MARS ground penetrating radars tracks GIS vector layers Documentation

Release 0.9

Federico Cantini, Anton Ivanov (eSpace-EPFL)

September 09, 2016

CHAPTER

ONE

ABOUT THE DB

This is a short documentation for the use of a database of **geometric data and metadata** of the tracks of the two MARS ground penetrating radars existing so far: MARSIS (http://sci.esa.int/mars-express/34826-design/?fbodylongid=1601) and SHARAD(http://mars.nasa.gov/mro/mission/instruments/sharad/). The data are available as **vector layers** to be used with **GIS software**.

The database is a result of the contribution of the EPFL Space Engineering Center (http://espace.epfl.ch/) **to the iMars grant agreement n°607379 (http://www.i-mars.eu/) of the European Union's Seventh Framework Programme (FP7/2007-2013).**

The layers are currently available through a direct connection to the PostGIS/PostgreSQL DBMS (http://postgis.net/ http://www.postgresql.org/) implementing the db. Availability of the layers through WFS (http://www.opengeospatial.org/standards/wfs) protocol using MapServer (http://mapserver.org/) is planned but not yet implemented.

CHAPTER

TWO

AVAILABLE LAYERS

Several layers are available for MARSIS and SHARAD tracks data.

_point layers contain, for each radar sampling point, geometric data and other metadata. For the list of the included data see Included data (page 5).

_lines layers are intended for a quick visualization of the orbit footprint without providing further details included in the *_point* layers. They can be used to show the track number on the QGIS canvas and maps.

In the layers named with the **_180_ suffix**, the **longitude** is represented **between -180° and +180°**. In the other layers the longitude is represented **between 0° and 360°**.

2.1 Using PostGIS DB connection

For the PostGIS connection parameters refer to PostGIS connection parameters (page 7)

2.1.1 MARSIS layers

- *orbit_point*: MARSIS sampling points (longitude between 0° and 360°. Features geometry type is *point*)
- *marsis_orbit_points_180*: MARSIS sampling points (longitude between -180° and 180°. Features geometry type is *point*)
- *marsis_orbit_lines*: MARSIS orbit tracks (longitude between 0° and 360°. Features geometry type is *line*)
- *marsis_orbit_lines_180*: MARSIS orbit tracks (longitude between -180° and 180°. Features geometry type is *line*)

2.1.2 SHARAD layers

- *orbit_point*: SHARAD sampling points (longitude between 0° and 360°. Features geometry type is *point*)
- *sharad_orbit_points_180*: SHARAD sampling points (longitude between -180° and 180°. Features geometry type is *point*)
- *sharad_orbit_lines*: SHARAD orbit tracks (longitude between 0° and 360°. Features geometry type is *line*)
- *sharad_orbit_lines_180*: SHARAD orbit tracks (longitude between -180° and 180°. Features geometry type is *line*)

2.2 Tracks layers in SQLite format

The aforementioned tracks layers are also available in *SQLite* format from TBA.

Both for MARSIS and SHARAD, the DB is partitioned as follow:

CHAPTER

THREE

INCLUDED DATA

For each radar sampling point, the _points_ layers provide the follwing data:

3.1 MARSIS layers:

- *point_id*: id of the corresponding radargram column
- *scetw*: SCET timeof the frame (whole)
- *scetf*: SCET timeof the frame (frac)
- *ephemt*: Ephemeris time (number of seconds since Jan 1 2000 - 12:00 UTC)
- *geoep*: Ephemeris time in UTC format
- *sunlon*: Mars solar longitude
- *sundist*: Mars to Sun distance
- *orbit*: Orbit number of the related dataproduct
- *target*: Celestial body observed
- *tarscx*: Target posistion (X component)
- *tarscy*: Target posistion (Y component)
- *tarscz*: Target posistion (Z component)
- *scalt*: Distance from the Mars Express spacecraft to the reference surface
- *scelon*: Longitude of the footprint location
- *sclat*: Latitude of the footprint location
- *tarscvx*: Mars Express spacecraft velocity vector in the reference frame of the target body (X component)
- *tarscvy*: Mars Express spacecraft velocity vector in the reference frame of the target body (Y component)
- *tarscvz*: Mars Express spacecraft velocity vector in the reference frame of the target body (Z component)
- *tarscradv*: Radial component of the Mars Express spacecraft velocity vector in the reference frame of the target body
- *tarsctanv*: Tangential component of the Mars Express spacecraft velocity vector in the reference frame of the target body
- *locsunt*: Local true solar time
- *sunzenith*: Solar zenith angle
- *dipx*: Unit vector directed along MARSIS dipole Antenna in the reference frame of the target body (X component)
- *dipy*: Unit vector directed along MARSIS dipole Antenna in the reference frame of the target body (Y component)

- *dipz*: Unit vector directed along MARSIS dipole Antenna in the reference frame of the target body (Z component)
- *monox*: Unit vector directed along MARSIS monopole Antenna in the reference frame of the target body (X component)
- *monoy*: Unit vector directed along MARSIS monopole Antenna in the reference frame of the target body (Y component)
- *monoz*: Unit vector directed along MARSIS monopole Antenna in the reference frame of the target body (Z component)
- *f1*: Values in Hz of the first radar frequency
- *f2*: Values in Hz of the second radar frequency
- *snr_f1_m1*: [[Signal to noise ratio]] of the first frequency, filter -1
- *snr_f1__0*: [[Signal to noise ratio]] of the first frequency, filter 0
- *snr_f1_p1*: [[Signal to noise ratio]] of the first frequency, filter 1
- *snr_f2_m1*: [[Signal to noise ratio]] of the second frequency, filter -1
- *snr_f2__0*: [[Signal to noise ratio]] of the second frequency, filter 0
- *snr_f2_p1*: [[Signal to noise ratio]] of the second frequency, filter 1

3.2 SHARAD layers:

The data provided in the SHARAD layers are those included in the SHARAD geometric data files (http://pds-geosciences.wustl.edu/mro/mro-m-sharad-5-radargram-v1/mrosh_2001/data/geom/)

- *point_id*: id of the corresponding radargram column
- *epoch*: UT date and time of observation
- *lat*: Latitude of the footprint location
- *lon*: Longitude of the footprint location
- *mars_r*: Radius of Mars at the footprint time
- *sc_r*: Distance from center of mass to MRO
- *rad_v*: MRO radial velocity
- *tan_v*: MRO tangential velocity
- *sza*: Solar zenith angle
- *phase*: Signal phase distortion
- *orbit*: Orbit number of the related dataproduct

CHAPTER

FOUR

POSTGIS CONNECTION PARAMETERS

4.1 eSpace MARSIS layers DB

Host: redmine-espace.epfl.ch
Port: 5432
Database: MARSIS
Username: marsisuser

4.2 eSpace SHARAD layers DB

Host: redmine-espace.epfl.ch
Port: 5432
Database: SHARAD
Username: marsisuser

CHAPTER

FIVE

GETTING LAYERS SUBSETS USING GDAL'S OGR2OGR

"GDAL (http://www.gdal.org/) is a translator library for raster and vector geospatial data formats that is released under an X/MIT style Open Source license by the Open Source Geospatial Foundation (http://www.osgeo.org/)."

Using the proper GDAL utility it is possible to **download subsets of data from MARSIS and SHARAD layers and saving it in one of the format managed by GDAL**. This can be useful to work without a network connection and can also lead to performance improvement using QGIS.

5.1 Download GDAL

Information about GDAL download and installation for GNU/Linux, OSX and Windows operating systems can be found here: (https://trac.osgeo.org/gdal/wiki/DownloadingGdalBinaries)

GDAL sources can be downloaded from here (http://download.osgeo.org/gdal/)

5.2 Getting layers subsets

The GDAL utility to fetch layers subsets is *ogr2ogr* (http://www.gdal.org/ogr2ogr.html). It is included in the GDAL installation.

5.2.1 Connecting to PostGIS db

The generic syntax of *ogr2ogr* command is the following:

```
ogr2ogr -f "*driver*" *filename* PG:"host=redmine-espace.epfl.ch user=\ *dbuser* dbname=\ *dbname
```

- *driver*: name of the GDAL driver to use to write data
- *filename*: name of the output file
- *dbuser*: database username. Please refer to [[PostGIS connection parameters]]
- *dbmane*: name of the database to fetch data from. Please refer to [[PostGIS connection parameters]]
- *password*: password provided to the users
- *layer_name*: name of the layer to fetch data from. Please refer to [[Available layers]]
- *min_lon min_lat max_lon max_lat*: longitude and latitude extent
- *restricted_where*: list of attribute to include in the output. Please refer to [[Included data]]

examples:

```
ogr2ogr -f "GML" file.gml PG:"host=redmine-espace.epfl.ch user=\ *dbuser* dbname=\ *dbname* passw
```

Fetches data of MARSIS sampling points from table *marsis_orbit_points_180* with **longitude between 10°W and 10°E and latitude between 30°S and 30°N** and save it in *file.gml* using GML format.

```
ogr2ogr -f "SQLite" file.sqlite PG:"host=redmine-espace.epfl.ch user=\ *dbuser* dbname=\ *dbname*
```

Fetches data of MARSIS sampling points from table *marsis_orbit_points_180* with **orbit number between 8000 and 8999** and save it in *file.sqlite* using SQLite format.

```
ogr2ogr -f "SQLite" file.sqlite PG:"host=redmine-espace.epfl.ch user=\ *dbuser* dbname=\ *dbname*
```

Fetches data of MARSIS sampling points from table *marsis_orbit_points_180* **restricted to orbit number, orbit point id and solar zenith angle** and save it in *file.sqlite* using SQLite format.

- For a detaild description* of *ogr2ogr* syntax please refer to http://download.osgeo.org/gdal/ or the documentation of your GDAL installation.

Service via WFS protocol is planned but not yet available.

5.3 Connecting to WFS service

To be implemented

Independently on the service used, the files containing the **fetched data can be open with QGIS or other software** (i.e. GRASS GIS) depending on the format used for saving it.