**Constraint Analyzer Project**

**Data Collection:**

* **Location of all the files mentioned in this document:**
* **S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Data**

Needed data – Transmission outages data, Constraint data, Auction Contingency Data, Auction Mapping Document

\***For now this project is only developed for ERCOT region, once completed it can be developed for other regions as well.**

**\*This document contains information about all the file extensions except .py extension for which documentation is done in CodeFiles\_Documentation.docx file**

**Source of Historical Constraint Data:**

* Location:
* S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Constraint-Project\Data\Constraint Data
* Data is taken from the years 2014 to 2019.
* **Source - Yes Energy** is the vendor from where Historical Constraint data is pulled out.

**Source of Mapping Document:**

* Location:
* S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Constraint-Project\Data\Mapping Documents
* These mapping documents are taken from ERCOT.
* The mapping document can be taken from any year (preferably latest year’s data should be taken) as it contains the list of constraint-contingency that have happened over the period.

**Source of Transmission Outage Data:**

* Location:
* S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Constraint-Project\Data\Transmission Outages
* Data is taken from the years 2014 to 2019
* Source of data from all the years is **Yes Energy** – data pulled using the Get\_Historical\_Trans\_Outage\_v1.py
* Link to .py file - S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Constraint-Project\Code Files\Get\_Historical\_Trans\_Outage\_v1.py

**uniquePairList2014-2019.xlsx:**

* Location:
* S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Constraint-Project\Data\UniqueConstraintContingencyPair\uniquePairList2014-2019.xlsx
* This file is the output of ConstraintContingencyUniquePairCreation.py file.
* This file contains unique pairs of constraint-contingency for the years 2014-2019.
* Sheet Constraint-Contingency:
* Columns named –constraint, contingency, hrs, % Hrs, Total $, Avg $, max $, Constraint ID, Contingen. ID, Reported Name, Source Shift Factor, Sink Shift Factor, Spread Shift Factor are already present in the output obtained by python file.
* Column named – Code is created manually as follows:

1. Insert 3 columns next to ‘Constraint’ column in the excel sheet.
2. Select ‘Constraint’ column of the excel sheet.
3. Go to Data (Excel Ribbon) 🡪 Text to Column 🡪Select delimited 🡪 Click Next 🡪 Check space (uncheck all others) 🡪 Click Next 🡪 Click on Destination (in Convert Text to Columns Wizard – Step 3 of 3) 🡪 Select 4 newly inserted columns in the sheet after the ‘Constraint’ column 🡪 Click Finish
4. Pick the 3rd newly inserted column and name that column as ‘Code’. Delete the remaining columns.

* Save the excel file.
* This excel file is used as an input by ConstraintContingencyMapping.py

**uniquePairList2014-2019Changed.xlsx:**

* Location:
* S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Constraint-Project\Data\UniqueConstraintContingencyPair\uniquePairList2014-2019Changed.xlsx
* This file is the output of ConstraintContingencyMapping.py file.
* This file contains unique pairs of constraint-contingency for the years 2014-2019 in sheet ‘Constraint-Contingency’, auction mapping data of year 2019 September in ‘AuctionMapping2019SEP’ sheet and auction contingency data for the year 2019 September in sheet ‘AuctionContingency2019SEP’.
* Sheet Constraint-Contingency:
* Columns named –constraint, contingency, hrs, % Hrs, Total $, Avg $, max $, Constraint ID, Contingen. ID, Reported Name, Source Shift Factor, Sink Shift Factor, Spread Shift Factor are already present in the output obtained by python file.
* Column named – contingency\_from\_bus\_number, contingency\_from\_bus\_name, contingency\_to\_bus\_number, contingency\_to\_bus\_name, contingency\_circuit\_id, constraint\_from\_bus\_number, constraint\_from\_bus\_name, constraint\_to\_bus\_ number, constraint\_to\_bus\_name, constraint\_circuit\_id are fetched using VLOOKUP formula
* contingency\_from\_bus\_number:

=IFERROR(VLOOKUP(C3,AuctionContingency2019SEP!$A:$I,5,FALSE)," ")

* contingency\_from\_bus\_name:

=IFERROR(VLOOKUP(C3,AuctionContingency2019SEP!$A:$I,6,FALSE)," ")

* contingency\_to\_bus\_number:

=IFERROR(VLOOKUP(C3,AuctionContingency2019SEP!$A:$I,7,FALSE)," ")

* contingency\_to\_bus\_name:

=IFERROR(VLOOKUP(C3,AuctionContingency2019SEP!$A:$I,8,FALSE)," ")

* contingency\_circuit\_id:

=IFERROR(VLOOKUP(C3,AuctionContingency2019SEP!$A:$I,9,FALSE)," ")

* constraint\_from\_bus\_number:

=IFERROR(VLOOKUP(B2,AuctionMapping2019SEP!$C:$J,4,FALSE)," ")

* constraint\_from\_bus\_name:

=IFERROR(VLOOKUP(B2,AuctionMapping2019SEP!$C:$J,5,FALSE)," ")

* constraint\_to\_bus\_ number:

=IFERROR(VLOOKUP(B2,AuctionMapping2019SEP!$C:$J,6,FALSE)," ")

* constraint\_to\_bus\_name:

=IFERROR(VLOOKUP(B2,AuctionMapping2019SEP!$C:$J,7,FALSE)," ")

* constraint\_circuit\_id:

=IFERROR(VLOOKUP(B2,AuctionMapping2019SEP!$C:$J,8,FALSE)," ")

* Columns named

1. Interface Name is fetched using formula –

=IF(AND(O2<>" ",P2<>" ",Q2<>" ",R2<>" ",T2<>"",U2<>" ",V2<>" ",W2<>" "),CONCATENATE(U2,"-",W2," ","FLO"," ",P2,"-",R2),IF(AND(T2<>" ",U2<>" ",V2<>" ",W2<>" "),CONCATENATE(U2,"-",W2," ","FLO"," ","BASE"),"NaN"))

1. Element a (Contingency) is fetched using formula-

=IF(AND(O2<>" ",P2<>" ",Q2<>" ",R2<>" "),CONCATENATE("BRANCHOPEN"," ",O2," ",Q2," ",S2),"NaN")

1. Element b (Monitored) is fetched using formula –

=IF(AND(T2<>" ",U2<>" ",V2<>" ",W2<>" "),CONCATENATE("BRANCH"," ",T2," ",V2," ",X2),"NaN")

1. Meter Far is given a value – “NO”
2. Weight is given a value – “1”

* Copy the formula to all the cells
* Save the excel file.
* This file is used as an input by the ConstraintContingencySeparation.py

**uniquePairList2014-2019Separation.xlsx:**

* Location:
* S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Constraint-Project\Data\UniqueConstraintContingencyPair\uniquePairList2014-2019Separation.xlsx
* This file is the output of ConstraintContingencySeparation.py file.
* Sheet ‘mappedConstraints’ – all the columns are already present in the output obtained by python file.
* Sheet ‘mappedContingency’ - all the columns are already present in the output obtained by python file.
* Sheet ‘unmappedConstraints’ - all the columns are already present in the output obtained by python file except ‘match’ which is created as follows:
* Select ‘Constraint’ column of the excel sheet.
* Go to Data (Excel Ribbon) 🡪 Text to Column 🡪Select delimited 🡪 Click Next 🡪 Check space (uncheck all others) 🡪 Click Next 🡪 Click on Destination (in Convert Text to Columns Wizard – Step 3 of 3) 🡪 Select 3 newly columns in the sheet after the last column 🡪 Click Finish
* Pick the 3rd newly inserted column and name that column as ‘match’. Delete the remaining columns.
* Sheet ‘unmappedContingency’ - all the columns are already present in the output obtained by python file.
* Sheet ‘mappedBoth’ - all the columns are already present in the output obtained by python file.
* Save the excel file.
* This file is used as an input by the UnmappedConstraintsApproximateStringMatching.py file.

**UnmappedConstraintsApproximateStringMatching.xlsx:**

* Location:
* S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Constraint-Project\Data\UniqueConstraintContingencyPair\uniquePairList2014-2019Separation.xlsx
* This file is the output of UnmappedConstraintsApproximateStringMatching.py file.
* Manual processing carried out in this file is as follows:
* Create a new sheet named ‘manual\_mapped’
* Copy ‘constraint’, ‘match’, ‘matching\_code’ to the new sheet.
* Manually check strings separated by ‘\*’ in the matching code column with the match column for similarity.
* If they mostly match, check for the from and to bus name in the auction mapping document of the year 2019 September and compare it with first part of the ‘constraint’ column.
* If all the checks are correct or mostly match keep the row and delete the rows that don’t match.
* In the matching code column, keep only that string which matches the most removing the ‘\*’ as well.
* In the sheet named “All” create a column named ‘matching\_code\_mapped’ which is filled using this formula: =IFERROR(VLOOKUP(D2,manual\_mapped!$A:$C,3,FALSE)," ")
* Sheet named ‘unmapped’ consist the constraints that remained unmapped after manual mapping. Filter was applied in the ‘All’ sheet and then the data obtained was copy pasted into ‘unmapped’ sheet.
* Sheet named ‘total\_$\_analysis’ contains pivot table to analyze in what total\_$ range does the unmapped constraints mostly fall.
* Sheet ‘AuctionMappingSEP2019’ is copied from “S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Constraint-Project\Data\Mapping Documents\2019.SEP.Monthly.Auction.MappingDocument.xlsx”
* This file is used as one of the inputs in the ConstraintContingencyFinal.py file.

**ConstraintContingencyFinal.xlsx:**

* Location:
* S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Constraint-Project\Data\UniqueConstraintContingencyPair\Constraint-ContingencyFinal.xlsx
* This file is the output of ConstraintContingencyFinal.py file.
* Sheet ‘mappedConstraints’ contains final list of mapped constraints
* Sheet ‘unmappedConstraints’ contains final list of unmapped constraints
* Sheet ‘mappedContingency’ contains final list of mapped contingencies
* Sheet ‘unmappedContingency’ contains final list of unmapped contingencies
* Sheet ‘mappedBoth’ contains final list of mapped Constraint and Contingency pair.

**Preprocessing of Transmission Outages Files:**

* These files are used as an input by the TransmissionOutagesMapping.py file.
* Location:
* Preprocessing of the files listed below must be done before using it as an input. All files are:

1. S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Constraint-Project\Data\Transmission Outages\ERCOT\_TransmissionOutage\_2014-01-01\_to\_2014-12-31.xlsx
2. S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Constraint-Project\Data\Transmission Outages\ERCOT\_TransmissionOutage\_2015-01-01\_to\_2015-12-31.xlsx
3. S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Constraint-Project\Data\Transmission Outages\ERCOT\_TransmissionOutage\_2016-01-01\_to\_2016-12-31.xlsx
4. S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Constraint-Project\Data\Transmission Outages\ERCOT\_TransmissionOutage\_2017-01-01\_to\_2017-12-31.xlsx
5. S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Constraint-Project\Data\Transmission Outages\ERCOT\_TransmissionOutage\_2018-01-01\_to\_2018-12-31.xlsx
6. S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Constraint-Project\Data\Transmission Outages\ERCOT\_TransmissionOutage\_2019-01-01\_to\_2019-07-24.xlsx

* Following step are to be performed on all the files mentioned above:

1. Select ‘reported name’ column of the excel file.
2. Go to Data (Excel Ribbon) 🡪 Text to Column 🡪Select delimited 🡪 Click Next 🡪 Check Space (uncheck all others) 🡪 Click Next 🡪 Click on Destination (in Convert Text to Columns Wizard – Step 3 of 3) 🡪 Select 4 columns in the sheet after the last column i.e. ‘RELIABILITY\_SCORE’ 🡪 Click Finish
3. In the newly created columns, only keep column 1st and 3rd. Delete the other two columns.
4. Name the newly created 1st column as ‘from’ and 3rd column as ‘to’.

**UniqueTransmissionOutagesMappedYEAR.xlsx (where YEAR can be – 2014,2015,2016,2017,2018,2019)**

* Location:
* S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Constraint-Project\Data
* These files are output generated by TransmissionOutagesMapping.py file.
* Delete the index column in ‘Outages’ sheet, ‘AuctionMapping2019JUL\_LINES’ sheet and ‘AuctionMapping2019JUL\_AUTOS’ sheet created by the python code.
* In the ‘Outages’ sheet Columns named – iso, facility, fromstation, tostation, kv, fromzone, tozone, facility\_type, type, type\_detail, status, status\_detail, startdate, enddate, planned\_startdate, planned\_enddate, open\_close, ticketid, facilityid, lastchangedate, publishdate, reported\_name, fromstationid, tostationid, reliability\_score, reliability\_accuracy, from, to are all fetched by the python code file itself.
* In the AuctionMapping2019JUL\_AUTOS sheet, insert new columns as follows:

1. Insert 3 blank columns after ‘operations\_name’.
2. Select ‘operations\_name’ column of the excel sheet.
3. Go to Data (Excel Ribbon) 🡪 Text to Column 🡪Select delimited 🡪 Click Next 🡪 Check Other – enter underscore symbol (uncheck all others) 🡪 Click Next 🡪 Click on Destination (in Convert Text to Columns Wizard – Step 3 of 3) 🡪 Select 3 columns in the sheet inserted after column i.e. ‘operations\_name’ 🡪 Click Finish
4. Name the 1st column as ‘Operations’ and delete the rest.
5. Insert a column next to ‘Operations’ named ‘Combine’ and use this formula =C2&"|"&E2

To obtain the values for this column.

* In the ‘Outages’ sheet Columns which are created manually other than the one present originallyafter executing python code to obtain this file are:

1. combine: formula used for obtaining column value

=AA2&"|"&AB2

1. from\_bus\_number: formula used for obtaining column value =IF(H2="XFMR",IFERROR(VLOOKUP(AC2,AuctionMapping2019JUL\_AUTOS!$D:$J,3,FALSE)," "),IFERROR(VLOOKUP(AB2,AuctionMapping2019JUL\_LINES!$C:$J,4,FALSE)," "))
2. from\_bus\_name: formula used for obtaining column value =IF(H2="XFMR",IFERROR(VLOOKUP(AC2,AuctionMapping2019JUL\_AUTOS!$D:$J,4,FALSE)," "),IFERROR(VLOOKUP(AB2,AuctionMapping2019JUL\_LINES!$C:$J,5,FALSE)," "))
3. to\_bus\_number: formula used for obtaining column value =IF(H2="XFMR",IFERROR(VLOOKUP(AC2,AuctionMapping2019JUL\_AUTOS!$D:$J,5,FALSE)," "),IFERROR(VLOOKUP(AB2,AuctionMapping2019JUL\_LINES!$C:$J,6,FALSE)," "))
4. to\_bus\_name: formula used for obtaining column value =IF(H2="XFMR",IFERROR(VLOOKUP(AC2,AuctionMapping2019JUL\_AUTOS!$D:$J,6,FALSE)," "),IFERROR(VLOOKUP(AB2,AuctionMapping2019JUL\_LINES!$C:$J,7,FALSE)," "))
5. circuit\_id: formula used for obtaining column value =IF(H2="XFMR",IFERROR(VLOOKUP(AC2,AuctionMapping2019JUL\_AUTOS!$D:$J,7,FALSE)," "),IFERROR(VLOOKUP(AB2,AuctionMapping2019JUL\_LINES!$C:$J,8,FALSE)," "))
6. reported\_name\_duplicate: copy and paste column values of ‘reported\_name’ column.

* In the ‘AuctionMapping2019JUL\_LINES’ sheet create a new column next to the last column named ‘circuit\_id’ and name this new column as ‘outages\_reported\_names\_mapped’. The values for this column are obtained using the following formula:

=IFERROR(VLOOKUP(F2,Outages!$AD:$AI,6,FALSE)," ")

* In the ‘AuctionMapping2019JUL\_AUTOS’ sheet create a new column next to the last column named ‘circuit\_id’ and name this new column as ‘outages\_reported\_names\_mapped’. The values for this column are obtained using the following formula:

=IFERROR(VLOOKUP(F2,Outages!$AD:$AI,6,FALSE)," ")

* All these files are used as an input by the UniqueTransmissionOutagesConcat.py file.

**UniqueTransmissionOutagesMapped2014-2019.xlsx:**

* Location:
* S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Constraint-Project\Data\UniqueTransmissionOutagesMapped2014-2019.xlsx
* This file is an output file which is generated by a python code file named – UniqueTransmissionOutagesConcat.py
* Sheet – ‘All’ consists of all the transmission outages
* Sheet – ‘Mapped’ consists of mapped transmission outages
* Sheet – ‘Unmapped’ consists of unmapped transmission outages
* In this sheet apart from the columns fetched by python code file, a new column is inserted as follows:

1. Select ‘facility’ column of the excel file.
2. Go to Data (Excel Ribbon) 🡪 Text to Column 🡪Select delimited 🡪 Click Next 🡪 Check Space (uncheck all others) 🡪 Click Next 🡪 Click on Destination (in Convert Text to Columns Wizard – Step 3 of 3) 🡪 Select 4 columns in the sheet after the last column i.e. ‘type\_detail’ 🡪 Click Finish
3. In the newly created columns, only keep column 1st. Delete the other columns.
4. Name the newly created 1st column as ‘match’.

**MappedOutagesVerification.xlsx:**

* This file is created by copying sheet named ‘Mapped’ of excel file UniqueTransmissionOutagesMapped2014-2019.xlsx.
* Location:
* S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Constraint-Project\Data\Transmission Outages Verification\MappedOutagesVerification.xlsx
* Apart from the columns that are copied, ‘from\_split\_1’, ‘from\_split\_2’ are inserted as follows:

1. Insert 2 columns after ‘from’ column.
2. Select ‘from’ column of the excel file.
3. Go to Data (Excel Ribbon) 🡪 Text to Column 🡪Select delimited 🡪 Click Next 🡪 Check Others and type hyphen (-) (uncheck all others) 🡪 Click Next 🡪 Click on Destination (in Convert Text to Columns Wizard – Step 3 of 3) 🡪 Select the 2 newly inserted columns in the sheet after the column named ‘from’ 🡪 Click Finish
4. Name the newly created 1st column as ‘from\_split\_1’ and 2nd column as ‘from\_split\_2.

**MappedOutagesVerificationResult.xlsx:**

* This file is an output generated by MappedOutagesVerification.py file.
* Location:
* S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Constraint-Project\Data\Transmission Outages Verification\MappedOutagesVerificationResult.xlsx
* This file is used as an input by MappedOutagesVerificationHistogram.py file.

**UnmappedTransmissionOutagesAprroximateStringMatch.xlsx:**

* Location:
* S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Constraint-Project\Data\UnmappedTransmissionOutagesAprroximateStringMatch.xlsx
* This file is the output of ApproximateMatchingUnmappedTransmissionOutages.py file.
* Manual processing carried out in this file is as follows:
* Create a new sheet named ‘manual\_mapped’
* Copy ‘facility’, ‘match’, ‘matching\_code’ to the new sheet.
* Manually check strings separated by ‘\*’ in the matching code column with the match column for similarity.
* If they mostly match, check for the from and to bus name in the auction mapping document of the year 2019 September and compare it with first part of the ‘facility’ column.
* If all the checks are correct or mostly match keep the row and delete the rows that don’t match.
* In the matching code column, keep only that string which matches the most removing the ‘\*’ as well.
* In the sheet named “unmappedAll” create a column named ‘matching\_code\_mapped’ which is filled using this formula: =IFERROR(VLOOKUP(D2,manual\_mapped!$A:$C,3,FALSE)," ")
* This file is used as an input file by SeparationOfUnmappedTransmissionOutages.py file.

**UnmappedTransmissionOutagesAprroximateSeparation.xlsx:**

* Location:
* S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Constraint-Project\Data\UnmappedTransmissionOutagesAprroximateSeparation.xlsx
* This file is the output of SeparationOfUnmappedTransmissionOutages.py file.
* Sheet – ‘mapped’ consist of transmission outages mapped manually.
* Sheet – ‘unmapped’ consist of transmission outages which remained unmapped.
* Sheet – ‘unmapped\_kv\_analysis’ is a pivot table created to analyze the kv range in which most of the transmission lines fall.
* This file is used as input by TransmissionOutagesFinal.py file.

**Statistics:**

**Constraint data:**

Total – 18843

Mapped Constraints – 12543

Unmapped Constraints – 6300

Mapped Contingencies – 16068

Unmapped Contingencies - 1929

Mapped Both – 11282

**Transmission Outages:**

Total – 9343

Mapped Outages – 7766

Unmapped Outages - 1791

**Why is mapping done?**

The data that we get from Yes Energy or Energy Velocity consists the names of Monitored Element and Contingency whereas PowerWorld only consists of numbers i.e. From Bus Number and To Bus Number. So, mapping of names to numbers needs to be done to perform calculations of sensitivity factors such as PTDF, LODF and TLR.

Similar process must be carried out for transmission outages to determine which transmission line is open/close in the power system network.

**Parameters that are thought to be considered in training dataset:**

* All transmission outages as different parameter (can have a 0/1 value or LODF)
* Load
* Fuel Price
* PTDF
* On Peak/Off Peak
* Generators
* Date
* Month
* Year
* Week
* Time
* Shadow Price