Scheme rate and trade changes

Contents

[Introduction 2](#_Toc150863730)

[Identify whether old or new style calculator 2](#_Toc150863731)

[Changing a new style calculator 3](#_Toc150863732)

[Changing an old style scheme 4](#_Toc150863733)

[Search for tables in database 4](#_Toc150863734)

[Find tables using WPD 4](#_Toc150863735)

[Versioning 5](#_Toc150863736)

[Test the Scheme in TGSL 7](#_Toc150863737)

[Release to live TGSL 8](#_Toc150863738)

# Introduction

This document assumes an understanding of Schemes and calculators. For more information see the Creating a new Scheme.docx documentation in one of the following locations:

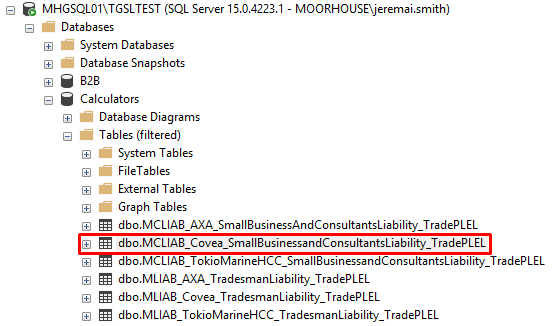
* GIT\Documentation\Scheme
* <https://constructaquote.monday.com/boards/671792066/pulses/674060287>

It is impossible to provide a single set of step-by-step instructions for rate and trade changes, as every Scheme is different. Some older schemes are version controlled with a version number and require every record in every table to be duplicated with the new version number.

Some schemes hold a rate against each individual trade, while others hold rates against bandings which are applied to the trades. However, there are similarities and this document gives an overview.

# Identify whether old or new style calculator

Historically, the calculator logic has been built into files called WPDs (the file extension is .wpd) with calculations written in TGSL’s proprietary TCAS calculator language. New style schemes still require a WPD file to interface with TGSL. However, a standard file is now created which acts as little more than a link, providing the risk information to SQL procedures, functions and views that exist in the Calculators database.

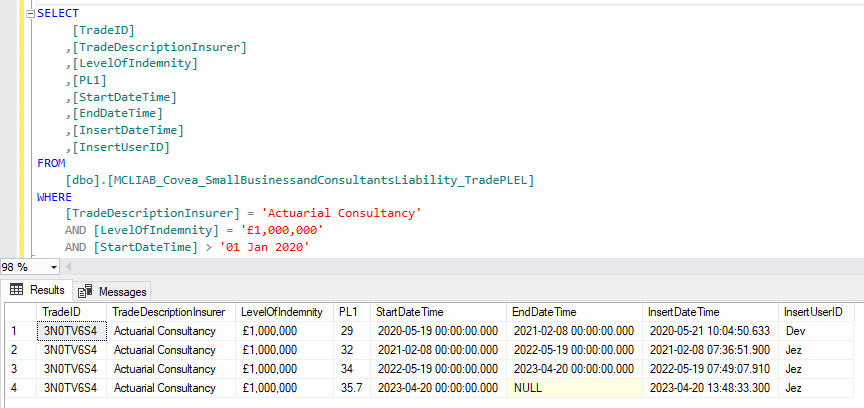
1. Launch SQL Server management Studio and connect to **MHGSQL01\TGSLTEST**
2. Expand the **Calculators** database and then expand the **Tables** folder
3. Look for a table named “<Product code>\_<Insurer>\_<Product>\_Trade…” or “<Product code>\_<Insurer>\_<Product>\_Rate…” for the Scheme you need to change. For example if the change is for the Covea Small Business and Consultants Liability Scheme you can see there is a trade table for this:  
     
   Note, the Tables folder can be filtered by right-clicking and selecting **Filter > Filter Settings** to help find a table.

|  |
| --- |
| **If you find tables for the Scheme, this means it already has a new style calculator. Follow the instructions for** [**Changing a new style calculator**](#_Changing_a_new) **below.**  **If the Product does NOT already have a new style calculator, follow the instructions for** [**Changing an old style schem**](#_Changing_and_old)**e below.** |

# Changing a new style calculator

After identifying the tables to be updated in the previous step, the basic way to change a trade or rate is to create a new row in the table with a new start date, and then set the end date on the old row to expire it.

For example, the following query shows the changes in the one person public liability rate for £1m limit of indemnity in the Covea Small Business and Consultants Liability trade table for the Actuarial Consultancy trade:



The public liability rate (PL1 column) has changed annually and so a new row has been inserted each time.

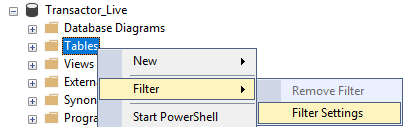
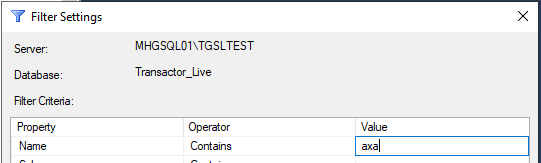
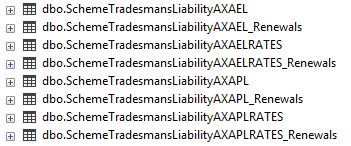
For any change to a table, write a SQL script:

1. Script insert of new row or rows. Trade and rate changes are usually done in bulk, e.g. increasing the rates across all trades. This may require loading up a spreadsheet of new rates into a staging table and referencing this table in the script. The staging table can be created in the StagingTables database.
   1. Copy any values that are not changing from the previous version of the row
   2. Unless a start date is specified on the ticket, use the current date (at run time, since the live release may happen days or weeks later), without a time element
   3. The InsertUserID column, and UpdateUserID column if present, are actually free text and you can enter your name however you want it to appear
2. Script previous row expiry by setting the EndDateTime to the same as the StartDateTime value of the new row (if new trades are being inserted for the first time then obviously there will be no previous row to expire)
3. Save the script in a new subfolder in the following location:   
   L:\Public\IT\Development\Tickets\Monday  
   The new subfolder can be named with the ticket number from the Monday.com board. The script will be used later for the live release.
4. Execute this script, making sure you are connected to the UAT server (MHGSQL01\TGSLTEST)
5. Proceed to the [Test the Scheme in TGSL](#_Test_the_Scheme) section

# Changing an old style scheme

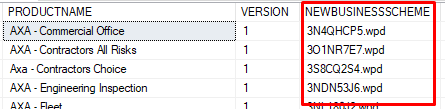
If trade and rate tables are not found in the Calculator database, then they are likely to exist in the Transactor\_Live database, but the table names are less standardised.

## Search for tables in database

1. Launch SQL Server management Studio and connect to **MHGSQL01\TGSLTEST**
2. Expand the **Transactor\_Live** database
3. Right-click on the **Tables** folder and select **Filter > Filter Settings**
4. Enter part of the insurer name, for example “axa”, in the name filter and click **OK**:  
   
5. Expand the Tables folder and it will now be filtered. Look for trade and rate tables for the Scheme you need to change. For example if the change is for the AXA Tradesman Liability Scheme you can see there are tables for this:  
   

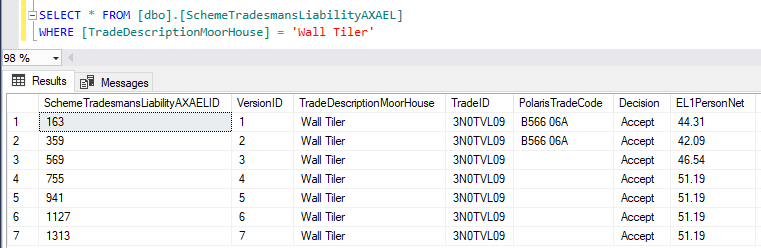
## Find tables using WPD

If unable to locate tables in the database using the method above, you will need to open the WPD file to find out what procedures, views or tables it uses.

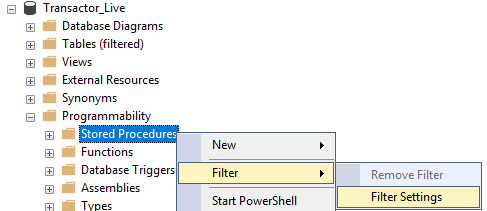
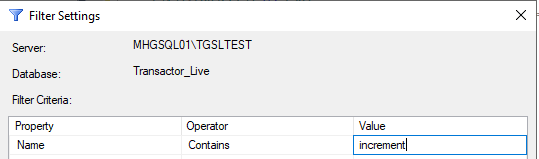
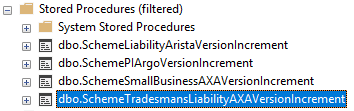
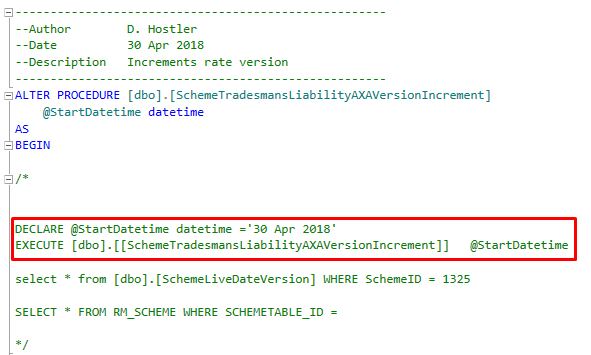
1. Still within the **Transactor\_Live** database, run the following query, changing the insurer name as required:  
   SELECT \* FROM SYSTEM\_SCHEME\_DEFINITIONS WHERE PRODUCTNAME LIKE 'Axa%'  
   This will show the names of the WPD files for the various Schemes:  
   
2. Launch File Explorer and navigate to L:\Test\TCAS\Schemefiles
3. In a text editor, open the WPD file for the Scheme that needs to be changed
4. Search the file for “ApplyTable”. There may be more than one occurrence of this. These will show you the database objects that the WPD invokes to return a premium, e.g.:  
   

The WPD may read directly from a table or it may call a stored procedure or view, as in the example above. If a stored procedure or view is shown, you will need to locate these objects in the database and look at the scripts to find the underlying tables that will need to be updated.

## Versioning

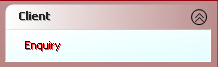
Having located the trade or rate table that needs to be updating, check whether it has a VersionID column. For example, the following table shows seven versions of the record for the Wall Tiler trade:

Sometimes there is no difference in any of the other columns from one version to the next, but because changes have been made to other trades, the entire Scheme gets a new version number and all rows are replicated. If your scheme uses a version number, you will need to do likewise.

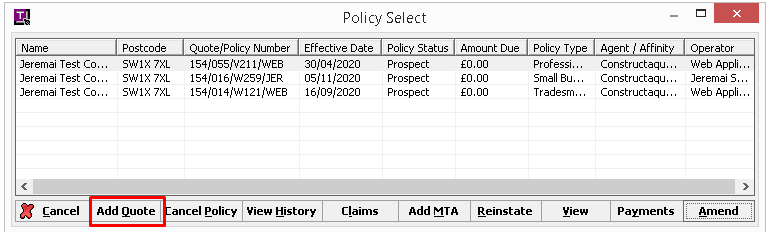
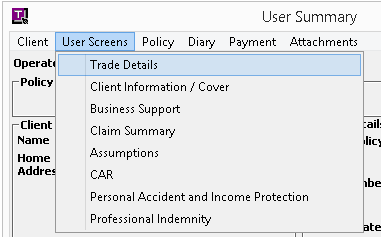
1. Still within the **Transactor\_Live** database, expand the Programmability folder
2. Right-click on the **Stored Procedures** folder and select **Filter > Filter Settings**  
   
3. Enter “increment” in the name filter and click **OK**:  
   
4. Expand the Stored Procedures folder and it will now be filtered. Look for a version increment procedure for the Scheme you need to change. For example if the change is for the AXA Tradesman Liability Scheme you can see there is a procedure for this:  
   
5. Right-click on the stored procedure and click **Modify**. The procedure script will open in a new query window.
6. Copy and the execution code from the test harness near the top of the procedure (if present), e.g.:  
   
7. Paste this into a new query window. If there is no test harness, manually type the execution statement.
8. Set the start date parameter to the current date (at run time, since the live release may happen days or weeks later)
9. Save the script in a new subfolder in the following location:   
   L:\Public\IT\Development\Tickets\Monday  
   The new subfolder can be named with the ticket number from the Monday.com board. The script will be used later for the live release.
10. Execute this script, making sure you are connected to the UAT server (MHGSQL01\TGSLTEST)
11. Add to the script to update the rows that have been created under the new version number with the required changes according to the ticket. Trade and rate changes are usually done in bulk, e.g. increasing the rates across all trades. This may require loading up a spreadsheet into a staging table and referencing this table in the script. The staging table can be created in the StagingTables database.
12. Execute the updates, making sure you are connected to the UAT server (MHGSQL01\TGSLTEST)
13. Save the changes to the script
14. Proceed to the [Test the Scheme in TGSL](#_Test_the_Scheme) section

If the Scheme does not use versioning then the updates can be made by inserting new rows with a new StartDateTime and expiring existing rows, in the same way as the instructions for [Changing a new style calculator](#_Changing_a_new) above.

# Test the Scheme in TGSL

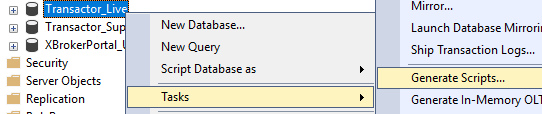
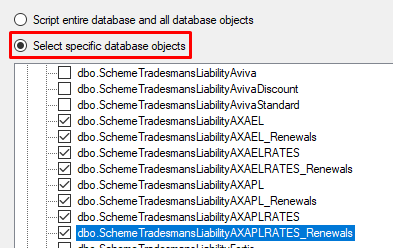
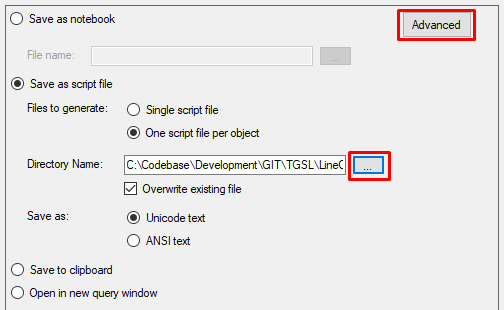
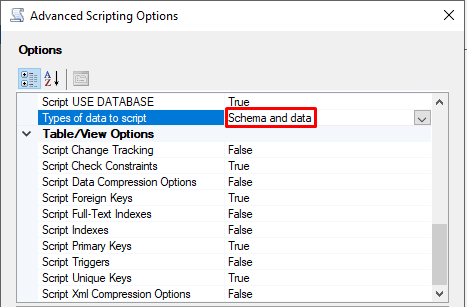
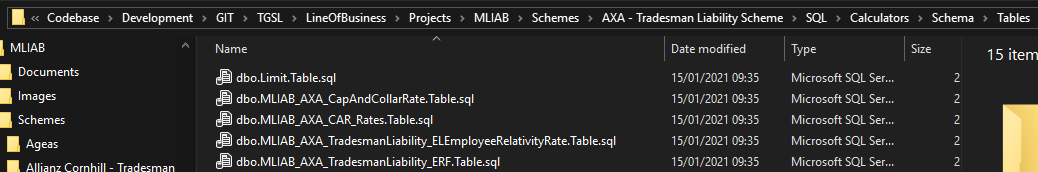
1. Launch a remote desktop connection to MHGTGSL01Test
2. Click the start button and search for TCAS, then click to open:  
   
3. Log in
4. Perform an enquiry to find a suitable test customer:  
   

Client Search 
Search By 
O 
Policy/Quote Number 
Surname and/or Postcode 
• 
O 
Surname and Renewal Date 
O 
Surname and Date of Birth 
O 
Registration Number 
O 
Invoice Number 
O 
Client Reference 
O 
Company Name and/or Postcode 
O 
Insurer Policy Number 
O 
Claim Number 
O 
Telephone Campaign 
Sumame/Co_ Name Clements 
Postcode CFB3 2RA 
Cancel 

1. Once you've found a client then click **Policies**. If you already have a Policy Type for the appropriate Product click **Amend** and skip the next step. Otherwise click **Add Quote**:  
   
2. Select the Agent and Product from the Product Selection screen and click **OK**.
3. Fill in the quote screens, clicking **OK** or **Next** to progress through the screens. If you are amending an existing quote, the previously filled in screens can be accessed from the **User Screens** menu on the User Summary screen:   
     
   Ensure you enter information that will allow you to test the changes that have been made, e.g. trade and level of indemnity.
4. Upon reaching the User Summary screen, click **Quote** and then **OK**.
5. The Scheme should appear in the list of quotes. If the Refer/Decline outcome column shows ‘Refer’ or ‘Decline’, double-click this row and check the messages for the reason, then go back and correct the risk information on the user screens to return a quote.
6. Click on the **Breakdown** button to view the premium sections, and check that the premiums have been calculated according to the rates or other changes that have been made.
7. Respond to the ticket on the Monday.com board, advising that changes are now ready for testing.

# Release to live TGSL

After the ticket is approved for live release:

1. Launch SQL Server management Studio and connect to **MHGSQL01\TGSL** (live)
2. Open the previously saved release script
3. Change the database connection to the **Transactor\_Live** or **Calculators** database depending on whether you are updating an old or new style Scheme:  
   master 
   FastCodeLive 
   FastCodeUAT 
   FleetSystem_Live 
   IdentityUsers 
   Marketing 
   master 
   model 
   msdb 
   Product 
   ProductTemp 
   Rule8asedPremiumAdjustment 
   Shared 
   StagingTabIes 
   tempdb 
   Transactor Live 
   Transactor Live Archive 
   Transactor_Support 
4. Execute the script
5. Respond to the ticket on the Monday.com board, advising that changes are now live.
6. Export table creation and data population scripts for all tables that have been updated. Within SQL Server Management Studio, right-click on the appropriate database and click **Tasks > Generate Scripts…**:  
   
7. Click the radio button for ‘Select specific database objects’, expand the Tables list and place a check mark against all tables that have been updated, e.g.:  
   
8. Click **Next**
9. Choose ‘Save as script file’ and ‘One script file per object’:  
   
10. Click the **Advanced** button highlighted above
11. Scroll down almost to the end of the Advanced Scripting Options window that appears and change the **Types of data to script** option to ‘Schema and data’ so that the new records are added to the script:  
    
12. Click **OK**
13. Click the ellipsis button outlined above and choose the Git location for the Scheme objects, i.e. GIT\TGSL\LineOfBusiness\Projects\<Product Code>\Schemes\<Scheme name>\SQL  
    Note some Scheme folders may be set up differently and have separate subfolders for tables, views, procedures, etc, for example:  
    
14. Click **Next**, then **Next** again, then **Finish**
15. Check that the scripts have been created in the specified folder
16. Commit the changes in Git and push to the Github repository