



QDABRA DATABASE ACCELERATOR V2.3

How to IPFS-enable a codeless DBXL solution

With DBXL v2.3 you can quickly integrate your solution with SharePoint and IPFS in order to create a browser-based solution. This document will illustrate an IPFS scenario using the Publish Library and XSN feature in DBXL.

Note that this step will not work if the form has code. If that is your scenario, please see a document called [How to IPFS-enable a DBXL Solution that contains Code](#).

PUBLISH A NEW SHAREPOINT FORM LIBRARY FROM DBXL

In this scenario we will take advantage of a new feature in DBXL v2.3 that allows us to create a new SharePoint Form Library directly from DBXL.

1. Create a new form
 - a. Open InfoPath and click **Design a form template**.
 - b. Under **Design a new**, select **Form template**.
 - c. Under **Based on**, select **Blank**.
 - d. Check the check box to **Enable browser-compatible features only**, and then click **OK**.
2. Create a Submit data connection
 - a. Select **Tools > Data Connections** and click **Add**.
 - b. Select **Create a new connection** to and select **Submit data**. Click **Next**.
 - c. Select **To a Web service** and click **Next**.
 - d. Enter the URL to the DBXLDocumentService, usually `http://<servername>/QdabraWebService/DBXLDocumentService.asmx`, and click **Next**.
 - e. Select **SubmitDocument** from the list of operations and click **Next**.
 - f. Select the `tns:xml` parameter, select the **Entire form** toggle, and check the checkbox for **Submit data as string**.
 - g. Click **Next**, and then click **Finish**.
3. Convert the data connection to UDC and approve it in SharePoint

This step is optional. The form will work from the browser and from InfoPath with or without UDC connections.

- a. Select the Web Service Submit connection created in step 2 and click **Convert**.
 - b. Enter the URL to the data connection library and click **OK**. The URL may look like this: `http://<servername>/MyDataConnections/SubmitScenario2.udcx`.
 - c. In the **Data Connections** window, click **Close**.
 - d. Open the SharePoint data connections library in Internet Explorer.
 - e. Click the down arrow next to the UDC file created in step 4 and select **Approve/reject**.
 - f. Select the toggle for **Approved** and click **OK**.
4. Design your form.
 - a. For the purposes of this example, add three **Text Boxes**.
 - b. Add a button and double-click on it to open the **Button Properties**.
 - c. Change the **Action** dropdown to **Submit** and click on **Submit Options**.



- d. Check the checkbox for **Allow users to submit this form** and select **Web Service** from the **Send form data to a single destination** dropdown.
 - e. Make sure that the Web Service submit data connection created in step 3 is selected in the **Choose a data connection for submit** dropdown.
 - f. Click **OK** twice to close the dialogs.
 - g. Back in InfoPath, select **File > Save** to save the form to your local drive.
5. Create and configure the new Document Type in DBXL
- a. In DBXL, click on **New Configuration**.
 - b. Enter the **Document Type Name**.
 - c. Click on the file attachment control and attach the InfoPath form you saved in step 4.
 - d. Go to the **SharePoint** tab.
 - e. Click on **Insert Mapping**.
 - f. Enter the URL of the SharePoint library you wish to create. The SharePoint server should be in the same machine where DBXL is installed.
 - g. Select **Object Model**.
 - h. Check the checkboxes for **Publish form and XSN** and **Use Forms Services**.

DBXL automatically detects the Document Type Name and uses it to set the parameter in the main Submit Data Connection of the form. The Use Forms Services checkboxes transmits this information to IPFS, thus allowing IPFS to correctly submit to the DocType in DBXL.

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

On a cold server you might encounter a timeout error even though DBXL attempted to create the library. This can be handled in two ways: you could simply click **Save** again. The alternative is to delete the library from SharePoint and then click **Save** again.

6. Enable the SharePoint library for IPFS
- a. In Internet Explorer, open the SharePoint library you created, click on **Settings** and select **Form Library Settings**.
 - b. Click on **Advanced settings** in the **General Settings** section.
 - c. Select **Display as a webpage** radio button in the **Browser-enabled Documents** section. Click **OK**.
7. Test the solution
- a. In Internet Explorer, open the SharePoint library created in step 7.
 - b. Click on **New**. Fill out the form and click **Submit**.
 - c. Go to the **Documents** tab in DAT and click **New**. Fill out the form in InfoPath and click **Submit**.
 - d. You can also test:
 - i. From DAT, open and submit the form that was created from the browser.
 - ii. From Internet Explorer, open and submit the form that was created from DAT.

ADDITIONAL NOTES

- The use of IPFS-enabled solutions is not supported if the installations of SharePoint and DBXL lie on different machines. This scenario involves UDC files and SSO configuration, and is not supported in DBXL v2.3. If your solution requires IPFS, we recommend installing SharePoint and DBXL on the same machine, and using the Object Model method when creating your mapping as describe in this document.



- Whenever mapping DBXL Document Types to SharePoint, we recommend using the "Publish Library and XSN" checkbox. However, should you need to create the form library from InfoPath, using File | Publish, you'll find that the forms' version numbers will differ. The workaround is to publish to SharePoint from InfoPath, then attach the published form from the SharePoint form library (<http://<servername>/FormLibrary/forms/template.xsn>) when creating the DBXL Document Type in DAT.
- On a cold server SharePoint submit could time out (even though it is successful). The reason is that SharePoint can take longer than 30 seconds to process a new document. As a result, the submit operation succeeds but IPFS reports to the user that it failed due to timeout. There is no way to change timeout in InfoPath without writing code.
- When you use the "Publish form and XSN" and "Use Forms Services" checkboxes in the SharePoint tab in DAT to IPFS-enable a solution, DBXL attempts to automatically determine the DocType name with every submit. However, this fails for Receive-type data connections that attempt to submit data. Instead, please use a Submit-type data connection.
- You could also use DBXLDocumentServiceEx.asmx. This "Ex" API differs from its "non-Ex" counterpart only in the way errors are handled. Methods defined in the "Ex" pages return a SoapException when an error occurs. This allows InfoPath's internal mechanism to detect the error and display an error message. The "non-Ex" methods return a StatusInfo object with a Status node, of value true or false, and error messages. Calling the non-Ex methods requires form logic to check the Success indicator and handle any errors.