Routine GNSS Processing Reference & User Guide Dionysos Satellite Observatory, NTUA

Xanthos Papanikolaou Demitris Anastasiou

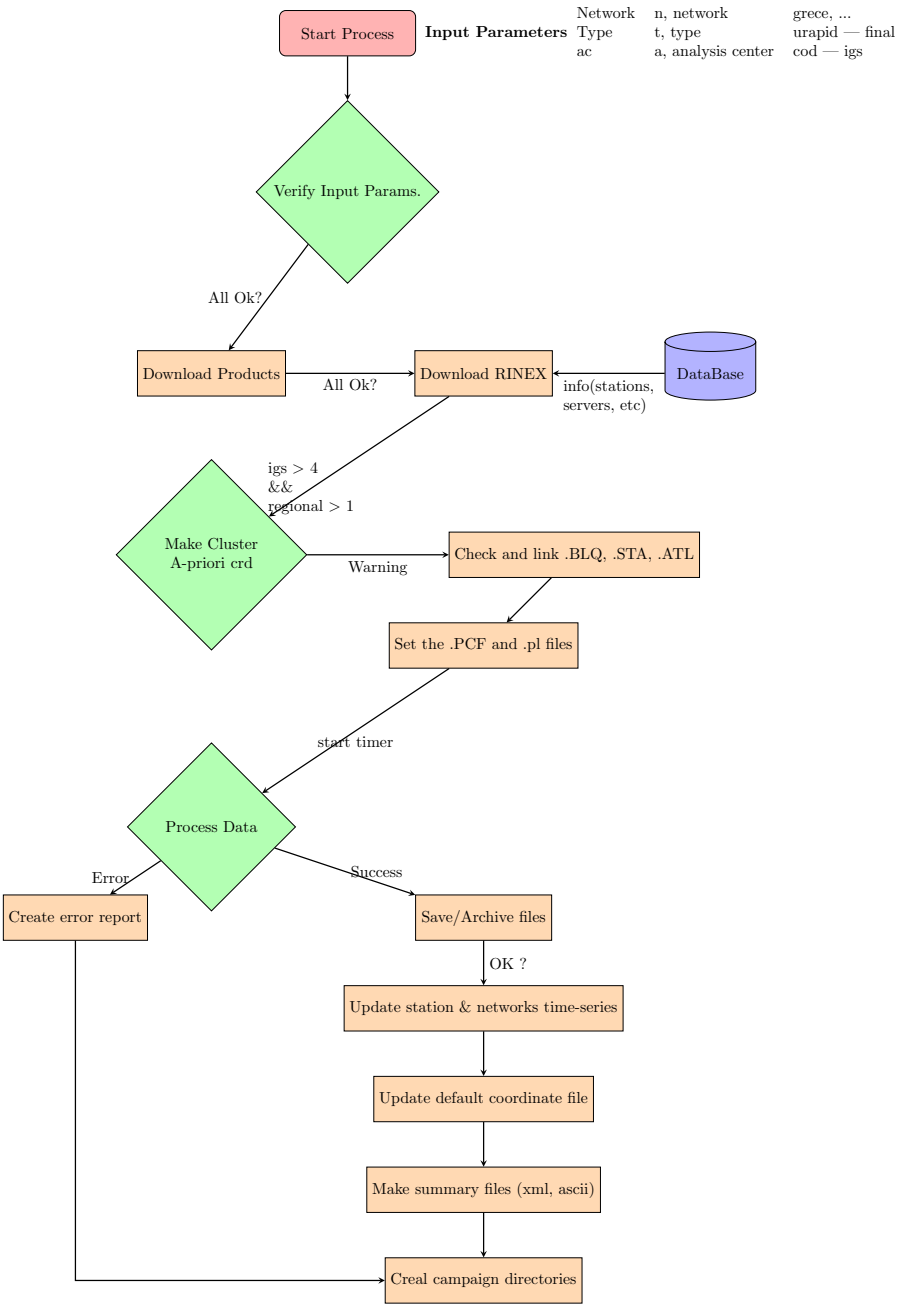
July 22, 2015

First Revision: June 10, 2015
Library Uri: `http://dionysos.survey.ntua.gr`
Version: v1.0-0

Abstract

This document describes the routine processing of GNSS data as developed at Dionysos Satellite Observatory (DSO), of National Technical University of Athens (NTUA).

Contents


Start Process
Input Parameters
Network n, network grece, ...
Type t, type urapid — final
ac a, analysis center cod — igs
Verify Input Params.
All Ok?
Download Products
All Ok?
Download RINEX
DataBase
info(stations, servers, etc)
igs > 4 && regional > 1
Make Cluster A-priori crd
Warning
Check and link .BLQ, .STA, .ATL
Set the .PCF and .pl files
start timer
Process Data
Error
Success
Create error report
Save/Archive files
OK ?
Update station & networks time-series
Update default coordinate file
Make summary files (xml, ascii)
Creal campaign directories

1 Programs

1.1 ddprocess

1.1.1 Purpose

`ddprocess.sh` is a bash (Shell) script to process a network in network mode, using the double-difference approach.
Location: `src/bash/ddprocess.sh`

1.1.2 Usage

```
ddprocess -y YYYY -d DDD [OPTIONS]
```

Switches:

- `-a --analysis-center=`[option]
  Specify the analysis center. This can be either

  a. igs, or

  b. cod

  Default value: cod.

- `-b --bernese-loadvar=` /foo/bar/LOADGPS.setvar
  Specify the Bernese LOADGPS.setvar file; this is needed to resolve the Bernese-related variables.

- `-c --campaign=`[option]
  Specify the campaign name. This name should be exactly the name of the campaign used within Bernese. Inside the script it is truncated to upper-case, so the options `--campaign=greece` and `--campaign=GREECE` are equivelant. For a list of all the files depending on this variable, see subsubsection 1.1.7, Note 1. Only specify the name of the campaign do **NOT** include the path.

- `-d --doy=`[option]
  Specify the day of year (as integer).

- `-e --elevation-angle=`
  Specify the elevation cut-off angle. The angle is expressed as integer in degrees. Default value: 3.

- `-f --ion-products=`[option]
  Specify (a-priori) ionospheric correction file identifier. If more than one, use a comma-seperated list (e.g. `-f FFG,RFG`). See subsubsection 1.1.7, Note 2.

- `-i --solution-id=`[option]
  Specify solution id (e.g. `FFG`). See subsubsection 1.1.7, Note 3.

- `-l --stations-per-cluster=`[option]
  Specify the number of stations per cluster. Input should be apositive integer. Default value is 5.

- `-m --calibration-model=`[option]
  The extension (model) used for antenna calibration files. This can be e.g. `I01, I05 or I08`. What you enter here, will be appended to the pcv filename (provided via the `-p` switch) and all calibration-dependent Bernese processing files (e.g. `SATELLITE.XXX`). See subsubsection 1.1.7, Note 4.

- `-p --pcv-file=`[option]
  Specify the PCV file to be used. Do not provide the extension (this is automatically appended using the `-m` switch). See subsubsection 1.1.7, Note 4.

- `-r --save-dir=`[option]
  Specify directory where the solution will be saved; note that if the directory does not exist, it will be created.

- `-s --satellite-system=`[option]
  Specify the satellite system; this can be:

  a. gps, or

  b. mixed (i.e. gps+glonass)

  Default value is gps.

- `-t --solution-type=`[option]
  Specify the solution type; this can be:

  a. final, or

  b. urapid

- `-u --update=`[option]
  Specify which records/files should be updated; valid values are:

  a. `crd` : update the default network crd file.

  b. `sta` : update station-specific files, i.e. time-series records for the stations.

  c. `ntw` : update update network-specific records.

  d. `all` : all both the above.

  More than one options can be provided, in a comma seperated string e.g. `--update=crd,sta`.

- `-y --year=`[option]
  Specify the year as a 4-digit integer.

- `-x --xml-output`
  Produce an xml (actually docbook) output summary report.
- `--force-remove-previous`
  Remove any files from the specified save directory (`-r --save-dir=`) prior to start of processing.
- `--add-suffix=`[option]
  Add a suffix (e.g. `_GPS`) to saved products of the processing.
- `-h --help`
  Display (this) help message and exit.
- `-v --version`
  Dsiplay version and exit.

1.1.3 Prerequisites

1.1.4 Exit Status

On sucess, the program returns `0`.

1.1.5 ToDo

1.1.6 Bugs

Send reports to:
Xanthos Papanikolaou mailto:xanthos@mail.ntua.gr
Demitris Anastasiou mailto:danast@mail.ntua.gr

1.1.7 Notes

1 A list of files are expected to be present in the tables directory, specified by the campaign name. See item 1:

Expected File	Linked to
${TABLES}/pcv/${PCV_FILE}	${X}/GEN/${PCV_FILE}
${TABLES}/sta/${CAMPAIGN}.STA	${P}/STA/${CAMPAIGN}.STA
${TABLES}/blq/${CAMPAIGN}.BLQ	${P}/STA/${CAMPAIGN}.BLQ
${TABLES}/atl/${CAMPAIGN}.ATL	${P}/STA/${CAMPAIGN}.ATL
${TABLES}/crd/${CAMPAIGN}.igs	
${TABLES}/crd/${CAMPAIGN}.epn	
${TABLES}/crd/${CAMPAIGN}.reg	

2 The ionospheric correction file, must be in the Bernese-specific ION format. These files should reside in the product area, specified by the variable ${PRODUCT_AREA} stored as ${PRODUCT_AREA}/YYYY/DDD/XXXYYDDD0.ION.Z, where `XXX` is the solution identifier specified by the `-f` option.

If none of these files are found (or if the `-f` switch is not used), then the script will try to download a Bernese-specific ION file from CODE's ftp, using the program wgetion. This downloaded files can be final, rapid or ultra-rapid.

3 The solution id will have an effect on the naming of the Final, Preliminary and Size-Reduced solution files. If e.g. the solution-id is set to `NTA`, then the Final solution files will be named `NTA`, the preliminary `NTP` and the size-reduced `NTR`.

4 The pcv file must reside in the `tables/pcv` folder, and will be linked by the script to the {%GEN} directory. Do not provide the extension; it will be automatically generated using the pcv file and the extension given via the calibration model (`-m`). E.g. using `-p GRE_PCV` and `-m I08`, then the script will search for the pcv file ${TABLES}/pcv/GRE_PCV.I08.

1.2 rnxdwnl

1.2.1 Purpose

`rnxdwnl.py` is a Python script to download RINEX files.
Location: `src/python/rnxdwnl.py`

1.2.2 Usage

```
rnxdwnl -y YYYY -d DDD [OPTIONS]
```

Switches:

- `-s, --stations=` station1,stations2,...
  Specify a comma-seperated list of stations to be downloaded. The stations specified here, must be included in the database. The name of stations specified, must be the one used by DSO (a 4-char string). In some, rare cases, this may not match the 'official' 4-char name of the station. When the rinex file is searched for on the web, the 'official' name will be used.

- `-n, --networks=` network1,network2,...
  Specify a comma-seperated list of networks. Every station, belonging to these networks will (try to) be downloaded. The name of the network(s) specified, must match valid networks in the database.

- `-y, --year=` YYYY
  Specify the year for which the RINEX are requested. This must be a (valid), 4-digit integer.

- `-d, --doy=` DDD
  Specify the day of year for which the RINEX are requested. This must be a (valid), integer.

- `-f, --force-remove`
  By default, if the RINEX to be downloaded already exists, with size > 0, then the downloading step is not performed (other steps -if specified- are perfored normaly). If this switch is turned on, then if a file exists, named exactly as the one to be downloaded, then this file is removed and a normal, download is performed.

- `-u, --uppercase`
  If turned on, the downloaded RINEX file(s) will be truncated to uppercase.

- `-p, --path=` /foo/bar
  Specify the directory (path) where the downloaded files will be downloaded to.

- `-z, --uncompress`
  Uncompress the downloaded RINEX file(s). Note that this will only work for UNIX compressed '.Z' files.

1.2.3 Prerequisites

The Python library MySQLdb must be installed and available for importing.

The program needs to connect to a (MySQL) database called `procsta`, and make various queries about the stations, networks, servers, etc. The structure of this database is strict, cause the queries are hardcoded in the source code of this program.

For more information on the used database, ask Mitsos.

The program is designed to work on UNIX-like systems. It will call the Shell to issue commands (like downloading, compressing, etc). Depending on the specific use, the following may be required:

- `compress` and/or `uncompress` utilities; needed to compress or uncompress UNIX-compressed files (i.e. `.Z`).

- `wget` needed to download remote files, when the protocol is `http`, `ftp`, or `https`.

- `scp` needed to download remote files, when the protocol is `ssh`. Note that if a non-standard port is used for `ssh`, then it must be hardcoded into the source code. Also, the remote and server sites, must be able to connect without using explicit passwords (they must hold the ssh keys).

1.2.4 Exit Status

On sucess, the program returns `0`.

1.2.5 ToDo

Date	What	Status
11,Jun,15	add switch for excluding certain stations	Waiting ...

1.2.6 Bugs

Send reports to:
Xanthos Papanikolaou mailto:xanthos@mail.ntua.gr
Demitris Anastasiou mailto:danast@mail.ntua.gr

Date	What	Status
11,Jun,15	Add help switch	Waiting ...

1.3 syncwbern52

1.3.1 Purpose

`syncwbern52.py` is a bash (Shell) script to synchronize (mirror) a local Bernese `GEN` directory, with the remote one, which can be found at AIUB's ftp server `fpt://ftp.unibe.ch/aiub/BSWUSER52/GEN/`.
Location: `src/bash/syncwbern52.sh`

1.3.2 Usage

`syncwbern52 [OPTIONS]`
at least either `--target-directory=` or `--bernese-loadvar=` must be specified as command line arguments.
Switches:

- `-t --target-directory=` /foo/bar
  Specify the (local) target directory in localhost (See subsubsection 1.3.7, Note 1).
- `-b --bernese-loadvar=` /foo/bar/LOADGPS.setvar
  Specify a Bernese source file (i.e. the file `BERN52/GPS/EXE/LOADGPS.setvar`) which can be sourced; if such a file is set, then the local target directory is defined by the variable $X/GEN. (See subsubsection 1.3.7, Note 1)
- `-o --logfile=` [logfile]
  Specify log file. Default value is /dev/null.
- `-q --quite`
  Do not show progress on screen.

- `-s --stamp-file=` [stampfile]
  Specify a file where the time stamp of current run will be written, so that the user can keep track of when the last mirroring was done.

- `-h --help`
  Display (this) help message and exit.

- `-v --version`
  Dsiplay version and exit.

1.3.3 Prerequisites

- `getopt`,

- `lftp`,

- `hash`

1.3.4 Exit Status

On sucess, the program returns `0`.
Else, the return status is >0.

1.3.5 ToDo

1.3.6 Bugs

Send reports to:
Xanthos Papanikolaou mailto:xanthos@mail.ntua.gr
Demitris Anastasiou mailto:danast@mail.ntua.gr

1.3.7 Notes

1 Note that if both `-t` and `-b` switches are used, the target directory is the one specified by the `-b` option (i.e. using the LOADGPS.setvar file).

2 All files with `.EPH` extension will be excluded from synchronization.

1.4 updatecrd

1.4.1 Purpose

`updatecrd.py` is a Python script used to update the (cartesian) coordinates in a Bernese-formated `.CRD` file, using a second `.CRD` file as reference.
Location: `src/python/updatecrd.py`

1.4.2 Usage

`updatecrd -r [.CRD-file] -u [.CRD-file] [OPTIONS]`
at least `--update-file=` and `--reference-file=` must be specified as command line arguments.

Switches:

- `-u --update-file=` [.CRD-file]
  The `.CRD` file to be updated.
- `-r --reference-file=` [.CRD-file]
  The `.CRD` file to be used as reference.
- `-s --stations=` [station1,station2]
  A comma-seperated list of stations, whose coordinates should be updated. If not specified, then all stations found in the file-to-be-updated will be updated, if they are macthed in the reference .CRD file.
- `-f --flags=` [A,W]
  Only update stations flaged with specific characters in the reference file. The flags should be specified using a comma-seperated list, e.g. `--flags=W,A`. If this switch is not specified, all matching stations will be updated regardless of their flags.
- `--include-unmatched`
  The final output CRD file, will contain all the stations recorded in the reference file (either matched or not).
- `--delete-unmatched`
  By default, all stations recorded in the file-to-be-updated will be included in the resulting file. If this (`--delete-unmatched`) switched is turned on, then stations that are not matched in the reference file will not be included in the resulting file.
- `--no-marker-number`
  If specified, the marker number will not be used when trying to match stations (i.e. `YEBE 13420M001` will be matched to `YEBE`).
- `-h --help`
  Display (this) help message and exit.
- `-v --version`
  Dsiplay version and exit.

1.4.3 Prerequisites

- `bernutils.berncrd` Python (library) module.

1.4.4 Exit Status

On sucess, the program returns `0`.
Else, the return status is >0.

1.4.5 ToDo

1.4.6 Bugs

Send reports to:
Xanthos Papanikolaou mailto:xanthos@mail.ntua.gr
Demitris Anastasiou mailto:danast@mail.ntua.gr

Date	What	Status
JUN, 2015	help and version switch not working. Fix	Waiting ...

1.5 getvmf1

1.5.1 Purpose

`getvmf1.py` is a Python script to download VMF1 (i.e. Vienna Mapping Function 1) grid files. These files are located on the web at `http://ggosatm.hg.tuwien.ac.at`. Grid files older than today are always placed in `http://ggosatm.hg.tuwien.ac.at/DELAY/GRID/VMFG/YYYY` while prediction files (including today and forward) are placed in `http://ggosatm.hg.tuwien.ac.at/DELAY/GRID/VMFG_FC/YYYY`.
The grid files hold values per 6 hours (so we have 4 files per day), at 00, 06, 12 and 18 hours and they are named as `VMFG_YYYYMMDD.HSS`, where:

- `YYYY` is the year,
- `MM` is the month,
- `DD` is the day of month and
- `SS` is the hour (00, 06, 12 or 18)

Note that the grid files previous than 01/01/2009 are compressed in `.gz` format; the routine will automatically uncompress these files.

1.5.2 Usage

`getvmf1 -y [year] -d [doy] [OPTIONS]`
at least `--year=` and `--doy=` must be specified as command line arguments.

Switches:

- `-y --year=`[YYYY]
  The `YEAR` for which the grid file is needed.

- `-d --doy=`[Day Of Year]
  The `day of year` for which the grid file is needed.
- `-r --hour=`[Hour]
  If we only want a single grid file to cover a certain time interval, then this switch can be used. The `hour` argument should be a valid integer of float in the range [0,24).
- `-o --outdir=`[output directory]
  Specify the directory where the downloaded file(s) will be saved.
- `-h --help`
  Display (this) help message and exit.
- `-v --version`
  Dsiplay version and exit.

1.5.3 Prerequisites

- `bernutils.berncrd` Python (library) module.

1.5.4 Exit Status

On sucess, the program returns `0`.
Else, the return status is >0.

The routine will also report the list of downloaded and saved files to `stdout`, e.g.

```
$> getvmf1 -y 2007 -d 10 -o /foo/bar/data
Downloaded http://ggosatm.hg.tuwien.ac.at/DELAY/GRID/VMFG/2007/VMFG_20070110.H00.gz to /foo/bar/data/VMFG_20070110.H00
Downloaded http://ggosatm.hg.tuwien.ac.at/DELAY/GRID/VMFG/2007/VMFG_20070110.H06.gz to /foo/bar/data/VMFG_20070110.H06
Downloaded http://ggosatm.hg.tuwien.ac.at/DELAY/GRID/VMFG/2007/VMFG_20070110.H12.gz to /foo/bar/data/VMFG_20070110.H12
Downloaded http://ggosatm.hg.tuwien.ac.at/DELAY/GRID/VMFG/2007/VMFG_20070110.H18.gz to /foo/bar/data/VMFG_20070110.H18
```

1.5.5 ToDo

Date	What	Status
20,Jul,15	add switch for help and version	Waiting ...

1.5.6 Bugs

Send reports to:
Xanthos Papanikolaou mailto:xanthos@mail.ntua.gr
Demitris Anastasiou mailto:danast@mail.ntua.gr

Date	What	Status
JUN, 2015	help and version switch not working. Fix	Waiting ...

Index