Python and powershell scripts to prepare coinsurance (leader) statements

Barneedhar Vigneshwar

Oct 2022

Contents

Purpose of making the script

Currently, we need to download two reports from GC Core to prepare coinsurance statements where our operating office is the leader. The reports from GC Core have multiple shortfalls which require some manual intervention to get accurate reports.

Improvements made to Claims receivable register

- Issue 1: Claims receivable report provides "Name of receiver" column instead of the more useful "Name of insured". Name of the surveyor/hospital/claimant/patient clutters the report and is generally not relevant for our purpose anyway.

  Our script will replace "Name of receiver" with "Name of insured" based on the policy number available in the report.

- Issue 2: The report does not mention the policy start date or the end date. Policy period is sought by followers for their accounting purpose.

  Based on the policy number in the claims receivable report, we will add policy start date and policy end date.

- Issue 3: Claims receivable entry do not show the percentage share the follower has in that particular policy. If the 100% claim amount paid crosses 25 lakhs, the follower generally requests for claim documents for their records.

  Based on the policy number, we will include the percentage share column in the report generated by the script.

Improvements made to Premium payable register

1. The admin charges calculated by the reports does not consider the GST (at the rate of 18%) to be charged on admin charges.
2. Whererever terrorism premium is involved, the reports should provide breakup of the same. Preferably, total premium along with terrorism premium and other than terrorism premium should be present in all entries irrespective of the nature of the policy.
3. Brokerage rate is not explicitly mentioned in the reports. This makes it
4. TPA Service charges rate is not explicitly mentioned in the reports.

Other general improvements

- Issue 1: For each of the coinsurer company, we need to cut/copy and paste claims reports and premium reports into different spreadsheets and share it with them.

The script will create folders for each coinsurer company, containing claims receivable and premium payable in two different sheets of the same file. This makes the sharing process with companies a relatively smooth process, reducing mundane repetitive manual work.

- Issue 2: To verify our manually prepared reports, we need to add up the premium payable amount and claim receivable amount for each of the follower office code and cross-verify with the coinsurance settlement module.

  The script will generate a summary file for each office code showing values of premium payable and claims receivable as well as net payable/receivable amount. This amount can then be verified against the coinsurance settlement module.

C9		f_x Σ ▾ =		
	A	B	C	D
1	Follower Office Code	Net Premium payable	Claims receivable	Net Payable by UIIC
2	111202	61,499	6,87,419	-6,25,921
3	120500	10,25,419	0	10,25,419
4	970000	3,79,908	19,54,717	-15,74,810
5	Total	14,66,825	26,42,137	-11,75,311
6				
7				

Figure 1: Summary file generated by the script

Setting up the script in our local machine

Step 0

1. Ensure you are connected to the internet before we proceed any further.
2. Make a note of the bit width of your computer, whether 32-bit or 64-bit.

Step 1: Installing python

- Visit the official python website and download the 32-bit or 64-bit installer for your computer.

Or you can use the direct link below for downloading the installers:

- 32 bit
- 64 bit

After downloading the installer, select "Add path" checkbox before proceeding further with the installation.

Step 2: Disable alias

Step 3: Download python libraries

- Open powershell and run the following command to install the relevant libraries:

    pip3 install pandas openpyxl Jinja2 xlsxwriter

Note: The above steps 1 to 3 have to be done only once for each computer.

Step 4: Download the python scripts from our Github profile

- You can visit the https://github.com/jokerdino/coinsurance) and click the big green button labelled Code and scroll down to "Download Zip" button.

Direct link to the download zip file of the scripts

- After the download is over, unzip the folder and save the folder somewhere easily accessible as we will be saving reports from GC in this folder for further processing.

Downloading reports from Dashboard and GC reports

Step 5.1: Download the reports from Dashboard

1. Log in to dashboard through "Mypage login" and download the UW reports under underwriting menu. Save the file as `premium_data`.
2. From the same dashboard, save the claims report under claims menu and save that file as `claims_data`.

Step 5.2: Download reports from GC Reports

1. From Reports1 role in GC Core, download Monthly commission billl for the period you want to prepare the statements. Save this file as `commission-merge`.
2. In the reports role, download the TPA service charges register and save it as `tpa-register`.
3. In reports role, download claims receivable register and save it as `claims-receivable`.
4. In reports role, download the premium payable register and save it as `premium-payable`.

How to use the script

Step 6: Running our script

1. Open powershell and move our current directory to where the files and scripts are saved.

2. Run the following command:

   ```
   ./all.ps1
   ```

3. If you get a permission error, we shall bypass that error by the following command instead:

   ```
   powershell.exe -noprofile -executionpolicy bypass -file ./all.ps1
   ```

4. Wait for the script to do its thing.

Step 7: Checking with the settlement module

1. When the statements are generated, please open the summary file and cross verify that the amount is same as the one in settlement module.

Step 8 (optional): Generating summary file again

If there is a need to generate a new summary file, delete the existing 'Summary.xlsx' file and close the statement file. Then open powershell, `cd` to the relevant directory and run the following command:

```
python pivot-table.py
```

What to do if summary file does not tally with settlement module

If the summary generated by the script does not tally with the coinsurance settlement module, here are some of the pointers that may be checked for:

1. Uncheck if any incoming entries (both premium and claims) are present in the coinsurance settlement module.
2. Any settlement entries made in coinsurance settlement module will also be present in the claims receivable and premium payable reports. Check if there are any such entries in the report and remove them from our statement. The voucher number column can be used to identify such settement entries. After removal of such entries, you may follow Step 8 to regenerate a new summary file.
3. The script assumes that United India pays 100% of the brokerage (including followers' share). If only our own share of brokerage is paid, you may remove the brokerage amount and recalculate the "Net premium payable". **Note**: The statement generated does not have any Excel formula. So, you

would have to add formulas to "Net premium payable" column to calculate the new amount.

4. Some of the claim receivable entries may be recovery entries and our claim receivable sheet has to be updated to reflect the same.
5. Some policies issued with terrorism premium might not have corresponding brokerage. The script assumes all terrorism premium are paid brokerage at 5% rate.

The above pointers are not exhaustive and the discrepancy between the settlement module and our generated statements has to be analysed on a case-by-case basis for accuracy. If any corrections/modifications are made in the statement, save the file and close it before generating a new summary.

Making statements for multiple months at one go

Generating statements for a different period in the future

Create a new folder and name it as per your requirement. Then proceed to redownload the script as per Step 4 and unzip it to the folder just created. You may now proceed further with Step 5 onwards.

Feedback

If you have any doubts, comments, suggestions, improvements or any other feedback, you may contact me at my official email address (Employee number: 44515).