**BIAN Automatic Swagger Generation Scripts Manual**

Create Config files <SD>ModelConfig, <SD>PathConfig and <SD>ISOMapping. Samples are provided in the zip.

**Creating the <SD>ModelConfig**

* Copy the entire content of the Control Record in cell AC in API Directory Excel
* You can opt to remove lines for Service Configuration and leave everything else starting from the CR line defining the master control record.
* Please refer to the sample. Also, at the bottom of the file, add an empty line.
* Save the config file as SDNameModelConfig e.g. CustomerOfferModelConfig

**Creating the <SD>PathConfig**

The PathConfig is used to generate the API path definitions. The table below describes each config parameter and which cell in API directory excel it needs to be taken from:

|  |  |  |
| --- | --- | --- |
| **Param** | **Excel Cell** | **Example** |
| sd | Service Domain in column J | sd=Card ECommerce Gateway |
| sdpath | Asset Type in column N converted to hyphenated lowercase | sdpath=e-commerce-gateway |
| crpath | Control Record in column O converted to hyphenated lowercase | crpath=e-commerce-gateway-operating-session |
| crr | From Control Record in column AC | crr=eCommerce Gateway Service Session |
| mcr | Value of CR from Behavior Qualifiers column T | mcr=OperatingSession |
| bqs | All BQ names from column T in a single line separated by Space | bqs=Capture Batch Authentication Authorization Update Reporting |

After the above parameters comes a line called CONFIG below which everything from column U through AA that define the API operations need to be extracted and pasted as is. Example:

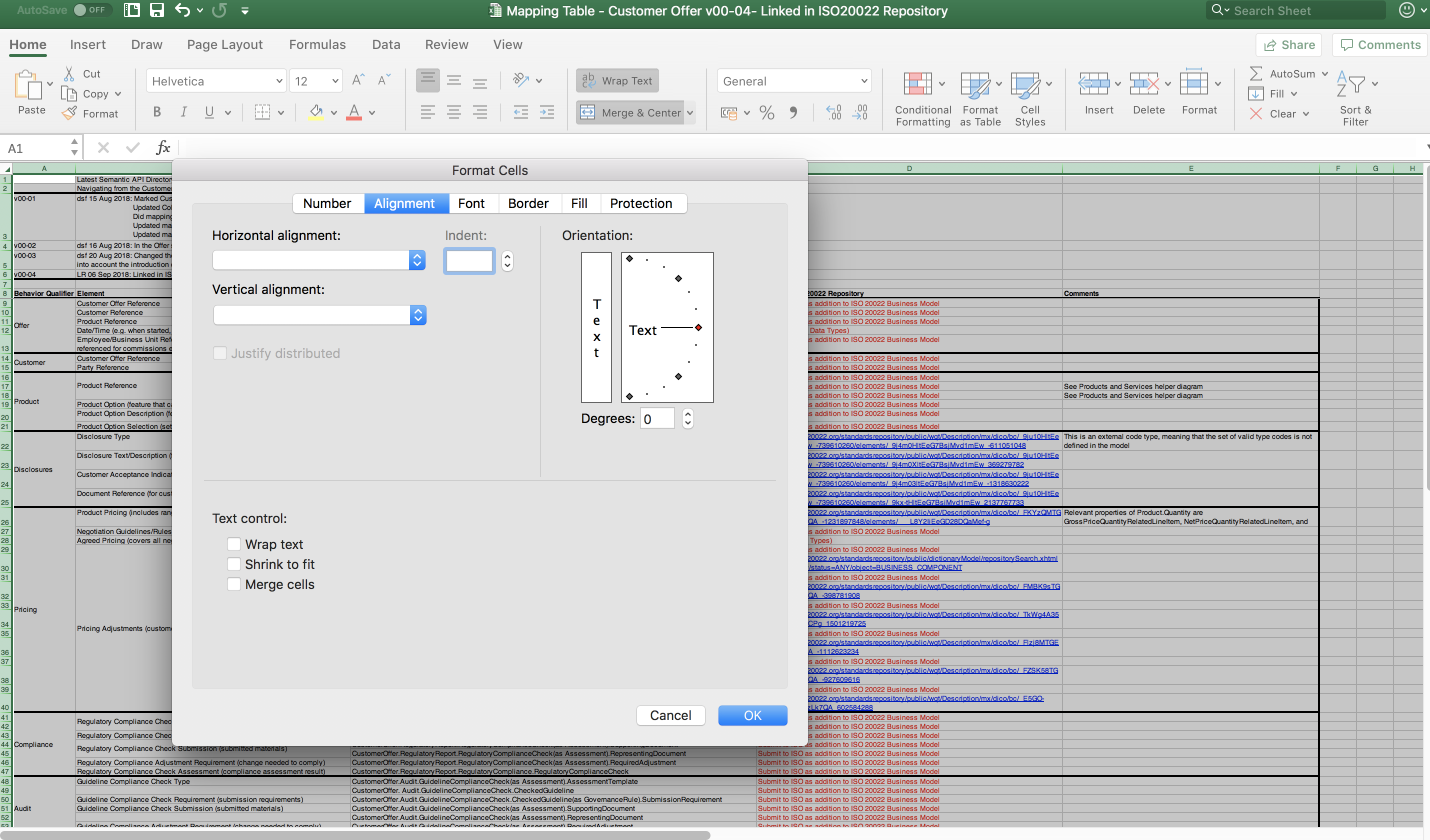


In some cases Default Action term cells are empty as they are a continuation of an action term defined in an upper cell. In such cases must fill in the empty cell with the appropriate action term. You can choose to delete rows for which we do not need operations e.g. INTERNAL action term in the example above. However, leaving them in the config will also not create any issues as the tool will ignore rows for which an Extended Service Operation and/or Possible External API are not defined.

Save the file as SDNamePathConfig e.g. CustomerOfferPathConfig

**Creating the <SD>ISOMapping file**

From the ISO Mapping spreadsheet for the SD, first un-merge cells as shown:



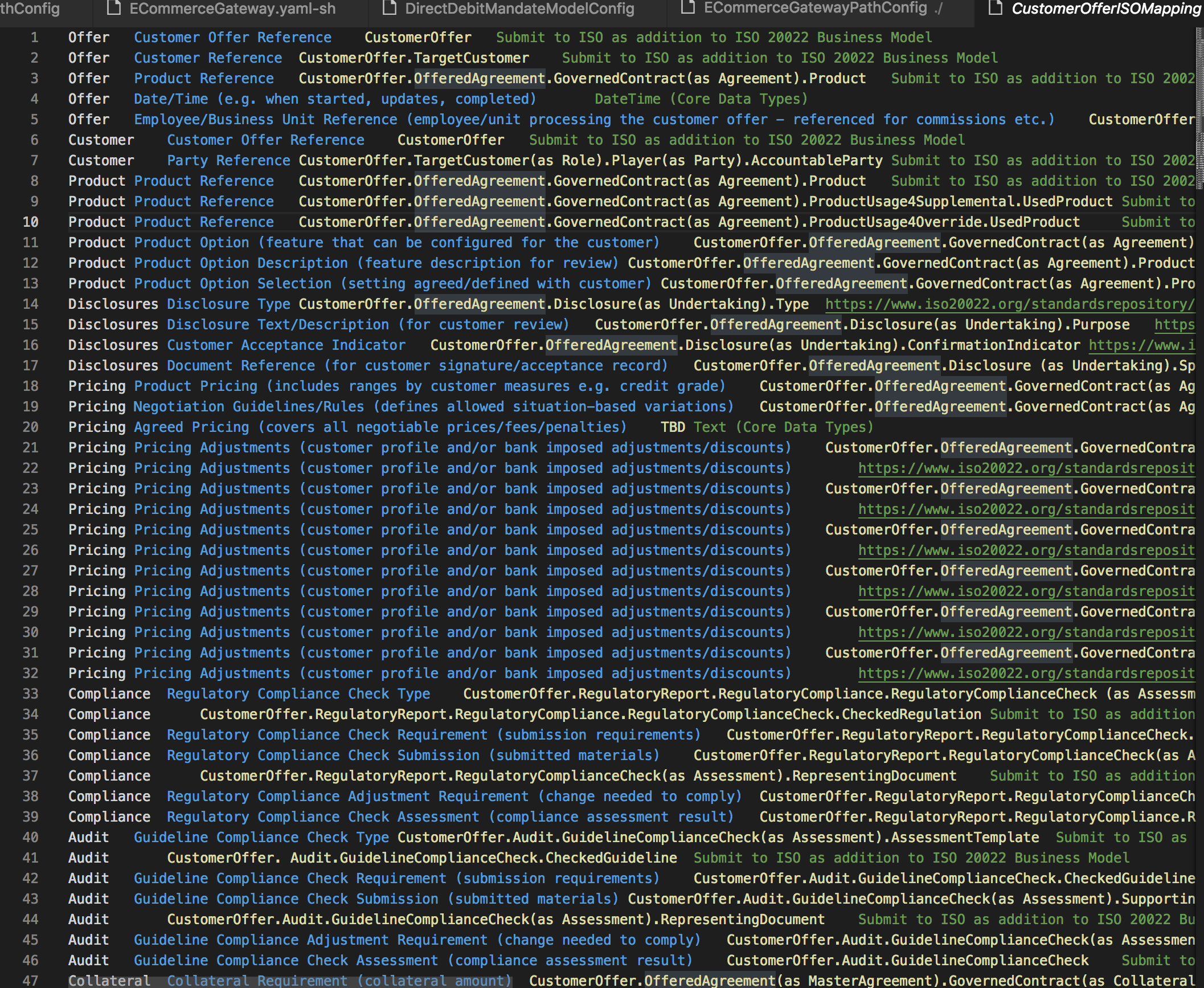
Remove rows above the mapping rows. Also the remove the row with the headings above the mapping rows.

Fill in empty cells in the excel with appropriate values. Example – the BQs are genrally merged so unmerging will create empty cells that need to be filled in with the same BQ name. Also any cell in Element, Mapping and Linkage to ISO columns need to be filled in if empty.

Remove the comments column

Copy the entire content and paste it in the file <SDName>ISOMapping e.g. CustomerOfferISOMapping

The content should look like this. Sample file provided in the zip.



If ISO Mapping is not available for an SD, no need to create the ISOMappingFile. Create only the ModelConfig and PathConfig files and script will automatically add for each attribute its BIAN core-data-type mapping. Only if ISOMapping file is created, it will read it and map to ISO mappings.

**Running the script**

Run script generateSwagger.sh in a bash shell. The script internally invokes the processModelAndISO.sh and processPath.sh. Each of these scripts can also be run independently to generate only the model definitions or only the Path definitions.



<SD>.yaml would be generated as output with all models and API definitions created.

Things to do manually on generated YAML are:

Add examples for collection filters

Pluralize BQs where necessary where BQs are part of the path

Convert “Record” Models to generic models

Tool also adds descriptions, summary and example values. Go through them and make corrections in grammar if necessary