



Indexing in DBXL v2.8

There are three different types of indexing that can be used to improve performance in DBXL.

1. **QueryDocumentNodeSet indexing** [\[Page 2\]](#)

If you are using QueryDocumentNodeSet, a checkbox in the General tab of the DBXL Admin Tool (DAT) will allow you to index documents and improve the performance of your queries. Note that this will increase the size of the database.

2. **DBXL Indexing** [\[Page 4\]](#)

To improve the performance of other web services within DBXL, a section in the General tab of DAT allows you to configure an index.

3. **SQL Indexing** [\[Page 8\]](#)

To improve performance of QueryDB queries and of Database mappings, an index can be added directly to SQL.

a) DBXL Database Mapping - add the index to the identity column

b) QueryDB scenario

- If you are filtering in your QueryDB query, index the column being compared.
- If you are sorting in your QueryDB query, index the column being user for sorting.
- If your QueryDB isn't performing filtering nor sorting, no index can be added.



ENABLE XML INDEXES

For the first scenario described above, simply edit the document type you wish to index, and check the Enable XML Indexes checkbox in the General tab (screenshot below).

DATABASE ACCELERATOR Administration Tool **QDABRA** SOFTWARE

Document Type: QdExpenseReport-Sample Save Save As... Catalog

General Database Permissions SharePoint Documents Taxonomy Tree

Document Type Details

Name	Solution Group
QdExpenseReport-Sample	
Description	
Last Modified By	
AUTONOMYSYSTEMS\ernestom	

InfoPath Form Template

XSN File	Form Template Properties
QdExpenseReport.xsn Microsoft InfoPath Form Template 11.9 KB Open	Form Version: 1.0.0.4 Last Modified: 8/7/2013 4:36:35 PM ☐ Form template is signed ☐ URN-based solution
URL to Form Template	
http://QSH2013:8000/QdabraWebService/Forms/ID/84/template.xsn	

Ref. ID

Format for Ref. ID	Base Reference #	XPath to Inject Ref. ID Into (Leave Blank to Disable)
	3	

Indexes

Name	XPath
Insert index	

Additional Options

☐ Enable Xml Indexes (requires extra storage space)

After checking the checkbox, click the Save button:



<http://www.qdabra.com>

Last updated on 7/1/2014 10:37 AM

Copyright © 2013 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

Document Type Details

Name	Solution Group
QdExpenseReport-Sample	 
Description	
Last Modified By	
AUTONOMYSYSTEMS\ernestom	

InfoPath Form Template

XSN File	Form Template Properties
 QdExpenseReport.xsn Microsoft InfoPath Form Template 11.9 KB Open	Form Version: 1.0.0.4 Last Modified: 8/7/2013 4:36:35 PM ☐ Form template is signed ☐ URN-based solution
URL to Form Template	
http://QSHP2013:8000/QdabraWebService/Forms/ID/84/template.xsn	

Ref. ID

Format for Ref. ID	Base Reference #	XPath to Inject Ref. ID Into (Leave Blank to Disable)
	3	

Indexes

Name	XPath
 Insert index	

Additional Options

☒ Enable Xml Indexes (requires extra storage space)



INDEXING BASED ON AN XPATH

To add this index, we'll add a node to the Expense Report sample to store the timestamp when the document was created.

The image shows two parts of a software interface. On the left is a 'Field or Group Properties' dialog box with a red border. It has three tabs: 'Data', 'Merge', and 'Details'. The 'Data' tab is active. Inside, the 'Name' field is set to 'timestamp'. The 'Type' is 'Field (element)'. The 'Data type' is 'Date and Time (dateTime)'. There are checkboxes for 'Repeating', 'Cannot be blank (*)', and 'Non-empty occurrence required', all of which are currently unchecked. The 'Default Value' section has a 'Value' field and a formula icon (fx). Below it, an example is shown: 'Example: 12/31/2000 11:59:59 PM'. At the bottom of the dialog are 'OK' and 'Cancel' buttons. On the right is the 'Fields' pane, which has a title bar with a dropdown arrow and a close button. Below the title is the instruction 'Drag a field to add it to the form.' Underneath is a tree view labeled 'Fields:'. The tree shows a hierarchy: 'expenseReport' (folder), 'employee' (folder), 'manager' (folder), 'expenseCode' (file), 'startDate' (file), 'endDate' (file), 'reportDate' (file), 'purpose' (file), 'items' (folder), and 'timestamp' (file). The 'timestamp' file is selected and highlighted in blue.

A simple Form Load rule will ensure this field is populated when the form is first created:



Rules ▼ ✕

Form Load [Icons: Add, Copy, Paste, Delete]

Populate timestamp on first open

✱ New ▼

Details for:
Populate timestamp on first open

Condition:
timestamp is blank

Rule type:
Action

Run these actions: * Add ▼

Set a field's value: timestamp = now()

This updated form template (XSN) is then attached to the QdExpenseReport-Sample document type in DAT, and the document type is saved.

Now we're ready to create the index:

1. In the General tab of DAT, click **Insert index**.

Indexes	
Name	XPath
Insert index	

2. Enter a name for this index, and enter the xpath to the timestamp node.

Indexes		Update
Name	XPath	
timestamp_index	/my:expenseReport/my:timestamp	
Insert index		

3. Click Save to save the document type.
4. Check the index in SQL by going to the QdabraDBXL database and checking the DocumentIndex table.



<http://www.qdabra.com>

Last updated on 7/1/2014 10:37 AM

Copyright © 2013 Autonomy Systems, LLC. All rights reserved.

SQLQuery2.sql - qshp2013\...\E...

```
SELECT *
FROM [QdabraDBXLv28].[dbo].[QdDocumentIndexField]
WHERE Name='timestamp_index'
```

Results Messages

	ID	Name	XPath
1	4588885B-EA47-4D9D-83D2-59A516B08CAE	timestamp_index	/my:expenseReport/my:timestamp

- Submit a document and find the DocID in the Documents tab:

Documents							Delete	Delete All	Reshred	Reshred All
Doc. ID	Ref. ID	Version #	Name	Author	Description	New				
890	4	1	2014-05-09	John Doe	This is a test	Open				

- Then verify the IndexValue to check the index was recorded correctly:

SQLQuery4.sql - qshp2013\...\E...

```
SELECT *
FROM [QdabraDBXLv28].[dbo].[IndexValue]
```

Results Messages

	IndexValueID	DocID	QdIndexId	Value	OptimisticLockField
1	1	890	4588885B-EA47-4D9D-83D2-59A516B08CAE	2014-05-09T16:32:42	0

If you have existing documents in your document type, the indexes will need to be updated. To do this:

- Click **Update Indexes** in the General tab

Indexes		Update
Name	XPath	
timestamp_index	/my:expenseReport/my:timestamp	Update Indexes

☒ Insert index

- Click **Reshred All** in the Database tab:



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=qshp2013\sqlexpress;Initial Catalog=QdabraUtilityv28;Integrated Security=True;Application Name="Qdabra DBXL"

Test Connection

Submit Options

☒ Enable History
☒ Report Shredding Errors to the Form User

If reshredding more than 100 documents, please use the DBXL Migration Tool

Database Map

 Reshred All Documents
 Refresh Database

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

Reshred All Documents (Including documents not listed)

Note: If the document count exceeds 200, Qdabra recommends using the [Migration Tool](#) to ensure no timeouts happen while reshredding all the documents.



<http://www.qdabra.com>

Last updated on 7/1/2014 10:37 AM

Copyright © 2013 Autonomy Systems, LLC. All rights reserved.

SQL INDEXES

Since there are two scenarios where adding a SQL index will improve performance, this section will cover each separately.

DATABASE MAPPING

To illustrate this example, we'll use the Expense Report document type included in every installation of DBXL and its QdExpenseReportItem table. The document type included in DBXL already comes with the following database mapping:

Table Name (9 Columns) ▼		Node Path		
QdExpenseReportItem ▼		my:items/my:item  		
Column Name		Key Type	Parent Column Name	
ExpenseReportId ▼		Column	DocId ▼	
☒ Insert Foreign Key				
Column Name	<u>Add Column</u>	DB Type	Key	Node Path
AdditionalInfo ▼		String(Inn ▼)	☐	my:additionalInfo  
AdditionalInfoPlain ▼		String ▼	☐	my:additionalInfo  
Amount ▼		Decimal ▼	☐	my:amount  
Category ▼		String ▼	☐	my:category  
Date ▼		DateTime ▼	☐	my:date  
Description ▼		String ▼	☐	my:description  
EntertainmentPlace ▼		String ▼	☐	my:entertainmentDetails/my:entertainmentPla  
EntertainmentPurpose ▼		String ▼	☐	my:entertainmentDetails/my:entertainmentPui  
MerchantName ▼		String ▼	☐	my:merchantName



The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'QdabraUtilityv28', with the table 'dbo.QdExpenseReportItem' selected. The right pane shows the table's structure with the following columns:

Column Name	Data Type	Allow Nulls
ExpenseReportId	bigint	☐
Date	datetime	☒
Description	nvarchar(128)	☒
Amount	decimal(18, 2)	☒
Category	nvarchar(50)	☒
MerchantName	nvarchar(128)	☒
EntertainmentPurpose	nvarchar(512)	☒
EntertainmentPlace	nvarchar(100)	☒
AdditionalInfo	nvarchar(MAX)	☒
AdditionalInfoPlain	nvarchar(MAX)	☒
		☐

The QdExpenseReportItem table does not contain an identity column, so we'll add one.



QSH2013\SQL...seReportItem*

Column Name	Data Type	Allow Nulls
AdditionalInfo	nvarchar(MAX)	☒
AdditionalInfoPlain	nvarchar(MAX)	☒
ID	int	☐

Column Properties

(General)

(Name)	ID
Allow Nulls	No
Data Type	int
Default Value or Binding	

Table Designer

Collation	<data:
Computed Column Specification	
Condensed Data Type	int
Description	
Deterministic	Yes
DTS-published	No
Full-text Specification	No
Has Non-SQL Server Subscriber	No
Identity Specification	Yes
(Is Identity)	Yes
Identity Increment	1

Save the updated table, and then execute:

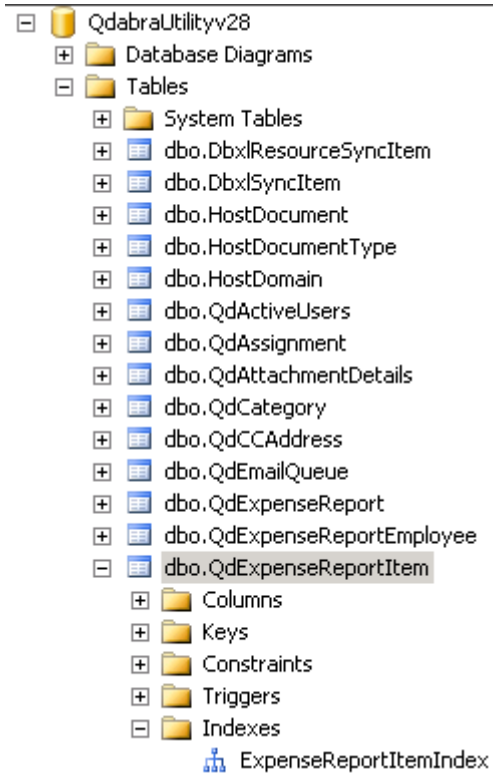
```
CREATE INDEX index_name ON table_name (column_name)
```

In our case, we execute:

```
CREATE INDEX ExpenseReportItemIndex ON QdExpenseReportItem (ID)
```

The table now has an index, as shown in the screenshot below. This index will help speed up the Database Mapping operations.

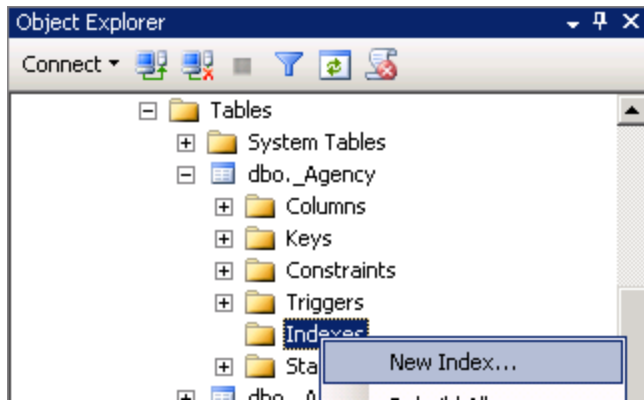




QUERYDB

To improve performance of QueryDB, an index can be added directly to SQL.

To add in index, you can use SQL Server Management Studio. Locate the table being queried, and add a new index:



Enter a name for the Index, then click Add to select the appropriate column.

- If you are filtering in your QueryDB query, index the column being compared.
- If you are sorting in your QueryDB query, index the column being user for sorting.

Then click OK, and the index is ready!



New Index

Select a page

- General
- Options
- Included Columns
- Storage
- Spatial
- Filter

Script Help

Table name: Agency

Index name: QueryDBIndex

Index type: Nonclustered

☐ Unique

Index key columns:

Name	Sort Order	Data Type	Size
Category	Ascending	nvarchar(50)	100

Add...
Remove
Move Up
Move Down

Connection

Server: qshp2013\sqlexpress

Connection: AUTONOMYSYSTEMS\Ernestok

[View connection properties](#)

Progress

Ready

OK Cancel

