Database Indexer 1.0

User Guide

Database Indexer creates dtSearch compatible indexes of SQL databases, on premises, in the cloud, or both.

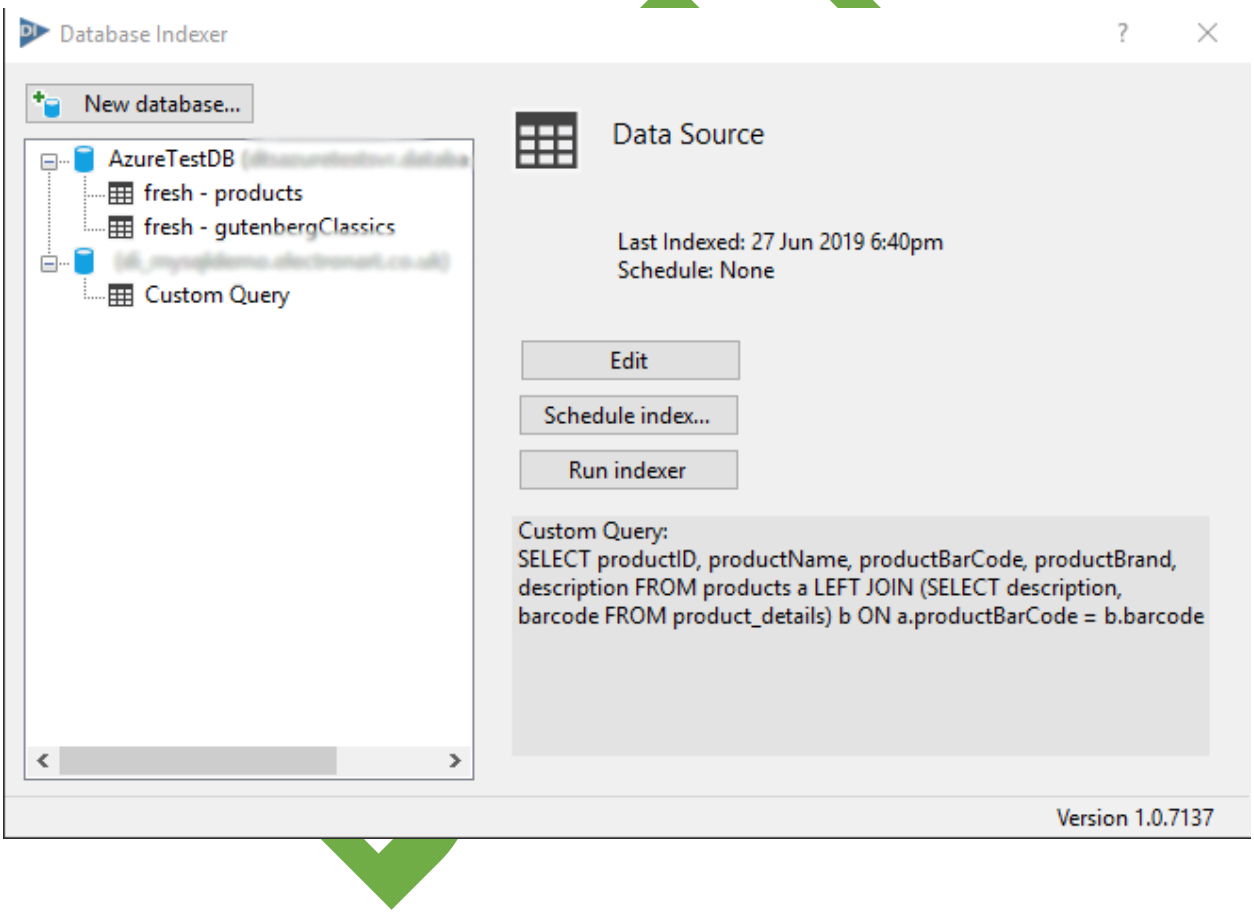
Contents

Getting Started

The steps to create an index are Add a New Database, Add a data source, set the index options and finally set a schedule for the index updating if needed. All the steps can be edited later.

If you are evaluating, you might want to try connecting to our demonstration databases first before testing with your own data. You will find details of how to connect and create indexes in the Demo section. Contact us to obtain the URLs and Passwords.

Evaluation mode is limited to 10,000 rows per index. If you have a serial number, enter it to remove the limit, you can always find your serial number by clicking on the help (?) icon in the title bar.

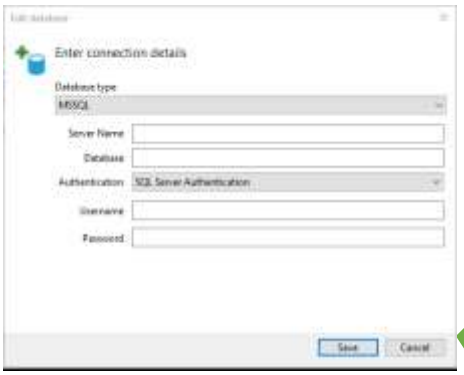


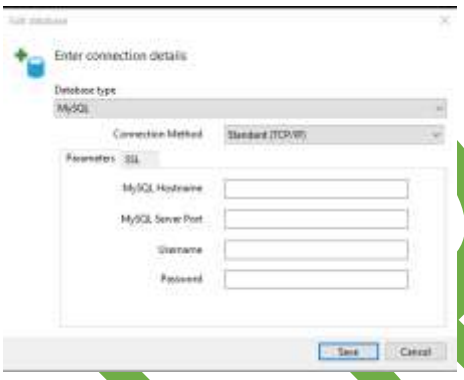
Add a New Database

Select the **database type** from the drop-down list and enter the requested parameters of your database.

You may prefer to use our cloud hosted demonstration databases while evaluating. See next section. Contact us for the latest links and passwords.

The images below show the options for the MS SQL and mySQL database types.





Demonstration databases

These are the cloud hosted demonstration databases available as at 16 July 2019.

1. Azure SQL
2. mySQL hosted on AWS
3. Custom Query

Evaluation using the demo Azure SQL database:

Run DatabaseIndexer.exe

Connect to the Azure SQL database as follows:

1) Click the New Database button (press Enter)
2) Select Database Type: MSSQL
3) Enter Server name: *
4) Enter Database: Azure DB
5) Select Authentication: SQL Server (or Windows)
6) Enter Username: *
7) Enter Password: *
8) Click the Save button (Enter)

Add demo data source (a) as follows:

1) Click Datasource button
2) Select Schema: fresh
3) Select Table: products (18,560 records of supermarket items)
4) Click All (13 fields text, Boolean, numeric) productID, productName, productCreationDate, productBarcode, productLastModified, productDeleted, suggestedExpiry, suggestedLocation, ImpDate, hasAmazonLink, hasWaitroseLink, hasImageInFS, productBrand.
5) Leave bottom item as "Don't Extract"
6) Click Next (Enter)

Choose the Index Location:

1) Select New Index
2) Name: fresh – products
3) Location: Browse to your dtSearch Desktop index folder (e.g. c:\...\Documents\dtsearch\)
4) Click Save (Enter)
5) Click the Run Index button

18,560 records should be indexed in approx. 10 seconds.

Searching:

Run dtSearch Desktop and select the `fresh – products` index (If it doesn't appear in the list of indexes, from the Index menu>Index Manager>Recognise index…).

In the Search dialog note that clicking on the **fields…** button should show 14 fields

(13 + Filename)

Type a search query: `xfirstword`

It should return 5000 results by default. From the **Search Dialog>More Search Optio**ns tab - edit the limit to 20,000 so that it will show all the results from the index.

Try some field searches, for example:

Click on the **fields…** button, choose `productBarCode`, press OK and enter 3263670041279

i.e. (`productBarCode` contains (3263670041279))

Evaluation using the demo Azure SQL database containing documents

1. Select the AzureDB database
2. Click the Datasource button
3. Select fresh
4. Select gutenbergClassics
5. Select All Columns (id, name, author, year, filecontent)
6. Click Setup…
7. Select File Data Column: Extract filecontent
8. Interpret as File content Binary
9. Filename Column: name
10. Click save
11. Select data to Index
12. Select Schema fresh
13. Select Table gutenbergClassics
14. Select All Columns
15. Document Extraction Setup…
16. Next…

Set Index Destination
New index
Name fresh – gutenbergClassics
Location – c:\...\Documents\Database Indexes
Save
Run Index 5 records approx. 1 second.

Search
In dtSearch Desktop search on 'abandoned', this should return 3 results from A Tale of two Cities, Jane Eyre and Dracula with the word abandoned highlighted.

The document properties are displayed at the end of the document.

Evaluation of Custom Queries using MySQL:

Contact us for the Hostname, Port, Username, and Password.

1. Ensure you're running Database Indexer 1.0.7109 or later
2. Create a new database connection with the following settings:
    - Database type: *MySQL*
    - Hostname: *
    - Port: *
    - Username: *
    - Password: *
3. Using the newly created database, create a new data source:
    - Set Schema to *fresh*
    - Select Custom Query and Set the following:
        - Add the following query:
          SELECT productID, productName, productBarCode, productBrand, description FROM products a LEFT JOIN (SELECT description, barcode FROM product_details) b ON a.productBarCode = b.barcode
        - Click *Test Query*. This will enable the Pagination Column Combo box.
        - Select *productid* as the pagination column.
        - Press OK. Columns will automatically be detected and displayed as options based on the result set of the custom query.
    - Select all columns, leave document extraction set to don't extract.

The indexer can be run as normal.

The created index will contain combined data from two tables (the products table, and product_details table) Where *productid, name, barcode, brand* come from the products table and the *description* from the product_details table.

Evaluation of Custom Query using Azure MSSQL

Whilst there is no demo pair of tables that can be used for a join statement, it's possible to demonstrate that custom queries are working in MSSQL as well.

1. Create a new data source for the Azure database with schema set to `fresh`.
2. Set the custom query to `SELECT * FROM fresh.products`
3. Select all the columns, save and run the index.

Note that a subtle difference between MySQL and MSSQL is that MySQL does not need the schema in the select statement i.e. (SELECT * FROM products) but MSSQL needs (SELECT * FROM fresh.products)

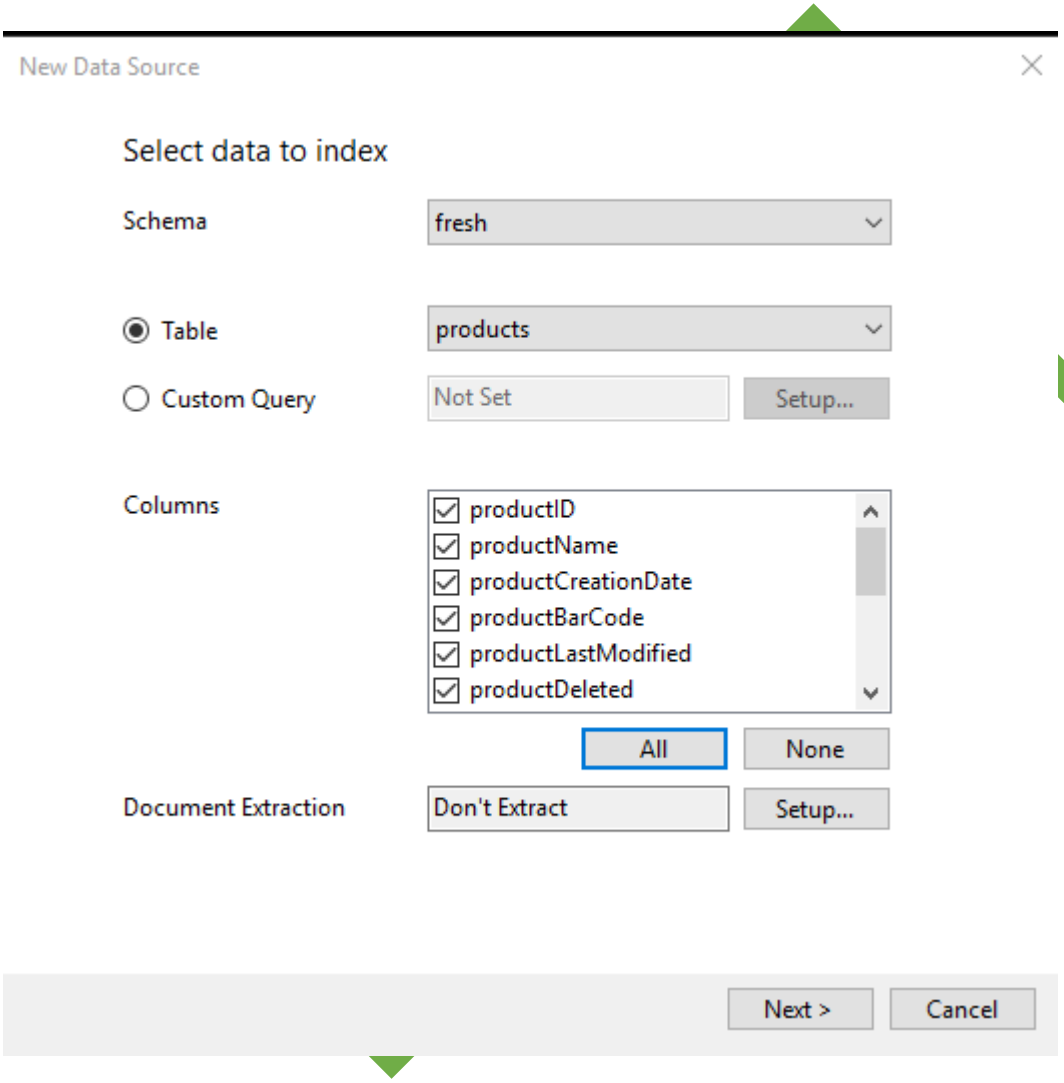
Add a data source

After you have added a new database, press the **New Data Source** button.

Select the **schema** from the drop-down list.

Choose to select a **Table** or create a **Custom Query**.

If any of the columns contain a document or file path to a document, you might want to set up **Document Extraction**.



Finally press the **Next >** button and set up the index.

Set up the index

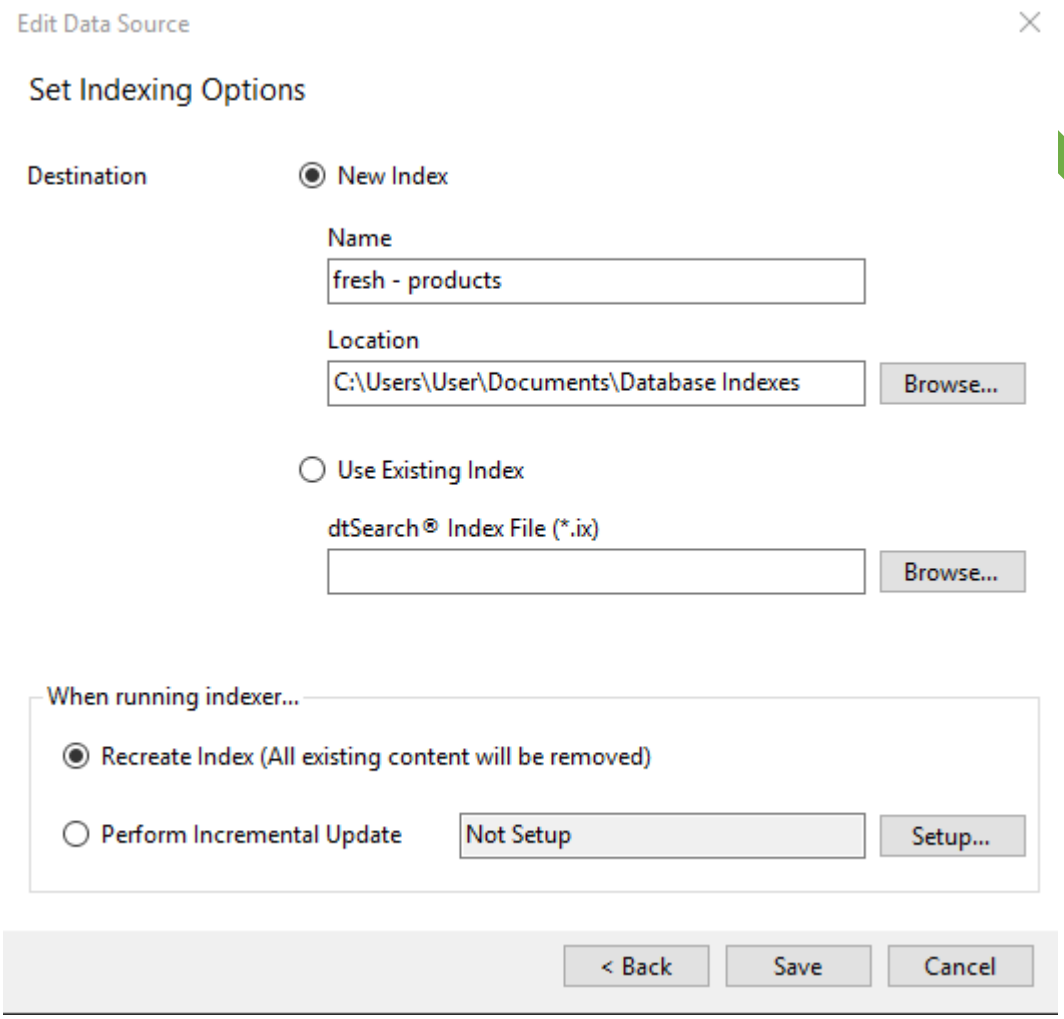
You can create a **New Index** or use an existing dtSearch index. The index name can consist of several words and can contain spaces.

Choose an index **Location** on the same machine where you have read/write permissions.

CAUTION: If you choose an existing index, and select to recreate the index, any data not previously indexed by the Database Indexer will be lost. (The software will also warn you if you select this option)

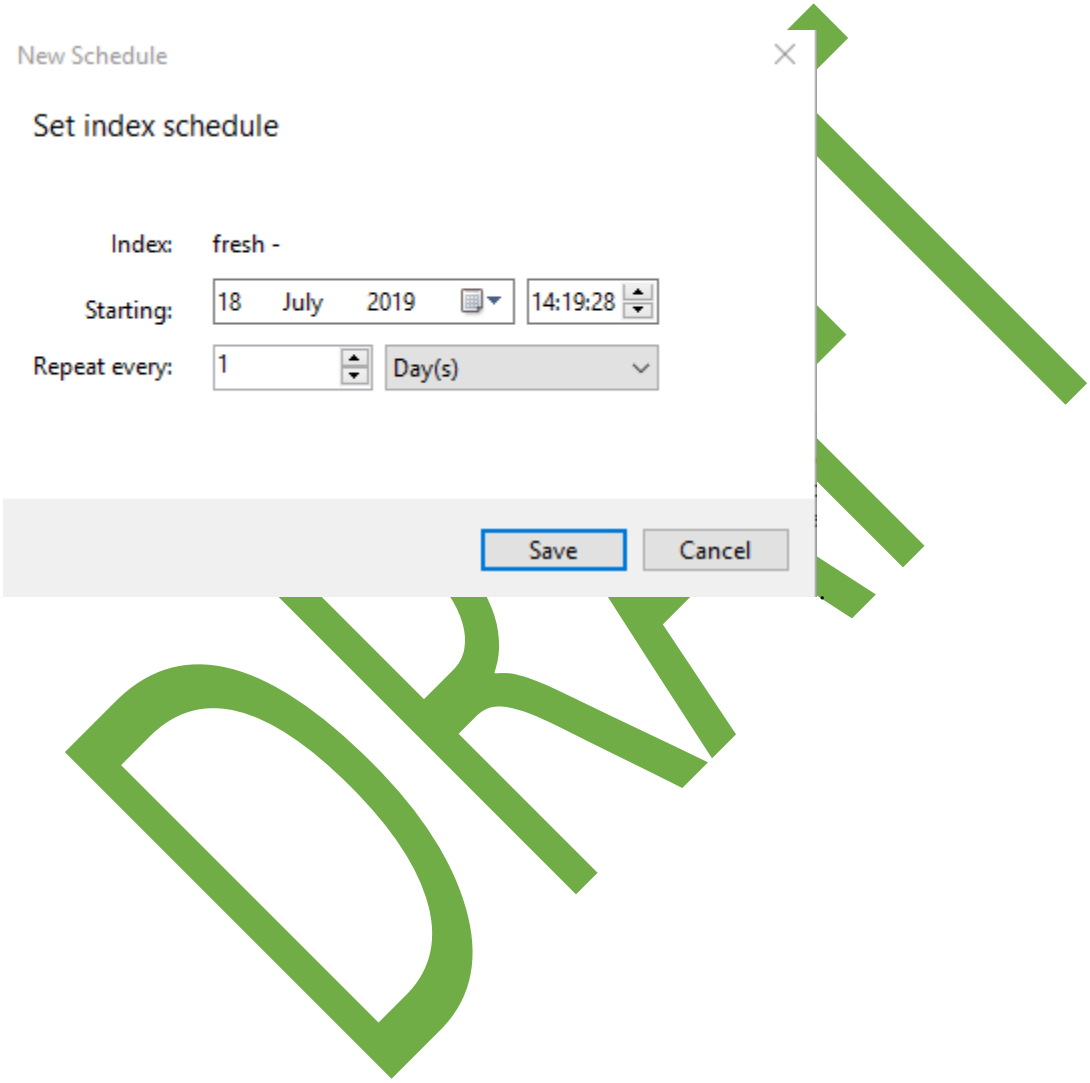
Incremental index update requires a column in the table that has a 'last modified' date. Database Indexer needs read-only permission to the database.

Press the **Save** button.



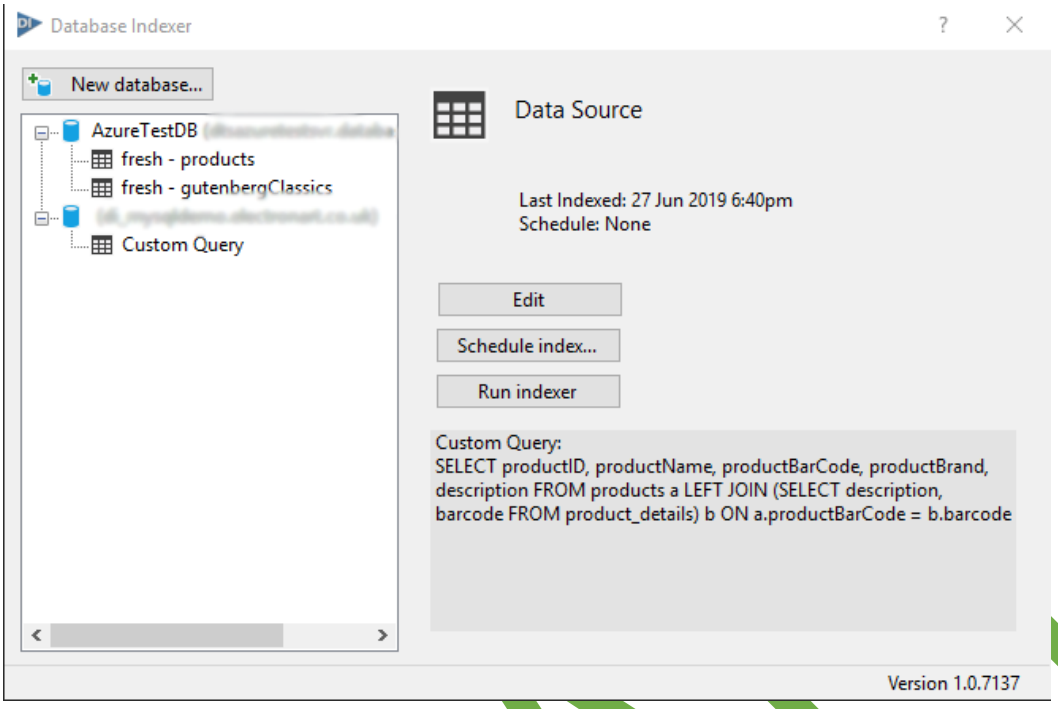
Schedule the Index updates

If you want to rebuild or update the index daily, you need to ensure that it can be rebuilt or updated within 24 hours. For large databases rebuilding might not be possible within 24 hours, it is recommended to use incremental updates on a scheduled basis. Incremental updates only re-index those rows in the table that have been modified since the last update.



Remove an index

Custom Queries



Get help

If you need assistance with evaluating or setting up your own databases, please contact support@dtsearch.co.uk.

Settings

This initial release has default settings for the alphabet file.

If document extraction is configured the document will be cached in the index.

All other index options (i.e. those listed on the index options dialog in dtSearch Desktop) are 'off'.

Feedback

Please give us feedback on this user guide so that we can provide content that's useful and helpful. Thank you!