

## QGIS Plugin - Rubber Sheeting, etc.

by Spatial Tapestry  
email: SpatialTapestry@gmail.com

|  |   |
|--|---|
| <b>Donations can be made to:</b><br><br>paypal.me/SpatialTapestry (@SpatialTapestry)<br>ANZ Bank: Account Name: Spatial Tapestry P/L<br>BSB: 012430<br>Account Number: 493104982<br>Swift: <b>ANZBAU3M</b> | <b>It is assumed the user knows:</b><br><br>1/ how to make one or more themes 'Active' in the QGIS Menu<br>2/ the correct database connection parameters for their database<br>3/ the CRS/SRID/EPG that the themes will be stored in<br>4/ the database connectors have been installed, see <b>Installation Instructions</b> below. |
|--|---|

QGIS seems to be missing 'Spatial Upgrade' tools to things like Rubber Sheeting or Two Point Transformations.

Spatial Tapestry has been doing 'Spatial Upgrades' for Utilities and Local Government for 30 years, and has an excellent Windows Desktop Application which is far more advanced than this Plugin. The effort to port that functionality to QGIS would be maybe 3 to 5 years. Please feel free to email the author for the User Manual.

However for many the Rubber Sheeting algorithm provided will adequate. This is nothing more than calculating the weighted average shift from the 10 nearest Shift Vectors.

Spatial Tapestry would love to hear back from anyone with ideas for other algorithms, happy to look at coding them if they sound interesting.

This plugin has 2 tools. Rubber Sheeting and Two Point Transformation. Both operate on the 'Active Themes' in the QGIS Legend, and store the results in either a Postgres/PostGIS, Oracle or SQL Server database.

### User Instructions:

There are 3 parts to this, setting up the storage (which must be a database), setting up the Shift Vector scenario to do a Rubber Sheeting process over the active themes, and setting up a Two Point Transformation process over the active themes.

### Setting up the Storage

Select the database where the data is to be stored. Note that Oracle is supported via both the 'oracledb' and 'Devart' connectors.

Enter the database connection parameters and the SRID. The SQL can be written to a local file, in the user's home directory on both Windows and Linux, if that is wanted then check the 'Write SQL to (ASCII) File' CheckBox. The SQL can also be executed against the database directly, if that is wanted then check the 'Execute SQL against the Database' CheckBox. If both are wanted then check both CheckBoxes.

Click "Save Project File". The Project file will be loaded everytime that database type is selected.

Click "Test Database Connection, User Name and Password" if there is a need to test the connection. The plugin will attempt to open the connection and create a 'testtable'.

The 'Delete the QGIS Selected Themes from Serv' button can be used to delete one or more themes from the database. This may be useful for several reasons such as MSSQL not accepting invalid coordinates, where one fixes a geometry in QGIS and needs to reload the theme, bearing in mind that MSSQL does not support 'IF EXISTS' on a 'DROP THEME' command. It is also useful for dropping themes when no longer needed.

The 'Process the QGIS Selected Themes to Serv' is how the SQL is generated and stored to file and/or executed to the database. As this is running, one can look at their home directory and see the SQL and/or TXT (Report) Files being created as each theme is processed.

### Database specifics:

Postgres needs a schema to work in and is generally the easiest one to manipulate from SQL with the 'IF EXISTS' and 'IF NOT EXISTS' clauses. The SQL Files are compatible for PGAdmin 3 and 4, and DBeaver.

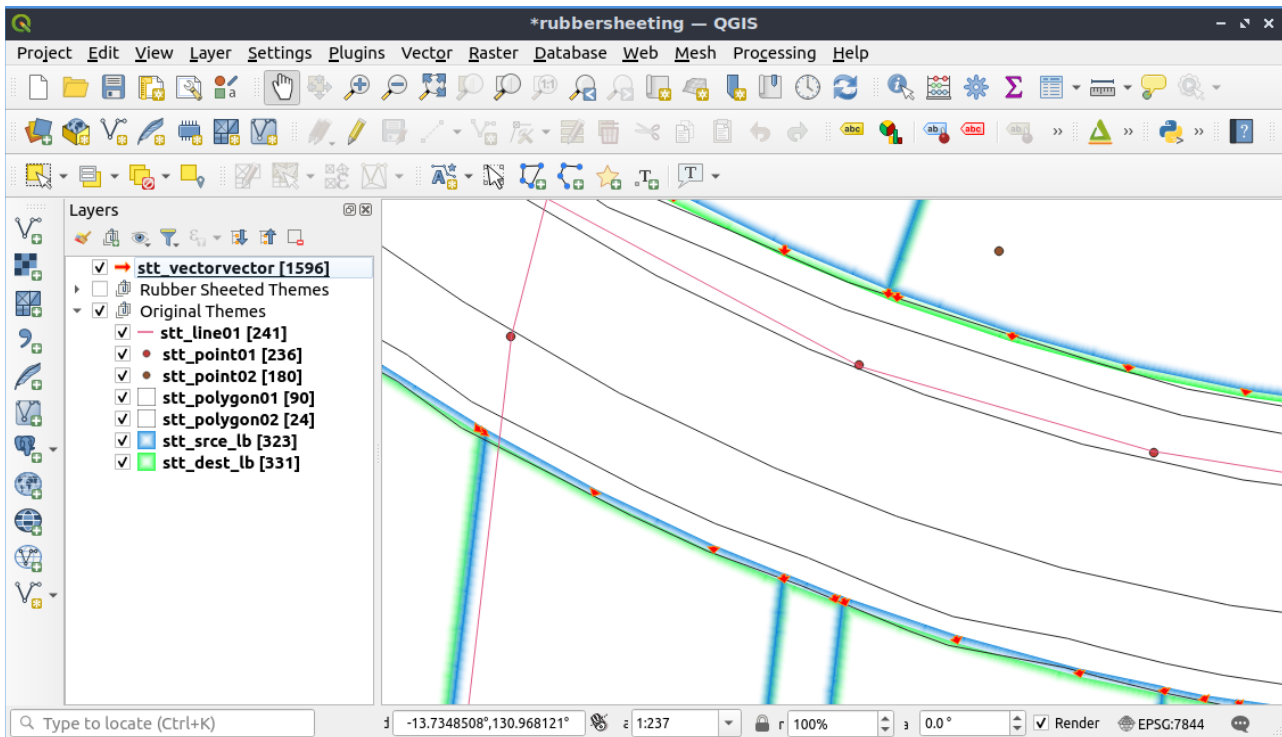
Oracle supports 4000 characters in client side SQL and 32700 characters with binding/clobs for the geometry so some very large features may not load. **Thanks to Simon Greener for help with Bind Variables and Clobs.** Oracle only supports schemas for the user themselves. The SQL Files are compatible for SQL Developer and DBeaver.

MS SQL does not support 'IF EXISTS' and 'IF NOT EXISTS' clauses so reprocessing can be tedious. Schemas are supported but seems to be simply prefixes to the table name. The SQL Files are compatible for MS SQL Server Admin, and DBeaver.

In all cases it falls to the user to have all the connection details sorted, so basically if DBeaver and other tools cannot connect then neither can this Plugin.

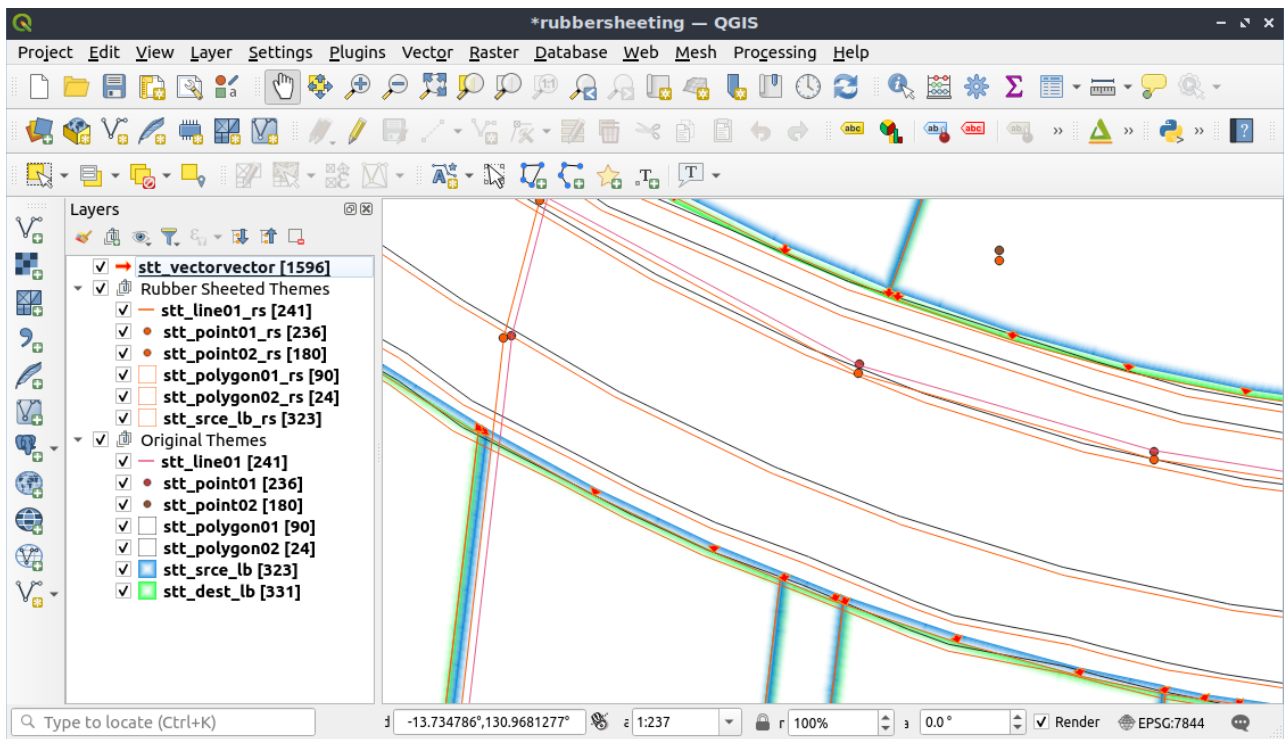
### Rubber Sheeting Instructions:

There is a sample GeoPackage and QGIS Project provided. There must be a theme in the legend called 'stt\_vectorvector' before the "Write out Rubber Sheeted X,Y" can be Ticked (and enabled). This theme must have a LineString geometry type and contain geometries containing the Shift Vectors. Mostly they represent joins between an older (less accurate) landbase/cadastre and a newer (more accurate) one. However they can be joins to a orthophoto, geodetic shifts, anything really.



Simply make the themes active and click "Process the QGIS Selected Themes" button to write out the Rubber Sheeted geometries to the destination. Note that the theme names will have an "\_rs" appended to the theme name.

Turning on the "Rubber Sheeted Themes" will show the themes like the image below:



Whilst the Rubber Sheeting algorithm provided will do a reasonable job for most themes in most circumstances, there is no attempt at maintaining Rigid Themes (e.g. buildings), Straight Lines (e.g. power lines or sewer mains), Implied Vertical Topology (where a house sewer line joins (visually) a street sewer line and there is no matching point in the street sewer line - i.e. Vertical Topology), etc. As mentioned elsewhere, if you have these types of geometries, please contact the author who can help through a consultation process.

## Two Point Transformation Instructions - with known CRS/SRID:

To access this, click "Two Point Transformation - known CRS/SRID" at the bottom right of the form.

A Two Point Transformation can be useful when geometries are found to be in the wrong location. Simply export those features to say a GeoPackage, open the themes in QGIS, make them active, set the transformation parameters and import them back into new themes in your database, and use QGIS to replace the original features.

### Manage Database, User and Schema Details:

Please choose your Database:

☒ Postgres and PostGIS  
☐ Oracle (oracledb) (asWkt max(32,000 string length))  
☐ Microsoft SQL Server (Devart)    ☐ Oracle (Devart)

Note: Maximum of 2000 Features per Theme!  
See RubberSheetingEtc.html for full support.

IP Address:     Test Database Connection, User and Password  
 Database Name:   
 User Name:     Save Project File  
 Password:       
 Schema Name:   
 Port:    

Choose Output Locations (Either or Both):

☒ Write SQL to (ASCII) File    ☒ Execute SQL against the Database

### RubberSheeting with ST\_VectorVector

#### Two Point Transformation - known CRS/SRID

From Easting/Longitude:   
 From Northing/Latitude:   
 To Easting/Longitude:   
 To Northing/Latitude:   
 Rotation (Dec Deg):   
 Scale Factor:

X and Y above are in the "Save as SRID"!

NOTE: Selected Themes MUST have defined CRS/SRID/EPG.

#### Two Point Transformation - Local Coordinates

Another use for this is to reproject themes and store them in your database.

Simply make the themes active and click "Process the QGIS Selected Themes" button to write out the Rubber Sheeted geometries to the destination. Note that the theme names will have an "\_tp" appended to the theme name.

## Two Point Transformation Instructions - Local Coordinates:

To access this, click "Two Point Transformation - Local Coordinates" at the bottom right of the form.

A Two Point Transformation can be useful when importing geometries such as airports, buildings, bridges, interchanges, landXML Survey Plans, etc, which are often in a Local Coordinate System.

Note: The 'Units' can say 'degrees', as below:

## Coordinate Reference System (CRS)

|                       |   |
|-----------------------|---|
| <b>Name</b>           | Undefined geographic SRS                        |
| <b>Units</b>          | degrees   |
| <b>Method</b>         | Lat/long (Geodetic alias)                       |
| <b>Celestial body</b> | Earth   |
| <b>Reference</b>      | Static (relies on a datum which is plate-fixed) |

What matters is that the extent is representative of the theme's extent in Metres:

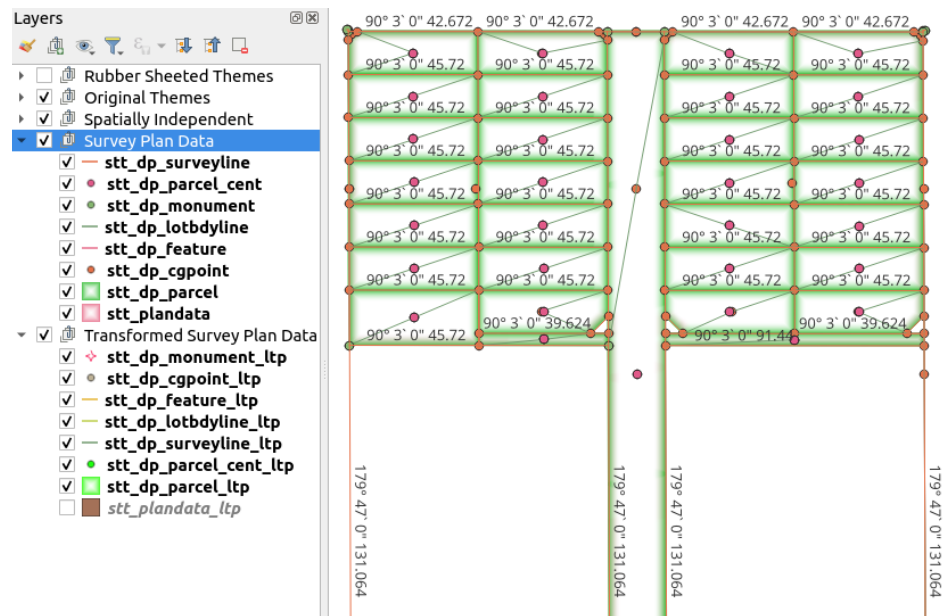
**Extent** 2062.0131835937500000,0.0000000000000000 : 2266.5377312982200237,242.7969452785899875

A good way to do this is to get the coordinates of one corner in the the source theme, get the coordinate of the same point in the GIS Coordinate System, work out any rotation and scale factors, and enter these in the form above.

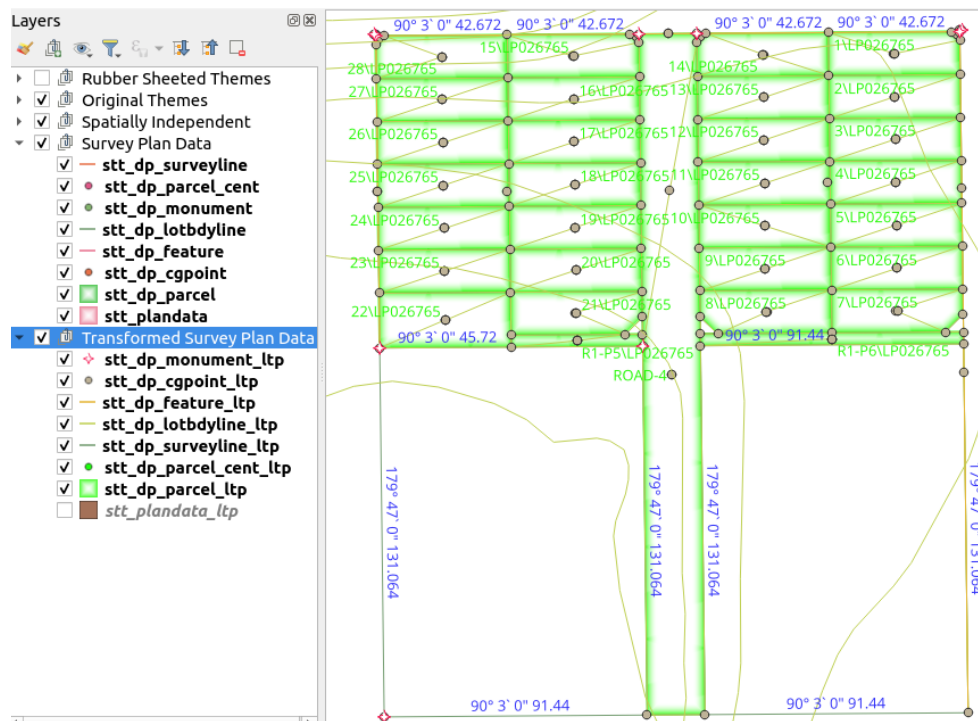
Simply make the themes active and click "Process the QGIS Selected Themes" button to write out the Rubber Sheeted geometries to the destination. Note that the theme names will have an "\_ltp" appended to the theme name.

Note: If the themes are in feet, treat it as metres but change the Scale Factor to 0.3048.

There is a LP026765\_NoCRS.gpkg file with an Australian Survey Plan loaded, with no CRS/SRID assigned. This should simulate .DWG files loaded to .GPKG via QGIS's Import functionality. The QGIS Project file, "rubbersheeting.qgs" will display these as:



Running the "Two Point Transformation - Local Coordinates" will produce:



## Installation Instructions:

QGIS 3.28 is recommended. Do not install Python, instead we will use the Python installed with QGIS. (Installing Python 3.12 on Windows causes problems)

All 3 database connectors must be installed otherwise code will return errors. Alternatively the Python Source code can be modified to remove references to any unwanted connectors.

### Ubuntu

```
pip3 install -U pycpg2
pip3 install -U oracledb
pip3 install -U pyodbc
```

### Windows 10/11

On Windows the "PythonPath" Environment Variable must be set - be sure to adjust this to your particular computer and QGIS Version:  
PYTHONPATH

C:\Program Files\QGIS 3.28.15\apps\qgis\python;C:\Program Files\QGIS 3.28.15\apps\qgis\python\plugins;C:\Program Files\QGIS

```
3.28.15\apps\Qt5\plugins;C:\Program Files\QGIS 3.28.15\share\gdal;
C:\Program Files\QGIS 3.28.15\apps\Python39\Scripts\pip3 install -U psycopg2
C:\Program Files\QGIS 3.28.15\apps\Python39\Scripts\pip3 install -U oracledb
C:\Program Files\QGIS 3.28.15\apps\Python39\Scripts\ pip3 install -U pyodbc
```

### Devart components

The Devart components are used by the 'pyodbc' connector to deal with the MS SQL Server and Oracle. i.e. If you are uploading to PostgreSQL or Oracle (and the oracledb connector is working) then the Devart components are not necessary.  
Note: QGIS 3.28.15 has Python 3.9 as part of its installation!

The following URLs will help with the DEVART components.

<https://www.devart.com/python/oracle/download.html>

Python Connector for Oracle 1.0 - Python 3.9 for Windows

Or

Python Connector for Oracle 1.0 - Python 3.9 for Linux

```
cd C:\Users\r\Downloads\devart_oracle_cp39_win\DevartPythonOracle\whl
"C:\Program Files\QGIS 3.28.15\apps\Python39\Scripts\pip3" install devart_oracle_connector-1.0.1-cp39-cp39-win_amd64.whl
```

<https://www.devart.com/python/sqlserver/download.html>

Python Connector for SQL Server 1.0 - Python 3.9 for Windows

Or

Python Connector for SQL Server 1.0 - Python 3.9 for Linux

```
cd C:\Users\r\Downloads\devart_sqlserver_cp39_win\DevartPythonSqlServer\whl
"C:\Program Files\QGIS 3.28.15\apps\Python39\Scripts\pip3" install devart_sqlserver_connector-1.0.1-cp39-cp39-win_amd64.whl
```

### Sample Database .config files:

The plugin stores the database connection details for easy reuse. These are included in the Plugin's folder but need to be moved to the user's home folder before they can be accessed.

The first 3 characters indicate the connection type:

#### PGS\_ST\_SDL.Config (Postgres)

```
host=192.168.56.1
database=gisdb
user=rogermsu
password=rogermsu
port=15432
schemaname=test123
tableSRID=EPSG:7844
DatabaseType=PGS
Output2=Serv
```

#### MSS\_ST\_SDL.Config (Microsoft SQL Server via Devart)

```
host=192.168.56.1
database=gisdb
user=drrog
password=drrog
port=11433
schemaname=testSchema
tableSRID=EPSG:7844
DatabaseType=MSS
Output2=Both
```

#### ORA\_ST\_SDL.Config (Oracle)

```
host=192.168.56.1
database=xex
user=drrog
password=drrog
port=11521
schemaname=testSchema
tableSRID=EPSG:7844
DatabaseType=ORA
Output2=Both
```

#### ORD\_ST\_SDL.Config (Oracle via Devart)

```
host=192.168.56.1
database=free
user=drrog
password=drrog
port=1521
schemaname=testSchema
tableSRID=EPSG:7844
DatabaseType=ORD
Output2=Both
```

**Some Postgres code to get started with (PGADMIN4):**

```
CREATE DATABASE gisdb WITH OWNER = postgres ENCODING = 'UTF8' TABLESPACE = pg_default CONNECTION LIMIT = 300;  
CREATE ROLE rogermsu LOGIN PASSWORD 'rogermsu' SUPERUSER INHERIT CREATEDB CREATEROLE NOREPLICATION;  
CREATE SCHEMA testdata AUTHORIZATION rogermsu;
```

**Some Oracle code to get started with (SQL Developer)**

```
Note: With Oracle each user has their own schema  
Enter user-name: system  
Enter password:  
Last Successful login time: Wed Jan 10 2024 14:33:56 +08:00  
Connected to:  
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production  
Version 21.3.0.0.0  
SQL> create user drrog identified by drrog;  
create user drrog identified by drrog  
*  
ERROR at line 1:  
ORA-65096: invalid common user or role name  
SQL> alter session set "_ORACLE_SCRIPT"=true;  
Session altered.  
SQL> create user drrog identified by drrog;  
User created.  
SQL>  
-- USER SQL  
CREATE USER "drrog" IDENTIFIED BY "drrog" ;  
-- ROLES  
GRANT "CONNECT" TO "DRROG" ;  
GRANT "DBA" TO "DRROG" ;  
-- SYSTEM PRIVILEGES  
GRANT ALTER ANY TABLE TO "DRROG" ;  
GRANT DROP ANY TABLE TO "DRROG" ;  
GRANT INSERT ANY TABLE TO "DRROG" ;  
GRANT CREATE ANY TABLE TO "DRROG" ;
```

**Some MSSQL comments to get started with:**

Remember to open up TCPIP on the server and create the user, etc.