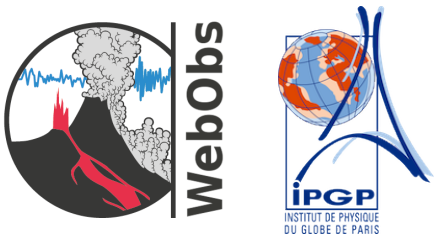


WebObs : An integrated web-based system for networks management and data monitoring in observatories

User and Administration Manual

François Beauducel and Didier Lafon
Institut de Physique du Globe de Paris Observatories

Release v2.1.5a, April 25, 2020

Contents

Introduction

Seismological and Volcanological observatories have common needs and often common practical problems for multi disciplinary data monitoring applications. In fact, access to integrated data in real-time and estimation of measurements uncertainties are keys for an efficient interpretation and decision making. But instruments variety, data sampling, heterogeneity of acquisition systems lead to difficulties that may hinder crisis management. Since early 2001, we faced this problem in the Guadeloupe volcanological observatory and we developed an operational system that attempts to answer these questions in the context of a pluri-instrumental observatory. Based on a single computer server, open source scripts (Perl, Bash, Matlab with compiled binaries, Octave, Python) and a Web interface (Apache), the system named **WebObs** proposes:

- an extended database for networks management, stations and sensors (maps, station file with log history, technical characteristics, meta-data, photos and associated documents);
- web-form interfaces for manual data input/editing and export (like geochemical analysis, repetition deformation measurements, ...);
- routine data processing with dedicated automatic scripts for each technique, production of validated data outputs, static graphs on preset moving time intervals, possible e-mail alarms;
- acquisition processes, stations and individual sensors status automatic check for technical control.

In the special case of seismology, **WebObs** includes a digital stripchart multichannel continuous seismogram compatible with international standards (SEED) associated with EarthWorm [1] and SeisComP3 [2] event database, event classification database, automatic shakemaps and regional catalog with associated hypocenter maps accessed through a user request form.

This system leads to a real-time Internet access for integrated monitoring and becomes a strong support for scientists and technicians exchange, and is widely open to interdisciplinary real-time modeling. At the time of this document, it has been set up in different observatories where it is used as one of the main operational tool: Guadeloupe, Martinique (Lesser Antilles), La Réunion (Indian Ocean), Java (Indonesia), and Paris (France).

[1] see http://www.isti.com/products/earthworm/
[2] see http://www.seiscomp3.org/

Chapter 1

Installation

1.1 Overview

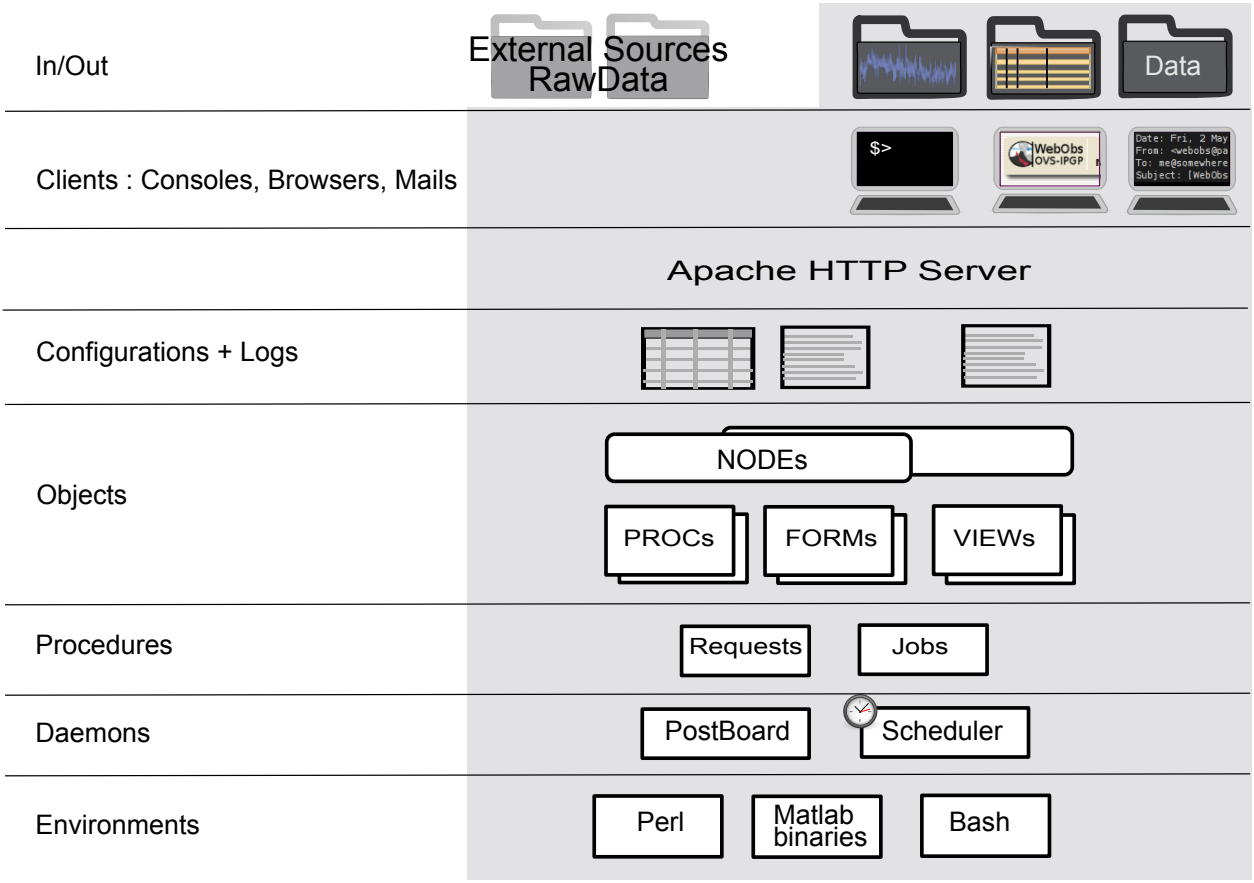


Figure 1.1: **WebObs** big picture

1.2 System requirements

1.2.1 System and package

WebObs server can run on **Linux** and **Mac OS X**.

It has been successfully installed/tested on:

- Linux 3.13.0 , x86_64 , Ubuntu 14.04 LTS
- Linux 4.13.0 , x86_64 , Ubuntu 16.04 LTS

- Linux 2.6.32, i386 , Debian 2.6.32-48squeeze4
- Linux version 3.2.0-4-amd64, Debian 3.2.51-1
- Linux version 3.2.0-4-amd64, Debian 4.6.3-14
- Linux version 4.9.0-5-amd64, Debian 4.9.65-3
- Mac OS X 10.11.5, Darwin 15.5.0

WebObs (browser) clients have been successfully tested with:

- FireFox (up to FireFox 47)
- Safari

Download the **WebObs** latest Release and **MatLab** Runtime Compiler from:
http://www.ipgp.fr/~beaudu/webobs.html

1.2.2 Software requirements

Required installations

The following softwares will be tested for existence by the **setup** installation procedure:

- Perl 5.14+ (**setup** will also check for additional Perl modules)
- Apache 2.2+
- Sqlite 3.7.9
- ImageMagick 6.6.9 (convert+identify)
- Mutt 1.5+
- MatLab MCR R2011b

Bundled software

The following lists softwares included in **WebObs** package:

- SeisComP3 (slinktool + arclink_fetch)
- JavaScript extensions:
  - JQuery
  - flot
  - markItUp
  - MultiMarkDown
  - overlib

1.3 Initial installation

You must have **root** privileges to execute **WebObs** installation.

Installation procedure:

- Choose/create your target **WebObs** directory, and **cd** to it. For demonstration purposes in this document we will use **/opt/webobs/** as the target **WebObs** directory.
- Download **WebObs** package and **MatLab RunTime Compiler**. For demonstration purposes in this document we will use **WebObs-2.1.5a.tgz** as the **WebObs** package.
- Choose/create the system's **WebObs** user+group (aka **WebObs Owner**) if you don't want **setup** procedure to create one itself.

  The **WebObs user**, and its corresponding **group**, is the required **WebObs** administration account. It must have a home directory. It will be the owner of the **WebObs** CONF,LOGS,DATA,WWW,OUTx directories. The Apache http server user will be made a member of **WebObs** user's group. The **WebObs** user can be used to launch the **WebObs** Scheduler and Postboard daemons.

- Untar **WebObs** package. This will create/populate the **WebObs** version subdirectory: **/opt/webobs/WebObs-2.1.5a**.
- Run the **setup** procedure (again, with **root** privileges):
  - must be called as **/opt/webobs/WebObs-2.1.5a/SETUP/setup**
  - **setup** will gather information/location from your system/environment, check for some dependencies (see Requirements section above), build the **WebObs** structure, optionaly customize Apache's **WebObs** Virtual Host, set required system's ownerships and access-rights, and populate your brand new **WebObs** with ready to use demonstration data and templates.
  - once completed, **setup** will display **WebObs** configuration (see **qsys** command below).
  - **WebObs** now ready, you need to activate the **scheduler**, **postboard** and (re)start Apache http server.

1.4 Upgrades

You must have **root** privileges to execute **WebObs** upgrades.

The **setup** procedure used for Initial Installation, is also used to upgrade **WebObs** to a new release. It will automatically detect that an upgrade is intended rather than a first time installation.

Upgrade procedure:

- Download **WebObs** package corresponding to the version you want to upgrade to.
- Untar **WebObs** package.
- Run the **setup** procedure (**/opt/webobs/WebObs-2.1.5a/SETUP/setup**).
- **setup** reports Version changes and important information in the **SETUP.CONF.README** file. Please read it carefully.
- **setup** will also, if you choose to do so, walk through executions of **vimdiff**s between your configuration files and corresponding non-customized **WebObs** version configurations that were changed / added.

1.5 qsys - query configuration

qsys script, also automatically executed when **setup** ends, will display your base **WebObs** configuration.

qsys example

```
                                        __ __ __
            ...oooooWooooooo..      W \ V  V /          Id: ipgp
         .oo.. ............o..o.. W  \_/\_/      Version: WebObs-2.0.0
       .oo.      ..............  ..oW   ___
     .oo.        ..............     W  / -_)         Owner: webobs[webobs]
    .W..      .  .............o.... W  \___|          Root: /opt/webobs
   .W.............o.........oWWWoo.W   _            Config: /opt/webobs/CONF/WEBOBS.rc
  .W     .    .  . ............o.....W  | |__          Logs: /opt/webobs/LOGS
  W.         ..    .WWWWo....  .     W  | '_ \
␣.W␣␣␣␣␣␣␣␣.Wo.␣.Wxxxxx....␣␣␣␣␣␣␣␣W␣␣|_.__/␣␣␣␣␣␣␣␣␣␣MCR:␣/usr/local/MATLAB/MATLAB_Compiler_Runtime/v7...
␣.W␣␣␣␣␣␣␣␣.WWW.WxxxxxxW...␣␣␣...␣␣W␣␣␣___␣␣␣␣␣␣␣␣Rawdata:␣/opt/rawdata
␣.W␣␣␣␣␣␣␣␣.WWWWWxxxxxxxxW..........W␣␣/␣_␣\␣␣␣␣␣␣␣Sefran:␣/opt/sefran
␣␣Wo..ooWxxxxxxWxxxxxxxxxo........W␣␣\___/
␣␣.xxxxxxxxxxxxxxWxxxxxxxxxW.......W␣␣␣_␣␣␣␣␣␣␣␣␣␣␣␣HTTP:␣http://webobs.ipgp
␣␣␣␣.xxxxxxxxxxxxxWWWWWxxxxxxW.....oW␣␣|␣|__␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣/etc/apache2/sites-available/webobs
␣␣␣␣␣.WxxxxxxxxxWxxWWWWWWWxxxxxWWo..W␣␣|␣'_ \                running on Apache/2.2.22 (Ubuntu)
     .oxxxxxxxxxxxxxWWWWWWWxxxxxxxWW   |_.__/
       .oWxxxxxxxxxxxxWWWWWWWWxxxWW    ___       Scheduler: not running
          .oWxxxxxxxxxxxxWWWWWxW.. W   (_-<       PostBoard: not running
             ...oWWWxxxWWWWo..     W   /__/
                                                          qsys run 2014-08-08 12:05:06
```

1.6 Configuration files syntax

WebObs configuration files (*.rc, *.conf, or *.cnf) used to customize your installation and described in this document, typically define one functionnal parameter (**key**) per line, made up of one or more associated values (**fields**). They can share the same set set of syntactic rules for parsing/interpretation:

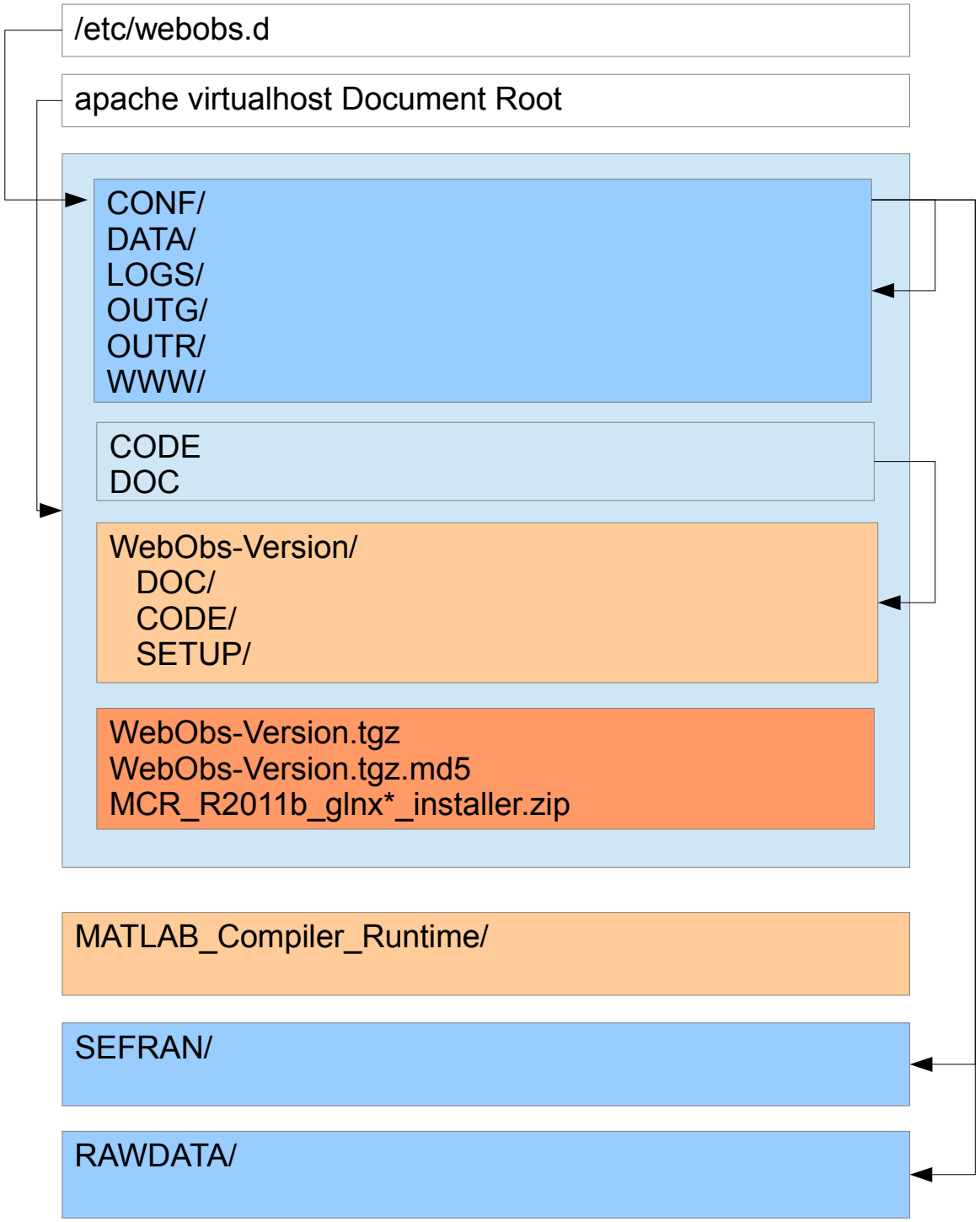


Figure 1.2: Overview of **WebObs** installed disk structure

1) In order to be parsed/interpreted according to the following rules, the files must contain a so-called 'definition line' (identified with = in column 1) as the first interpreted line. This definition line is also used to further define parsing, that comes in two (2) flavors:

=key|value	one value per key (Perl's equiv. $X{key} =>value)
=key|name1|...|nameN	multiple named values per key (Perl's equiv. $X{key}{name1} =>value)

2) Any text following a # is considered a comment and discarded.

3) Blank lines are discarded, leading and trailing blanks too.

4) Fields separator character, within interpreted lines, is |(pipe).

5) | and # characters that must belong to a field value may be 'escaped' (ie. not interpreted as separator or comment respectively) by prefixing them with a \.

6) Field value substitution (interpolation) is allowed in **=key|value** format:

${key} in value	will be replaced with the value of the key |value pair of the current file.
$WEBOBS{key} in value	will be replaced with the value of the **WebObs** main configuration **key** value.

1.7 WebObs tree

CONF/	configurations
WEBOBS.rc	main **WebObs** configuration file
***.{rc,conf,cnf}**	other configuration files
FORMS/	FORMS definitions
formname/	definitions for FORM formname
PROCS/	PROCS definitions
procname/	definitions for PROC procname
VIEWS/	VIEWS definitions
viewname/	definitions for VIEW viewname
GRIDS2FORMS/	links from PROCs to FORMS
PROC.pname.formname	->../formname
GRIDS2NODES/	links from GRIDs to NODES
PROC.pname.nodename	->../../DATA/NODES/nodename
VIEW.vname.nodename	->../../DATA/NODES/nodename
CODE/	**WebObs** code
bin/	executables
cgi-bin/	Perl CGIs
css/	HTML Style Sheets
html/	static HTML pages
icons/	HTML icons, static images
js/	javascript
matlab/	matlab (+ compiled)
shells/	bash commands
tplates/	configurations templates
DATA/	data
DB/	built-in tools data
***.DAT**	
DEM/	Digital Elevation Model files
NODES/	NODES configurations and data
nodename/	nodename
***.txt**	system features descriptions
***.cnf**	configuration
***.clb**	calibration file
DOCUMENTS/	documents
FEATURES/	features
INTERVENTIONS/	events log
PHOTOS/	pictures
DOC/	documentations
LOGS/	tools, appl, daemons logs
OUTG/	batch Procs Outputs

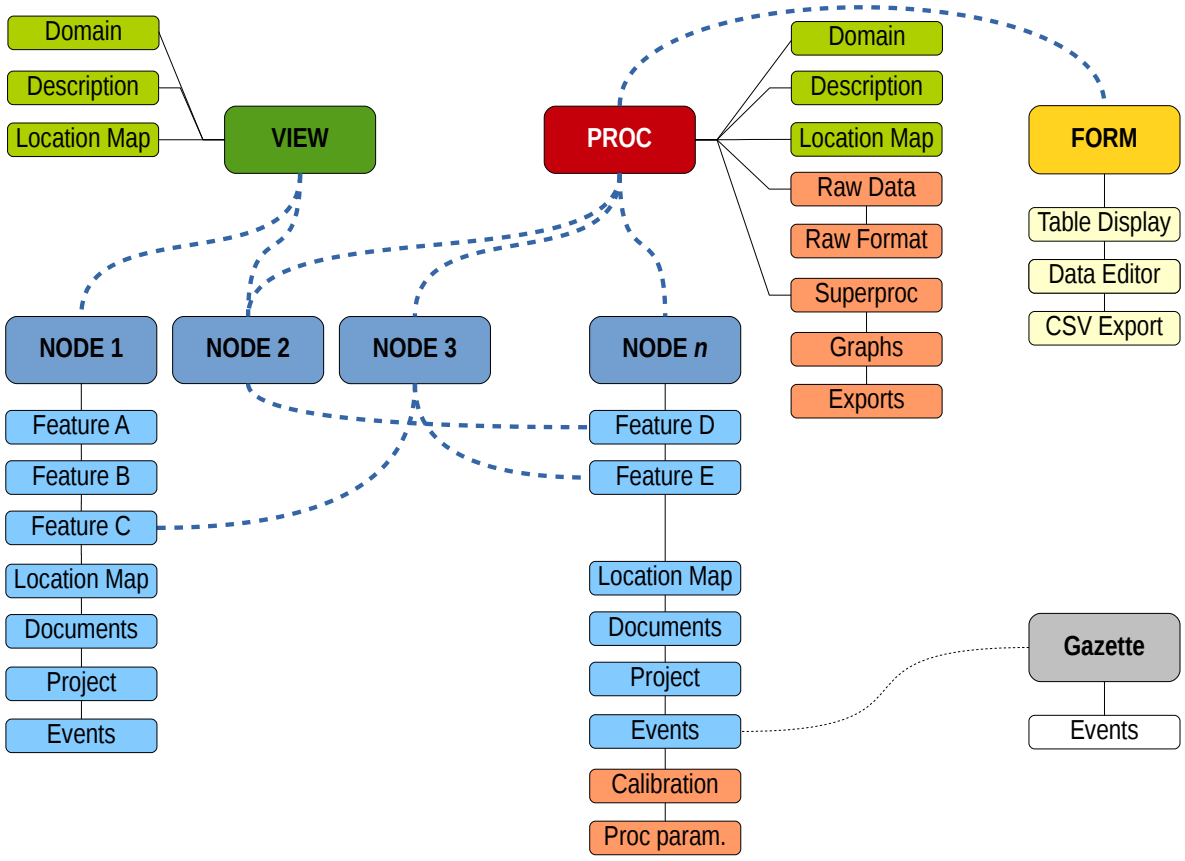
PROC.procname/	PROC procname outputs
exports/	
graphs/	
maps/	
events/	
OUTR/	Procs Requests Outputs
20140720_084340_85.199.99.199_userX/	Dated Request
REQUEST.rc	Request parameters
PROC.procname/	Requested Proc outputs
graphs/	

1.8 License

Chapter 2

Reference

2.1 Overview



2.2 Nodes: Elementary WebObs Objects

A NODE is the central **WebObs** element associated with following attributes:

- a long name (NAME) and short name (ALIAS);
- an optional short description (TYPE);
- a lifetime period with start and end dates (both optional);
- an optional location (latitude, longitude, elevation) associated with a location map (graph), Google Maps and Google Earth links;
- optional text contents for "informations", "installation" and "access";

- an optional sensor description associated with a calibration table of channels parameters;
- a list of user-defined features (also free text contents);
- attached documents, photos and diagrams;
- a project;
- events log associated with date and operator list;
- a list of associated grids (VIEWS and/or PROCS, see 2.3);
- a validity flag (for admin users);
- optional functional parameters when the NODE is associated to a PROC: data code (FID), network code (FDSN), data format (RAWFORMAT) and source (RAWDATA), time zone (TZ), acquisition period and delay, and a calibration file that describes each channel characteristics and history:
  - date and time of validity;
  - channel number, name, unit, code, S/N, offset, factor, gain, min/max values, azimuth, latitude, longitude, elevation, depth, sampling frequency, dynamic, location code.

Examples of what a NODE can be:

- an instrumental station or a part of it,
- a site or place for data sampling or measurement,
- a place of any interest,
- a mobile equipment, an instrument, a building, a vehicle, ...
- a journal board, an event description (e.g. an historical earthquake), ...

NODES.rc

```
=key|value

PATH_NODES|$WEBOBS{ROOT_DATA}/NODES

SPATH_INTERVENTIONS|EVENTS
SPATH_PHOTOS|PHOTOS
SPATH_DOCUMENTS|DOCUMENTS
SPATH_SCHEMES|SCHEMES
SPATH_FEATURES|FEATURES
SPATH_THUMBNAILS|THUMBNAILS

PATH_EVENTNODE_TRASH|$WEBOBS{ROOT_DATA}/trash/EVENTNODE
PATH_NODE_TRASH|$WEBOBS{ROOT_DATA}/trash/NODES
EVENTNODE_NOTEBOOK|NO
EVENTNODE_NOTIFY_DEFAULT|NO
# list of categories (valid keys are: grid,alias,feature,author,remote,startdate,enddate,title,comment,notebook,outcome)
EVENT_SEARCH_CATEGORY_LIST|grid,alias,feature,author,remote,startdate,title,comment,notebook,outcome
EVENT_SEARCH_MAXDISPLAY_LIST|15,50,75,100
EVENT_SEARCH_DEFAULT|alias
EVENT_SEARCH_DEFAULT2|comment


CGI_SHOW|showNODE.pl
CGI_SRCH|srchNODES.pl
CGI_FORM|formNODE.pl

THUMBNAILS_ON|ALL
THUMBNAILS_PIXH|100
THUMBNAILS_PIXV|100
THUMBNAILS_EXT|jpg

FILE_NETWORKS|$WEBOBS{ROOT_CONF}/networkcodes.csv
FILE_NODES2NODES|$WEBOBS{ROOT_CONF}/nodes2nodes.rc
FILE_TELE|$WEBOBS{ROOT_CONF}/TRANStypes.conf
FILE_POS|$WEBOBS{ROOT_CONF}/POSITIONtypes.conf
FILE_FILTER_NOTES|$WEBOBS{PATH_DATA_WEB}/Notes_Filtre_Stations.txt

# node's status (threshold in %)
SQL_DB_STATUS|$WEBOBS{PATH_DATA_DB}/NODESSTATUS.db
STATUS_THRESHOLD_WARNING|90
STATUS_THRESHOLD_CRITICAL|10
STATUS_STANDBY_VALUE|-1
```

2.2.1 Create, edit or delete a NODE

Modifying NODE is possible only through a GRID (see next section 2.3). Creation of new NODE or edition of an existing NODE deserves to USERS with edit level for the associated GRID. Deletion of a NODE is reserved to administrator level USERS.

2.2.2 Names and codes

Main code (ID)

Each NODE has a unique ID code in **WebObs** . The code can be any string of characters, but it is recommended to use a comprehensive and rational code (see appendix 5.10) when creating a new NODE. Attribution of this unique code deserves to USERS with administrator level when creating a new NODE. It cannot be modified after.

Name

The long NAME of a NODE is used for display purposes in tables and graphs. It cannot be empty. The NAME is displayed in the main GRID's table in front of the corresponding NODE.

Alias (short name)

The ALIAS is the short name of a NODE, used for display purposes in table and graphs. It cannot be empty. The ALIAS is displayed in the main GRID's table in front of the corresponding NODE.

Type

The TYPE of a NODE is an optional free string that will be used for any short description of the NODE. The TYPE is displayed in the main GRID's table in front of the corresponding NODE.

2.2.3 Lifetime and validity

Start/end dates

Start and end dates of a NODE define the period of activity of this NODE. Both or any of these two dates can be undefined or incompletely defined, while respecting the ISO 8601 standard: Date values are ordered from the most to the least significant: year, month and day; any number of values may be dropped from any of the date representations, but in the order from the least to the most significant. For example, 1976 and 2005-04 are both reduced precision valid dates.

The lifetime of a NODE has potential impact on processes that use this NODE or associated data.

Valid NODE

Independently of the lifetime period, a NODE can be valid or invalid. When invalid, a NODE is simply ignored by most of the **WebObs** processes and invisible for standard USERS. Nevertheless, if a USER has an administrator level , he will be able to see an invalid NODE in the lists and tables.

2.2.4 Grids memberships

A NODE can be associated with one or more GRIDS. See section 2.3.

2.2.5 Geographic location

Date	Type	Lat. S (WGS84)	Lon. E (WGS84)	Alt. (m)	Transverse Mercator	East (m)	North (m)	
2014-07-11	GPS meas.	7.14270 ° 07 ° 08.562 ' 07 ° 08 ' 33.7 "	107.84080 ° 107 ° 50.448 ' 107 ° 50 ' 26.9 "	2249	UTM48 WGS84:	813798	9209508	

A NODE can be georeferenced using latitude, longitude and elevation coordinates on Earth. When a NODE has valid coordinates defined, a location map will be automatically built (see below and section 3.2.3), and it will appear on associated GRID's maps (see section 2.3).

WGS84 coordinates

Latitude and longitude must be in the WGS84 geodetic reference system, and expressed in decimal form, using negative value for southern latitude and western longitude. Use the dot as decimal point (not coma).

In the NODE's display page, coordinates are presented in different formats: DDD.ddddd (decimal degrees), DDD°MM.mmm' (degrees and decimal minutes) and DDD°MM'SS.s" (degrees, minutes and decimal secondes).

The coordinates are displayed in the main GRID's table in front of the corresponding NODE. Also, the Universal Transverse Mercator (UTM) projection is computed and displayed in the NODE's page.

Timestamp and type of coordinates

This optional information provides the timestamp (date) of coordinates, and a type of positioning from a list.

POSITIONtypes.conf

```
=key|value
# Definitions of Positionning types
#
#  Cle: type used in scripts, forms, database, ...
#  Nom: corresponding text string to be displayed
#
# Cle|Nom
0|unknown
1|map
2|GPS meas.
```

Google Maps API link

Link to Google Maps API is available if the NODE has coordinates. The link will open a pop-up window. This functionality is associated with some variables in the main configuration file **WEBOBS.rc**:

WEBOBS.rc (excerpt)

```
CGI_GOOGLE_MAPS|googleMAPS.pl
GOOGLE_MAPS_LINK|1
GOOGLE_MAPS_API|http://maps.google.com/maps?file=api
GOOGLE_MAPS_API_KEY
GOOGLE_MAPS_LINK_INFO|See with Google Maps (ATTENTION: error ~20 m)
GOOGLE_MAPS_ICON|/icons/gdot.png
GOOGLE_MAPS_TYPE|G_HYBRID_MAP
GOOGLE_MAPS_WIDTH_VALUE|500
GOOGLE_MAPS_HEIGHT_VALUE|500
GOOGLE_MAPS_ZOOM_VALUE|17
```

Google Earth KML link

Link to Google Earth KML file is available if the NODE has coordinates. The link will download a local KML file, which may be open with local application Google Earth (if installed). This functionality is associated with some variables in the main configuration file **WEBOBS.rc**:

WEBOBS.rc (excerpt)

```
GOOGLE_EARTH_LINK|1
GOOGLE_EARTH_LINK_INFO|KML file for Google Earth (ATTENTION: error ~20 m)
IMAGE_LOGO_GOOGLE_EARTH|/icons/google/google_earth.gif
```

Location map

A location map with 4 different scales is automatically made and updated by LOCASTAT application (see section 3.2.3 for associated parameters), using the coordinates of the NODE.

2.2.6 Transmission

When a NODE represents, for instance, an instrumental station with data transmission, the type of transmission can defined using a reference list (below) and other NODEs can be defined as intermediate repeaters or final acquisition point.

The transmission chain is displayed with dynamic links to associated NODEs. It appears also on the location map (see section 3.2.3) and the GRID maps (see section 3.2.2).

Type

Type of transmission is defined in a list of transmission type associated with plot line style, width and color.

TRANStypes.conf

```
# WEBOBS
#
# Transmission types definitions
#
#       key: used in {show,form}NODE.pl and readnode.m
#       name: name
#       style: line style (Matlab style: -,--,-.,:)
#       width: line width
#       rgb: line color red,green,blue (values between 0 and 1)
#
=key|name|style|width|rgb
#
0|-|none|0|0,0,0
1|Analog Radio|-|1|0,0,0
2|Hard-wired|-.|1|0,0,0
3|Phone/GSM|:|1|0,0,0
4|Laser|--|1|1,0,0
5|Wireless|-|.75|0,0,0
6|Satellite|:|2|0,0,0
```

Acquisition and repeaters

The transmission chain is described by defining a list of NODE's ID, from station to acquisition.

2.2.7 Proc's parameters and data status

When associated with a PROC (see section 2.3.3), data from a NODE is used to produce outputs. Several additional parameters must be defined for this.

Network code

A NODE can be associated with a standard network code, from the International Federation of Digital Seismograph Networks (FDSN) [1]. This is mostly the case for permanent seismological stations. This code is mandatory when using Arclink or Seedlink data requests (see sections 3.14.1 and 3.14.1).

The list of network codes is also described at IRIS [2].

The official FDSN codes list can be overwritten using a local configuration file **CONF/networkcodes.csv**.

Raw format

A list of possible data format that overwrites, if selected, the `RAWFORMAT` parameter in the PROC/*.conf. This allows to associate different data format/source to the NODEs associated to a single PROC.

Raw data source

A free text string that overwrites, if not blank, the `RAWDATA` parameter in the PROC/*.conf. This allows to associate different data format/source to the NODEs associated to a single PROC.

FID

A FID code is a short string which may contain a functional code (or a coma-separated list of codes) that will be used by associated PROCs. For example, it can be the official station code for a seismic station or a GNSS receiver, while the data files will contain this code. It is generally set to the same string as the ALIAS.

Some PROCS are able to deal with a coma-separated list, as gnss: this allows to define multiple codes pointing to different data, and associate them to a single station.

FID_*

Some raw formats may need additional codes or parameters to process data from the NODE. It is possible to define specific parameters with the prefix FID_: they will be listed and editable below the main FID. Creation/edit of a FID_SUBNAME is reserved to USERS with administrator level.

[1] see http://www.fdsn.org/
[2] see http://www.iris.edu/ds/nodes/dmc/services/network-codes/

Time zone

Time zone is the offset delay from UTC, expressed in hours. Examples: $+0$ is UTC, -4 is Lesser Antilles time.

Acquisition period

Acquisition period defines the normal time interval between two data sample, expressed in days. Examples: 1/24 corresponds to 1 sample per hour, 1/86400 is 1 Hz sampling. This value might be used to compute average acquisition performance rating (see below).

Acquisition delay

This value defines the maximum delay allowed between the last available data and the present time, expressed in days. This delay is used to determine the NODE's status (see below).

Status and link to data

If the associated PROC is set to compute the NODE status, the following information is displayed:

- last status check timestamp;
- sampling rate performance (in %): number of valid samples over the last time period, compared to theoretical acquisition period (see above);
- status (in %): existence of valid samples in the last time delay for all channels.

The status informations are displayed in the main GRID's table in front of the corresponding NODE.

If the associated PROC produces output graphs, direct links to each outputs are available.

Channels parameters (calibration file)

Channels

Date	Time	Ch. Nb	Ch. Name	Unit	S/N	Ch. Code	Offset	Calib. Fact.	Gain	Min. Value	Max. Value	Azim. (°N)	Latitude (°)	Longitude (°)	Elevation (m)	Depth (m)	S.F. (Hz)	Dyn. (bit)	LC
2008-06-21	00:00	1	Eastern	m		E	0	1	1			0	16.27268	-61.76509	618	0			
2008-06-21	00:00	2	Northern	m		N	0	1	1			0	16.27268	-61.76509	618	0			
2008-06-21	00:00	3	Vertical	m		U	0	1	1			0	16.27268	-61.76509	618	0			

Calibration formula: D = d × (Calib. Fact.) × (Gain) + (Offset)

For most of PROCS, it is mandatory to define a list of data channels. This can be done through a "calibration file" that contains description and detailed parameters of the NODE's data channels: date and time of validity, name, unit, S/N, functional code, gain and multiplier factors, offset, min/max values, sensor azimuth, location and depth, sampling frequency, digital dynamic and "location code" (SEED standard). Each channel must have one or more series of parameters.

The date and time of validity applies until real-time or next line of parameters for this channel number.

Each PROC might be associated to a selection of channels. Active channels appear in bold font, while inactive appear in gray.

2.2.8 Features

Each NODE can have user-defined features which are free text contents (HTML and Wiki syntax allowed) that appear in the main table as a series of table cells.

List of user-defined features

Feature name is limited to short names and must avoid special characters. Edit the list of features to be displayed using the NODE configuration form. Features with void content won't be displayed for USERS without edit or admin rights.

Node-features-nodes association list

It is possible to associate a NODE's feature to an other NODE as its feature's children. An automatic link will be displayed in the feature's table cell before the text content of the feature. In the children NODE page, the parent feature will appear with automatic name "feature of" and a link to the parent NODE.

Configuring the node2node association list is made by editing a configuration file containing a 3-field definition (pipe delimited):

nodes2nodes.rc (excerpt)

```
PARENT_NODEID|featurename|CHILDREN_NODEID
```

2.2.9 Installation, information and access

These 3 fields are free-length text content. Wiki syntax allowed.

2.2.10 Photos, diagrams and associated documents

Three different types of documents can be uploaded and associated to a NODE:

- Photos: any JPEG file that will be displayed as thumbnail with a link to full screen mode;
- Diagrams: any image or picture file (JPEG, GIF, PNG, PDF, ...) that will be displayed as thumbnail with a link to the full resolution image;
- Documents: any file (PDF, DOC, TXT, ...) that will be downloadable and sometimes displayed through the navigator.

2.2.11 Project

A NODE may have one (and only one) PROJECT description: a free-length text file, accepting MultiMarkDown syntax, a list of users and possible associated photos. Project is a special EVENT without date.

2.2.12 Events

A NODE can be associated to dated events. A NODE event has following characteristics:

- start date and time;
- end date & time;
- a title;
- a content of free-length text,
- author(s) from the list of **WebObs** users;
- selected feature (from the node's feature list);
- sensor/data outcome flag;
- notebook number (optional);
- notebook forward flag (optional);
- associated photos (image files).

An event can include one or more sub-events (children). Event files accept the MultiMarkDown syntax. See EVENTS LOGGING section (2.4) for a description of Nodes and Grids events coding.

2.3 Grids: networks of nodes

A GRID is a group of NODES. Each NODE can be associated to one or multiple GRIDS. There is two kind of GRIDS:

- A VIEW is a list of NODES we simply want to group to be seen or accessed together,
- A PROC is a list of NODES associated to a common data processing that produces outputs (graphs and/or elaborated data), and/or optional editable data FORMS.

GRIDS are presented through a sorted table allowing access to dedicated page for each GRID with:

- domain: a category used to group similar GRIDS;
- purpose (text content);
- specifications: operator owner, number of NODES, type, optional external link;
- list of NODES: a complete table with alias, names, location, start/end dates, type, project, status (for active PROC); ...
- an interactive location map (if at least one NODE is georeferenced);
- informations (text content);
- references (text content);
- a graphical representation of NODES links with associated GRIDS.

Text content sections are editable with Wiki syntax possibilities.

GRIDS.rc

```
=key|value

# title for default "all grids" display (listGRIDS.pl)
SHOW_GRIDS_TITLE|Grids

# grids table options
SHOW_TYPE|Y
SHOW_OWNER|Y

# nodes table options
DEFAULT_COORDINATES|latlon
DEFAULT_NODES_FILTER|all
DEFAULT_PROJECT_FILTER|on
DEFAULT_PROCPARAM_FILTER|on

# ===== do not edit below this line =====
# (parameters only for backward compatibily purpose)

CGI_SHOW_GRID|showGRID.pl
CGI_SHOW_GRIDS|listGRIDS.pl

PATH_GRIDS|$WEBOBS{ROOT_DATA}/GRIDS

SPATH_INTERVENTIONS|EVENTS
SPATH_PHOTOS|PHOTOS
SPATH_DOCUMENTS|DOCUMENTS
SPATH_SCHEMES|SCHEMAS
SPATH_FEATURES|FEATURES
SPATH_THUMBNAILS|THUMBNAILS

PATH_EVENTGRID_TRASH|$WEBOBS{ROOT_DATA}/trash/EVNTRASH

DESCRIPTION_SUFFIX|_description.txt
PROTOCOLE_SUFFIX|_protocole.txt
BIBLIO_SUFFIX|_bibliography.txt
```

2.3.1 Domains

A DOMAIN is a generic category that is used to group GRIDS together: in the GRID's table presentation and in most of the page links between GRIDS and NODES. By default, DOMAINS are defined as a list of standard scientific methods as used in volcanological observatories: Seismology, Deformations, Geochemistry, Imagery, ... but these categories can be changed to better fit the **WebObs** needs. For example, if the **WebObs** has to handle multiple targets (like volcanoes), you might want to make the DOMAINS as a list of these targets.

Changes to DOMAINS list can be done by administrator by editing the sqlite database. New domains can be easily added by inserting an entry into the `domains` table. To remove an existing DOMAIN, you must remove any GRID associated to this DOMAIN first (this can be done using the GUI of GRID configuration editor), then delete manually the entry in the database.

WEBOBS.rc (excerpt)

```
SQL_DOMAINS|${ROOT_CONF}/WEBOBSDOMAINS.db
SQL_TABLE_DOMAINS|domains
SQL_TABLE_GRIDS|grids2domains
```

2.3.2 Views

A VIEW is a simple group of NODES.

To create a new VIEW, go to any GRID table page (menu GRIDS, VIEWS or PROCS) and click on the edit icon in the Name header of table, enter an unique short name and click Edit. This will open the GRID configuration editor. The form allows to freely edit the parameters file (key/value pairs), to select the Domain and to associate/disassociate NODES.

VIEW.DEFAULT template

```
=key|value
# Generic VIEW template for use with cgi-bin/formGRID.pl

NAME|Generic View
OWNCODE|MyCompany
TYPE|A Group of nodes
URL|http://myview.org
COPYRIGHT|MyView

# node parameters (for display and maps)
NODE_NAME|node
NODE_MARKER|^
NODE_SIZE|15
```

```
NODE_RGB|red
NODE_FONTSIZE|12

# set to YES to use SRTM1 30m resolution tiles
DEM_SRTM1|NO

# optional user-defined DEM (Arcinfo format, lat/lon) overwriting SRTM/ETOPO default
#DEM_FILE|$WEBOBS{PATH_DATA_DEM}/highresdem_latlon.asc
#DEM_TYPE|LATLON
#DEM_COPYRIGHT|DEM: myDEM

# additional zoomed map(s) using fixed axis limits: LON1,LON2,LAT1,LAT2
#MAP1_XYLIM|
#MAP2_XYLIM|
# do not show node alias in zoom map(s)
NODE_SUBMAP_ALIAS|N
```

All the following parameters are valid for GRIDS, that is any VIEW or PROC (see the next section).

GRID name and attributes

- NAME: Character string for the GRID long name. This name will appear in the GRID page and maps of NODES.
- OWNCODE: Character string for the GRID's owner name. This name will be displayed in the GRIDS tables and GRID summary, if SHOW_OWNER key is true in **GRIDS.rc**. For backward compatibility, the key might contains a short code with corresponding long names in **CONF/OWNERS.conf**.
- TYPE: Character string giving a short attribute to the GRID, if SHOW_TYPE key is true in **GRIDS.rc**.
- URL: Optional URL related to the GRID. Any valid URL is allowed, e.g. an external web link.
- COPYRIGHT: Optional character string for the copyright of the GRID.

NODES attributes

- NODE_NAME: Short name of a generic NODE. This name will be used instead of "node" in tables and maps.
- NODE_MARKER: Type of marker used to plot NODES on maps. See valid markers in table 5.3.
- NODE_SIZE: Size of the NODE marker for maps (in points).
- NODE_RGB: Color of the NODE marker for maps: it can be a color name string (see **CODE/matlab/htm2rgb.m** for a list of standard colors), or a vector of 3 scalars (red,green,blue), each value between 0 and 1.
- NODE_FONTSIZE: If not empty, font size to display NODE's alias names on maps besides the marker. A

Digital Elevation Model (DEM)

For any VIEW or PROC that needs topographic data, **WebObs** will use either SRTM3, SRTM1, ETOPO5, ETOPO1 or user-defined DEM file. SRTM tiles will be downloaded automatically from internet, as needed by the PROC, and stored in the PATH_DATA_DEM local directory (see **WEBOBS.rc**). ETOPO5 is available in the **WebObs** package offline. ETOPO1 must be downloaded manually from the web. The selection between these different sources respects the following policy:

1. SRTM3 is the default, 90m resolution worldwide DEM, available through 1×1^o tiles. It is convenient for any maps from about 20km to 500km width. **WEBOBS.rc** defines a limit of SRTM_MAX_TILES as maximum amount of tiles to avoid memory issues (default is 25). If the covered area needs more than this limit, it will switch automatically to ETOPO with a warning in the logs.
2. SRTM1, a 30m resolution DEM can be activated setting DEM_SRTM1 to Y in the PROC's configuration file. Each tile is 1×1^o and 9 times bigger than SRTM3. This is convenient for smaller areas less than 20km width. **WEBOBS.rc** defines a limit of SRTM1_MAX_TILES maximum amount of tiles to avoid memory issues (default is 4). If the covered area needs more than this limit, it will switch automatically to SRTM3 with a warning in the logs.
3. if SRTM3 or SRTM1 limits are exceeded for any large area, **WebObs** will use ETOPO worldwide DEM with bathymetry. ETOPO5 is 5 arc-min resolution (about 9km) which is convenient for maps greater than 2,000km width. If ETOPO1 has been installed (which is recommended at installation), it offers 1 arc-min resolution (about 1.8km) which is convenient for maps width greater than 500km. So with SRTM and ETOPO1 installed, most situations will produce nice basemaps.
4. if the area is less than about 5km width, even SRTM1 is not enough to produce decent maps. In that case it is more appropriate to use a local DEM file (in ArcInfo ascii format, elevation values in meters) that can be defined in the PROC's configuration file with DEM_FILE as full path of the file, and DEM_TYPE as LATLON (preferred) or UTM (might takes more time). If this DEM does not cover the whole area, the PROC will switch automatically to SRTM/ETOPO data. Nevertheless, it is possible to force the use of your local DEM by setting DEM_FORCED to Y but when the area exceeds the DEM data, map will be surrounded by gray zones where there is no data available. A last option DEM_COPYRIGHT allows to specify the DEM copyright.

2.3.3 Procs

A PROC is a list of NODES that are associated to a specific and common data processing called SUPERPROC. It has the same basic parameters as VIEWS (see previous section 2.3.2) plus additional settings that depend on the process you want to achieve. **WebObs** proposes some built-in SUPERPROCS (see chapter 3), each of them are specific. There is on default SUPERPROC called 'genplot' which is a simple plot of time series data (see section 3.4).

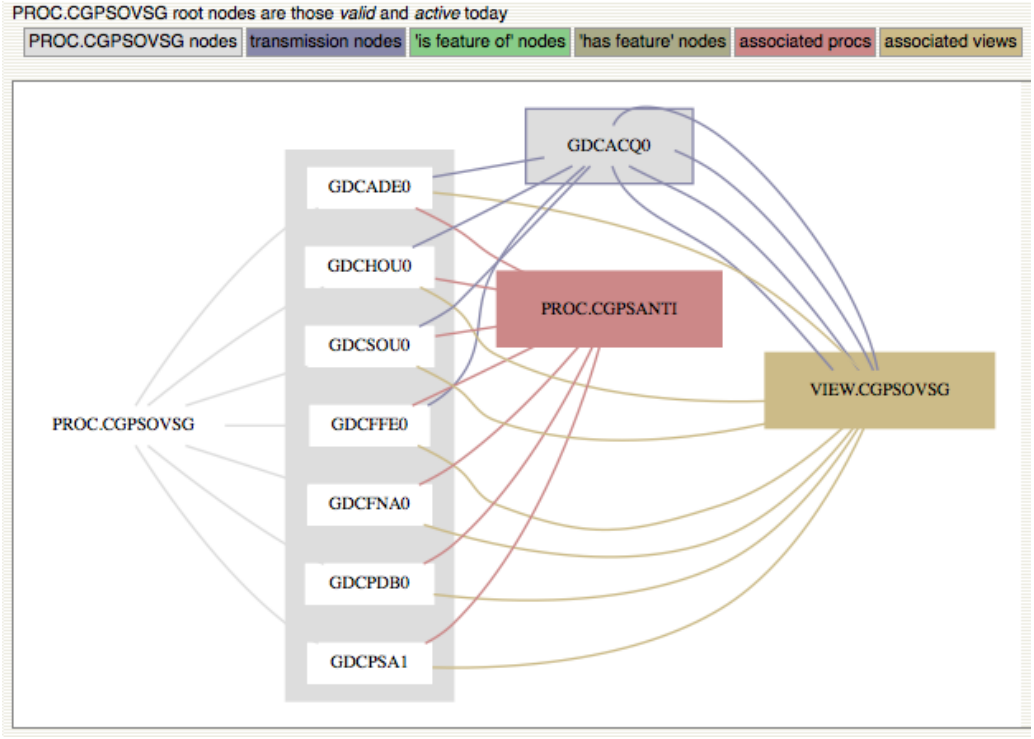
To create a new PROC, go to any GRID table page (menu GRIDS, VIEWS or PROCS) and click on the edit icon in the Name header of table, select "PROC: Generic time series" in the Grid type list, enter an unique short name and click Edit. This will open the PROC configuration editor. The form allows to freely edit the parameters file (key/value pairs), to select the Domain, optional FORM and to associate/disassociate NODES.

A FORM is a web interface for manual data input, edit, display and export. A FORM is an optional tool of a PROC that defines a specific database format that contains data from associated NODES. Some SUPERPROCS are working specifically with FORM databases.

2.3.4 Events and Projects

A GRID can have events and project like for NODES: see sections **??** and **??**.

2.3.5 TMap: GRIDS diagram



2.4 Events Logging

Nodes and Grids can have EVENT. These events are free timestamped text files (Wifi syntax being supported), editable by users having at least Write authorization on the corresponding NODE or GRID. Events live in specific repositories (known as INTERVENTIONS), and are managed through functions of the **CODE/cgi-bin/WebObs/Events.pm** module and **CODE/cgi-bin/vedit.pl**. Their filenames reflect both their Node or Grid membership and their timestamp.

An event may also have attached images (PHOTOS) and/or sub-events: both are collectively referred to as EVENT EXTENSIONS. Subevents are themselves events, thus building up a tree structure for each event.

Events base directories (interventions)

```
$GRIDS{PATH_GRIDS}/gridtype/gridname/$GRIDS{SPATH_INTERVENTIONS}/
$NODES{PATH_NODES}/nodename/$NODES{SPATH_INTERVENTIONS}/
```

Events 'trash' directories (for deleted events)

```
$NODES{PATH_EVENTNODE_TRASH}
$GRIDS{PATH_EVENTGRID_TRASH}
```

Events files and extensions naming conventions

```
event_file       :=  event.txt
event_extensions :=  event/
event            :=  name_YYYY-MM-DD_HH-MM{_v}  |  name_YYYY-MM-DD_NA{_v}
name             :=  { gridname | nodename }
v                :=  so-called version number (automatically generated to make event name unique)
NA               :=  "NA" for unknown/undefined HH-MM
```

Special event file: the Project

```
only one allowed per Node or Grid :

project_file     :=  name_Projet.txt
```

Unfolded example for node NODEA INTERVENTIONS

```
$NODES{PATH_NODES}/NODEA/$NODES{SPATH_INTERVENTIONS}/
        NODEA_Projet.txt
        NODEA_2001-01-01_20-00.txt          Event 2001-01-01_20-00 file
        NODEA_2001-01-01_20-00/             Event 2001-01-01_20-00 extensions
                PHOTOS/                             Event 2001-01-01_20-00 photos
                        *.[jpg,pdf]
                        THUMBNAILS/
                NODEA_2002-02-02_02-02.txt          subEvent 2002-02-02_02-02
                NODEA_2002-02-02_02-02/             subEvent 2002-02-02_02-02 extensions
                        PHOTOS/                             subEvent 2002-02-02_02-02 photos
                                *.[jpg,pdf]
                                THUMBNAILS
                        NODEA_2003-03-03_03-03.txt          subsubEvent 2003-03-03_03-03
                        NODEA_2010-02-02_22-30.txt          Event 2010-02-02_22-30
```

2.5 Web pages

2.5.1 Main menu

2.5.2 Home page

2.5.3 Wiki pages

2.5.4 Tools pages

2.5.5 Application pages

Chapter 3

Built-in applications

3.1 The Gazette

3.1.1 Overview

The GAZETTE is the observatory's logbook and calendar. It is a collection of **articles** in a relational DB table. An article is defined by:

- a Start and an End timestamp,
- a Category (from a predefined/editable list),
- a list of Users (WebObs-registered and additional),
- a Place,
- a Subject.

The GAZETTE has its own html interfaces for visualization and edition, implemented in **CODE/cgi-bin/WebObs/Gazette.pm** and **CODE/cgi-bin/Gazette.pl** modules.

3.1.2 Configuration

The GAZETTE is defined by a configuration file, whose location is pointed to by the main WebObs configuration variable GAZETTE_CONF.

WEBOBS.rc (excerpt)

```
GAZETTE_CONF|${ROOT_CONF}/Gazette.rc
```

Gazette.rc

```
=key|value

DB_NAME|$WEBOBS{PATH_DATA_DB}/Gazette.db
CATEGORIES_FILE|$WEBOBS{ROOT_CONF}/Gazette_categories.conf

BANG|2001
TITLE|Gazette
FUTURE_YEARS|2

# defines group(s) of "active" users available for new events
ACTIVE_GID|+ADMIN,+DUTY

EMPTY_SELECTION_MSG|No match found
CALENDAR_WEEKNUMBER|VERTICAL
SHOW_BYDATE_ONGOING|TEXT
CALENDAR_TRUNCLENGTH|25

# DEFAULT_VIEW values: calendar, datelist or categorylist
DEFAULT_VIEW|calendar
# DEFAULT_DATE values: today, tomorrow, yesterday, allyear, thisweek
DEFAULT_DATE|thisweek
# see CATEGORIES_FILE for available keys
DEFAULT_CATEGORY|
```

Among other parameters, GAZETTE_CONF file points to the Gazette DataBase (DB_NAME) and the Gazette categories definitions (CATEGORIES_FILE).

The GAZETTE also uses the WebObs authorization mechanism with one resource associated to each category, plus one generic resource representing all categories. These resources belong to the authorization 'misc' type (authmisc table), and are identified as "GAZETTE*categoryKey*" (eg. GAZETTEMissions, GAZETTEField).

Gazette_categories.conf

```
# Gazette articles categories
#
# - Key    : value used by scripts, forms fields, database, ...
#            ATT: a single word with NO special character nor space
#            ATT: reserved keys = "ALL"
#            Key is also used as prefix for authorization resource names (GAZETTE_key)
# - Name   : long name for user interface
# - RGB:     color of text display (HTML/CSS format)
# - RGBlight: corresponding color used as background
# - Auto: category is automatic (not selectable in form)
# - Format: any string from : ndol, ndlo, ldon, dlon, andol, or adon
#            encoding fields order of appearance, where:
#            n = name (operator), d = date, o = comment, l = place, a = others
#
# Allowed modifications:
#        - modify Name, RGB or Level values
#        - add new keys
#
# Forbidden:
#        - modify/delete existing Keys (or corresponding Keys must be updated in
#          the database file)
#        - duplicate reserved Keys, or delete them
#
=Key|Name|RGB|RGBlight|Auto|Format
ALL|All|\#000000|\#7F7F7F|1
Duty|Duty Scientist|\#EFC700|\#F6E27F|0|ndol
Missions|Staff Missions|\#DD5555|\#FFD4D4|0|ndlo
Holiday|Off / Holiday|\#DD5555|\#EDA9A9|0|ndol
Field|Field work|\#00BB00|\#7FDC7F|0|ldon
Meeting|Meeting / Officials|\#007F7F|\#7FBEBE|0|dlon
Media|Medias|\#5555FF|\#D4D4FF|0|dlon
Outreach|Outreach|\#5555FF|\#D4D4FF|0|dlon
Teaching|Teaching|\#5555FF|\#A9A9FE|0|dlon
Visitor|Visitors|\#B200FF|\#D87FFE|0|andol
Training|Students Training|\#B200FF|\#FF7FFF|0|adon
Building|Buildings|\#777777|\#CCFF66|0|ldon
Misc|Misc|\#777777|\#BABABA|0|dlon
Event|Events|\#ED7C3F|\#DE8F65|1|ldon
```

The GAZETTE also includes holidays definitions that can be adapted to any country using a specifications file pointed to by `FILE_DAYSOFF` in the main WebObs configuration WEBOBS.rc .

WEBOBS.rc (excerpt)

```
FILE_DAYSOFF|${ROOT_CONF}/Holidays.conf
```

Holidays.conf (example)

```
# List of some international public holidays
#
# Configuration file read by subfunction "readFeries" in WebObs::Dates
# File defined by the WEBOBS.rc key FILE_DAYSOFF
#
# - EXP : expression used by the Linux function 'date -d "EXP"' to determine the date
#         with possible parameters:
#           $Y = current year
#           $PQ = Easter Sunday
#           $PQ n = n days from Easter Sunday
# - Name : name of the day
#
#
# EXP|Name
$Y-01-01|New Year
$PQ 1|Easter Monday
$PQ 39|Ascension
$PQ 50|Pentecost Monday
$Y-08-15|Assomption
$Y-11-01|All Saints' Day
$Y-12-25|Christmas
```

3.1.3 Display/Edit Gazette

CODE/cgi-bin/Gazette.pl is the html interface used to request and display the Gazette. The greyed banner of the page is a selection form used to specify what and how Gazette articles should be displayed:

- What : Date or Date-range selection
  - **via a monthly calendar**
  - **using predefined periods**
- What : Category and Filter
- How : Presentation
  - **Calendar**: a calendar-type weekly table,
  - **List by categories**: a list sorted by categories,
  - **List by date**: a list sorted by chronological start-date,
  - **iCalendar**: iCal format text,
  - **dump**: reserved for administrators, raw DB-table display

The banner also displays a 'Create Article' button to trigger the edition of a new article. The form allows to edit the fields for the new event. Note the list of author names is limited to valid users (see section 4.1.2) or to any users in the group defined by `ACTIVE_GID` configuration key.

Note: 'Event' category is not editable through html interface.

Developer's note: A Gazette display html code can also be imported into other pages using the WebObs/Gazette.pm Show function. See description in Gazette.pm perldoc and Welcome.pl page as an example.

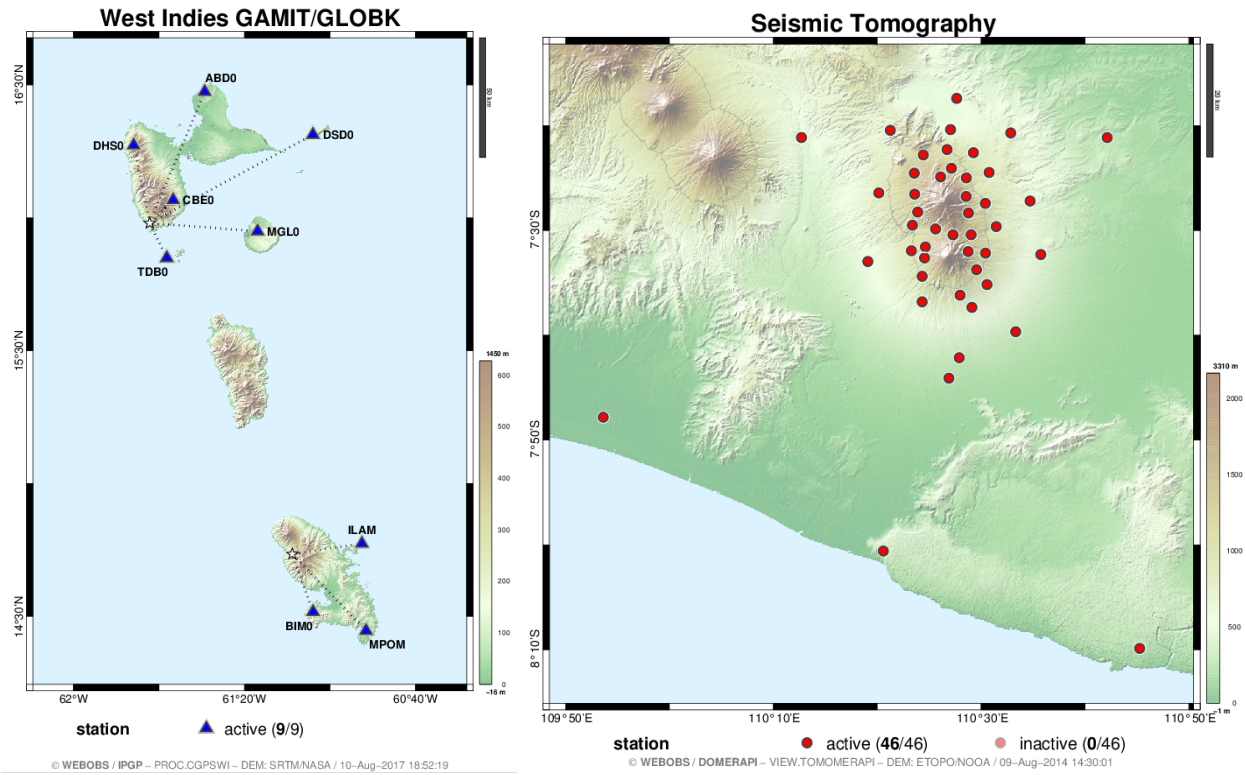


Figure 3.1: Examples of GRID's maps created by GRIDMAPS.

3.2 GRIDMAPS and LOCASTAT: maps of georeferenced nodes

3.2.1 Overview

Any NODE may be associated with a location with geographic coordinates. A PROC might use these coordinates for a specific processing, e.g. for tilt or gnss deformation modelling, or to produce dedicated maps. There is also two built-in applications that will use these coordinates: GRIDMAPS and LOCASTAT.

3.2.2 GRIDMAPS

When a NODE is associated to a GRID, **WebObs** will automatically produce maps with these georefenced NODES. Background basemap will be an illuminated elevation model of the area, using SRTM, ETOPO or user-defined DEM (see section 2.3.2 for DEM policy).

Configuration

Each GRID map is depending on two configuration files: one is common for all grids to define the basemap parameters:

GRIDMAPS.rc

```
=key|value

NAME|Networks
# --- basemap options
# paper size (maximum width/height)
PAPERSIZE_INCHES|10
# density (in pixels per inch) for PostScript to PNG convertion
DPI|100
# basemap DEM rendering options (see CODE/matlab/dem.m documentation)
LINEWIDTH|1.5
LANDCOLORMAP|landcolor
#SEACOLORMAP|seacolor
SEACOLORMAP|[0.7,0.9,1]
COLOR_LIGHTENING|2
LIGHT_AZIMUTH|-45
LIGHT_CONTRAST|1
ZCUT|0.1

# oversampling in case of too small image
OVERSAMPLING|500

# minimum size of map around NODES
MIN_SIZE_KM|5
```

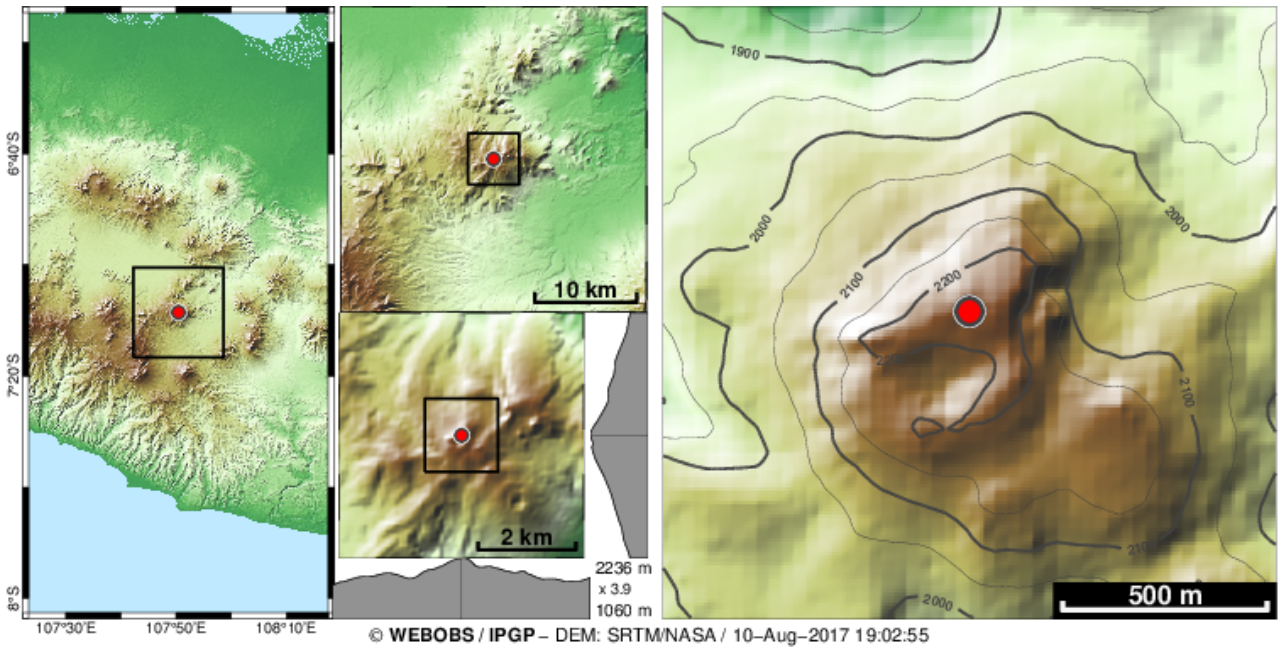


Figure 3.2: Example of NODE's map created by LOCASTAT.

```
# maximum X/Y ratio in case of automatic sizing around NODES
MAX_XYRATIO|1.5
BORDER_ADD|0.1
DEM_SRTM1|N

# --- contour lines
CONTOURLINES|Y
CONTOURLINES_RGB|.5,.5,.5
CONTOURLINES_ZERO_LEVEL|N
CONTOURLINES_MINOR_LINEWIDTH|.1
CONTOURLINES_MAJOR_LINEWIDTH|1
CONTOURLINES_LABEL|Y

# adds transmission lines between NODES (if defined)
PLOT_TRANSMISSION|Y

# list of GRID keys available for edit in request
REQUEST_GRID_KEYLIST|NODE_SIZE,NODE_RGB,NODE_FONTSIZE,NODE_MARKER
SUBMIT_COMMAND|$WEBOBS{JOB_MCC} gridmaps -
SUBMIT_RESOURCE|gridmaps
```

and the other is the configuration file of the GRID itself to define the parameters associated to NODES (see for example the VIEW example in section 2.3.2).

Activation

GRIDMAPS is set by default when installing **WebObs** . It should be active at the first scheduler start. Parameters are as follows:

jid:	gridmaps
res:	gridmaps
xeq1:	$WEBOBS{JOB_MCC} gridmaps
interval:	86400
logpath:	gridmaps
valid:	Y

You may check if it runs correctly in the Scheduler Runs page (see Section 4.3).

3.2.3 LOCASTAT

The location map appears in each NODE page if coordinates are defined. The map is automatically made and updated by LOCASTAT application. Maps are updated when the map timestamp is older than NODE's configuration file. So to force the update of a location map, you may modify any parameter in the NODE configuration, typically the coordinates values or positioning date.

The map contains 4 maps at different scales, all centered on the NODE position, from left to right with a progressive zoom

effect. Basemaps are built from the free worldwide topography data SRTM [1] and specific colormap and rendering methods. It is possible to specify a user-defined DEM (Digital Elevation Model) for the highest resolution scale map (right frame).

LOCASTAT.rc

```
=key|value

# --- wide-scale left frame
FRAME0_WIDTH_KM|100

# intermediate-scale middle-top frame
FRAME1_WIDTH_KM|30
FRAME1_SCALE_KM|10

# --- intermediate-scale middle-bottom frame
FRAME2_WIDTH_KM|5
FRAME2_SCALE_KM|2
FRAME2_RESAMPLING|200

# --- high-resolution right frame (in UTM)
FRAME3_WIDTH_KM|1.5
FRAME3_SCALE_KM|.5
FRAME3_RESAMPLING|400
FRAME3_SRTM1|NO

# --- optional user-defined DEM frame (right) => applies to all nodes
# DEM grid filename (ArcInfo format)
FRAME3_DEM_FILE|$WEBOBS{PATH_DATA_DEM}/exampledem.asc
# DEM coordinate system: UTM or LATLON
FRAME3_DEM_TYPE|UTM
COPYRIGHT2|MyDEM

DPI|80
LINEWIDTH|1.5
COLORMAP|landcolor
COLOR_LIGHTENING|1.2
LIGHT_AZIMUTH|-45
LIGHT_CONTRAST|1
SEACOLOR|[0.7,0.9,1]

PLOT_TRANSMISSION|Y
```

Activation

LOCASTAT is set by default when installing **WebObs** . It should be active at the first scheduler start. Parameters are as follows:

jid:	locastat
res:	locastat
xeq1:	$WEBOBS{JOB_MCC} locastat
interval:	3600
logpath:	locastat
valid:	Y

You may check if it runs correctly in the Scheduler Runs page (see Section 4.3).

3.2.4 DEM basemaps

GRIDMAPS and LOCASTAT will use automatic mapping with the same standards and options as for GRIDS maps, i.e., using either SRTM3, SRTM1, ETOPO5, ETOPO1 worldwide elevation data or user-defined DEM file. See section 2.3.2 for details.

[1] see http://www2.jpl.nasa.gov/srtm/

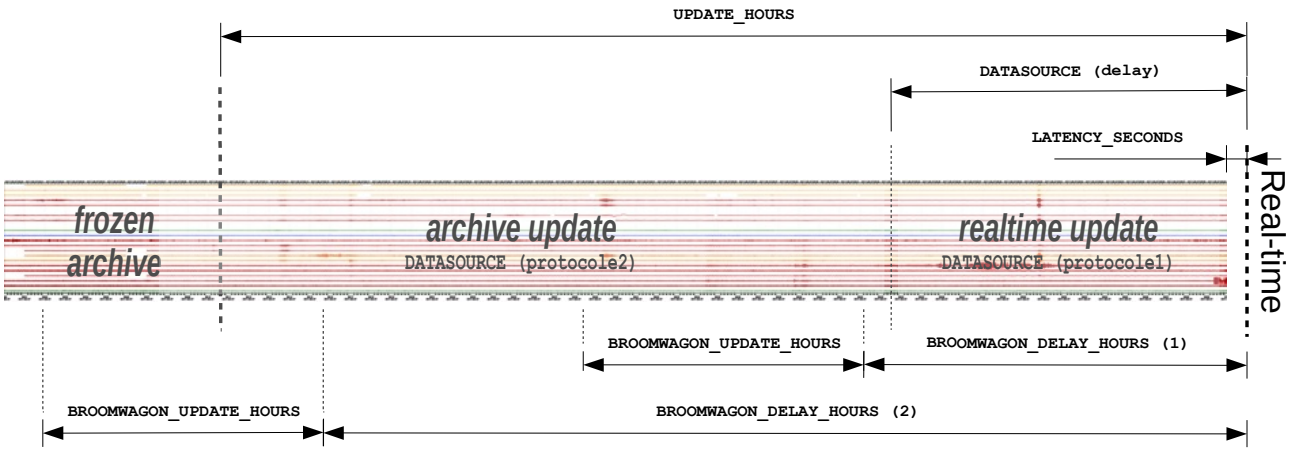


Figure 3.3: Schematic diagram of SEFRAN3 main parameters.

3.3 SEFRAN3/MC3: seismic chart and bulletin

3.3.1 Overview

The SEFRAN3 is a graphical interface to operate seismic data flux, manual and semi-automatic detection of events, and earthquake catalog bulletin management.

The name "SefraN" is a contraction of *Sefram Numérique*; it comes from a 70's paper strip-chart recording instrument from the French factory *SEFRAM*®, used during decades in French observatories. Starting 2001, the system has been replaced by a numerical simulation using local data files archives in SUDS format. SEFRAN3 is the third version of SefraN which now uses data flow from several data flow protocols (Seedlink, Arclink, FDSNWS, ...).

A second tool called "Main Courante" hereafter named MC3, is a database of seismic events that is linked to SEFRAN3 interface and possibly to external earthquake catalogs like a local SeisComP3 or any FDSN-webservice compatible database like EMSC or USGS.

The SEFRAN3 works with a fixed selection of channels (up to 15) coming from a single server source, that will be displayed together as time series using 2 different time scales (normal and high) that simulate the paper speed. SEFRAN3 has 4 different GUI:

1. Main page showing hourly thumbnail images of seismic signals for a given period of time. Identified events are shown as overlaying colored tags. Bottom part of the page is showing realtime state of each channel with some statistics on the data quality. There is two display modes:
   - real-time automatic refreshing page, for the last X hours/days of data;
   - any date selection for one or more days.
2. One hour display of full-resolution image in a wide window that can be spanned and scrolled through time.
3. A form showing a single event at high-speed time resolution of seismic signals, with possibility of editing data and submitting to the database.
4. A detailed table of events list with date selection, filters, and dynamic graphs.

Notice: SEFRAN3 is using extensively the network connection (for data flux) and external programs like **arclink_fetch** (from *SeisComP3*), **slinktool** (from *IRIS*) and **convert** (from *ImageMagick*). If you experience any trouble, check first the network, the data availability on servers, and the third-party programs. You might add the variable `DEBUG|Y` in the configuration file to make the logs more verbose.

3.3.2 SEFRAN3 installation

SEFRAN3 is based on 1-minute images of seismic traces. It works within a loop that will end after a minimum duration run. Each loop will scan the existing images on disk, and make new images (real-time first, then look at older periods for gaps) or update them using broom wagons that check the data completeness after some time delay. There is also a short delay of few seconds to take into account the data flux lateness from real-time, and a longer delay to switch from SeedLink data request (convenient for real-time data flux) to ArcLink data request (convenient for archived data). Figure 3.3 is a summary of main parameters.

Configuration files

To configure a new SEFRAN3, copy the template files **SEFRAN3.conf** and **SEFRAN3_Channels.conf** to, for example, **MYSEFRAN.conf** and **MYSEFRAN_Channels.conf**.

MYSEFRAN_Channels.conf file sets the channel list. Each channel is defined by its alias (a short code used for display), the stream string (used for the data request), the sensitivity factor (to convert counts to m/s), the filter (median, trend or spline removal), the peak-to-peak signal amplitude, and the color. A good conduct is to order the channels from North to South, and give different colors for specific regions. There is no limit to the number of channels, but a maximum of 15 channels is recommended for a normal screen resolution.

MYSEFRAN.conf file is the main configuration file. It defines a lot of things like output paths, server address and all parameters of graphical outputs and SEFRAN3 behavior.

SEFRAN3_Channels.conf

```
# SEFRAN3 template channnel file
#
# Definition of the SEFRAN3 channels:
#       - Alias: channel alias code (for display)
#       - Stream: channel stream full name (Network.Station.LocId.Channel)
#       - Sensitivity: sensitivity factor (in counts per m/s)
#       - Filter: applyes a filter to the signal:
#                       [X]: constant offset value [X] (in count)
#                    median: median value correction (formerly 'auto')
#                     trend: linear detrend correction
#                     sp[X]: spline filter using [X] seconds interval points
#                            ([X] is a positive integer)
#       [ft][fn][N],[F],[S]: digital filters
#                           ft = 'lp' (lowpass), 'hp' (highpass),
#                                'bp' (bandpass) or 'bs' (bandstop);
#                           fn = 'bu' (Butterworth), 'be' (Bessel),
#                                'c1' or 'c2' (Chebyshev type I or II);
#                            N = positive integer;
#                            F = frequency (in Hz), use [FL,FH] for 'bp'
#                                and 'bs' type;
#                            S = stopband attenuation/ripple (in dB) for
#                                Chebyshev only.
#       - PP: define the signal amplitude (peak-to-peak, in m/s)
#       - RGB: define the signal color (HTML format: hexa or standard name)
#         ATTENTION: for hexa format use double-quote "#FF0000" to avoid comment !
#
#
# Alias Stream          Sensitivity     Filter          PP      RGB
COYC    G.COYC.00.LHZ   5.944420E+09    hpbu3,0.2       5e-5    "#008800"
FDF     G.FDF.00.LHZ    2.516580E+09    median          5e-5    "#008800"
SSB     G.SSB.00.LHZ    5.977820E+09    median          5e-5    "#FF0000"
MBO     G.MBO.00.LHZ    4.998380E+09    median          5e-5    "#FF0000"
NOUC    G.NOUC.00.LHZ   5.931530E+09    median          5e-5    "#FF0000"
#NOUC   G.NOUC.00.LHZ   5.931530E+09    hpbu3,0.2       5e-5    "#FF0000"
PPTF    G.PPTF.00.LHZ   5.680760E+09    median          5e-5    "#FFA500"
#PPTF   G.PPTF.00.LHZ   5.680760E+09    hpbu3,0.3       5e-5    "#FFA500"
RER     G.RER.00.LHZ    5.643860E+09    median          5e-5    "#FFA500"
TAM     G.TAM.00.LHZ    3.885600E+09    median          5e-5    "#000088"
UNM     G.UNM.00.LHZ    5.615490E+09    median          5e-5    "#000088"
```

SEFRAN3.conf

```
=key|value
############################################################
# SEFRAN3 configuration file
############################################################
TITRE|SEFRAN3: GEOSCOPE World Seismicity
BANG|2014
ROOT|$WEBOBS{SEFRAN_ROOT}/sefran3
CHANNEL_CONF|$WEBOBS{ROOT_CONF}/SEFRAN3_Channels.conf
CSS|SEFRAN3.css

BIGARROWS|YES

PATH_WEB|$WEBOBS{SEFRAN_ROOT_ALIAS}/sefran3

# sub-directories: not modifiable after a first run!
PATH_IMAGES_MINUTE|minute
PATH_IMAGES_HOUR|hour
PATH_IMAGES_HEADER|header

# --- Data source and format
# combined format : DATASOURCE|protocole1;protocol2;delay
DATASOURCE|slink://${SEEDLINK_SERVER};arclink://${ARCLINK_SERVER}?user=${ARCLINK_USER};${ARCLINK_DELAY_HOURS}
# notes:
#         protocole1 is real-time data
#         protocole2 is archived data
#         delay is time limit (in hour) to switch from protocole1 to protocole2
# protocoles can be slink, arclink, fdsnws or file (miniseed)
# examples:
#DATASOURCE|slink://rtserver.ipgp.fr:18000;arclink://eida.ipgp.fr:18001?user=sefran3;12
#DATASOURCE|slink://rtserver.ipgp.fr:18000;fdsnws://http://rtpriv-cdd.ipgp.fr/fdsnws/dataselect/1/query?;2
# latency delay for realtime data
LATENCY_SECONDS|${SEEDLINK_DELAY_SECONDS}

# seedLink server: source of real-time data (needed to display the status table)
SEEDLINK_SERVER|rtserver.ipgp.fr:18000
```

```
SEEDLINK_SERVER_TIMEOUT_SECONDS|30

# for backward compatibility only (to be used for default DATASOURCE if needed)!
SEEDLINK_DELAY_SECONDS|300
ARCLINK_SERVER|eida.ipgp.fr:18001
ARCLINK_USER|sefran3
ARCLINK_DELAY_HOURS|12

# cleans overlaps in data streams
CLEAN_OVERLAPS|1

# --- Run-time parameters
# moving time window which is processed
UPDATE_HOURS|6
# minimum beat of the main loop (in seconds)
BEAT|2
# minimum loop runtime before exit
MIN_RUNTIME_SECONDS|600
# maximum number of processed images in a single run
MAX_IMAGES_IN_RUN|10

# --- Broomwagon: reprocess images if needed, after a while
BROOMWAGON_ACTIVE|0
# minimum age(s) of an image to be considered (in hour)
# multiple broomwagon allowed in a coma-separated list
BROOMWAGON_DELAY_HOURS|2,12
# time window which will be processed
BROOMWAGON_UPDATE_HOURS|1
# threshold parameters
BROOMWAGON_MAX_DEAD_CHANNELS|1
BROOMWAGON_MAX_GAP_FACTOR|0.2

# extra seconds of signal needed for filters
FILTER_EXTRA_SECONDS|0

# --- Associated "Main Courante" (seismic bulletin)
MC3_NAME|MC3
MC3_EVENT_OPACITY|0.4
MC3_EVENT_DISPLAY_LOC|N
DISPLAY_LAST_MC|2
DISPLAY_LAST_MC_HOURS|1

# main page refrech (in real-time mode)
DISPLAY_REFRESH_SECONDS|60
REF_NORTC|1
# max number of days (in a page)
DISPLAY_DAYS|7

# --- Main graphical parameters
# pixel-per-inch for all PNG images output
VALUE_PPI|100
# image height (in inches): 7.8 means 780 pixels at 100 ppi
HEIGHT_INCH|7.8
# equivalent paper speed in inches per minute
VALUE_SPEED|1.2
# paper high-speed (in/mn)
VALUE_SPEED_HIGH|4.8
# intertrace: 1 means no overlap, 0.8 means 20% overlap
INTERTRACE|0.8
TRACE_LINEWIDTH|1
# to obtain better PNG, images are oversampled
PRINT_OVERSAMPLING_FACTOR|2
# labels for minute-image (in pixels)
LABEL_TOP_HEIGHT|23
LABEL_BOTTOM_HEIGHT|55
XTICK_INTERVAL_SECONDS|1
XTICK_LABEL_INTERVAL_SECONDS|10

# --- Hourly thumbnails (sizes in pixels)
HOURLY_WIDTH|900
HOURLY_HEIGHT|90
HOURLY_CONVERT_GAMMA|0.4
LASTHOUR_WIDTH|320

NOTES|$WEBOBS{PATH_USERS_HTML}/Sefran3_Notes.htm
MENU_FILE|$WEBOBS{PATH_USERS_HTML}/Sefran3_Menu.htm

TIME_INTERVALS_CONF|$WEBOBS{ROOT_CONF}/SEFRAN3_TimeIntervals.conf
TIME_INTERVALS_DEFAULT_VALUE|6

# channel status threshold
STATUS_OFFSET_WARNING|2
STATUS_OFFSET_CRITICAL|5
STATUS_NOISE_WARNING|1e-6
STATUS_NOISE_CRITICAL|5e-6
STATUS_DELAY_WARNING|300
STATUS_DELAY_CRITICAL|86400
```

Activation

To activate the SEFRAN3, add a new job in the scheduler, for example with following parameters:

jid: mysefran
res: mysefran
xeq1: $WEBOBS{JOB_MCC} sefran3
xeq2: MYSEFRAN
interval: 600
logpath: mysefran
valid: Y

After activation, check that it runs correctly in the Scheduler Runs page (see Section 4.3).

To access the main SEFRAN3 page, use the following URL (can be set for instance in the menu bar):

/cgi-bin/sefran3.pl?header=1&s3=MYSEFRAN

Further options are available to access all display possibilities. See **perldoc CODE/cgi-bin/sefran3.pl**.

3.3.3 MC3 configuration

MC3.conf

```
=key|value
################################################################################
# MC3 Main Courante configuration parameters
################################################################################
TITLE|Main Courante Seismicity
BANG|2014
ROOT|$WEBOBS{SEFRAN_ROOT}/mc3
PATH_WEB|$WEBOBS{SEFRAN_ROOT_ALIAS}/mc3
EVENT_CODES_CONF|$WEBOBS{ROOT_CONF}/MC3_Codes.conf
DURATIONS_CONF|$WEBOBS{ROOT_CONF}/MC3_Durations.conf
AMPLITUDES_CONF|$WEBOBS{ROOT_CONF}/MC3_Amplitudes.conf
TABLE_HEADERS_CONF|$WEBOBS{ROOT_CONF}/MC3_Headers.conf

PATH_SC3_QML|
PATH_TMP|$WEBOBS{SEFRAN_ROOT}/tmp
PATH_TMP_WEB|$WEBOBS{SEFRAN_ROOT_ALIAS}/tmp
PATH_FILES|files
PATH_IMAGES|images
FILE_PREFIX|MC_
FRAME_WIDTH|600
WINDOW_LENGTH_MINUTE|5
IMAGES_MAX_CAT|60
# default table interval display (in days)
DEFAULT_TABLE_DAYS|30
# local time zone: used only for date/hour selection (display and statistics) - data always remain in UTC !
SELECT_LOCAL_TZ|0
DEFAULT_SELECT_LOCAL|N

NOTES|$WEBOBS{PATH_USERS_HTML}/MC3_Notes.htm
FILTER_POPUP|$WEBOBS{PATH_USERS_HTML}/MC3_Filter_PopUp.htm

# external catalog visit link (formely USGS_URL)
VISIT_LINK|<A href="http://www.emsc-csem.org/Earthquake/" target="_blank"><B>EMSC</B></A>
#VISIT_LINK|<A href="https://earthquake.usgs.gov/earthquakes/map" target="_blank"><B>USGS</B></A>
#USGS_URL|https://earthquake.usgs.gov/earthquakes/map

AUTOPRINT|0
PRINTER|
CSS|MC3.css
P_WAVE_VELOCITY|6
VP_VS_RATIO|1.75

# set to 0 to keep the S-P value when changing P-phase
NEW_P_CLEAR_S|1

# default amplitude value for new event: see MC3_Amplitude.conf for valid keys
DEFAULT_AMPLITUDE|AVERAGE
# default type value for new event: see MC3_Codes.conf for valid keys
DEFAULT_TYPE|UNKNOWN

# small ruler parameters (obsolete)
RULER_DISPLAY|0
RULER_LENGTH|60
RULER_TICKS|5
RULER_TICKS_LABEL|10
RULER_POS_DX|0
RULER_POS_DY|-50

# shakemaps
SHAKEMAPS|0
SHAKEMAPS_PATH|$WEBOBS{SEFRAN_ROOT}/B3
SHAKEMAPS_URN|$WEBOBS{URN_SEFRAN}/B3
```

```
TREMBLEMAPS_PROC|B3

# Cities for event location
# ----------------------------------------
CITIES|$WEBOBS{ROOT_CONF}/Cities.conf
CITIES_PLACE|Guadeloupe

#for MC2 compatibilty
PATH_DESTINATION_SIGNAUX|$WEBOBS{SEFRAN_ROOT}/A_Depouiller
WEB_DESTINATION_SIGNAUX|$WEBOBS{SEFRAN_ROOT_ALIAS}/A_Depouiller
HYPO_USE_FMTO|
HYPO_USE_FMT1|$WEBOBS{OVPFHYP_PATH}
DISPLAY_LOCATION_DEFAULT|1
LOCATION_MIN_PHASES|4

# ===== SeisComP3 database parameters =====
SC3_EVENTS_ROOT|$WEBOBS{SEFRAN_ROOT}/sc3_events
# 4-letter prefix of events ID
SC3_EVENTS_ID_PREFIX|abcd
# time window to scan new events (in days)
SC3_UPDATE_DAYS|15
SC3_USER|SC3
# comma separated list of unwanted event types (not imported in MC3)
SC3_EVENT_TYPES_BLACKLIST|not existing
# comma separated list of unwanted event types for displaying location in MC3
#   Note to SC3 FDSNWS users: the QuakeML standard used by FDSNWS does not
#   define any "not locatable" event type. This event type is therefore mapped
#   to "other event" in FDSNWS, and should be included in the definition
#   of SC3_EVENT_TYPES_NOLOCATION below to hide their location in the MC3.
#   Be aware that this will also hide events of type "other event" as well as
#   "duplicate" events (that are also mapped to "other event" for the same
#   reason).
SC3_EVENT_TYPES_NOLOCATION|outside of network interest,not locatable
# wo2sc3 module parameters
WO2SC3_HOSTNAME|localhost
WO2SC3_PORT|30003
WO2SC3_MOD_ID|1
WO2SC3_MOD_TYPE|1

# ===== FDSN WebService event parameters =====
# Uses SC3_EVENT_TYPES_BLACKLIST, SC3_EVENT_TYPES_NOLOCATION and SC3_UPDATE_DAYS
# Default server (when only one)
FDSNWS_EVENTS_URL|https://earthquake.usgs.gov/fdsnws/event/1/query?
# WebService options for catalog search : area of interest, magnitude of interest
FDSNWS_EVENTS_SEARCH|minmagnitude=6
# WebService options for event details retrieval : include arrivals
FDSNWS_EVENTS_DETAIL|includeallmagnitudes=false&includeallorigins=false

# Extra EMSC FDSN events server (may have more than one)
FDSNWS_EVENTS_URL_EMSC|https://earthquake.usgs.gov/fdsnws/event/1/query?
# WebService options for catalog search : area of interest, magnitude of interest
FDSNWS_EVENTS_SEARCH_EMSC|minmagnitude=5&maxmagnitude=6&minlatitude=11&maxlatitude=19&minlongitude=-64&maxlongitude=-59
# WebService options for event details retrieval : include arrivals
FDSNWS_EVENTS_DETAIL_EMSC|includeallmagnitudes=false&includeallorigins=false&includearrivals=true
```

MC3_Codes.conf

```
# WEBOBS
# Configuration file for seismic bulletin "Main Courante"
#
# Input to mc3.pl and sefran3.pl, to define events.
#
#       key = Event-type code as recorded in files
#               [Note: UNKNOWN and AUTO are RESERVED keywords ]
#
#       Name = Full text of event-type as presented to user
#
#       Color = Associated color in display/graphs ... html hexa format #RRGGBB
#
#       Md = how to compute duration magnitude :
#               1 = requires duration and S-P
#               0 = requires duration only (assumes S-P = 0 if not defined)
#               -1 = never compute
#
#       asVT = whether event is counted as VT in seismic bulletin
#               1 = yes
#               0 = no
#
#       asRF = whether event is counted as rockfall in seismic bulletin
#               1 = yes
#               0 = no
#
#       Location = whether location info will be displayed
#               1 = yes (except if SC3 type in the SC3_EVENT_TYPES_NOLOCATION list)
#               0 = no
#
#       WO2SC3 = whether SeisComP3 eventID creation is checked
#               1 = yes
#               0 = no
#
```

```
=key|Name|Color|BgColor|Md|asVT|asRF|Location|WO2SC3
#
UNKNOWN|Unknown event|\#535353|\#FFFFFF|-1|0|0|1|1
VOLCTECT|Volcano-Tectonic|\#FA8072|\#FFFFFF|0|1|0|1|1
VOLCLP|Volcanic Long-Periode|\#DC143C|\#FFFFFF|0|0|0|1|1
VOLCTREMOR|Volcanic Tremor|\#FF4500|\#FFFFFF|-1|0|0|1|1
ROCKFALL|Rockfall/Landslide|\#8B4513|\#FFFFFF|-1|0|1|0|1
TECT|Tectonic|\#228B22|\#FFFFFF|1|0|1|1|1
TELE|Teleseism|\#ADFF2F|\#FFFFFF|1|0|0|0|0
TPHASE|T-Phase|\#4682B4|\#FFFFFF|-1|0|0|0|0
SOUND|Sound|\#7FFFD4|\#FFFFFF|-1|0|0|0|0
ANTHROP|Anthropogenic|\#FFD700|\#FFFFFF|1|0|0|1|0
AUTO|AUTOMATIC|\#808080|\#F6CECE|-1|0|0|1|0
COMMENT|Comment|\#FFFF22|\#666666|-1|0|0|0|0
```

MC3_Amplitudes.conf

```
# WEBOBS
# Configuration file for MC3 amplitude signals definition
#
#       - Key = key for database (do not modify or erase)
#       - Name = name for display
#       - Value = numerical value (in signal units, i.e., m/s or counts), will be
#                 used for amplitude event selection (with operators <= - >=)
#
# Key|Name|Value
WEAK|Weak|0
AVERAGE|Average|500
STRONG|Strong|1000
OVERSCALE|Overscale|2000
```

MC3_Durations.conf

```
# WEBOBS
# Configuration file for MC3 signal durations definition
#
#       - Key = key for database (do not modify or erase)
#       - Name = name for display
#       - Sec = number of seconds
#
# Key|Name|Sec
s|seconds|1
mn|minutes|60
h|hours|3600
d|days|86400
```

3.3.4 Links with earthquake event catalogs

SeisComP3

EarthWorm

3.4 GENPLOT: generic time series

3.4.1 Overview

GENPLOT is the default superproc to plot time series from any source of data, particularly the time series formats (see section 3.14.2). GENPLOT is able to produce graphics and text outputs from any data channels of associated NODES for a list of preset time scales and two types of graphs:

- one graph per NODE showing separated subplot for each selected channel;
- one summary graph combining all NODES on each channel subplot using different colors. This graph is activated by setting `SUMMARYLIST` parameter to the keyword `SUMMARY` (this is the default). To remove the summary graph just empty or comment the parameter.

GENPLOT and all other superprocs have the same configuration parameters of a standard VIEW, mainly for the PROC's name, NODES options and DEM parameters (see section 2.3.2). Additional parameters are specific to each superproc, but most of the GENPLOT parameters below will apply also to other superprocs.

Data source, format and timezone

The generic behavior of superprocs is to set a global data source and format for all associated NODES using:

- `RAWFORMAT` keyword (see table 5.1);
- `RAWDATA` full path or web-service url (depending on each format).

This global PROC data source can be overwritten by any individual NODE's data source (see section 2.2.7), so a PROC is able to deal with different data formats for each of its associated NODES.

The timezone of the raw data must be defined in each of the NODE's configuration (see section 2.2.7) otherwise it will be considered as GMT by default (TZ=0). The PROC will then display and export the data using the GENPLOT parameter applying to all NODES and outputs:

- `TZ` in hours (default is 0 for GMT).

Preset time scales (real-time moving windows)

Time scales are defined by the `TIMESCALELIST` parameter that contains coma-separated codes as defined by table 5.2, which can be one of the following:

- a fixed duration expressed in hour, day, week, month or year as a moving window ending at the present time;
- a variable duration from a fixed reference date until the present time (extending window);
- all available data, from the first to the last (adapted window for each NODE).

These preset time scales will produce graphs with moving time limits that will be automatically updated by the scheduler (see section 4.3). To make specific graphs from selected dates and times limits, use the manual PROCS graph and data request form (see section 3.13).

Channel selection

Each associated NODE must have a calibration file that defines all channel parameters (see section 2.2.7). By default, all available channels are plotted in the per-node and summary graphs, but selection can be made using respectively `PERNODE_CHANNELS` and `SUMMARY_CHANNELS` parameters and coma-separated channel number list, as defined in the calibration file (**Ch. Nb**). Setting optional multiple comas to extend the previous subplot height in proportion to others. Example:

```
PERNODE_CHANNELS|1,2,3,4
SUMMARY_CHANNELS|1,,4,2
```

sets the channels 1 to 4 (from top to bottom) with equal subplot size for each per-node plot, and the channels 1, 4 and 2 for the summary plot, but the first subplot (channel 1) will be double-height compared to the second (channel 4) and third subplots (channel 2).

Data filtering

Some filters can be applied to the whole set of data:

- FLAT_IS_NAN set to YES replaces flat signal, i.e. two consecutive strictly identical values, by Not-a-Number (NaN which disappear from graph); this is useful to detect some sensor/digitizer anomaly;
- PICKS_CLEAN_PERCENT removes data picks using a threshold as % of min/max of the data values after linear detrend;
- PICKS_CLEAN_STD removes data picks using a threshold as a multiple of standard deviation of data around the mean value after linear detrend;
- MEDIAN_FILTER_SAMPLES applies a moving median filter on a number of data samples. A value of 3 makes a robust filter to remove individual spikes, without affecting too much the data;
- DECIMATELIST defines a decimation factor for each timescale (coma-separated values, default is 1); the decimation will use average of data samples (no other filtering);
- MOVING_FILTER_SAMPLES keeps the original data but add an additional plot with a moving average, using the same color but lighter.

Graphical options

Titles, labels, line and marker style and other page settings can also be configured:

- set PERNODE_TITLE and/or SUMMARY_TITLE to customize the title; string can contain some TeX commands, any of the PROC variables (e.g. ${NAME}) and a list of internal variables that will be substituted in the text strings:
  - $node_name = node long name (for pernode plots),
  - $node_alias = node alias (for pernode plots),
  - $timescale = time scale string (e.g. '1 year'),
  - $last_data = last data date and time string.
- set PERNODE_LINESTYLE and/or SUMMARY_LINESTYLE to a line specification string, a combination of two components:
  - a line type (see possible values in table 5.4),
  - a marker symbol type (see possible values in table 5.3),

  to choose between drawing markers and/or line and define their style; note you might specify a marker only with no line, then only the markers are plotted;
- the line width and marker size can be different for each time scale using respectively LINEWIDTHLIST and MARKERSIZELIST, that should both define a size to use (in pt) for each of the timescales defined in TIMESCALELIST, coma separated values;
- colors are chosen automatically: for the summary graph it will be one color per node, for the per-node graph it will be one color per channel;
- CONTINUOUS_PLOT set to YES will join all data points by a line (if linestyle contains a line component), while the default is to show gaps where two data points are separated more than 1.5 times the acquisition period (if this parameter is set in the NODE's configuration);
- PLOT_GRID adds grid lines to all axis;
- YLOGSCALE set to YES will make all Y-axis logarithmic;
- EVENTS_FILE set a filename (or a coma-separated list of filenames) containing a list of time-referenced events that will be plotted in the background of each graph (see example below).

Events file example

```
# Configuration file for time referenced events associated with a proc
# using EVENTS_FILE that will be plotted in the background of graphs,
# as shaded color areas with a text comment.
#
#       Date1: start date of the event (YYYY-MM-DD hh:mm format, time is optional)
#       Date2: end date of the event (YYYY-MM-DD hh:mm format) or same as Date1 for "instant" event.
#   LineWidth: line width (in pt) for "instant" event.
#         RGB: color (HTML format, e.g., 'Red' or \#FFFFFF)
#        Name: pop-up window's title
#     Comment: pop-up window's long text
#
# ------------------------------------------------------------
# Date1|Date2|LineWidth|RGB|Name|Comment
#
2004-11-21 11:40|2004-11-21 12:00|.5|LightGreen|Les Saintes Mw6.3|Earthquake in Guadeloupe, FWI
2007-11-29 19:00|2007-11-29 20:00|.5|LightGreen|Martinique Mw7.4|Earthquake in Martinique, FWI
2010-10-26 10:02|2010-11-07 00:00|.5|IndianRed|Merapi 2010|Volcano eruption in Indonesia
2014-08-29 00:00|2015-02-27 00:00|.5|LightPink|Bardarbunga|Volcano eruption in Iceland
```

- PAGESIZE sets the page width and height in inches (default is 8.5,11 for usletter) for PostScript image output;

- PPI sets the point-per-inch resolution to convert the EPS in PNG raster images (default is 100 so standard image outputs are 800×1100 pixels);
- PAGES_MAX_SUBPLOT will extend the page height if the number of subplots exceeds this value (default is 8), to avoid shrinky graphs;
- a logo can be added to the page at upper-left corner using LOGO_FILE (full path of image, default is a **WebObs** logo defined by LOGO_DEFAULT in **WEBOBS.rc**) and LOGO_HEIGHT (a fraction of page's height, default is 0.04, i.e., 4%). The aspect ratio of image will be preserved. It is also possible to set a second logo that will be plotted at upper-right corner using LOGO2_FILE and LOGO2_HEIGHT; LOGO_FILE and LOGO2_FILE might contain a list of coma separated filenames so images will be horizontally appended from left to right, with same height;
- two copyrights strings can be set using COPYRIGHT and COPYRIGHT2 (some TeX allowed). They will appear below the graph title;
- DISPLAY_AUTOREFRESH_SECONDS sets the time delay for web browser autorefresh of the PROC's output graph pages. Note that a dedicated icon is available in all graphs submenu to refresh manually if necessary.

3.4.2 Configuration

Some parameter keys of GENPLOT are common for all PROCS and GRIDS so are identical with the VIEW configuration (see section 2.3.2).

GENPLOT template

```
=key|value
# Generic PROC template
# It can be used for instance with superproc genplot (generic time series).
# See $WEBOBS{ROOT_CODE}/matlab/superprocs/genplot.m for further information.

# long name of the PROC
NAME|Generic time series
# proc TYPE and OWNER
TYPE|
OWNCODE|?
# default raw data format (might be overwritten in each node)
RAWFORMAT|
# raw data path (might be overwritten in each node configuration)
RAWDATA|$WEBOBS{ROOT_RAWD}/mydatasource
# URN link to data
URNDATA|$WEBOBS{URN_RAWD}/mydatasource

DEM_SRTM1|N
# optional user-defined DEM (Arcinfo format, lat/lon, overwrites SRTM/ETOPO default)
#DEM_FILE|$WEBOBS{PATH_DATA_DEM}/highresdem_latlon.asc
#DEM_TYPE|LATLON
#DEM_COPYRIGHT|DEM: myDEM

# time zone for outputs
TZ|0

# additionnal external link(s): TEXT1,URL1;TEXT2,URL2;...
URL|

# lists of parameters (number of comma-separated rows must be consistent)
# TIMESCALELIST valid suffix: h (hour), d (day), w (week), m (month), y (year)
# TIMESCALELIST reserved words are: all (all available data), ref (reference date until now)
TIMESCALELIST|24h,01y
DECIMATELIST|1,1
CUMULATELIST|1,1
DATESTRLIST|-1,-1
MARKERSIZELIST|6,2
LINEWIDTHLIST|2,1
STATUSLIST|1,0
# defines a reference date to use with 'r01' timescale in TIMESCALELIST
REF01_DATE|2000-01-01
# year and month timescales use true durations
TIMESCALE_TRUEVALUE|Y

# execution command (for runproc shell)
SUBMIT_COMMAND|$WEBOBS{JOB_MCC} genplot $SELFREF -
SUBMIT_RESOURCE|myproc

# list of keys editable in requests
REQUEST_KEYLIST|NAME

# auto refresh of graphic display
DISPLAY_AUTOREFRESH_SECONDS|3600

# --- node parameters
NODE_NAME|node
NODE_SIZE|10
NODE_RGB|Red
NODE_FONTSIZE|10
NODE_MARKER|s
```

```
# set to YES to use SRTM1 30m resolution tiles
DEM_SRTM1|NO

# optional user-defined DEM (Arcinfo format, lat/lon) overwriting SRTM/ETOPO default
#DEM_FILE|$WEBOBS{PATH_DATA_DEM}/highresdem_latlon.asc
#DEM_TYPE|LATLON
#DEM_COPYRIGHT|DEM: myDEM

# additional zoomed map(s) using fixed axis limits: LON1,LON2,LAT1,LAT2
#MAP1_XYLIM|
#MAP2_XYLIM|
# do not show node alias in zoom map(s)
NODE_SUBMAP_ALIAS|N

# file for background time referenced events
EVENTS_FILE|$WEBOBS{ROOT_CONF}/events_World.conf

# --- page format and outputs
# optional paper size width,height in inches
PAPER_SIZE|
# graphical density for PNG outputs (in pixel per inches)
PPI|100
# landscape format for all outputs
LANDSCAPE|NO
# maximum number of subplots in a standard page height
PAGE_MAX_SUBPLOT|8
# plot grid for all axes
PLOT_GRID|YES
# makes a PDF file for each graph (in addition to PNG, EPS and JPG thumbnail)
PDFOUTPUT|NO
# main logo file
LOGO_FILE|$WEBOBS{LOGO_DEFAULT}
LOGO_HEIGHT|$WEBOBS{LOGO_HEIGHT}
# main copyright
COPYRIGHT|MyProc
# optional secondary logo file
LOGO2_FILE|
LOGO2_HEIGHT|$WEBOBS{LOGO_HEIGHT}
# optional secondary copyright
COPYRIGHT2|
# exports text data files
EXPORTS|YES

# --- data processing and plot
# removes data picks using STD filter (around mean)
PICKS_CLEAN_STD|0
# removes data picks using median filter (% of min/max values)
PICKS_CLEAN_PERCENT|0
# replaces flat signal (two consecutive identical values) by NaN
FLAT_IS_NAN|NO
# applies a median filter to the data
MEDIAN_FILTER_SAMPLES|0
# superimpose a moving average on N samples
MOVING_AVERAGE_SAMPLES|0
# set to YES to plot continuous line ignoring gaps
CONTINUOUS_PLOT|NO
# set Y-axis with a log scale
YLOGSCALE|NO

# --- per node graphs
# coma separated list of channels to plot (default is all channels)
PERNODE_CHANNELS|${NODE_CHANNELS}
# graph title
PERNODE_TITLE|{\fontsize{14}{\bf$node_alias: $node_name} ($timescale)}
# line style: combination of a marker and/or a line type (see user's manual)
PERNODE_LINESTYLE|-
# set to YES to substract the median value in all plots
PERNODE_RELATIVE|NO

# --- summary graph (all nodes together)
# empty or comment this key to not plot the summary graph
SUMMARYLIST|SUMMARY
# coma separated list of channels to plot (default is all channels)
SUMMARY_CHANNELS|
# graph title
SUMMARY_TITLE|{\fontsize{14}{\bf${NAME}} ($timescale)}
# line style: combination of a marker and/or a line type (see user's manual)
SUMMARY_LINESTYLE|-
# set to YES to substract the median data value in all plots
SUMMARY_RELATIVE|NO
```

3.5 HYPOMAP: Earthquake hypocenter maps from seismic catalog

3.5.1 Overview

3.5.2 Configuration

Some parameter keys of HYPOMAP are common for all PROCS and GRIDS so are identical with the VIEW configuration (see section 2.3.2).

HYPOMAP template

```
=key|value
# Generic PROC template for superproc hypomap
# Superformat is 'quakes': see readfmtdata_quakes.m for compatible RAWFORMAT defined
# in associated NODE's RAWFORMAT field
# See $WEBOBS{ROOT_CODE}/matlab/superprocs/hypomap.m for further information.

# long name of the PROC
NAME|Earthquake hypocenter maps from seismic catalog
# proc TYPE and OWNER code (see OWNERS.conf for OWNCODEs)
RAWFORMAT|quakes
TYPE|
OWNCODE|?
# raw data path
RAWDATA|
# URN link to data
URNDATA|
# time zone for outputs
TZ|0
# additionnal URL
URL|

# lists of parameters (number of comma-separated rows must be consistent)
# TIMESCALELIST valid suffix: h (hour), d (day), w (week), m (month), y (year)
TIMESCALELIST|01y,10y,all
DATESTRLIST|-1,-1,-1
MARKERSIZELIST|5,3,2
# defines a reference date to use with 'r01' timescale in TIMESCALELIST
REF01_DATE|2000-01-01
# year and month timescales use true durations
TIMESCALE_TRUEVALUE|Y

# additional summary graphs keywords (comma separated)
SUMMARYLIST|Area1
SUMMARY_CHANNELS|

# execution command (for runproc shell)
SUBMIT_COMMAND|$WEBOBS{JOB_MCC} hypomap $SELFREF -
SUBMIT_RESOURCE|hypomap

# list of keys editable in requests
REQUEST_KEYLIST|NAME,SUMMARYLIST,LATLIM,LONLIM,MAGLIM,DEPLIM,PLOT_BG_ALL,BUBBLE_PLOT

# node parameters
NODE_NAME|node
NODE_SIZE|10
NODE_RGB|1,0,0
NODE_FONTSIZE|10
NODE_MARKER|o

# auto refresh of graphic display
DISPLAY_AUTOREFRESH_SECONDS|3600

# --- optional user-defined DEM, in ArcInfo format (overwrites SRTM/ETOPO default)
DEM_FILE|$WEBOBS{PATH_DATA_DEM}/exampledem.asc
# DEM coordinate system: UTM or LATLON
DEM_TYPE|LATLON
DEM_COPYRIGHT|

# additional zoomed map(s) using fixed axis limits: LON1,LON2,LAT1,LAT2
#MAP1_XYLIM|
#MAP2_XYLIM|
# do not show node alias in zoom map(s)
NODE_SUBMAP_ALIAS|N

# graphical density for PNG outputs (in pixel per inches)
PPI|100
# value = 1 to produce EPS graphs
POSTSCRIPT|1
# value = 1 to export text data files
EXPORTS|1
# main logo file
LOGO_FILE|$WEBOBS{ROOT_CODE}/icons/ipgp/logo_WOVS.png
# secondary logo file
LOGO2_FILE|
# main copyright
COPYRIGHT|IPGP
# secondary copyright
COPYRIGHT2|
```

```
# --- proc specific parameters
# for catalog data request
EVENTTYPE_EXCLUDED_LIST|not existing,not locatable,outside of network interest,sonic boom,duplicate,other event
EVENTSTATUS_EXCLUDED_LIST|automatic
EVENTCOMMENT_EXCLUDED_REGEXP|AUTOMATIC
SC3_LISTEVT|
LATLIM|13,19
LONLIM|-64,-58
MAGLIM|3,10
DEPLIM|-10,300
MSKLIM|1,12
GAPLIM|0,360
RMSLIM|0,1
ERHLIM|0,100
ERZLIM|0,100
NPHLIM|3,Inf
CLALIM|0,4
QUALITY_FILTER|0.7
# for graph
MARKER_LINEWIDTH|1
PLOT_BG_ALL|.3
DEM_OPT|'WaterMark',2,'FontSize',7
SHAPE_FILE|$WEBOBS{PATH_DATA_SHAPE}/antilles_faults.bln
#STATIONS_GRID|VIEW.
# Hypocenters without individual circle line (1 yes, 0 no)
BUBBLE_PLOT|1

# -- Maps of areas: duplicate MAP_areaname_* keys to make specific maps
MAP_Area1_TITLE|Area1 - Lesser Antilles
# map limits: lon0,lat0,width (in degree) or former lon1,lon2,lat1,lat2 (in degree)
MAP_Area1_XYLIM|-61,16,6
# magnitude limits (for size scale only, not a filter)
MAP_Area1_MAGLIM|3,7
# depth limits (for color scale only, not a filter)
MAP_Area1_DEPLIM|-2,200
# 1 or 2 optional profile(s)
# Profile 1 (bottom): center latitude, center longitude, azimuth (degree North), lateral distance (km), depth (km)
MAP_Area1_PROFILE1|-61.4651,16.5138,55,100,200
# Profile 2 (right): same parameters as profile1, but drawn on the right side of the map
# Colormap used: must specify the number of colors
MAP_Area1_COLORMAP|jet(256)
# Color reference for markers: 'depth' (default) or 'time'
MAP_Area1_COLORREF|time
# Optional time plot: give a list of parameters (versus time) ==> alpha version not fully functional
#MAP_Area1_TIMEPLOT|latitude,longitude,depth,magnitude,profile1,profile2
```

3.6 HELICORDER: Seismic helicorder

3.6.1 Overview

3.6.2 Configuration

Some parameter keys of HELICORDER are common for all PROCS and GRIDS so are identical with the VIEW configuration (see section 2.3.2).

HELICORDER template

```
=key|value
# Helicorder PROC template
# Makes helicorders from node channels, using "events" output type.
# See $WEBOBS{ROOT_CODE}/matlab/superprocs/helicorder.m for further information.

# long name of the PROC
NAME|Helicorder
# proc TYPE and OWNER code (see OWNERS.conf for OWNCODEs)
TYPE|
OWNCODE|?
RAWFORMAT|
# raw data path
RAWDATA|
# URN link to data
URNDATA|/rawdata/myproc

# time zone for outputs
TZ|0
# additionnal URL
URL|

# lists of parameters (number of comma-separated rows must be consistent)
# TIMESCALELIST valid suffix: h (hour), d (day), w (week), m (month), y (year)
TIMESCALELIST|02d
DECIMATELIST|1
CUMULATELIST|1
DATESTRLIST|-1
LINEWIDTHLIST|1
```

```
MARKERSIZELIST|4
STATUSLIST|1
# defines a reference date to use with 'r01' timescale in TIMESCALELIST
REF01_DATE|2000-01-01
# year and month timescales use true durations
TIMESCALE_TRUEVALUE|Y

# execution command (for runproc shell)
SUBMIT_COMMAND|$WEBOBS{JOB_MCC} helicorder $SELFREF -
SUBMIT_RESOURCE|helicorder

# list of keys editable in requests
REQUEST_KEYLIST|NAME,HELICORDER_DURATION_DAYS,HELICORDER_TURNS,HELICORDER_SCALE,HELICORDER_PAPER_COLOR,HELICORDER_PAPER_C

# node parameters
NODE_NAME|node
NODE_SIZE|10
NODE_RGB|1,0,0
NODE_FONTSIZE|10
NODE_MARKER|o

# auto refresh of graphic display
DISPLAY_AUTOREFRESH_SECONDS|3600

# --- optional user-defined DEM, in ArcInfo format (overwrites SRTM/ETOPO default)
DEM_FILE|$WEBOBS{PATH_DATA_DEM}/exampledem.asc
# DEM coordinate system: UTM or LATLON
DEM_TYPE|LATLON
DEM_COPYRIGHT|
DEM_SRTM1|N
# additional zoomed map(s) using fixed axis limits: LON1,LON2,LAT1,LAT2
#MAP1_XYLIM|
#MAP2_XYLIM|
# do not show node alias in zoom map(s)
NODE_SUBMAP_ALIAS|N

# file for background time referenced events
EVENTS_FILE|$WEBOBS{ROOT_CONF}/events_World.conf

# graphical density for PNG outputs (in pixel per inches)
PPI|100
# landscape format for all outputs
LANDSCAPE|Y
# value = 1 to produce EPS graphs
POSTSCRIPT|1
# value = 1 to export text data files
EXPORTS|1
# main logo file
LOGO_FILE|$WEBOBS{ROOT_CODE}/icons/ipgp/logo_WOVS.png
# secondary logo file
LOGO2_FILE|
# main copyright
COPYRIGHT|MyProc
# secondary copyright
COPYRIGHT2|

#STREAM_CHANNEL_SELECTOR|HHZ
#DATALINK_DELAY_SECONDS|600

# --- proc's specific parameters
# main duration for a single graph
HELICORDER_DURATION_DAYS|1
# number of turns per duration
HELICORDER_TURNS|24*2
# vertical scale (relative to the signal standard deviation)
HELICORDER_SCALE|100
# background paper color (accepts html names, see CODE/matlab/htm2rgb.m)
HELICORDER_PAPER_COLOR|#0F0F0F
# trace color or color list
#HELICORDER_TRACE_COLOR|black,red,mediumblue,green
HELICORDER_TRACE_COLOR|whitesmoke
# ytick labels
HELICORDER_YTICK_HOURS|2
# galvanometer rotation effect (use 0 to unset)
HELICORDER_RADIUS|1
# traces trend effect
HELICORDER_TREND|N
# vertical time direction
HELICORDER_TOPDOWN|N

# --- data processing
# removes data picks using median filter (% of min/max values)
PICKS_CLEAN_PERCENT|0

# replaces flat signal (two consecutive identical values) by NaN
FLAT_IS_NAN|NO

# applies a median filter to the data
MEDIAN_FILTER_SAMPLES|0
```

3.7 RSAM: Realtime Seismic Amplitude Measurement

3.7.1 Overview

3.7.2 Configuration

Some parameter keys of RSAM are common for all PROCS and GRIDS so are identical with the VIEW configuration (see section 2.3.2).

RSAM template

```
=key|value
# RSAM PROC template
# See $WEBOBS{ROOT_CODE}/matlab/superprocs/rsam.m for further information.

# long name of the PROC
NAME|Realtime Seismic Amplitude Measurement
# proc TYPE and OWNER
TYPE|
OWNCODE|?
# default raw data format (might be overwritten in each node)
RAWFORMAT|
# raw data path (might be overwritten in each node)
RAWDATA|
# URN link to data
URNDATA|/rawdata/myproc
# additionnal URL
URL|
# time zone for outputs
TZ|0

# lists of parameters (number of comma-separated rows must be consistent)
# TIMESCALELIST valid suffix: h (hour), d (day), w (week), m (month), y (year)
TIMESCALELIST|04h,24h
DECIMATELIST|1,1
CUMULATELIST|1,1
DATESTRLIST|-1,-1
MARKERSIZELIST|2,1
LINEWIDTHLIST|.2,.1
STATUSLIST|1,0
# defines a reference date to use with 'r01' timescale in TIMESCALELIST
REF01_DATE|2000-01-01
# year and month timescales use true durations
TIMESCALE_TRUEVALUE|Y

# execution command (for runproc shell)
SUBMIT_COMMAND|$WEBOBS{JOB_MCC} rsam $SELFREF -
SUBMIT_RESOURCE|rsam

# list of keys editable in requests
REQUEST_KEYLIST|NAME,REF_DATE

# --- node parameters
NODE_NAME|node
NODE_SIZE|10
NODE_RGB|Red
NODE_FONTSIZE|10
NODE_MARKER|s

# auto refresh of graphic display
DISPLAY_AUTOREFRESH_SECONDS|3600

# --- optional user-defined DEM, in ArcInfo format (overwrites SRTM/ETOPO default)
#DEM_FILE|$WEBOBS{PATH_DATA_DEM}/exampledem.asc
# DEM coordinate system: UTM or LATLON
DEM_TYPE|LATLON
DEM_COPYRIGHT|
DEM_SRTM1|Y

# additional zoomed map(s) using fixed axis limits: LON1,LON2,LAT1,LAT2
#MAP1_XYLIM|
#MAP2_XYLIM|
# do not show node alias in zoom map(s)
NODE_SUBMAP_ALIAS|N

# file for background time referenced events
EVENTS_FILE|$WEBOBS{ROOT_CONF}/events_World.conf

# --- page format and outputs
# optional paper size width,height in inches
PAPER_SIZE|
# graphical density for PNG outputs (in pixel per inches)
PPI|100
# landscape format for all outputs
LANDSCAPE|NO
# maximum number of subplots in a standard page height
PAGE_MAX_SUBPLOT|8
# makes a PDF file for each graph (in addition to PNG, EPS and JPG thumbnail)
PDFOUTPUT|NO
```

```
# main logo file
LOGO_FILE|$WEBOBS{ROOT_CODE}/icons/ipgp/logo_WOVS.png
# main copyright
COPYRIGHT|Processing by Taisne et al./IPGP/EOS
# optional secondary logo file
LOGO2_FILE|
# optional secondary copyright
COPYRIGHT2|
# events file to plot in the background of each graph
#EVENTS_FILE|$WEBOBS{ROOT_CONF}/events_World.conf
# exports text data files
#EXPORTS|YES

# --- data processing and plot
# removes data picks using median filter (% of min/max values)
PICKS_CLEAN_PERCENT|0
# replaces flat signal (two consecutive identical values) by NaN
FLAT_IS_NAN|NO
# applies a median filter to the data
MEDIAN_FILTER_SAMPLES|0
# superimpose a moving average on N samples
MOVING_AVERAGE_SAMPLES|12*60
# set to YES to plot continuous line ignoring gaps
CONTINUOUS_PLOT|NO
# set Y-axis with a log scale
YLOGSCALE|YES

# --- alarm
ALARM_THRESHOLD_LEVEL|1e-5
ALARM_COLOR|IndianRed
NOTIFY_EVENT|rsamalert.

# --- per node graphs
# graph title
PERNODE_TITLE|{\fontsize{14}{\bf$node_alias: $node_name} ($timescale)}
# line style
PERNODE_LINESTYLE|-

# --- summary graph (all nodes together)
# comment this key to not plot the summary graph
SUMMARYLIST|SOURCEMAP

# coma separated list of channels to plot (default is all channels)
SUMMARY_CHANNELS|1
# graph title
SUMMARY_TITLE|{\fontsize{14}{\bf${NAME}} ($timescale)}
# line style
SUMMARY_LINESTYLE|-

# --- source mapping
# number of maps will be N*N (allowed values are 1,2,3 or 4)
SOURCEMAP_N|3
SOURCEMAP_COLORMAP|jet
SOURCEMAP_COLORMAP_ALPHA|0,1
SOURCEMAP_CAXIS|0,2e-5
SOURCEMAP_PLOT_MAX|Y
```

Figure 3.4: Example of overview graph page created by a proc GNSS.

3.8 GNSS: GPS time series, vectors and modelling

3.8.1 Overview

GNSS is a superproc dedicated to displacement data from global navigation satellite systems processing for instance solutions from Gispy, Gamit/Globk, ... It handles local velocity referential, some data filtering, and produces following graphs (summary graphs activated by setting `SUMMARYLIST` to the appropriated keyword list, coma separated), for each timescale:

- default is one graph per NODE showing time series of the 3 component displacements (East, North and Up);
- one summary graph `SUMMARY` as time series combining all NODES on the 3 component subplots;
- one summary graph `BASELINES` as time series computing 1-D distance variation between a selection of node pairs;
- one summary graph `VECTORS` as map of velocity vectors and amplitude vs. distance optional graph;
- one summary graph `MODELLING` showing the potential source of deformation, estimating the location, volume variation / flux rate and source shape as result of point-source inversion in elastic medium (source type is isotropic or complex source);
- one experimental summary graph `MODELTIME` showing time series of model sources and time location evolution.

3.8.2 Configuration

Some parameter keys of GNSS are common for all PROCS and GRIDS so are identical with the VIEW configuration (see section 2.3.2) and genplot PROC configuration (see section 3.4).

Data format

Following generic behavior of superprocs, data source and format can be set for all associated NODES using `RAWDATA` and `RAWFORMAT` keywords (see section 5.1), or overwritten for any individual NODE. The GNSS superproc needs a particular data structure which is:

- Channel 1: Eastern coordinate UTM (in meters)
- Channel 2: Northern coordinate UTM (m)
- Channel 3: Elevation or altitude coordinate (m)
- Channel 4: Orbit type (0 for final, 1 for rapid, 2 for ultra)

The GNSS superformat built-in formats returns this data structure (see section 3.14.4).

Preset time scales (real-time moving windows)

Time scales are defined as for GENPLOT superproc (see section 3.4).

Data filtering

Some filters can be applied to the whole set of data:

- `PICKS_CLEAN_PERCENT` removes data picks using a threshold as % of min/max of the data values after linear detrend;
- `PICKS_CLEAN_STD` removes data picks using a threshold as a multiple of standard deviation of data around the mean value after linear detrend;

Generic graphical options

Other page settings can also be configured:

- the line width and marker size can be different for each time scale using respectively LINEWIDTHLIST and MARKERSIZELIST, that should both define a size to use (in pt) for each of the timescales defined in TIMESCALELIST, coma separated values;
- colors are chosen automatically: for the summary graph it will be one color per node, for the per-node graph it will be one color per channel;
- PLOT_GRID adds grid lines to all axis;
- EVENTS_FILE set a filename (or a coma-separated list of filenames) containing a list of time-referenced events that will be plotted in the background of each graph (see section 3.4).
- PAGESIZE sets the page width and height in inches (default is 8.5,11 for usletter) for PostScript image output;
- PPI sets the point-per-inch resolution to convert the EPS in PNG raster images (default is 100 so standard image outputs are 800×1100 pixels);
- a logo can be added to the page at upper-left corner using LOGO_FILE (full path of image, default is a **WebObs** logo defined by LOGO_DEFAULT in **WEBOBS.rc**) and LOGO_HEIGHT (a fraction of page's height, default is 0.04, i.e., 4%). The aspect ratio of image will be preserved. It is also possible to set a second logo that will be plotted at upper-right corner using LOGO2_FILE and LOGO2_HEIGHT; LOGO_FILE and LOGO2_FILE might contain a list of coma separated filenames so images will be horizontally appended from left to right, with same height;
- two copyrights strings can be set using COPYRIGHT and COPYRIGHT2 (some TeX allowed). They will appear below the graph title;
- DISPLAY_AUTOREFRESH_SECONDS sets the time delay for web browser autorefresh of the PROC's output graph pages. Note that a dedicated icon is available in all graphs submenu to refresh manually if necessary.

GNSS template

```
=key|value
# Generic PROC template for superproc gnss
# Superformat is 'gnss': see readfmtdata_gnss.m for compatible RAWFORMAT defined
# in associated NODE's RAWFORMAT field
# See $WEBOBS{ROOT_CODE}/matlab/superprocs/gnss.m for further information.


# long name of the PROC
NAME|GNSS time series, vectors and modelling
RAWFORMAT|gipsyx
RAWDATA|$WEBOBS{ROOT_RAWD}/gipsyx
URNDATA|$WEBOBS{URN_RAWD}/gipsyx

# time zone for outputs
TZ|0

# additionnal external link(s): TEXT1,URL1;TEXT2,URL2;...
URL|

# lists of parameters (number of comma-separated rows must be consistent)
TIMESCALELIST|01y,all
DECIMATELIST|
CUMULATELIST|
DATESTRLIST|-1,-1
MARKERSIZELIST|3,1
LINEWIDTHLIST|2,.5
STATUSLIST|1,0
# defines a reference date to use with 'r01' timescale in TIMESCALELIST
REF01_DATE|2000-01-01
# year and month timescales use true durations
TIMESCALE_TRUEVALUE|Y

# summary graphs available: SUMMARY,VECTORS,MOTION,BASELINES,MODELLING,MODELTIME
SUMMARYLIST|SUMMARY,VECTORS,MOTION,BASELINES

# for scheduler requests
SUBMIT_COMMAND|$WEBOBS{JOB_MCC} gnss $SELFREF -
SUBMIT_RESOURCE|gipsyx

# list of keys editable in requests
REQUEST_KEYLIST|NAME,SUMMARYLIST,VECTORS_RELATIVE

DEM_SRTM1|N
# optional user-defined DEM (Arcinfo format, lat/lon)
#DEM_FILE|$WEBOBS{PATH_DATA_DEM}/highresdem_latlon.asc
#DEM_TYPE|LATLON
#DEM_COPYRIGHT|DEM: myDEM

# additional zoomed map(s) using fixed axis limits: LON1,LON2,LAT1,LAT2
#MAP1_XYLIM|
#MAP2_XYLIM|
# do not show node alias in zoom map(s)
```

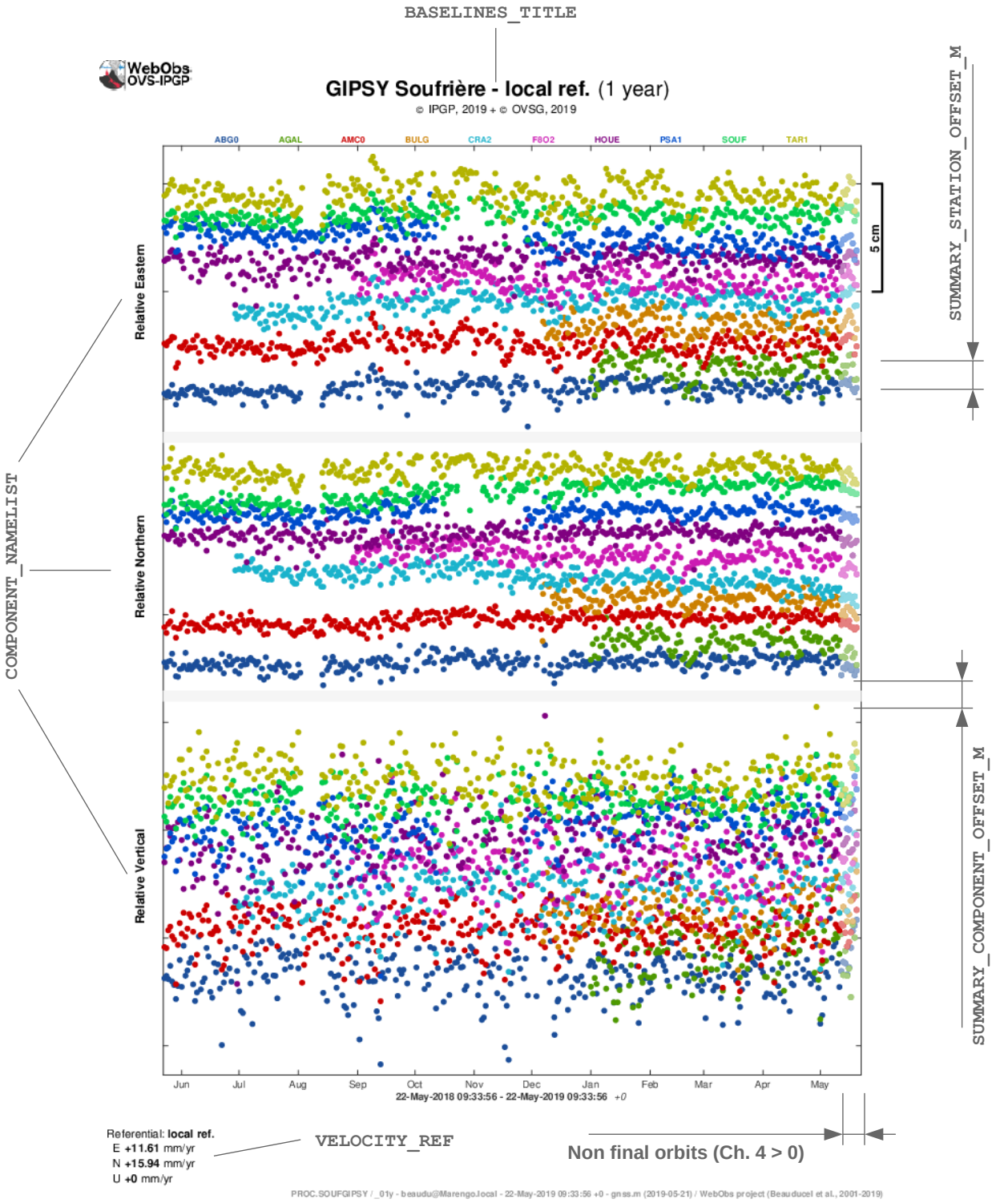


Figure 3.5: **GNSS superproc SUMMARY graph.** Example of summary graph where the raw coordinates are plotted in a local referential defined by the 3-component vector of velocities `VELOCITY_REF` (coma-separated, in mm/yr). All nodes (stations) are plotted for each component, using default color order, and possible watermark effect for the last data with non-final orbits. Line and marker type is set by `SUMMARY_LINESTYLE` (see tables 5.3 and 5.4) and error bars are not plotted. Stations time series are shifted by `SUMMARY_STATION_OFFSET_M` (in m) from their median value, overlaps are allowed. Each component is separated by an offset of `SUMMARY_COMPONENT_OFFSET_M` (in m) that separates min and max data between components (no overlap if the offset is positive). Title is defined by `SUMMARY_TITLE` string (some variables and TeX allowed). Y-label is defined by `COMPONENT_NAMELIST` key containing the 3 coma-separated strings for East, North and Vertical components respectively (some variables and TeX allowed).

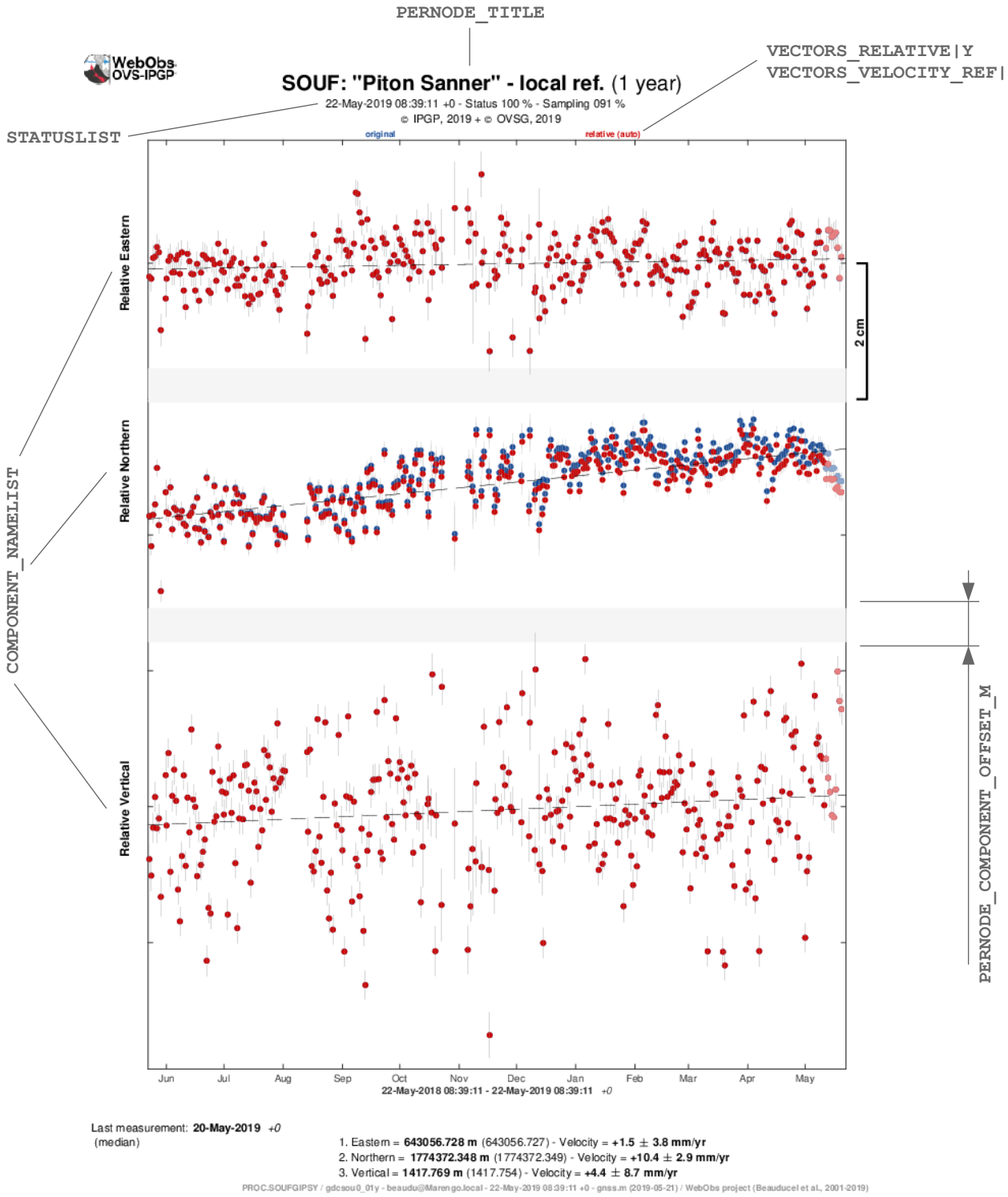


Figure 3.6: **GNSS superproc PERNODE graph.** Example of per-node graph that will be plotted for all nodes (stations). In this example, we use `VECTORS_RELATIVE` key as `Y` so original data (in blue) are corrected by a global velocity trend to make final relative components (in red). `VECTORS_VELOCITY_REF` is set to automatic (empty) so the global trend is computed as the network velocity mean including all station velocities but `VECTORS_EXCLUDED_NODELIST`. Line and marker type is set by `PERNODE_LINESTYLE` (see tables 5.3 and 5.4) and error bars as also plotted. Each component is separated by an offset of `PERNODE_COMPONENT_OFFSET_M` (in m). Title is defined by `PERNODE_TITLE` string (some variables and TeX allowed). Additional subtitle may contain last data and node status if the timescale is selected in `STATUSLIST`. Y-label is defined by `COMPONENT_NAMELIST` key containing the 3 coma-separated strings for East, North and Vertical components respectively (some variables and TeX allowed).

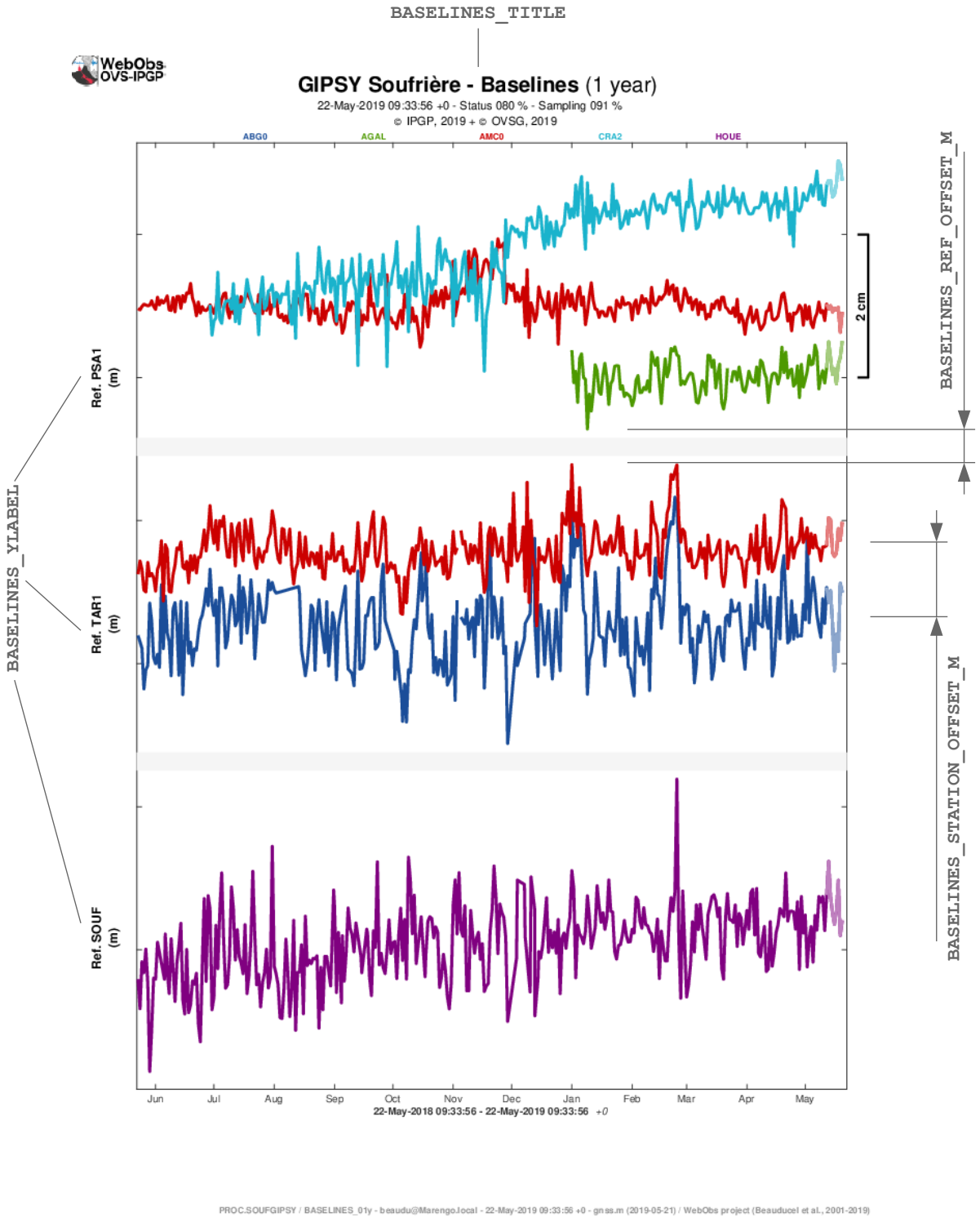


Figure 3.7: **GNSS superproc BASELINES graph.** Example of baselines made using `BASELINES_NODEPAIRS` key as `PSA1,AMC0,AGAL,CRA2;TAR1,ABG0,AMC0;SOUF,HOUE`, i.e., baselines relative to 3 separated reference stations: **PSA1** to **AMC0**, **AGAL** and **CRA2**; **TAR1** to **ABG0** and **AMC0**; and **SOUF** to **HOUE**. Line and marker type is set by `BASELINES_LINESTYLE` (here the default solid line `-`, see tables 5.3 and 5.4). Colors are the default color-order assigned to destination stations (identical to other graphs). Stations time series are shifted by `BASELINES_STATION_OFFSET_M` (in m) from their median value, overlaps are allowed. Each reference station subplot is separated by an offset of `BASELINES_REF_OFFSET_M` (in m) that separates min and max data between each subplot (no overlap if the offset is positive). Title is defined by `BASELINES_TITLE` string (some variables and TeX allowed). Y-label is defined by `VECTORS_YLABEL` string (some variables and TeX allowed).

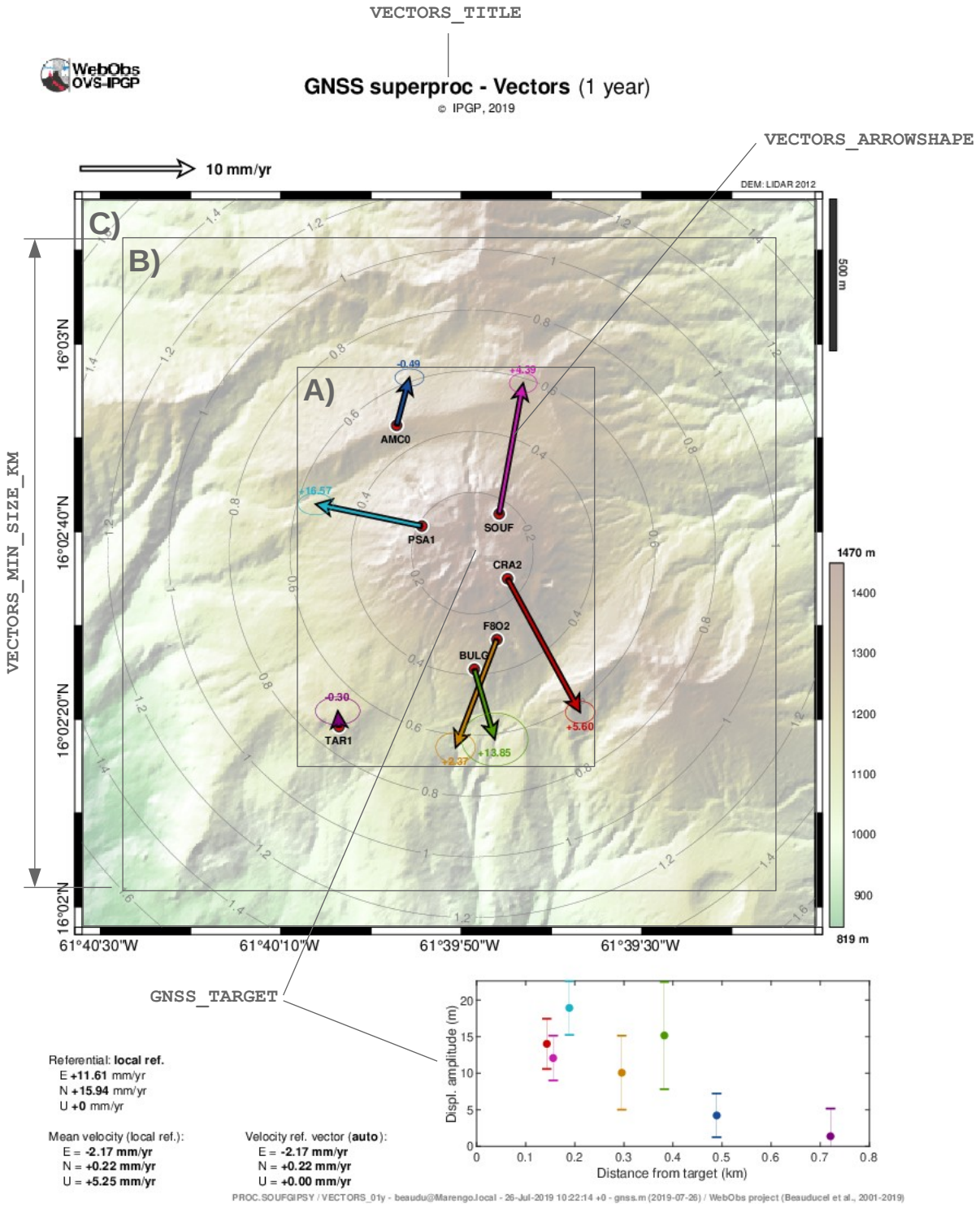


Figure 3.8: **GNSS superproc VECTORS graph.** Velocity vectors are drawn using default colors for each node and `VECTORS_ARROWSHAPE` parameters to define the arrow shape relative to the legend arrow size (see table 5.10). End-arrow numbers indicate the vertical component values. Map limits are computed as follows: A) automatic box surrounding all stations and vectors including error ellipses; B) eventually extended to fit the `VECTORS_MIN_SIZE_KM` minimum size and/or to respect the `VECTORS_MAX_XYRATIO` size ratio between width and height (default is `1` to force a square map); C) finally extended by 10% borders. Background map uses SRTM by default or user-defined DEM if specified and does cover the whole needed area. Title is defined by `VECTORS_TITLE` string (some TeX allowed).

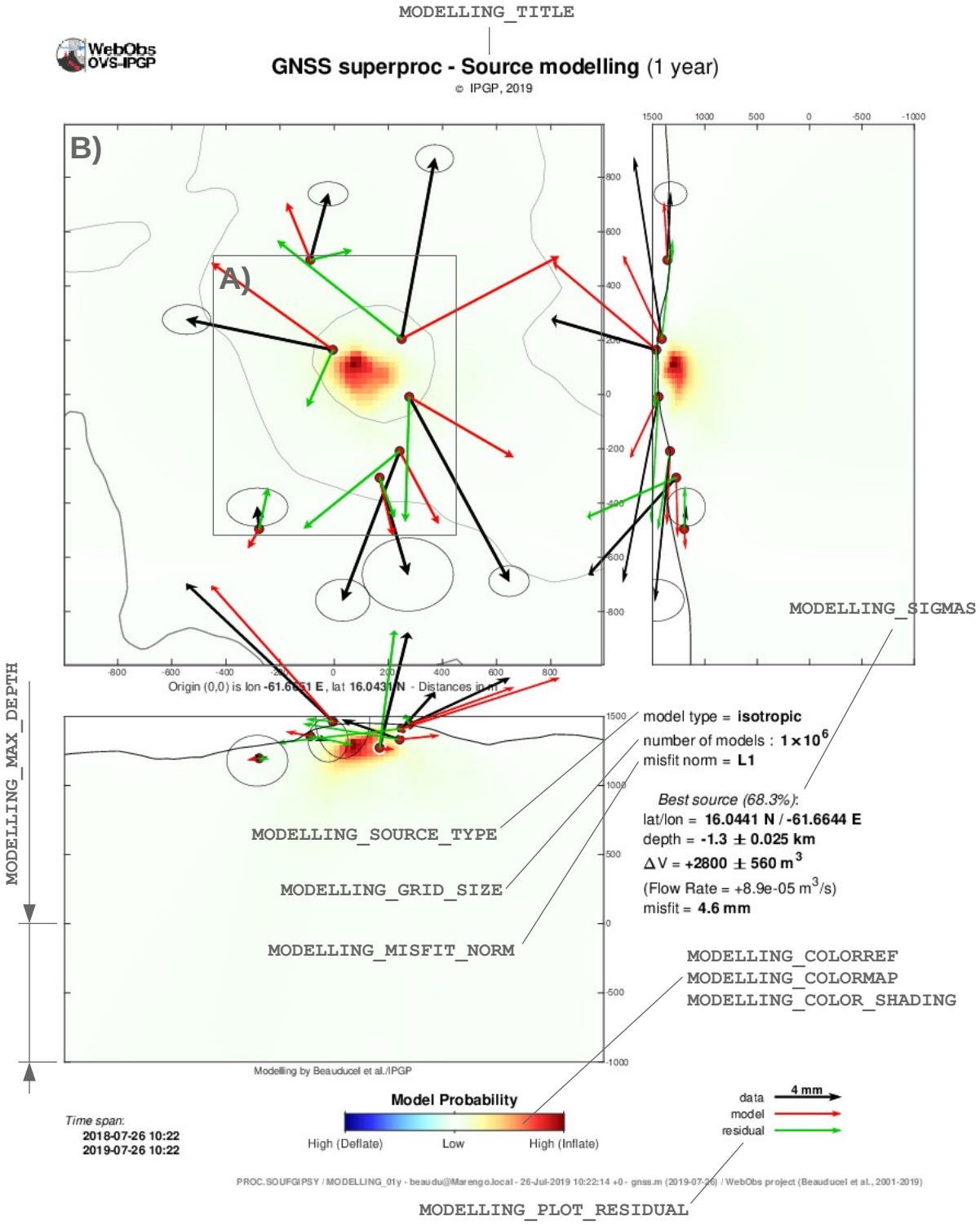


Figure 3.9: **GNSS superproc MODELLING graph.** Displacement vectors are drawn for all stations with error ellipses (in black), best model vectors are plotted in red, and optionally residuals vectors in green (if `MODELLING_PLOT_RESIDUAL` is set to `Y`). 3-D model space limits are computed as follows: A) automatic box surrounding all stations; B) extension of horizontal limits of `MODELLING_BORDERS` (in m) and extension of vertical depth to `MODELLING_MAX_DEPTH` (in m below sea level). The topography is using SRTM3 exclusively. Model space grid has the cubed `MODELLING_GRID_SIZE` elements (here 100^3). Analytical point-source model type is `MODELLING_SOURCE_TYPE` (default `isotropic`). Misfit norm is `MODELLING_MISFIT_NORM` (default `L1`). Estimations of best source model parameter uncertainties are `MODELLING_SIGMAS` sigma (default $1\sigma = 68.3\%$). Colormap used for model probability image is `MODELLING_COLORREF` type (default `volpdf`), `MODELLING_COLORMAP` colormap (default `jet`, see table 5.6) with a shading factor of `MODELLING_COLOR_SHADING` (default `0.8`). Title is defined by `MODELLING_TITLE` string (some TeX allowed).

```
NODE_SUBMAP_ALIAS|N

# file for background time referenced events
EVENTS_FILE|$WEBOBS{ROOT_CONF}/events_World.conf

PAPER_SIZE|8,11
PPI|100
LANDSCAPE|N
PLOT_GRID|N

FONTSIZE|8
POSTSCRIPT|1
EXPORTS|1
LOGO_FILE|$WEBOBS{ROOT_CODE}/icons/ipgp/logo_WOVS.png
LOGO2_FILE|
COPYRIGHT|
COPYRIGHT2|

NODE_NAME|station
NODE_MARKER|o
NODE_SIZE|6
NODE_RGB|1,0,0
NODE_FONTSIZE|8

# auto refresh of graphic display
DISPLAY_AUTOREFRESH_SECONDS|3600


# ----------------------------------------------------------------------
# specific proc's parameters


# maximum error on position (in m) before excluding the data from graphs and calculations
FILTER_MAX_ERROR_M|0.05

PICKS_CLEAN_PERCENT|0
TREND_ERROR_MODE|1
# minimum time window (in days) needed to compute a trend
TREND_MIN_DAYS|5

# ITRF reference (this is a string displayed on graph title, not functional)
ITRF_REF|ITRF08
# Relative velocity reference E,N,U (mm/yr) from ITRF = constant trend
# substracted to all data before any other processing
VELOCITY_REF|0,0,0
VELOCITY_REF_ORIGIN_DATE|2000-01-01

# component names used as Y-label for SUMMARY and PERNODE graphs (use '\ ' to avoid word splitting)
COMPONENT_NAMELIST|Relative Eastern,Relative Northern,Relative Vertical

# --- PERNODE graphs
PERNODE_LINESTYLE|o
PERNODE_TITLE|{\fontsize{14}{\bf$node_alias: $node_name - $velref} ($timescale)}
# offset between components (in m)
PERNODE_COMPONENT_OFFSET_M|0.01
# adds a secondary subplot time-zoomed for the most recent data (fraction of timescale)
PERNODE_TIMEZOOM|

# --- SUMMARY graphs
SUMMARY_LINESTYLE|o
SUMMARY_TITLE|{\fontsize{14}{\bf${NAME} - $velref} ($timescale)}
# adds a secondary subplot time-zoomed for the most recent data (fraction of timescale)
SUMMARY_TIMEZOOM|
# offset between components (in m)
SUMMARY_COMPONENT_OFFSET_M|0.01
# offset between stations (in m)
SUMMARY_STATION_OFFSET_M|0.01

# --- BASELINES graphs
# ignores vertical component
BASELINES_HORIZONTAL_ONLY|Y
# - default behavior: all pairs of nodes, with possible node exclusion and/or node reference list
BASELINES_EXCLUDED_NODELIST|
BASELINES_REF_NODELIST|
# - fixed pairs (overwrites BASELINES_EXCLUDED_NODELIST and BASELINES_REF_NODELIST): REF1,STA1;REF2,STA1,STA2,STA5
BASELINES_NODEPAIRS|
# interpolation method between nodes: linear or nearest
BASELINES_INTERP_METHOD|linear
BASELINES_TITLE|{\fontsize{14}{\bf${NAME} - Baselines} ($timescale)}
BASELINES_UNIT|mm
BASELINES_YLABEL|Ref.\ $ref_node_alias (${BASELINES_UNIT})
BASELINES_LINESTYLE|-
# offset between references (in m)
BASELINES_REF_OFFSET_M|0.001
# offset between stations (in m)
BASELINES_STATION_OFFSET_M|0.01
# adds a secondary subplot time-zoomed for the most recent data (fraction of timescale)
BASELINES_TIMEZOOM|

# --- VECTORS: excluding NODES from graphs and calculations
VECTORS_EXCLUDED_NODELIST|
```

```
VECTORS_RELATIVE|N
# VECTORS: if VECTORS_RELATIVE is Y, substracts a reference vector in VECTORS_VELOCITY_REF:
# - 'auto' (default): automatic mean velocity (average vector of all NODES)
# - NODE's FID name or list of NODES: use the velocity vector (or mean velocity vector)
# - E,N,U: sets fixed offset values (in mm/yr)
VECTORS_VELOCITY_REF|auto
VECTORS_RELATIVE_HORIZONTAL_ONLY|Y
# background map parameters (see CODE/matlab/dem.m for available options)
VECTORS_DEM_OPT|'watermark',3,'interp','legend','seacolor',[0.7,0.9,1]
VECTORS_TITLE|{\fontsize{14}{\bf${NAME} - Velocity vectors} ($timescale)}
VECTORS_MIN_SIZE_KM|5
VECTORS_MAX_XYRATIO|1
VECTORS_ARROWSHAPE|.15,.15,.12,.03
VECTORS_TARGET_INCLUDED|Y

# defines a main target (latitude,longitude coordinates)
GNSS_TARGET_LATLON|

# --- MOTION summary graph
MOTION_EXCLUDED_NODELIST|
MOTION_MAFILTER|10
MOTION_SCALE_MM|0
MOTION_MIN_SIZE_KM|10
MOTION_COLORMAP|jet(256)
MOTION_DEM_OPT|'colormap',.5*ones(64,3),'watermark',2,'interp'
MOTION_TITLE|{\fontsize{14}{\bf${NAME} - Motion} ($timescale)}
MOTION_TARGET_INCLUDED|Y

# --- MODELLING graph
# excluding NODES from graphs and calculations
MODELLING_EXCLUDED_NODELIST|
MODELLING_EXCLUDED_FROM_TARGET_KM|
# ignores vertical component
MODELLING_HORIZONTAL_ONLY|Y
# recomputes relative velocities before modelling (instead of vectors results)
MODELLING_FORCE_RELATIVE|N
# modelling source: 'isotropic' (formerly 'mogi') or 'pcdm' (point Compound Dislocation Model)
MODELLING_SOURCE_TYPE|isotropic
# models the residuals with a secondary source (EXPERIMENTAL)
MODELLING_MULTIPLE_SOURCES|1
# misfit norm calculation: 'L1' (default) or 'L2'
MODELLING_MISFITNORM|L1
# a priori horizontal error around the target or grid center (in STD, km), 0 = no a priori
MODELLING_APRIORI_HSTD_KM|0
# set a minimum error on displacements, absolute (mm) and relative (%) to avoid NaN PDF when misfit is too high
MODELLING_MINERROR_MM|5
MODELLING_MINERROR_PERCENT|1
# vector scale (mm), 0 for automatic
MODELLING_VMAX_MM|0
# maximum vector length to plot width ratio
MODELLING_VMAX_RATIO|.25
# clip vectors outside the plot axis
MODELLING_VECTORS_CLIP|NO
# indicates the best source solution on graph
MODELLING_PLOT_BEST|Y
# adds residual arrows (data - model)
MODELLING_PLOT_RESIDUAL|Y
# number of sigma (STD) to compute uncertainty intervals for best model
MODELLING_SIGMAS|1
# RGB color for topo/bathy basemap
MODELLING_TOPO_RGB|.5,.5,.5
# color reference for model space: 'pdf' or 'volpdf' (source volume sign x pdf, new default)
MODELLING_COLORREF|volpdf
MODELLING_COLORMAP|jet(256)
MODELLING_COLOR_SHADING|0.8
MODELLING_MAX_DEPTH|8000
MODELLING_BORDERS|5000
MODELLING_GRID_SIZE|50
MODELLING_TITLE|{\fontsize{14}{\bf${NAME} - Source modelling} ($timescale)}

# specific pCDM parameters for MODELLING
MODELLING_PCDM_ITERATIONS|5
MODELLING_PCDM_RANDOM_SAMPLING|100000
MODELLING_PCDM_NU|0.25
MODELLING_PCDM_DVLIM|-1e7,1e7
MODELLING_PCDM_ALIM|0,1
MODELLING_PCDM_BLIM|0,1
MODELLING_PCDM_OXLIM|-45,45
MODELLING_PCDM_OYLIM|-45,45
MODELLING_PCDM_OZLIM|-45,45
MODELLING_PCDM_HEATMAP_GRID|50
MODELLING_PCDM_HEATMAP_SATURATION|0.4
MODELLING_PCDM_HEATMAP_SMOOTH_SPAN|5
MODELLING_PCDM_HEATMAP_SMOOTH_DEGREE|1
MODELLING_PCDM_NEW_THRESHOLD|2
MODELLING_PCDM_NEW_LIMIT_EDGE_RATIO|20
MODELLING_PCDM_NEW_LIMIT_EXTEND|1
MODELLING_PCDM_SUPPLEMENTARY_GRAPHS|N

# --- MODELTIME: defines the moving periods, minimum sampling, maximum models
# [NOTE] MODELTIME uses all MODELLING parameters but forces an isotropic source
```

```
MODELTIME_PERIOD_DAY|30,15
MODELTIME_SAMPLING_DAY|2
MODELTIME_MAX_MODELS|50
MODELTIME_FLOWRATE|Y
MODELTIME_MAP_PERIODLIST|
MODELTIME_MARKER_LINEWIDTH|1
MODELTIME_COLORMAP|jet(256)
MODELTIME_MARKERSIZE|10
MODELTIME_TITLE|{\fontsize{14}{\bf$name - Source best model timeline} ($timescale)}
```

3.9 EXTENSO: Extensometry time series and vectors

3.9.1 Overview

3.9.2 Configuration

Some parameter keys of EXTENSO are common for all PROCS and GRIDS so are identical with the VIEW configuration (see section 2.3.2).

EXTENSO template

```
=key|value
# Generic PROC template for superproc extenso
# This PROC must be associated with the FORM "EXTENSO" which contains
# compatible data and needed complementary configuration files.
# See $WEBOBS{ROOT_CODE}/matlab/superprocs/extenso.m for further information.

NAME|Extensometry time series and vectors
RAWDATA|
URNDATA|
TZ|-4
URL|

# lists of parameters (number of coma-separated rows must be consistent)
TIMESCALELIST|10y,all
DECIMATELIST|
CUMULATELIST|
DATESTRLIST|-1,-1
MARKERSIZELIST|16,12
LINEWIDTHLIST|2,1
STATUSLIST|1,0
# defines a reference date to use with 'r01' timescale in TIMESCALELIST
REF01_DATE|2000-01-01
# year and month timescales use true durations
TIMESCALE_TRUEVALUE|Y

# additional summary graphs keywords (comma separated)
SUMMARYLIST|VECTORS

# execution command (for runproc shell)
SUBMIT_COMMAND|
SUBMIT_RESOURCE|myproc

NODE_NAME|site
NODE_MARKER|^
NODE_RGB|1,0,0
NODE_SIZE|10
NODE_FONTSIZE|8

# set to YES to use SRTM1 30m resolution tiles
DEM_SRTM1|NO

# optional user-defined DEM (Arcinfo format, lat/lon) overwriting SRTM/ETOPO default
#DEM_FILE|$WEBOBS{PATH_DATA_DEM}/highresdem_latlon.asc
#DEM_TYPE|LATLON
#DEM_COPYRIGHT|DEM: myDEM

# additional zoomed map(s) using fixed axis limits: LON1,LON2,LAT1,LAT2
#MAP1_XYLIM|
#MAP2_XYLIM|
# do not show node alias in zoom map(s)
NODE_SUBMAP_ALIAS|N

# auto refresh of graphic display
DISPLAY_AUTOREFRESH_SECONDS|3600

# time zone for outputs
TZ|0

# additionnal external link(s): TEXT1,URL1;TEXT2,URL2;...
URL|

# file for background time referenced events
EVENTS_FILE|$WEBOBS{ROOT_CONF}/events_World.conf

PPI|100
POSTSCRIPT|1
EXPORTS|1
LOGO_FILE|$WEBOBS{ROOT_CODE}/icons/ipgp/logo_WOVS.png
LOGO2_FILE|
COPYRIGHT|Processing by Beauducel et al./IPGP
COPYRIGHT2|

# specific PROC's parameters
FILTER_MAX_ERROR_MM|10
ZONE1_NAME|Zone Nord
ZONE1_NODELIST|FNW1,DUP1,DUP2,DUP3,FNO1
ZONE2_NAME|Zone Sud-Est
ZONE2_NODELIST|NAP1,F8J1,BLK1,LCX1,CSD1
```

```
ZONE3_NAME|Zone Sud
ZONE3_NODELIST|DOL1,DOL2,PEY1,F302,F303

VECTORS_EXCLUDED_NODELIST|BAT0,HOU1
VECTORS_ARROWSHAPE|.15,.1,.1,.04
VECTORS_VELOCITY_SCALE|0
VECTORS_MIN_SIZE_KM|.5
VECTORS_MAX_XYRATIO|2
```

3.10 TILT: Tiltmetry time series, vectors and modelling

3.10.1 Overview

3.10.2 Configuration

Some parameter keys of TILT are common for all PROCS and GRIDS so are identical with the VIEW configuration (see section 2.3.2).

TILT template

```
=key|value
# Generic PROC template for superproc tilt
# See $WEBOBS{ROOT_CODE}/matlab/superprocs/tilt.m for further information.

# long name of the PROC
NAME|Tiltmetry time series, vectors and modelling
# proc TYPE and OWNER code (see OWNERS.conf for OWNCODEs)
TYPE|
OWNCODE|?
RAWFORMAT|
# raw data path
RAWDATA|
# URN link to data
#URNDATA|/rawdata/myproc
# time zone for outputs
TZ|+0
# additionnal URL
URL|

# lists of parameters (number of comma-separated rows must be consistent)
# TIMESCALELIST valid suffix: h (hour), d (day), w (week), m (month), y (year)
TIMESCALELIST|24h,01w
DECIMATELIST|1,1
CUMULATELIST|1,1
DATESTRLIST|-1,-1
MARKERSIZELIST|6,3
LINEWIDTHLIST|1,1
STATUSLIST|1,0
# defines a reference date to use with 'r01' timescale in TIMESCALELIST
REF01_DATE|2000-01-01
# year and month timescales use true durations
TIMESCALE_TRUEVALUE|Y

# additional summary graphs keywords (comma separated)
SUMMARYLIST|VECTORS
SUMMARY_CHANNELS|1,2,3

# execution command (for runproc shell)
SUBMIT_COMMAND|$WEBOBS{JOB_MCC} tilt $SELFREF -
SUBMIT_RESOURCE|myproc

# list of keys editable in requests
REQUEST_KEYLIST|NAME,VECTORS_EXCLUDED_NODELIST,MODELLING_EXCLUDED_NODELIST

# node parameters
NODE_NAME|station
NODE_SIZE|10
NODE_RGB|1,0,0
NODE_FONTSIZE|10
NODE_MARKER|o

# auto refresh of graphic display
DISPLAY_AUTOREFRESH_SECONDS|3600

# --- optional user-defined DEM, in ArcInfo format (overwrites SRTM/ETOPO default)
DEM_FILE|$WEBOBS{PATH_DATA_DEM}/exampledem.asc
# DEM coordinate system: UTM or LATLON
DEM_TYPE|LATLON
DEM_COPYRIGHT|

# additional zoomed map(s) using fixed axis limits: LON1,LON2,LAT1,LAT2
#MAP1_XYLIM|
#MAP2_XYLIM|
# do not show node alias in zoom map(s)
NODE_SUBMAP_ALIAS|N
```

```
# file for background time referenced events
EVENTS_FILE|$WEBOBS{ROOT_CONF}/events_World.conf

LOGO_FILE|$WEBOBS{ROOT_CODE}/icons/ipgp/logo_WOVS.png
COPYRIGHT|Processing by Beauducel et al./IPGP
LOGO2_FILE|
COPYRIGHT2|

# graphical density for PNG outputs (in pixel per inches)
PPI|100
# value = 1 to produce EPS graphs
POSTSCRIPT|1
# value = 1 to export text data files
EXPORTS|1
PAPER_SIZE|8,11

PICKS_CLEAN_PERCENT|0
FLAT_IS_NAN|NO
# applies a median filter to the data
MEDIAN_FILTER_SAMPLES|0
TREND_ERROR_MODE|1

# defines the channels
TILTX_CHANNEL|1
TILTY_CHANNEL|2
TEMPERATURE_CHANNEL|3
TILT_TARGET_LATLON|

PERNODE_TEMPERATURE_BACKGROUND|N
PERNODE_TEMPERATURE_COLOR|1,.7,.7

MOTION_EXCLUDED_NODELIST|
MOTION_MAFILTER|1
MOTION_SCALE_RAD|0
MOTION_MIN_SIZE_KM|10
MOTION_COLORMAP|jet(64)
MOTION_DEM_OPT|'colormap',.5*ones(64,3),'watermark',2,'interp'
MOTION_TITLE|{\fontsize{14}{\bf$name - Motion} ($timescale)}

VECTORS_TITLE|{\fontsize{14}{\bf$name} ($timescale)}
VECTORS_EXCLUDED_NODELIST|
VECTORS_DEM_OPT|'watermark',2,'interp','legend'
VECTORS_VELOCITY_SCALE|
VECTORS_TARGET_INCLUDED|Y
VECTORS_MIN_SIZE_KM|10
VECTORS_MAX_XYRATIO|1.5
VECTORS_ARROWSHAPE|.1,.1,.08,.02
VECTORS_TOPO_RGB|
VECTORS_SHAPE_FILE|

MODELLING_TITLE|{\fontsize{14}{\bf$name} ($timescale)}
MODELLING_EXCLUDED_NODELIST|
MODELLING_MAX_DEPTH|5000
MODELLING_BORDERS|2500
MODELLING_GRID_SIZE|100
MODELLING_SIGMAS|1
```

3.11 METEO: meteorological time series

3.11.1 Overview

3.11.2 Configuration

Some parameter keys of METEO are common for all PROCS and GRIDS so are identical with the VIEW configuration (see section 2.3.2).

METEO template

```
=key|value
# Generic PROC template for superproc meteo
# See $WEBOBS{ROOT_CODE}/matlab/superprocs/meteo.m for data format and further information.


# long name of the PROC
NAME|Meteorological time series
RAWDATA|$WEBOBS{ROOT_RAWD}/campbell
RAWFORMAT|cr10xasc

# proc TYPE and OWNER code (see OWNERS.conf for OWNCODEs)
TYPE|
OWNCODE|?
LOGO_FILE|$WEBOBS{ROOT_CODE}/icons/ipgp/logo_WOVS.png
TZ|0

# lists of parameters (number of comma-separated rows must be consistent)
TIMESCALELIST|24h,30d,01y,10y,all
```

```
DECIMATELIST|1,1,6,6,6
CUMULATELIST|1/24,1,1,30,30
DATESTRLIST|-1,-1,-1,-1,-1
MARKERSIZELIST|6,4,2,1,1
LINEWIDTHLIST|1,1,1,1,1
STATUSLIST|1,0,0,0,0
# defines a reference date to use with 'r01' timescale in TIMESCALELIST
REF01_DATE|2000-01-01
# year and month timescales use true durations
TIMESCALE_TRUEVALUE|Y

#SUMMARYLIST|

# execution command (for runproc shell)
SUBMIT_COMMAND|$WEBOBS{JOB_MCC} meteo $SELFREF -
SUBMIT_RESOURCE|mymeteo

# list of keys editable in requests
REQUEST_KEYLIST|NAME

# node parameters
NODE_NAME|station
NODE_SIZE|15
NODE_RGB|1,0,0
NODE_FONTSIZE|10
NODE_MARKER|o

# auto refresh of graphic display
DISPLAY_AUTOREFRESH_SECONDS|3600

# --- optional user-defined DEM, in ArcInfo format (overwrites SRTM/ETOPO default)
DEM_FILE|$WEBOBS{PATH_DATA_DEM}/exampledem.asc
# DEM coordinate system: UTM or LATLON
DEM_TYPE|LATLON
DEM_COPYRIGHT|

# additional zoomed map(s) using fixed axis limits: LON1,LON2,LAT1,LAT2
#MAP1_XYLIM|
#MAP2_XYLIM|
# do not show node alias in zoom map(s)
NODE_SUBMAP_ALIAS|N

# file for background time referenced events
EVENTS_FILE|$WEBOBS{ROOT_CONF}/events_World.conf

# graphical density for PNG outputs (in pixel per inches)
PPI|100
# landscape format for all outputs
LANDSCAPE|N
# value = 1 to produce EPS graphs
POSTSCRIPT|1
# value = 1 to export text data files
EXPORTS|1
# main logo file
LOGO_FILE|$WEBOBS{ROOT_CODE}/icons/ipgp/logo_WOVS.png
# secondary logo file
LOGO2_FILE|
# main copyright
COPYRIGHT|MyProc
# secondary copyright
COPYRIGHT2|

# ------------------------------------------------------------------------
# specific proc's parameters

RAIN_CHANNEL|6
RAIN_CUMSUM_DATA|NO
WIND_SPEED_CHANNEL|5
WIND_AZIMUTH_CHANNEL|4
WIND_ROSE_STEP|10

XY_CHANNELS|3,8
NODE_CHANNELS|1,2,7,4,5,3,8

RAIN_ALERT_THRESHOLD|50
RAIN_ALERT_INTERVAL|1
RAIN_ALERT_DELAY|3
RAIN_ALERT_RGB|1,.3,.3
RAIN_ALERT_DELAY_RGB|1,.6,.6
```

3.12 WATERS: chemical analysis

3.12.1 Overview

3.12.2 Configuration

Some parameter keys of WATERS are common for all PROCS and GRIDS so are identical with the VIEW configuration (see section 2.3.2).

WATERS template

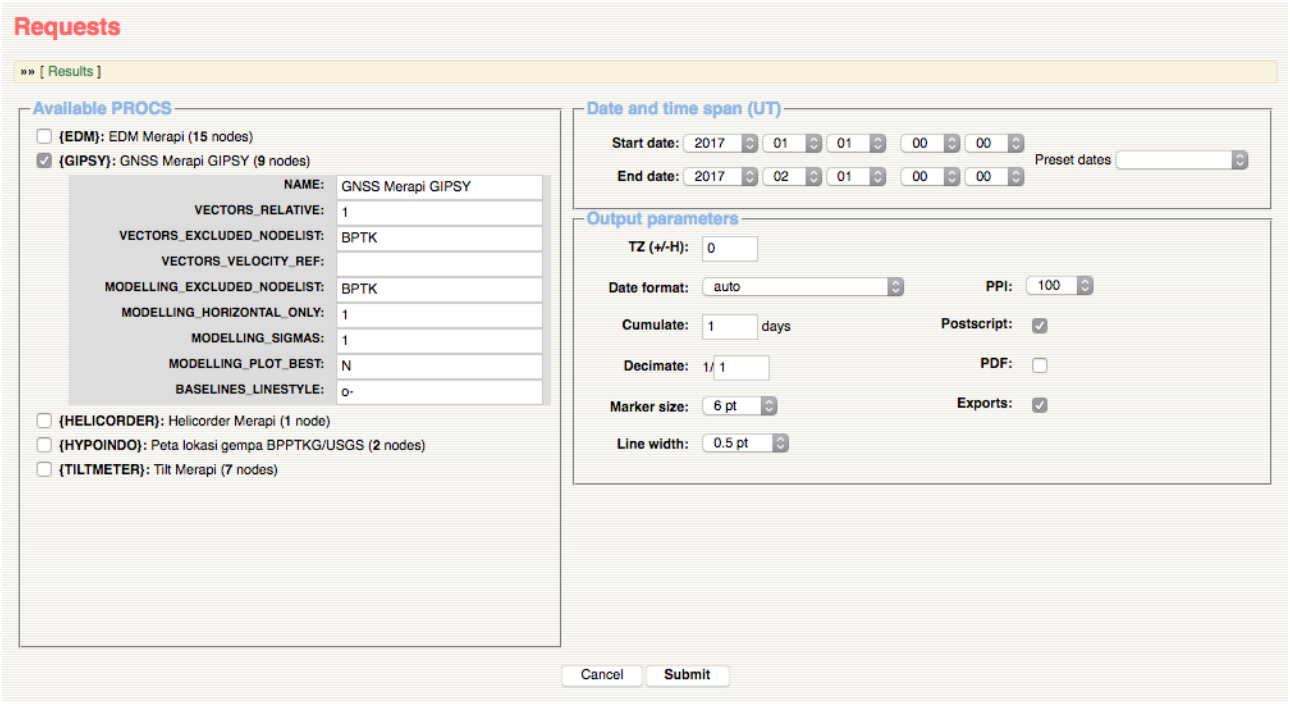
```
=key|value
# Generic PROC template for superproc waters
# This PROC must be associated with the FORM "EAUX" which contains compatible
# data and needed complementary configuration files.
# See $WEBOBS{ROOT_CODE}/matlab/superprocs/waters.m for further information.


# long name of the PROC
NAME|Waters chemical analysis

# proc TYPE and OWNER code (see OWNERS.conf for OWNCODEs)
TYPE|
OWNCODE|?
LOGO_FILE|$WEBOBS{ROOT_CODE}/icons/ipgp/logo_WOVS.png
TZ|0

# auto refresh of graphic display
DISPLAY_AUTOREFRESH_SECONDS|3600

# lists of parameters (number of comma-separated rows must be consistent)
TIMESCALELIST|10y,50y,all
DECIMATELIST|
CUMULATELIST|
DATESTRLIST|10,10,10
MARKERSIZELIST|4,2,2
LINEWIDTHLIST|1,.5,.1
STATUSLIST|1,0,0
# defines a reference date to use with 'r01' timescale in TIMESCALELIST
REF01_DATE|2000-01-01
# year and month timescales use true durations
TIMESCALE_TRUEVALUE|Y

SUMMARYLIST|

# execution command (for runproc shell)
SUBMIT_COMMAND|$WEBOBS{JOB_MCC} waters $SELFREF -
SUBMIT_RESOURCE|waters

# list of keys editable in requests
REQUEST_KEYLIST|NAME

# node parameters
NODE_NAME|spring
NODE_SIZE|15
NODE_RGB|0,1,1
NODE_FONTSIZE|10
NODE_MARKER|s

# file for background time referenced events
EVENTS_FILE|$WEBOBS{ROOT_CONF}/events_World.conf

# --- optional user-defined DEM, in ArcInfo format (overwrites SRTM/ETOPO default)
DEM_FILE|$WEBOBS{PATH_DATA_DEM}/exampledem.asc
# DEM coordinate system: UTM or LATLON
DEM_TYPE|LATLON
DEM_COPYRIGHT|

# additional zoomed map(s) using fixed axis limits: LON1,LON2,LAT1,LAT2
#MAP1_XYLIM|
#MAP2_XYLIM|
# do not show node alias in zoom map(s)
NODE_SUBMAP_ALIAS|N

# graphical density for PNG outputs (in pixel per inches)
PPI|100
# landscape format for all outputs
LANDSCAPE|N
# value = 1 to produce EPS graphs
POSTSCRIPT|1
# value = 1 to export text data files
EXPORTS|1
# main logo file
LOGO_FILE|$WEBOBS{ROOT_CODE}/icons/ipgp/logo_WOVS.png
# secondary logo file
LOGO2_FILE|
# main copyright
COPYRIGHT|MyProc
# secondary copyright
```

```
COPYRIGHT2|
```

3.13 PROCS graph and data request

3.13.1 Overview

A dedicated form **/cgi-bin/formREQ.pl** allows to make user request to get any outputs (graphs and data) from PROCS sharing the same set of time span and parameters, mostly independent from the SCHEDULER's jobs. The identified USER has to be authorized for reading on the PROC to perform a request on it.



3.13.2 Description

The form presents a list of available PROCS with empty check boxes. One or more PROCS can be selected, in that case a parameter list might appear for each of selected PROC, allowing the user to change the values. Please note there is no validity check of the values so a request may fail in case of invalid fields.

The main parameter to define is the date and time span: **Start date** and **End date** with time. Default is the last full month. A list of preset dates is also available. The date and time must be in UT.

Also some output parameters can be defined:

- **TZ:** the output time zone, in hours. Will affect graphs and data exports.
- **Date format:** the format of dates for time series plots axis ticks label.
- **Cumulate:** time for cumulating data (when cumulative allowed in the PROC), in day. Use fraction or any arithmetic formula if needed.
- **Decimate:** number of sample for decimation of the raw data (time series).
- **Marker size:** maker size in points (if the PROC uses markers).
- **Line width:** line width in points (if the PROC uses lines).
- **PPI:** resolution for PNG output images, in pixel per inch.
- **Postscript:** outputs the EPS vector graphic images (default is cheched).
- **PDF:** outputs a PDF version of images.
- **Exports:** outputs data (text files or others depending on the PROC).

After submit the request, each of the PROC will be submitted to SCHEDULER as specific jobs. The name of the job is made from date and time of the request, hostname and user login name.

The run of each PROC can be followed on the scheduler runs page. If a job ends with success, a notification email is sent to the user through the POSTBOARD. The email provides two links: a first to access the request results (graphs and exports) through a web interface similar to the routine PROC graphs. The second link allows to download a .tgz archive containing all images and files of the request.

A dedicated page is also available to access request results: **/cgi-bin/showREQ.pl**. The page will show the user's requests and results (if the request job has ended successfully), or all the existing requests for ADMIN users.

3.13.3 Configuration

PROC.conf (excerpt)

```
SUBMIT_COMMAND|$WEBOBS{JOB_MCC} genplot $SELFREF -
SUBMIT_RESOURCE|myproc
REQUEST_KEYLIST|NAME,SUMMARY_RELATIVE,PERNODE_LINESTYLE
```

The request form displays any PROC containing a not-empty SUBMIT_COMMAND parameter in its configuration. This parameter is the routine execution command line, ie. equivalent to the value of a XEQ1 in the SCHEDULER (see scheduler.pl doc) and, as such, supporting $WEBOBS parameters substitution.

The SUBMIT_RESOURCE is the optional routine execution mutex name (process lock) of as defined in SCHEDULER if the PROC is one of the routine jobs. This is to avoid possible conflicts of simultaneous runs.

Optionally, the REQUEST_KEYLIST parameter is used to specify a list of comma-separated keys of existing parameters, that will be presented to the user so that (s)he will have a chance to overwrite corresponding values for request execution.

The user defined output parameters **Date format**, **Cumulate**, **Decimate**, **Marker size** and **Line width** correspond to table values DATESTRLIST, CUMULATELIST, DECIMATELIST, MARKERSIZELIST and LINEWIDTHLIST, respectively.

The list of available preset values for PPI resolution, marker size and line width can be defined in **WEBOBS.rc** :

WEBOBS.rc (excerpt)

```
REQ_PPI_LIST|75,100,150,300,600
REQ_MARKERSIZE_LIST|1,2,4,6,10,15,20
REQ_LINEWIDTH_LIST|0.1,0.25,0.5,1,1.5,2,3
```

To use the notification email facility, POSTBOARD must be running, and the special event **formreq.** must be defined and valid (see **WebObs** Users Manager page).

3.14 Data formats available for PROCS

Formats are defined for a whole PROC in the RAWFORMAT PROC's variable, or for individual NODE in the RAWFORMAT NODE's parameter which overwrites the PROC value. The RAWDATA PROC's variable can be defined for all associated NODES, and any individual NODE's RAWDATA may overwrite it. A special variable **$FID** might be used and will be replaced by each NODE's value. For most of the formats, the Calibration File of each associated NODE will define the list of available channels and associated parameters.

See the source codes **CODE/matlab/readfmtdata.m** help for more details.

3.14.1 Waveforms formats

These formats are standards in seismology for waveforms data, but they are also used for other types of geophysical sensors. The standards use local files in specific format or dedicated protocol request from distant servers. Particularly, the full channel stream must be defined for each NODE, i.e.:

NET	network code	FDSN_NETWORK_CODE NODE's parameter
STA	station code	FID NODE's parameter
CHA	channel code	calibration file "Chan. Code" channel parameter
LOC	location code	calibration file "LC" channel parameter

{miniseed}: miniSEED files

Single of multiple local files in miniSEED format. RAWDATA defines the filename(s) using standard bash syntax (accepts wildcards). Some limitations may apply due to bash line size limit.

{seedlink}: SeedLink request

SeedLink protocol request from a distant or local server. RAWDATA defines the server with **host:port**. The format uses external program **slinktool** defined by the **WEBOBS.rc** SLINKTOOL_PRGM parameter. DATALINK_DELAY_SECONDS defines the delay in seconds from real-time for the last data.

{arclink}: SeisComP3 ArcLink request

ArcLink protocol request from a distant or local server. RAWDATA defines the server with **host:port**. Optional parameter ARCLINK_USER can be defined (default user is 'wo'). The format uses external program **arclink_fetch** defined by the **WEBOBS.rc** ARCLINKFETCH_PRGM parameter.

{combined}: SeisComP3 combined SeedLink/ArcLink request

The combined format will use SeedLink protocol for recent data and ArcLink protocol for data older than a delay. `RAWDATA` must contain the string **seedlinkhost:seedlinkport;arclinkhost:arclinkport;delayhours**. It will use both external programs defined in **WEBOBS.rc** `SLINKTOOL_PRGM` and `ARCLINKFETCH_PRGM` parameters.

{fdsnws-dataselect}: FDSN web-service dataselect request

Distant waveform request using the FDSN web-service protocol available at most of seismological data centers. `RAWDATA` must contain the base URL, for example: **http://service.iris.edu/fdsnws/dataselect/1/query?** for IRIS.

{winston}: EarthWorm Winston wave server request

EarthWorm Winston wave server (WWS) protocol request from a distant or local server. `RAWDATA` defines the server with **host:port**.

3.14.2 Generic time series

These formats will return time series of data channels, like the waveform formats do but usually the data sampling frequency is lower than for seismic waveforms, so it can be managed using data files stored in local directories. Number of channels depends on the data and can be selected and calibrated using the NODE's calibration file.

{ascii}: Generic ASCII text files

Attempt to read generic text files with regular data columns. `RAWDATA` contains the full path and filename(s) using bash wildcard facilities. The data files must be organized as regular columns of numbers (strings will produce a column of NaN), any separator character, and the date and time must be defined as 3 (year, month, day) or 6 columns (year, month, day, hour, minute, second) at some place (default is the 6 first columns). If there is no calibration file for a NODE, the header line will be used to get the channel names.

For this format you may define optional additional `FID_*` keys for each NODE to specify the format:

`FID_FS`	field separator character (default is semicolumn),
`FID_TIMECOLS`	index vector of columns defining date&time in order: year month day hour minute second,
`FID_NF`	number of data columns in the file, considering all non-numeric as separator (default is automatic),
`FID_HEADERLINES`	number of header lines (default is 1).

Generic ASCII format example 1: time + data channels 1 to 3.

```
              Date;                P_0;              SO2_0;              H2S_0;
06/06/2017 18:00:00;          752.579529;          -0.061299;          -0.031172;
06/06/2017 18:00:09;          724.445852;          -0.071515;          -0.024938;
```

`FID_*` parameters to read example 1 file.

```
FID_FS|;
FID_TIMECOLS|3,2,1,4,5,6
FID_NF|9
FID_HEADERLINES|1
```

Generic ASCII format example 2: data channel 1 (all NaN) + time + data channels 2 to 4.

```
VMAB    01/01/12        06:20:11        132     338     40.6
VMAB    01/01/12        06:40:11        135     337     40.7
VMAB    01/01/12        07:00:11        133     336     40.6
```

`FID_*` parameters to read example 2 file.

```
FID_FS|\t
FID_TIMECOLS|4,3,2,5,6,7
FID_NF|10
FID_HEADERLINES|0
```

{sql-table}: SQL-table request

Request to a SQL database using external program **mysql**. `RAWDATA` must contain the full command that will return the data in the text format **yyyy-mm-dd HH:MM:SS data1 data2 data3 ...**. The command must include two variables **$date1** and **$date2** that will be replaced by the timespan request. Example:

mysql -h host -u user -ppasswd -Ddatabase -N -B -e 'SELECT time,data1,data2,data3 from $FID WHERE time between "$date1" and "$date2";'

will make a request from the **database** at server **host** on the table **$FID** and return timestamp and data columns. The calibration file must define these 3 channels in that order.

{cr10xasc}: Campbell Scientific CR10X ASCII files

Daily data files from data loggers CR10X archived in a specific directory structure. `RAWDATA` contains the main directory path, in which files are stored in the following subpath and name: **FID/YYYY/YYYYMMDD.DAT**. Each file has the data format: **PRGM,yyyy,doy,HM,data1,data2, ... ,dataN**, where **PRGM** is the program number, **yyyy** the 4-digit year, **doy** the day of the year (ordinal day), **HM** the hour and minute with leading blanks, and the data.

Campbell CR10X format example

```
121,2013,365,2340,8.6971,-20.168,0,91.3,22.48,0,826.42,12.514,99999,0,0
121,2013,365,2350,8.7294,-20.016,0,88.4,20.02,0,826.43,12.521,99999,0,0
```

{t0a5}: Campbell Scientific T0A5 ASCII files

Daily data files from Campbell Scientific data loggers in the T0A5 output format, archived in a specific directory structure. `RAWDATA` contains the main directory path, in which files are stored in the following subpath and name: **FID/YYYY/FIDYYYYDDD.DAT**. Each file has the data format: **"yyyy-mm-dd HH:MM:SS",data1,data2, ... ,dataN**.

Campbell T0A5 format example

```
"2014-01-11 00:00:00",68148,12.1,16.07,15.29,100,938,0.02,0.02,0,7.875425,98.67353,10.86884,0,0,0,0,0,0,0
"2014-01-11 00:10:00",68149,12.1,15.96,14.76,100,938,0.021,0.021,0,7.200668,101.9742,11.35311,0,0,0,0,0,0,0,0
```

{porkyasc}: USGS Porky ASCII files

Daily data files from USGS Porky data systems, archived in a specific directory structure. `RAWDATA` contains the main directory path, in which files are stored in the following subpath and name: **FID/YYYY/YYYYMMDD.DAT**. Each file has the data format: **DD-MMM-YYYY HH:MM data1 data2 ... dataN**.

USGS Porky format example

```
01-JAN-2014 00:00  00000  00029  00013  03181  00001 -00998 -00998 -00998
01-JAN-2014 00:05  00001  00040  00010  03182  00001 -00998 -00998 -00998
```

3.14.3 Quakes catalogs

These are specific formats for PROCS dedicated to earthquake catalogs. These formats returns a list of event with preset channels, like Latitude, Longitude, Depth, Magnitude, etc... There is no calibration file (if it exists it will be ignored). It is possible to link this format with a Main Courante (MC3) database using the NODE's `FID_MC3` with the MC3 name. In that case some information from MC3 might be associated to identified events.

An other specificity of these formats is that all catalogs from different associated NODES will be concatenated in a single data matrix. This allows to merge for instance, a distance worldwide catalog like USGS (for large earthquakes), a local catalog from a local network, and an historical catalog in an old-fashion file format.

{hyp71sum2k}: Quake Hypo71 summary lines year 2000 compatible

Single file in the HYPO71 ASCII format identified by `RAWDATA` with full filename and path. The standard format is completed by two last columns: **SCode** for a 5-letter identification code, and **File** for the waveform filename. Note the file is column formatted, without any delimiter, there is no leading zeros but blanks, and the longitude value is positive towards the West.

HYPO71 format example

```
#    DATE ORIGIN      LAT_N      LON_W      DEPTH     MAG NO GAP DMIN  RMS  ERH  ERZ Q  SCode File
18430208 1440 00.00 16-44.00  61-10.00 000.00    8.00 00 000 00.0 0.00 00.0 00.0 0  TE9GM
20141005 1819 07.34 14-48.70  61-10.33  -0.24 D 1.52  8 166  0.3 0.26  0.6  0.8 C  EB1    20141005_181900.mq0
```

{fdsnws-event}: Quake FDSN WebServices event request

Distant event data request using the FDSN web-service protocol available at most of seismological data centers. It accepts the QuakeML 1.2 format only. `RAWDATA` must contain the base URL, for example: **http://service.iris.edu/fdsnws/event/1/query?** for IRIS.

{scevtlog-xml}: Quake SeisComP3-xml files

Reads a files architecture created by the SeisComP3 scevtlog module. `RAWDATA` defines the path root where events are stored in a subdirectory structure as **YYYY/MM/DD/eventID/eventID.last.xml** in the SC3ML format.

3.14.4 GNSS solutions

These are specific formats for PROCS dedicated to positioning data from GNSS (Global Network Satellite Systems) like GPS or GLONASS. These formats returns preset channels: Eastern, Northern, Vertical and Orbit type.

{gipsy}: JPL GIPSY-OASIS TDP files

TDP (Time Dependant Parameter) files results of the JPL GIPSY-OASIS processing in IRTF. The format uses only the position part of the data: **Time Dinit Dfinal error STA c ssss** where Time is GPS date in seconds past J2000, **c** is the component in geocentric referential, **ssss** the station name. `RAWDATA` contains the path root directory where daily solutions files are stores in a subdirectory structure as **FID/YYYY/FID/YYYY-MM-DD.FID.tdp***.

GIPSY-OASIS TDP format example

```
 476712000.0000   1797.11460400000      1797.11463384421      9.751E-07  STA Z   ABD0
 476712000.0000  -5375.91966100000     -5375.91974126823      2.297E-06  STA Y   ABD0
 476712000.0000   2920.34971800000      2920.34975977472      1.377E-06  STA X   ABD0
```

Non-final orbits are coded using additional file extension ***.tdp.qlR** for rapid orbits, ***.tdp.ultra** for ultra orbits.

{gipsyx}: JPL GipsyX TDP files

TDP (Time Dependant Parameter) files results of the JPL GipsyX processing in IRTF. The format uses only the position part of the data: **Time Dinit Dfinal error ssss.State.Pos.c** where Time is GPS date in seconds past J2000, **c** is the component in geocentric referential, **ssss** the station name. `RAWDATA` contains the path root directory where daily solutions files are stores in a subdirectory structure as **FID/YYYY/FID/YYYY-MM-DD.FID.tdp***.

GipsyX TDP format example

```
 613482900 -1.272247348200000e+06 -1.272245892239674e+06  1.052583883237580e-03 .Station.GLOR.State.Pos.Z
 613482900  4.591963170400000e+06  4.591962844942605e+06  3.727601983103848e-03 .Station.GLOR.State.Pos.Y
 613482900  4.238487354600000e+06  4.238485753608068e+06  3.355552475393279e-03 .Station.GLOR.State.Pos.X
```

Non-final orbits are coded using additional file extension ***.tdp.qlR** for rapid orbits, ***.tdp.ultra** for ultra orbits. This extension comes from bash script **gnss_run_gipsyx**.

{globkval}: MIT GAMIT-GLOBK VAL files

Single file output of Gamit-GlobK processing in ITRF referencing. `RAWDATA` defines the full path filename of the .VAL result file which contains solution timeseries for each component and each station.

GAMIT-GLOBK VAL format example

```
 Combination of ALL networks
ILAM_GPS to E Solution  1 +  32197594.810 m

 2012 12 19 11 59  32197594.80990    0.00626   -0.00183   0.00626
 2012 12 20 11 59  32197594.81087    0.00433   -0.00090   0.00433
 ...
 2013 12 17 11 59  32197594.81866    0.00519   -0.00799   0.00519

Wmean   32197594.8184 m +- 0.0003 from  340 data. WRMS   5.2 mm, NRMS  1.19
```

```
Slope     15.01 +-      0.84 mm/yr, WRMS    3.1 mm, NRMS   0.70, dur   0.99 <> 2013.41 yr

 Combination of ALL networks
ILAM_GPS to U Solution   1 +         -0.750 m

 2012 12 19 11 59         -0.74977     0.01910     0.01410    0.01910
 2012 12 20 11 59         -0.74919     0.01048     0.01471    0.01048
 ...
 2013 12 17 11 59         -0.80060     0.01370    -0.02614    0.01370

Wmean          -0.7686 m +- 0.0006 from   340 data. WRMS    9.3 mm, NRMS   0.82
Slope    -10.66 +-      2.20 mm/yr, WRMS    8.8 mm, NRMS   0.78, dur   0.99 <> 2013.41 yr
```

3.14.5 Other specific formats

These formats are basically time series but the channels are predefined.

{teqc-qc}: TEQC Rinex quality check

{naqs-soh}: NAQS State of Health

{wodbform}: WebObs database forms

This format is not selectable. It becomes active automatically when a PROC is associated to a FORM and its specific database. In that case the data columns are determined by the FORM type.

Chapter 4

Administration

4.1 Users, Groups and Authorizations

4.1.1 Overview

WebObs uses its own AUTHORIZATION system, in addition to the Apache Authentication system, to identify its HTTP USERS and control their individual ACCESS-RIGHTS to **WebObs** RESOURCES (ie. logical entities refering to files, processes, html-pages, whatever).

AUTHORIZATION system elements:

- a USERS TABLE that further identifies the USERS defined in the Apache Authentication files (eg. .htpasswd),
- a GROUPS TABLE that merely defines groups of USERS, to simplify (reduce number of) access-rights definitions,
- RESOURCES are fully identified as **resourceType.resourceName**,
- RESOURCES TABLES are the **resourceType** tables containing their own **resourceName** descriptions,
- a **resourceName** description defines the relationship **uid-or-gid has access-rights**,
- Supported access-rights are:
  - **R** = 1 Read
  - **E** = 2 Edit = Read + Write
  - **A** = 4 Admin = Edit + Create/Delete

4.1.2 Users table and Groups table

A **WebObs** USER is identified by its LOGIN (string) as also defined in the HTTP Authentication system. A row in the USERS TABLE further defines a USER with the following information:

- LOGIN
- FULLNAME, the user's name
- UID, a short identification string, usually the user's name initials, to be used for access-rights and other functionnal needs
- EMAIL, the email address (somebody@somewhere) used by the **WebObs** POSTBOARD system,
- VALIDITY, to determine if the user is able to access some resource or not.

Two special UIDs are reserved for system use:

- **?** to identify a GUEST user (granted to undefined users for temporary/limited access to **WebObs**),
- **!** to identify the WEBOBS OWNER.

Records of the GROUPS TABLE associate UIDs to GROUP names (aka GID). A GID must starts with a '+' sign. A USER may be a member of more than one GROUP. A USER inherits all access-rights defined for the GROUP(s) it belongs to.

Four (4) special GROUPs are pre-defined in **WebObs** : **+ADMIN**, **+DUTY**, **+OBSERVER** and **+VISITOR**. They are initially used to define access-rights to **WebObs** built-in tools and/or applications.

4.1.3 Resource tables

The fully qualified name of a **WebObs** RESOURCE is **resourceType.resourceName**.

There are five (5) resourceType tables: **authviews**, **authprocs**, **authforms**, **authwikis** corresponding to the base **WebObs** objects and **authmisc** for any additional, unclassified, resourceNames definitions. They are already populated with resourceNames related to **WebObs** built-in tools and applications.

The special resourceName '*' stands for "all resourceNames of this resourceType".

resourceNames are strings, defined and documented by the developers of the **WebObs** tools or applications.

4.1.4 Managing Users and Authorizations

The USERS ADMIN page **/cgi-bin/usersMgr.pl** (built-in tool), initially restricted to the +ADMIN group, is used to create/-modify/delete user and resources definitions.

USERS and RESOURCE TABLES have customization variables in the main configuration file **WEBOBS.rc**:

WEBOBS.rc (excerpt)

```
SQL_DB_USERS|${ROOT_CONF}/WEBOBSUSERS.db
SQL_TABLE_USERS|users
SQL_TABLE_AUTHPROCS|authprocs
SQL_TABLE_AUTHVIEWS|authviews
SQL_TABLE_AUTHFORMS|authforms
SQL_TABLE_AUTHWIKIS|authwikis
SQL_TABLE_AUTHMISC|authmisc
SQL_TABLE_GROUPS|groups
SQL_DB_USERS_AUTOREGISTER|YES
```

4.1.5 Developing with Users and Authorizations system

The **WebObs::Users** perl module is the built-in interface to the USERS/AUTHORIZATIONS objects and functions system. Detailed programming information can be found in its 'perldoc' documentation, such as:

- global variables USERS, USERIDS and CLIENT
- the special 'path-like' specification for resourceNames
- functions: **allUsers, clientHasRead, clientHasEdit, clientHasAdm, listRNames**

Developers may add/define/use their own resourceName(s) for their specific needs.

4.1.6 Adding a new user

Registration of a new user is done in 3 steps:

1. new user must fill the form (see screenshot 4.1) by connecting to WebObs interface and click 'Cancel' when asked to login. The data (with encrypted password) will be stored as a new pipe-separated line in the file **DATA/DB/reglog** and an e-mail will be sent to user and **WebObs** owner;
2. two alternatives depending on the value of `SQL_DB_USERS_AUTOREGISTER` key:
   (a) if 'YES', then the new user will be automatically added in the database with validity 'N' and without any associated group. UID is made from initials of the full name, if necessary adding suffix number. The encrypted password will be automatically added into **CONF/htpasswd** apache file;
   (b) if 'NO', administrator must add the user manually using Users Admin interface (see section 4.1.2), and add manually the encrypted password as a new line into the file **CONF/htpasswd**;
3. for both alternatives, administrator must validate the new user, eventually modify its UID, and associate it to a group or add specific resource access authorization. This step must be conducted with care since it gives an access (or edit/admin) to all or part of **WebObs** resources.

4.2 PostBoard

4.2.1 Overview

WebObs tools and applications may wish to send (email) alerting/warning/information messages to **WebObs** USERs when detecting special processing conditions or other events. Deciding who needs/wishes to receive such messages, and actually sending them, should be as easy as possible from the developers point of view; furthermore, **WebObs** administrators should be

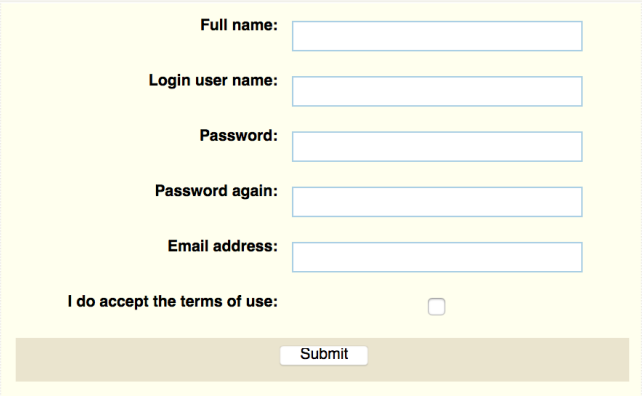


Figure 4.1: Registration form for new users.

able to easily filter/choose which users should receive what, based on operationnal needs, authorizations concerns, and even user's choice of being (or not being) alerted.

The **WebObs** POSTBOARD system (notifications/subscriptions) addresses these needs. Elements of POSTBOARD architecture:

- Tools and applications simply and unconditionnaly send identified messages (NOTIFICATIONS) to POSTBOARD
- NOTIFICATIONS basically look like "eventname|senderId|message"
- POSTBOARD is a daemon that tries to match the eventname of the NOTIFICATIONS it receives against active SUBSCRIPTIONS that tell it what to do: either send a mail to a UID (or GID), or trigger a command, or both
- a SUBSCRIPTION is a row in the **WebObs** NOTIFICATIONS TABLE with the following fields:

eventname	identifying the SUBSCRIPTION to match NOTIFICATIONS sent to POSTBOARD
validity	indicating active/inactive
uid	UID or GID to whom mail the NOTIFICATION
subject	subject of the mail being sent
attachment	optional, path of a file to attach to mail
action	optional, a command to be executed

4.2.2 Event names

Event names identify and associate NOTIFICATIONS to SUBSCRIPTIONS:

- eventname = string[.[string]]
- string.string is known as the **majorname.minorname** form of an event-name
- **majorname.minorname** identifies a single subscription named majorname.minorname AND
- a **majorname.** subscription, if defined as such (don't forget the ending dot!), will also match all **majorname.minorname** notifications This is the way to define common mail/action to a set of notifications.
- some eventnames are already defined for internal **WebObs** usage. These reserved eventnames are : **eventnode , formreq. , scheduler.alert , scheduler.warning , submitrc.**

Example: a **WebObs** application may issue (notify) NOTIFICATIONs identified with **myevent** eventname; The SUBSCRIPTION **myevent,Y,UID,mysubject,-,-** is registered in the NOTIFICATION TABLE; the following mail will eventually be sent by POSTBOARD when the application notifies "myevent||the application message" :

mail for myevent notification

```
From: webobs@webobsaddr
To: UID's mailaddr
Subject: [WEBOBS_ID] mysubject
User-Agent: Mutt/1.x.xx (2000-01-01)
the application message
```

4.2.3 Managing PostBoard Subscriptions

The USERS ADMIN page **/cgi-bin/usersMgr.pl** (built-in tool), initially restricted to the +ADMIN group, is used to create/-modify/delete the POSTBOARD SUBSCRIPTIONs in the NOTIFICATIONS TABLE.

		Event	V	Uid	Mail Subject	Mail Attachm.	Action
✎	⊘	eventnode	Y	!	webobs node event	-	-
✎	⊘	formreq.	Y	!	your graph request is ready	-	-
✎	⊘	scheduler.alert	Y	!	-	pp/ff.txt	-
✎	⊘	scheduler.warning	Y	!	scheduler warning	-	-
✎	⊘	submitrc.	Y	!	-	-	-

CODE/cgi-bin/postboard.pl is the Perl daemon.
CODE/shells/postboard is the command line interface to start/stop, query status and even send NOTIFICATIONs to POSTBOARD. Usage: **postboard [start|stop|status|kill|notify]**

POSTBOARD also has customization variables in the main configuration file **WEBOBS.rc**:

WEBOBS.rc (excerpt)

```
SQL_DB_POSTBOARD|${SQL_DB_USERS}
SQL_TABLE_NOTIFICATIONS|notifications
POSTBOARD_NPIPE|/tmp/WEBOBSNP
POSTBOARD_MAILER|mutt
POSTBOARD_MAILER_OPTS|-nx
POSTBOARD_MAILER_DEFSUBJECT|WebObs notification
```

4.2.4 Developing with Notifications

The **WebObs::Config** perl module exports the **notify** function to be used to send NOTIFICATIONS to POSTBOARD. The **notify.m** module plays the same role from MatLab code.

Detailed programming information can be found in **CODE/cgi-bin/postboard.pl** perldoc documentation, such as:

- the **WebObs::Config::notify** function syntax,
- eventnames naming conventions,
- notification string syntax: **event-name|sender-id|message** and automatic timestamp ,
- **message** component interpretation and special keywords,
- the special **submitrc.** eventname,
- POSTBOARD MAILER considerations,
- subscription's ACTIONs considerations

4.3 Scheduler

4.3.1 Overview

The SCHEDULER is the daemon that controls the execution of **WebObs** batch JOBS. It has been developed to meet the following needs that, for some of them, would have been more difficult to tackle, or simply have required as much development, with a regular crontab architecture:

- schedule execution of a jobs based on the elapsed time since their previous execution,
- manage parallel executions of jobs,
- implement a mutually exclusive locking mechanism between jobs execution,
- implement a simple checking of CPU load to accept or delay jobs execution,
- centralize and normalize the jobs definitions, also with run-time parameters substitutions,
- standard output and error archiving and consultation,
- centralize reporting/history with housekeeping and HTML interface,
- accept dynamic jobs submission in addition to regular jobs,
- use **WebObs** POSTBOARD for errors/warnings and end-of-jobs notifications,
- provide both command line and HTML interfaces to JOBS and EXECUTIONS management

The SCHEDULER daemon is **CODE/cgi-bin/scheduler.pl** whose execution is controlled with the command line interface **CODE/shells/scheduler**.

4.3.2 Configuration and Tables

The main configuration file **WEBOBS.rc** holds the **CONF_SCHEDULER** variable that points to the SCHEDULER's configuration file used to customize its execution environment. Default configuration file is **scheduler.rc** :

scheduler.rc

```
BEAT|2                                      # internal processing loop frequency in seconds
MAX_CHILDREN|10                             # maximum number of simultaneously started jobs
LISTEN_ADDR|localhost                       # hostname or local IP address of the command interperter
PORT|7761                                   # commands' interpreter UDP port number
SOCKET_MAXLEN|1500                          # commands' interpreter max command size
SQL_DB_JOBS|$WEBOBS{ROOT_CONF}/WEBOBSJOBS.db # sqlite database for JOBS and RUNS tables
DAYS_IN_RUN|30                              # number of days JOBS are kept in RUNS table
DITTO_LOG_MAX|500                           # controls repeating messages in log
DITTO_NTF_MAX|1000                          # controls repeating messages notified
CANCEL_SUBMIT|600                           # max seconds submitted JOBS wait in Queue
CLEANUP_RUNS|999,zombie                     # how to tag zombie RUNS (unknown end of job)
LOADAVG1_THRESHOLD|0.7                      # max 1-sec cpu load threshold to start JOBS
LOADAVG5_THRESHOLD|0.7                      # max 5-sec cpu load "
LOADAVG15_THRESHOLD|0.7                     # max 15-sec cpu load "
LMISS_BIAS|10                               # seconds to delay candidates on load-threshold
EMISS_BIAS|4                                # seconds to delay candidates on enq busy
PATH_STD|$WEBOBS{ROOT_LOGS}/jobslogs        # root path for all STDOUT and STDERR of JOBS
PATH_RES|$WEBOBS{ROOT_LOGS}/res             # directory to hold JOBS resources (locks)
```

JOBS TABLE

JOBS to be scheduled are uniquely identified with a JID and defined into the JOBS TABLE by the following fields:

JID	JOB's ID unique string
VALIDITY flag	indicates wether JID is eligible for execution
RESOURCE name	a string identifying a mutually exclusive jobs lock
XEQ1, XEQ2 and XEQ3	3 components of the actual command that is started for JOB's execution
INTERVAL	required elapsed time (seconds) between two executions of the JOB
LOAD THRESHOLD	max CPU LOAD value to allow execution of the JOB
LOGS PATH	path of the JOB's STDOUT and STDERR files
LAST START TIMESTAMP	(not editable)

RESOURCE syntax

A RESOURCE is simply identified by its freely choosen name (a string not containing double-dash, ie –). It may also be a set of individual resources (a '+' separated list of names): all of these individual resources must be simultaneously free for the job to be executed.

XEQ1, XEQ2, XEQ3 syntax

XEQ1, XEQ2 and XEQ3 will be concatenated, in this order, to build the actual JOB command to be executed. Those fields have no special meanings for execution, except XEQ2 for LOGPATH (see below), and only one is obviously required to build a valid command; but they may ease maintenance and lisibility.

They all accept variables interpolation: ie. they may specify any number of variable names from the main configuration file **WEBOBS.rc**, coded as **$WEBOBS**{**variableName**}.

JOBS LOGS PATHS syntax

JOBs can redirect/build their own STDOUT and STDERR, however the following rules are implemented as a default behavior in the SCHEDULER:

- All JOBS outputs as a whole will be placed into the common **$SCHED**{**PATH_STD**} directory,
- JOBs' specific LOGPATH definitions are relative to this common directory.
- By default, the standard output and error output of the job process will be written to diffent files, respectively suffixed `.stdout` and `.stderr`.
- If the scheduler configuration variable **MERGE_JOB_LOGS** (that is, **$SCHED**{**MERGE_JOB_LOGS**})is true (meaning its value is either `y`, `yes`, in lower of upper case, or `1`), the standard error output will be merged in the same file as the standard output, and the ouput file will have the suffix `.log`.

The following table shows **LOGPATH** syntax (left) and its full interpretation (right). Any subdirectories will be dynamically created if needed.

logpath value	Path to the logfile(s)
name	$SCHED{PATH_STD}/name.*ext*
name/	$SCHED{PATH_STD}/name/*pid.ext*
name/name/out	$SCHED{PATH_STD}/name/name/name/out.*ext*
(null)	$SCHED{PATH_STD}/*pid.ext*

In the table above, *pid* is the job process PID and the file extension *ext* will be :

- `stdout` for the standard output if **$SCHED{MERGE_JOB_LOGS}** is false or missing;
- `stderr` for the standard error output if **$SCHED{MERGE_JOB_LOGS}** is false or missing;
- `log` if **$SCHED{MERGE_JOB_LOGS}** is true;

The following two redirection rules apply to any of the above forms of the *logpath* value:

>**name**	(the default if '>' is omitted) overwrites previous file with same name
>>**name**	appends to previous file with same name

The following TAGS are also available in the name(s) you supply for easier specification of unique log files:

{**TS**}	replaced with job's start-timestamp
{**RTNE**}	replaced with job's XEQ2 string, with any blanks (spaces) chars changed to _ underscores.

For example, with the default configuration, a *logpath* value of `sefran3/sefran3_{TS}` will generate log files with names similar to `sefran3_946699200.54321.stdout` and `sefran3_946699200.54321.stderr` (or `sefran3_946699200.54321.log` only, if **$SCHED{MERGE_JOB_LOGS}** is true) in the directory `/opt/webobs/LOGS/jobslogs/sefran3/`.

4.3.3 Jobs selection and execution

The SCHEDULER continously scan the JOBS TABLE to find CANDIDATES for execution: ie those VALID JOBS whose LAST RUN TIMESTAMP is now older than their INTERVAL.

JOBS may also be submitted for immediate execution from the command line, or from another application, regardless of their INTERVAL normal delay and of their VALIDITY flag. Submitted JOBS are automatically CANDIDATES and placed in a JOB REQUEST QUEUE (JOBRQ) where they can stay no more than $SCHED{CANCEL_SUBMIT} seconds.

The SCHEDULER then scans all CANDIDATES, starting with the JOBRQ, to actually start (execute) JOBS that fullfill their CPU LOAD THRESHOLD and RESOURCE (lock) conditions. JOB execution's command is the concatenation of the JID's XEQ1, XEQ2 and XEQ3 strings, in this order, and with ${WEBOBS} variables interpolation. Started JOBS are placed into a RUNQ for monitoring and future end-of-job processing, and in the RUNS TABLE for reporting/history.

CANDIDATES that are not moved to the RUNQ because they don't fullfill their CPU LOAD THRESHOLD and RESOURCE (lock) conditions, will automatically be candidates again on the next scheduler's beat; to avoid unnecessary overload and reporting, the scheduler may delay these jobs from being candidates again by a small amount of seconds. Delay to be used are defined by $SCHED{LMISS_BIAS} for CPU LOAD THRESHOLD condition and $SCHED{LMISS_BIAS} for RESOURCE busy condition. Set these to 0 to disable delay.

JOBS are started as independent, parallel processes, children of the SCHEDULER, in their own process group; from there on they are known as KIDS. The SCHEDULER doesn't forget its KIDS once it forked them ! It waits for their termination (non-blocking wait) to perform housekeeping and reporting about execution (mainly unlocking RESOURCEs, saving return code and elapsed time to update the RUNS TABLE).

The precision at which JOBS are scheduled/executed is $SCHED{BEAT} seconds.

Scheduler's job RESOURCEs, used as a locking mechanism between scheduled jobs, may be shared with external processes: thus it is possible to also synchronize execution of scheduler's jobs with non-scheduler machine's activities and/or conditions. The Scheduler's commands ENQ and DEQ are the unique scheduler's entry points to the locking mechanism.

Command line submit syntax

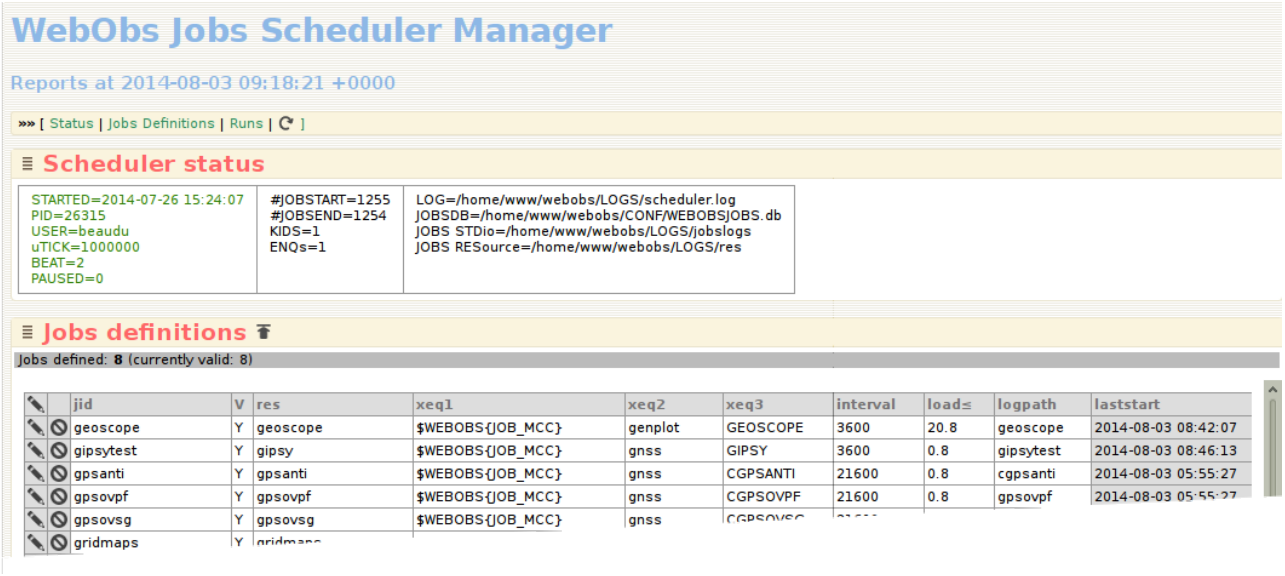
- JOBS may be submitted, for immediate execution, to the SCHEDULER using **CODE/shells/scheduler** and its **submit** command
- specifying the JOB comes in two flavors:
  - JID=<job's id from JOBS TABLE >; Example: **scheduler submit JID=myjob**
  - as a string defining the JOB, with the following comma-separated keywords:
    - **XEQ1: , XEQ2: and XEQ3:** to specify the JOB's command
    - **LOGPATH:** optional, to specify the directory, relative to $SCHED{PATH_STD}, for JOB's STDOUT and STDERR
    - **RES:** optional, JOB's RESOURCE (lock)

    - **MAXSYSLOAD:** optional, CPU LOAD THRESHOLD
    - **UID:** optional, UID to be used for end-of-job notification
  - Example: **scheduler submit 'XEQ1:perl,XEQ2:/path/to/jobtst.pl,RES:mylock,UID:DL'**

Note: submitted JOBs are given a unique, negative, JID.

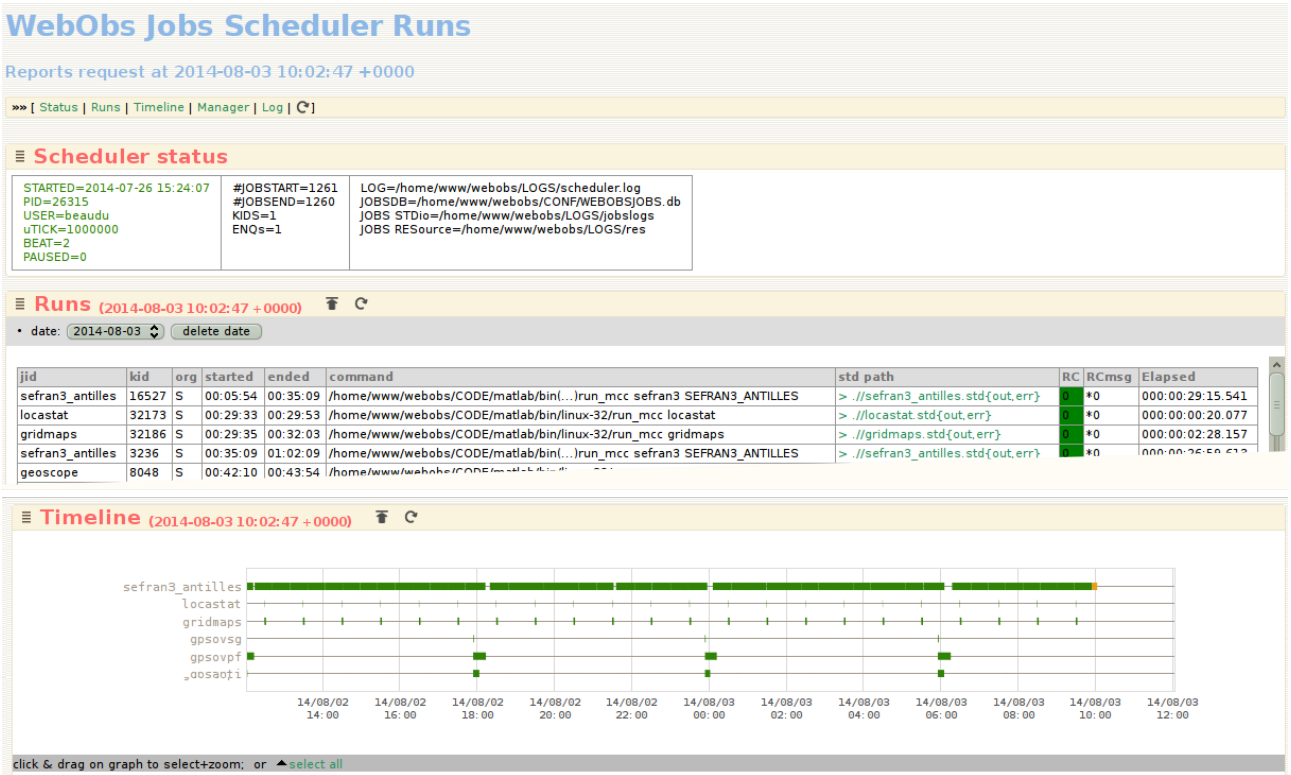
4.3.4 Scheduler manager

The SCHEDULER MANAGER **CODE/cgi-bin/schedulerMgr.pl** built-in page, initially restricted to the +ADMIN group, is used to create/modify/delete JOBS of the JOBS TABLE.



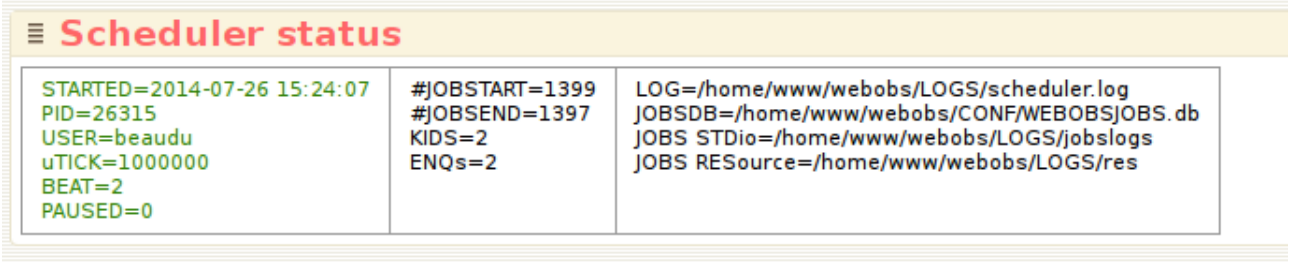
4.3.5 Scheduler runs

The SCHEDULER RUNS **CODE/cgi-bin/schedulerRuns.pl** built-in page, initially restricted to the +ADMIN group, is used to display the RUNS TABLE (one day at a time) along with its corresponding, zoomable TIMELINE chart, to better visualize JOBs executions elapsed times and parallelism.



4.3.6 Scheduler status

Both the SCHEDULER MANAGER and SCHEDULER RUNS built-in pages show the STATUS of the SCHEDULER:



where:

STARTED, PID, USER	**when**, under **which** PID and **who** started the SCHEDULER
uTICK, BEAT	internal frequency in microseconds and main SCHEDULER loop frequency
PAUSED	wether SCHEDULER is in PAUSE mode (not scanning JOBs, = 1)
#JOBSTART, #JOBSEND	number of **started** and **ended** JOBs since 'STARTED'
KIDS	number of currently started JOBs (KIDS)
ENQs	number of currently held JOBs' RESOURCEs
Paths of main SCHEDULER's files	

4.3.7 Scheduler command line

The SCHEDULER command line interface is **CODE/shells/scheduler**.

shells/scheduler

```
me@here:/opt/webobs/CODE/shells$ ./scheduler
Usage: ./scheduler {enq|deq|flog|jobq|terminate|kill|pause|ps|qs|
                    quiet|resume|runq|start|status|stop|submit|killjob|verbose}
```

CODE/shells/scheduler sub-commands :

start	start, if not already active
stop	stop, waiting for any active kids to end
terminate	stop, killing any running jobs
pause	hold execution (suspend jobs processing)
resume	resume execution
enq	enq a resource
deq	deq a resource
kill	forced stop, not recommended
verbose	log verbosity on
quiet	log verbosity off
status	display status/indicators/settings
jobq	display JOBRQ
runq	display RUNQ
ps	display executing KIDS trees
flog	force log backup and cleanup
submit	place a job on the JOBRQ, requiring additional arguments, either: 1) **jid=n** where n is the JOB's ID from the JOBS TABLE 2) **keyword:value[,keyword:value,...]** where keyword in [XEQ1,XEQ2,XEQ3,MAXSYSLOAD,LOGPATH,RES,UID]

example scheduler status

```
me@here:/opt/webobs/CODE/shells$ ./scheduler status
Pls wait...
STATTIME=2014-02-05 06:04:58
STARTED=2014-02-05 06:04:02
PID=6725
USER=root
uTICK=1000000
```

```
BEAT=2
LOG=/opt/webobs/LOGS/scheduler.log
JOBSDB=/opt/webobs/CONF/WEBOBSJOBS.db
JOBS STDio=/opt/webobs/LOGS/jobslogs
JOBS RESource=/opt/webobs/LOGS/res
PAUSED=0
#JOBSTART=2
#JOBSEND=2
KIDS=0
ENQs=0
```

example runq and ps

```
me@here:/opt/webobs/CODE/shells$ ./scheduler runq
Pls wait...
RUNQ(1407212134.21663) started on 2014-08-05 00:15:34
   ORG=R
   jid=-2
   kid=7567
   kidcmd=/opt/webobs/CODE/cgi-bin/jobtester.pl 40
   logfd=./
   logfn=undef
   res=nil
   started=1407212135.2194
   uid=nil
RUNQ(1407211890.24512) started on 2014-08-05 00:11:30
   ORG=S
   jid=sefran
   kid=7060
   kidcmd=/opt/webobs/CODE/matlab/bin/linux-64/run_mcc sefran3
   logfd=./
   logfn=sefran3
   res=sefran3
   started=1407211890.46864
   uid=nil

me@here:/opt/webobs/CODE/shells$ ./scheduler ps
  PID  PGRP     ELAPSED %CPU %MEM CMD
 2769  2769    02:41:40  0.0  0.1 /bin/bash
 7050  7034       04:21  0.1  0.2 perl scheduler.pl -v
 7060  7060       04:19  0.0  0.0   /bin/sh /opt/webobs/CODE/matlab/bin/linux-64/run_mcc sefran3
 7068  7060       04:18  6.7  3.3     /opt/webobs/CODE/matlab/bin/linux-64/sefran3
 7567  7567       00:14  0.0  0.0   /usr/bin/perl /opt/webobs/CODE/cgi-bin/jobtester.pl 40
```

Note that when using *systemd* to manage the **scheduler.pl** background process, the command **systemctl stop woscheduler.service** (which needs to be run as *root*) will kill any running jobs, as it acts just like the **scheduler terminate** command. This will usually not do much harm, but using **scheduler stop** (as your webobs user) instead is recommended if you need to stop the scheduler for administrative purposes like on updates (*systemd* will not restart the *woscheduler* service in this case, as the main process will exit on a success code).

4.4 Postboard and Scheduler system integration

In order to automatically start and stop the Postboard and the Scheduler when the system starts up and shuts down, you need to create some configuration in your system. You only need to do this when you first install WebObs.

The configuration to create depends on the system and service manager your system uses. Major GNU/Linux distributions nowadays use the *systemd* manager, so the rest of this section assumes your system use this manager. Should your system use a different one, you will have to refer to its documentation, and potentially write an init script that will start and stop the postboard and the scheduler based on the `postboard` and `scheduler` scripts in the `CODE/shells` directory.

Configuring services in systemd

To start, stop and manage the Postboard and the Scheduler using systemd, you will need to create a *service* for each of these processes.

The WebObs package provides templates for these services in the `SETUP/systemd` directory, named `wopostboard.service` and `woscheduler.service`. They call the scripts `postboard` and `scheduler` from the `CODE/shells` directory with the *start* subcommand to start the perl scripts `postboard.pl` and `scheduler.pl`. When they are stopped, the services send a TERM signal to the perl scripts that will then exit cleanly.

To use these services, you will first have to do the following actions as the user *root*,:

- copy these files to the directory `/etc/systemd/system`

```
cp SETUP/systemd/wo*.service /etc/systemd/system/
```

- edit these copied files (for example using the command `nano /etc/systemd/system/wopostboard.service`) to change the lines `User=` and `Group=` to use the name of your WebObs system user and its group (this usually will be something like `User=webobs` and `Group=webobs`, although some installation use the `wo` user and group).

Once these service files are installed and adapted, use these commands as *root* to reload the systemd configuration and enable the services:

```
systemctl daemon-reload
systemctl enable wopostboard
systemctl enable woscheduler
```

If you want to immediately start the services, use the following commands:

```
systemctl start wopostboard
systemctl start woscheduler
```

The first command is not mandatory here, as the *woscheduler* service is configured to start the *wopostboard* service before, if it is not already started.

Once these services are enabled and started, *systemd* will start the Postboard and the Scheduler (in this order) when the system starts up, and will stop them in reverse order when the system shuts down. *Systemd* will also restart them immediately if the corresponding perl scripts were killed or happened to crash.

Activities from these scripts and they standard output will usually be sent to your *syslog* daemon (this is the case on most GNU/Linux distributions), but the scripts mainly log their activies to the `postboard.log` and `scheduler.log` files in the WebObs log directory (by default `/opt/webobs/LOGS`). Additionally, the error output of the scripts will by default be redirected to the respective files `postboard_error.log` and `scheduler_error.log` in this directory (unless you changed or disabled the error file names in the configuration).

Managing the Postboard and Scheduler processes using systemd

When using systemd services to start, stop and monitor the Postboard and the Scheduler, you can manually stop these services using the following commands:

```
systemctl stop woscheduler
systemctl stop wopostboard
```

However, be aware that stopping the Scheduler this way will immediately kill any running job, just like the command `scheduler terminate` would.

If you need to stop the scheduler for administrative purposes (e.g. before upgrading your WebObs installation), it is recommended to run the command `scheduler stop` (as your WebObs user). This will wait for the running jobs to stop before the scheduler cleanly exits, which will cause a clean stop of the systemd service (it only automatically restart the scheduler in case of error).

While waiting for the running jobs to finish (which could be long), remember that remaining running jobs can be monitored using the command `scheduler runq`.

Once you are finished with your task (for example once the WebObs installation upgrade is finished), you can restart the Postboard and the Scheduler with this simple command (as already mentioned, it is not required to start the *wopostboard* service, as the *woscheduler* service will take care of that before starting the scheduler):

```
systemctl start woscheduler
```

4.5 WOC

WOC, the **WebObs Console**, is a command-line tool to query and update internal **WebObs** structures. Initially built as a developer's set of debugging tools and coding examples, **WOC** also contains some **WebObs** administrator's tools.

WOC can be run in interactive mode, interpreting and executing woc-commands at the console's WOC prompt, or batch mode, executing a single woc-command passed as argument. **WOC** is also available from a WebObs html page thru the use of woc.html + woc.js .

List of woc-commands :

%WEBOBS [key]	dump %WEBOBS key or all keys
-%WEBOBS value	query which %WEBOBS key(s) holds value
%OWNERS	dump all %OWNRS
%DISCP [discp]	dump %DISCP discp discipline or all
%USERS [login]	dump %USERS login or all

authres	list all auth resources
user login	query DB USERS for login
newuser	add a user
newgroup	add a users group
deluser	delete a user
delgroup	delete a users group
grant auth	grant access in auth table
auth login	dump login authorizations
%NODES [key]	dump %NODES key or all
proc [proc]	dump PROC proc or all
form [form]	dump FORM form or all
view [view]	dump VIEW view or all
node [node]	dump NODE node or list node names
newnode node as other	define a new node as othernode
delnode node	delete a node
nodegrids [node]	list grids that reference node
nodedev [node]	list features+devices for node (or all dev)
statnodes	statistics on node+grids
readcfg file	readCfg file
dbjobs	list all jobs definitions
newjob	add a job definition
dbruns	list all jobs last run info
sys	print system information
! cmd	exec shell cmd (WebObs vars single-quoted for interpolation)
= expr	exec perl expr (interactive mode only)
dd	keys of main hashes and their occurence
ddxref	keys of main hashes + their occurence + cross-reference
help	this help text
quit	make a guess

WOC session example

```
WOC version 1.6, Apr2013
At WOC prompt: command , 'help', or 'quit'

<WOC> statnodes
    14 node directories
     2 nodes have no grid
       GCSCC21  GCSREV1
     0 node has no proc
     1 node has no view
       ISBFDF0

<WOC> sys

Linux 3.2.0-67-generic #101-Ubuntu SMP Tue Jul 15 17:46:11 UTC 2014 GNU/Linux
Perl $^V = v5.14.2
$ENV{PATH} = /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:
@INC : /opt/webobs/WebObs-beta-1.6.5/CODE/cgi-bin:/etc/perl:/usr/local/lib/perl/5.14.2:/usr/local/share/perl/
$POSIX::VERSION = 1.24
POSIX::tzname = CET CEST
$ENV{TZ} =
/etc/localtime -> CET-1CEST,M3.5.0,M10.5.0/3
local now: 2014-08-05 08:46:33 CEST (+0200) 1407221193 (1407221193)
UTC   now: 2014-08-05 06:46:33 1407217593
$ENV{LANG} = en_US.UTF-8
Locale(s) Supported/Installed: en_US = S/I; fr_FR = S/I;
UMASK 002
PID 12492 started 1407221193 by someone (1000/1000) in /opt/webobs/CONF


"WebObs-Paris" WebObs-beta-1.6.5 [/etc/webobs.d -> /opt/webobs/CONF]

<WOC> quit
Bye.
```

Chapter 5

Developments

5.1 Use PROC's output graphics facilities

To display PROC's output graphics and data, **WebObs** uses a script named **/cgi-bin/showOUTG.pl** that can be used with external data. Products must be images in PNG format, with a thumbnail in JPG format and optional EPS and TXT files. The script proposes two different display formats:

1. per time scales: each image corresponds to a preset moving time window;
2. per dated events: each image is associated to a timestamp (date and time).

First, you must create a new PROC using the web interface with name MYPROC for example, and from any template (choose Generic time series for instance). Then you will have to name and store image files by respecting some rules explained below. Note that all files must be readable by the apache user (in **WebObs** automatic processes, the files are owned by the **WebObs** owner and group-readable).

5.1.1 Graphs per time scale

To display graphs per time scale, the PROC must define the minimum following keys:

MYPROC.conf

```
=key|value
NAME|My PROC title
TIMESCALELIST|ts1,ts2,ts3
SUMMARYLIST|SUMMARYA,SUMMARYB
```

ts1, **ts2**, **ts3** are 2-letter minimum length time scale keys (at least one key is needed, see Table 5.2 for valid keys and syntax). **SUMMARYA** and **SUMMARYB** are optional short names for additional summary graphs (all nodes). You may also associate NODES to this PROC if you want to show some per-node graphs. The outputs must be named and placed as:

/opt/webobs/OUTG/PROC.MYPROC/graphs/SUMMARY{**A,B**}_{**ts1,ts2,ts3**}.{**png,jpg,eps**}

where **.png** are the full resolution images, and **.jpg** files (same name as .png) are the thumbnails. These two files are mandatory. Optional **.eps** extension containing a vectorial image will give access to it through a link.

If you want to show individual associated node graphs, for instance from NODEID1, you name the files as:

/opt/webobs/OUTG/PROC.MYPROC/graphs/nodeid1_{**ts1,ts2,ts3**}.{**png,jpg,eps**}

Note the NODE's ID must be written in lower case in the filename. **.eps** extension file is optional.

You can have one data export file for each summary and node output. The files must be placed as:

/opt/webobs/OUTG/PROC.MYPROC/exports/{**SUMMARY*,nodeid1**}_{**ts1,ts2,ts3**}**.txt**

If the **WebObs** has been installed and configured by changing the default root path and subdirectories, you might look into **/etc/webobs.d/WEBOBS.rc** for the following variables to define the paths:

$WEBOBS**{ ROOT_OUTG}**/PROC.MYPROC/$WEBOBS{ PATH_OUTG_GRAPHS}

$WEBOBS**{ ROOT_OUTG}**/PROC.MYPROC/$WEBOBS{ PATH_OUTG_EXPORTS}

5.1.2 Graphs per event

To display graphs per event, the PROC must define only one key:

MYPROC.conf

```
=key|value
NAME|My PROC title
```

for the page title.

Events must be referenced to a date (year, month, day) and will be presented sorted by month in one page per year, showing the image thumbnails. One event can contain multiple images that will be shown together as thumbnails when clicking on it. The last display level is the full scale image itself. There is no rule for eventID and images filenames (excepted the file extensions):

/opt/webobs/OUTG/PROC.MYPROC/events/YYYY/MM/DD/eventID1/*.{**png,jpg,eps,pdf,txt**}

But it is better to use self-explanatory filenames since it will be displayed as popup windows on mouse over the thumbnails. Optional extensions **.eps**, **.pdf** and **.txt** will give access to supplementary files through links. You must also define at least one preferred image to be display as thumbnail on the main page, by creating a symbolic link to the **.jpg** file. The link basename has no importance but the extension.

/opt/webobs/OUTG/PROC.MYPROC/events/YYYY/MM/DD/eventID1/link.jpg → **maineventimage.jpg**

5.2 SUPERPROCS: Templates for applications development

5.2.1 Superprocs

Appendix

Table 5.1: Data raw formats for PROCS

```
# WEBOBS Raw formats definition for PROCS and NODES
#
#       key: used in readfmtdata.m, readnode.m and {show,form}NODE.pl
#       supfmt: superformat (for display grouping)
#       name: name for display in {show,form}NODE.pl
#       FID: additionnal FID_x keys
#
# Updated: 2019-12-24
#
=key|supfmt|name|FID
#
||-|
dsv|ASCII|delimiter-separated values file|PREPROCESSOR,FS,TIMECOLS,NF,HEADERLINES
sql-table|DB|SQL database with simple table|
miniseed|MSEED|miniSEED file|
seedlink|MSEED|SEEDLink data stream|
arclink|MSEED|SeisComP3 ArcLink data request|
combined|MSEED|SeisComP3 SeedLink/ArcLink data request|
fdsnws-dataselect|MSEED|FDSN WebServices data select|
winston|EARTHWORM|EarthWorm Winston Wave Server data request|
gipsy|GNSS|JPL GIPSY-OASIS .tdp file|ANTENNA,RECEIVER,XYZ
gipsyx|GNSS|JPL GipsyX .tdp file|ANTENNA,RECEIVER
globkval|GNSS|MIT GAMIT/GLOBK VAL file|
sbe37-ascii|GNSS|sbe37 OBP text file|DATA_ERROR,DATA_DECIMATE
usgs-rneu|GNSS|USGS RNEU text file|
ics-neu|GNSS|ICS NEU text file|
ogc-neu|GNSS|OGC NEU text file|
cr10xasc|CAMPBELL|Campbell Scientific CR10X data file|
toa5|CAMPBELL|Campbell Scientific TOA5 data file|
tob1|CAMPBELL|Campbell Scientific TOB1 data file|
porkyasc|ASCII|USGS Porky data file|AFMALARM
hyp71sum2k|QUAKE|Quake Hypo71 summary lines|MC3,MAGTYPE_DEFAULT,MAGERR_DEFAULT
fdsnws-event|QUAKE|Quake FDSN WebServices event request|MC3,MAGTYPE_DEFAULT,MAGERR_DEFAULT
scevtlog-xml|QUAKE|Quake SeisComP3-xml files|MC3,MAGTYPE_DEFAULT,MAGERR_DEFAULT
mc3|MC3|MainCourante database|
naqs-soh|ASCII|NAQS State of Health|NANOSTATYPE
teqc-qc|ASCII|TEQC Rinex Quality Check|
meteofrance|ASCII|Meteo-France raingauge|
mat-file|WO|Matlab MAT-file|
```

Table 5.2: Time scale keys (x is an integer).

Key	Time scale
xs	x seconds
xn	x minutes
xh	x hours
xd	x days
xw	x weeks
xm	x months
xy	x years
xl	last x data
rx	reference to date REFx_DATE
all	all data

Table 5.3: Marker type list

Symbol	Marker
.	Point
+	Plus sign
*	Asterisk
x	Cross
o	Circle
s	Square
d	Diamond (vertical rhombus)
^	Upward-pointing triangle
v	Downward-pointing triangle
>	Right-pointing triangle
<	Left-pointing triangle
p	Five-pointed star (pentagram)
h	Six-pointed star (hexagram)

Table 5.4: Line Style list

Symbol	Line style
-	Solid line
--	Dashed line
:	Dotted line
-.	Dash-dot line

Table 5.5: Some basic R,G,B colors

R,G,B	Color
0,0,0	Black
1,1,1	White
1,0,0	Red
0,1,0	Green
0,0,1	Blue
1,1,0	Yellow
1,0,1	Magenta
0,1,1	Cyan

Table 5.6: Some built-in colormaps

Name	Description
jet	
hsv	
landcolor	
seacolor	
white	
gray	
hot	
cool	
bone	
pink	
copper	
spring	
summer	
autumn	
winter	

Table 5.7: Date string format list. "Datenum" column indicates if the format is valid for date/time string as key value

Number	Datenum	String	Example	Comment
−1		automatic		default value
0	OK	'dd-mmm-yyyy HH:MM:SS'	01-Mar-2000 15:45:17	
1	OK	'dd-mmm-yyyy'	01-Mar-2000	
2		'mm/dd/yy'	03/01/00	
3		'mmm'	Mar	
4		'm'	M	
5		'mm'	03	
6		'mm/dd'	03/01	
7		'dd'	01	
8		'ddd'	Wed	
9		'd'	W	
10		'yyyy'	2000	
11		'yy'	00	
12		'mmmyy'	Mar00	
13		'HH:MM:SS'	15:45:17	
14		'HH:MM:SS PM'	3:45:17 PM	
15		'HH:MM'	15:45	
16		'HH:MM PM'	3:45 PM	
17		'QQ-YY'	Q1-96	
18		'QQ'	Q1	
19		'dd/mm'	01/03	
20		'dd/mm/yy'	01/03/00	
21		'mmm.dd,yyyy HH:MM:SS'	Mar.01,2000 15:45:17	
22		'mmm.dd,yyyy'	Mar.01,2000	
23	OK	'mm/dd/yyyy'	03/01/2000	
24		'dd/mm/yyyy'	01/03/2000	
25		'yy/mm/dd'	00/03/01	
26		'yyyy/mm/dd'	2000/03/01	
27		'QQ-YYYY'	Q1-1996	
28		'mmmyyyy'	Mar2000	
29	OK	'yyyy-mm-dd'	2000-03-01	ISO 8601
30	OK	'yyyymmddTHHMMSS'	20000301T154517	ISO 8601
31	OK	'yyyy-mm-dd HH:MM:SS'	2000-03-01 15:45:17	

Table 5.8: TeX stream modifiers and main escape characters allowed in titles and labels when specified, in addition to standard Greek letters and mathematical symbols.

TeX Modifier	Comment
^	Superscript or exponent (use braces for more than 1 character)
_	Subscript (use braces for more than 1 character)
\bf	Bold font
\it	Italic font
\rm	Normal font
\fontname{fontname}	Specify the name of the font family to use
\fontsize{fontsize}	Specify the font size in FontUnits
\color(colorSpec)	Specify color for succeeding characters
\backslash	backslash character
\{ or \lbrace	left brace character
\} or \rbrace	right brace character
_	underscore (low line) character
\^ or \hat	hat character

Table 5.9: Suggestion for NODE ID code: N D T S S S S

Letter	Code	Comment
N = Network	I	IPGP
	G	OVSG
	M	OVMP
	R	OVPF
	P	PVMBG
D = Domain	S	Seismology
	D	Deformations
	G	Geophysics
	C	Chemistry
	I	Imagery
	M	Meteorology
	P	Phenomenology
	A	Acquisition
DT = Technique	SB	Broad-band
	SZ	Short-period
	DC	Continuous GPS
	DT	Tilmetry
	DD	Distancemetry
	DE	Extensometry
	GM	Magnetometry
	GE	Electric
	CS	Hot Springs Analysis
	CG	Gas Analysis
	CD	DOAS
	MW	Weather station
	PJ	Journal Phenomenology
	PE	Eruption
	AT	Transmission
	AB	Buildings

Table 5.10: *_ARROWSHAPE key's parameter syntax is a 4-element vector of scalars, coma or space separated = HEADW,HEADL,HEADI,LINEW. Values are ratios relative to the length of reference scale legend arrow equals 1 (see **CODE/-matlab/arrows.m** for further details).

Parameter	Default value	Comment
HEADW	0.15	arrow's head width
HEADL	0.15	arrow's head length
HEADI	0.12	arrow's head inside length
LINEW	0.03	arrow's line width

Acknowledgments

A 19-year history summary.

Episode I. The **WebObs** project was born in October 2000 when *François Beauducel* has been assigned to the Guadeloupe volcanological observatory in Lesser Antilles. First ideas of an integrated monitoring and management system have risen thanks to fruitful discussions with *Christian Anténor-Habazac*, *Jean-Christophe Komorowski* and *Stéphane Acounis*. Quickly (and dirtily) developed in about a year of sparse hours, a first version of **WebObs** was presented in Paris on January 2002 [2], containing most of the present content: station files, networks, automatic graphs for seismic, deformation, geochemistry and weather stations, shared calendar... During the first two years of the project, there was a single developer, moreover, a scientist during its overtime work and not a dedicated computer specialist!

Episode II. From August 2002 to December 2005, *Didier Mallarino* was the first computer engineer who invested a part of its time to improve the codes and configuration files, especially by developing more robust and flexible Perl GCI scripts [3, 4, 1]. In 2004, a version of **WebObs** has been partially duplicated and adapted for a public website (CDSA).

Episode III. From 2006 to 2010, some code improvements where made by a second computer engineer *Alexis Bosson*: particularly, the system was internationalized and a first effort was made to integrate **WebObs** with observatory acquisition chain seismology standards [5]. During these years, the system worked in a relatively stable production state, and it was adapted and partly installed in different observatories: Paris (thanks to *François Truong* [6]), Addis-Abeba (thanks to *Alexandre Nercessian*), Martinique (thanks to *Jean-Marie Saurel* and *Benoît Costes*), Montserrat (thanks to *Alexis Bosson* and *Roderick Stewart*) and later in 2012 at La Réunion (thanks to *Patrice Boissier*, *Florent Brenguier* and *Philippe Kowalski*).

Episode IV (*A New Hope*). In 2012, **WebObs** obtained its first dedicated funding support from the French Ministry of Ecology, thanks to *Steve Tait*, *Arnaud Lemarchand* and *Pierre Agrinier*. A very significant contribution has been made by *Didier Lafon*, the first computer engineer working 100% on the project. Taking advantage of 10 years of production feedback, he reassessed the whole coding concept, improved and standardized the codes, made library modules and administration tools, wrote technical documentation, put all this under a versioning control system and built the first Linux installation package. This allowed to install a first alpha and beta version at Merapi observatory (thanks to *Ali A. Fahmi*), then the same codes in Guadeloupe, Martinique and La Réunion observatories, and start a real collaborative development. During this last period, we welcomed additional contributors, as developers or end-users: *Xavier Béguin*, *Jean-Marie Saurel*, *Stephen Roselia*, *Patrice Boissier*, *Laura Henriette*, and of course all the observatory teams under the direction and enthousiast support of *Jean-Bernard de Chabalier*, *Valérie Clouard*, *Andrea Di Muro*, *Nicolas Villeneuve*, *Céline Dessert*, *Aline Peltier*, *Roberto Moretti*, and *Anne-Marie Lejeune*.

Episode V. In October 2018, **WebObs** code has been released on https://github.com/IPGP/webobs.

Bibliography

[1] F. Beauducel. Surveillance opérationnelle des volcans français : développements récents à la Guadeloupe. *Géosciences BRGM*, 4:64–68, 2006.

[2] F. Beauducel and C. Anténor-Habazac. Quelques éléments d'une surveillance opérationnelle... In *Observatoires Volcanologiques*. Institut de Physique du Globe de Paris, 2002.

[3] F. Beauducel, C. Anténor-Habazac, and D. Mallarino. WEBOVS : Integrated monitoring system interface for volcano observatories. In *IAVCEI General Assembly, Chile*, 2004.

[4] F. Beauducel, C. Anténor-Habazac, and D. Mallarino. WEBOVS: Integrated monitoring system interface for volcano observatories. In *European Seismological Commission Annual Workshop, Saint-Claude Guadeloupe*. IAVCEI / IPGP, 2005.

[5] F. Beauducel, A. Bosson, F. Randriamora, C. Anténor-Habazac, A. Lemarchand, J.-M. Saurel, A. Nercessian, M.-P. Bouin, J.-B. de Chabalier, and V. Clouard. Recent advances in the Lesser Antilles observatories Part 2 : WebObs - an integrated web-based system for monitoring and networks management. In *European Geophysical Union General Assembly, Vienna*, April 2010.

[6] F. Truong, X. Lalanne, and A. Chulliat. MAGIS: The information system of IPGP magnetic observatories. In *Proceedings of the XIIIth IAGA Workshop on Geomagnetic Observatory Instruments, Data Acquisition and Processing, June 9-18 2008, Golden, USA*, 2009.