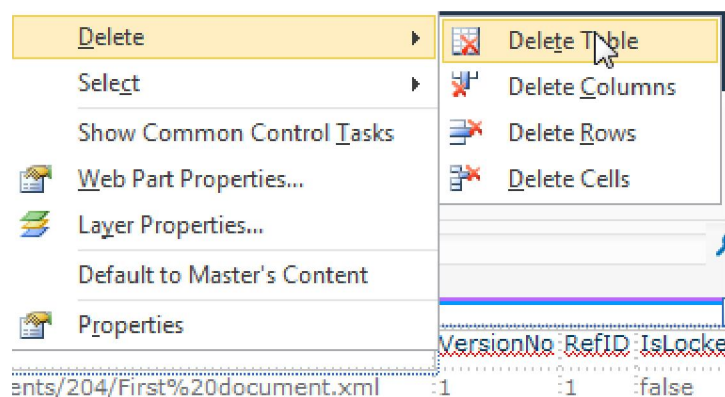
Name	Value	Element	Type	Runtim...
listName	SampleForm		s:string	☐
viewName	SampleForm		s:string	☐
query				☐
viewFields				☐
rowLimit	1000		s:string	☐
queryOptions				☐

14. Return to the AllItems.aspx edit view. SharePoint adds some generic layout tables and controls to this view, which you can remove or modify as desired.
15. Place the cursor on the location where you wish to enter the data view.
16. From the ribbon option **Insert**, select **Data View** and then select the **GetListItems** data source we created above. If it doesn't display among the options, you might need to select **More Data Sources**. If the **Data View** menu item is not enabled, your cursor is not in a location where the data can be displayed properly.



You now have a basic Data View, but it might not display as desired. The next steps will show you how to modify it. You'll notice that SharePoint designer selected a few columns to display. However, we might want to display a different set of columns.

17. Right click anywhere on the table displaying the data view, and select **Delete > Delete Table**.





18. You'll see that the Data Source Details are displayed in the right hand frame. Select the columns you wish to display by using Control+click. Once you have selected the desired columns, drag and drop into the location in the view where you wish to display the data view table.

You can also modify the available columns by going to **Options > Add/Remove columns** in the ribbon.

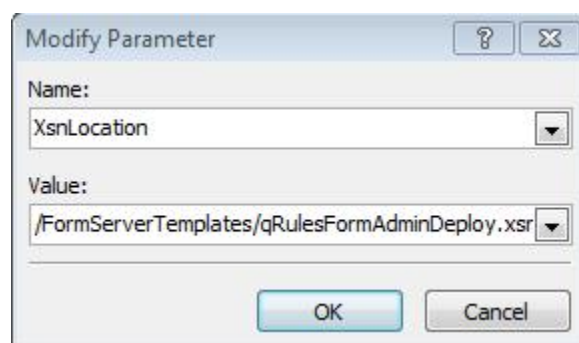
C. Add a hyperlink to the browser form

1. Select the column item you wish to use as a hyperlink, and then select **Insert > Hyperlink** from the ribbon.



2. Click on the **Parameters** button.
3. In the Path text field, enter `http://<your server name>/_layouts/FormServer.aspx`
4. Click **Add** button. Enter **XsnLocation** in the **Name** field.
5. Enter the value for your XSN location.

NOTE: Remember that previously in this document we published the XSN as an admin deployed form template. The (absolute) XSN location should be something like: `http://<your server name>/FormServerTemplates/qRulesFormAdminDeploy.xsn`. In the Parameter screen, we can use the relative path only, as shown in the screenshot below.



6. Click the **Add** button again.
7. Enter **DocID** and select **XSL:DocID** in the **Value** dropdown. Remove the '@' of DocID in the **Name** field and click **OK**.



Modify Parameter

Name: DocID

Value: {@DocID}

OK Cancel

8. Click the **Add** button. Enter **OpenIn** for **Name** and **Browser** as **Value**. Then click **OK**.
9. Click the **Add** button. Enter **Source** for **Name**, and the URL of your data view as **Value**. Then click **OK**. (This is the URL where the user will be taken once the browser form is closed.)

You will now have your parameters listed as follows:

Hyperlink Parameters

Path: https://shp.formsboard.com/_layouts/FormServer.aspx

Insert Field Value

Query string

Name	Value
XsnLocation	/FormServerTemplates/qRulesFormAdminDeploy.xsn
OpenIn	Browser
DocID	{@DocID}
Source	https://shp.formsboard.com/Lists/qRulesFormsList/AllIt

Move Up Move Down

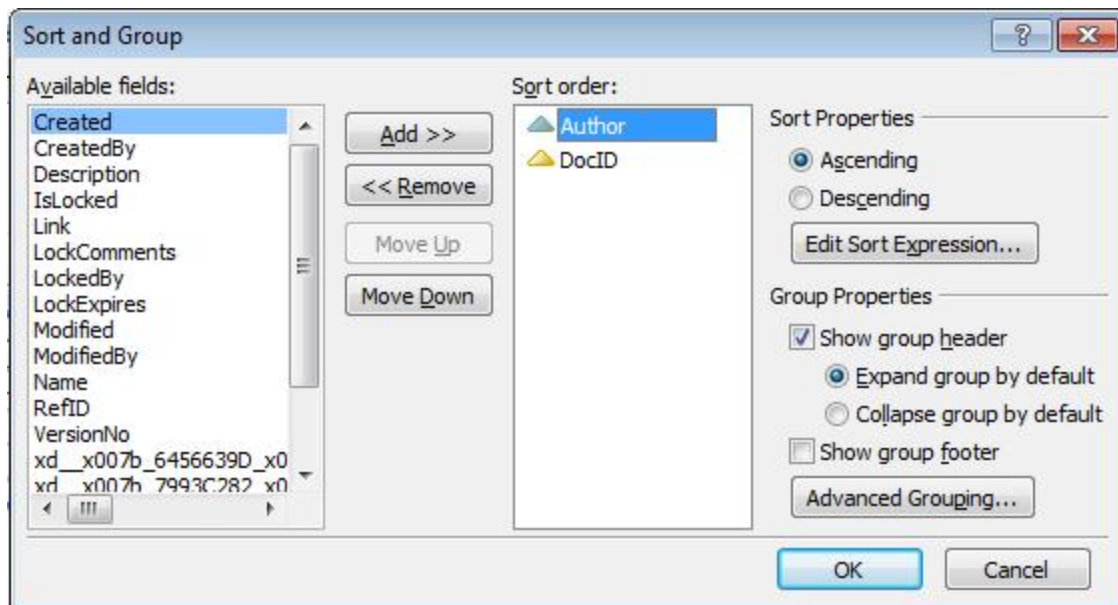
Add... Modify... Remove Clear

OK Cancel

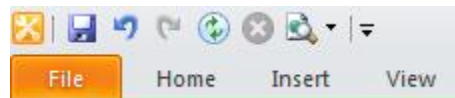
10. Click **OK** twice.

D. Data View sorting and grouping (*Optional*)

1. Place the cursor on the table and select **Options > Sort and Group**.
2. Select **DocID** and click **Add >>**. Repeat for **Author**.
3. Click **Author** in **Sort order**, click **Move Up**, and then click **Show group header**. Click **OK**.



4. Click the **Save** icon to save the changes to your SharePoint List.



Should your document type contain a large amount of documents, sorting and grouping may not be enough. You can use paging to improve performance. You can add paging by going to **Options > Paging**.