

Constraint Generator

Introduction

The TVStudy Data Processing and Constraint Generation (Constraint Generator) software is used to process data produced by the FCC's *TVStudy* software to create pairwise interference constraint files in the format adopted by the FCC for use in the repacking process of the upcoming broadcast incentive auction.

The Constraint Generator comprises two distinct steps. First, SQL scripts process raw point data produced by *TVStudy* to generate a database of pairwise station interference data. Next, a Java program outputs the constraint files in a specified format using the processed data from the SQL database.

FCC Releases

- FCC's LEARN Repacking page (<http://wireless.fcc.gov/incentiveauctions/learn-program/repacking.html>)
- Incentive Auction Report & Order (https://apps.fcc.gov/edocs_public/attachmatch/FCC-14-50A1.pdf)
- Repacking Data PN (https://apps.fcc.gov/edocs_public/attachmatch/DA-13-1613A1.pdf)
- Feasibility Checking PN (https://apps.fcc.gov/edocs_public/attachmatch/DA-14-3A1.pdf)

License

Constraint Generator is released under the GNU General Public License (GPL) – <http://www.gnu.org/copyleft/gpl.html> (<http://www.gnu.org/copyleft/gpl.html>) .

System Requirements

- **Platforms:** Constraint Generator runs on Mac OS X, Linux (<https://github.com/fcc/Constraint-Generator/README.md>) , and Windows platforms.
- **Software Pre-requisites:** Java Runtime Environment (JRE7) (<http://java.com/download>) , PostgreSQL 9.3.1 (<http://postgresql.org/download>) or higher, Git, and *TVStudy* 1.3.1 (<http://data.fcc.gov/download/incentive-auctions/OET-69/>)
- **Disk Space:** 1 TB (recommended).
Note: Each run of *TVStudy* produces over 500GB of data that must be loaded into a robust database. Once loaded, indexing of data requires additional space.
- **Memory & Processing:** 4GB RAM (recommended) & modern multi-core processor.

Version

This manual is for Constraint Generator v1.0.0 (Windows version). To install Constraint Generator on Mac OS X or Linux, please see the README.md (<https://github.com/fcc/Constraint-Generator/README.md>) file.

Installation

1. Download the Constraint Generator software using git by typing `git clone https://github.com/fcc/Constraint-Generator/` at the command prompt or by using the GitHub for Windows client to clone `https://github.com/fcc/Constraint-Generator` to your local machine.
2. Install PostgreSQL 9.3.1 or higher on your local machine or a robust computer using the default settings.
3. In the Constraint-Generator folder, extract `tvstudyprocessing.zip` to create the `.\tvstudyprocessing\` subfolder
 - To extract the .zip file, right click `tvstudyprocessing.zip` select "Extract All" and then browse to the Constraint-Generator base folder (**note:** do NOT extract to the default path in Windows, or else you will create a subfolder `.\tvstudyprocessing\tvstudyprocessing\` and the scripts will not work)

Setting up the Database

PostgreSQL Setup

1. Open the pgAdmin III GUI tool, then right-click on "PostgreSQL 9.3 (localhost)" in the object browser and click connect (enter the password set during installation if prompted). Expand the Databases object, select the default `postgres` database and then click Plugins -> PSQL Console.
2. At the psql console prompt, create a PostgreSQL user (default: `db_ia_owner`) and set a password (default: `changeme`)

```
CREATE USER db_ia_owner WITH PASSWORD 'changeme';
```

3. Create a PostgreSQL database `db_ia` for TVStudy data processing

```
CREATE DATABASE db_ia OWNER db_ia_owner;
```

4. Grant privileges on the `db_ia` database to the user `db_ia_owner`

```
GRANT ALL PRIVILEGES ON DATABASE db_ia TO db_ia_owner;
```

Reference Table Setup

Note: This step only needs to be run once.

1. Open the pgAdmin III GUI tool, then right-click on "PostgreSQL 9.3 (localhost)" in the object browser and click connect (enter the configured password if prompted). Expand the Databases object, select the newly created db_ia database and then click Plugins -> PSQL Console.
2. Create the ia schema as the db_ia_owner

```
CREATE SCHEMA ia AUTHORIZATION db_ia_owner;
```

3. Grant additional privileges to db_ia_owner

```
GRANT USAGE ON SCHEMA ia TO db_ia_owner;  
GRANT SELECT ON ALL TABLES IN SCHEMA ia TO db_ia_owner;
```

4. Change the working directory to the base of the Constraint-Generator directory (e.g., C:\Users\FCC\Documents\GitHub\Constraint-Generator\)

```
\cd C:/Users/FCC/Documents/GitHub/Constraint-Generator/
```

5. Run the SQL code that creates and loads data into all the reference tables

```
\i 'scripts/reference.sql'
```

Note: the step above, and all other SQL scripts, expects .csv data to exist in the .\tvstudyprocessing\ folder. Please make sure that you have downloaded and extracted the tvstudyprocessing.zip file to this folder.

TVStudy Output Data Processing

In order to use this code to process *TVStudy* output data, you must run the *TVStudy* software 8 times using the following specific sets of channel replications:

- Channel 2, 3, 4
- Channel 5, 6, 33, 34, 35, 36
- Channel 7, 8, 9, 10, 11, 12, 13
- Channel 14, 15, 16, 17, 18, 19, 20
- Channel 21, 22, 23, 24, 25, 26
- Channel 27, 28, 29, 30, 31, 32
- Channel 38, 39, 40, 41, 42, 43, 44
- Channel 45, 46, 47, 48, 49, 50, 51

To replicate TV Stations onto specific channels using the FCC's *TVStudy* software, please follow the TVStudy instructions (<http://data.fcc.gov/download/incentive-auctions/OET-69/>) .

Note: These steps must be run each time you would like to generate constraints from different *TVStudy* runs. Given the volume of data, the processing typically takes at least 5–8

hours.

1. Move `stations.csv`, `service.csv` and `interference.csv` from the Channel 02-04 *TVStudy* run to the `.\tvstudyprocessing\ch0204\` folder.
2. Move `service.csv` and `interference.csv` from the Channel 05-06 & 33-36 *TVStudy* run to the `.\tvstudyprocessing\ch05063336\` folder.
3. Move `service.csv` and `interference.csv` from the Channel 07-13 *TVStudy* run to the `.\tvstudyprocessing\ch0713\` folder.
4. Move `service.csv` and `interference.csv` from the Channel 14-20 *TVStudy* run to the `.\tvstudyprocessing\ch1420\` folder.
5. Move `service.csv` and `interference.csv` from the Channel 21-26 *TVStudy* run to the `.\tvstudyprocessing\ch2126\` folder.
6. Move `service.csv` and `interference.csv` from the Channel 27-32 *TVStudy* run to the `.\tvstudyprocessing\ch2732\` folder.
7. Move `service.csv` and `interference.csv` from the Channel 38-33 *TVStudy* run to the `.\tvstudyprocessing\ch3844\` folder.
8. Move `service.csv` and `interference.csv` from the Channel 45-51 *TVStudy* run to the `.\tvstudyprocessing\ch4551\` folder.

Process the TVStudy Output Data

Note: It is very important to move the *TVStudy* output .csv files to the correct `.\tvstudyprocessing\` subfolders. All code references the locations in those subfolders.

1. Open the pgAdmin III GUI tool, then right-click on "PostgreSQL 9.3 (localhost)" in the object browser and click connect (enter the configured password if prompted). Expand the Databases object, select the `db_ia` database and then click Plugins -> PSQL Console.
2. Change the working directory to the base of the Constraint-Generator directory (e.g., `C:\Users\FCC\Documents\GitHub\Constraint-Generator\`)

```
\cd C:/Users/FCC/Documents/GitHub/Constraint-Generator/
```

3. Run the SQL code that conducts the initial processing of data to determine the baseline interference-free population of each station

```
\i 'scripts/initial.sql'
```

4. Once the previous step is complete, run the SQL code to process the Channel 02-04 data

```
\i 'scripts/ch0204.sql'
```

5. Next, run the SQL code to process the Channel 05-06, 33-36 data

```
psql> \i 'scripts/ch05063336.sql'
```

6. Next, run the SQL code to process the Channel 07-13 data

```
psql> \i 'scripts/ch0713.sql'
```

7. Next, run the SQL code to process the Channel 14-20 data

```
psql> \i 'scripts/ch1420.sql'
```

8. Next, run the SQL code to process the Channel 21-26 data

```
psql> \i 'scripts/ch2126.sql'
```

9. Next, run the SQL code to process the Channel 27-32 data

```
psql> \i 'scripts/ch2732.sql'
```

10. Next, run the SQL code to process the Channel 38-44 data

```
psql> \i 'scripts/ch3844.sql'
```

11. Next, run the SQL code to process the Channel 45-51 data

```
psql> \i 'scripts/ch4551.sql'
```

12. Finally, run the SQL code that conducts the final processing to merge all intermediate pairwise interference tables into a single table

```
psql> \i 'scripts/final.sql'
```

13. It is possible to parallel process Steps 3-12 by opening multiple psql console connections (assuming you have adequate computing resources)
14. The SQL code is written in such a way that each piece of code is committed during execution (rather than doing a single commit into the database at the completion of the script).
15. The code is written in such a way that if you run Steps 3-12 a second time, it will delete the previous tables.
16. If processing of a particular script is interrupted or fails, it is safe to simply re-run the .sql script that failed (this will not cause any issues with previously completed .sql scripts)
17. If you wish to keep previous runs of TVStudy output please make to export a CSV of the previous result or rename the table, for example by running this command in psql:

```
ALTER TABLE ia.tvsoftware_pairwise_result_final RENAME TO tvsoftware_pairwise_result_final_oct2014
```

Constraint Generation

Configuration

There is 1 model available:

- CS_RepackUSFixedCAFixedMX_Option2

Edit the parameters as necessary in the file located in the `profiles` subfolder.

Configurable Parameters:

- MODEL parameters
 - `database_name` – name of the database (default: `db_ia`)
 - `database_user_name` – database account username (default: `db_ia_owner`)
 - `database_password` – database account password (default: `changeme`)
 - `database_ip_address` – IP address of the database (default: `127.0.0.1` localhost)
 - `database_port` – database port number (default: 5432 or 5433 on Mac OS X)
 - `nationwide_acceptable_interference_pct` – acceptable threshold of interference in terms of the interference free population of a TV station. This threshold defines whether any pair of stations can be co-channel or adjacent-channel ($\text{adj}+1/\text{adj}-1$) (default: 0.5 percent)
- DB_TABLE parameters
 - `tv_stations_tablename` – name of the table that stores US, Canada, and Mexico station information (default: `tvsoftware_stations`)
 - `lm_station_tablename` – name of the table that stores US Land Mobile (LM) station information (default: `ia_lm_master`)
 - `lmw_station_tablename` – name of the table that stores Land Mobile Waiver (LMW) station information (default: `ia_lmw_master`)
 - `lm_lmw_interference_tablename` – name of the table that stores pairwise interference truth table for protecting Land Mobile (LM) and Land Mobile Waiver (LMW) stations (default: `ia_lm_lmw_interference_table`)
 - `mx_interference_tablename` – name of the table that stores pairwise interference truth table for protecting Mexican stations (default: `ia_mx_interference_table`)
 - `tvstudy_interference_tablename` – name of the table that stores pairwise interference population percentages to protect repacking stations from having additional interference above the threshold defined in the `nationwide_acceptable_interference_pct` parameter (default: `tvsoftware_pairwise_result_final`)

Constraint Generator Usage

Open a command prompt window and change directory to the Constraint-Generator base folder (e.g., `C:\Users\FCC\Documents\GitHub\Constraint-Generator\`)

```
cd \Users\FCC\Documents\GitHub\Constraint-Generator
```

Run the following command:

```
constraintgen.bat CS_RepackUSFixedCAFixedMX_Option2
```

Note: if you receive an error about memory allocation, please ensure that you have a 64-bit version of the Java Runtime Environment installed or edit the `constraintgen.bat` file and reduce the Java max memory flag `-Xmx4096M` to `-Xmx1500M` or smaller.