Creating a new Scheme

Contents

[Introduction 2](#_Toc171323861)

[Scheme Components 2](#_Toc171323862)

[WPD File 2](#_Toc171323863)

[Standard database objects 2](#_Toc171323864)

[High level diagram 3](#_Toc171323865)

[Set up Scheme Folder 3](#_Toc171323866)

[Check Product Type ID / Backup and Restore Databases 4](#_Toc171323867)

[Creating a new Scheme 4](#_Toc171323868)

[Create a working folder 4](#_Toc171323869)

[TGSL LOB Scheme Builder steps 4](#_Toc171323870)

[Create Scheme in Transactor\_Dev database and update SchemeTableID 6](#_Toc171323871)

[Update and re-un TGSL LOB Scheme Builder 7](#_Toc171323872)

[Customise the Calculator Scripts 8](#_Toc171323873)

[Release and test the Calculator 9](#_Toc171323874)

[Release to UAT TCAS 10](#_Toc171323875)

[Test Constructaquote.com web site steps 11](#_Toc171323876)

[Product database steps 11](#_Toc171323877)

[Relationship Manager steps 11](#_Toc171323878)

[Release to Live TCAS 13](#_Toc171323879)

[Release to Live Constructaquote.com web site 14](#_Toc171323880)

[Troubleshooting 14](#_Toc171323881)

[Debugging Calculator procedures 14](#_Toc171323882)

[Scheme not quoting on Constructaquote.com 14](#_Toc171323883)

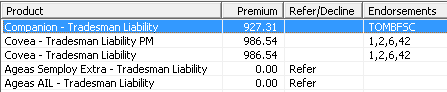
[Appendices 14](#_Toc171323884)

[Appendix I: XBroker commission insert example script 14](#_Toc171323885)

[Appendix II: Product database scheme insert example script 15](#_Toc171323886)

# Introduction

A Product (Line of Business) and an insurer together make a Scheme. For example, when quoting a prospective customer for a Tradesman Liability policy (this is the Product) in TGSL, results will be returned for multiple insurers. Each of these is a separate Scheme:



A calculator for each Scheme processes the risk information (level of cover, trade, number of employees, etc.) against the rules and rates provided by each insurer to return the quoted premium, premium breakdown, referral messages, decline messages, excesses, endorsements and a summary of the policy details.

As shown above, a Scheme name conventionally consists of the insurer name, followed by a hyphen, followed by the Product name.

# Scheme Components

## WPD File

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. This is an extremely verbose language which is difficult to code, read, and debug.

The early calculators were written using the TCAS product modeller tool which outputs a .wpd file. The rates tables were kept within the Transactor\_live database. Many calculators are still in this format, and a guide to the syntax can be found on Twiki:

<http://intranet/twiki/bin/view/Development/TGSLCalculatorSyntax>

Each new scheme still requires 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, and receiving the outputs (premium, referral messages, etc.) in the required format.

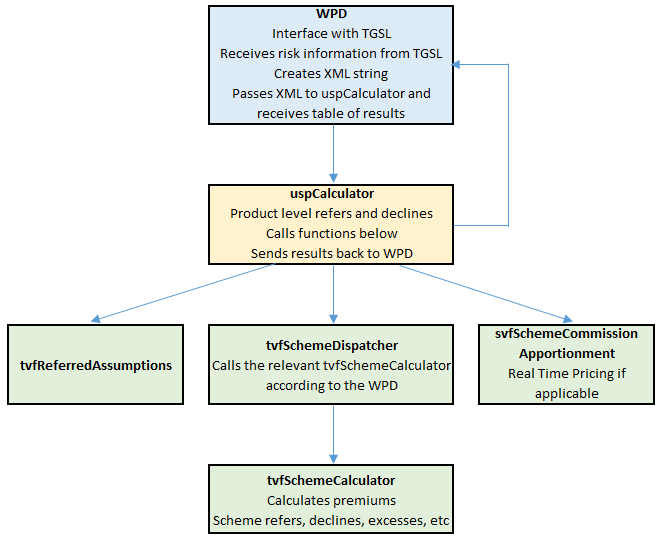
## Standard database objects

The following describes the flow of processing through the standard database objects that will be created or updated for the new Scheme.

The WPD passes risk XML to **Calculators.dbo.<product code>\_uspCalculator** which does the following:

* Converts the XML to variables and tables
* Calls function **Calculators.dbo.<product code>\_tvfReferredAssumptions** and returns a referral message if any of the assumptions in the question set have not been agreed to
* Returns Product level referral, summary, and decline messages, i.e. those that apply regardless of the Scheme
* Calls function **Calculators.dbo.<product code>\_tvfSchemeDispatcher**Top of FormTop of Form
  + The SchemeDispatcher calls the relevant function **Calculators.dbo.<product code>\_<scheme name>\_tvfCalculator** depending on the ScemeTableID passed in
  + May also call additional calculator functions depending on the Product. For example, Tradesman Liability calls Personal Accident and Professional Indemnity functions.
    - The calculator function calculates the premiums, refers, declines, breakdowns, excesses, endorsements, summaries, and product details. There may be additional functions called for these depending on the Scheme, e.g. **<product code>\_<scheme name>\_tvfExcess** and **…tvfTrades**.
* Calls procedure **Calculators.dbo.uspSchemePolicyDetailsInsert** to save the calculated details to **Calculators.dbo.SchemePolicyDetails** table (depending on the Scheme)
* Calls function **Calculators.dbo.svfSchemeCommissionApportionment** to apply rule based premium / commission adjustments (known as ‘Real Time Pricing’) if applicable
* Passes the results back to the WPD

## High level diagram



# Set up Scheme Folder

This assumes you are using Git and have the TGSL repository cloned locally.

1. Create a new project folder in the following location:  
   …GIT\TGSL\LineOfBusiness\Projects\<product code>\Schemes  
   Name the folder <InsurerName> - <Line Of Business Name> e.g. “Tokio Marine HCC - Tradesman Liability”
2. Create sub folders called “Documents” and “SQL”
3. A new Scheme specification should be received from Operations, containing the rates and rules that apply. Save this specification in the Documents folder.

# Check Product Type ID / Backup and Restore Databases

Each Scheme is assigned a Scheme Table ID in TGSL. It is important that this ID is the same between Dev, UAT and Live databases because the ID gets hard coded into files that are produced by the TGSLLOBSchemeBuilder. This will cause errors when releasing the scheme from Dev to UAT or from UAT to Live if the ID is different.

It is best practise to restore the Dev and UAT Transactor\_Live databases from a backup of Live before beginning any new project. This is especially important if the IDs are out of sync. To check the current highest ID that is in use, launch SQL Server management Studio and connect to **MHGSQL01\TGSLDEV**. Run the following query in the Transactor\_Live database:

SELECT MAX([SCHEMETABLE\_ID]) FROM [dbo].[RM\_SCHEME]

Then connect to both **MHGSQL01\TGSLTEST** and **MHGSQL01\TGSL** and repeat the query.

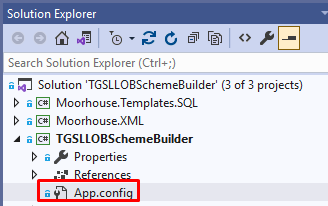
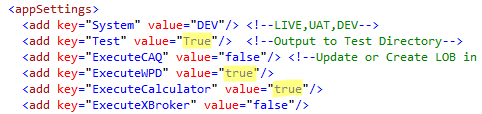
# Creating a new Scheme

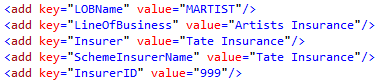
## Create a working folder

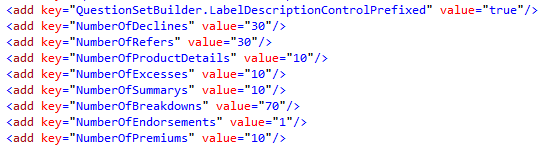
The LOB Scheme Builder steps below will produce a number of files, some of which may not be needed, and which may be overwritten if the builder needs to be run multiple times. Create a working folder where these files can be output to. This could be a local folder on your developer desktop, or a folder with the number of the Monday.com ticket in L:\Public\IT\Development\Tickets\Monday.

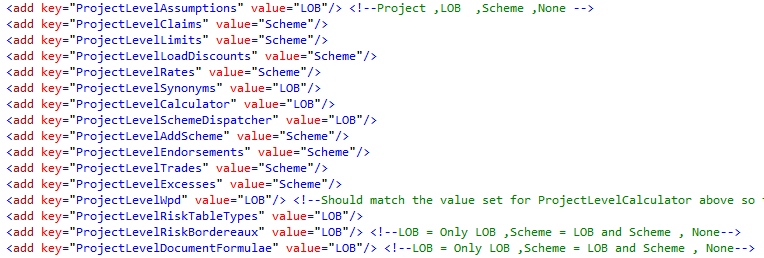
## TGSL LOB Scheme Builder steps

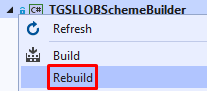
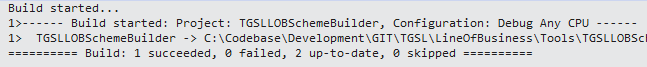
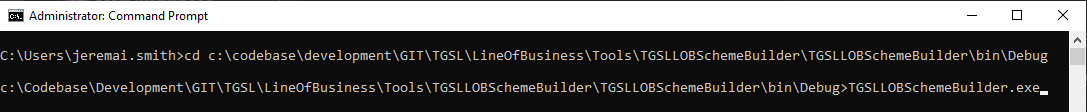
This assumes you are using Git and have the TGSL repository cloned locally.

1. From your developer desktop, launch Microsoft Visual Studio
2. Open the solution GIT\TGSL\LineOfBusiness\Tools\TGSLLOBSchemeBuilder\ TGSLLOBSchemeBuilder.sln
3. From the Solution Explorer on the right of the screen, double click **App.config** under TGSLLOBSchemeBuilder to open it:  
   
4. Set System to **Test, ExecuteWPD** and **ExecuteCalculator** to true, ensuring the other Execute… options are set to false:  
   
5. The **ProjectRootPath** should be set to the location of the Screen Designer project files. If this is the first Scheme for a new Product, the files may only be in the Dev folder location. Set the **Testpath** to the location of the working folder created earlier:  
   
6. Ensure the Line of Business and Insurer options are set accordingly. The Insurer ID is the primary key from the LIST\_INSURER table in the Transactor\_Live database:



1. Note, the **SchemeTableID** can be ignored for now as the Scheme doesn’t yet exist in the Transactor\_Live database. We will come back and update this later.  
   Set **SchemeLinkAgents** to a comma separated list of the Agent IDs that the Scheme needs to be linked to. These should be documented in the Scheme specification document and will commonly be Constructaquote, Constructaquote.com, and Moorhouse, plus XBroker if applicable. The Agent IDs are the primary key from the RM\_Agent table in the Transactor\_Live database:  
   
2. Set the **CommissionPercent** according to the Scheme specification document:
3. Set the **RangePrefix** that will be used in creating Policy numbers. This should be provided in the specification but usually consists of a two letter code for the insurer followed by three letters for the Product. The **wpdFileName** option sets the name of the WPD file that can be created. This can be set as desired, but it makes sense to use the same code as the **RangePrefix** followed by 001 for the first version of the WPD:  
   
4. Set **RateStartDateTime** to the start date of the Scheme rates. For a new scheme this doesn’t matter too much as there will be no existing rates. For testing purposes, use today’s date:  
   
5. Leave the following options set to the default values:  
   
6. Set the **PremiumSections** value is to the sections that are required for this Product (multiple values can be separated with a comma):  
   
7. Set the **ProjectLevel…** options to “Scheme” or “LOB” according to the examples below:

  
Selecting “Scheme” will output the scripts and the database objects they create with the Scheme name as part of the object name. Selecting “LOB” will create a single object that is used across all schemes for the Product. These options can be adjusted depending on the specification of the Scheme being created.

1. If this is the first calculator to be created for a Line of Business, set the RealTimePricingApplies value to true if the Line of Business will use Real Time Pricing. This will cause the <product code>\_uspCalculator script to include a call to the svfSchemeCommissionApportionment function.  
     
   If there are already calculators for this Line of Business, then the uspCalculator procedure will already exist and so this setting is redundant.
2. Right-click on TGSLLOBSchemeBuilder in the Solution Explorer and click **Rebuild**:  
   
3. The output window will show if the build succeeded (this may auto hide if not pinned):  
   
4. Launch Command Prompt and change directory to your local path for the following folder: GIT\TGSL\LineOfBusiness\Tools\TGSLLOBSchemeBuilder\TGSLLOBSchemeBuilder\bin\Debug  
   Then type or paste TGSLLOBSchemeBuilder.exe and press Enter to run it:  
   

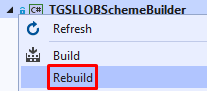
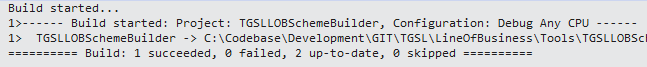
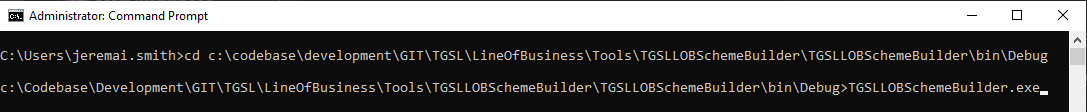
Note, you can also double-click to run TGSLLOBSchemeBuilder.exe from within File Explorer, but this method does not allow you to read any error messages returned.

1. Check that the scripts have been output to the working folder location that was chosen earlier.

## Create Scheme in Transactor\_Dev database and update SchemeTableID

1. Create a subfolder in GIT\TGSL\LineOfBusiness\Projects\<product code>\Schemes\<scheme name>\SQL called “Transactor\_Live”. This will hold scripts for release to the Transactor\_Live database.
2. Copy the output script <product code\_scheme>\_AddScheme.Script.sql from the output folder you specified and paste it into GIT\TGSL\LineOfBusiness\Projects\<product code>\Schemes\<scheme name>\SQL\Transactor\_Live
3. Open the script and adjust the commission percentages according to the Scheme specification if necessary. The Scheme creation procedure only creates a single row in the RM\_COMMISON table, so:
   1. If the Scheme is linked only to direct agents (e.g. Constructaquote, Constructaquote.com, Moorhouse Gorup) then the three partner commission percentage fields (NB\_Partner\_Percent, MTA\_Partner\_Percent and REN\_Partner\_Percent) should be updated to the same amount
   2. If the Scheme is linked only to indirect agents that sell through brokers (e.g. XBroker), the three partner commission percentage fields should be updated as above, the three agent commission percentage fields (NB\_Agent\_Percent, MTA\_Agent\_Percent and REN\_Agent\_Percent) should be updated with the agent commission amounts, and the three subagent commission percentage fields (NB\_SubAgent\_Percent, MTA\_SubAgent\_Percent and REN\_SubAgent\_Percent) should be updated with the subagent commission amounts.
   3. If the Scheme is linked to both direct and indirect agents, only update the partner commission percentage fields on the main record as per point a above. Then copy and paste the script from [Appendix I](#_Appendix_I:_XBroker) into the AddScheme script. This will create an additional row in the RM\_COMMISSION table for the agent and subagent commission rates. Amend the percentages and Scheme name where highlighted.
4. Check that the @InternetAvailable parameter is set to 1 if the scheme is to be available on the Constructaquote.com web site.
5. Launch SQL Server management Studio and connect to **MHGSQL01\TGSLDEV**. Run the script in the Transactor\_Live database.
6. Run the following query, changing the name highlighted in yellow to the name of the new scheme, and make a note of the SchemeTableID that has been assigned:  
   SELECT [SCHEMETABLE\_ID] FROM [dbo].[RM\_SCHEME] WHERE [Name] = 'Tate Insurance - ArtistsInsurance'

## Update and re-un TGSL LOB Scheme Builder

1. Return to the TGSLLOBSchemeBuilder in Microsoft Visual Studio
2. Update the app.config with the new SchemeTableID that has been created:  
   
3. Right-click on TGSLLOBSchemeBuilder in the Solution Explorer and click **Rebuild**:  
   
4. The output window will show if the build succeeded (this may auto hide if not pinned):  
   
5. Launch Command Prompt and change directory to your local path for the following folder: GIT\TGSL\LineOfBusiness\Tools\TGSLLOBSchemeBuilder\TGSLLOBSchemeBuilder\bin\Debug  
   Then type or paste TGSLLOBSchemeBuilder.exe and press Enter to run it:  
   

Note, you can also double-click to run TGSLLOBSchemeBuilder.exe from within File Explorer, but this method does not allow you to read any error messages returned.

This will overwrite the WPD file and Scheme Dispatcher TVF script with new versions containing the correct new SchemeTableID.

1. Copy the .wpd file to GIT\TGSL\LineOfBusiness\Projects\<product code>\Schemes\<scheme name>
2. Copy the .wpd file to L:\Dev\TCAS\Schemefiles

## Customise the Calculator Scripts

The TGSLLOBSChemeBuilder will have output scripts to create various tables, stored procedure and functions that will form the Calculator for the new Scheme.

These scripts are standard templates and will need to be customised to insert the rates, limits, etc., according to the Scheme specification.

It is impossible to provide step by step instructions as every Scheme is different. The new Scheme may have additional rules or rating factors that require additional columns and tables to be created.

If the new Scheme contains hundreds of rows for rates it may be easier to import them into a staging table and then select from the staging table to insert them into the new rate table.

Below is a list of the standard database object scripts that will be created, along with some suggestions where changes may need to be made.

Table Types

dbo.{Prefix}RiskTables.Type.sql

Synonyms (provide access to Transactor\_live tables and views from the Calculators database)

dbo.{Prefix}ListTable.Synonyms

Claims

dbo.{Prefix}tvfClaims.UserDefinedFunction.sql

Assumptions

dbo.{Prefix}Assumptions.Table.sql

dbo.{Prefix}Assumption.TableData.sql

dbo.{Prefix}tvfReferredAssumptions.UserDefinedFunction

Limits

dbo.{Prefix}Limit.Table.sql

dbo.{Prefix}Limit.TableData.sql

* Remove any unrequired limits
* Update required limit values
* Change the Insurer and LineOfBusiness text columns to SchemeTableID if preferred

dbo.{Prefix}svfLimitSelect.UserDefinedFunction

Excess

dbo.{Prefix}Excess.Table.sql

* Add any missing required excess template strings to table dbo.Excess. These take the following format:  
  Description : Value : List\_Excess\_Section.Excess\_Section\_ID : LIST\_EXCESSTYPE.ExcessType\_ID
* Populate dbo.{Prefix}Excess.Table with values and relatedIDs

Endorsements

{Prefix}List\_Endorsement.TableData.sql

* Add Insert statements for any new endorsement codes and text
* Move this script to the GIT\TGSL\LineOfBusiness\Projects\<product code>\Schemes\<scheme name>\SQL\Transactor\_Live folder as this will form part of the Transactor\_Live database release instead of the Calculators database release

Trades

dbo.{Prefix}Trades.Table.sql

dbo.{Prefix}tvfTrades.UserDefinedFunction.sql

LoadDiscount

dbo.{Prefix}LoadDiscount.Table.sql

dbo.{Prefix}LoadDiscounts.TableData.sql

dbo.{Prefix}svfLoadDiscountsSelect.UserDefinedFunction.sql

Rate

dbo.{Prefix}Rate.Table.sql

dbo.{Prefix}Rates.TableData.sql

dbo.{Prefix}svfRatesSelect.UserDefinedFunction.sql

Calculator

dbo.{Prefix}tvfCalculator.UserDefinedFunction.sql

* This is the main Calculator for the scheme that will reference most of the other objects

dbo.{Prefix}tvfSchemeDispatcher.UserDefinedFunction.sql

* This is a Product level function that calls the individual schemes Calculators. It may already exist if this is not the first scheme built for this Product, in which case copy the case statement for the new Scheme into the existing function.

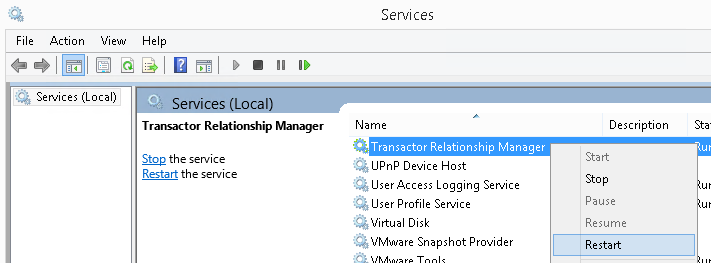
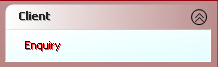
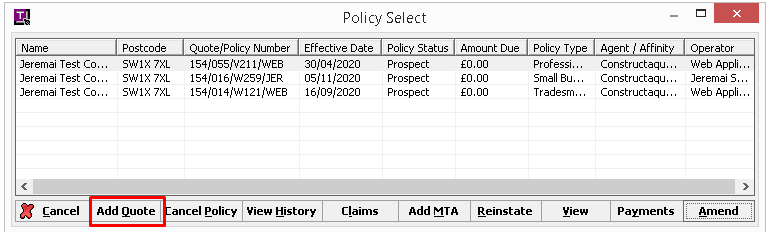
dbo.{Prefix}uspCalculator.StoredProcedure.sql

* This is a Product level function that interfaces with the WPD file. It may already exist if this is not the first scheme built for this Product.

Move all of the scripts that have been edited or are required into GIT\TGSL\LineOfBusiness\Projects\<project code>\Schemes\<scheme name>\SQL\Calculators

## Release and test the Calculator

Note, you may wish to back up and restore the Calculators database from **MHGSQL01\TGSL** to **MHGSQL01\TGSLDev** depending on how out of date it is, and whether anything else is being worked on in the Dev Calculators database.

1. In SQL Server Management Studio connect to **MHGSQL01\TGSLDEV**.
2. Run the GIT\TGSL\LineOfBusiness\Projects\<product code>\Schemes\<scheme name>\SQL\Transactor\_Live\{Prefix}List\_Endorsement.TableData.sql script in the Transactor\_Live database.
3. Run all of the scripts in saved in GIT\TGSL\LineOfBusiness\Projects\<project code>\Schemes\<scheme name>\SQL\Calculators into the Calculators database.
4. Launch a remote desktop connection to MHGTGSL01Dev.
5. Click the start button and search for Services, then click to open:  
   
6. Scroll down to the **Transactor Relationship Manager** service, right-click it and choose **Restart**:  
   
7. Click the start button and search for TCAS, then click to open:  
   
8. Log in
9. Perform an enquiry to find a suitable test customer:  
   
10. Click on **Add Quote**:  
    
11. Select the Agent and Product from the Product Selection screen and click **OK**.
12. Fill in the quote screens, clicking **OK** or **Next** to progress through the screens.
13. Upon reaching the User Summary screen, click **Quote** and then **OK**.
14. The new Scheme should appear in the list of quotes with either a premium or a Refer/Decline outcome. If it has referred or declined, double click the row and check that the refer or decline message is consistent with the Scheme specification. It is highly likely on first attempt that it will decline and show a SQL error in the decline message. See [Debugging Calculator procedures](#_Debugging_Calculator_procedures) in the Troubleshooting section.
15. Once all SQL errors are resolved, adjust the risk information to try and obtain a premium quote. If a premium is quoted, click on the **Breakdown**, **Endorsements** and **Excesses** buttons and check that all the results are correct.

# Release to UAT TCAS

Note, you may wish to back up and restore the Calculators database from **MHGSQL01\TGSL** to **MHGSQL01\TGSLTest** depending on how out of date it is, and whether anything else is being worked on in the UAT Calculators database.

When happy that the Scheme is ready for UAT:

1. Open SQL Server Management Studio and connect to **MHGSQL01\TGSLTEST**.
2. Run all of the scripts in saved in GIT\TGSL\LineOfBusiness\Projects\<product code>\Schemes\<scheme name>\SQL\Transactor\_Live into the Transactor\_Live database.
3. Run all of the scripts in saved in GIT\TGSL\LineOfBusiness\Projects\<project code>\Schemes\<scheme name>\SQL\Calculators into the Calculators database.
4. Copy the .wpd file to L:\Test\TCAS\Schemefiles.
5. Launch a remote desktop connection to MHGTGSL01Test.
6. Follow the same steps as in the [Release and test the Calculator](#_Release_and_test) section above to restart the Transactor Relationship Manager service and test the Scheme.
7. When happy, notify Ops or the creator of the ticket that the Scheme is ready for testing.

# Test Constructaquote.com web site steps

There is currently no Dev environment for the web site, so these steps apply to Test only.

## Product database steps

In SQL Server Management Studio, connect to **MHGSQL01\TGSLTEST.** Create a script to insert the Scheme into the Content.Scheme table in the Product database. An example script can be found in [Appendix II: Product database scheme insert example script](#_Appendix_I:_Product). Note that the ImportInformation field should be populated according to other schemes for the same insurer in this table. If in doubt, leave it NULL.

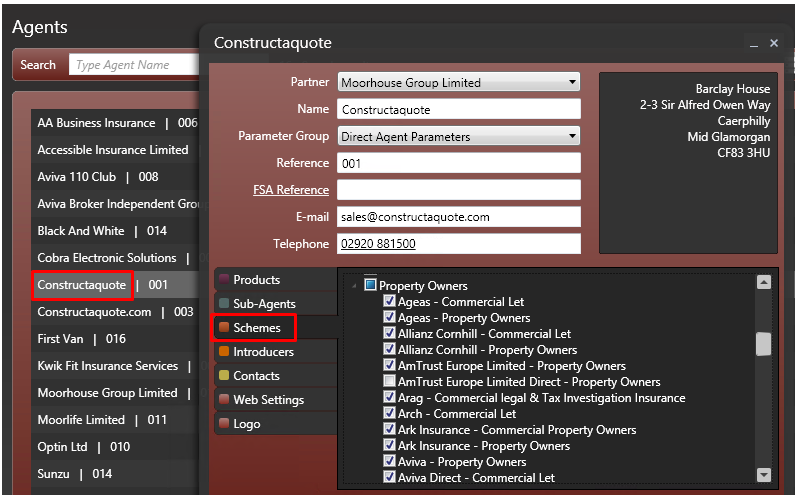
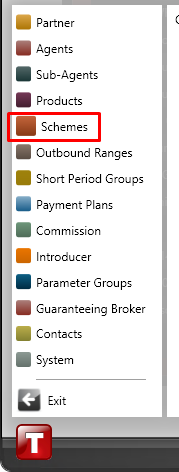
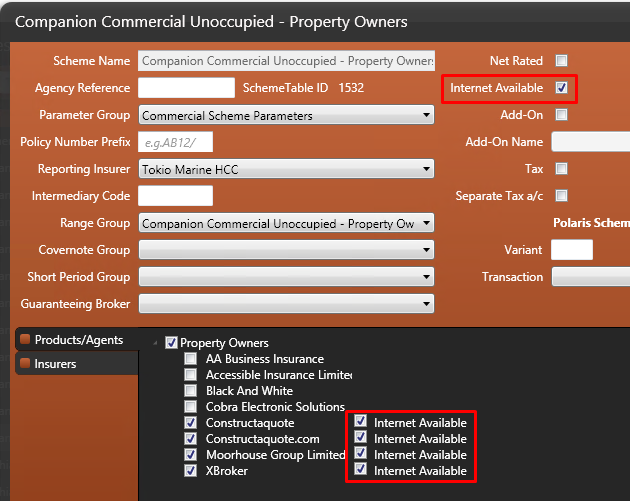
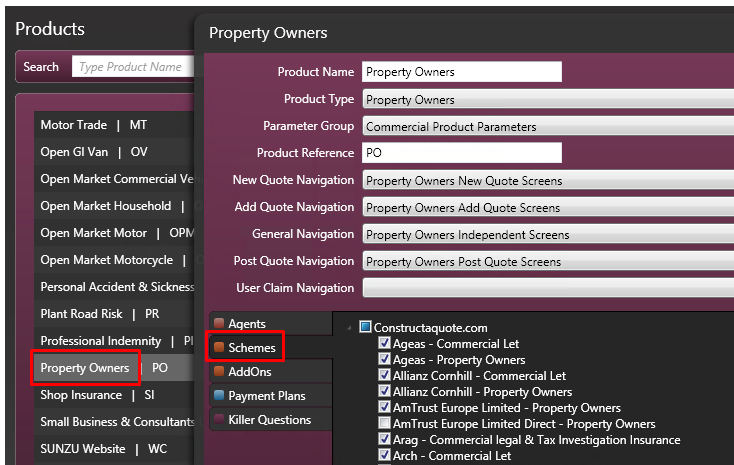
Execute the script and save it in the following folder:

GIT\TGSL\LineOfBusiness\Projects\<product code>\Schemes\<scheme name>\SQL\Product

This script will then form part of the live release.

## Relationship Manager steps

If the correct settings were used in TGSL LOB Scheme Builder / add scheme script, then the following settings should all be set correctly already.

1. Launch a remote desktop connection to MHGTGSL01Test
2. Click the start button and search for the Relationship Manger application, then open it:  
   
3. The Agent screen will be displayed. Select the Constructaquote agent, then click on **Schemes**. Expand the relevant Product type and ensure there is a check mark against the new Scheme:  
   
4. Repeat this step for other applicable agents, e.g. Constructaquote.com and XBroker.
5. Select the “T” menu button at bottom left and click **Schemes**:  
   
6. Locate the new Scheme and ensure that the main Internet Available option I checked, as well as for each agent:  
   
7. Lastly, select the “T” menu button at bottom left and click **Products**
8. Select the Product type, then click on the **Schemes**. Expand the relevant agents and ensure there is a check mark against the new Scheme under each applicable agent:  
   

# Release to Live TCAS

One tested and approved to go live:

1. Open SQL Server Management Studio and connect to **MHGSQL01\TGSL**.
2. Run all of the scripts in saved in GIT\TGSL\LineOfBusiness\Projects\<product code>\Schemes\<scheme name>\SQL\Transactor\_Live into the Transactor\_Live database.
3. Run all of the scripts in saved in GIT\TGSL\LineOfBusiness\Projects\<project code>\Schemes\<scheme name>\SQL\Calculators into the Calculators database.
4. Copy the .wpd file to L:\TCAS\Schemefiles.
5. Inform Operations that the Scheme has been released, pending a restart of the Transactor Relationship Manager service. DO NOT restart the service on any of the live servers as it will affect users using TCAS. There are two automatic restarts scheduled every day, or Operations may schedule an additional restart.
6. Commit any uncommitted changes in Git and push to the Github repository.

# Release to Live Constructaquote.com web site

1. In SQL Server Management Studio, connect to **MHGSQL01\TGSL**
2. Run all of the scripts in saved in GIT\TGSL\LineOfBusiness\Projects\<product code>\Schemes\<scheme name>\SQL\Product into the Product database.
3. Commit any uncommitted changes in Git and push to the Github repository.
4. Launch a remote desktop connection to TGSL-Farm (live servers) and open Relationship Manager. Follow the same steps as in the [Relationship Manager steps](#_Relationship_Manager_steps) section above.
5. Copy the XSLT file produced by Screen Designer from L:\Test\TCAS\Projects\<product code>\Website\stylesheets\client\read\policy\user
6. Navigate to the folders below on the live web server and paste the file (if you do not have access, contact the web developers and ask them to save the file):

\\ MHGWEB01\d$\ web\_roots\Live\TGSLWebsiteLive\quote\Stylesheets\Client\read\policy\user

# Troubleshooting

## Debugging Calculator procedures

If a quote in TGSL returns a SQL error or unexpected results, the XML string containing the risk information that was sent to the Calculator procedures can be obtained from the SchemeCommandDebug table in the Transactor\_Live database. Change the following query where highlighted to search for the name of the scheme calculator being tested:

SELECT \* FROM [dbo].[SchemeCommandDebug] WHERE [SchemeCommandText] LIKE '%MLETPROP\_uspCalculator%'

ORDER BY [QuoteDateTime] DESC

The XML string returned can then be used for debugging the Calculator stored procedures and functions within SQL Server Management Studio.

Note that the SchemeCommandDebug table grows quickly as it stores every quote. It can be cleared down periodically to make it easier to work with since it only exists for debugging and is not used for any other purpose.

## Scheme not quoting on Constructaquote.com

* Check that RM\_SCHEME.INTERNETSCHEME is set to 1 in the Transactor\_Live database on the appropriate server (MHGSQL01\TGSLTEST for UAT or MHGSQL01\TGSL for live). This should have been set when running the [Scheme creation](#_Create_Scheme_in) script.

# Appendices

## Appendix I: XBroker commission insert example script

-- Insert XBroker commission row:

INSERT INTO [dbo].[RM\_COMMISSION] (

[COMMISSION\_ID]

,[COMMISSION\_GROUP\_ID]

,[AGENT\_ID]

,[SUBAGENT\_ID]

,[EFFECTIVEDATE]

,[COMMISSIONLIMIT]

,[NB\_PARTNER\_PERCENT]

,[NB\_PARTNER\_FLAT]

,[NB\_AGENT\_PERCENT]

,[NB\_AGENT\_FLAT]

,[NB\_SUBAGENT\_PERCENT]

,[NB\_SUBAGENT\_FLAT]

,[MTA\_PARTNER\_PERCENT]

,[MTA\_PARTNER\_FLAT]

,[MTA\_AGENT\_PERCENT]

,[MTA\_AGENT\_FLAT]

,[MTA\_SUBAGENT\_PERCENT]

,[MTA\_SUBAGENT\_FLAT]

,[REN\_PARTNER\_PERCENT]

,[REN\_PARTNER\_FLAT]

,[REN\_AGENT\_PERCENT]

,[REN\_AGENT\_FLAT]

,[REN\_SUBAGENT\_PERCENT]

,[REN\_SUBAGENT\_FLAT]

,[PARTNER\_DISCOUNT]

,[AGENT\_DISCOUNT]

,[SUBAGENT\_DISCOUNT]

,[EXPIRYDATE]

,[ANYSUBAGENT]

)

SELECT

REPLACE(NEWID(),'-','') AS [COMMISSION\_ID]

,[C].[COMMISSION\_GROUP\_ID]

,'0F849A389DD4477CAF66BBCBECA49AA4' AS [AGENT\_ID] -- XBroker

,[C].[SUBAGENT\_ID]

,[C].[EFFECTIVEDATE]

,[C].[COMMISSIONLIMIT]

,[C].[NB\_PARTNER\_PERCENT]

,[C].[NB\_PARTNER\_FLAT]

,5 AS [NB\_AGENT\_PERCENT]

,[C].[NB\_AGENT\_FLAT]

,5 AS [NB\_SUBAGENT\_PERCENT]

,[C].[NB\_SUBAGENT\_FLAT]

,[C].[MTA\_PARTNER\_PERCENT]

,[C].[MTA\_PARTNER\_FLAT]

,5 AS [MTA\_AGENT\_PERCENT]

,[C].[MTA\_AGENT\_FLAT]

,5 AS [MTA\_SUBAGENT\_PERCENT]

,[C].[MTA\_SUBAGENT\_FLAT]

,[C].[REN\_PARTNER\_PERCENT]

,[C].[REN\_PARTNER\_FLAT]

,5 AS [REN\_AGENT\_PERCENT]

,[C].[REN\_AGENT\_FLAT]

,5 AS [REN\_SUBAGENT\_PERCENT]

,[C].[REN\_SUBAGENT\_FLAT]

,[C].[PARTNER\_DISCOUNT]

,[C].[AGENT\_DISCOUNT]

,[C].[SUBAGENT\_DISCOUNT]

,[C].[EXPIRYDATE]

,1 AS [ANYSUBAGENT]

FROM

[dbo].[RM\_Commission\_Group] AS [CG]

JOIN [dbo].[RM\_COMMISSION] AS [C] ON [C].[COMMISSION\_GROUP\_ID] =[CG].[COMMISSION\_GROUP\_ID]

WHERE

[CG].[Name] = 'Companion Commercial Unoccupied - Property Owners'

;

## Appendix II: Product database scheme insert example script

USE [Product]

GO

INSERT INTO [Content].[Scheme] (

[SchemeID]

,[InsurerID]

,[Name]

,[KeyProductInformation]

,[PolicyDetails]

,[ImportantInformation]

,[StartDateTime]

,[EndDateTime]

)

SELECT

[SCHEME\_ID]

,[INSURER\_ID]

,[NAME]

,NULL AS [KeyProductInformation]

,NULL AS [PolicyDetails]

,'<ul><li>Underwritten by Tokio Marine HCC</li><li>Offering speciality insurance in more than 180 countries</li><li>Part of a premier global company founded in 1879</li><li>Employs almost 3,000 industry-leading experts</li></ul>' AS [ImportantInformation]

,CONVERT(date, GETDATE()) AS [StartDateTime]

,NULL AS [EndDateTime]

FROM [Transactor\_Live].[dbo].[RM\_SCHEME]

WHERE [NAME] = 'Companion Commercial Unoccupied - Property Owners'

GO