

# MYOB Exo Business White Paper

## 2019.4.1 Upgrade - Unicode Database Conversion

Last Updated: 21 November 2019

**myob**

# Contents

Introduction	1
Before You Begin.....	1
Size of Database After an Update.....	2
Converting Custom Tables to Unicode	3
Before You Begin.....	3
Using Wildcards.....	3
Converting Custom Tables.....	4
Performing the Database Update	5
Conversion of Database Triggers to Unicode.....	7
Logging.....	7
Handling of Extra Search Fields and Extra Fields.....	8

# Introduction

The major change in release 2019.4.1 of Exo Business involves updates to the database to support Unicode. You'll need to run the database updater tool to apply those changes.

The process to update the database hasn't changed from previous versions of Exo Business. There is, however, an additional step (that occurs after all changes for version 2019.2 have been completed) to convert standard [varchar](#) columns in Exo's database to their Unicode [nvarchar](#) equivalents.

**Note:** You can pause and prompt between major versions by pressing CTR+ALT+SHIFT+F12.

MYOB recommends that you only perform the update on installations of Exo Business at version 2017.1 or newer — ideally version 2019.2. During our testing, we upgraded databases directly from version 2016.3 of Exo Business, but we don't recommend this for larger sites.

**MYOB cannot guarantee the update will always succeed or that you won't run into usage issues.**

**Note:** If the upgrade process fails, the updater rolls back the failed transaction and reports the error. You should correct the error and rerun the update process multiple times. Run tests on a copy of the database you plan to upgrade before upgrading a production database.

## Before You Begin

This update makes extensive changes to the Exo Business database. **You cannot revert those changes after performing the upgrade.** You cannot use previous versions of Exo Business with a database that has been converted to version 2019.4.1 or above.

Before updating, **back up your database and test that the backup is valid.** If anything goes wrong, you can restore the older version of your database and try again. Or, you can continue to use the previous version of Exo Business until you resolve the issues causing the upgrade failure.

If your SQL database contains a large amount of data, and has not been optimised recently, consider [shrinking your database](#) to optimise what's on disk.

**Note:** Shrinking the database can take quite a bit of time if you have a large database.

Sites with custom tables, database triggers, routines, and views should follow the instructions in the section [Converting Custom Database Tables to Unicode](#) before performing the upgrade.

## Size of Database After an Update

We've tested the impact on disk space when running the database upgrade on large databases and determined that the size of the database can change significantly. How much the database grows depends on the number of text columns the updater must modify in a table, and the number of transactional rows in that table.

After the conversion process completes, your database may be between 10% to 20% larger.

**Note:** The Exo Business Database Update utility cannot manage the size of your database log files. SQL Server manages the logs independently.

Because the transactional log file rapidly expands to log these changes, MYOB recommends that you have *five times* the amount of free space that your database files currently occupy on disk.

# Converting Custom Tables to Unicode

As is part of the database update process, the Exo Configurator Database Update tool can automatically convert custom database triggers, routines, views, and tables to Unicode. Enabling the conversion of custom elements in the database requires some setup work on your part.

## Before You Begin

You must create these text files and save them in the Exo Business *Exe* folder:

- CUSTOM\_TRIGGERS.LST
- CUSTOM\_ROUTINES.LST
- CUSTOM\_VIEWS.LST
- CUSTOM\_TABLES.LST

The files specify which database triggers, routines, views, and tables to convert. Or, you can specify which elements to exclude from the conversion process.

**Note:** Ensure that the file names are spelled correctly. If they aren't, the database update process ignores the files.

Each item that you need to convert must be on its own line in a file. So, if you need to convert a dozen custom database triggers, the file CUSTOM\_TRIGGERS.LST must contain 12 lines.

## Using Wildcards

**Note:** *Do not* use wildcards with custom routines, custom tables, or custom triggers.

You can specify database views to convert to Unicode with these wildcards:

- \* - specifies *all* characters
- ? - specifies a single character

In a text file, add those characters anywhere in the name of a view.

**Note:** If the database updater can't find a view specified in the .LST file, it ignores what's missing and continues with the update.

# Converting Custom Tables

The file `CUSTOM_TABLES.LST` works differently from the other files that you set up for the conversion. You can use wildcards with table names — for example, `*` to convert all tables, or `CLIENT*` to convert all tables whose names start with `CLIENT`.

**Note:** The database updater always converts standard Exo tables to Unicode, even if those tables are listed in the file `CUSTOM_TABLES.LST`.

You can also add the name of a column after the name of a table. Just put a comma after name of the table, followed by the name of the column. For example:

**CLIENT,STOCKCODE**

To convert all database tables that contain the `STOCKCODE` column, add this to the file `CUSTOM_TABLES.LST`:

**\*,STOCKCODE**

You can add a dash (-) in front of the name of a table to exclude it from a conversion. For example:

**-CLIENT\*,STOCK\***

That excludes from the conversion all columns whose names start with `STOCK` in database tables whose names start with `CLIENT`. However, the rest of the columns in the `CLIENT` database tables **are** converted.

You can also add a plus sign (+) in front of the name of a table to explicitly include it in the conversion. For example:

**+MYOBJECT\***

# Performing the Database Update

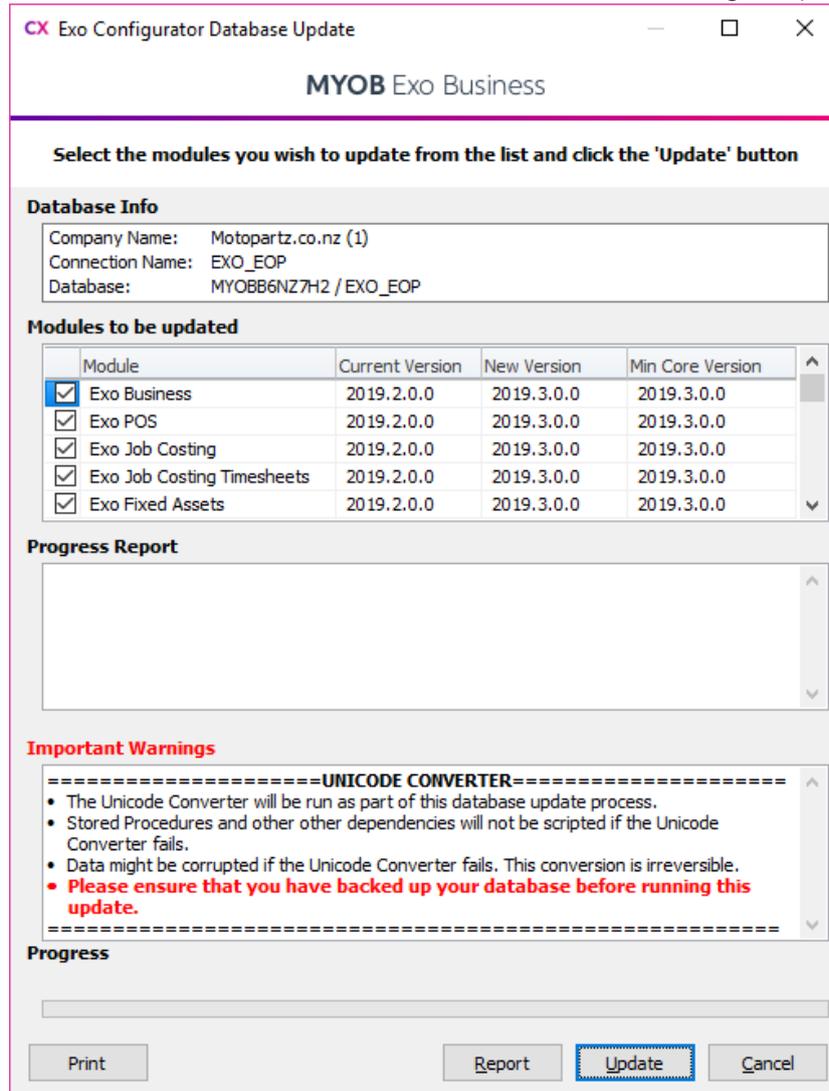
This update follows the steps of the Exo Business database update process from previous releases. However, the update incorporates steps to convert varchar database fields to support Unicode, and optionally, to [convert custom tables](#), triggers, views, and functions to Unicode.

**Note:** The amount of time required for an update depends on the size of your database and the performance of your systems. In a medium-sized implementation, this may be 10 or 15 minutes.

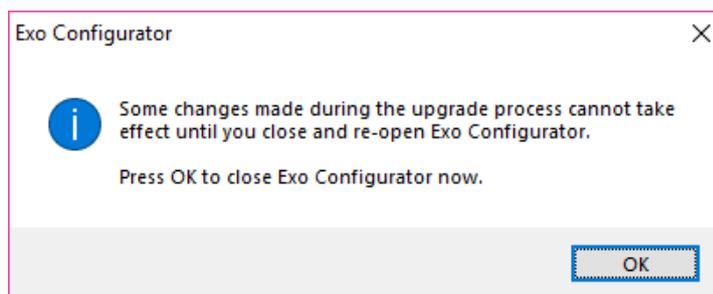
After installing the latest version of Exo Business, perform the database update by:

1. Starting Exo Configurator.
2. Logging in using the administrator's user name and password.
3. Clicking **OK**. The Exo Configurator Database Update screen displays.
4. Deselecting modules to exclude from the update (MYOB *does not* recommend this), and then clicking the **Update** button.  
A warning message displays if your database contains customised stored procedures or other data.  
You must convert the stored procedures or data to the standard Exo Business format *before* you can continue with the upgrade.
5. Once the update completes, the next step is to convert the database to Unicode.

Information about the database to convert and warnings display:



6. If you click:
  - a. **Update**, Exo Configurator drops all indexes and constraints in the database. Exo Configurator then converts all tables, views, and routines in the database to Unicode.
  - b. **Cancel**, the Database Update screen closes. Back up your database and then log back into Exo Configurator.  
When you click the **Update** button on the Database Update screen, the message that displayed in step 5 pops up. Click **Yes** to convert the database to Unicode.
7. When the update finishes, this message displays:



Click **OK**.

If the database service fails or unexpectedly stops, Exo Business halts the conversion process. The last change that failed is rolled back. When you log back into Exo Configurator, the conversion process starts from where it left off.

## Conversion of Database Triggers to Unicode

The database update utility converts associated triggers, functions and stored procedures that handle text variables to Unicode.

The update utility drops all triggers before converting database tables to Unicode. The utility then recreates the triggers and converts the fields in the triggers to [nvarchar](#) data types (which can handle Unicode data).

**Note:** If you need to convert *custom* database triggers, you must [set up an exclusion file](#) before running the database update.

## Logging

The progress of a conversion displays in the **Progress Report** section on the Exo Business Configurator Database Update screen. However, that information scrolls too quickly for you to view it in detail.

All information about the conversion is written to the file *Unicode\_Conversion\_log.txt*, which Exo Business Configurator Database Updater saves to the Exo Business *Documents* folder. If that folder doesn't exist, or if the updater can't find the folder, then the database updater tries to save the log file to your computer's public *Documents* folder. If it can't do that, then a file save dialog box displays, where you can specify the folder in which to save the log file.

Errors encountered during the update process are listed at the end of the log file. Check the log file and fix the errors. Then, run the database update again.

# Handling of Extra Search Fields and Extra Fields

[Extra search fields](#) declarations are converted to Unicode as part of the database update process. Exo Business now recognises them as string fields so that a column name can display non-Western characters.

[Extra fields](#), however, are not converted to Unicode during the database update. This ensures that third-party software, which might not support Unicode, can continue to integrate with Exo Business and access the extra fields.

If you see the message “(WIDEMEMO)” in an extra field after upgrading, you will need to either:

- Set the Field type in Extra Fields to “Memo”, OR
- Make the Field type shorter (NVARCHAR(4000) or shorter)—this is strongly recommended for performance reasons.