Bernese GNSS Software Version 5.2

Voume 2

Edited by

Rolf Dach, Simon Lutz,
Peter Walser, Pierre Fridez

November 2015

UNIVERSITÄT BERN

AIUB

Astronomical Institute, University of Bern

User manual of the *Bernese GNSS Software*, Version 5.2

Please send comments on the *Bernese GNSS Software* and
this manual as well as requests for copies to:

PD Dr. Rolf Dach		
Astronomical Institute	Phone:	+41 – 31 – 631 85 91 (secretary)
University of Bern	E-mail:	`rolf.dach@aiub.unibe.ch`
Sidlerstrasse 5		
CH–3012 Bern		
Switzerland		

For contact and support send e-mail to:

Pierre Fridez	Phone:	+41 – 31 – 631 85 93
	E-mail:	`bernese@aiub.unibe.ch`

ISBN: 978-3-906813-05-9 ; University of Bern, Bern Open Publishing.
DOI: 10.7892/boris.72297
Freely available online at `http://www.bernese.unibe.ch`.

November 2015

Produced in Digital Print by Publikation Digital AG, Biel in December 2015.

Bernese GNSS Software Version 5.2

Astronomical Institute, University of Bern

R. Dach,

F. Andritsch, D. Arnold, S. Bertone, P. Fridez, A. Jäggi,
Y. Jean, A. Maier, L. Mervart, U. Meyer, E. Orliac,
E. Ortiz–Geist, L. Prange, S. Scaramuzza, S. Schaer,
D. Sidorov, A. Sušnik, A. Villiger, P. Walser,

C. Baumann, G. Beutler, H. Bock, A. Gäde, S. Lutz,
M. Meindl, L. Ostini, K. Sośnica, A. Steinbach, D. Thaller

Important External Contributors:

Federal Office of Topography swisstopo, Wabern, Switzerland
E. Brockmann, D. Ineichen, S. Lutz, S. Schaer

IAPG, Technische Universität München, Germany
U. Hugentobler, C. Rodriguez–Solano, P. Steigenberger

Federal Agency for Cartography and Geodesy, BKG, Germany
D. Thaller

IPG, Technische Universität Dresden, Germany
M. Fritsche, A. Rülke

The Version 5.2 is based on the Version 5.0 that had been developed until May 2004 by many former institute members and several external contributors.

Contents

Part II.

Technical Aspects

21. The Menu System

21.1. Introduction

The *Bernese GNSS Software* is a very complex tool consisting of dozens of closely cooperating programs. It became clear already during the development of early versions of the software that there is a great desire for a supporting menu program. Such a tool should allow to use the *Bernese GNSS Software* effectively, helping the user to run the individual programs and to specify reasonable options for them. However, the available programming tools were quite restricted at that time and the only possible solution was a kind of a line–oriented menu system, based (in principle) on the vt100 terminal command set. The old line–oriented menu system had been used successfully for many years up to Version 4.2.

However, during the preparation of the Version 4.2 a number of difficulties became evident which were due to the old menu system. The main problems were the following:

- The lack of ANSI cursor control support under some operating systems resulted in restrictions of the use of the software on those platforms.
- Differences in cursor libraries on different platforms led to platform–dependent and even compiler–dependent parts of source code.
- The maintenance of the old menu system became increasingly difficult due to its complicated structure.
- The line–oriented menu system did not fulfill the requirements of today's users that are familiar with modern graphical interfaces.

Taking into account the importance of the user interface the decision was to develop a completely new software tool that would solve the problems listed above. The result of this development was the menu program of Version 5.0 , providing a user friendly graphical interface with the well–known elements of modern windows programs.

For the *Bernese GNSS Software* Version 5.2 the menu system is based on the one for Version 5.0 , however with slight modifications and extensions. It is written in C++ and based on the Qt library (`http://www.qt.io` , see Section 21.10.1 for details). It establishes one single and generic user interface program, allowing to specify sessions and campaigns, to view and edit data files, execute processing programs, start processing through the BPE, in short words, to control all functions available in the *Bernese GNSS Software* package.

This chapter gives a general overview of the menu system, introduces its features, and provides technical details on the mechanisms it is based on.

21.2. Starting the Menu System

The entire menu system consists of the following parts:

1. The only executable file is called `menu` on UNIX platforms or `menu.exe` on Windows platforms. It is located in the `${XQ}` directory on UNIX platforms (`%XQ%` for Windows).
2. Menu–related input files define the configuration of the menu system:

   `${U}/PAN/MENU.INP` is the primary menu input file and its name has to be specified when starting the menu system (see below). The paths to the remaining input files are specified in this primary file.

   `${X}/PAN/MENU_CMP.INP` contains the list of campaigns and it is common to all users (see Sections 3.3).

   `${U}/PAN/MENU_EXT.INP` defines the paths and extensions for the Bernese file types (see Section 21.4.2).

   `${U}/PAN/MENU_PGM.INP` assigns the Bernese Fortran programs to the menu items (see Section 21.4.3).

   `${U}/PAN/MENU_VAR.INP` manages the menu variables (see Section 21.6).
3. The program–specific input files are located in the user–specific directory `${U}/PAN`. To each Bernese main program corresponds one input file (default extension `INP`), where the basename is identical with the program name. It serves as the interface between the menu and the Fortran processing program. A detailed description of the input file format (default extension `INP`) is given in Section 21.10.2.
4. Help files in html format are located in `${X}/HLP` (default extension `HLP`). They have the same names as the input files.

Before starting the menu the Bernese environment has to be loaded. On UNIX platforms this is done by sourcing the file `${X}/EXE/LOADGPS.setvar` (which may be done by the login script). On Windows platforms the environment variables are defined in the registry by the installation procedure.

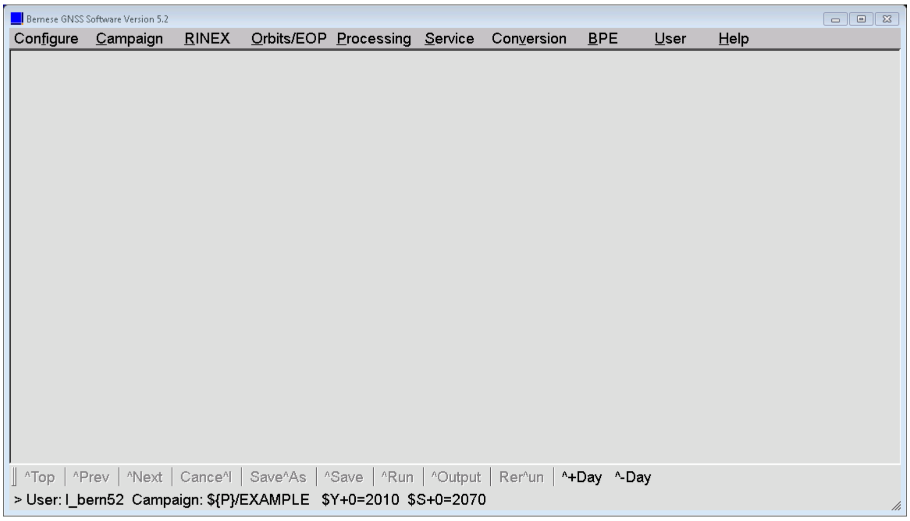
To start the menu program on Windows platforms click on the *Bernese GNSS Software* icon. Internally the name of the executable file (`menu.exe`) and the primary input file `MENU.INP` as the parameter is specified:

```
C:> %XQ%\menu %U%\PAN\MENU.INP
```

On UNIX platforms we start the executable (`menu`) indirectly using the shell script `menu.sh`. It ensures that the environment variable `${QTDIR}` is set to the correct version of the Qt library. The primary menu input file is a parameter, too:

```
> $XQ/menu.sh $U/PAN/MENU.INP
```

You may simply type `G` on the command line to start the menu. The shell script `${X}/EXE/G` is prepared for performing this task. (`${X}/EXE` is added to the `${PATH}` when the Bernese environment is loaded). In both cases the Bernese menu window appears as shown in Figure 21.1. To leave the menu use "Menu>Configure>Quit" (shortcut “Ctrl+q”).



> User: l_bern52 Campaign: ${P}/EXAMPLE $Y+0=2010 $S+0=2070

Figure 21.1.: Bernese menu system on startup.

21.3. Elements of the Bernese Menu Window

The graphical surface is composed of four parts (from top to bottom, see Figure 21.1):

Menu bar: Contains a number of drop–down menus allowing to define session and campaign, select the program to be executed, configure the menu, request help, and quit the menu.

Panel area: Represents the main part of the menu that displays the input fields for the selected program or tool (empty on startup).

Command bar: Contains command buttons for navigating forward and backward through the program panels, save the inputs, execute the programs, view the program output files and change the day of the session.

Status bar: Displays user name, active campaign, current session, and other relevant information for a specific program run.

Users that are accustomed to graphical interfaces of windows applications will feel familiar with the Bernese menu system. The buttons, drop–down menus, dialogs, and other elements can be controlled by mouse or keyboard in a very standard way.

Let us have a look at the different elements in the following sections.

21.3.1. Menu Bar

The menu bar is the top line in the menu window and contains a number of drop–down menus. The first two allow to configure the menu (fonts and colors), the data structure (data file path and extensions) and the processing setup (campaign and session). See Section 21.4.1 for details on the configuration of processing defaults. The "Menu>Configure" drop–down menu also contains the “Quit”–button (see Section 21.10.3).

The next few drop–down menus provide the tabs for accessing the processing programs. They are ordered in a logical sequence representing the processing steps successively executed in an analysis procedure (import of data, preparation of orbits and Earth orientation parameters, data processing). The "Menu>Service" drop–down menu offers a variety of tools such as the Bernese simulation program (described in Chapter 20), utilities related to station information files, observation files, checking and displaying of residuals, station coordinate and velocity files (including the time series analysis using the program FODITS, see Chapter 11), clock RINEX files , tools for the ionosphere model estimation and to support the automated processing as well as programs specifically related to LEO and SLR processing. This menu item also provides the possibility to browse program output or error message files (see Section 21.7).

Through the "Menu>Conversion" section the programs for converting binary files into ASCII or vice versa can be accessed. In addition you find there programs to extract the solution and further information from SINEX files and to convert antenna phase center offsets and patterns from the ANTEX into the Bernese format. The full list of Bernese processing programs with their location in the menu system is provided in Table 1.2, page 11 in Section 1.3 .

The "Menu>BPE" drop–down menu collects the programs for the automated processing in the framework of the Bernese Processing Engine (BPE, see Chapter 22).

The user may add his own programs to the menu system. Such programs have to follow a few rules which are described in Section 21.10.7. The programs are accessible with "Menu >User".

Finally, "Menu>Help" provides a quick access to a number of readme files and extended help texts covering general aspects as well as specific information for all Bernese programs (see Section 21.3.6).

Using the keyboard the drop–down menus may be activated using the "Alt–" sequences. The underlined characters are those that have to be used. If the drop–down menu is open only the underlined letter is necessary to continue navigation.

21.3.2. Command Bar

The command bar below the panel area provides the functions for navigating through the panels (one panel forward by using `^Next`, one panel backward by using `^Prev` or to the first panel by using `^Top`), to save chosen options in the input files, to execute the programs, to view the program output files and to quickly change the day of the session. Only the relevant buttons in a given context are active. The `^Next` button, e.g., gets inactive if the last panel for a program is displayed. The letter following the ^–sign in the button names indicates the keyboard–shortcut that may be used together with the "Ctrl"–key to activate the corresponding function.

After pressing the `^Run` button the edited input file is saved and the main program is started (either in foreground or background, see Section 21.4.4) using the current options. Alternatively the `^Save` and `Save^As` buttons that allow to store the edited options without starting the program. The input file may thus be prepared for a later execution of the program run outside the Bernese menu system (see Section 21.9). The `Cance^l` button discards the changes in the input file.

The `^Output` button allows to view the program output file after executing the program. If an error output file was specified, this file is displayed, too, if an error or warning message was generated by the program. Output and error files may also be viewed using the drop–down menus "Menu>Service>Browse program output" and "Menu>Service>Browse error message". For more details on job output and error handling see Section 21.7.

Using the Rer^un button the panels for the last executed program may be displayed again making it unnecessary to navigate every time through the drop–down menus to find the program. This is particularly useful if a program is executed several times with a few options modified.

The buttons ^+Day and ^-Day allow for a quick increase or decrease of the session's day in steps of one. This is useful if a program is run for several consecutive or close–by days, making it unnecessary to call "Menu>Configure>Set session/compute date" each time.

21.3.3. Status Bar

The status bar at the bottom of the menu window displays information specific to the active processing setup. It shows the user name, the active campaign, the current year and session, the job identifier (if defined), and the filename of the displayed program panel.

Session ($S+0), year ($Y+0), and job identifier ($J) are labeled with the corresponding names of the menu variables (see Section 21.6.1). Active campaign and current session may be changed using the drop–down menus "Menu>Campaign>Select active campaign" and "Menu>Configure>Set session/compute date" respectively (see Sections 3.3 and 3.4).

21.3.4. Program Panels

The main task of the menu system is to start the Bernese processing programs and to handle their input options. The input options are stored in user–specific input files (one file per program in directory ${U}/PAN). Within an input file, the options are thematically combined and ordered into different panels.

A panel contains a set of options which are displayed together and which are logically related to each other. For one Bernese program usually a number of panels are provided, each having an unambiguous title. Using the command buttons ^Next, ^Prev, and ^Top the user can proceed from one panel to the next, return to the previous panel, or even to the primary (top) panel as many times as wished. Already when navigating through the panels the menu system checks the user's input. If necessary, it warns the user about non–existing files, long filenames[1], missing or obviously wrong options, etc. The corresponding message box can be quit either with the Back button to return to the input file mask or with the Ignore button to force the menu to accept the entry. Of course, at this stage it is not possible to check everything (e.g., the selected files may exist but their content may be entirely wrong). When running the corresponding program, it will also check the input values and, if an unexpected values was found, stop with an appropriate error message.

Not all panels are necessarily shown. Some panels may be displayed only if the values of specific options fulfill a defined condition. If, e.g., the option "Use time window" is enabled, a subsequent panel allowing to define the time window is activated. It is also possible that the value in an input field renders another field inactive if user input in that field has no

[1] The length of the filenames including the path and extensions is limited to 32 characters in *Bernese GNSS Software*, see Section 3.5.

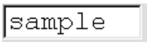
sense in combination with the first option. One or more options from different panels may be involved in the definition of the condition.

If you are interested in the technical details of the underlying mechanisms we refer to Section 21.10.

21.3.5. Panel Controls

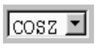
Each panel contains a couple of control elements, so–called widgets (window gadgets), that allow the users to set the options. The selected options are transferred into the Bernese processing programs via one keyword per option. These keywords become visible in a message bubble if you are keeping the mouse on the widget.

Here we give a complete list of widgets used in the Bernese menu system. There are the following five standard widgets:

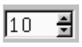


The *lineedit* is used for simple user input which may consist of comment text, a real– or integer–valued number, or an output file name. For filenames the extension for the specific file type appears just right of the field.

The *checkbox* is used if a simple yes/no answer is expected. With the box marked the corresponding option is activated, with the box unmarked the option is not active.



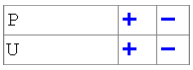
The *combobox* is used if the user has to select one of several predefined alternatives. Depending on the option the list of defined values is complete or additional values may be input. The last entry of each combobox is a variable name which, when selected, allows for a flexible definition of values in an automated processing. The value is then set via a user variable with the same name, defined, e.g., as a BPE variable, see Section 22.5.5.



The *spinbox* is used if an integer value in a given range is expected. The arrows allow to increase or decrease the value by particular amounts, but any desired value can be entered manually. As for the combobox, the 'lowest' entry is a variable name that allows to set the value via a user variable.

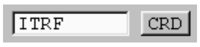
The *radiobuttons* are used if the user has to select one of several alternatives. Some other control elements may depend on the setting of the radiobuttons.

Apart from the standard control elements, three widgets were prepared especially for the Bernese menu:



The *qtable* is used to generate input tables of variable length. It replaces the *uniline* widget used in Version 5.0 and consists of editable fields and two buttons. Use a double–click with your mouse to activate an input field for editing. If an input field is already activated you can alternatively navigate with the "Tab" key or the "Shift+Tab" combination to the next or previous input field. The +–button repeats the line (the new line can then be edited), the

-–button deletes the line (the last remaining line cannot be deleted). In some cases the qtable lines appear without + and -–buttons. In this case only the activated input fields accept user input.



ITRF

The *selwin* is typically used for selecting input files. This widget consists of a lineedit and a selection button just right to it labeled with the extension of the expected file type. Pressing the button, a standard file–selection dialog appears allowing the user to select the desired input file(s). Before pressing the selection button the user may restrict the selection using wildcards * and ?. For example, if you specify F* in the editable field and presses the selection button CRD you may select from files ${P}/MYCAMP/STA/F*.CRD.

Depending on the specific input field, more than one input file may be selected. In that case the string SELECTED appears in the field. Pushing the selection button allows to check the names of the selected files. The selwin file selection dialogue offers the possibility to browse ASCII files. In some cases the widget is also used to select other items than filenames, e.g., a subset of station names from a list.

${X}/PAN

The *comment* displays a variable text like a normal text (e.g., the name of the file currently edited in the panel). It can not be edited by the user.

Path and extensions for input and output files are file–type specific and predefined for each selwin (input files) and for each lineedit (output filenames). Even if it is not recommended they may be changed in the panels "Menu>Configure>Paths and extensions", see Section 21.4.2.

Note that some of the widgets may be inactive if they are not relevant in the chosen setup. Their content is not used for the processing. Also note that the user input is not translated to upper case. This is important, e.g., for filenames on UNIX systems (UNIX systems are case sensitive).

A consequent use of menu time variables in filenames and for the definition of time windows in the program panels significantly improves the flexibility of the program system in case processing steps have to be performed repetitively for different sessions. The lineedit, qtable and selwin input fields accept menu time and user variables (see Section 21.6).

Navigation through the options inside a panel using the “Tab”–key is supported. Use the “space”–key (or left mouse button) to select an item. In spinboxes or comboboxes the “up”–key resp. “down”–key may be used to change the values. When selecting from a list (e.g., within the file–selection dialog), use “Ctrl” + left mouse button for selecting separate files and “Shift” + left mouse button for selecting a range of files.

21.3.6. Help

The "Menu>Help" provides extended help covering general aspects as well as specific information for all Bernese programs. The help files are located in the directory ${X}/HLP and are written in html–format. The help window of the menu program is based on the html–window provided by the Qt library. Alternatively, you may use any html–browser to view the help files (e.g., to print them). Options in the program panels are linked to the corresponding section in the program help file using the option–related keyword as an anchor in the html–based help file. This allows to immediately access the help section for the specific option (select the option and then "Menu>Help>Help on context"). The online help provides default values for many of the program options.

Apart from the help files, "Menu>Help>Readme" provides direct access to all README files in the directory ${X}/DOC. These contain useful information concerning the installation and update of the software and the JPL ephemeris, upgrade from Version 5.0, first steps, the preparation of the oceanic and atmospheric tidal loading coefficients and the VMF1 coefficients, and the anonymous ftp server of the AIUB. Furthermore, the README files describing the example BPEs (see Chapter 23) can also be accessed (located in the directory ${U}/PCF).

The help–browser allows to search for strings in a help panel. Links to different sections in the same panel or to other panels help to quickly collect the information you require. Arrow keys at the bottom of the help window support you in navigating forward and backward through the different links you selected (backward jumps return to the position of a defined anchor point in the text which need not be the original position in the text in all cases).

21.4. Processing Defaults

21.4.1. Configuring the Menu

The user may change color, fonts, and font sizes separately for the different components of the menu using the items in the submenu "Menu>Configure>Menu layout". Note that for the panel font, you should not select a proportional font. Using "Menu>Configure>Menu layout >Reset menu layout" it is always possible to reset the settings to factory defaults (use the key sequence "Alt+f l r y" in case you selected an illegible font or color). The size of the menu may be changed by dragging the window border.

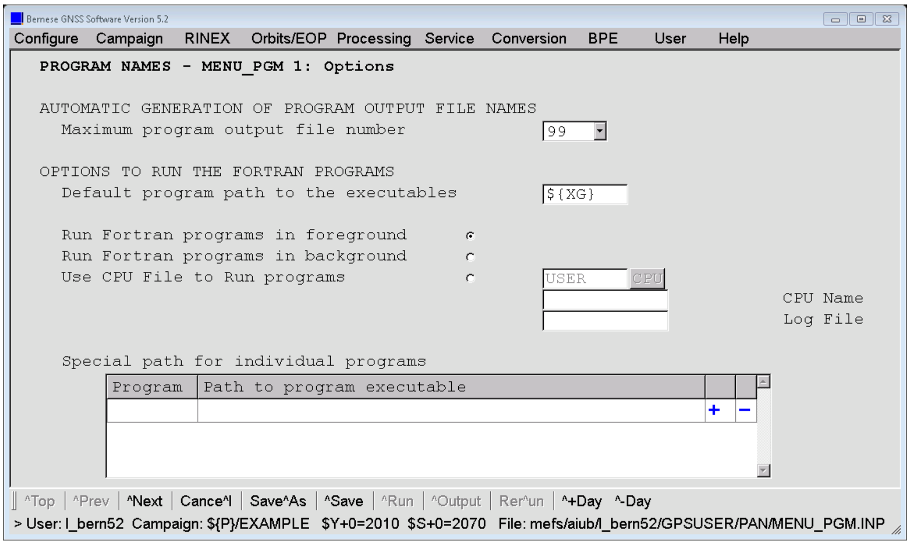
The settings are recorded in the user–specific configuration file ${U}/PAN/MENU.INP when exiting the menu. It includes the active campaign and the current session, too. When opening the menu the next time the configuration will again be the same. We refer to Section 21.10.3 for a detailed description of the mechanism — especially if you use more than one menu in a given user environment at the same time.

How to specify the campaign, to manipulate the session table, to define the current session and the job identification character is described in Sections 3.3 and 3.4.

21.4.2. Path and Extensions of Data Files

Path and extensions for all data files may be changed using "Menu>Configure>Paths and extensions". Scrolling through panels may be useful to get an overview over the available files and their location in the data structure. It is, however, *not recommended to change these defaults.*

The definitions of path and extensions are saved in the user–specific file ${U}/PAN/MENU_EXT.INP.



Bernese GNSS Software Version 5.2
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un ^+Day ^-Day

```
PROGRAM NAMES - MENU_PGM 1: Options

AUTOMATIC GENERATION OF PROGRAM OUTPUT FILE NAMES
  Maximum program output file number           99

OPTIONS TO RUN THE FORTRAN PROGRAMS
  Default program path to the executables      ${XG}

  Run Fortran programs in foreground      ●
  Run Fortran programs in background      ○
  Use CPU File to Run programs            ○    USER  CPU
                                                              CPU Name
                                                              Log File

  Special path for individual programs
      Program | Path to program executable
```

> User: l_bern52 Campaign: ${P}/EXAMPLE $Y+0=2010 $S+0=2070 File: mefs/aiub/l_bern52/GPSUSER/PAN/MENU_PGM.INP

Figure 21.2.: Options to run the Fortran programs, "Menu>Configure>Program names".

21.4.3. Path to Executables, Names of Programs and Input Files

Using "Menu>Configure>Program names" the path to the Fortran program executables are defined in option "Default program path to the executables" (see Figure 21.2). System environment variables may be used (enclosed in curly brackets). The default value is `${XG}` and should not be changed. The list in option "Programs executed from special paths" allows to specify a different path for specific programs. This may be useful when working with a modified version of a Bernese program or when including user programs into the menu (see Section 21.10.7).

The menu expects that the filenames of the executable, of the program input file, and of the corresponding help file are identical (apart from path and extension). It is not recommended to change any of the options in the subsequent panels. The panels itself are stored the file `${U}/PAN/MENU_PGM.INP`.

21.4.4. Program Execution Mode

If a processing program is started in foreground — the default setting — the menu system is blocked until the main program ends. The advantage is that the user is notified about the end of a program run and the output of the program is displayed by pressing the `^Output`–button.

On UNIX platforms programs can be started either in foreground or in background. This can be specified with "Menu>Configure>Program names", option "Run Fortran programs in background" (see Figure 21.2). By starting the program in background the menu system loses control over it and it is up to the user to check whether the program has finished or not. On the other hand, starting the program in background allows to continue working with the menu system immediately. Take care on output and auxiliary filenames to avoid conflicts if you run several programs or one program several times in parallel (remember, the scratch files are specified in the "General Files" panel of the corresponding program and is displayed only on request). It is a good idea either to include the variable for the jobID ($J) somehow into the filenames or make consequent use of session specific file naming.

This option is ignored on Windows–platforms where programs are always executed in foreground.

21.4.5. Execution of Programs Using a CPU Control File

If you are working on a system with a job queuing/distribution system you may run a single program from the menu in this system. To do so, select "Menu>Configure>Program names", option "Use selected CPU file to run Fortran programs". The command is specified in a CPU control file (as it is also used for the BPE, see Section 22.4) which you have to specify. You can select a CPU name (from the CPUs specified in the CPU control file) and the name of the log file collecting all output. The only difference to the usage of the CPU control file for BPE usage is that you have to run the script ${XQ}/start_pgm.sh instead of ${BPE}/RUNBPE.pm, e.g.:

```
RSH_PGM  rsh host ${XQ}/start_pgm.sh <ARGV> >> <LOG> 2>&1 &
```

The job commands are constructed like for the BPE. Remember, that the resulting command needs to run in the background. For more details on setup CPU control file entries we refer to Section 22.4.

This option is ignored on Windows platforms.

21.5. The Active Campaign and the Current Session

21.5.1. Active Campaign

Data processing in the *Bernese GNSS Software* is campaign–oriented. All input data, intermediate files, and results are read from or written to subdirectories in a campaign directory. The menu needs to know the campaign you are currently processing. The *active campaign* is listed in the status bar of the menu. All campaigns are listed in the campaign list with the default name ${X}/PAN/MENU_CMP.INP. This is a general file accessed by all users and is the only file in the master panel directory which needs write–access for all users.

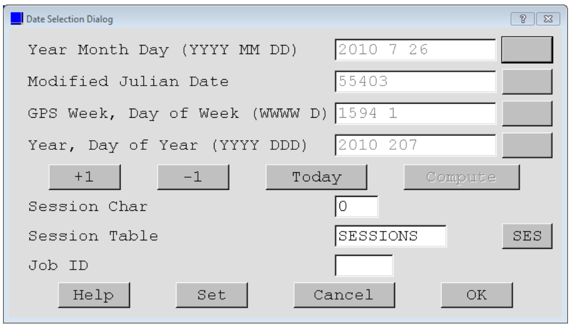
The drop–down menu "Menu>Campaign" provides the items

- to select the active campaign from the campaign list,
- to add or remove a campaign from the list, and
- to create a new campaign directory structure.

How to create a new campaign is described in Section 3.3.

21.5.2. Current Session

A session is the time interval covering all observations to be processed together. A four–character session string is assigned to a time window in a session table (see Section 3.4).



Date Selection Dialog

```
Year Month Day (YYYY MM DD)        2010 7 26
Modified Julian Date               55403
GPS Week, Day of Week (WWWW D)     1594 1
Year, Day of Year (YYYY DDD)       2010 207
Session Char                       0
Session Table                      SESSIONS
Job ID
```

Figure 21.3.: Date selection dialogue to set the current session.

A *current session* has to be defined for data processing with the *Bernese GNSS Software* Version 5.2 . The current session is used to set the menu time variables which are recommended to be used in the input fields, e.g., as part of filenames or to specify a time window, for an efficient processing (see Section 21.6.1 for details concerning menu time variables).

The current session can be set using "Menu>Configure>Set session/compute date" which provides a date selection dialogue as shown in Figure 21.3. You may specify the date for the session using one of the four supported formats. Specify the session character in field "Session Char" and the name of a session table in the field "Session Table". The current session is set using the buttons `Set` (dialogue box remains open) or `OK` (dialogue box is closed). With the button `Compute` the other fields are filled with the input date in the corresponding format. The session dialogue can thus be used as a date converter. Using the buttons `+1` and `-1` you may increment or decrement the current day by one. The date for the current session can also be adjusted using the `Today`–button.

If the session `ssss` is not defined in the session table, a warning message (`Session ssss not found in session table`) will appear when closing the date selection dialogue. The resulting current session is displayed in the status line of the menu. Change the active campaign before you choose the current session if the session tables of both campaigns are not compatible.

21.5.3. Job Identification Character

The Job ID can be used to distinguish multiple input files for a given program. It may be defined using "Menu>Configure>Set session/compute date" (see Figure 21.3). The Job ID is appended to the extension of the input files in your user specific panel directory (e.g., `${U}/PAN/GPSEST.INP_T1`). In this way it is possible to predefine panels with different settings of options for the same program which you may in the following address with the job identification character. If you would like to run more than one job of the same program at the same time, you have to use different job identification characters to obtain independent input files for the program runs.

The Job ID defines the value of the predefined menu variable `$J` (see Section 21.6.1), which you may use in all input panels. It may be used, e.g., to easily distinguish the temporary file if you run the same program more than once at the same time. Technically, there are no restrictions about the length of the Job ID. However, we recommend to specify not more than two characters for the Job ID.

Table 21.1.: Predefined menu variables.

Variable:			Format:	Description:
`$Y`			`yy`	2–digit year
`$M`			`mm`	Month
`$D`			`dd`	Day of month
`$J`			`jj`	Job ID
`$+n`	`$-n`	`$+-`	`ddd`	Day of year, DoY
`$S+n`	`$S-n`	`$S+-`	`dddf`	DoY, session character
`$Y+n`	`$Y-n`	`$Y+-`	`yyyy`	Year
`$W+n`	`$W-n`	`$W+-`	`wwww`	GPS week
`$M+n`	`$M-n`	`$M+-`	`yymm`	Year, month
`$JD+n`	`$JD-n`	`$JD+-`	`ddddd`	Modified Julian date
`$WD+n`	`$WD-n`	`$WD+-`	`wwwwd`	GPS week and day
`$YD+n`	`$YD-n`	`$YD+-`	`yyddd`	Year and DoY
`$YSS+n`	`$YSS-n`	`$YSS+-`	`yydddf`	Year, DoY, sess. char.
`$YMD_STR+n`	`$YMD_STR-n`	`$YMD_STR+-`	`yyyy mm dd`	Year, month, day

21.6. Menu Variables

When processing GNSS data, it is often necessary to repeat a program run several times with slightly different option settings. A typical example would be the processing of several sessions of data. The names of the observation files are session dependent as the session number is usually part of their filenames. It would be very cumbersome to repeat all the runs selecting the correct files manually every time. For the BPE an automatization is mandatory. For such cases the Bernese menu system provides so called menu variables. The menu variables are defined in the user–specific menu input file `${U}/PAN/MENU_VAR.INP` and they are accessible through "Menu>Configure>Menu variables". Three kinds of menu–variables are available: predefined variables (also called menu time variables), user–defined variables, and system environment variables.

21.6.1. Menu Time Variables

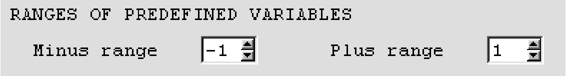
Most of the predefined menu variables are time variables based on the current session. Table 21.1 gives the list of all predefined variables. Wherever possible, it is recommended to use time variables in all input fields of the program panels, especially for input and output filenames, e.g., `IGS$YD+0`. The widgets lineedit, selwin, qtable, and combobox (with user input) accept menu variables as input. A consequent use of time variables increases the efficiency of the menu system considerably because you can process different sessions by simply changing the current session without changing any field in the program panels. When using the BPE for automatic processing the use of menu time variables is indispensable to set the input options in program panels (see Section 22.7.2).

The syntax of the predefined variables needs an explanation. Except for the first four variables (`$Y`, `$M`, `$D` and `$J`) in Table 21.1 the variable names consist of a first part (the dollar sign plus zero or more capital letters) and a second part. The second part may be:

- Plus sign and one digit (e.g., `+0`),
- minus sign and one digit (e.g., `-1`),
- plus–minus sign (i.e., `+-`), or
- two tilde signs (i.e., `~~`).

The meaning of this syntax is the following: Any appearance of `$S+0` — as an example — is replaced by the current session. `$S+1` corresponds to the next session according to the session table, `$S+2` to the second next session, etc. Similarly `$S-1` corresponds to the previous session. If `n` is larger than 9 the number has to be put into brackets, e.g., `$S+(200)` meaning the session 200 sessions ahead of the current session. Similarly the variable `$Y+0` holds the 4–digit year of the current session, the `$Y+1` the year of the next session (that may or may not be the same as `$Y+0`), etc..

The syntax with the `+-` or `~~` sign used for so–called range variables is very powerful, too. On the top of panel "MENU_VAR 2: Ranges" (accessible "Menu>Configure>Menu variables") the so–called ranges are defined:



RANGES OF PREDEFINED VARIABLES

Minus range -1   Plus range 1

The range variables are not translated into a single variable but into a list of variables within the specified ranges. Here is an example: Assuming that the current (daily) session corresponds to December 31, 2012, and the ranges are −1 and +1 the string `NEQ$YD+-` or `NEQ$YD~~` then resolves into the following three strings:

```
NEQ12364
NEQ12365
NEQ13001
```

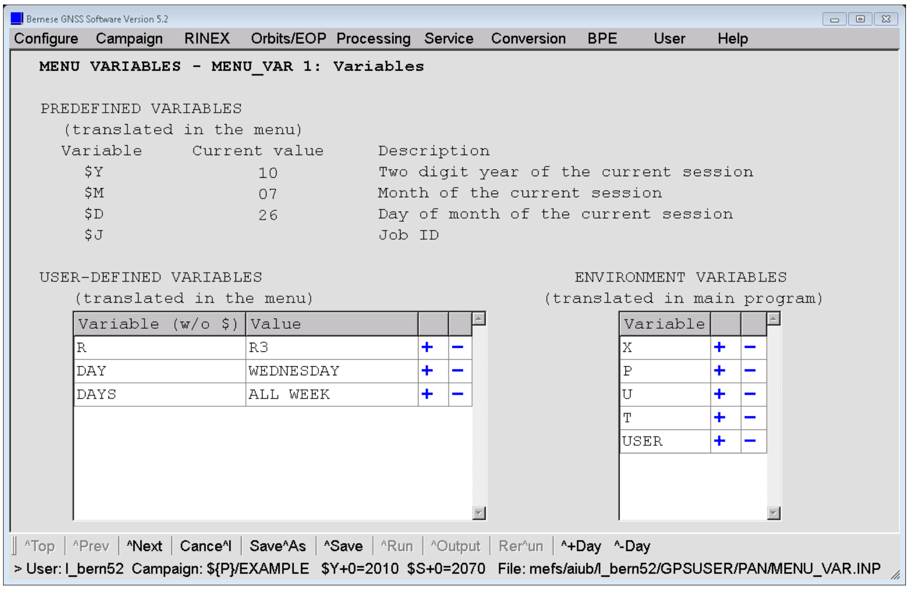
The range variables are very useful, e.g., if one wants to select several files from consecutive sessions. Range variables may be used for widgets where a list of values may be specified (i.e., more than one input file for the widget selwin or a range of sessions to specify a time window).

The difference between the range variables `NEQ$YD+-` and `NEQ$YD~~` is the following: Using the variable `NEQ$YD+-`, all the files selected by this range variable must exist, otherwise the program aborts. On the other hand, files selected by `NEQ$YD~~` are not mandatory; if some files are missing, the program does not stop and simply uses the existing subset of files.

Note that the menu time variables always refer to the *sessions* as defined in the session table. This must not necessarily be days (especially for a fixed session table or an open session table with more than one session per day). E.g., for a fixed session table the variable `$S+1` is undefined (*empty*) if `$S+0` refers to the last session in the table. For an open session table, however, `$S+1` refers to the next session in the table which may be the first session of the following day. For more examples and details we refer to the online help.

21.6.2. User Variables

The user may define an arbitrary number of variables using the qtable widget "User-defined variables" in the first of the menu variable panels ("Menu>Configure>Menu variables", see Figure 21.4). In this example the variable `$(DAY)` is translated by the menu system to


Bernese GNSS Software Version 5.2
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un ^+Day ^-Day

```
MENU VARIABLES - MENU_VAR 1: Variables

PREDEFINED VARIABLES
  (translated in the menu)
 Variable     Current value     Description
   $Y              10           Two digit year of the current session
   $M              07           Month of the current session
   $D              26           Day of month of the current session
   $J                           Job ID

USER-DEFINED VARIABLES                          ENVIRONMENT VARIABLES
   (translated in the menu)                   (translated in main program)
```

Variable (w/o $)	Value		
R	R3	+	−
DAY	WEDNESDAY	+	−
DAYS	ALL WEEK	+	−

Variable		
X	+	−
P	+	−
U	+	−
T	+	−
USER	+	−

> User: l_bern52 Campaign: ${P}/EXAMPLE $Y+0=2010 $S+0=2070 File: mefs/aiub/l_bern52/GPSUSER/PAN/MENU_VAR.INP

Figure 21.4.: Example for the menu and environment variable definition in *Bernese GNSS Software* Version 5.2 ("Menu>Configure>Menu variables").

WEDNESDAY wherever it is used. For 1–character variables the brackets are not necessary (you may, e.g., use $R in an input field). For variables with more than one character the brackets are mandatory to get an unambiguous variable names. The user–defined variables may be used in all widgets where menu time variables are permitted. Be aware that the variable names are case–sensitive.

The widgets combobox and spinbox provide a special entry referring to the related keyword, e.g., $(SAMPL). This allows to specify the content of these elements also as user variables. Even if this feature is mainly intended to be used for automated processing in the BPE environment, the values may also be defined in the table "User-defined variables". It is obvious that the values for these variables must be compliant with the specifications of the related widget, e.g., an integer number within the specified range for a spinbox.

The variables $Y, $M, $D, and $J are predefined (see previous Section) and are reserved. They may not be used as user variables.

21.6.3. System Environment Variables

System environment variables that must be known to the Fortran programs have to be defined in option "Environment variables" ("Menu>Configure>Menu variables", see Figure 21.4). At least the variables

U	path to user environment,
T	path to temporary data environment,
X	path to general files environment,
P	path to campaign drive, and
USER	user name

have to be defined. The environment variables are not interpreted by the menu but made available to the Fortran programs using the hidden keyword ENVIRONMENT in the program input files (see Figure 21.5 in Section 21.10 for an example). The Fortran program may then replace these variables

- before a filename is used for any disk–operation, or
- when the title section of the program output file is generated (the variable `${USER}` is used to print the user name).

The environment variables are used in path names, e.g., in the panels "Menu>Configure>Paths and extensions" defining the path names for the data files and "Menu>Campaign>Edit list of campaigns" defining the path and names of the campaigns. This allows to cope with the limitation of filenames (including path and extension) to 32 characters in the *Bernese GNSS Software.*

Environment variables always have to be enclosed in curly brackets to be distinct from user variables (that are translated by the menu). Because the environment variables are handled by the menu and the Fortran programs their syntax is platform independent. Use the same notation also on Windows platforms.

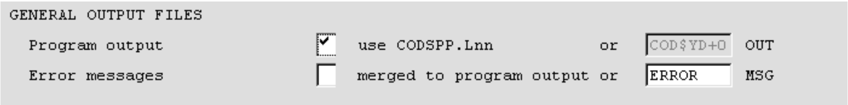
A message box `System variable not defined` indicates that an undefined environment variable is used in an input field. If the Fortran program cannot open a file and the error message looks like:

```
### SR RPLENVAR: Variable X not resolved in: ${X}/GEN/CONST.


*** SR OPNERR: OPEN FAILED
               FILE NAME : ${X}/GEN/CONST.
               PGM./SUBR.:     DEFCON
               IOSTAT    :    175
               FILE UNIT :    999
```

The environment variable has to be added to the list of variables in option "Environment variables" in the first panel of "Menu>Configure>Menu variables" displayed in Figure 21.4. Be aware that the variable has to be added to `MENU_VAR.INP` in each BPE–option directory in `${U}/OPT`.

21.7. Program Output and Error Handling

Each Bernese main program reports its result in a program output file and (optionally) in an error message file. The names of these files can be specified by the user for each program panel individually. Here is an example:



```
GENERAL OUTPUT FILES
  Program output     ☑  use CODSPP.Lnn              or  COD$YD+0  OUT
  Error messages     □  merged to program output or  ERROR     MSG
```

The user may specify a name for the program output file or use the default name. The default name consists of the main program's name and the extension `Lnn` where `nn` is a counter incremented every time the input file is saved. The counter starts with `00` and runs up to a maximum value which is specified in option "Maximum program output file number" ("Menu>Configure>Program names", see Figure 21.2). The number `nn` to be used for a subsequent program run is stored in a file `pgmnam.J` in the campaign's `OUT`–directory. This file is locked with an extension `_lk` while it is updated. If the `J`–file is deleted the counting starts again at `00` and a new `J`–file is created. The maximum program output file number may be any value up to 999. If it is greater than 99 the extension switches from Lnn to nnn. We recommend to specify explicit filenames (corresponding to result filenames) for all important program runs. In case that a lock file was not removed the extension for the output is set to XXX.

Error and warning messages can either be merged into the program output file or written into a separate file located in the ${U}/WORK directory with default extension MSG. An error message starts with three star–characters (***). It is issued if the program cannot continue execution for any reason. A warning message starts with three hash characters (###). It is issued to point out an aspect of the recent program run which, depending on the application, may or may not be critical (e.g., if a satellite was not found in the standard orbit file).

The menu system remembers the name of the output file of the last program run. This file can easily be displayed using the ^Output–button. Older program output files and error message files may be found through "Menu>Service>Browse program output" and "Menu>Service >Browse error message". The browser for program output files and error message files offers the possibility to search for a string. This may help to navigate through long output files.

In case a program encounters an unhandled error and aborts, a message usually appears on the program's standard output. This message may only be seen in the terminal window from where the menu was started (on UNIX platforms only). If this window was closed or if the menu was not started from a command line (e.g., under Windows), this output cannot be seen. In order to get such messages on the screen, execute the program without the menu (see Section 21.9).

21.8. Change Options, Update Panels

The two functions described in the following two sections are integral parts of the menu system and can, therefore, not be executed outside of the menu.

21.8.1. Change General Options

Program CHNGEN ("Menu>Configure>Change general options") allows to change the values for a specific keyword occurring in all panels in a specified list of directories. This may be useful if you have to change an option that appears in several input files in a consistent way, e.g., to change the nutation model filename. You may specify

- one or more program input files in ${U}/PAN,
- all panels in the directory ${U}/PAN resp. all panels in one or several BPE option directories (wildcards are allowed, e.g., ${U}/OPT/*, ${U}/OPT/PPP_GEN, or ${U}/OPT/PPP_*), or
- all panels in directories listed in an input file (see Section 24.3.20 for a file format description).

The last two options take into account not only the files with the extension INP, but also the input files extended by the Job ID INP_* (see Section 21.5.3).

The names of general files such as the file of constants, the satellite information file, the file containing the geodetic datum parameters, etc., may be changed to new names in a straightforward way (see panel "CHNGEN 2: Change General Input Files"). The values for any keyword may be changed using the options in panel "CHNGEN 3: Change any keywords – use with care" (only available when the option "Change any keywords" in the panel "CHNGEN 2: Change General Input Files" is activated). Use this second option with care. In general, keywords need not to be unique across input panels of different programs, and with this option, it would be easily possible to unintentionally change keywords with the same name in different program panels. See the online help for more information.

21.8.2. Update Input Files

Program UPDPAN ("Menu>Configure>Update program input files") can be used to update program input files in the users panel directory ${U}/PAN or in one or more BPE option directories ${U}/OPT/* (wildcards are allowed). You may also specify a file containing a list of directories with panels to be updated (see Section 24.3.20 for a file format description). Input files with the default extension (INP) and extensions including a Job ID (INP_*, see Section 21.5.3) will be updated.

The program is useful whenever panels for the Bernese programs are changed and new input options are added (e.g., with certain updates of the software or due to changes by the user).

In this case new panels will be created by the program adding the new options from a master set of panels, while retaining the user settings for existing options. All fields that cannot be edited by the user are taken from the master panel in any case (apart from the main menu input file ${U}/PAN/MENU.INP). When using the program be sure to use UPDATE for option "Update options: update/copy" in order not to overwrite your files. See the online help for more information.

21.9. Calling Programs Without the Menu System

All Bernese Fortran programs can be started without the menu system as well. Every Bernese program has exactly one input file. How to start a Bernese program is system dependent (Windows and UNIX systems). On Windows platforms a program can be started by typing the following command on the command line

```
 C:>  %XG%\program %U%\PAN\program.INP
```

In other words, the name of the program input file must appear as a command line parameter.

Because of the lack of a standard way of reading command line parameters by a program written in Fortran, the name of the program input file is read from the standard input on UNIX platforms. This means that a Bernese main program can be started by typing, e.g., the following command

```
 > echo $U/PAN/program.INP | $XG/program
```

On UNIX systems the command

```
 > SJ program
```

may be used to start a program using the input files located in $U/PAN from the command line.

Starting Bernese programs without the menu system will probably not be used very often. This possibility is useful mainly for debugging purposes to get access to the standard output.

Any program, also programs not available within the menu (e.g., AMBCHK) may be started under UNIX using the command

```
RUNGPS AMBCHK
```

This command starts the menu with the panel for the selected program displayed. The options may then be set and the program executed as usual. After the execution of AMBCHK the input panel is displayed again. A second argument of the command RUNGPS is the path to the executable. Default is ${XG} . The command expects that the input file has the same name as the program. If a corresponding input file exists in the current directory, this file is used. Otherwise the input file from ${U}/PAN is taken.

Another way to start a program that is not supported by the menu is to include it as user program in "Menu>Configure>Program names" and start it with "Menu>User". If the path to the executable is different from the default ${XG}, it has to be specified in the option "Programs executed from special paths" (see Figure 21.2). This mechanism also works on Windows platforms.

21.10. Technical Details

21.10.1. The Qt Library

The menu system is a single application written in C++. Its source code has approximately 18,000 lines. One of the most important features of the menu system is its platform independence (on source code level). Nowadays, there are literally dozens of toolkits and libraries supporting the development of a Graphical User Interface (GUI) on a large variety of computer platforms. However, in case of multi–platform (or platform–independent) GUIs there are only a few good alternatives each of them having its pros and cons. After a number of tests we decided to use the Qt library that is currently owned by the Finnish company Digia (see http://www.qt.io). Due to the importance of this library for the Bernese menu system several remarks concerning Qt should be given:

- Qt is a C++ object–oriented library.
- Qt is the so–called GUI–emulating toolkit. It allows the menu system to be compiled for a large variety of platforms (Microsoft Windows 8, 7, Vista, XP[2] and various UNIX's) without any platform–dependent pieces of source code.
- Qt is a commercial product. However, users of an application written with it do not have to pay a license fee. There is no such thing as a runtime license fee. In other words: the Astronomical Institute of the University of Bern, has paid the developer's license, the users of the *Bernese GNSS Software* are not subject to any license fee.

[2] see http://www.qt.io

- There is one difference between the Windows and the UNIX version of our menu system. For Windows platforms we distribute the menu program in an executable form and the user does not need to know anything about the Qt library at all. On the other hand the UNIX version of the menu program is distributed in source code and the user has to install the (freely available) Qt library on his machine before compiling the menu system himself. For installing the Qt library we refer to its online documentation and to the web page `http://www.qt.io`. For *Bernese GNSS Software* Version 5.2 , the Qt library version 4.x is necessary. See the `install.txt` file in the Bernese distribution for information on downloading the Qt library.

21.10.2. Input File Format

The input files play a central role in the design of the Bernese menu system. These ASCII–files provide the interface between the Bernese main programs (written in Fortran) and the menu program itself. Their format is sufficiently generic to fulfill the requirements of all Bernese programs and at the same time providing the menu program with all necessary information to present the necessary options in the GUI . The Bernese main programs consider the input files to be read–only (with two exceptions: PUTKEY and MENUAUX). They read all options at the beginning of the program run. On the other hand, the menu program reads and writes the input files. From this point of view the menu program can be seen as a specialized editor.

As stated above, each program has exactly one input file. The file has the same name as the corresponding program. The default location is `${U}/PAN` and default extension is `INP`. The file consists of an arbitrary number of keywords and panels.

21.10.2.1. Keywords in Input Files

As an example, part of the ADDNEQ2 program input file is displayed in Figure 21.5. The keyword structure requires several comments and explanations.

- Lines containing the exclamation mark (`!`) as the first non–blank character are treated as comments by the menu system and the main program.
- The keywords are written in capital letters. They start at the beginning of the line.
- The keyword is followed by the number of values associated with it.
- Keyword values are enclosed by double quotes. If the number of values is equal to one, the value is on the same line as the keyword (e.g., keyword `SHOWGEN` in Figure 21.5). The values are listed on separate lines if more than one value per keyword is present (e.g., keyword `INPFILE` in Figure 21.5).
- Lines beginning with two hash marks (`##`) are only read by the menu system and contain all the information for displaying and handling the keyword by the menu program. They are ignored by the main programs. A list of all widgets and their valid metacommands is given in Table 21.2. Multiple lines with metacommands are allowed (see, e.g., keyword `COVCOMI` in Figure 21.5).
- If the value to be displayed in the panel input field is different from the keyword value to be used by the main program, a line with a single hash mark (`#`) gives the value the user entered in the input field. Examples can be found in Figure 21.5, showing, e.g., the use of wildcards for keyword `INPFILES` or the use of menu variables for keyword `COORD`.

```
...
! Environment variables
! ---------------------
ENVIRONMENT 5
  "P" "/aiub"
  "X" "/aiub_sw/BERN50/GPS"
  "U" "/u/aiub/bernese/GPSUSER"
  "T" "/scratch/bernese"
  "USER" "bernese"
  ## widget = initmenu; pointer = ENVIR_LIST

! ========================================================================
! ADDNEQ2 1: Input Files
! ========================================================================
! Show all general files
! ----------------------
SHOWGEN 1  "1"
  ## widget = checkbox

MSG_SHOWGEN 1  "Show all general files"

! Normal equations
! ----------------
INPFILE 4
  "${P}/EXAMPLE/SOL/F1_021430.NQ0"
  "${P}/EXAMPLE/SOL/F1_021440.NQ0"
  "${P}/EXAMPLE/SOL/F1_031380.NQ0"
  "${P}/EXAMPLE/SOL/F1_031390.NQ0"
  ## widget = selwin; path = DIR_NEQ; ext = EXT_NEQ
  # F1_*

MSG_INPFILE 1  "Normal equations"

INPFILE_TXT_COL_1 1  "Name"

! Variance rescaling factors
! --------------------------
COVCOMI 0
  ## widget = selwin; path = DIR_WGT; ext = EXT_WGT; maxfiles = 1
  ## emptyallowed = true
  #

DESCR_COVCOMI 1  "Variance rescaling factors"

! Station coordinates
! -------------------
COORD 1  "${P}/EXAMPLE/STA/REF02143.CRD"
  ## widget = selwin; path = DIR_CRD; ext = EXT_CRD; maxfiles = 1
  # REF$YD+0

DESCR_COORD 1  "Station coordinates"
...
```

Figure 21.5.: Extraction of a program input file (`${U}/PAN/ADDNEQ2.INP`).

- One blank line (at least) is used to separate keywords.
- The comments in field–specific message boxes generated by the menu are taken from keywords `MSG_keyword` or `DESCR_keyword` present for each keyword. Keywords `DESCR_keyword` accompany each keyword specifying a single filename (e.g., keyword `COORD` in Figure 21.5). This information is used by the Fortran programs to generate a list of input and output filenames at the beginning of the program output file. The title for a list of files printed to the program output file is given by a keyword similar to `INPFILE_TXT_COL_1` in Figure 21.5.

The input file mechanism includes a special construct to use one set of file base names for various entry fields. This allows the user, e.g., to enter the names of the observation header files only. The names of the corresponding observation data files are then automatically created in the Fortran programs by exchanging path and extension by the values given in keywords like `PSFILES_PTH_COL_2` and `PSFILES_EXT_COL_2`. The indicator `_2` in this example numbers the "columns" of different file types. Usually these two keywords are

Table 21.2.: List of metacommands defining the behavior of the widgets.

Metacommand	Description	Widgets	default
`widget`	Type of the widget to be displayed	all widgets	
`cards`	List of items that can be selected in a combobox	combobox	
`editable`	Combobox is editable (`true`) or not editable	combobox	not editable
`emptyallowed`	User input may remain empty (`true`) or an input is required (`false`)	selwin lineedit	`false` `true`
`ext`	File extension (`EXT_xxx` from `MENU_EXT.INP`)	lineedit, selwin	no extension
`group`	Groups of radiobuttons	radiobutton	
`keepvariables`	Do not translate menu (time and user) variables for this keyword when saving the input file	selwin, lineedit, qtable, initmenu	false (translate)
`maxfiles`	Maximum number of selectable files (integer)	lineedit, selwin	unlimited
`multilineallowed`	Multiple values for a lineedit are allowed (`true`)	lineedit	`false`
`numlines`	Number of lines to be displayed by the qtable widget (integer)	qtable	
`path`	Absolute path (`PTH_xxx`) or campaign directory (`DIR_xxx`) from `MENU_EXT.INP`	lineedit, selwin	no path
`pointer`	Points to another keyword, e.g., selwin selects from the list defined by this keyword	initmenu, selwin	
`radiokeys`	List of keywords that are activated by the radiobutton	radiobutton	
`range`	Range and step for spinbox (integer for start, end, step)	spinbox	
`activeif`	Keyword is active under the specified condition	all widgets	
`check_max`	Maximum value of a user input	lineedit, qtable	
`check_min`	Minimum value of a user input		
`check_strlen`	Maximum length of strings (integer)		
`check_type`	Expected user input (`integer`, `real`, `date`, or `time`)		
`menuaux`	Keyword value is prepared by the auxiliary Fortran program (MENUAUX)	(`MENUAUX` in all cases)	
`action`	Action performed by program MENUAUX		
`menuauxkeys`	List of keywords. The current values of these keywords will be passed to program MENUAUX	lineedit, selwin, qtable	
`updateifchanged`	The value of the keyword is updated only if a keyword in this list changes		
`updateifsave`	Update the content of the keyword using the MENUAUX–mechanism if the panel is saved		
`updateifactive`	Update the content of the keyword using the MENUAUX–mechanism only if at least one of the keywords behind is active		

Required metacommands are		
	combobox:	`cards`
	radiobutton:	`group`
	selwin:	either `path`/`ext` or `pointer`
	spinbox:	`range`
	qtable:	`numlines` (a range between 5 and 30 is expected)
	menuaux:	`action`

assigned to the widget initmenu pointing to the keywords for directory and extension (e.g., DIR_PSO resp. EXT_PSO) in the file ${U}/PAN/MENU_EXT.INP. The values of keywords like PSFILES_TXT_COL_2 accompanying these multicolumn keywords are used to generate titles in input file tables generated by the Fortran programs.

A number of metacommands are available to check the correctness of user input. The command check_type = integer, e.g., checks if an input is an integer value. If the user enters a value with an unexpected type the menu asks to correct the entry upon leaving the panel (by navigating to a different panel or by saving the panel file). The user may go back to the panel to correct the value or insist on his input.

The metacommand activeif allows to define conditions under which an input field is active or not. If the specified condition on the values of other keywords is not fulfilled, the field remains inactive and does not accept user input. The metacommand activeif = COORD / _ is, e.g., used to activate an input field if a coordinate file is specified in field COORD. Several conditions may be combined by AND or OR. They are analyzed from left to right.

The keyword ENVIRONMENT is the interface for the Bernese environment variables. The environment variables listed in option "Environment variables" ("Menu>Configure>Menu variables", see Figure 21.4) are the first column of the keyword values. The second column lists the corresponding values of the environment variables at the moment when the input panel was saved.

21.10.2.2. Menu Panels in the Input Files

An arbitrary number of keywords delimited by at least one blank line makes up the first part of the input file. However, this first part does not provide all the information needed by the menu program. The menu program must know how to organize the widgets for displaying them to the user. This information is stored in the second part of the input file consisting of an arbitrary number of panels. As an example the ASCII definition of a panel is given in Figure 21.6.

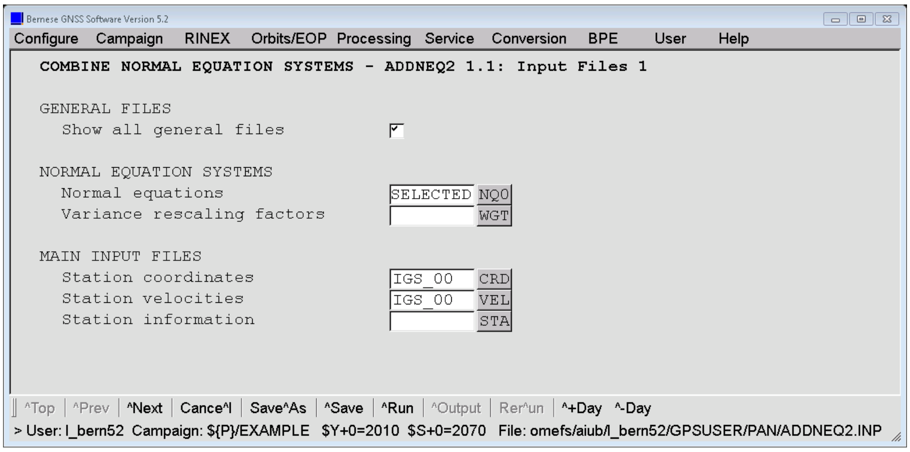
Each panel is defined by lines containing the hash mark (#) as the first non–blank character. It may be insightful to look at the resulting panel displayed in Figure 21.7. It can be seen that a part of the panel was directly printed to the canvas of the menu window. The rightmost part of the panel is not shown and the fields like > %%%%%%% < (*masks* in our terminology) are replaced by the widgets.

Nevertheless, we should add some comments:

- The panel actually defines only the size and the position of the widget on the canvas. All remaining information (first of all the widget type) is given in the description line of the corresponding keyword.
- The first line of the panel may contain a condition that has to be fulfilled to show the panel. This provides a mechanism to hide panels that are not necessary under a specific constellation of options. With NO_CONDITION the panel is always displayed.
- The second line of each panel provides a title line containing the program name, an unambiguous panel number, and a text describing the content of the panel.
- Each panel consists of lines that might contain the so–called masks. Each mask corresponds to one keyword. The keyword name is given at the same line in its rightmost part. There may be more than one mask per line.

```
# BEGIN_PANEL NO_CONDITION #####################################################
# COMBINE NORMAL EQUATION SYSTEMS - ADDNEQ2 1.1: Input Files 1                 #
#                                                                              #
# GENERAL FILES                                                                #
#   Show all general files      > % <                                          # SHOWGEN
#                                                                              #
# NORMAL EQUATION SYSTEMS                                                      #
#   Normal equations            > %%%%%%%% <                                   # INPFILE
#   Variance rescaling factors  > %%%%%%%% <                                   # COVCOMI
#                                                                              #
# MAIN INPUT FILES                                                             #
#   Station coordinates         > %%%%%%%% <                                   # COORD
#   Station velocities          > %%%%%%%% <                                   # VELAPR
#   Station information         > %%%%%%%% <                                   # STAINFO
# END_PANEL ####################################################################
```

Figure 21.6.: ASCII definition of a panel, see Figure 21.7.



```
COMBINE NORMAL EQUATION SYSTEMS - ADDNEQ2 1.1: Input Files 1

GENERAL FILES
  Show all general files          ☑

NORMAL EQUATION SYSTEMS
  Normal equations                SELECTED  NQ0
  Variance rescaling factors                WGT

MAIN INPUT FILES
  Station coordinates             IGS_00    CRD
  Station velocities              IGS_00    VEL
  Station information                       STA
```

> User: l_bern52 Campaign: ${P}/EXAMPLE $Y+0=2010 $S+0=2070 File: omefs/aiub/l_bern52/GPSUSER/PAN/ADDNEQ2.INP

Figure 21.7.: Graphical representation of a panel, see Figure 21.6.

- The panel is never changed by the menu.

The relationship between the panel and the keyword structure is given by the keywords (see Figure 21.5 in our example).

21.10.3. Menu Configuration

In interactive mode the menu command is executed with a single input argument `${U}/PAN/MENU.INP` as described in Section 21.2. The user–specific file `MENU.INP` (see Figure 21.8) contains in the first section the names of files containing the processing defaults, the menu variables, and the campaign list. The second section provides the settings for the current session and the active campaign. The third section contains the settings of the menu such as colors, fonts, size of the window, etc. The settings are read when the menu is started and are written to the configuration file when exiting the menu. Therefore, the chosen settings will be kept and recovered when opening the menu next time.

```
! Release Information Notes
! -------------------------
RELEASE_INFO 1  "${X}/DOC/RELEASE.TXT"

! List of Remaining Menu Input Files (full Pathes)
! ----------------------------------------------
MENU_CMP_INP 1  "${X}/PAN/MENU_CMP.INP"

MSG_MENU_CMP_INP 1  "Edit list of campaigns"

MENU_VAR_INP 1  "${U}/PAN/MENU_VAR.INP"

MSG_MENU_VAR_INP 1  "Menu variables"

MENU_EXT_INP 1  "${U}/PAN/MENU_EXT.INP"

MSG_MENU_EXT_INP 1  "Paths and extensions"

MENU_PGM_INP 1  "${U}/PAN/MENU_PGM.INP"

MSG_MENU_PGM_INP 1  "Program names"

! Active Campaign
! ---------------
ACTIVE_CAMPAIGN 1  "${P}/EXAMPLE"

! Modified Julian Date
! --------------------
MODJULDATE 1  "55403.000000"

! Session (only the last character is relevant)
! -------------------------------------------
SESSION_CHAR 1  "0"

SESSION_TABLE 1  "${P}/EXAMPLE/STA/SESSIONS.SES"
  ## widget = selwin; path = DIR_SES; ext = EXT_SES; maxfiles = 1
  # SESSIONS

! Job character
! -------------
JOB_ID 1  ""

! Size of the Main Window
! -----------------------
MAIN_WIDTH 1  "1142"

MAIN_HEIGHT 1  "742"

! Colors (RGB Button + RGB Background)
! -----------------------------------
MAIN_PALETTE 1  "192 192 192 222 222 222"

! Font Types and Font Sizes
! -------------------------
FONTTYPE_BASE 1  "Arial"

FONTSIZE_BASE 1  "14"

FONTBOLD_BASE 1  "0"

....
```

Figure 21.8.: Example of a primary menu input file (${U}/PAN/MENU.INP).

You are able to start the menu more than once for the same user. The configuration of the menu is independent for each running menu. As a consequence the configuration file contains the information from the menu you have left last.[3]

[3] On UNIX platforms the shell script ${X}/EXE/G makes a copy of the original ${U}/PAN/MENU.INP file which is then used to run the menu. When quitting the menu the configuration file is copied back to the original location.

Table 21.3.: Keywords for starting the menu in special modes.

Keyword	Description
INP_FILE_NAME	Name of the program input file to be read by the menu
OUT_FILE_NAME	Name of the program output file to be written by the menu
MODUS	Execution mode of menu (INTERACTIVE) If the menu is started with two arguments it normally runs in the non–interactive mode.
RUN_BPE	Name of the input file for starting a BPE (${U}/PAN/RUNBPE.INP), see Section 22.8.2
RESETCPU	Name of the CPU–control file to be reset, see Section 22.4
PRINT_PID	Prints the system process ID of the menu program to the error output
UPDPAN	Executes the program UPDPAN to update program input files, see Section 21.8.2

21.10.4. Running the Menu in Special Modes

The menu may be started in interactive (see Section 21.2) or non–interactive mode. In non–interactive mode the menu needs a menu command file as an additional (second) argument:

```
${XQ}/menu.sh ${U}/PAN/MENU.INP $U/WORK/MENUTASK.INP            # for UNIX platforms
```

```
%XG%\menu  %U%\PAN\MENU.INP %U%\WORK\MENUTASK.INP               # for Windows platforms
```

This way of starting the menu is used, e.g., when starting the BPE from the command line (see Section 22.8.2) or to run the menu in a BPE user–script (see Section 22.6.3). The auxiliary input file MENUTASK.INP contains information on how the menu shall be started and what it shall do. The file has to be created either manually or with a user–script using an arbitrary location and filename.

The information in the file MENUTASK.INP is keyword–annotated using the same notation as in usual panel files. The keywords for special modes of the menu are listed in Table 21.3. In addition to these keywords it is possible to overwrite the values of any keyword in the input files MENU.INP, MENU_EXT.INP, MENU_PGM.INP, and MENU_VAR.INP, allowing to modify the entire configuration of the menu. In the non–interactive start of the BPE, e.g., the active campaign and session are overwritten by the start script.

The keyword MODUS allows to start the menu in interactive mode, an option which is, e.g., used in the script RUNGPS for starting programs which are not available through the menu. The script defines the input file name (keyword INP_FILE_NAME), and the path to the executables (by overwriting the value of the option "Default program path to the executables" in panel MENU_PGM.INP, "Menu>Configure>Program names") using the user's input.

21.10.5. Remote Execution of the Menu

If you want to use the menu system to process GNSS data on a remote server, one possibility is to locally display the remote menu via XServer. However, if your internet connection is slow and if you have a local installation of the menu, a better method is to remotely execute the menu. To do so, first establish an ssh tunnel using the port, e.g., 1234 on both sides:

```
local> ssh -L 1234:remotehost:1234 user@remotehost
```

Then, launch ${XQ}/BSWD.pl as a daemon on the remote system:

```
  remote> ${XQ}/BSWD.pl 1234 &
```

Finally, start the menu locally in the "MODEM" mode:

```
  local> ${XQ}/menu '${U}/PAN/MENU.INP' MODEM localhost 1234
```

Notice that the single quotation marks in the last call are necessary for the system variable $U to be translated only on the remote system.

21.10.6. The MENUAUX–Mechanism

The menu program does not access Bernese data files directly in order to avoid the duplication of file formats in different reading and writing routines (in Fortran and in C++). To access data files a special mechanism is implemented using the Fortran program MENUAUX. This allows, e.g., to display selection lists that are generated from input files such as, for example, station names for datum definition extracted from binary Bernese observation files or from normal equation files. In principle any specialized action may be performed using this mechanism.

The MENUAUX–mechanism is initiated by the metacommand menuaux in association with a panel input field (see Section 21.10.2) specifying the auxiliary program to be executed (usually MENUAUX). Required is the additional metacommand action describing the requested action. The mechanism may be described by the following example associated with an input field allowing to select stations for datum definition in GPSEST:

```
STALST      0
  ## widget = selwin; menuaux = MENUAUX; action = OBS
  ## menuauxkeys = CONST PSFILES CSFILES PZFILES CZFILES RZFILES
```

If the user presses the button related to the selwin widget pointing to STALST the menu generates an input file for program MENUAUX. It specifies the action OBS and containing the actual values (corresponding to the current user selections) of the keywords in the GPSEST–panel listed in metacommand menuauxkeys. With this input file ${U}/WORK/MENUAUX.INP_pid_nnn (pid is the system process ID and nnn a running number) the auxiliary program MENUAUX is executed which processes the requested action (in the example extraction of the station names from the binary observation files listed in the menuauxkeys, action OBS) and returns the list of station names in a keyword MENUAUX_RESULTS in the same file that acted as MENUAUX input file. The menu in its turn reads this file, checks for possible error messages (see below), and inserts the list of stations into the selwin widget which then is displayed to the user. The user does not realize that a Fortran program was executed in the background and selects the stations to be constrained from the selection window.

A total of 67 actions are implemented in the menu auxiliary program for Version 5.2 , see subroutine ${LG}/MENUAUXS.f90. They include all edit actions associated with Bernese files (e.g., CRD_EDIT, CRD_SAVE for editing of Bernese coordinate files) and actions such as browsing of binary observation files, extraction of antenna names from observation file headers, extraction of header information from residual files, and many more.

MENUAUX runs for each keyword associated with a menuaux metacommand

- if a panel is opened containing the keyword (widgets `lineedit` or `qtable`),
- if the user activates a widget associated with the keyword (widget `selwin`), or
- if the input file is saved (except of the case that the metacommands `updateifsave` or `updateifactive` are present and the related conditions are false).

The MENUAUX–mechanism represents a powerful generic tool allowing to handle a multitude of special actions that may depend in detail on a specific situation and processing options. All special actions are implemented in a separate auxiliary program and are activated through the program input files while the menu program itself keeps its generic nature.

Special MENUAUX Metacommands

A special mechanism is required to suppress the execution of MENUAUX in order that keywords retain the user–selected values when the user is returning to a panel containing `menuaux`–keywords.

The situation may be illustrated by an example: In the panel of program HELMR1 the user may select two coordinate files (keywords `COORD1` and `COORD2`). When opening panel "HELMR1 2.1: Manual Station Selection" the program MENUAUX provides a list of all stations with non–blank flags that are available in both files and the user may mark each station with a flag `E` (exclude station) or `M` (compute residuals but do not use station to compute Helmert transformation parameters). If the user then browses through other panels and returns to panel "HELMR1 2.1" his selection of the flags should not be overwritten by another execution of MENUAUX.

Figure 21.9 shows the part of the program input file `HELMR1.INP` that is relevant for the description of the built–in mechanism to solve the described problem. The keyword `STASELECT` contains the list generated by MENUAUX and edited by the user. A hidden

```
...

STASELHLP     8
  "BRUS 13101M004" "P" "A" ""
  "FFMJ 14279M001" "P" "A" ""
  "MATE 12734M008" "I" "A" ""
  "ONSA 10402M004" "I" "A" ""
  "PTBB 14234M001" "P" "A" ""
  "VILL 13406M001" "I" "A" ""
  "ZIMJ 14001M006" "P" "A" ""
  "ZIMM 14001M004" "P" "A" ""
  ## widget = uniline; menuaux = MENUAUX; action = HELMERT
  ## menuauxkeys = DATUM COORD1 COORD2 USESTA OTHSTA
  ## updateifsave = RADIO_2 = 1

STASELECT     8
  "BRUS 13101M004" "P" "A" "M"
  "FFMJ 14279M001" "P" "A" "M"
  "MATE 12734M008" "I" "A" ""
  "ONSA 10402M004" "I" "A" ""
  "PTBB 14234M001" "P" "A" "M"
  "VILL 13406M001" "I" "A" ""
  "ZIMJ 14001M006" "P" "A" "M"
  "ZIMM 14001M004" "P" "A" "M"
  ## widget = uniline; menuaux = MENUAUX; action = HELMERT
  ## menuauxkeys = DATUM COORD1 COORD2 USESTA OTHSTA
  ## updateifsave = RADIO_2 = 1; updateifchanged = STASELHLP

MSG_STASELECT 1 "Station name / Flags / selection"
...
```

Figure 21.9.: Extraction of the `HELMR1.INP` for the manual station selection with an automatic generation of the default settings using the program MENUAUX.

keyword STASELHLP contains the original list provided by MENUAUX and is linked to keyword STASELECT by the metacommand updateifchanged. As long as the content of the hidden keyword does not change the content of keyword STASELECT is not updated — even if the input file is reopened after executing program HELMR1. If the user selects another input coordinate file the hidden keyword STASELHLP is updated and the values for keyword STASELECT is refreshed, too.

Keyword values are also updated if the input file is saved. The content of the keywords STASELECT and STASELHLP is, however, only relevant if manual station selection is enabled (keyword RADIO_2 is 1). The metacommand updateifsave allows to execute MENUAUX only if a specific condition of keyword values is fulfilled when the panel file is saved (see example in Figure 21.9).

MENUAUX Error Handling

The MENUAUX–mechanism has a built in error handling mechanism. The menu detects if the program MENUAUX cannot be executed or whether the auxiliary program encountered a handled or a fatal error and informs the user with a corresponding error message. To this purpose the menu defines in the input file for MENUAUX the name of an error output file as well as a return code (keyword MENUAUX_IRCODE) which is initialized to 9. As a first action MENUAUX sets the value to 5. After processing the requested action the return code is set to 0 — if no error occurred — or to 2 — if an error was encountered. If the menu then finds a return code of 5 in the file returned by MENUAUX the auxiliary program did start but failed with a fatal error. If the return code 2 is found the menu displays the corresponding error or warning message provided by MENUAUX.

The input file and the error message file of a MENUAUX–run is deleted by the menu immediately after the execution. To make them available (e.g., for debugging purposes) the keyword MENUAUX_DEBUG in file ${U}/PAN/MENU_EXT.INP can be set to 1 using an ASCII editor.

21.10.7. User–Specific Additions to the Menu System

We have tried to make the menu system a little bit extensible and give the user an opportunity to use the Bernese menu system for his own programs. The user programs may be written in any language (e.g., C++, Fortran, Perl). There are some requirements

- The program has to have exactly one input file in a given format (see Section 21.10.2). On Windows platform the program has to read the name of its input file as a command line parameter, on UNIX platform it has to read the input file name from the standard input as described in Section 21.9.
- The input file is expected to be located in the directory ${U}/PAN with the extension INP.
- A help file may exist in the directory ${X}/HLP with the extension HLP.
- The path to the executable may be specified in "Menu>Configure>Program names" (see Figure 21.2). The default is the directory ${XG} or make use of the option "Programs executed from special paths"

Adding a user program into the menu system is done simply by specifying the name of the program in "MENU_PGM 3: Menu - User" ("Menu>Configure>Program names"). The user programs are then accessible from the menu under "Menu>User".

22. Bernese Processing Engine (BPE)

22.1. Introduction

Nowadays permanent GNSS tracking networks have been established for different purposes in many regions of the world. The daily amount of data collected by such networks calls for a highly automated GNSS data processing. Large campaigns are also being organized with a few tens to a few hundreds of sites occupied. For this type of tasks the *Bernese Processing Engine (BPE)* has been developed allowing for a highly automated GNSS data processing. The first version of the BPE has been provided with the *Bernese GNSS Software* Version 4.0 . Version 5.2 of *Bernese GNSS Software* includes now the second generation of BPE that was developed from scratch in 2002 with a significantly simplified architecture.

The BPE has been primarily designed for the automation of processing permanent GNSS networks. However, experienced users use the tool not only for processing data stemming from permanent networks but even in many other cases when a certain level of automation of the processing is desirable. The BPE was rated many times as an extremely useful tool and has become a very important part of the *Bernese GNSS Software.* It is used at the CODE AC of the IGS for the routine processing of the global IGS network since 1995.

The BPE is based on the following features:

- Server–client architecture, communication between the server and the clients is based on TCP/IP.
- The server is written in C++ and integrated into the menu program.
- The client is written in Perl and can be used under diverse operating systems (Windows, various UNIX, and Mac OS systems).
- On UNIX platforms the so–called user scripts may be written in (almost) any programming language. We use (and recommend to use) Perl. On Windows platforms scripts have to be written in Perl.
- The BPE can be run in interactive mode (with the graphical user interface) as well as in non–interactive mode. In the non–interactive mode the BPE process can be started from the command line or its start can be scheduled using, e.g., the cron utility on UNIX systems or a task scheduling utility on Windows. The different ways of running the BPE are described in Section 22.8.

The BPE may be used to perform all possible tasks, sequentially or in parallel. A processing sequence of Bernese programs can be defined, scripts may be included which download data, perform quality reports and processing protocols, submit or archive result files. The BPE processing tasks are defined by the user in so–called Process Control Files (PCFs). The BPE is able to run more than one task at a time on different CPUs, and can even divide a single task across different CPUs. The BPE also includes control structures that allow to perform conditioned jumps forward and backward in the processing sequence.

Seven sophisticated example BPEs are delivered together with the software (description in Chapter 23) which can be used directly or may be adapted by the user to his particular needs.

In this chapter we describe the three main parts of the BPE (server, client, and user scripts), important files used by the BPE and the way the BPE can be setup and executed. However, prior to this description we have to make a remark concerning the directory structure used by the BPE.

22.2. Directory Structure Used by the BPE

The BPE uses several directory trees. Standard Bernese environment variables contain the names of these directories (e.g., the variable `${X}` may contain the directory name `/home/bernese/BERN52/GPS` on a UNIX system or something like `C:\BERN52\GPS` on a Windows machine) and we use these variables when we refer to the corresponding directories in a platform–independent manner. The directories may be divided into the following four categories (see also Section 3.1):

1. Software–directories (`${BPE}`, `${X}/EXE`, `${XG}`, `${XQ}`). The following directories are particularly important for the BPE:

   `${XG}` Executables of the Bernese processing programs.
   `${XQ}` The menu program (acting as BPE server at the same time).
   `${X}/EXE` The file `LOADGPS.setvar`, that defines the Bernese environment, is located here together with some useful scripts.
   `${X}/SCRIPT` Auxiliary scripts.
   `${BPE}` The so–called BPE client and several BPE tools used by the BPE (written in Perl).

2. User area (`${U}`, and subdirectories). The following directories are particularly important for the BPE:

   `${U}/PAN` The Bernese input files and the so–called CPU control file (see Section 22.4).
   `${U}/PCF` The so–called Process Control Files (see Section 22.5).
   `${U}/SCRIPT` The so–called user scripts (see Section 22.6).
   `${U}/OPT` Subdirectories with input files prepared for the various BPE processes (see Section 22.7).

3. Temporary area (`${T}`, and subdirectories). This area is used by BPE clients during their run. It contains a temporary copy of a part of the `${U}` area. After having successfully finished the BPE client removes all temporary files unless the “no clean” option is used in "Menu>BPE>Start BPE processing".

4. The area where the GNSS data reside (e.g., `GPSDATA`) consisting of three main structures:

   DATAPOOL (`${D}`): area where a local copy of all relevant input data for the GNSS processing exists
   CAMPAIGN–area (e.g., `${P}`): campaign–directories with their subdirectories for all GNSS processing activities with the Bernese processing programs
   SAVEDISK–area (`${S}`): area where the main results of the campaign–specific processing are stored

   The example BPEs introduced in Chapter 23 make use of such a directory structure.

22.3. BPE Components and Environment

22.3.1. BPE Server

The BPE server is the central part of the BPE. It is written in C++ (using the classes from the Qt Network module) and is part of the Bernese menu program. The executable resides in the `${XQ}` directory.

After the BPE server has started, it performs the following tasks according to the flow chart given in Figure 22.1:

1. The BPE server reads its input options. The input options contain the name of the *Process Control File* (PCF) and the *CPU Control File*. The first file defines the sequence of tasks that will be executed and the second defines which hosts can be used. These files are described in detail in Sections 22.4 and 22.5, respectively.
2. According to the `PCF` and `CPU` file, the server starts the first client (it may happen that several clients are started simultaneously). The `CPU` file provides the command how a client is started. If the client is processed on a remote machine, the remote shell `rsh` or the secure shell `ssh` command may be used (not available on Windows systems).
3. The first task of the newly started client is to open the TCP/IP connection to the server. From the moment when the connection is accepted by the server, the client is permanently connected until it terminates (if the connection is lost for some reasons the client is able to reestablish the connection). Figure 22.2 displays the sequence of communications between server and client.

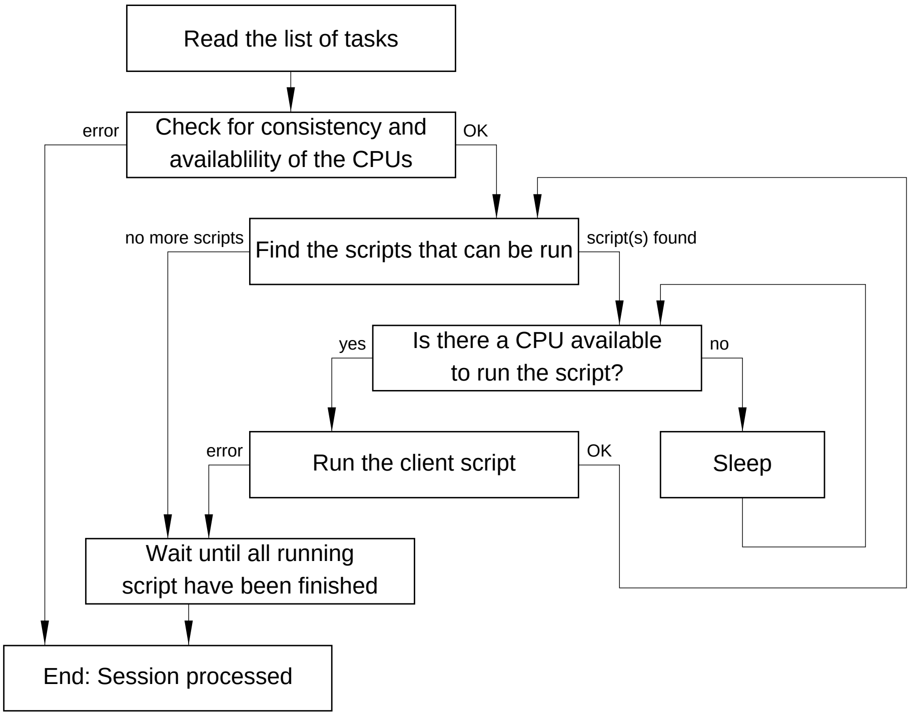

Read the list of tasks
error
Check for consistency and availablility of the CPUs
OK
no more scripts
Find the scripts that can be run
script(s) found
yes
Is there a CPU available to run the script?
no
error
Run the client script
OK
Sleep
Wait until all running script have been finished
End: Session processed

Figure 22.1.: BPE server flowchart.

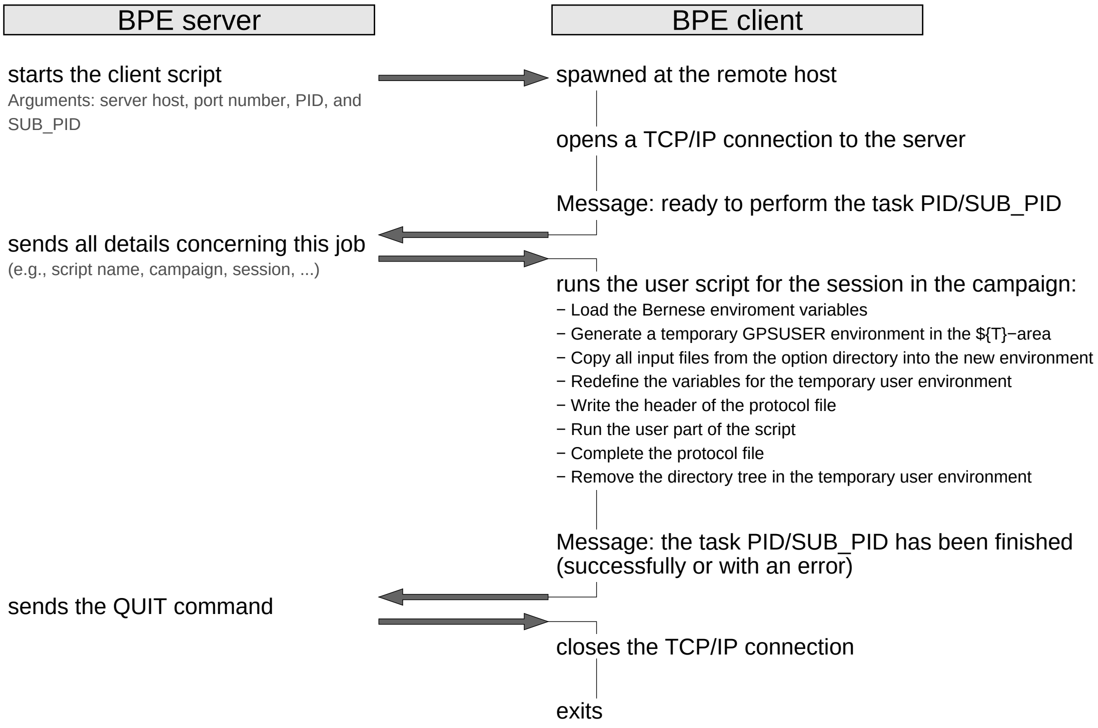

BPE server
BPE client
starts the client script
Arguments: server host, port number, PID, and SUB_PID
spawned at the remote host
opens a TCP/IP connection to the server
Message: ready to perform the task PID/SUB_PID
sends all details concerning this job
(e.g., script name, campaign, session, ...)
runs the user script for the session in the campaign:
− Load the Bernese enviroment variables
− Generate a temporary GPSUSER environment in the ${T}−area
− Copy all input files from the option directory into the new environment
− Redefine the variables for the temporary user environment
− Write the header of the protocol file
− Run the user part of the script
− Complete the protocol file
− Remove the directory tree in the temporary user environment
Message: the task PID/SUB_PID has been finished (successfully or with an error)
sends the QUIT command
closes the TCP/IP connection
exits

Figure 22.2.: Communications between BPE server and BPE client.

4. After successfully establishing the connection between server and client, the server starts to send commands to the client and the client sends back its responses. The first message from the server usually commands the client to run a specific *user script* (see Section 22.6). The last response from the client informs the server about finishing the user script. The last command from the server tells the client to disconnect and terminate.

5. In the way described above the BPE server keeps the information about all running clients. After finishing one client the BPE server decides (according to the PCF) whether new client(s) may be started. This means that steps (2) — (5) are repeated until the entire tasks defined in the PCF have been finished or one of the client scripts reported an error.

6. Server and clients report the information about their execution as well as CPU– and run–time statistics into several output files. See Section 22.10 for details.

22.3.2. BPE Client

The client of the BPE process is the single Perl module `${BPE}/RUNBPE.pm` written in Perl object–oriented style. As stated above the client is started by the BPE server. Figure 22.2 displays the sequence of communications between BPE server and BPE client. After being started the client performs the following tasks:

1. The client establishes the TCP/IP connection to the server. The needed information (address and port number of the server) is received by the client as command line arguments when it is started.

2. On server command the client prepares the environment and starts the user script that resides in the directory `${U}/SCRIPT`. The environment prepared by the client is described in detail in Section 22.3.3.
3. The user script reports to the server as soon as it is finished (including the status: either successfully or with an error message).
4. On server command the client closes the connection and exits.

22.3.3. The Environment of the Client

The environment (i.e., the set of variables that are defined and may be used in various scripts) of the BPE stems from the standard environment defined in the file `${X}/EXE/LOADGPS.setvar` (on Windows systems, this environment is stored in the registry at installation time under HKCU/Environment). However, in distinction to the standard environment there are two differences:

1. many additional variables are defined (see Section 22.6.2), and
2. the variable `${U}` is re–defined by the BPE client in such a way that it points to a (newly created) client–specific directory.

The environment variables for *Bernese GNSS Software* are handled differently under Windows and UNIX systems. On a Windows machine all the necessary variables are defined as system variables. A special variable `BERNESE_VARIABLES` contains a list of those system variables that are made available for the *Bernese GNSS Software* client scripts. The field "Client's environment file" in the first panel of "Menu>BPE>Start BPE processing" is not used under Windows and may remain empty. The user may run the BPE server on a Windows computer. We assume that under Windows the BPE clients run on the same host. In that case it is automatically assured that server and clients have the same set of system variables (see the note in Section 22.6.2 about general variables available in user scripts). Technically the BPE on Windows platforms offers also the opportunity to run client jobs remotely on any other Windows computer. Unfortunately there is no standard way for such a procedure offered by the operating system.

On UNIX and Mac platforms the environment for *Bernese GNSS Software* is prepared by reading the file `${X}/EXE/LOADGPS.setvar` which has to be available on each host on which clients are started. There are two possible problems with this concept:

1. The BPE server and the BPE client may generally source different files having thus different environments. This allows a certain flexibility but may easily also cause consistency problems.
2. Typically, `${X}/EXE/LOADGPS.setvar` may be sourced during the startup of the login shell. It is, e.g., executed with the file `.bashrc` (in case of Bourne–again shell) or with `.tcshrc` (in case of `tcsh`). A BPE client started using the `rsh` or `ssh` command may, however, not execute the startup files at all.

The user is fully responsible for solving the first problem. Possibly different `LOADGPS.setvar` files on different hosts must be compatible, i.e., they must create the same variables pointing to the same directories.[1]

[1] Actually using the same directories is not mandatory. However, using different directories for the server and for the client would probably be very rare.

The second problem is solved in the client script ${BPE}/RUNBPE.pm. The client receives the name of the environment file from the server (from option "Client's environment file" in "RUNBPE 1: Client Environment/Session Selection" "Menu>BPE>Start BPE processing", see Figure 22.13 on page 514) and the file (usually ${X}/EXE/LOADGPS.setvar) is read and interpreted by the client itself on each client host, see Section 22.6.2.

Take care when you are using environment variables in the path to this configuration file (e.g., ${X}): the value of this input field is transferred to the BPE client as it is given in the input field including all system variables. In consequence, a system variable must be defined in the login procedure before the BPE client is started on the remote host.

The Temporary User Environment

One remark concerning the temporary area ${T} and the re–definition of the variable ${U} should be added. Bernese programs (the menu program and the Fortran processing programs) expect to have their input files in the directory ${U}/PAN, some auxiliary files go into the directory ${U}/WORK, etc. The BPE must ensure that each user script has its corresponding input files in directory ${U}/PAN and that there are no interferences between the files belonging to two different scripts. Therefore the BPE client ${BPE}/RUNBPE.pm performs the following steps:

1. A new subdirectory is created in the ${T} directory. It has the name BPE_*pcfname_port_year_session_PID_SUB_ PID* that is sufficiently unique. Subsequently three subdirectories are created in this new directory: PAN, INP, and WORK
2. The input files are copied from the ${U}/OPT/xxx directory (xxx is the option directory specified in the PCF for each script) into the new PAN subdirectory.
3. The ${U} variable is redefined in such a way that

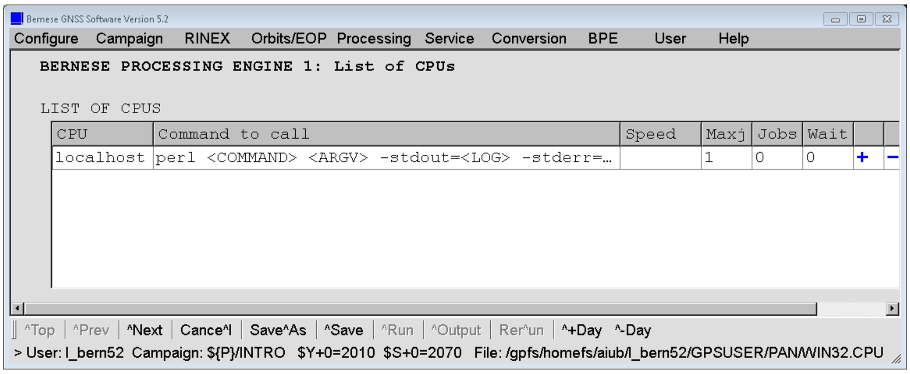
   ${U} = ${T}/BPE_*pcfname_port_year_session_PID_SUB_ PID*
4. The user script is started.
5. Before the client terminates it deletes the (previously created) temporary directory tree ${U} unless the "Do not remove temporary user environment" option is specified in panel "RUNBPE 2: Process Control Options", "Menu>BPE>Start BPE processing".

22.4. CPU Control File

The CPU control file defines which hosts (CPUs) are available and how the clients are started on different locations. CPU control files are located in the ${U}/PAN directory (with default extension CPU). They may be edited through "Menu>BPE>Edit CPU file".

Two different examples for CPU control files are listed in Figure 22.3. The column "CPU" contains a "nickname" for the CPU (may also be named host or queue). This name will be reported in the protocol file of the BPE as the CPU on which a script has been running. One may specify this name in the PCF to force a script to this particular queue.

In column "Command to call" the user has to specify how the client is started on the given host. Three reserved strings are expanded before performing the command:

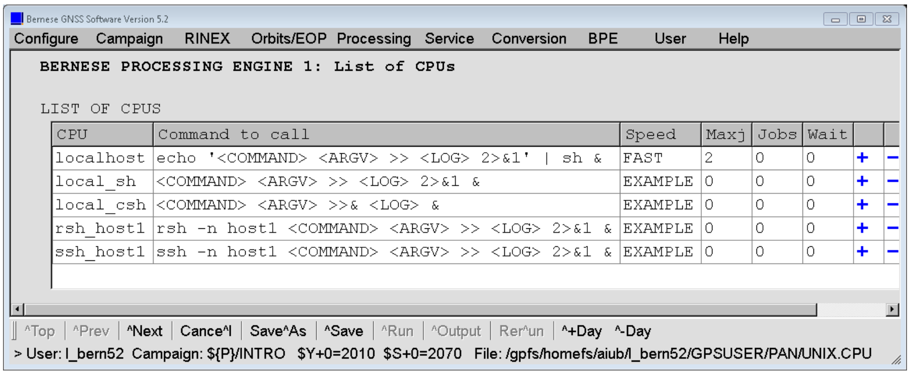

Bernese GNSS Software Version 5.2
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un ^+Day ^-Day

BERNESE PROCESSING ENGINE 1: List of CPUs

LIST OF CPUS

CPU	Command to call	Speed	Maxj	Jobs	Wait		
localhost	perl <COMMAND> <ARGV> -stdout=<LOG> -stderr=…		1	0	0	+	−

> User: l_bern52 Campaign: ${P}/INTRO $Y+0=2010 $S+0=2070 File: /gpfs/homefs/aiub/l_bern52/GPSUSER/PAN/WIN32.CPU


Bernese GNSS Software Version 5.2
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un ^+Day ^-Day

BERNESE PROCESSING ENGINE 1: List of CPUs

LIST OF CPUS

CPU	Command to call	Speed	Maxj	Jobs	Wait		
localhost	echo '<COMMAND> <ARGV> >> <LOG> 2>&1' | sh &	FAST	2	0	0	+	−
local_sh	<COMMAND> <ARGV> >> <LOG> 2>&1 &	EXAMPLE	0	0	0	+	−
local_csh	<COMMAND> <ARGV> >>& <LOG> &	EXAMPLE	0	0	0	+	−
rsh_host1	rsh -n host1 <COMMAND> <ARGV> >> <LOG> 2>&1 &	EXAMPLE	0	0	0	+	−
ssh_host1	ssh -n host1 <COMMAND> <ARGV> >> <LOG> 2>&1 &	EXAMPLE	0	0	0	+	−

> User: l_bern52 Campaign: ${P}/INTRO $Y+0=2010 $S+0=2070 File: /gpfs/homefs/aiub/l_bern52/GPSUSER/PAN/UNIX.CPU

Figure 22.3.: CPU control file for Windows (top) and UNIX (bottom) platforms.

`<COMMAND>` expands into the name of the client script with full path (i.e. `${BPE}/RUNBPE.pm`)[2],

`<ARGV>` expands into the list of arguments of the `RUNBPE.pm` script consisting of:

- Server host name (e.g., “localhost”, “ubecx.unibe.ch”, etc.) — the BPE server gets its host name from the system variable `BPE_SERVER_HOST`.
- Port on which the server is listening. An appropriate port is automatically selected by the BPE server itself.
- PID — ID of the user script to be executed by this particular client (as defined in the PCF — see Section 22.5).
- SUB_PID — sub–ID of a parallel script which shall run by the client (for all non–parallel scripts it has the value “000”).

`<LOG>` expands into the fully specified name of the BPE log file where all standard and error output of the script running on this client shall be collected.

It is the user’s responsibility that the script can be executed on a given host using the command defined in the CPU control file. In particular the use of the remote–shell (`rsh`) or secure–shell (`ssh`) commands on a UNIX system may need the assistance of the system administrator. To test if job submission using these commands is possible, the user can type the simple command rsh hostname ls or ssh hostname ls, respectively. It is important that the command works in background (which means that authentication in case of remote–shell or secure–shell must be done without any interaction). We refer to the corresponding man pages on how to use the remote (or secure) shell utility.

[2] The system environment variable `${BPE}` is expanded on the host where the client is started from. In consequence you have to make sure that on each remote host this file is found with this name, if needed, symbolic or hard links may help.

Of course, if the client is to be started on the same host as the BPE server, no remote (or secure) shell is necessary. Here is an example of the command that starts the client on a local machine:

```
  perl <COMMAND> <ARGV> --stdout=<LOG> --stderr=<LOG> &
```

The switches --stdout and --stderr re–direct the standard output and the error output to the BPE log–file. They are handled by the BPE client and the syntax works on all (Windows and UNIX) platforms. Of course it is assumed that the Perl interpreter is correctly installed and accessible through the path where the operating systems searches for executables. On UNIX systems it is important to ensure that the client is started in the background. This is done by the trailing & . On Windows the ampersand is not mandatory.

The string in the column "Speed" in the CPU control file defines a group–name of the corresponding CPU. If this string is specified as the CPU name for a script in the PCF, the script will run on one of the CPUs of this group only. Any CPU group name is allowed (except ANY or IDLE, see Section 22.5).

The field "Maxj" contains the maximum number of jobs allowed to run on the corresponding CPU simultaneously. The number of currently running jobs is counted in the ${U}/WORK environment (specified in "Scratch for CPU control file" in panel "MENU_EXT 10: Bernese Processing Engine (BPE)" accessible via "Menu>Configure>Paths and extensions"). Each time the BPE server starts a new job it creates a file

${U}/WORK/CPU_*cpufile*/*cpu*.*port_pcfname_year_session_PID_SUB_ PID*

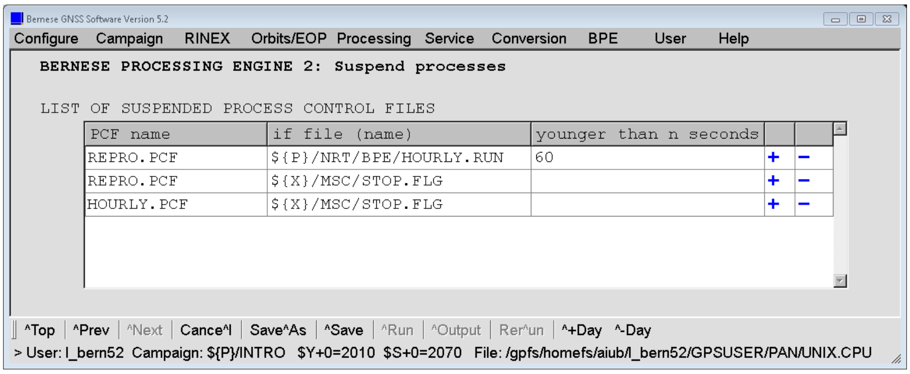
(where *cpu* is the nickname of the CPU). The file is removed by the BPE server if the job has been finished. The number of existing files starting with a specific CPU name corresponds to the number of currently running jobs on this CPU. Because of this extensive access to this directory when a BPE is running it might be worth to relocate this directory to a location with a quick disk access (e.g., a /scratch/local–directory).

The field "Jobs" has no meaning anymore but is still present for backward compatibility with older versions.

The field "Wait" requires an explanation. The BPE server starts a client script on the remote CPU to run a BPE script. When the client has become "alive" on the remote host it opens a TCP/IP connection to the server. If the client does not establish the TCP/IP connection within the time interval (in seconds) specified in the field "Wait" , the BPE server will stop with an error. It may then be assumed that, e.g., the remote host is not alive or the network connection does not work correctly. A blank field is equivalent to an infinite waiting time, which is needed, e.g., if the client scripts are launched via a job queuing system.

Resetting a CPU Control File

The function "Menu>BPE>Reset CPU file" allows to reset the statistics on the currently running jobs of a CPU control file to zero. This is simply done by removing all files from the directory ${U}/WORK/CPU_*cpufile*. This is particularly important if a previously executed BPE server terminated with a fatal error or has been killed. The CPU file may even be reset without consequences while a BPE is running (apart from potentially additional jobs exceeding the limit from the field "Maxj"). Use the reset–button if a BPE seems to wait indefinitely for an unknown reason.



BERNESE PROCESSING ENGINE 2: Suspend processes

LIST OF SUSPENDED PROCESS CONTROL FILES

PCF name	if file (name)	younger than n seconds
REPRO.PCF	${P}/NRT/BPE/HOURLY.RUN	60
REPRO.PCF	${X}/MSC/STOP.FLG	
HOURLY.PCF	${X}/MSC/STOP.FLG	

> User: l_bern52 Campaign: ${P}/INTRO $Y+0=2010 $S+0=2070 File: /gpfs/homefs/aiub/l_bern52/GPSUSER/PAN/UNIX.CPU

Figure 22.4.: Process suspension in a CPU control file.

Suspending BPE Processing

Option "List of suspended PCF" in "BERNESE PROCESSING ENGINE 2: Suspend processes" allows to prevent a specific PCF from starting new clients. However, this does not terminate the currently running clients.

The full `PCF` file name (e.g., RNX2SNX.PCF) must be specified to put the corresponding BPE in a "sleeping mode". This may be useful to halt a BPE in order to make the CPU resources available for a higher priority BPE run. As an advanced feature, the suspension of a PCF may depend on the existence (field "if file (name)") and age (field "younger than n seconds") of a certain file.

In this way, reprocessing activities may share the same computer system with near real–time applications: as long as the near real–time application is running, it regularly updates the BPE server status file (default extension `RUN`, see Section 22.10.1, e.g., `${P}/NRT/BPE/HOURLY.RUN` in the example of Figure 22.4) every 5 seconds. If this file is specified as the suspension condition for the reprocessing–related PCF (`REPRO.PCF` in Figure 22.4) with an age of more than 1 minute this BPE stops issuing further clients as long as the near real–time BPE is running. Another application is the use of "STOP"–files (`${X}/MSC/STOP.FLG` in Figure 22.4) to prevent jobs from running. In this way, a maintenance window can be created where no BPE script is executed to allow updating the source code or changes in station related files — of course all PCFs need to be listed as shown for two examples in Figure 22.4 .

22.5. Process Control File (PCF)

The Process Control File (PCF) defines which scripts should run and in what order they should be executed. In addition to listing the scripts to run, the PCF

- defines which scripts must wait for other scripts before they can run,
- defines parameters that are to be passed into the scripts,
- the CPU to be used by the client, and
- the option panels to be used by the Bernese programs executed in the scripts.

The Process Control Files are ASCII files. Figure 22.5 shows one of the example PCFs delivered with the *Bernese GNSS Software* Version 5.2 . The file consists of three sections:

```
#
# =============================================================================
# BASTST.PCF
# =============================================================================
#
# Purpose:      Does GPSEST runs baseline by baseline for trouble shooting.
#
# -----------------------------------------------------------------------------
#
# A description of the BPE, the use of the BPE server variables, input and
# output files as well as the important points to be checked in the processing
# summary are provided in
#              ${U}/PCF/BASTST.README       (UNIX/Linux)
#                %U%\PCF\BASTST.README     (Windows)
#
# -----------------------------------------------------------------------------
#
# Author     :  R. Dach
# Created    :  10-Aug-2011               Last modified: 10-Aug-2011
#
# =============================================================================
#
#
PID SCRIPT   OPT_DIR  CAMPAIGN CPU      P WAIT FOR....
3** 8******* 8******* 8******* 8******* 1 3** 3** 3** 3** 3** 3** 3** 3** 3** 3**
#
# Compute baseline solutions and extract relevant information
# ------------------------------------------------------------
101 GPSESTAP BAS_GEN           ANY      1
102 GPSEST_P BAS_GEN           ANY      1 101
103 GPSXTR   BAS_GEN           ANY      1 102
111 RESRMS   BAS_GEN           ANY      1 102
112 RESCHK   BAS_GEN           ANY      1 111
#
# Create summary file and delete files
# -------------------------------------
901 BAS_SUM  BAS_GEN           ANY      1 103 112
991 BPE_CLN  PPP_GEN           ANY      1 901
#
# End of BPE
# ----------
999 DUMMY    NO_OPT            ANY      1 991
#
#
PID USER         PASSWORD PARAM1   PARAM2   PARAM3   PARAM4   PARAM5   PARAM6   PARAM7   ...
3** 12********** 8******* 8******* 8******* 8******* 8******* 8******* 8******* 8******* ...
#
# Compute baseline solutions and extract relevant information
# ------------------------------------------------------------
101                       $101     V_A
102                       PARALLEL $101
#
# Create summary file and delete files
# -------------------------------------
#
#
#
VARIABLE DESCRIPTION                              DEFAULT
8******* 40************************************** 30****************************
V_A      Output and residual statistics           BAS
V_CRDINF Abbreviation filename                    EXAMPLE
V_APRFIL A priori (coordinate) information        APR$YSS+0
V_ORBFIL Orbit/ERP information                    COD$YSS+0
V_BLQINF BLQ FILE NAME, CMC CORRECTIONS           EXAMPLE
V_ATLINF ATL FILE NAME, CMC CORRECTIONS           EXAMPLE
V_HOIFIL Ionosphere model for higher order iono   HOI$YSS+0
V_SATSYS Select the GNSS (GPS, GPS/GLO)           GPS/GLO
V_SAMPL  Sampling interval                        30
V_PCV    Absolute/relative PCV model              I08
V_SATINF Satellite information file               SATELLIT
V_PCVINF PCV information file                     PCV
V_SATCRX Satellite problem file                   SAT_$Y+0
#
# DO NOT USE V_D, V_J, V_M, V_Y VARIABLES!
# (they are used already by the menu)
#
```

Figure 22.5.: Example PCF file BASTST.PCF.

1. Section defining the sequence of the scripts, option directories and CPUs to be used, and wait conditions (starting with header string PID SCRIPT, see description below).
2. Section defining parameters and special actions (starting with header string PID USER).
3. Section defining values for variables passed to the scripts (starting with header string VARIABLE).

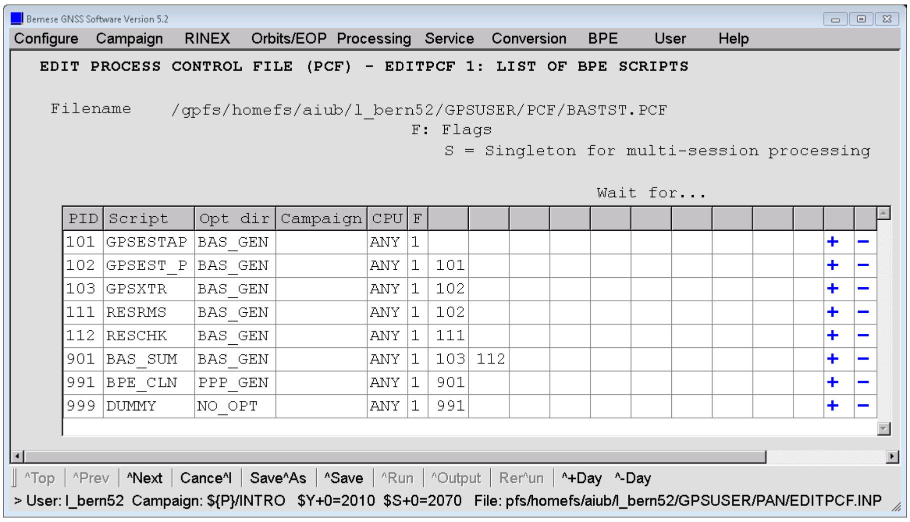
As you can see in the example of Figure 22.5 comments may be added to the PCF in any place, even between a sequence of scripts. Comment lines have to start with a hash–character (#).

The PCF files may be edited in any ASCII editor. Alternatively, you may edit them using the edit facility provided by the menu item "Menu>BPE>Edit process control file (PCF)". Note, that in the second case you do not have the possibility to add comment lines.

22.5.1. General Definition of the Scripts

Figure 22.6 shows the first panel that appears when the user starts to edit a PCF in "Menu>BPE>Edit process control file (PCF)" (same PCF as displayed in Figure 22.5):

- The user has to specify the names of the scripts in column "Script" and has to assign a unique PID (Process Identifier) number to each script.
- The column "Opt_dir" contains the name of the option directory that will be used to get input files for the processing programs and the menu configuration. The option directories must reside in ${U}/OPT.
- Column "Campaign" may contain the name of the campaign where the script has to be processed. Normally this field is left blank to run the script in the active campaign of the BPE. This allows to process any campaign using the same PCF. The field is intended to execute individual scripts in different campaigns.



EDIT PROCESS CONTROL FILE (PCF) - EDITPCF 1: LIST OF BPE SCRIPTS

Filename /gpfs/homefs/aiub/l_bern52/GPSUSER/PCF/BASTST.PCF

F: Flags
S = Singleton for multi-session processing

Wait for...

PID	Script	Opt dir	Campaign	CPU	F												
101	GPSESTAP	BAS_GEN		ANY	1											+	−
102	GPSEST_P	BAS_GEN		ANY	1	101										+	−
103	GPSXTR	BAS_GEN		ANY	1	102										+	−
111	RESRMS	BAS_GEN		ANY	1	102										+	−
112	RESCHK	BAS_GEN		ANY	1	111										+	−
901	BAS_SUM	BAS_GEN		ANY	1	103	112									+	−
991	BPE_CLN	PPP_GEN		ANY	1	901										+	−
999	DUMMY	NO_OPT		ANY	1	991										+	−

> User: l_bern52 Campaign: ${P}/INTRO $Y+0=2010 $S+0=2070 File: pfs/homefs/aiub/l_bern52/GPSUSER/PAN/EDITPCF.INP

Figure 22.6.: Process Control File, definition of scripts.

- Column "CPU" is the name of the computer/queue to which the script should be submitted. It must correspond to the name in the CPU control file (nickname of the given CPU or the group–name — see Section 22.4). If the name ANY is used, then the script will be submitted to the first available computer found in the CPU list. Another possibility is IDLE which means that a computer/queue is chosen where no other BPE script is running yet.
- Column "F" is intended for flags. The only flag defined so far is S (Singleton), which indicates that the corresponding script is executed only once, even if the PCF is processed for several sessions.
- The remaining fields specify the PIDs of the scripts that must be finished before the script can be started. Up to ten entries are possible. If a script has to wait for more than ten scripts, a DUMMY script may be inserted which does nothing than wait for part of these scripts.

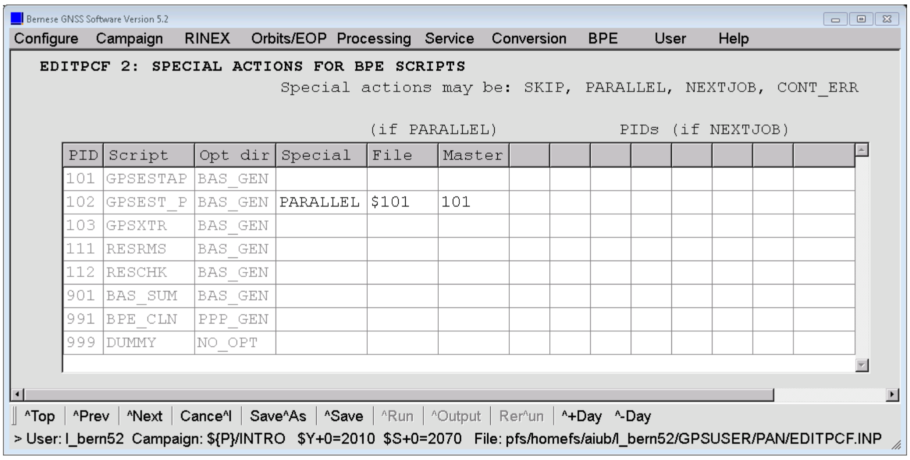
22.5.2. Parallel Processing

The basic PCF is linear, sequentially executed. A certain level of parallelization will happen if two different scripts may run at the same time due to appropriate wait conditions. In the example of Figure 22.6 the extraction of the GPSEST program output (GPSXTR, PID 103) and the creation of the residual summary table (RESRMS, PID 111) can both be started if the PID 102 (script GPSEST_P running GPSEST) has been finished. As far as two CPUs are free at that time, both scripts will run in parallel.

Parallel PCFs are slightly more complicated than linear PCFs. They have the advantage, however, that they can split up a single task into multiple tasks, each of which may be executed on a separate computer. For example, to run GPSEST for ambiguity resolution on all single–difference observation files, a BPE user script could be written that processes one baseline file after another executing GPSEST in a linear fashion. With parallel scripts, it is possible to divide up the observation files into groups and have different computers working on different groups of files. Improvements in speed can be achieved even by running two groups of observation files on the same computer. This is because while one group is using the disk I/O of the computer, the other can use the CPU and vice versa.

Parallel PCFs require pairs of scripts that work together in conjunction: a master and a slave script. The first script of each such pair is the *master script* and its job is usually (a) to create a list of files (e.g., observation files) to be processed one–by–one in parallel or (b) to divide up a list of files that are to be processed in groups. In both cases a *control file* is created. The number of lines in this file is equal to the number of slaves that should run in parallel. Each line contains the information required by the corresponding slave. In case (a) each line just contains the name of the file that should be processed. In case (b) it may contain all file names belonging to the group or a name of the file that contains the list of files belonging to the group (Section 22.6.4 provides a detailed description). Within the master script the name of the control file is accessible through the variable $$bpe{CONTROL_FILE}. The BPE defines the name of the control file automatically and it ensures that the name is unique.

The second script of the pair is the so–called *slave script* which actually does the work. The slave script is executed once for each line in the control file. Each slave script knows exactly one line from the control file. The content of this line is given in the variable $$bpe{CONTROL_FILE_LINE}.


Bernese GNSS Software Version 5.2
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un ^+Day ^-Day

EDITPCF 2: SPECIAL ACTIONS FOR BPE SCRIPTS

Special actions may be: SKIP, PARALLEL, NEXTJOB, CONT_ERR

(if PARALLEL) PIDs (if NEXTJOB)

PID	Script	Opt dir	Special	File	Master
101	GPSESTAP	BAS_GEN			
102	GPSEST_P	BAS_GEN	PARALLEL	$101	101
103	GPSXTR	BAS_GEN			
111	RESRMS	BAS_GEN			
112	RESCHK	BAS_GEN			
901	BAS_SUM	BAS_GEN			
991	BPE_CLN	PPP_GEN			
999	DUMMY	NO_OPT			

> User: l_bern52 Campaign: ${P}/INTRO $Y+0=2010 $S+0=2070 File: pfs/homefs/aiub/l_bern52/GPSUSER/PAN/EDITPCF.INP

Figure 22.7.: Process Control File, master and slave scripts.

Let us assume that among the scripts in the example of Figure 22.6 a parallel baseline–by–baseline processing shall be constructed. At first we have to define the master–and–slave pair: The script GPSESTAP (PID 101) will act as the master preparing parallel runs of GPSEST, the script GPSEST_P (PID 102) is the corresponding slave executing GPSEST. The script becomes a "slave" by specifying the value PARALLEL in the field "Special" in the panel "EDITPCF 2: SPECIAL ACTIONS FOR BPE SCRIPTS" (see Figure 22.7). In addition, the PID of the master script must be specified in field "Master". In our example the script 101 is the master of the script 102. The field "File" is currently not used but the menu program requires a non–blank string here due to the compatibility with the old versions of the BPE. The corresponding part of the second section in the PCF file then looks like:

```
PID USER         PASSWORD PARAM1   PARAM2   PARAM3   PARAM4   PARAM5   PARAM6   ...
3** 12********** 8******* 8******* 8******* 8******* 8******* 8******* 8******* ...
101                       $101
102                       PARALLEL $101
```

See Section 22.6.4 for parallel script examples. Perl utilities are provided with Version 5.2 to facilitate the writing of parallel user scripts. See Section 22.6.5.3 for more details or inspect the example scripts delivered with the software.

22.5.3. Loops and Jumps

The action NEXTJOB allows to change the sequence in which the scripts are executed. The corresponding script must decide which script will be submitted next. The two cases shall be illustrated with typical examples:

- A certain sequence of scripts shall be repeated.
  For instance, in case of automated updating of the list of fiducial stations for the datum definition, the final session solution with the program ADDNEQ2 needs to be repeated. Depending on the output of the program HELMR1 (responsible for the verification of the datum station list) a jump back needs to be initiated to repeat the generation of the final solution.

  Similar loops are needed if stations or satellites are identified as "misbehaving" and shall be excluded from the processing.

- A certain sequence of scripts shall be skipped.
  In case of an hourly processing scheme several hourly RINEX observation files shall be merged into one RINEX file per station. This sequence is not needed in case of a daily processing scheme and may be skipped depending on the session definition.

  As another example one can imagine that the protocols and the list of files to be copied from the campaign into the savedisk area may be different for an operational or reprocessing usage of a PCF . Depending on the mode, one or the other sequence may be skipped at the end of the BPE .

We recommend to store the PIDs of possible targets into the fields "PIDs (if NEXTJOB)" (which correspond to script parameters `PARAM2`, `PARAM3`, etc., see Section 22.5.5). The second section in the PCF may then look like:

```
PID USER         PASSWORD PARAM1   PARAM2   PARAM3   PARAM4   PARAM5   PARAM6   ...
3** 12********** 8******* 8******* 8******* 8******* 8******* 8******* 8******* ...
114                       NEXTJOB  100      100      120
```

The jump is achieved by calling the method `RUNBPE::PRT_GOTO pid` in the BPE user script, e.g., using the command `$bpe->PRT_GOTO($$bpe{PARAM2});` (see Section 22.6.5.4) in the user script which writes a message to the script's protocol file and initiates the jump. In case of a jump back, all running scripts between the current script and the target PID have to be finished before the script at the target PID is started. In case of a jump forward the script at the target PID is immediately started. Any necessary waiting conditions have to be specified by the user.

The BPE variable `$$bpe{STARTCOUNT}` keeps track of the number of times a user script has been started in the same BPE run as a consequence of backward jumps. It is useful to limit the number of loops (iterations).

In Section 22.6.3 an example script is displayed and in Section 22.6.5.4 an example is given on how to use the BPE utilities for writing user scripts performing jumps.

22.5.4. Further Special Actions: SKIP and CONT_ERR

In addition to `PARALLEL`, the user may specify three other special actions in the panel displayed in Figure 22.7. If action `SKIP` is specified, the script will not execute. In this way scripts may be disabled without having to renumber the `Wait for`–PIDs in the PCF. The second section in the PCF may then look like:

```
PID USER         PASSWORD PARAM1   PARAM2   PARAM3   PARAM4   PARAM5   PARAM6   ...
3** 12********** 8******* 8******* 8******* 8******* 8******* 8******* 8******* ...
200                       SKIP
```

If action `CONT_ERR` is specified, the BPE continues with the next script even if the script has finished with an error. This feature is intended for scripts where a failure does not harm the processing chain. It is activated in the second section of the PCF exactly like `SKIP`:

```
PID USER         PASSWORD PARAM1   PARAM2   PARAM3   PARAM4   PARAM5   PARAM6   ...
3** 12********** 8******* 8******* 8******* 8******* 8******* 8******* 8******* ...
200                       CONT_ERR
```

22.5.5. Script Parameters and BPE Variables

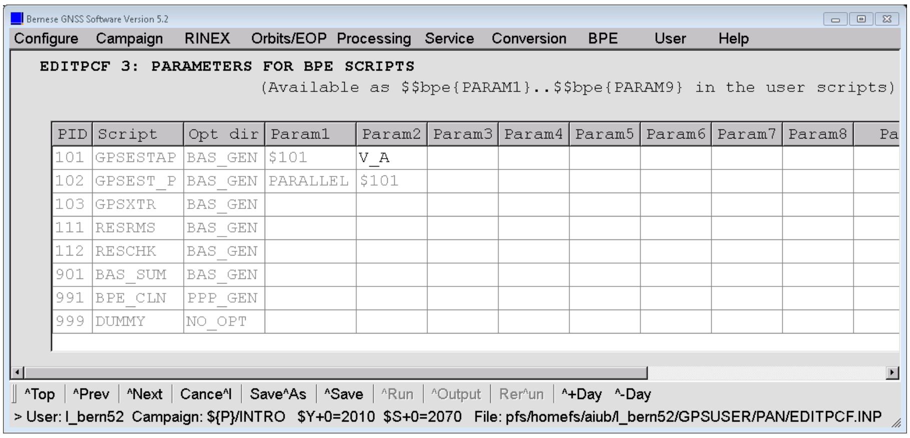
The remaining two panels define the parameters of the user scripts and the so–called BPE variables. The parameters can be specified in the panel shown in Figure 22.8. Up to nine parameters can be defined which are available in the Perl user scripts as `$$bpe{PARAM1}`, `$$bpe{PARAM2}`, etc. However, in case of one of the special actions described above (`PARALLEL`, `NEXT_JOB`, `SKIP`, or `CONT_ERR`), the parameter `$$bpe{PARAM1}` is reserved for the keyword indicating the action. The `PARAM1` field is also reserved in case of a master script for parallel processing to indicate the related slave script.

No PCF parameters can be defined for the slave scripts. The slave scripts should be given all necessary information through the variable `$$bpe{CONTROL_FILE_LINE}`. We have stated already that this variable contains exactly one line (different for each parallel running slave) of the control file. In addition to the information specific for the individual slave scripts (e.g., name of baseline to be processed), the master script may add the values of the BPE variables `PARAM2`, `PARAM3`, etc., to `$$bpe{CONTROL_FILE_LINE}` as blank–delimited list. These parameters are then available in the slave script as `$$bpe{PARAM2}`, `$$bpe{PARAM3}`, etc. See Section 22.6.4 for an example.

In the last panel (Figure 22.9) the BPE variables may be defined. These are additional variables that may be used in user scripts as well as in the program input files. The user may define the values of the BPE variables when executing the BPE. He may, e.g., specify a solution ID which is part of the result filenames, some general input files to be used by the processing programs, or even individual processing options to be inserted in one of the fields of the input panels.

In order to remain compatible with older version of the BPE, we still use the convention that BPE variables have to start with the string `V_`. They may be accessed in user scripts with the syntax `$$bpe{V_xxx}` or as menu variables in input fields of the program input files where the string `$(xxx)` has to be used for accessing the BPE variable `V_xxx`. BPE variables have precedence over menu variables defined in `MENU_VAR.INP` in the script's option directory. See Section 21.6 for details on menu variables and Section 22.6.2 for details on variables in user scripts.

Some variables have a special meaning: The special parameters `V_MINUS` and `V_PLUS` specify the ranges for session variables. If defined, their values are used instead of the

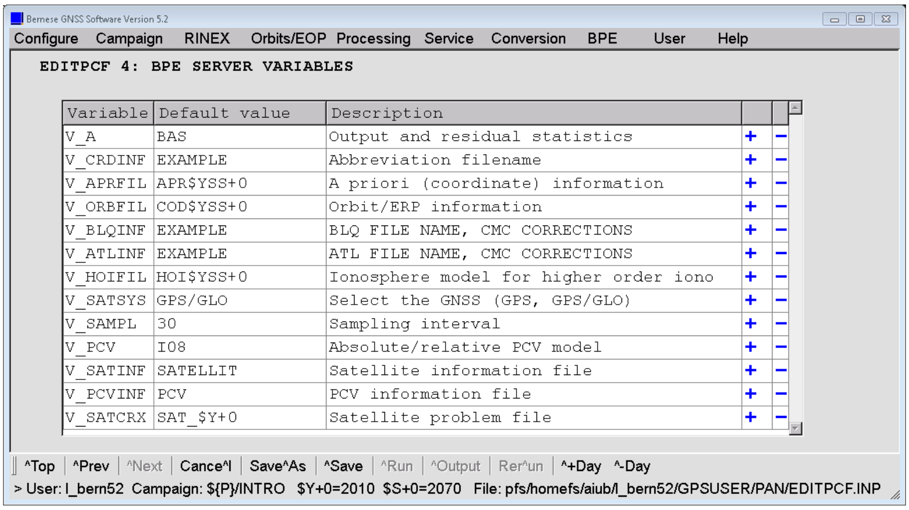

Bernese GNSS Software Version 5.2
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un ^+Day ^-Day

EDITPCF 3: PARAMETERS FOR BPE SCRIPTS
(Available as $$bpe{PARAM1}..$$bpe{PARAM9} in the user scripts)

PID	Script	Opt dir	Param1	Param2	Param3	Param4	Param5	Param6	Param7	Param8	Pa
101	GPSESTAP	BAS_GEN	$101	V_A							
102	GPSEST_P	BAS_GEN	PARALLEL	$101							
103	GPSXTR	BAS_GEN									
111	RESRMS	BAS_GEN									
112	RESCHK	BAS_GEN									
901	BAS_SUM	BAS_GEN									
991	BPE_CLN	PPP_GEN									
999	DUMMY	NO_OPT									

> User: l_bern52 Campaign: ${P}/INTRO $Y+0=2010 $S+0=2070 File: pfs/homefs/aiub/l_bern52/GPSUSER/PAN/EDITPCF.INP

Figure 22.8.: Process Control File, parameters for scripts.


Bernese GNSS Software Version 5.2
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un ^+Day ^-Day

EDITPCF 4: BPE SERVER VARIABLES

Variable	Default value	Description		
V_A	BAS	Output and residual statistics	+	−
V_CRDINF	EXAMPLE	Abbreviation filename	+	−
V_APRFIL	APR$YSS+0	A priori (coordinate) information	+	−
V_ORBFIL	COD$YSS+0	Orbit/ERP information	+	−
V_BLQINF	EXAMPLE	BLQ FILE NAME, CMC CORRECTIONS	+	−
V_ATLINF	EXAMPLE	ATL FILE NAME, CMC CORRECTIONS	+	−
V_HOIFIL	HOI$YSS+0	Ionosphere model for higher order iono	+	−
V_SATSYS	GPS/GLO	Select the GNSS (GPS, GPS/GLO)	+	−
V_SAMPL	30	Sampling interval	+	−
V_PCV	I08	Absolute/relative PCV model	+	−
V_SATINF	SATELLIT	Satellite information file	+	−
V_PCVINF	PCV	PCV information file	+	−
V_SATCRX	SAT_$Y+0	Satellite problem file	+	−

> User: l_bern52 Campaign: ${P}/INTRO $Y+0=2010 $S+0=2070 File: pfs/homefs/aiub/l_bern52/GPSUSER/PAN/EDITPCF.INP

Figure 22.9.: Process Control File, BPE variables.

values of the keywords `VAR_PLUS` and `VAR_MINUS` in file `MENU_VAR.INP` of the script's option directory. See Section 21.6 for details on menu range variables.

The variable `V_CAMP` may be used to define the campaign in which the BPE has to run. The BPE will stop if it is started in another campaign. If the variable is not specified, the BPE may run in any campaign. The variable `V_RERUN` is used to restart the user script in case of an error. The value of this variable specifies the number of additional attempts to run the script. This may make sense if the user script fails due to system or network problems. If this variable is not contained in the list (or has the value 0), no reruns are performed. The variables `V_D`, `V_J`, `V_M`, `V_Y` are reserved. Do not use them.

22.6. User Scripts

After installing the *Bernese GNSS Software* a couple of Perl scripts can be found in directory `${U}/SCRIPT`. These small programs are user scripts and complement the BPE examples delivered with the software. Most of them are very generic and may be used in a number of different BPEs. The user may take them as examples for the development of his own sophisticated user scripts when creating his own BPE procedures.

These scripts are started by the BPE–client and are responsible for running the main Bernese programs or performing other specific tasks such as copying or downloading files or executing auxiliary programs. On UNIX platforms the user scripts may be written in almost any programming language (actually the only requirement is that the environment variables are accessible from within the user script and some system commands can be performed). Old UNIX user scripts were, e.g., written in Bourne shell and may still be used (with a few modifications, see Section 26.2.2).

However, we recommend to use Perl for writing new scripts. Scripts written in Perl may profit from inherited methods defined in the BPE client `${BPE}/RUNBPE.pm` and efficiency is increased because the user scripts do not need to be executed as a separate child process of the client. On Windows platforms the use of Perl as scripting language is required.

DOS batch files are no longer supported. The “skeleton” of Perl user scripts is very simple as shown in the following section.

22.6.1. User Script Skeleton

From the user point of view the most important method of the RUNBPE object (that is defined in ${BPE}/RUNBPE.pm) is the method run. This method is responsible for starting the *user script*. In the terminology of object–oriented programming the method should be looked at as virtual. Therefore, it is possible to use the technique of method overloading. A user script written in Perl inherits its behavior from the RUNBPE object and overrides the RUNBPE::run method in such a way that it directly fulfills the required task.

Let us start with a trivial example of a user script. We call this script DUMMY and it does nothing else but directly exiting. The implementation of the script is given here (of course, the line numbers are not a part of the script):

```
1 package DUMMY;
2 @ISA = ("RUNBPE");
3 use strict;
4 sub run {
5   my $bpe = shift;
6
7 }
```

The first line is the name of the script. The second line says that the behavior of the new package DUMMY (or object DUMMY) is inherited from the package RUNBPE. Line 3 requires explicit declaration of all variables (and is included for the sake of a good Perl programming style). Lines 4–7 redefine the method run. In Perl object–oriented style the first parameter of each method is the pointer to the object itself. Line 5 stores this pointer into the variable $bpe (using the name $bpe is a useful convention but not a strict demand). The body of the script doing the actual work would be specified in lines 6 and following.

Of course, real scripts will have to perform more complicated tasks (e.g., running a Bernese program). However, the lines 1–5 from the example above may remain without any changes. Before we proceed with a more sophisticated example we have to make a remark concerning the variables that are accessible in user scripts.

22.6.2. Variables in User Scripts

All information required by the user scripts is available in form of system variables. This means that, e.g., old scripts written in Bourne shell can directly use the system variable $U and the Perl scripts could use the corresponding syntax $ENV{U}. However, new Perl scripts that inherit the run method from the RUNBPE object should refer to the environment variables through the object reference $bpe (see the example in the previous section). There are two possibilities:

$$bpe{variable_name} or $bpe–>{variable_name} .

Both syntax are equivalent. The getKeys method from RUNBPE provides a general mechanism to access variables, see Section 22.6.5.1.

All the variables available in user scripts may be divided into three categories:

BPE_CLIENT	$BPE/RUNBPE.pm	PARAM1	
BPE_SERVER_HOST	carina2.unibe.ch	PARAM2	
CAMPAIGN	INTRO	PARAM3	
CAMP_DRV	$P/	PARAM4	
CAMP_PTH	/home/bernese/GPSDATA/CAMPAIGN52/	PARAM5	
PTH_BPELOG	/home/bernese/GPSDATA/CAMPAIGN52/INTRO/BPE/	PARAM6	
PTH_BPEPRT	/home/bernese/GPSDATA/CAMPAIGN52/INTRO/BPE/	PARAM7	
CLIENT_ENV	$X/EXE/LOADGPS.setvar	PARAM8	
COMMAND	START	PARAM9	
CPU	localhost	PID	113
CPUFIL	/home/bernese/GPSUSER52/PAN/UNIX.CPU	PORT	36915
S_PTH_OPT	/home/bernese/GPSUSER52/OPT	REQ_CPU	ANY
S_PTH_SCRIPT	/home/bernese/GPSUSER52/SCRIPT	SCRIPT	ORBGENH
DAY	26	SESSID	0
DAYWEEK	1	SESSION	2070
DAYYEAR	207	SESSION_TABLE	SESSIONS.SES
DEBUG	1	SINGLETON	N
EXT_BPELOG	LOG	STARTCOUNT	1
EXT_BPEPRT	PRT	SUB_PID	000
GPSWEEK	1594	TASKID	NN
MJD	55403	USER	bernese
MONTH	07	WAIT_FOR	112
NOCLEAN	0	YEAR	10
OPT_DIR	R2S_GEN	YR_4	2010
PCFFIL	RNX2SNX.PCF		

Figure 22.10.: Server–defined client variables.

1) Environment Variables

The first category contains variables defined in LOADGPS.setvar file (on UNIX platforms — see Section 22.3.3) or listed in the special variable BERNESE_VARIABLES (under Windows). The user may arbitrarily expand the list of these variables, however at least the following variables should always be included: ${BPE}, ${C}, ${T}, ${U}, ${X}, ${XG}, ${XQ} (remember, however, that in the BPE the variable ${U} has been re–defined, its original value is accessible by $$bpe{U_OLD}).

In general, environment variables should be accessed in Perl user scripts using the methods $$bpe{variable_name} or $bpe–>{variable_name} instead of $ENV{U}. It may, e.g., be that the client has started on a host where the Bernese environment variables are not available. On UNIX systems the client gets the name of the LOADGPS.setvar file from the server, extracts the variable names from there and inserts them into the client's namespace without defining them as system environment variables.

2) Client Variables

The BPE server sends a list of variables and their values to each client when it commands the client to start the user script. Table 22.10 gives a list of the variables together with their values stemming from a run of script ORBGEN in the example BPE RNX2SNX.PCF. Note, that the script was neither master nor a slave and therefore the variables $CONTROL_FILE and $CONTROL_FILE_LINE are not defined.

Additional variables are defined and available as client variables. An interesting variable is $$bpe{SYSOUT} which contains the name of the program output file of the last program executed in a Perl user script using the run method. The script may then, e.g., append the file to a protocol file.

In addition to the variables listed in Figure 22.10, the plus–minus time variables such as DAYMP1 (listed on page 380f of the documentation of Version 4.2) are still available for backward compatibility. It is, however, not recommended to use these variables. Use the method RUNBPE::getKeys instead (see Section 22.6.5.1).

3) User–defined Server Variables

The user may define additional variables that are sent to the client together with the general variables and the standard server variables. For compatibility with the previous Versions 4.x of the BPE, the variable names defined in process control files must start with V_ .

There are three ways how to define these variables:

1. Define the variable directly in the last section of the PCF (see Section 22.5.5).
2. Define the variables in "Menu>BPE>Start BPE processing". The menu program works in such a way that variables defined in the PCF are displayed and the user may edit the list of variables arbitrarily. The entered values remain defined until another PCF is edited. The values entered here have precedence over the values defined in the PCF.
3. The variables may be defined when starting the BPE in non–interactive mode (see Section 22.8.2).

Access the variables in user scripts by $$bpe{V_xxx}. The user–defined server variables may also be used in panels when preparing the options for the BPE process in the same way as the user–variables (defined in "Menu>Configure>Menu variables", see Section 21.6.2). The only difference is that the prefix V_ has to be stripped from the variable name. I.e., use $(xxx) for accessing the variable V_xxx in a program panel input field. For widgets with a limited input capability (spinboxes and comboboxes) the corresponding keyword, preceded by V_, may be used like a user–defined server variable

Thanks to the metacommand keepvariables = true added to the option "BPE server variables" in panel "RUNBPE 4: Server Variables Sent to Client" ("Menu>BPE>Start BPE processing"), the BPE server variables may even contain time variables that are translated only later on for each individual session that is processed. For example, a variable V_SATCRUX may be defined with the value SAT_$Y+0 . Independent from the session defined when the BPE was started for a multi–year processing, the time variable $Y+0 is translated with the correct year for each individual session that is processed.

22.6.3. Running Bernese Programs in User Scripts

The following example shows a very simple script that executes the Bernese program COOVEL. Again the line numbers are not a part of the script:

```
package COOVEL;
@ISA = ("RUNBPE");
# ============================================================================
# Name     :  COOVEL
# Purpose  :  Execute COOVEL
# ============================================================================
use strict;
sub run {
  my $bpe = shift;

# Run program
# -----------
  my $PGMNAM = "COOVEL";
  $bpe->RUN_PGMS($PGMNAM);
}
```

The meaning of the lines 1–2 and 7–9 was already described in Section 22.6.1. Let us look at the lines 13 and 14: In this way the Bernese main program is started. The format of these two lines is mandatory to allow the menu program to find all programs that are started in a given user script in "Menu>BPE>Edit PCF program input files" (see Section 22.7). The method RUNBPE::RUN_PGMS is defined in the module `${BPE}/RUNBPE.pm`.

The method RUNBPE::RUN_PGMS executes the following steps:

1. Makes sure that the "Error message" is written into a separate file but not merged into the "Program output" .
2. Copies the program input file from `${U}/PAN` to `${U}/INP` while updating the active campaign, the current session as well as all relevant menu variables.
3. Executes the related program using the options given in the corresponding input file prepared in `${U}/INP` in the previous step.
4. Posts warning or error messages into the protocol file.
5. Defines the BPE server variable SYSOUT containing the name of the program output file for further use.
6. Stops the user script if the program reported an error, or returns to the next step of the user script if the program was successfully executed.

The following example shows a more complex user script available in the example BPE. The name of the script is `HELMCHK`. It verifies the list of fiducial stations to be used for the datum definition (see Section 10.2.3). For this purpose the Bernese main program HELMR1 is executed. If stations with big residuals are listed in a specific file, the solution is repeated with an updated list of fiducial sites.

```perl
package HELMCHK;
@ISA = ("RUNBPE");
# ====================================================================
# Name     :  HELMCHK
# Purpose  :  Execute HELMR1 to verify the datum definition
# PARAMs   :  PARAM2 - goto PID for new datum definition
# ====================================================================
use strict;

use File::Copy;

use lib "$ENV{BPE}";
use bpe_util qw (prtMess prtGoto);

sub run {
  my $bpe = shift;

# Get variables
# -------------
  my ($startCount,$param2) =
  $bpe->getKeys
     ('STARTCOUNT','PARAM2');

# Outlier rejection only in the first iteration
# ---------------------------------------------
  if ($startCount > 1) {
    prtMess($bpe,"Outlier␣rejection␣disabled");
    $bpe->putKey("$ENV{U}/PAN/HELMR1.INP","REJECT","0");
 }

# Run program
# -----------
  my $PGMNAM = "HELMR1";
  $bpe->RUN_PGMS($PGMNAM);

# Copy the result from the outlier rejection, jump back (if needed)
# -----------------------------------------------------------------
  if ($startCount == 1) {
    my $listFil = $bpe->getKey("$ENV{U}/INP/HELMR1.INP","LISTFIL");
    if ( -e $listFil ) {
      prtGoto($bpe,$param2);
    }
 }
}
```

The program HELMR1 is started in lines 33–34. With method $bpe–>putKey (line 28) a value is assigned to the specified input panel and keyword. Its complement is method $bpe–>getKey (line 39) where the value of a keyword is extracted. It is important to understand the difference in the path of the HELMR1.INP file: setting a keyword before the program is executed has to be done in ${U}/PAN, whereas the extraction of values typically has to be done after updating the campaign and session by the menu program within the RUN_PGMS method — it means in ${U}/INP.

In contrast to $bpe–>getKey which extracts a value of a keyword in a specific input file (typically a program input file), the method $bpe–>getKeys (called in line 21) takes a list of keywords related to the menu configuration or BPE variables. In most cases the values are assigned to user script variables for the further use within the script (see Section 22.6.5.1).

In lines 27 and 41 two BPE utility functions are called (see Section 22.6.5.5 and Section 22.6.5.4). The corresponding module is loaded in lines 12–13. The first function writes a message into the BPE protocol file and the second function initiates a jump of the BPE processing sequence with the action NEXTJOB to the PID specified by PARAM2.

22.6.4. Parallel User Scripts

The examples in this section show a master and slave user script for parallel processing. The first example demonstrates the preparation of a baseline–wise processing, e.g., for ambiguity resolution:

```
 1 use bpe_util qw(initPar_Bl);
 2
 3 sub run{
 4   my $bpe = shift;
 5
 6 # Get variables
 7 # -------------
 8   my           ( $dirPSH,  $extPSH, $sess, ) =
 9   $bpe->getKeys('DIR_PSH','EXT_PSH','$S+0' );
10   ...
11
12 # Get the list of files to be processed
13 # -------------------------------------
14   my @basLin = ();
15   foreach my $bslFil ( glob("${dirPSH}????${sess}.${extPSH}") ) {
16     push @basLin, substr(basename($bslFil),0,4);
17  }
18
19 # Initialize parallel run
20 # -----------------------
21   initPar_Bl($bpe,@basLin);
```

The lines 14 to 17 assign the four character IDs from all baselines of a certain session to the variable @basLin. In line 21 the BPE utility function initPar_Bl is called with the array @basLin containing the filenames of the baselines. It automatically creates a control file from the entries of @basLin.

In fact, the master scripts for the ambiguity resolution in the distributed processing examples (e.g., GNSQIFAP) do something similar. Instead of directly selecting all observation files of a session (see line 14 of the example above), they use the Bernese program BASLST to select the baselines according to their length:

```
 1   ...
 2 # Run program
 3 # -----------
 4   my $PGMNAM = "BASLST";
 5   $bpe->RUN_PGMS($PGMNAM);
 6
 7 # Get baselines to be processed
 8 # ----------------------------
 9   my $basFil = $bpe->getKey("$ENV{U}/INP/BASLST.INP","SELBSL");
10
11   my @basLin = ();
12   open (BAS,$basFil);
13   push (@basLin,<BAS>);
14   close BAS;
15   chomp(@basLin);
16
17 # Initialize parallel run
18 # -----------------------
19   initPar_Bl($bpe,@basLin);
```

In line 5 BASLST is executed and the list of four character abbreviations for the baselines is created. In lines 11–15 the filenames of the baselines are read from the output file and stored into the array @basLin. With the function initPar_Bl the control file is created here too. It looks like:

```
GAJO
HEZI
JOLA
JOON
JOWS
JOWT
MAZI
PTWZ
TLZI
WTWZ
WTZI
ZIZM
```

The content of this file is automatically posted from the parallel master script to the BPE server to initialize twelve parallel runs of the slave script GNSQIF_P. Each parallel running slave script gets one line of the control file. The baseline abbreviation, e.g., GAJO is assigned to the parameter $$bpe{PARAM1}. An extract of the relevant lines of GNSQIF_P follows:

```
 1 use bpe_util qw(setUserVar);
 2
 3 sub run{
 4   my $bpe = shift;
 5
 6 # Get variables
 7 # -------------
 8  my $param1 = $bpe->getKeys('PARAM1');
 9
10 # Set filename variable
11 # ---------------------
12   setUserVar($bpe,"FFFF",$param1);
13
14 # Run program
15 # -----------
16   my $PGMNAM = "GPSEST";
17   $bpe->RUN_PGMS($PGMNAM);
```

In line 8 the baseline abbreviation (e.g., GAJO) is read from $$bpe{PARAM1} and assigned to the local variable $param1. In line 12 a user variable FFFF is created using the BPE utility function setUserVar and is assigned to the baseline abbreviation. In all program panels the baseline abbreviation is thus available through the variable $(FFFF). That means in the panel "GPSEST 1.1: Input Files 1" the name of the phase file to be processed has to be specified with the string $(FFFF)$S+0 which is independent of the particular baseline. Finally in line 17 GPSEST is run to process the specific baseline.

In a similar way the master script could group files to clusters, if the computing resources available for parallel processing are limited. For each cluster the individual filenames are written to a selection file by the BPE utility function initPar_Cl (e.g., with name `OBSssssnnn.BPE` in the campaign's `BPE` directory with `ssss` being the session and `nnn` being the cluster number). The names of these cluster–specific files are written as separate lines to a control file (note that `${P}` will be replaced by the path to the campaign in practice):

The control file contains:

```
${P}/EXAMPLE/BPE/OBS2080001.BPE
${P}/EXAMPLE/BPE/OBS2080002.BPE
```

Cluster file
`${P}/EXAMPLE/BPE/OBS2080001.BPE`:

```
${P}/EXAMPLE/OBS/GAJO2080.PSH
${P}/EXAMPLE/OBS/HEZI2080.PSH
${P}/EXAMPLE/OBS/JOLA2080.PSH
${P}/EXAMPLE/OBS/JOON2080.PSH
${P}/EXAMPLE/OBS/JOWS2080.PSH
${P}/EXAMPLE/OBS/JOWT2080.PSH
```

Cluster file
`${P}/EXAMPLE/BPE/OBS2080002.BPE`:

```
${P}/EXAMPLE/OBS/MAZI2080.PSH
${P}/EXAMPLE/OBS/PTWZ2080.PSH
${P}/EXAMPLE/OBS/TLZI2080.PSH
${P}/EXAMPLE/OBS/WTWZ2080.PSH
${P}/EXAMPLE/OBS/WTZI2080.PSH
${P}/EXAMPLE/OBS/ZIZM2080.PSH
```

The slave script, in turn, gets the name of one selection file. It may then put the list of files from that file as file selection list into the program panel using the BPE utility function replaceSel (line 17):

```
 1 use bpe_util qw(setUserVar replaceSel);
 2
 3 sub run {
 4   my $bpe = shift;
 5
 6 # Get variables
 7 # -------------
 8   my ($ssss , $param1 , $subPid) =
 9   $bpe->getKeys
10      ('$S+0','PARAM1', 'SUB_PID');
11
12 # Set cluster number and input files
13 # ---------------------------------
14   my $cluster = "$ssss$subPid";
15   setUserVar($bpe,"CLUSTER",$cluster,"CCC",$subPid);
16
17   replaceSel($bpe,"GPSEST","PSFILES",$param1);
18
19 # Run program
20 # -----------
21   my $PGMNAM = "GPSEST";
22   $bpe->RUN_PGMS($PGMNAM);
```

The variables constructed in lines 14 and 15 are used in the input fields to define the output and result filenames. After these preparation steps, the program GPSEST is launched in line 22.

BPE parameters may be passed to the slave script either via the BPE utility functions initPar_Bl or initPar_Cl or directly by adding their values to `CONTROL_FILE` as a blank–delimited list:

```
  ...

# Get the list of files to be processed
# ------------------------------------
  my @basLin = ();
  foreach my $bslFil ( glob("${dirPSH}????${sess}.${extPSH}") ) {
    push @basLin, substr(basename($bslFil),0,4) . "␣PARAM2␣PARAM3";
 }

# Initialize parallel run
# -----------------------
  initPar_Bl($bpe,@basLin);
```

This results in a control file like

```
GAJO PARAM2 PARAM3
HEZI PARAM2 PARAM3
JOLA PARAM2 PARAM3
JOON PARAM2 PARAM3
...
```

In the slave script the values of the parameters (PARAM2 and PARAM3 in this example) are then accessible through the variables $$bpe{PARAM2} and $$bpe{PARAM3} (while $$bpe{PARAM1} contains the name of the baseline).

See Section 22.6.5.3 for details on the use of the utilities introduced above, or check the example user scripts delivered with the software in the directory ${X}/USERSCPT and copied during the installation to the user environment of ${U}/SCRIPT.

22.6.5. Useful Utilities for User Scripts

The example in Section 22.6.3 shows how the methods defined in ${BPE}/RUNBPE.pm can be used in user scripts that inherit the RUNBPE behavior. The general syntax is

```
  $bpe->functionName(list_of_arguments);
```

In addition, useful functions can be found in the Perl module ${BPE}/bpe_util.pm. Several of them are just shortcuts that simplify the usage of methods and standard procedures.

22.6.5.1. Accessing Variables and Program Panels

In Section 22.6.3 one particularly useful method — RUNBPE::getKeys — has been introduced. This method can be used to read a list of keyword values from menu input files (${U}/PAN/MENU_VAR.INP, ${U}/PAN/MENU_EXT.INP, etc.) and may, e.g., be used to get menu time variables or path and extensions of specific file types. It takes one or more keyword names as its arguments and returns an array of all relevant keyword values. The special feature of this method is that all menu variables (see Section 21.6) are correctly expanded, i.e, variables that depend on the currently processed campaign or session are correctly set. For example, after the call

```
  my ($sess,$sess30) = $bpe->getKeys('$S+0','$S-(30)');
```

the Perl variables $sess and $sess30 contain the identifier of the current session and of 30 sessions earlier respectively. After the call

```
  my $sess = $bpe->getKeys('$S+-');
```

the variable $sess contains the list of sessions within the V_MINUS and V_PLUS range around the current session as a list delimited by a newline character.

The method RUNBPE::getKey may be used to get the values of one keyword from one specified program input file. In distinction to the method RUNBPE::getKeys the input file has to be specified, e.g., $$bpe{U}/INP/GPSEST.INP:

```
  $bpe->getKey(inputFileName, keyword);
```

If the string FILNAM is added as the (optional) third parameter, the paths and extensions are stripped from the results (if applicable). If the string FILEXT is specified as the third parameter, the paths are stripped from the results (but not the extensions). Examples:

```
  my $sampl  = $bpe->getKey("$$bpe{U}/INP/GPSEST.INP","SAMPLE");
  my $lstFil = $bpe->getKey("$$bpe{U}/INP/HELMR1.INP","LISTFIL","FILEXT");
```

Note that in the examples the panels are accessed in the subdirectory INP in the BPE temporary directory. The panels are available in the subdirectory INP after the execution of the program with all menu variables replaced by their values according to the current session and campaign (see Section 22.7.1).

Another useful method is RUNBPE::putKey which may be used to set a specific value for a keyword in a program input file. It is called with three arguments to set one value for a keyword

```
  $bpe->putKey(inputFileName, keyword, value);
```

or with five arguments

```
  $bpe->putKey(inputFileName, keyword, value, action, selectionFileName);
```

where action is one of REPLACE, APPEND, or PREPEND. The method replaces (appends or prepends) the selection list of the specified keyword by the content of the file selectionFileName. The value is usually SELECTED. Note that a script ${X}/EXE/NPUTKEY is available to perform the same task outside of a BPE user script. Examples:

```
  $bpe->putKey("$$bpe{U}/PAN/CODXTR.INP","FILLST",$$bpe{SYSOUT});
  $bpe->putKey("$$bpe{U}/PAN/GPSEST.INP","PSFILES",
                                    "SELECTED","REPLACE",$selFil);
```

Note that in contrast to the above examples concerning RUNBPE::getKey the values here are set in the PAN directory to be used for the following execution of the program.

22.6.5.2. Methods to Handle Lists of Files

Because replacing of a list of files is used quite often, the bpe_util.pm utility provides with the function replaceSel a shortcut:

Usage: replaceSel($bpe,$inpFil,$keyWrd,$filNam)
Parameters: $bpe BPE object
$inpFil name of the input file without path and extensions
(in most cases simply the program name, e.g., ADDNEQ2)
$keyWrd keyword where the new selection should replace the old one
$filNam BPE cluster file containing the filenames (usually $PARAM1)

The last example from the previous section would simply result in:

```
  replaceSel($bpe,"GPSEST","PSFILES",$selFil);
```

Other methods useful to handle lists of files are replaceSelLst to replace the place holder SELECTED in a panel by a list of files and redFileList to remove non existent files from a list of input files (e.g., composed by the menu based on plus/minus variables).

Usage:	replaceSelLst($s,$prog,$keyWd,@filLst)	
Parameters:	$bpe	BPE object
	$prog	name of program
	$keyWd	keyword where SELECTED should be replaced
	@filLst	list with file names

Usage:	redFileList($s,$prog,$keyWd)	
Parameters:	$bpe	BPE object
	$prog	name of program
	$keyWd	keyword containing the file list

This method may be obsolete, since Version 5.2 offers time range variables that allow already the reduction of a selection list according to the available files (see Section 21.6.1). The function is mainly kept in the distribution to ensure the compatibility with older user scripts.

22.6.5.3. Methods for Parallel Running of User Scripts

Several methods for the parallelization of BPE user scripts are provided in the `${BPE}/bpe_util.pm`. They create the control file in the master script to prepare the parallel processing of user scripts in the BPE and help to simplify the mechanisms described in Section 22.6.4.

1. file–wise parallel processing (e.g., baseline by baseline):

Usage:	initPar_Bl($bpe,@filLst,@args)	
Parameters:	$bpe	BPE object (or $$bpe{CONTROL_FILE})
	@filLst	list with filenames to be processed in parallel
	@args	optional list with additional arguments for BPE control files; $args[] are passed to the parallel running scripts as PARAM2, PARAM3, etc.

2. process a list of files in a specified number of clusters:

Usage:	initPar_Cl($bpe,$bpeFil,@filLst,$numClu,$numFil,$minClu,@args)	
Parameters:	$bpe, @filLst, @args	as in (1)
	$bpeFil	name of BPE cluster files (a cluster number and an extension will be appended, specify $bpeFil=${dirBpe}CODSPP to generate cluster files ${dirBpe}CODSPP001.BPE, ${dirBpe}CODSPP002.BPE, ...)
	$numClu	desired number of clusters
	$numFil, $minClu	not used if $numClu>0

3. process a list of files with a specified number of files in each cluster:

Usage:	initPar_Cl($bpe,$bpeFil,@filLst,$numClu,$numFil,$minClu,@args)	
Parameters:	$bpe, @filLst, @args	as in (2)
	$bpeFil	as in (1)
	$numClu	$numClu=0 to enable this option
	$numFil	desired number of files per cluster
	$minClu	a minimum number of clusters

4. use existing cluster files (e.g., output from the programs SNGDIF or MKCLUS) for the parallel processing:
   Usage: initPar_Fl($bpe,@cluFil)
   Parameters: $bpe as in (1)
   @cluFil list of the prepared cluster files

Examples for the use of these methods can be found in the user scripts of the example BPEs provided in the distribution of the *Bernese GNSS Software* Version 5.2 .

The methods initPar_Cl and initPar_Fl pass lists of files to be processed sequentially to the slave script. In the slave script the file selection in the program input file can be defined using the method replaceSel already described above in Section 22.6.5.2.

22.6.5.4. Method for Special Action NEXTJOB

Instead of directly calling the method $bpe–>PRT_GOTO($$bpe{PARAM2}); from the `${BPE}/RUNBPE.pm` (see Section 22.5.4) module, `${BPE}/bpe_util.pm` provides a subroutine for printing the GOTO_PID into the protocol file:
Usage: prtGoto($bpe,$pid)
Parameters: $bpe BPE object
$pid PID where to continue the processing

The following example shows a part of the user script `GOTO` which selects the PID on where to continue from the `NEXTJOB` parameters according to the value of the variable `STARTCOUNT`. That means for each start of the script, e.g., in an iteration loop, the next goto PID is used (see also Section 22.5.4):

```
# Get variables
# -------------
  my $iRun = $$bpe{STARTCOUNT} - 1;  # because starts counting by zero

  my @nextJob = ($$bpe{PARAM2},$$bpe{PARAM3},$$bpe{PARAM4},$$bpe{PARAM5},
                 $$bpe{PARAM6},$$bpe{PARAM7},$$bpe{PARAM8},$$bpe{PARAM9});

# Write the "next jobs" statement
# ------------------------------
  if (defined($nextJob[$iRun])) { $bpe->PRT_GOTO($nextJob[$iRun])}
```

22.6.5.5. Utilities for Output and Protocols

The module `${BPE}/bpe_util.pm` provides two methods for printing messages into the protocol files:

- Print a message line in the protocol file (`status` = `MSG`):
  Usage: prtMess($bpe,$msg)
  Parameters: $bpe BPE object
  $msg string to print
- Report an error in the protocol file (`status` = `ERR`):
  Usage: prtErr($bpe,$msg)
  Parameters: $bpe BPE object
  $msg string to print
  This method prints only the message into the protocol file. The user script should be terminated using the perl command die() to stop the BPE:

```
  prtErr($bpe,"SOME␣ERROR");
  die("BPE␣stopped␣by␣user␣script");
```

Warning messages of the Bernese programs are merged into the protocol file if they are executed in the BPE user script. They may be extracted into an additional file using the method getWarn of the `${BPE}/bpe_util.pm` module:

Usage: getWarn($bpe,$sumFil)
Parameters: $bpe BPE object
$sumFil file in which the warnings should be collected.
Messages are appended to this file.

As an example, to collect all warning messages from reading the RINEX observation files in a number of parallel running RXOBV3 scripts, you may add a command like getWarn($bpe,"${dirOut}RNX${yyssss}.ERR"); to the user script.

To concatenate summary, output, or result files from the processing of a BPE into a file summarizing the BPE run, a method appFile is available in the module `${BPE}/bpe_util.pm`:

Usage: appFile($title,$blank,$source,$target,$stop)
Parameters: $title title line (blank: no title printed)
$blank number of blank lines to append
$source file which should be appended
$target file to which the source file should be appended
$stop parameter left out: no message if the source file is missing
== 0: warning if the source file is missing
== 1: script stops if the source file is missing

The value for $stop may also be ≫ 1 and is interpreted as a Perl time stamp. In that case it is expected that the source file is younger than the time stamp from $stop. The Perl time stamp of the current source file is returned by the appFile function. If the source file is too old, the script stops. A small example shall illustrate this feature:

```
  my $filTim = gps_date("-ymd␣2000␣01␣01","-o␣%U");

# - RINEX pseudo-graphics
  my $sumFil = "${dirSmc}GRA${yyssss}.${extSmc}";
  $filTim = appFile("PART␣1:␣RINEX␣PSEUDO-GRAPHICS",
            2,$sumFil,$prcFil,$filTim);

# - Single-point-positioning summary
  $sumFil = "${dirOut}SPP${yyssss}.${extOut}";
  $filTim = appFile("PART␣2:␣SINGLE-POINT-POSITIONING␣SUMMARY",
            2,$sumFil,$prcFil,$filTim);
```

If for some reasons the processing has only updated the first summary file ${dirSmc}GRA${yyssss}.${extSmc} (e.g., none of the RINEX files has been accepted for the processing) but the second one (${dirOut}SPP${yyssss}.${extOut}) only exists from a previous processing of the same session the method appFile will stop with an error as in the case when the file is completely missing.

Finally, two methods allow the BPE client scripts to directly print to standard output (which is, in fact, the BPE log file):

- Print a text string:
  Usage: sysprint($text)
  Parameters: $text text string

- Print the content of a whole BPE file:
  Usage: prtBPEfile($BPEfile,$delete)
  Parameters: $BPEfile name of the file to be printed
  $delete flag whether the file should be deleted after printing ($\neq 0$) or not ($= 0$)

22.6.5.6. Preparation of Processing

The processing examples provided with Version 5.2 of *Bernese GNSS Software* assume the recommended dataflow where all external input files are collected in a datapool area before they are processed in the campaign directories. For this purpose a specific method copyRef is included in `${BPE}/bpe_util.pm`:

Usage: copyRef($source,@file_list)
Purpose: Copy project specific files from the reference directory into the campaign area
Parameters: $source directory where the source files are located
@file_list contains a list of files to be copied and a flag (1/0) after a blank to specify whether it is a mandatory or an optional file.

Remarks:

- Files are only copied if they are newer in the source directory
- First, copies of files with randomly generated names are created. Afterwards these are renamed to their final names to minimize the risk of incomplete files.
- ERP files that are expected in the international format (`IEP` for processing with *Bernese GNSS Software*) may also be accepted with the extension `ERP` in the datapool area as they are available in the international databases.
- Precise orbit files (default extension `PRE`) are alternatively excepted with the extensions `SP3` or `EPH` for compatibility with international databases.
- If the requested file exists in a UNIX– or gnu–compressed version (indicated by `.Z` or `.gz`) they are accepted and decompressed (using the program `gzip` that is expected to be installed in a way that it can called without the full path).

Example:

```
# Get variables
# ------------
  my ($wwwwd,
      $dirPre,  $dirIep,  $dirClk,
      $extPre,  $extIep,  $extClk) =


  $bpe->getKeys
     ('$WD+0',
      'DIR_PRE','DIR_IEP','DIR_RXC',
      'EXT_PRE','EXT_IEP','EXT_RXC');

# Construct list of external product files
# ---------------------------------------
  @filLst = ();
  push @filLst,"${dirPre}COD${wwwwd}.${extPre}␣1";
  push @filLst,"${dirIep}COD${wwwwd}.${extIep}␣1";
  push @filLst,"${dirClk}COD${wwwwd}.${extClk}␣1";
  push @filLst,"${dirClk}COD${wwwwd}.${extClk}_30S␣0";

# Copy External Product Files
# --------------------------
  $refDir = "$ENV{D}/COD";
  copyRef($refDir,@filLst);
```

With this sequence the CODE product series is copied from the datapool into the campaign area. Even if the orbit files are requested with the extension `PRE`, in the datapool also the file COD${wwwwd}.EPH.Z (as it is, for instance, available at http://www.aiub.unibe.ch/download/CODE/) is copied, decompressed and renamed. Similarly, the file COD${wwwwd}.ERP.Z is accepted and renamed as the ERP file in the international format to COD${wwwwd}.IEP in the campaign area. If no daily ERP file is available in the datapool, weekly files with the name COD${wwww}7.ERP.Z are accepted. The clock RINEX file with a 30 seconds sampling is optional, whereas all other files are mandatory. If these files are not found (considering the advanced compatibility features with respect to international databases), the script will stop.

RINEX observation files are nowadays typically available in the Hatanaka compression format. The ${BPE}/bpe_util.pm utility contains methods to decompress and compress the corresponding RINEX files. They are used, for instance, to copy the RINEX observation files from the datapool into the campaign area by the user script RNX_COP provided with the processing examples:

- Uncompress RINEX files (M/m, N/n, G/g, O/o, D/d):

Usage:	@done = crz2rnx(@sourc,e,[$dest],[–l|–u])	
Parameters:	@source	List with compressed RINEX files
	$dest	Path to destination directory Default: original directory
	–l/–u	Extensions in lower/upper case
	@done	List of uncompressed files

- Compress RINEX files (M/m, N/n, G/g, O/o, D/d):

Usage:	@done = rnx2crz(@source,[$dest],[–l|–u])	
Parameters:	@source	List with uncompressed RINEX files
	$dest	Path to destination directory Default: original directory
	–l/–u	Extensions in lower/upper case
	@done	List of compressed files

The latest versions of the executables RNX2CRX and CRX2RNX are expected in the PATH variable (accessible without specifying the full path).

22.6.5.7. Archiving of Results

At the end of the automated processing the result files shall be copied from the campaign area to the product archive (savedisk area). The utility ${BPE}/bpe_util.pm provides a dedicated routine for this purpose:

Usage:	copy2archive($source,$target,$flags,$title)	
Parameters:	$source	File to be copied into the archive (wildcards allowed)
	$target	Target where to put the file (may be a directory or a file)
	$flags	c: UNIX compress the file at the target z: gnu–zip the file at the target f: append a title line (may be used together with c/z)
	$title	title line

To make sure that the target directory for a file exists, the method check_dir is available:

Usage:	check_dir($dir)	
Parameters:	$dir	List of directories to check (and create, if non–existent)
Remark:	Returns the list of directories created.	

Examples for the use of these two routines may be found in the related user scripts of the provided example BPEs.

22.6.5.8. Define User Variables

You may add any user–defined variable or modify the actual value of the client variables and user–defined server variables within a user script. The variable names and the values have to be prepended to the keyword `USERVAR` in the `${U}/PAN/MENU_VAR.INP` (remember that `${U}` points to the temporary BPE user environment). The method `setUserVar` in the `${BPE}/bpe_util.pm` module performs this task in a very comfortable way:

Usage:	`setUserVar($bpe,%vars)`	
Parameters:	`$bpe`	BPE object
	`%vars`	hash array containing the variables as keys with corresponding values
Example:	`setUserVar($bpe,"CD4","$param1","pgm","GPSEST")`	
		defines user variables `$CD4` with the content of `$param1` and `$pgm` with the value `GPSEST`

A special case is the update of the ranges of session variables (see option "RANGES OF PREDEFINED VARIABLES" in panel "MENU_VAR 2: Ranges"). The module `${BPE}/bpe_util.pm` contains a corresponding method:

Usage:	`setMinusPlus($bpe,$minus,$plus)`	
Parameters:	`$bpe`	BPE object
	`$minus`	new value of minus–range in `MENU_VAR.INP` Default: `$PARAM1`
	`$plus`	new value of plus–range in `MENU_VAR.INP` Default: `$PARAM2`

Note that the range variables are changed in the file `MENU_VAR.INP` of the script's option directory. The variables `$$bpe{V_PLUS}` and `$$bpe{V_PLUS}` remain unchanged.

The aforementioned variables are only valid for particular user scripts. With the method provided by `${BPE}/RUNBPE.pm` even BPE server variables may be redefined or added by a user script.

Usage:	`setVar($varNam,$varVal)`	
Parameters:	`$varNam`	name of the variable to be changed
	`$varVal`	new value of the variable

Example:	`my $dirSav = "$ENV{S}/" . $result . "/$yyyy/";`
	`$bpe->setVar('DIRSAV',$dirSav);`	

		Define the variable `DIRSAV` as a new BPE server variable for the potential use in a subsequent user script.
Remark:		The BPE server variable may either already be defined in the PCF or may also be introduced first time in this user script.

Even if this method seems to provide a powerful tool for a flexible management of BPE tasks, you have to take care of the value of these variables if you do not start a BPE from its beginning.

22.6.5.9. Define Time Variables

With the method RUNBPE::getKeys all predefined time variables of the menu (see Table 21.1 on page 462 in Chapter 21) are accessible in the BPE user scripts. This method can only be used within the environment of the BPE client.

The module `${X}/EXE/Gps_Date.pm`, on the other hand, can also be used outside a BPE script (e.g., to start a BPE). The utility allows for all types of date and time conversions. Date and time strings in different formats are accepted as input. The output format can be specified separately:

```
gps_date("-yd  02 143",    "-o %X");                    # 2002-MAY-31
gps_date("-ymd 2002 5 31","-o COD%W%w");                # COD11685
@time = gps_date("-mjd 52417.5",  "-o %Y %j %I");       # Fill three values in an array
scalar(gps_date("-mjd 52417.5","-o %Y %j %I"));         # 2002 143 M
gps_date("-yd  02 143","-d +5","-o %y%j");              # 02148
```

The last line of this example demonstrates the possibility to add or subtract a number of days (hours or weeks). The important difference to the predefined time variables provided by the menu is that `${X}/EXE/Gps_Date.pm` works independently from the session table. For the complete list of input and output options we refer to the help display shown in Figure 22.11.

Moreover, special methods exist in `${BPE}/bpe_util.pm` to create formatted datum strings:

```
Usage: gps_date <[ -ymd | -yd | -wd | -mjd | -doy DATSTR ] [ -t[today] ]>
                 [ -hms | -hid             TIMSTR ]
                 [ -h   | -d  | -w         +/-INT ]
                 [ -o                      OUTSTR ]
                 [ -v   ]


Supported DATSTR: -ymd  ..  yy|yyyy   mm|mon   dd       with delimiters [-_\s]+
                  -yd   ..  yy|yyyy  doy|sess|doyid     with delimiters [-_\s]+
                  -mjd  ..  ddddd[.ddddd]
                  -wd   ..  wwww d                      with delimiters [-_\s]+
                  -doy  ..  doy                         may be negative
          TIMSTR: -hms  ..  hh|id mm ss                 with delimiters [-:\s]+
                  -hid  ..  hh|id                       work for hours only
          OUTSTR: -o    ..  any text + spec.chars


Spec. chars: year    = %y %Y       = [00-99] [1999]
             month   = %m %b %B %C = [01-12] [jan-dec] [JAN-DEC] [Jan-Dec]
             day     = %d          = [01-31]
             doy     = %j          = [001-365]
             dow     = %w %e %E %F = [0-6] [su-sa] [SU-SA] [Su-Sa]
             gpsweek = %W          = [0-????]
             gpssec  = %G          = [0-??????]
             woy     = %A          = [01-53]
             yow     = %a          = [00-99]
             modjul  = %J          = [?????.?????]
             modjul  = %IJ         = [?????]
             modjul  = %FJ         = [0.?????]
             hour,id = %H %I %i    = [00-23]  [A-X] [a-x]  (A=00,..,X=23)
             min     = %M          = [00-59]
             sec     = %S          = [00-59]

             dattim  = %V %v       = [YYYY-MM-DD HH:MM:SS] [YYYY-MON-DD HH:MM:SS]
             dat     = %X %x       = [YYYY-MM-DD]          [YYYY-MON-DD]
             tim     = %Z          = [HH:MM:SS]
             timprt  = %z          = [DD-MON-YYYY HH:MM:SS]
             seconds = %U          = seconds from 1.1.1970

             spec    = %t %n %%    = "\tab" "\newline" "%"
             pads    = %: %_       = "any_pad","blank_pad", (default="0_pad")
```

Figure 22.11.: Options for module `Gps_Date.pm`.

- Convert localtime–/gmtime–output to a formatted datum string, e.g., "14–JUL–2004 13:50:15":

Usage:	timstr(@time)	
Parameters:	@time	output from localtime or gmtime

- Return time stamp for processing protocols, e.g., "14–JUL–2004 13:50:15":

Usage:	timprt	
Options:	–gmt	
	–l[ocal]	(default)

22.6.5.10. More Useful Tools

A useful method, e.g., when screening observations is deleteFiles. The method deletes all files listed in an input list (e.g., observation files):

Usage:	deleteFiles($bpe,$delFil,$delete)	
Parameters:	$bpe	BPE object
	$delFil	file containing the list of files to be deleted
	$delete	flag whether the list–file should be deleted ($\neq 0$) or not ($= 0$)

Another helpful method of `${BPE}/bpe_util.pm` in automated processing is isHourly. It checks whether a session identifier indicates "hourly processing". It is true if a session ends with a character A..X:

Usage:	isHourly($session)	
Parameters:	$session	session identifier

22.6.6. Stop a User Script with an Error

To stop the user script in case of an error do not use the Perl command exit 1. If the user script is terminated with the command die the BPE client reports the error to the BPE server. Using exit instead of die stops the complete client and no ending message is sent to the BPE server. As a consequence, the BPE server will wait for this client script forever.

22.6.7. Old User Scripts and Non–Perl User Scripts

As stated above the user scripts on UNIX systems may be written in any programming language provided that the environment variables are accessible and the system commands can be executed. If the run method is not overloaded in the user script (overloading is possible if the user script is written in Perl), the default run method (defined in `${BPE}/RUNBPE.pm`) is called and performs the following system command:

```
system "$$bpe{U_OLD}/SCRIPT/$$bpe{SCRIPT}";
```

It means that non–Perl user scripts are started as separate processes without any arguments. These scripts cannot use directly the methods defined in `${BPE}/RUNBPE.pm`. Instead of that the non–Perl scripts can use the utilities that are located in the `${X}/SCRIPT` directory. Here is the list of available files with their brief description:

AUTO_UPD BEG_MENU DO_TAIL SET_SESS dummy scripts
(they have no functionality but they are called in old user scripts)
CLEAN_UP deletes some program–specific files
END_MENU sets the exit code according to error status
PRT_GOTO PRT_MESS RUN_PGMS
wrappers of the corresponding methods defined in ${BPE}/RUNBPE.pm

22.7. BPE Program Option Directories

22.7.1. Program Input Files

The program input files used by Bernese programs started in user scripts are located in the user's ${U}/OPT–directory tree. For each script a subdirectory of ${U}/OPT is specified in the PCF in the column "Opt_dir". Several scripts may access the same option subdirectory. This subdirectory has to contain not only the program input files but also all menu input files MENU*.INP (except of MENU_CMP.INP).

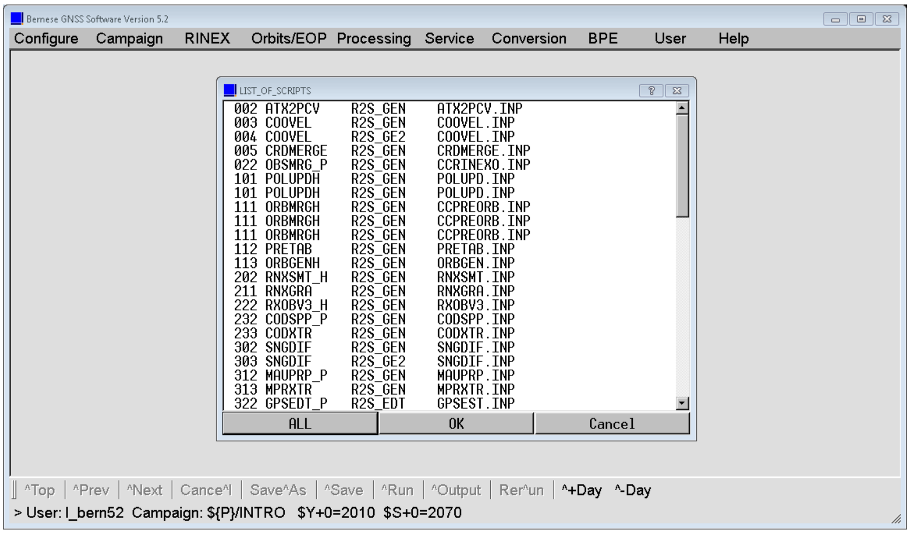
Before starting the user script the client copies the content of the corresponding option directory into the PAN subdirectory of the client–specific temporary area. Accessing program input files using the RUNBPE::putKey method modifies the files in this temporary directory. Before executing a Bernese program the client executes the menu in non–interactive mode which reads the input file and replaces all menu variables, paths and extensions with the correct values. The resulting program input file is stored in the INP subdirectory of the temporary area. This file is then used as input file for the Fortran program. The program panel in this subdirectory has to be accessed to get filenames using the command RUNBPE::getKey.

It is important to note that all ${U}/OPT directories must contain the menu input files MENU*.INP. They have to be copied into the directory manually by the user when preparing the program input options for a BPE process. The menu input files need to be available even if no Bernese program is started by the script at all.

22.7.2. Editing of Program Options

The menu provides a facility for easily checking and modifying options in the program input files. Selecting "Menu>BPE>Edit PCF program input files" opens a file dialogue where you may select a PCF file. The PCF is then displayed as a selection list similar to that shown in Figure 22.12. It lists the PID, script name, option directory, and program input file name for all user scripts that start a Bernese program. It is important to use the syntax described in Section 22.6.3 for starting programs to make this mechanism work (the menu program looks for the strings $PGMNAM="xxx" in the user scripts to compile the selection list).

From this list you may select one or several program input files. The corresponding program panels are then displayed successively and you may adapt the program options for the specific application. It is recommended to use menu time variables as part of filenames wherever possible. The use of BPE server variables for some specific options is beneficial because corresponding adaptions can be done when starting the BPE.



```
002 ATX2PCV     R2S_GEN     ATX2PCV.INP
003 COOVEL      R2S_GEN     COOVEL.INP
004 COOVEL      R2S_GE2     COOVEL.INP
005 CRDMERGE    R2S_GEN     CRDMERGE.INP
022 OBSMRG_P    R2S_GEN     CCRINEXO.INP
101 POLUPDH     R2S_GEN     POLUPD.INP
101 POLUPDH     R2S_GEN     POLUPD.INP
111 ORBMRGH     R2S_GEN     CCPREORB.INP
111 ORBMRGH     R2S_GEN     CCPREORB.INP
111 ORBMRGH     R2S_GEN     CCPREORB.INP
112 PRETAB      R2S_GEN     PRETAB.INP
113 ORBGENH     R2S_GEN     ORBGEN.INP
202 RNXSMT_H    R2S_GEN     RNXSMT.INP
211 RNXGRA      R2S_GEN     RNXGRA.INP
222 RXOBV3_H    R2S_GEN     RXOBV3.INP
232 CODSPP_P    R2S_GEN     CODSPP.INP
233 CODXTR      R2S_GEN     CODXTR.INP
302 SNGDIF      R2S_GEN     SNGDIF.INP
303 SNGDIF      R2S_GE2     SNGDIF.INP
312 MAUPRP_P    R2S_GEN     MAUPRP.INP
313 MPRXTR      R2S_GEN     MPRXTR.INP
322 GPSEDT_P    R2S_EDT     GPSEST.INP
```

> User: l_bern52 Campaign: ${P}/INTRO $Y+0=2010 $S+0=2070

Figure 22.12.: Selection dialogue for editing program input files.

As usual, you may browse forward and backward using the buttons `^Next`, `^Prev`, or `^Top` and save the panels with `^Save` or quit using `Cance^l`. If an input file is not yet available in the option directory the menu program gets it from `${U}/PAN` or alternatively from `${X}/PAN`. The program specific help is available as well. Checking of input field values is disabled.

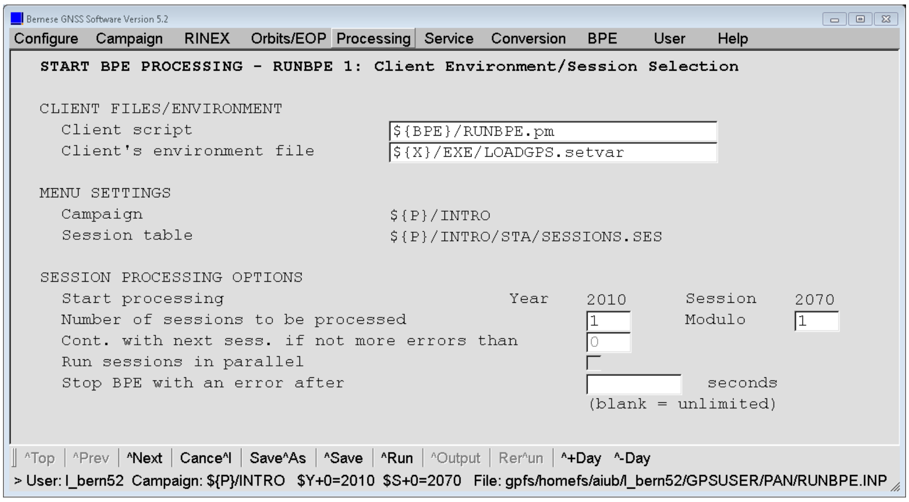
In order to edit menu input files (e.g., to modify the list of user variables in `MENU_VAR.INP` or a default extension in `MENU_EXT.INP`), the menu drop–down item "Menu>BPE>Edit single menu/program input file" has to be used. Navigate to the desired option directory and open the input file.

22.8. Starting the BPE

The BPE can be started either in interactive or in non–interactive mode.

22.8.1. Interactive Mode

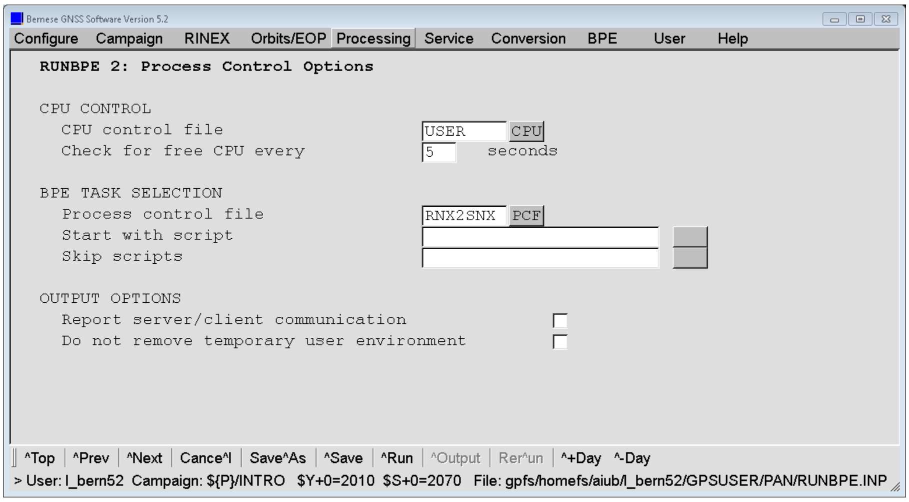
The BPE can be started in interactive mode directly from the running Bernese menu program in "Menu>BPE>Start BPE processing". In Figure 22.13 the first panel is displayed. There are several read–only options (“Campaign”, “Session table”, and “Start processing”). These options can be set using "Menu>Campaign>Select active campaign" or "Menu>Configure>Set session/compute date", respectively. The option “Client script” specifies the name of the BPE client peer. The script is described in Section 22.3.2. The option “Client's environment file” is used on UNIX only. The file name specified here is read on each client host. Under Windows the environment file is not used (the option may be left blank). The directory paths specified with the filenames in these options have to be correct on each host used for the BPE. We refer to Section 22.3.3 for more details. To avoid overlapping BPEs in, e.g., hourly processing, one may limit the run time with option “Maximum Time (in Seconds) for BPE Run” .


Bernese GNSS Software Version 5.2
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un ^+Day ^-Day

```
START BPE PROCESSING - RUNBPE 1: Client Environment/Session Selection

CLIENT FILES/ENVIRONMENT
  Client script                  ${BPE}/RUNBPE.pm
  Client's environment file      ${X}/EXE/LOADGPS.setvar

MENU SETTINGS
  Campaign                       ${P}/INTRO
  Session table                  ${P}/INTRO/STA/SESSIONS.SES

SESSION PROCESSING OPTIONS
  Start processing                          Year   2010     Session    2070
  Number of sessions to be processed               1        Modulo     1
  Cont. with next sess. if not more errors than    0
  Run sessions in parallel                         □
  Stop BPE with an error after                            seconds
                                                   (blank = unlimited)
```

> User: l_bern52 Campaign: ${P}/INTRO $Y+0=2010 $S+0=2070 File: gpfs/homefs/aiub/l_bern52/GPSUSER/PAN/RUNBPE.INP

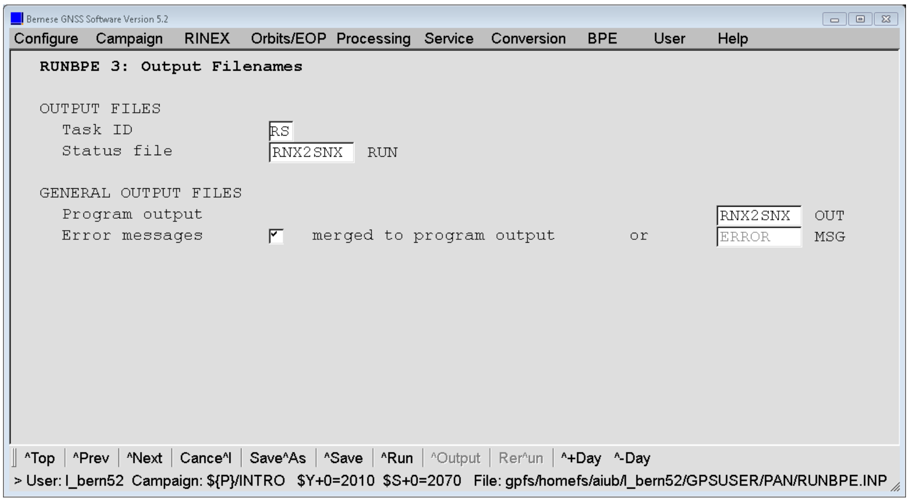
Figure 22.13.: Client environment/session selection.


Bernese GNSS Software Version 5.2
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un ^+Day ^-Day

```
RUNBPE 2: Process Control Options

CPU CONTROL
  CPU control file                       USER    CPU
  Check for free CPU every               5     seconds

BPE TASK SELECTION
  Process control file                   RNX2SNX PCF
  Start with script
  Skip scripts

OUTPUT OPTIONS
  Report server/client communication             □
  Do not remove temporary user environment       □
```

> User: l_bern52 Campaign: ${P}/INTRO $Y+0=2010 $S+0=2070 File: gpfs/homefs/aiub/l_bern52/GPSUSER/PAN/RUNBPE.INP

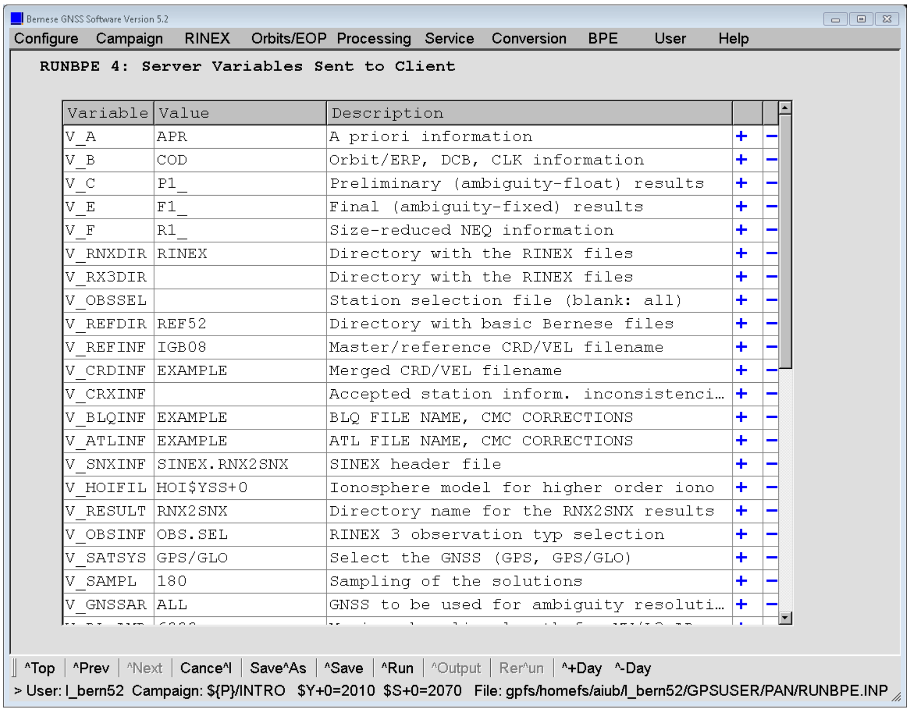
Figure 22.14.: Process control options.


Bernese GNSS Software Version 5.2
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un ^+Day ^-Day

```
RUNBPE 3: Output Filenames

OUTPUT FILES
  Task ID              RS
  Status file          RNX2SNX   RUN

GENERAL OUTPUT FILES
  Program output                                                 RNX2SNX   OUT
  Error messages       ☑   merged to program output       or     ERROR     MSG
```

> User: l_bern52 Campaign: ${P}/INTRO $Y+0=2010 $S+0=2070 File: gpfs/homefs/aiub/l_bern52/GPSUSER/PAN/RUNBPE.INP

Figure 22.15.: Output filenames.



RUNBPE 4: Server Variables Sent to Client

Variable	Value	Description		
V_A	APR	A priori information	+	−
V_B	COD	Orbit/ERP, DCB, CLK information	+	−
V_C	P1_	Preliminary (ambiguity-float) results	+	−
V_E	F1_	Final (ambiguity-fixed) results	+	−
V_F	R1_	Size-reduced NEQ information	+	−
V_RNXDIR	RINEX	Directory with the RINEX files	+	−
V_RX3DIR		Directory with the RINEX files	+	−
V_OBSSEL		Station selection file (blank: all)	+	−
V_REFDIR	REF52	Directory with basic Bernese files	+	−
V_REFINF	IGB08	Master/reference CRD/VEL filename	+	−
V_CRDINF	EXAMPLE	Merged CRD/VEL filename	+	−
V_CRXINF		Accepted station inform. inconsistenci…	+	−
V_BLQINF	EXAMPLE	BLQ FILE NAME, CMC CORRECTIONS	+	−
V_ATLINF	EXAMPLE	ATL FILE NAME, CMC CORRECTIONS	+	−
V_SNXINF	SINEX.RNX2SNX	SINEX header file	+	−
V_HOIFIL	HOI$YSS+0	Ionosphere model for higher order iono	+	−
V_RESULT	RNX2SNX	Directory name for the RNX2SNX results	+	−
V_OBSINF	OBS.SEL	RINEX 3 observation typ selection	+	−
V_SATSYS	GPS/GLO	Select the GNSS (GPS, GPS/GLO)	+	−
V_SAMPL	180	Sampling of the solutions	+	−
V_GNSSAR	ALL	GNSS to be used for ambiguity resoluti…	+	−

> User: l_bern52 Campaign: ${P}/INTRO $Y+0=2010 $S+0=2070 File: gpfs/homefs/aiub/l_bern52/GPSUSER/PAN/RUNBPE.INP

Figure 22.16.: Server variables.

Figure 22.14 shows the second panel that immediately follows the first panel if one session is processed. In case tmore sessions are to be processed in parallel, an additional panel is displayed (the multi session BPE is described in Section 22.9). The options shown in Figure 22.14 are self–explanatory and we refer to the corresponding help for their description.

In the third panel (see Figure 22.15) the user defines the names of the output file, error file (that may be merged into the output file), and the status files. Output file names have to be unique if several BPEs are started in the same campaign. Again, these options are self–explanatory and we refer to the corresponding help for their full description.

The last panel (see Figure 22.16) defines the so–called server variables. To be compatible with the old BPE, the variable names should start with `V_` (see Section 22.6.2). The menu program lists the variables defined in the PCF and the user may edit the list according to his needs. He may, e.g., specify the first characters of the result files individually for each BPE run, if the corresponding variables are consistently used as part of the output file names in BPE program panels.

After setting the options in all panels, the BPE server can be started in interactive mode using the `^Run` button. The BPE server is then started as independent process, and the server's window appears displaying the BPE status file (see Section 22.10.1) that is updated every 5 seconds. In the meantime, the menu may be used to perform other tasks or may even be closed without interrupting the execution of the BPE.

22.8.2. Non–Interactive Mode

The pre–requisite for starting the BPE in non–interactive mode is the correct selection of all options in exactly the same manner as in the interactive mode (see the previous section). Instead of running the BPE server interactively (by pressing the ^Run button) the options have to be stored using the ^Save button. The BPE server options are stored into a file with the default name RUNBPE.INP. This file has the standard Bernese input file format and resides in the ${U}/PAN directory. After creating the correct RUNBPE.INP file, the BPE server can be started in a non–interactive mode using the following command (see Section 21.10.4):

```
${XQ}/menu ${U}/PAN/MENU.INP ${U}/PAN/MENU_RUNPCF.INP
```

The first argument is the name of the primary input file, the second argument is the name of an auxiliary input file (the menu command file) that contains the following keyword:

```
RUN_BPE 1  "${U}/PAN/RUNBPE.INP"

PRINT_PID 1  "1"
```

The second line is not needed but helpful because it reports the process ID of the BPE server to the standard error. This allows to identify the related process if it must be stopped for any reason.

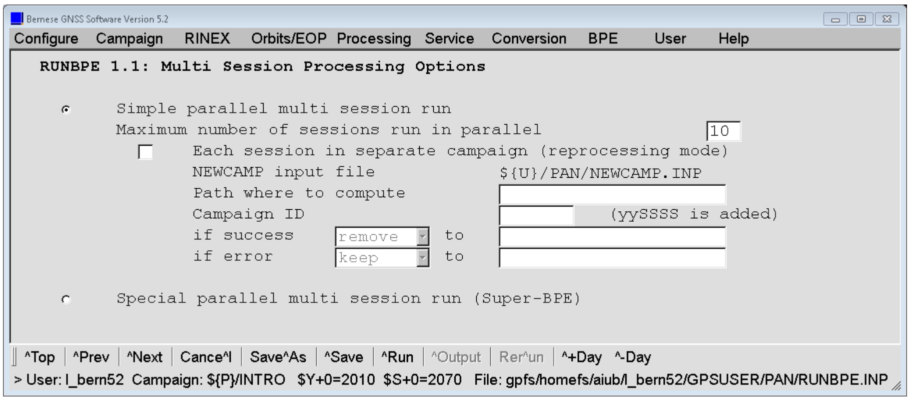
Furthermore, one remark concerning the input of the BPE server has to be added. Unlike the "normal" (i.e. the Fortran) Bernese programs the BPE server reads its input from several input files, namely from the MENU*.INP files and from its main input file RUNBPE.INP. This exception must be taken into account particularly with respect to four options: BPE_CAMPAIGN, SESSION_TABLE, YEAR, and SESSION (the campaign to be processed, the session table to be used, the year and session). These four options are declared as "pointer" in RUNBPE.INP which means that their actual values are taken from the file MENU.INP. By (intentionally or unintentionally) changing, e.g., the current session in MENU.INP you may thus execute the BPE non–interactively for another session.

In order to make the starting of BPE more simple, we have prepared a Perl module ${BPE}/startBPE.pm that is intended to be used in Perl scripts. We refer to the header of the file for detailed information on its usage. See also the example startup script ${U}/SCRIPT/rnx2snx_pcs.pl that is delivered together with the software.

22.9. Multi–Session Processing

The BPE allows to process many sessions in one run. Whether the sessions are processed sequentially or in parallel has to be decided by "Run sessions in parallel" in panel "RUNBPE 1: Client Environment/Session Selection" (see Figure 22.13). The sequential mode needs no specific comment compared to the processing of a single session only. Two additional options might become relevant depending on your application:

- "Continue if number of errors lower or equal than specified" that is sufficiently described in the online help.
- "Session Modulo" allows to process only every n^{th} session (e.g., in the case of daily sessions a modulo of "7" would activate weekly processing — where the current session defines which day of the session is processed).


Bernese GNSS Software Version 5.2
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un ^+Day ^-Day

```
RUNBPE 1.1: Multi Session Processing Options

●    Simple parallel multi session run
     Maximum number of sessions run in parallel              10
  □      Each session in separate campaign (reprocessing mode)
         NEWCAMP input file             ${U}/PAN/NEWCAMP.INP
         Path where to compute
         Campaign ID                           (yySSSS is added)
         if success    remove   to
         if error      keep     to

○    Special parallel multi session run (Super-BPE)
```

> User: l_bern52 Campaign: ${P}/INTRO $Y+0=2010 $S+0=2070 File: gpfs/homefs/aiub/l_bern52/GPSUSER/PAN/RUNBPE.INP

Figure 22.17.: Multi session processing options.

If the checkbox "Run sessions in parallel" is selected, an additional panel shown in Figure 22.17 appears that needs a bit more description. We recommend to use the "Simple parallel multi session run" mode unless you really need the "Special parallel multi session run (Super-BPE)" to support old–styled "SUPERBPEs" . Just one option needs to be specified that is self–explanatory: "Maximum number of sessions run in parallel". The optimal choice depends on the number and type of jobs within the PCF and the of available CPUs for these tasks.

Running too many sessions in the same campaign at the same time may not be optimal for several reasons. Firstly, the number of files in the directories of the campaign may increase to a level that degrades the performance of your operating system. In addition, all scripts and filenames (including all temporary files) must be fully sessions–independent what needs a significant extra effort. For this reason we offer an option to process "Each session in separate campaign". The campaign is prepared according to the specifications in ${U}/PAN/NEWCAMP.INP. In particular, in this application the option "Copy files into the campaign" is very useful to copy all necessary files that are needed before the BPE starts to process the particular session (see Section 3.3) into the campaign. In this case you have to specify the "Path where to compute" (that should be an area with a quick disk access) and a "Campaign ID" that is complemented by the two digit year and the session number[3]. After successful (or not) completion, the campaign may either be removed, kept, or moved to another location (e.g., in case of a job–queueing system where the campaign area is removed after the end of the process).

The "special" mode is made available in the new BPE for backward compatibility with the earlier BPE version. Actually the "special" mode does not conform with the fundamental philosophy of the new BPE — one server keeps the information about all running clients — and it may no longer be available in a future release. The basic idea behind the "special" mode is that there is one BPE run (so–called super–BPE) that

- runs user scripts that perform session–independent tasks before the actual session processing is started,
- prepare and start in parallel as many BPE servers as sessions that have to be processed. Each BPE sub–process is responsible for the processing of exactly one session. There is a sample of the master–slave pair of user scripts that accomplish this parallelization: SBPEAP and SPBE_P.

[3] Please keep in mind the limitation of filenames including the path to 32 characters.

- After finishing all sessions the super–BPE process runs user scripts that typically perform task like extracting summary information from individual runs, cleaning directories etc.

From the description of the "special" mode it is clear that options for two BPE processes have to be specified — the first set of options is used for the single super–BPE process, the second set of options is used for all session–specific BPE–subprocesses. This is handled by the menu program which displays separate panels for the so–called super–BPE in addition to the standard panels. One has to take into account that the names of BPE–output files (i.e., the standard output file, the error file, and the status file) have to contain variables that distinguish among processed sessions (otherwise, the parallel running BPEs overwrite them).

22.10. BPE Output and Protocol Files

Both BPE server and BPE client log information documenting their run. The BPE server writes an output file, abe error message file (that may be merged into the output file) and a so–called status file, containing summary information about a running BPE process.

22.10.1. Server Output and Status Files

The server output file and the error message file follow the same conventions as the corresponding files of any other Bernese program with the exception that the name of the output file must not be blank (i.e., the `*.Lnn` naming convention is not supported). The content of the output file differs in debug mode and non–debug mode (option "Report server/client communication" in panel "RUNBPE 2: Process Control Options"). In debug mode all the server–client communication is reported in addition to the standard file content.

An example of the server output (in non–debug mode) is given in the following figure which is based on the BPE example `BASTST.PCF`:

```
 Time                 Sess PID     Script   Option     Status
-------------------------------------------------------------------------------
 12-May-2015 06:19:09  2080 YR:2010  BASTST              : Server started at 54073
 12-May-2015 06:19:09  2080 101_000  GPSESTAP BAS_GEN    : Client started on: localhost
 12-May-2015 06:19:09  2080 101_000  GPSESTAP BAS_GEN    : Script started
 12-May-2015 06:19:09  2080 101_000  GPSESTAP BAS_GEN    : Script finished  OK
 12-May-2015 06:19:09  2080 YR:2010  BASTST              : Initialize parallel script 102
 12-May-2015 06:19:09  2080 102_001  GPSEST_P BAS_GEN    : Client started on: localhost
 12-May-2015 06:19:09  2080 102_002  GPSEST_P BAS_GEN    : Waiting for CPU: ANY
 12-May-2015 06:19:09  2080 102_001  GPSEST_P BAS_GEN    : Script started
 12-May-2015 06:19:50  2080 102_001  GPSEST_P BAS_GEN    : Script finished  OK

 ...

 12-May-2015 06:26:29  2080 102_012  GPSEST_P BAS_GEN    : Script finished  OK
 12-May-2015 06:26:29  2080 103_000  GPSXTR   BAS_GEN    : Client started on: localhost
 12-May-2015 06:26:29  2080 111_000  RESRMS   BAS_GEN    : Waiting for CPU: ANY
 12-May-2015 06:26:29  2080 103_000  GPSXTR   BAS_GEN    : Script started
 12-May-2015 06:26:30  2080 103_000  GPSXTR   BAS_GEN    : Script finished  OK
 12-May-2015 06:26:30  2080 111_000  RESRMS   BAS_GEN    : Client started on: localhost
 12-May-2015 06:26:30  2080 111_000  RESRMS   BAS_GEN    : Script started
 12-May-2015 06:26:32  2080 111_000  RESRMS   BAS_GEN    : Script finished  OK
 12-May-2015 06:26:32  2080 YR:2010  BASTST              : Session finished OK
-------------------------------------------------------------------------------
 Sessions finished: OK: 1    Error: 0            Total Time:   00:07:23
```

The file lists all events recorded by the server in chronological order. For each script the time of issuing the client's start command (`Client started`), of establishing the communication between server and client (`Script started`), and of termination of the script (`Script finished`) is given. For finishing scripts as well as for the entire processed PCF a status string `OK` or `ERR` is issued.

CPU Statistics in the Output File

At the end of the output file some statistics for the processed session are produced and the major time consumers are indicated. The first section is related to the overall processing of the sessions. It contains three parts that are split for printing in this manual. The first part of the section contains the time–related data :

```
-------------------------------------------------------------- ...
                 Statistics concerning time in sec             ...
Session      BPE     CPU  =  PGM  +  Aux + Delay Queuing       ...
-------------------------------------------------------------- ...
BT102080     443     441     429      12       0       1       ...
Max                   42      41       0       0       0
PID_SUB          102_003 102_003 102_009 102_004 102_009
-------------------------------------------------------------- ...
```

The columns have the following meaning:

`BPE` time between "Server started" and "Session finished" in the BPE server output file

`CPU` sum of the intervals between "Script started" and "Script finished" over all scripts of a session in the BPE server output file

`PGM+Aux` time between "SCRIPT STARTED" and "SCRIPT ENDED" as indicated in the BPE protocol files, summed over all scripts of a session

`PGM` time between "PROGRAM STARTED" and "PROGRAM ENDED" as indicated in the BPE protocol files, summed over all scripts of a session

`Queuing` sum of the intervals between "Client started" and "Script started" over all scripts of a session in the BPE server output file

The values in the columns `CPU` and `PGM+Aux` are expected to be identical otherwise the connection between the BPE server and the clients has some limitations. The column `Aux` indicates how much time the BPE client did need for all other actions apart from running Bernese processing programs, i.e., the execution of the Perl sequences.

The next part of the section is the middle part of the session–wise statistics to report the status of the individual scripts that have been executed within the session:

```
... ------------------------------------- ...
...     Statistics concerning # of jobs   ...
... PID  +  SUB   =  OK + Error   Rerun   ...
... ------------------------------------- ...
...   3      12      15       0       0   ...
```

The columns provide the following information:

`PID` Number of non–parallel scripts in the PCF for the particular session (`SUB_PID` = _000)

`SUB` Number of parallel scripts in the PCF for the particular session (`SUB_PID` ≠ _000)

`OK` Number of scripts finished with the status "OK"

`Error` Number of scripts finished with the status "Error"

`Rerun` Number of scripts that have been repeated due to the `V_RERUN` indicator

The number of started jobs (PID+SUB) is in any case equal to the number of finished tasks (OK+Error). The status Error may either be related to an error in a script that has stopped the session processing or it may also be induced by a script flagged with CONT_ERR.

In the last part of the session processing statistics some timestamps are provided that are related to start and end of the session processing:

```
... ---------------------------------------------------
...
... Start                End                  Duration
... ---------------------------------------------------
... 12-May-2015 06:19:09  12-May-2015 06:26:32  00:07:23
```

The second section is related to the statistics for the individual scripts of the PCF. If more than one session has been processed, the maximum CPU time is accomplished by the corresponding session. The meaning of these numbers is the same as in the session–related statistics above:

```
PCF-related BPE statistics:
===========================
${U}/PCF/BASTST.PCF
------------------------------------------------------------------------------------
                          Mean time  in sec
PID  Script    OPT dir     CPU = PGM + Aux  Max CPU  Session  PID_SUB  Rerun  Error
------------------------------------------------------------------------------------
101  GPSESTAP  BAS_GEN       0     0     0        0   102080   101_000      0      0
102  GPSEST_P  BAS_GEN      36    35     0       42   102080   102_003      0      0
103  GPSXTR    BAS_GEN       0     0     0        0   102080   103_000      0      0
111  RESRMS    BAS_GEN       2     1     0        2   102080   111_000      0      0
------------------------------------------------------------------------------------
```

Particularly in this section, the Aux column might be of importance indicating the CPU consumption of the user scripts outside from the Bernese processing programs. The value should correspond to the expected activities in the related Perl scripts.

At the very end, a summary for the involved CPUs is given:

```
CPU-related BPE statistics:
===========================
${U}/PAN/USER.CPU
------------------------------------------------------------------------------
                           CPU time  in sec     Queuing time
CPU name  # of jobs    Total  Mean    Max   Min  Total    Max  Session  PID_SUB
------------------------------------------------------------------------------
localhost        15      441    29     42     0      1     0   102080  102_009
------------------------------------------------------------------------------
```

For each CPU the number of jobs is counted in column #jobs. The other columns are constructed as described above. In particular, the Queuing time (and its relation to the total CPU time) is of interest to verify whether the job distribution is efficient or not.

You may check whether the duration for each script corresponds to your expectation or otherwise identify inefficient configurations:

- Long time for the menu program to prepare INP files (e.g., slow disk access),
- slow connection between server and client,
- transfer of an unexpected amount of data,
- long queuing time for a certain CPU–type.

BPE Status File

The status file is updated every 5 seconds showing the BPE execution status. In interactive mode its content is displayed in a BPE window. In the example below, the first script has finished and the BPE is processing script 102 in parallel. The following two scripts are waiting for execution, the last four are skipped intentionally.

```
Status of BASTST.PCF at 12-May-2015 06:19:24

Session 102080: running
  101 GPSESTAP BAS_GEN     finished
  102 GPSEST_P BAS_GEN     running   <      (12 remaining)
  103 GPSXTR   BAS_GEN     waiting
  111 RESRMS   BAS_GEN     waiting
  112 RESCHK   BAS_GEN     skipped
  901 BAS_SUM  BAS_GEN     skipped
  991 BPE_CLN  PPP_GEN     skipped
  999 DUMMY    NO_OPT      skipped
```

When the BPE has terminated the status file shows the status (finished or error) for each processed session:

```
Status of BASTST.PCF at 12-May-2015 06:26:32

Session 102080: finished
```

22.10.2. Client Protocol and Log Files

Every BPE client writes its own *protocol file*. The unique name of this file is:

```
   ${TASKID}${YEAR}${SESSION}_${PID}_${SUB_PID}
```

The `${TASKID}` is set in "RUNBPE 3: Output Filenames". By default the protocol files reside in the campaign's `BPE` directory and have the extension `PRT`. They contain the most important information about the execution of the clients including a parameter list. The files also contain error and warning messages issued by Bernese programs (irrespective on the error output selection in the program panels), with `MSG` or `ERR` showing the status of messages:

```
 PROTOCOL FILE FOR BPE SCRIPT
 ----------------------------
 Campaign           : ${P}/INTRO
 Year               : 10
 Session            : 2080
 PCF name           : RNX2SNX.PCF
 Script name        : HELMCHK
 Path to executables: ${XG}
 Option directory   : R2S_FIN
 Process ID         : 514
 Sub-process ID     : 000
 Server host        : carina.unibe.ch
 Remote host        : carina.unibe.ch  (system pid: 19922)
 CPU name           : localhost
 Path to work area  : /scratch/local/bern52/BPE_RNX2SNX_42423_10_2080_514_000
 User name          : bern52
 PARAM1             : NEXTJOB
 PARAM2             : 511

 Date         Time      Run time  Pgm.time  Sta Program   Message
 ----------------------------------------------------------------------------
 12-MAY-2015  06:17:08  00:00:00            MSG RUNBPE.pm SCRIPT  STARTED
 12-MAY-2015  06:17:08  00:00:00            MSG HELMR1    PROGRAM STARTED
 12-MAY-2015  06:17:08  00:00:00  00:00:00  MSG HELMR1    PROGRAM ENDED
 12-MAY-2015  06:17:08  00:00:00            MSG HELMCHK   GOTO PID 511
 12-MAY-2015  06:17:08  00:00:00            MSG RUNBPE.pm SCRIPT  ENDED
 ----------------------------------------------------------------------------
```

In the example first a Bernese processing program (HELMR1) is executed (indicated by PROGRAM STARTED and PROGRAM ENDED). Afterwards a jump backward of the execution of the BPE is requested (GOTO PID 511). If the script is repeated, the corresponding protocol is also duplicated.

The standard output and error output of the BPE client and user scripts may be re–directed into a log file (e.g., by specifying --stdout and --stderr in the CPU control file — see Section 22.4). In this case the file will contain debug output from programs and scripts as well as system messages in case of a hard crash of a program or script. The name and default location of the log file is the same as the name of the corresponding protocol file but the default extension is LOG.

22.11. Error Handling

22.11.1. Checks when Preparing a BPE

The chance of errors (particularly during the preparation of a new BPE run) is relatively high due to the complexity of the system. However, if one follows a few basic recommendations it should not be too difficult to eliminate the errors rapidly:

1. Before trying to work with the BPE make sure that the *Bernese GNSS Software* is behaving correctly in non–BPE mode.
2. Make sure that the CPU control file (see Section 22.4) is correct. If remote–shell or secure–shell commands are used in the CPU control file, test these commands outside the BPE. Make sure that the commands run in background on UNIX. The submission of a command must be possible without interaction (e.g., without entering a password when using ssh).
3. Before preparing your own BPE runs, check that the examples delivered with the *Bernese GNSS Software* work properly.
4. Before running any new BPE in non–interactive mode, try the interactive mode first. The error messages in both modes are the same but in non–interactive mode one has to know where to find them. In interactive mode the menu program tries to display the error messages automatically. Already before actually starting the BPE server the menu program checks several things (particularly the Process Control File) for logical errors (e.g., required waiting for a non–existing script etc.). A number of checks are already performed when editing a PCF file using "Menu>BPE>Edit process control file (PCF)".
5. Test new user scripts outside the BPE as far as possible before incorporating them into the BPE. The syntax of a Perl script may, e.g., be checked using the command perl –c script.
6. Use the die– instead of exit–command for handling errors in scripts.
7. Check carefully that the program output files generated by the BPE are unique so that they are not mutually overwritten by other scripts running in parallel. Make sure that the output file names specified in the program input files correspond to the names of input files needed by successive programs.
8. Make sure that each BPE option directory contains the menu input files MENU*.INP. Remember that the BPE uses these files from the option directories and not those

from the general user environment ${U}/PAN. This is particularly important if you add, e.g., a drive variable to MENU_VAR.INP.

22.11.2. Where to Find Error Messages

Even if the recommendations above are followed, it is necessary to know where to look for an error message in case the BPE does not work correctly. The following files should be checked:

1. Standard output file and standard error file of the BPE server (the error file may be merged into the standard output file). The PID of the failed process may be found here.
2. Protocol files (see Section 22.10.2). Generally the errors in Bernese main programs are reported in the protocol files. Messages reporting the failure of a program or script are labeled with a string ERR.
3. Log–files (see Section 22.10.2). Generally the errors in user scripts or fatal errors in Bernese programs are reported in log–files if in the CPU control file the client's standard output and error output is re–directed to the log file (see Section 22.4).
4. In case of a failure (or if option "Do not remove temporary user environment" in panel "RUNBPE 2: Process Control Options" is enabled) the client's temporary environment in the directory ${T} (with name ${T}/BPE_*pcfname_port_year_session_PID_SUB_PID*, see Section 22.3.3) is not removed after termination of the script. This offers the possibility to inspect the content of the program input files used by the programs (before substitution of menu variables in subdirectory PAN, after substitution in subdirectory INP). Input files from subdirectory INP may be used (copy them to ${U}/PAN) to manually repeat the run of a program executed in the BPE (after setting session and campaign to the desired values).

22.11.3. Server Waits and No Clients Are Started

You may also encounter the case that nothing happens after starting a BPE, even no window reporting the status of the BPE opens (in case of a non–interactive BPE the status file is not updated):

1. Open the BPE output file using "Menu>Service>Browse program output". It may contain a list of inconsistencies within the PCF, a report of missing scripts or option directories, or an indication that the active campaign does not correspond to the BPE server variable V_CAMP.
2. When the BPE output file only contains an entry that the session was started, the BPE is still on the way to clean up the BPE subdirectory of your campaign where the protocol and log files will be created. Depending on the number of files in this directory it may take even a few minutes.

It may also happen that the status file is generated and/or the BPE status window opens but the server waits indefinitely and no client seems to start working:

1. A frequent cause for this situation is a CPU control file which wrongly indicates that too many jobs are still running after a BPE server crashed or has been killed. Reset

the CPU control file using "Menu>BPE>Reset CPU file". This action is also allowed while a BPE is running. The only consequence may be that for a short time more than the allowed number of jobs are submitted.

2. It might also happen that a certain BPE does not make any progress in the execution of the scripts. A potential reason may be that other BPEs are running on the same user account on the same CPU control file. This is not critical because the remaining scripts will be executed as soon as the necessary CPUs are available when the other processes are finished. If needed, a priority handling as described in paragraph on suspending PCFs in Section 22.4 can be established.
3. Option "List of suspended PCF" in "Menu>BPE>Edit CPU file" allows to suspend jobs running in a specific PCF. Check this panel whether the particular PCF is contained in the list of suspended PCFs.
4. Depending on the particular setup of the job submission in the CPU control file and the type of error, a problem in a script may cause an uncontrolled stop of a client without notification to the server. Check the log files for error messages from the scripts or the operating system.
5. Jobs cannot be submitted because the `rsh` or `ssh` commands require interaction (e.g., manually typing a password) or fail due to an inappropriate configuration of the remote login procedure. Check the `rsh` resp. `ssh` access to the host.
6. Jobs were submitted to hosts which are not alive or where the particular queue is inactive; the file system on the client host is not mounted or the NFS server is not available. Check your system.

22.11.4. Potential Problems when Running a BPE

When a BPE is running the interaction with the operating system may also introduce some problems. The most common are:

1. a Bernese processing program cannot access a certain file, e.g., because of an intensive access by several parallel running BPE scripts. In that case it is sufficient to repeat the corresponding script using the `V_RERUN` option (see Section 22.5.5).

   If a rerun of a script has been initialized, the protocol file confirms this by an entry looking like

   ```
    12-MAY-2015  06:17:08  00:00:00           RER RUNBPE.pm SCRIPT  RERUN: 1
   ```

   In particular, if a program with a long duration fails because of a real problem that is not solved simply by repeating the script, it is worth to check the protocol file.
2. The bookkeeping for the numbering of program output files in a generic mode (described in Section 21.7) is performed by lock files in the `OUT` directory of the campaign named `pgmnam.J_lk`. If several user scripts are trying to increase the program output number at the same time, the progress of the BPE may be significantly delayed. For that reason, it is recommended to remove remaining lock files from previous BPE runs or preferably, to *not* use generic program output names at all (in particular, when scripts may run in parallel).
3. The Bernese menu, and likely also some of the user scripts, may need to screen a directory of the campaign for a certain pattern of files to be selected for the

processing. This may delay the processing progress if there are too many files in one particular directory of the campaign.

4. In some cases a BPE client process may get lost (e.g., killed by a job queuing system when a certain criterion was exceeded or simply stopped unintentionally). Without restarting the entire BPE process this particular script may be relaunched by the BPE server if a file `${TASKID}${YEAR}${SESSION}_${PID}_${SUB_PID}.JOB` in the `BPE` directory of your campaign is found (typically, creating a copy of the corresponding protocol file by exchanging the extension from `PRT` to `JOB` is the smartest way to generate this file).
5. If you are developing your own programs and you are using them in the BPE environment, you have to specify alternative locations of individual executables using the option "Programs executed from special paths" in the `MENU_PGM.INP` of the option directory (see Section21.4.3). The alternative location of the executable is confirmed in the protocol file:

```
12-MAY-2015  06:17:08  00:00:00            MSG HELMR1    PROGRAM STARTED
12-MAY-2015  06:17:08  00:00:00            MSG RUN_PGMS  PROGRAM PATH: /home/user/myEXE
12-MAY-2015  06:17:08  00:00:00  00:00:00  MSG HELMR1    PROGRAM ENDED
```

It is worth to check the protocol files for this statement from time to time to verify whether all these exceptional paths are really used on intention.

22.11.5. Killing a BPE Run

In interactive mode there is a possibility to "kill" a running BPE server using the `Kill` button in the server window. However, one has to take into account a few consequences:

1. Already started clients continue to run (they loose the TCP/IP connection to the server only). If necessary, you may have to stop these jobs manually using operating system commands on all hosts (the corresponding process IDs are reported in the protocol files of the individual scripts). A "hard" way how to kill all clients on a local UNIX machine may, e.g., be the kill −9 −1 command. Be careful in using this command since it stops not only the BPE jobs but *all* jobs running on the account!
2. The user area `${T}` is not cleaned — one has to delete the temporary directories manually.
3. The CPU control file remains in a non–updated state. Using the "Menu>BPE>Reset CPU file" the CPU control file may be cleaned again.

If a BPE is started in non–interactive mode it may be difficult to find the correct server to kill if other BPEs or interactive jobs are running at the same time. In order to facilitate the identification of the BPE server a mechanism is available that allows to print the PID of the server to the error output when starting the BPE non–interactively. This mechanism is initiated by adding the keyword `PRINT_PID` with value `1` to the menu command file, see Section 21.10.4 and Table 21.3.

22.12. Bernese Program Options for Automated Processing

Several programs in the *Bernese GNSS Software* may write deletion files (or selection files) for automatic removal of bad data files, or to select files according to particular criteria which makes them particularly useful for automated processing.

22.12.1. Selecting and Excluding Stations for Processing

22.12.1.1. Preselection at RINEX Level

In many cases it is important to select a subset of observation files for processing. The tracking statistics of the RINEX observation files obtained from program RNXGRA (program description in Section 4.2.5) serve a first selection. The options in panel "RNXGRA 3: Options for RINEX File Selection" define conditions for a minimum number of observations in the RINEX files. Furthermore, the number of selected RINEX files may be limited. Either the "File with the list of selected RINEX files" may be used for the selection of RINEX observation files for import into Bernese format with program RXOBV3 or the "File with the list of unselected RINEX files" may be used to delete RINEX observation files in your campaign directory.

22.12.1.2. Exclude Observation File while Importing from RINEX Format

The next program that can exclude observation files is RXOBV3 (description in Section 4.2.3). For the automatic processing we recommend to set the "ACTIONS IN CASE OF INCONSISTENCIES" for all relevant checks of the correctness of the RINEX header information to SKIP. In case of an inconsistency, the corresponding RINEX file is then not converted to Bernese format. Furthermore, no data from a RINEX file are imported if it contains less than a "Minimum number of epochs requested per file". Stations that have been excluded in the section `TYPE 003:  HANDLING OF STATION PROBLEMS` of the station information file are also automatically excluded from the data import.

22.12.1.3. Exclude Stations with Problems in the Data Preprocessing

Stations with problems in one of the preprocessing steps should be excluded from further processing in a robust automatic processing. In the case of the receiver clock synchronization (program CODSPP, see Section 6.3) and the preprocessing of phase observations (program MAUPRP, see Section 6.5), the corresponding extraction programs (CODXTR and MPRXTR) check the results and generate a file with a list of Bernese observation files to be deleted to prevent problems in further processing steps. For more details we refer to the description of these programs in Chapter 6.

The program RESCHK (description in Section 6.6.3) can detect bad stations from the summary table of the program RESRMS that is generated from post–fit residuals (usually from the program GPSEST). The program generates a deletion list containing the observation files from the detected misbehaving stations.

In case of bad stations in a network of baselines, usually the entire network has to be rebuild using the program SNGDIF, after deleting the zero–difference observation files of the misbehaving station.

22.12.1.4. Selection of Stations for Processing

The program MKCLUS ("Menu>Service>Automated processing>Form clusters") defines clusters for the processing of Bernese zero–difference or baseline observation files. A special application is the selection of a predefined number of Bernese zero–difference observation files for a single cluster. Select GLOBAL in option "Strategy for zero difference observations" to get an optimum distribution of the stations following one of the criteria:

- GEOMETRY: optimize the distribution of the stations using the maximum sum of the squared distances between the stations.
- DENSITY: minimize the redundancy of observations from the stations for each satellite and epoch (This may consume a lot of computing time because for the station selection, all Bernese observation files have to be read several times.)
- NUM.OBS: select the observation files with the biggest number of observations.
- CLOCKS: select the best clocks (smallest RMS of a linear fit in the input file "Station selection criterion" that may be generated by program CCRNXC).

Further conditions (e.g., "Minimum number of observations per file" and "Maximum number of ambiguities per file") may be defined for all selected files.

The list of selected files can be used for baseline file creation in program SNGDIF. In contrast to RNXGRA, program MKCLUS takes care of observations that are marked by the preprocessing in the Bernese observation files (e.g., in a zero–difference MAUPRP–run or in RNXSMT).

22.12.1.5. Selection of Baselines for Processing

The program BASLST ("Menu>Service>Automated processing>Select baselines") selects a set of Bernese baseline observation files using the length, the percentage of resolved ambiguities, or the receiver type of the stations as criteria. This is particularly useful to define the baselines for the different ambiguity resolution strategies (see Chapter 8) in an automatic processing.

22.12.2. Forming Clusters of Observation Files

If a network with a lot of stations (baselines) has to be processed not all observation files can be processed in one GPSEST run. In that case the observation files may be grouped into clusters and processed independently. The normal equations of the individual clusters may be combined using the program ADDNEQ2 for most of the non–epoch parameters. Clock estimates from a zero–difference processing can be combined with the program CCRNXC.

22.12.2.1. Clusters of Bernese Baseline Observation Files

If the "Correlation strategy" in the program GPSEST is not set to CORRECT, the clustering of the baseline observation files has no impact on the solution because all correlations between the baselines are ignored. A simple method (initPar_Bl or initPar_Cl) from the Perl module `${BPE}/bpe_util.pm` can be applied to form the clusters, see Section 22.6.5.3.

If the CORRECT "Correlation strategy" is selected, all baselines of a region should be processed together in one cluster in order to consider a maximum of correlations. The *Bernese GNSS Software* offers two possibilities for clustering baseline observation files in an automated processing mode:

1. A cluster file (default extension CLU, see Section 24.7.18 for a file format description) is used when forming the baselines in the program SNGDIF. The cluster number of the first station in a baseline defines to which cluster the baseline belongs. See Section 6.4 for more details.
2. A number of baseline observation files can be assigned to regional clusters using the program MKCLUS. No cluster file is necessary in this case. Those baselines are grouped to a cluster for which the distances between all stations are minimal. Either the maximum number of baselines per cluster or the number of clusters can be specified in the program input panel "MKCLUS 3: Regional Cluster Definition Options (Single Differences)".

22.12.2.2. Regional Clusters for the Zero–Difference Processing

Processing zero–difference observation files in regional clusters is useful, e.g., for the computation of post–fit residuals for data screening. In that case the number of stations observing the same satellites should be as big as possible. The reliability of the computed residuals benefits from this redundancy. In the panel "MKCLUS 3: Regional Cluster Definition Options (Zero Differences)" the program MKCLUS provides the possibility either to define the maximum number of observation files in a cluster or the number of clusters. The distance between the stations in a cluster is minimized.

If the satellite clocks are estimated in a regional cluster, there are always satellites that are observed by one station only. The residuals for those observations will be zero but this tells nothing about the quality of the observations itself. For the screening of the post–fit residuals of all observations a "Minimum number of observations per satellite at each epoch" may be defined with a value bigger than 1. As a consequence, a number of observation files will be assigned to more than one cluster.

22.12.2.3. Global Clusters for the Zero–Difference Processing

The main application of processing zero–difference observation files in global clusters is the computation of receiver and satellite clocks. One global solution created in a single run of the program GPSEST is, of course, the best way for clock estimation because all correlations can be considered. In order to include more stations into the solution, the zero–difference observation files may be divided into a number of global clusters that are processed independently with program GPSEST. In each global cluster solution a different set of stations is used to estimate the satellite clocks. The program CCRNXC can be used to combine the clock corrections from the individual cluster solutions to a consistent global solution.

The program MKCLUS provides several options to generate global clusters from a list of zero–difference observation files. The "Number of clusters" and the "Maximal number of stations per cluster" can be specified by the user in panel "MKCLUS 3: Global Cluster Definition Options (Zero Differences)". If a large number of observation files are specified, a selection of stations that are used in the global clusters for processing is possible. The criteria are described above. In addition, a minimum number of observations to each satellite can be requested in option "Minimum number of observations per satellite at each epoch" to achieve a reliable satellite clock estimation.

22.12.3. Rejecting Stations from the Definition of the Geodetic Datum

In an automatic processing, it might be important to automatically reject reference stations from the definition of the geodetic datum if a significant difference is found between estimated and a priori coordinates. The reasons for this are bad tracking data, unannounced antenna change, or shift of the station due to an earthquake. Such outliers may be detected by comparing coordinates of reference sites obtained with a no–net translation condition with the a priori coordinates using program HELMR1 (see Section 10.7.2). The program can be used to identify outliers and to write a new station selection file containing only those stations that passed the outlier criterion.

If reference stations are identified as outliers, the datum definition for the final solution has to be repeated with the modified station selection list by jumping back to the appropriate position in the processing sequence. The jump can be triggered by the existence of an outlier list that may be written by program HELMR1. This quality monitoring feature is included in the double–difference processing example (`RNX2SNX.PCF`) delivered with the *Bernese GNSS Software* Version 5.2 (description in Section 23.4.2).

23. Processing Examples

23.1. Introduction

The distribution of the *Bernese GNSS Software* contains several ready-to-use examples using the Bernese Processing Engine (BPE), including all necessary user scripts, option directories, and the Process Control Files (PCFs). The BPE allows to automatically process data following a scheme laid out in the PCF. These examples are intended to be used for standard processing tasks of the *Bernese GNSS Software* and to verify the installation. Of course you are free to adapt the provided PCFs to your specific problems within the BPE–environment. A technical description of the BPE and their components is given in Chapter 22 .

This chapter first describes the dataset of the example campaign, informs how to run the example PCFs for verification of the software installation, and describes the seven PCFs for the examples:

- Precise Point Positioning (`PPP_BAS.PCF`).
  This PCF serves to generate a priori information for further processing.
- extended Precise Point Positioning (`PPP_DEMO.PCF`).
  This PCF is an extension of the basic version of the PPP with some additional features, like pseudo kinematic positioning, high–rate troposphere estimation, and generation of ionosphere models. This PCF may also serve as an example how other BPEs can be adapted to these applications (e.g., high–rate troposphere estimates in the double–difference network solution).
- double–difference network processing (`RNX2SNX.PCF`).
  This is the default procedure for most cases (coordinate or troposphere results).
- baseline test (`BASTST.PCF`).
  This is an example of a straightforward PCF for baseline–wise processing (e.g., ambiguity resolution or residual analysis),
- zero–difference network processing (`CLKDET.PCF`).
  It may be used for clock estimation.
- precise orbit determination for LEOs (`LEOPOD.PCF`).
  This is an example for orbit determination using GNSS data from GRACE that may serve as the basis to process the data from other LEO missions.
- SLR orbit validation (`SLRVAL.PCF`).
  This PCF can be used to compute residuals of SLR measurements with respect to existing orbits for GNSS or LEO satellites.

A more detailed list of the individual BPEs is given in Table 23.1 .

At the end, we describe how to process your own data with the PCFs provided. Thus, the furnished examples open the gates to highly sophisticated, state of the art GNSS data processing using one of the most powerful software packages.

Table 23.1.: List of main features for the processing examples.

Feature	PPP_BAS.PCF	PPP_DEMO.PCF	RNX2SNX.PCF	CLKDET.PCF	LEOPOD.PCF	SLRVAL.PCF
allow the combined GPS/GLONASS processing	X	X	X	X		X
full ambiguity resolution scheme with all steps for GPS and			X			
so far supported also for GLONASS			X			
optimized preprocessing of the zero–difference examples		X		X	X	
cover more applications by additional example BPEs					X	X
for demonstration purposes equipment changes and	X	X	X	X		
a repositioning event shall be included	X	X	X	X	X	X
kinematic or at least pseudo–kinematic processing,		X			X	
high–rate troposphere estimation, and		X				
ionosphere model determination		X				
prepared for processing many stations (clustering)	X	X	X	X		
support hourly (sliding window) and	X	X	X			
daily processing as well as	X	X	X	X	X	X
reprocessing (daily independent) schemes	X	X	X	X	X	

The examples are provided together with a dataset to run the BPEs. The only exception is the JPL DE 405 ephemeris needed for the orbit integration by ORBGEN. Please follow the instructions in the `${X}/DOC/README_JPL_EPH.TXT` file in order to create this file.

Each example PCF comes along with a `README`–file containing a general description and comments, the input and output files, description of the BPE server variables, and remarks on the most important numbers in the resulting protocol file. The `README`–files are located in `${U}/PCF` or `${X}/PCF`.

While going through the detailed description of the PCFs in this chapter, it is recommended to inspect the corresponding program input files via "Menu>BPE>Edit PCF program input files". For each option, you can access the integrated help for further information. *Be careful not to accidentally change options in these files.*

23.2. Description of the Data Set

23.2.1. Observation Data

Data from thirteen European stations of the IGS network and from the EPN were selected for the example campaign. They are listed in Table 23.2. The locations of these stations are given in Figure 23.1. Three of the stations support only GPS whereas all other sites provide data from both GPS and its Russian counterpart GLONASS.

The observations for these stations are available for four days. Two days in year 2010 (day of year 207 and 208) and two in 2011 (days 205 and 206). For eight of these stations, coordinates and velocities are given in the IGb 08 reference frame, an IGS–specific realization of the ITRF 2008 (see `${D}/STAT_LOG/IGb08.snx`).

Table 23.2.: List of stations used for the example campaign including receiver and antenna type as well as the antenna height.

			Receiver type		Antenna
Station name	Location		Antenna type	Radome	height
GANP 11515M001	Ganovce, Slovakia		TRIMBLE NETR8		
			TRM55971.00	NONE	0.3830 m
HERT 13212M010	Hailsham, United Kingdom		LEICA GRX1200GGPRO		
			LEIAT504GG	NONE	0.0000 m
JOZ2 12204M002	Jozefoslaw, Poland		LEICA GRX1200GGPRO		
			LEIAT504GG	NONE	0.0000 m
LAMA 12209M001	Olsztyn, Poland	2010:	LEICA GRX1200GGPRO		
			LEIAT504GG	LEIS	0.0600 m
		2011:	LEICA GRX1200+GNSS		
			LEIAT504GG	LEIS	0.0600 m
MATE 12734M008	Matera, Italy		LEICA GRX1200GGPRO		
			LEIAT504GG	NONE	0.1010 m
ONSA 10402M004	Onsala, Sweden		JPS E_GGD		
			AOAD/M_B	OSOD	0.9950 m
PTBB 14234M001	Braunschweig, Germany		ASHTECH Z-XII3T		
			ASH700936E	SNOW	0.0562 m
TLSE 10003M009	Toulouse, France	2010:	TRIMBLE NETR5		
			TRM59800.00	NONE	1.0530 m
		2011:	TRIMBLE NETR9		
			TRM59800.00	NONE	1.0530 m
WSRT 13506M005	Westerbork, The Netherlands		AOA SNR-12 ACT		
			AOAD/M_T	DUTD	0.3888 m
WTZR 14201M010	Kötzting, Germany	2010:	LEICA GRX1200GGPRO		
			LEIAR25.R3	LEIT	0.0710 m
		2011:	LEICA GRX1200+GNSS		
			LEIAR25.R3	LEIT	0.0710 m
WTZZ 14201M014	Kötzting, Germany	2010:	TPS E_GGD		
			TPSCR3_GGD	CONE	0.2150 m
		2011:	JAVAD TRE_G3TH DELTA		
			LEIAR25.R3	LEIT	0.0450 m
ZIM2 14001M008	Zimmerwald, Switzerland		TRIMBLE NETR5		
			TRM59800.00	NONE	0.0000 m
ZIMM 14001M004	Zimmerwald, Switzerland		TRIMBLE NETRS		
			TRM29659.00	NONE	0.0000 m

Table 23.3.: List of antenna/radome combinations used in the example campaign together with the available antenna calibration values (IGS 08 antenna model).

	Type of calibration		used at
Antenna type	for GPS	for GLONASS	stations
`AOAD/M_B    OSOD`	`ADOPTED from NONE`	`ADOPTED from GPS`	`ONSA`
`AOAD/M_T    DUTD`	`ADOPTED from NONE`	—	`WSRT`
`ASH700936E  SNOW`	`ROBOT`	—	`PTBB`

`LEIAR25.R3  LEIT`	`ROBOT`	`ROBOT`	`WTZR,`
			`WTZZ(2011)`

`LEIAT504GG  NONE`	`ROBOT`	`ROBOT`	`JOZ2, HERT,`
			`MATE`

`LEIAT504GG  LEIS`	`ROBOT`	`ROBOT`	`LAMA`
`TPSCR3_GGD  CONE`	`ROBOT`	`ROBOT`	`WTZZ(2010)`
`TRM29659.00 NONE`	`ROBOT`	—	`ZIMM`
`TRM55971.00 NONE`	`ROBOT`	`ROBOT`	`GANP`
`TRM59800.00 NONE`	`ROBOT`	`ROBOT`	`TLSE, ZIM2`

Between these days in 2010 and 2011 the receivers (LAMA, TLSE, WTZR) and the full equipment (WTZZ) were changed. The receiver type, the antenna type, and the antenna height are also provided in Table 23.2. Notice, that for three antennas (GANP, WTZR, ZIM2) values from an individual calibration are available from the EPN processing. For all other antennas only type–specific calibration results from the IGS processing (`${X}/GEN/IGS08.ATX`) are available. More details are provided in Table 23.3. Only in two cases — where no calibration of the antenna/radome combination was available (ONSA, WSRT) — the calibration values of the antenna without radome were used instead. Except for ONSA, even system–specific calibrations for GPS and GLONASS measurements are available.

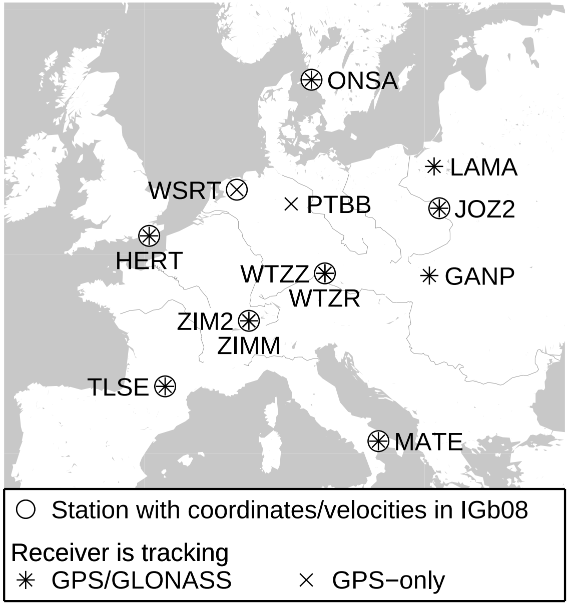


Figure 23.1.: Stations used in example campaign.

The distances between stations in the network are between 200 and 1000 km. There are two sites each hosting a pair of receivers included in the example dataset: in Zimmerwald, the distance between ZIMM and ZIM2 is only 19 m. In Kötzting the receivers WTZR and WTZZ are separated by less than 2 m — this is a short GPS/GLONASS baseline.

The receivers used at the stations MATE, ONSA, PTBB, and WSRT are connected to H–Maser clocks. The receiver type ASHTECH Z–XII3T used at PTBB was specifically developed for time and frequency applications. In 2011 both receivers in Kötzting (WTZR and WTZZ) were connected to the same H–Maser (EFOS 18).

23.2.2. Auxiliary Data

Besides the RINEX observation data, several other data files necessary for processing are distributed in the subdirectories of the DATAPOOL area (${D}):

RINEX: RINEX observation files for the GNSS ground stations
LEO: RINEX observation files and the corresponding attitude files (stored in separate subdirectories RINEX and ATTIT) — to be used by LEOPOD.PCF.
SLR_NP: SLR observation data in the normal point format (to be used by SLRVAL.PCF)
COD: Orbits, EOPs, and satellite clock corrections from CODE
BSW52: Ionosphere maps and DCB files
REF52: Reference frame files: IGB08_R.CRD, IGB08_R.VEL
Project–specific, station–related files: coordinate (CRD), velocity (VEL), station information (STA), abbreviation (ABB), cluster definition (CLU), tectonic plate assignment (PLD), and ocean (BLQ) as well as atmospheric (ATL) tidal loading corrections

In addition to these campaign–specific data, files from the ${X}/GEN directory are used (see the "General Input Files" panel of the processing programs). Let us mention here the user–specific files as well, such as the program input option panels in the ${U}/OPT, the PCFs in ${U}/PCF, and the scripts to be run by the PCFs in ${U}/SCRIPT directories. During the installation of the software, these files are copied from their original location in the ${X} structure to the user–specific directory structure GPSUSER. Chapter 24 contains a description for all Bernese file types. Examples for Bernese data files can be found in the ${X}/DOC directory.

23.3. Installation Verification Using the BPE Examples

The example PCFs are an excellent means to verify the completeness and correctness of the installation and the proper functioning of the software on your platform. We suggest to run the examples after installation and compare the obtained results with provided reference files (*.*_REF located in the SAVEDISK area, ${S}). At this point, please be sure that you have generated the ephemeris file ${X}/GEN/DE405.EPH, and that you have Perl available on your system.

23.3.1. How to Run the Examples

After installation, you will have to set the example campaign as active campaign:

- enter ${P}/EXAMPLE in the list of campaigns, "Menu>Campaign>Edit list of campaigns". You always have to specify the path (${P}) here.
- select the example campaign as active campaign: "Menu>Campaign>Select active campaign".

Use "Menu>Configure>Set session/compute date" to set the current session. Enter year 2010, and day of year 207 for the first of the 4 days of the example campaign. If necessary, specify SESSIONS for the session table, and specify 0 as session character. No other information is needed.

You can now run the first of the example PCFs, the PPP_BAS.PCF, for the first session by "Menu>BPE>Start BPE processing". The following four input panels are involved in this step:

RUNBPE 1: Here, you can verify the active campaign and session. No change in any options should be necessary.

RUNBPE 2: Verify the name of the "CPU control file". This should be USER.CPU[1]. Select the PPP_BAS.PCF in the option "Process control file". For now, the other fields may remain empty. Consult the online help for more information on these options.

RUNBPE 3: The option "Task ID" allows to specify a prefix to the BPE log files in the campaign's BPE directory, which is helpful to keep an overview of the runs. For the PPP_BAS.PCF, a natural choice would be, e.g., PP. The "Status file" displays the step–by–step progress of the BPE. This progress is displayed on screen, but it's often useful to save this information also in a file. An obvious choice at this point is PPP_BAS. A similarly evident name (PPP_BAS) may be chosen for the "Program output".

RUNBPE 4: Displays a list of variables defined in the PCF. Do not change any of these for now.

After filling in the options, the BPE can be started. The progress window will show up, displaying the status of the stepwise processing.

Further reading:

Chapter 3: "Directory Structure and Campaign Setup", page 57
Section 21.6: "Menu Variables", page 462
Section 22.4: "CPU Control File", page 484
Section 22.8.1: "Interactive Mode", page 513

23.3.2. Verification Using Provided Reference Solutions

After completion of the BPE, you can compare the results with the supplied reference solutions in the SAVEDISK area. For each "project" a separate directory is available (e.g., RNX2SNX). The files are located in individual subdirectories (SOL, ATM, STA, and ORB — all in yearly directories). We refer to ${S}/README_SAVEDISK.TXT for a detailed description of the structure.

It is important that the PPP_BAS.PCF for session 2070 of year 2010 is run first, otherwise your solutions will be based on other a priori coordinates (which has no impact on the results, but will lead to differences in several program outputs compared to the reference files). The other example PCFs are based on the PPP results for session 2070, year 2010. In addition, the zero–difference network solution (CLKDET.PCF) is expected to run after the double–difference network solution (RNX2SNX.PCF) because the resulting coordinates and troposphere parameters are introduced.

23.3.3. What to do in Case of Errors

The example PCFs will normally run without any error, provided that the installation was done correctly, no manual changes have been introduced in any files, and the required files

[1] The CPU control files may be adapted to the user's computer system according to the description in Section 22.4.

are available. Nevertheless, errors may (and sooner or later almost certainly will) occur, mainly when you run the BPE on your own data, or have modified the examples according to your needs. In Section 22.11 we provide numerous hints on how to solve a variety of possible problems.

23.4. Description of the Processing Examples

The PCF contains a list of scripts from the `${U}/SCRIPT` directory to be executed in a well–defined order by the BPE server. These scripts constitute modular building blocks, allowing to construct complete processing schemes (PCFs) tailored to specific needs. In essence, a script runs one (or more) Bernese program(s) (e.g., the script `COOVEL` runs the Bernese program COOVEL). Program input options are taken from the directory specified in the PCF. The BPE, furthermore, allows to control the processing flow of the user scripts (sequence, loops, interdependencies, parallelization, etc.). BPE variables, which can be used in the program input option panels, e.g., for filename creation, are specified in the PCF. These variables always have the prefix `V_`. For details the reader is referred to Chapter 22.

The following sections on the example PCFs provide an overview, describe the input and result files, explain the BPE variables, comment the scripts in logically connected blocks, and explain the protocol file for the quality control.

23.4.1. Precise Point Positioning (basic version: `PPP_BAS.PCF`; extended version: `PPP_DEMO.PCF`)

23.4.1.1. Purpose and Description

The purpose of both PCFs is to perform a Precise Point Positioning (PPP) using RINEX GNSS observations from a number of stationary ground receivers to obtain a set of station coordinates at a cm–level. This will be used as a priori information for the double–difference analysis done by `RNX2SNX.PCF`, described in Section 23.4.2. In addition, station–specific troposphere parameters and phase–consistent receiver clock offsets (typically at 5–minute intervals) for time transfer are computed.

It includes an update procedure to prepare files for new stations:

- update of station name abbreviation (`ABB`) files,
- computation of a priori velocities from NUVEL1A–model,
- propagation of station coordinate results to a predefined epoch,
- generation of a merged coordinate (`CRD`/`VEL`) file,
- detection of missing station entries in ocean and atmospheric tidal loading correction tables (`BLQ`, `ATL`).

Depending on the session table in the campaign, either daily or hourly files are processed (see Figure 3.3). In the latter case several hourly observation files are concatenated in a sliding window scheme.

On top of this functionality as it is provided by the basic version (`PPP_BAS.PCF`), the extended version (`PPP_DEMO`) may be used to estimate:

- geocenter coordinates from all datum stations included in the PPP procedure,
- pseudo–kinematic coordinates,

- high–rate troposphere parameters (15 minutes sampling), and
- station–specific/regional ionosphere models together with the corresponding DCB (P1−P2) parameters.

These additional features may be enabled/disabled by specifying a non–blank/blank solution identifier for the corresponding PCF variable.

Furthermore, this BPE demonstrates the two possibilities to preprocess phase data for zero–difference processing:

- RNXSMT: consistency check with the code measurements based on Melbourne–Wübbena linear combination (requires dual frequency data)
- MAUPRP: a sophisticated interpretation of epoch–difference residuals based on PPP (requires precise satellite clock corrections with the same sampling as that of the data)

These additional blocks in the extended version may easily be transferred into another double–difference (derived, e.g., from `RNX2SNX.PCF`, see Section 23.4.2) or zero–difference (based, e.g., on `CLKDET.PCF`, see Section 23.4.4) processing scheme.

Both PCFs write a processing summary file `PPPyyssss.PRC` (`yy`=year, `ssss`=session), which is described in Section 23.4.1.7.

23.4.1.2. Input Files

The input files are copied from the `DATAPOOL` area (`${D}`) into the campaign area in the BPE user script `PPP_COP`. With one exception the list of input files is identical for the basic as well as for the extended version of the PPP example. Note that most of the filenames and paths are constructed from BPE server variables described in Section 23.4.1.5:

- **External products files** (mandatory)

  `${D}/{V_B}/{V_B}wwwwd.PRE`
  `${D}/{V_B}/{V_B}wwwwd.CLK`
  `${D}/{V_B}/{V_B}wwww7.ERP` or `${D}/{V_B}/{V_B}wwwwd.ERP`

  The variable `V_B` defines the product series to be considered for the PPP (e.g., `COD` or `IGS`), where `wwwwd` stands for the GPS week and the day of week to be processed.

  For a successful PPP it is *essential* that orbit, ERPs and satellite clock corrections are *fully consistent*.

- **Bernese formatted products files** (mandatory)

  `${D}/BSW52/P1C1yymm.DCB`

  The DCB corrections are necessary to unify the reference for the different receiver tracking techniques (`yymm` is the 2-digit year and the 2-digit month of the processing session).

  `${D}/BSW52/CODwwwwd.ION`

  The ionosphere file is needed to correct for the HOI terms.

  Both files can be downloaded from `http://www.aiub.unibe.ch/download/CODE/`.

- **Reference frame files** (mandatory if V_UPD == 'Y')
  If the reference frame files shall be updated (enabled by the BPE server variable V_UPD == 'Y'), the reference frame coordinate and velocity files are needed:

  ```
  ${D}/{V_REFDIR}/{V_REFINF}_R.CRD
  ${D}/{V_REFDIR}/{V_REFINF}_R.VEL
  ```

  If the translation parameters (GCC, `V_G` is not blank) shall be computed in the extended version (`PPP_DEMO.PCF`), a list of reference frame stations is also required:

  ```
  ${D}/{V_REFDIR}/{V_REFINF}.FIX
  ```

- **Files to be updated/created on request** (if V_UPD == 'Y')

  ```
  ${D}/{V_REFDIR}/{V_CRDINF}.CRD
  ${D}/{V_REFDIR}/{V_CRDINF}.VEL
  ```

  If these files are available, they are copied into the campaign area. They are updated during the BPE with the PPP results (`CRD`) considering the NUVEL–1A velocity. For that purpose, the assignment to a tectonic plate is required from the corresponding file (mandatory if V_UPD == 'Y')

  ```
  ${D}/{V_REFDIR}/{V_CRDINF}.PLD
  ```

  The station abbreviation table is automatically updated for the stations processed in this PPP run if it already exists (otherwise it is created from scratch)

  ```
  ${D}/{V_REFDIR}/{V_CRDINF}.ABB
  ```

- **Station naming/equipment checking** (optional)
  If the station information file and/or a RINEX inconsistency file

  ```
  ${D}/{V_REFDIR}/{V_CRDINF}.STA
  ${D}/{V_REFDIR}/{V_CRXINF}.CRX
  ```

  are available, they are considered for the PPP and a warning in the PPP protocol file is issued in the case of an inconsistency.

- **Tidal loading corrections** (optional)

  ```
  ${D}/{V_REFDIR}/{V_BLQINF}.BLQ
  ${D}/{V_REFDIR}/{V_ATLINF}.ATL
  ```

  Ocean and atmospheric tidal loading may be corrected during the PPP procedure if the corresponding corrections are provided in the files.

  These files have also to contain corresponding center of mass corrections (CMC) values for the translation of the orbit positions from the Earth-fixed system in the center-of-mass system for the orbit integration.

  If this is the case the PPP protocol file will also contain a list of stations that are still missing in the two tables of corrections.

23.4.1.3. Observation Selection

The observation files are expected in RINEX format in the directory `${D}/{V_RNXDIR}/` (version 2) and/or `${D}/{V_RX3DIR}/` (version 3). The station/observation file selection is realized in the BPE user script `RNX_COP`.

Three possibilities to select stations for processing are implemented (considered in the following order):

1. A file with the name {V_OBSSEL}.CLU exists in the STA directory of the current campaign. The name of the cluster file is defined by the BPE server variable V_OBSSEL (see Section 23.4.1.5).

   All station names from this file are translated into the 4-character IDs using the abbreviation table {V_CRDINF}.ABB (also expected to be in the STA directory of the current campaign). These 4-character IDs are used to identify corresponding RINEX files from the source directory in the data base.

2. A file named {V_OBSSEL} resides directly in the top–level directory of your current campaign. It contains the RINEX abbreviations of the files to be processed. They have to be separated by blank characters or may start with a new line:

   ```
   GANP
   HERT
   JOZ2
   LAMA
   ZIM2 ZIMM
   ```

   (In this example, six stations are selected for the processing).

3. If the BPE server variable V_OBSSEL is empty all RINEX files of the specified session(s) are processed.

The selected RINEX files are copied and prepared for processing by the subroutine crz2rnx provided by the Perl module ${BPE}/bpe_util. The files are decompressed using the Hatanaka tool and/or any other usual compressing tool. For this purpose, the Hatanaka decompression tool CRX2RNX (Unix, Mac OS) or CRX2RNX.exe (Windows) is expected to be installed on your system and their location must be included in the system $PATH or %PATH% variable — the search path for executable programs. The same is expected for the decompression tool gzip or gzip.exe to handle the potential generic compression of the files.

Usually, only the RINEX files of the current session are processed. If you are going to process hourly sessions (session identifier must end with A..X — see method isHourly in ${BPE}/bpe_util.pm), all hours defined by the range of the last V_HOURS (see BPE server variables, Section 23.4.1.5) are concatenated and processed together under the label of the current session (of course the corresponding input files are copied, too).

Depending on the content of the BPE server variables V_RNXDIR and V_RX3DIR the corresponding RINEX files are considered in the related format versions 2 or 3 or even both. In case of RINEX version 3 the conventional filenames as well as the new long filenames are supported. If multiple versions of the same dataset are available, the selection is based on the given priority list defined in user script RNX_COP. The following preference list (in descending order) was defined:

1. long filename, RINEX version 3; generated by the receiver
   (indicated by data source flag R)
2. long filename, RINEX version 3; extracted from a data stream
   (indicated by data source flag S)
3. long filename, RINEX version 3 from an unknown source
   (indicated by data source flag U)
4. short filename, RINEX version 3 from an unknown source
5. RINEX version 2 (only short filenames available)

This priority list can of course be adapted according to your needs by modifying the value of the variable

```
  my @rx3Typ = ("R","r","S","s","U","u","short","rx2");
```

in the user script RNX_COP.

23.4.1.4. Result Files

The most important result files are copied in the BPE user script PPP_SAV into the ${S}/{V_RESULT} area (V_RESULT is defined below in the BPE server variables area) if the BPE server variable V_SAV == 'Y'. The files are located in yearly subdirectories (yyyy). These files are:

- ${S}/{V_RESULT}/yyyy/OUT/PPPyyssss.PRC
  Main protocol file containing the summary of the processing steps (more details are provided in the Section 23.4.1.7).
- ${S}/{V_RESULT}/yyyy/STA/{V_C}yyssss.CRD
  Resulting station coordinates for the stations from the PPP processing (epoch of the processed data).
- ${S}/{V_RESULT}/yyyy/ATM/{V_C}yyssss.TRP
  ${S}/{V_RESULT}/yyyy/ATM/{V_C}yyssss.TRO
  Troposphere estimates (Bernese troposphere format and troposphere SINEX format) as obtained from the PPP processing.
- ${S}/{V_RESULT}/yyyy/OUT/{V_C}yyssss.CLK.Z
  Clock RINEX file (UNIX compressed) with the receiver clock corrections for the stations from the PPP processing together with the used satellite clock corrections.
- ${S}/{V_RESULT}/yyyy/SOL/{V_C}yyssss.NQ0.gz
  Normal equation file (gnu–compressed) with the coordinate and troposphere parameters of all stations of the PPP processing.
- ${S}/{V_RESULT}/yyyy/SOL/{V_F}yyssss.NQ0.gz
  ${S}/{V_RESULT}/yyyy/SOL/{V_F}yyssss.SNX.gz
  Reduced version of the normal equation with pre–eliminated troposphere parameters containing, e.g., only the coordinate parameters.

The BPE server variables V_C and V_F are set by default to the identifiers PPP and RED, see Section 23.4.1.5.

The following list of result files is only generated by the extended version of the PPP example (PPP_DEMO.PCF). The corresponding BPE server variables must be defined with non–blank values to enable the generation of the related files:

- ${S}/{V_RESULT}/yyyy/STA/{V_G}yyssss.CRD
  ${S}/{V_RESULT}/yyyy/STA/{V_G}yyssss.GCC
  Coordinate and geocenter coordinates (translations of the PPP solution with respect to an external reference frame given in V_REFINF). These files are only available if V_G has a non–blank value.
- ${S}/{V_RESULT}/yyyy/OUT/{V_H}ssss{stat}.SUM
  ${S}/{V_RESULT}/yyyy/STA/{V_H}ssss{stat}.KIN

Kinematic coordinates (one independent set of station coordinates per epoch) and the GPSEST output summary reporting the deviation of the estimated coordinate per epoch from the static PPP solution with only one set of coordinates per processing interval together with some other statistical information. One pair of files will be stored per station in the PPP process. The generation of these results is enabled by defining a non–blank value for the PCF variable `V_H`.

- `${S}/{V_RESULT}/yyyy/ATM/{V_I}yyssss.TRP`
`${S}/{V_RESULT}/yyyy/ATM/{V_I}yyssss.TRO`
`${S}/{V_RESULT}/yyyy/STA/{V_I}yyssss.CRD`
  Result files for the high–rate troposphere estimation (internal Bernese and troposphere SINEX format) together with the estimated coordinates for validation purposes. A non–blank value for `V_I` is necessary to produce these results.
- `${S}/{V_RESULT}/yyyy/ATM/{V_K}yyssss.ION`
`${S}/{V_RESULT}/yyyy/ORB/{V_K}yyssss.DCB`
  Station–dependent ionosphere model in Bernese format (concatenated to one file per session) with the corresponding DCB estimates. These files are only available for a daily processing scheme and if `V_K` has a non–blank value.
- `${S}/{V_RESULT}/yyyy/ATM/{V_L}yyssss.ION`
`${S}/{V_RESULT}/yyyy/ATM/{V_L}yyssss.INX`
`${S}/{V_RESULT}/yyyy/ORB/{V_L}yyssss.DCB`
  A regional ionosphere model is generated from the observation files cleaned during the PPP process. The model is provided in the Bernese and IONEX format together with the resulting DCB estimates. These results are only generated if `V_K` has a non–blank value and if the daily processing scheme was used.

23.4.1.5. BPE Server Variables

There exist several variables having a default value in the PCF (last section of the file). When starting a BPE manually they may be modified in the fourth panel "RUNBPE 4: Server Variables Sent to Client".

Input and Result Filenames and Their Locations

This group of BPE server variables manages the directories of the input data:

V_REFDIR (default: REF52)
Directory in the `${D}` (`DATAPOOL`) area where the basic Bernese files for the processing are located. The detailed list of files is given in the section 23.4.1.2 on input files.

V_CRDINF (default: EXAMPLE)
The basename of the files used as basic Bernese input files for this PPP procedure. Copying these files is managed in the BPE user script `PPP_COP`.

V_STAINF (default: EXAMPLE)
Usually, the header information in the RINEX observation files is compared with the expected equipment information provided in the station information file. The basename of this file is given in this variable. It will be copied from `${D}/{V_REFDIR}/{V_STAINF}.STA` into your campaign by the script `PPP_COP`.

For a "quick–look" PPP this variable may be empty to skip the check of the RINEX header information. In this case, the station names in the resulting coordinate files or in a potentially created abbreviation table may be different than in a later processing when using this file. It may also happen that the receiver or antenna name is misspelled in the RINEX observation file header and that for this reason no entry in the antenna phase center offset or receiver information file is found.

V_CRXINF (default: empty)
Inconsistencies between the RINEX observation file headers and the station information file can be recorded in a file with "Accepted station information" inconsistencies in program RXOBV3. If you need such a file for your processing, you can specify its name here. It will be copied from `${D}/{V_REFDIR}/{V_CRXINF}.CRX` into the campaign.

V_BLQINF (default: EXAMPLE)
The basename of the file containing the ocean tidal loading corrections for the stations. The file has also to contain the CMC related to ocean tidal loading. These corrections are needed for orbit generation.

If the variable is empty, no displacement corrections to the stations and no CMC for the orbit generation are applied. The file `${D}/{V_REFDIR}/{V_BLQINF}.BLQ` is copied by the script `PPP_COP` into the campaign.

V_ATLINF (default: EXAMPLE)
The basename of the file containing the atmospheric tidal loading corrections for the stations. The file has also to contain the CMC connected to atmospheric tidal loading. These corrections are needed for orbit generation.

If the variable is empty, no displacement corrections to the stations and no CMC for the orbit generation are applied. The file `${D}/{V_REFDIR}/{V_ATLINF}.ATL` is copied by the script `PPP_COP` into the campaign.

V_HOIFIL (default: HOI$YSS+0)
Specify the filename used within your campaign area for the ionosphere model needed to compute the HOI corrections. The ionosphere model is copied from `${D}/BSW52/CODwwwwd.ION` into the `ATM`–directory of the campaign using the given filename.

An empty value for this variable disables the HOI corrections.

The following BPE server variables are related to the reference frame to be used for the processing:

V_REFINF (default: IGB08)
Reference frame files, used as master files when updating the project–specific coordinate and velocity files (`${D}/{V_REFDIR}/{V_CRDINF}.CRD` and `${D}/{V_REFDIR}/{V_CRDINF}.VEL`). The corresponding files are expected to be

```
${D}/{V_REFDIR}/{V_REFINF}_R.CRD
${D}/{V_REFDIR}/{V_REFINF}_R.VEL
```

V_REFEPO (default: 2005 01 01)
The epoch of the coordinates in the reference frame coordinate file (see above) needs to be specified here. In the case of IGS b08 it is "2005 01 01".

Another BPE server variable defines the location of the result files:

V_RESULT (default: PPP)
Directory in the `SAVEDISK` area (`${S}`) where the result files from the PPP processing are collected in yearly directories (saving of the result files is managed by the BPE user script `PPP_SAV`).

To write SINEX and/or IONEX result files the header information (e.g., institution name, hardware used, contacts, and information about the project) is taken from a skeleton file.

V_SNXINF (default: SINEX.PPP)
With this variable the SINEX header file may be selected for all BPE scripts where a SINEX file (with station coordinates) or a troposphere SINEX file is generated.

V_INXINF (default: IONEX.PPP)
The IONEX control file containing general information and some options may be selected with this variable for all BPE scripts where a IONEX file is generated.

Observation Selection

There are three variables related to the selection of the observation files. The procedure is described in Section 23.4.1.3.

V_RNXDIR (default: RINEX)
Directory in the `${D}` (`DATAPOOL`) area where the RINEX files in format version 2 are expected.

V_RX3DIR (default: empty)
Directory containing RINEX files in format version 3 in the `${D}` (`DATAPOOL`) area.

V_OBSSEL (default: empty)
This variable may contain a filename for a cluster file or a file with the RINEX station abbreviations.

The following variables are responsible for the data selection:

V_OBSINF (default: OBS.SEL)
This variable contains the filename of the receiver type related observation types priority list which is used when importing `RINEX3` data files, located in the the `${X}/GEN/` directory. The observation selection is done in RNXSMT. More details are given in Section 24.3.7 .

V_SATSYS (default: GPS)
Select the system(s) (usually GPS or GPS/GLO) to be used for the PPP processing. The value of this variable is directly introduced in the option "Satellite system to be considered" in panel "RXOBV3 2: Input Options 1" (user script `RXOBV3_H`).

Note: whether all selected systems are really used for the processing depends also on the availability of satellite orbits and clock corrections as well as on the content of the RINEX observation files. This option acts only as a filter to remove non–selected systems from the observation files. As a matter of fact, the only reason why the default value is GPS, is the lack of GLONASS satellite clock corrections.

V_SAMPL (default: 300)
Sampling interval for the final PPP solution: The value of this variable is directly introduced by the option "Sampling interval" in panel "GPSEST 3.1: General Options 1" (user script `PPPEDT_P`, process ID in the Bernese Processing Engine (PID) 502). If advanced preprocessing of the phase observations is not enabled (V_HRCLK /= 'Y') already the observations imported into the Bernese observation files are reduced by applying this option in addition to "Sampling interval" in panel "RXOBV3 2: Input Options 1" (user script `RXOBV3_H`, PID 222).

Note: ultimately, only epochs for which satellite clock corrections are available can be processed in the PPP procedure.

V_OBSTYP (default: BOTH)
You may select whether only the CODE, only the PHASE, or BOTH measurement types together shall be used for the PPP procedure.

V_HOURLY (default: 8)
In case of an hourly session definition, a number of hours can be specified that are prepended to the current session to stabilize the estimation of ambiguity parameters. It should not be much shorter than the length of a satellite path, e.g., 6 hours.

This variable is obsolete if a daily session definition is given.

General Model and Standard File Selection

In this sequence the BPE server variables related to the general processing files are listed:

V_PCV (default: I08)
Selection of the antenna phase center model identifier.

V_SATINF (default: SATELLIT)
Basename of the satellite information file. The resulting filename is `${X}/GEN/{V_SATINF}.{V_PCV}`.

V_PCVINF (default: PCV)
Basename of the antenna phase center correction file. The resulting filename is `${X}/GEN/{V_PCVINF}.{V_PCV}`.

V_SATCRX (default: SAT_$Y+0)
Name of the satellite problem file.

V_RECINF (default: RECEIVER.)
This variable contains the filename of the receiver characterization file located in the the `${X}/GEN/` directory. More details are given in Section 24.3.6.

Another capability of the example BPE is to update the antenna phase center correction file based on an ANTEX file. This feature is managed by a BPE server variable as well:

V_MYATX (default: empty)
Filename (including extension) for an ANTEX file containing receiver and possibly satellite antenna phase center corrections. The file is used to update the Bernese phase center file in `${X}/GEN/{V_PCVINF}.{V_PCV}`, e.g., in case of new antennas in the network.

It is expected that the ANTEX file is available at `${X}/GEN/`.

If you plan to use a file containing individually calibrated antennas, you need to change the settings for "Set number of generic antennas to 999999" in panel "ATX2PCV 2: ANTEX Conversion" (Script 002 `ATX2PCV`):

- checkbox unmarked means "type–specific calibrations"
- checkbox marked means "antenna–specific calibrations"

See also Section 18.2.4: "Technicalities on the Receiver Antenna Handling".

Selection of the Preprocessing Mode, Extended Version only

You may select how the phase measurements will be preprocessed. You have two options depending on the availability of satellite clock corrections:

V_HRCLK (default: YES, if possible)
NO checks only the consistency between the code and phase observations which limits the level of screening to the noise level of the code data.
YES uses a more involved algorithm but requires high–rate (30 seconds) and precise satellite clock corrections.

File and Product Naming

The following group of BPE server variables is used as solution identifiers, consequently used for file naming within the BPE processing:

V_B (default: COD)
Name of the external product for GNSS orbit, ERP, and satellite clock information. It is expected that the products are available in the `DATAPOOL` area (`${D}`) in a specific subdirectory `V_B`; the filenames must also contain this label: `{V_B}wwwwd` with `wwww` being the GPS week and `d` being the day of week (see also Section 23.4.1.2 on input files).

V_A (default: APR)
Station coordinates, improving with each processing step.

V_C (default: PPP)
Label of the program output and result files from the PPP process (`CRD`, `TRP`, `TRO`, `SNX`, `CLK`, `NQ0`).
It is used for the solution files of each individual station as well as for the combined files for each session.

V_F (default: RED)
Label of the program output and result files of a reduced version of the combined PPP normal equation file of the current session containing only station coordinates (troposphere and clock parameters are pre–eliminated).

V_E (default: REF)
When updating a given set of coordinates, the PPP results for the station coordinates are transferred to the same epoch. This coordinate set has the label defined in V_E.

The following group of solution identifiers is only available in the extended version of the PPP (`PPP_DEMO.PCF`). They are also used as switches to select certain optional parts of the BPE. If there is a non–blank value the corresponding part is executed, in case of blank value it is skipped:

V_G (default: GCC)
Label of the program output of the normal equation solution estimating translation parameters of the PPP results with respect to a given reference frame.

V_H (default: KIN)
Label of the program output, extraction, and result files of the kinematic PPP solution.

V_I (default: TRP)
Label of the program output and result files of the solution with high–rate troposphere parameters (one parameter every 15 minutes).

V_K (default: ION)
Label of the program output and result files of the station–specific ionosphere model and the corresponding DCBs. Note: the ionosphere estimation is only available when using a daily processing scheme.

V_L (default: RIM)
Label of the program output and result files of a regional ionosphere model and the corresponding DCBs obtained from the observation files cleaned in the PPP process. Note: this variable is not a separate switch to enable/disable but depends on the enable/disable decision for variable `V_K`.

Management of the Dataflow

The last group of BPE server variables is related to the file handling in the campaign:

V_UPD (default: Y)
Update the project–specific files in the `${D}/{V_REFDIR}` area with the results from the PPP process.

V_SAV (default: Y)
Save result files into the `${S}/{V_RESULT}` area if V_SAV == 'Y' .

V_SAVOBS (default: Y)
If V_SAVOBS == 'Y' the code and/or phase (depending on the value for the PCF variable `V_OBSTYP`) zero–difference observation files from the current session are copied into session–specific directories of the `SAVEDISK` area: `${S}/{V_RESULT}/yyyy/OBS/yyssss` .

V_DEL (default: Y)
Delete all files of the currently processed session from the campaign if V_DEL == 'Y' . Usually, the deletion should be enabled to keep the campaign clean. For debugging purposes and to identify sources for errors in the processing, it may be useful to keep all files.

Skip Sessions in Case of Reprocessing

In case of a reprocessing, it might be interesting to repeat only selected sessions for some reasons. A typical scenario could be that the BPE is executed once for all sessions. Possibly, some sessions will end with an error or with unsatisfying results, e.g., because of inconsistent equipment records between the station information file and the RINEX header for some stations. After fixing these problems, the reprocessing needs to be repeated only for a certain subset of sessions. Instead of starting the BPE manually for each of these sessions, it is more convenient to start the BPE again for the full interval of the reprocessing and let the existence of a specific file in the `SAVEDISK` area indicate whether a session shall be repeated or not.

V_SKIP (default: PPP$YSS+0.PRC; deactivated by PPP$YSS+0.PRC_)
The very first script SKIP_SES (PID 000) checks whether a file `${S}/{V_RESULT}/`

yyyy/OUT/{V_SKIP} exists, e.g., from a previous processing. The availability of the file indicates the successful processing of the session in a previous iteration of the reprocessing. Therefore, the BPE will directly jump to the last script (DUMMY; PID 999). If the indicated file does not exist, the BPE will execute all relevant scripts for the specific session.

We propose to use the protocol file in the SAVEDISK area as an indicator.

If a session shall be repeated even if it was executed up to the last script in a previous iteration, the indicator file (given in the V_SKIP variable) must be removed in the SAVEDISK area.

If all sessions shall be repeated regardless of the existence of the indicator file, the value of this variable can be left empty or at least be modified, e.g., from PPP$YSS+0.PRC to PPP$YSS+0.PRC_ .

23.4.1.6. Description of the PCF

Copy Required Files and Prepare the Hourly Processing

Prior to starting the processing, all necessary files need to be copied from the datapool into specific directories in the campaign area. In addition, some files are merged for the hourly processing scheme.

```
#
# Skip session (reprocessing)
# ---------------------------
000 SKIP_SES PPP_GEN           ANY      1
#
# Copy required files
# -------------------
001 PPP_COP  PPP_GEN           ANY      1 000
002 ATX2PCV  PPP_GEN           ANY      1 001
011 RNX_COP  PPP_GEN           ANY      1 001
021 OBSMRGAP PPP_GEN           ANY      1 011
022 OBSMRG_P PPP_GEN           ANY      1 021
031 ION_MRG  PPP_GEN           ANY      1 011
099 DUMMY    PPP_GEN           ANY      1 002 022 031
```

PID 000 SKIP_SES: This script checks whether the file ${S}/{V_RESULT}/yyyy/OUT/{V_SKIP} exists. The values for V_RESULT and V_SKIP are taken from the BPE server variables, whereas yyyy is the 4–digit year of the current session. If the file exists the BPE directly jumps to the last script. This mechanism may be used, e.g., to manage whether a specific session shall be repeated in the frame of a reprocessing or not.

PID 001 PPP_COP: This script copies the needed files (apart from the RINEX files) into the respective campaign directories. If appropriate, filenames are constructed based on BPE variables and session–specific time variables. The script distinguishes between mandatory and optional files. If a mandatory input file is missing, the BPE will stop with an error.

Note that only those files are copied that have been changed according to their modification date.

PID 002 ATX2PCV: If the BPE server variable V_MYATX contains the name of an ANTEX file, this file is used to update the "PCV information file" specified by the BPE variables V_PCVINF and V_PCV. The antenna names together with the GNSS supported by the related receiver types are read from the "Station information file" (defined by V_STAINF).

If the resulting "Bernese phase center offsets" file differs from the file currently used for the processing in ${X}/GEN, it is copied to this location.

PID 011 RNX_COP: The algorithm to select RINEX observation files, described in Section 23.4.1.3, is implemented in this script. It copies the selected files from the datapool into the campaign area and applies the Hatanaka decompression program if necessary.

The script checks whether an hourly processing scheme is applied. If this is the case the hourly observation files have to be merged using the next group of scripts. This is not needed for a daily processing scheme where the execution may continue with PID 099 .

If no observation file was found for the corresponding session, the processing ends with PID 999 . This allows to run through all sessions independent of whether data for all sessions are available or not.

PID 021 OBSMRGAP: This script and the following one form a unit. This first script prepares the parallelization, the actual processing is done in the next PID . The task of this step is to concatenate hourly RINEX files in case of an hourly processing scheme and/or to cut the observations according to the current session definition. The script deletes files that may be present from previous runs, and prepares a list of RINEX files for parallel processing.

PID 022 OBSMRG_P: The processing of the RINEX observation files is done by the program CCRINEXO. The observations that belong to the current session or to the range of hourly sessions (defined by the BPE variable V_HOURLY) are copied from the input RINEX files in the ORX directory of the campaign into the output files located in the campaign's RAW directory.

PID 031 ION_MRG: In case of an hourly processing scheme the ionosphere maps of several days are simply concatenated by the initial copy script PPP_COP. If the cumulated ionosphere maps contain duplicated epochs, they are cleaned by this script.

PID 099 DUMMY: Does nothing, closes this section of scripts.

If scripts/programs can run in parallel, in fact two scripts are needed in the PCF (for clarity, both should agree with the first six characters of their names, e.g., OBSMRGAP and OBSMRG_P in PIDs 021 and 022): the first of the pair defines how many processes will run in parallel and which input file has to be considered in which of the parallel running processes. Typically, this script also cleans up all potential result files from this step to prevent potential interferences with earlier runs of the BPE. This preparing script is usually indicated by AP at the end of the script name. The second script performs the processing steps itself where the input files for each of the parallel running scripts are given by the preparing script. Typically, the filename of this second script ends with _P.

The last script of this section (PID 099 DUMMY) does in fact nothing. It seems useless at first glance — but it simplifies the bookkeeping in that it provides a well-defined PID which marks the end of a section in the PCF. The waiting conditions for this script should be defined such that is it executed as the last script of the section. This way, one only needs to check the completion of these closing scripts to verify the completion of the numerous scripts in between.

Further reading:

Section 3.1: "Overview of the Directory Structure", page 57
Section 18.3: "ANTEX Converter ATX2PCV", page 411
Section 22.5.2: "Parallel Processing", page 490
Section 22.5.3: "Loops and Jumps", page 491

Prepare Pole, Orbit, and Clock Information

This block converts the pole information to Bernese format, generates the orbit information in Bernese standard orbit format, and extracts the satellite clock information from the clock RINEX file. You should always use consistent orbit, satellite clock, and pole files to obtain reasonable results for the PPP.

```
#
# Prepare pole, orbit, and clock information
# -----------------------------------------
101 POLUPDH  PPP_GEN          ANY      1 001
111 ORBMRGH  PPP_GEN          ANY      1 001
112 PRETAB   PPP_GEN          ANY      1 101 111
113 ORBGENH  PPP_GEN          ANY      1 112
121 CCRNXCH  PPP_GEN          ANY      1 001
122 RNXCLK   PPP_GEN          ANY      1 121
199 DUMMY    NO_OPT           ANY      1 113 122
```

PID 101 POLUPDH: Program POLUPD extracts ERP information from an IERS formatted pole file (extension `IEP`) into a Bernese formatted pole file (extension `ERP`).

The script checks the content of the input pole file and tries to find out the correct setting for option "Use ERP rates". Because of the diversity of the IERS/IGS formatted pole files, it might happen that the algorithm fails. In this case you have to define the option manually in the user script by replacing XYZ by the source of your products (value of BPE server variable `V_B`):

```
# Repeat the program if necessary
# ------------------------------
  if ( $v_b eq "XYZ" ) { $doRate = 1 }  # if products from "XYZ" are used
                                        # apply the polar rates.
  if ( $doRate != $isRate ) {
    $bpe->putKey("$ENV{U}/PAN/POLUPD.INP","USERATE","$doRate");

    my $PGMNAM = "POLUPD";
    $bpe->RUN_PGMS($PGMNAM);
  }
```

The letter `H` at the end of the script name indicates that the list of input files is adapted if an hourly processing scheme is used.

PID 111 ORBMRGH: Concatenates precise orbit files from several days if needed for an hourly processing scheme.

If products from the IGS shall be used (indicated by the BPE variable V_B == "IGS") and GLONASS shall be included (BPE variable V_SATSYS == "GPS/GLO") in the processing, the GPS and GLONASS orbit files are merged (`IGSwwwwd` and `IGLwwwwd`). Note that this is not needed if you use products from CODE because these precise orbit files contain the satellite positions for both systems.

PID 112 PRETAB: The program PRETAB converts the orbit information available in the precise orbits in SP3c format (Earth-fixed frame, extension `PRE`) into tabular positions in the inertial frame (extension TAB) for subsequent numerical integration by ORBGEN.

PID 113 ORBGENH: The program ORBGEN integrates the equations of motion using the positions given in the tabular orbit file to produce a Bernese standard orbit file (extension STD), used in all processing programs needing orbit information. The orbit is represented by six osculating elements and nine dynamical parameters associated with radiation pressure.

The letter H at the end of the script name indicates again a special action for an hourly processing scheme: the setup of stochastic pulses is switched off if the length of the processing interval (given by the BPE–server variable V_HOURLY) is too short compared to the sampling of the pulses as requested in the program input file.

PID 121 CCRNXCH: If an hourly processing scheme is set up, this script concatenates clock RINEX files from several days.

PID 122 RNXCLK: This program converts a clock RINEX file (prepared by PID 121) into a Bernese satellite clock file (extension CLK). The resulting file has the same name as the RINEX file but resides in the campaign's ORB directory.

PID 199 DUMMY: Does nothing, closes this section.

The orbit information prepared in these steps is used throughout the further processing, and is usually never changed.

Note that there are a few scripts with the letter H at the end of their names. This shall indicate that they contain a special section for hourly processing, e.g., to define the range of selected sessions according to the BPE server variable V_HOURLY .

Further reading:

Section 4.3: “Precise Orbit Files”, page 90
Section 4.4: “IGS and IERS Pole Files”, page 91
Chapter 5: “Preparation of Earth Orientation, GNSS Orbit, and Satellite Clock Information”, page 107
Section 15.3.1: “Extracting Satellite Clocks from Clock RINEX Files”, page 365

Preprocess, Convert, and Synchronize Observation Data

The purpose of this PCF block is to prepare the observation data (cleaning the observation data and conversion into Bernese format), to update the station information and coordinate files, and to synchronize the receiver clocks to GPS system time.

Depending on the availability of high–rate satellite clocks, the extended version of the PPP example (PPP_DEMO.PCF) offers two possibilities for preprocessing the phase observation files:

- using the program MAUPRP if high–rate satellite clock corrections are available (indicated by BPE server variable V_HRCLK == 'Y') or
- using the program RNXSMT; here, only the consistency between code and phase data of the two frequencies is checked.

The basic version of the PPP example (PPP_BAS.PCF) only offers the latter way of preprocessing.

```
#
# Preprocess, convert, and synchronize observation data
# -------------------------------------------------------
201 PPP_PRP  PPP_GEN           ANY      1 099
```

PID 201 PPP_PRP: Evaluates the value of the BPE server variable `V_HRCLK` and continues the processing at the corresponding PID:

211 for preprocessing code and phase data based on RNXSMT or
251 for using the program MAUPRP to clean the phase observations.

```
#
# Phase preprocessing with RNXSMT
211 RNXSMTAP PPP_GEN           ANY      1 201
212 RNXSMT_H PPP_GEN           ANY      1 211
221 RNXGRA   PPP_GEN           ANY      1 212
231 RXOBV3AP PPP_GEN           ANY      1 221
232 RXOBV3_H PPP_GEN           ANY      1 231
233 CRDMRDAT PPP_GEN           ANY      1 232
241 CODSPPAP PPP_GEN           ANY      1 199 233
242 CODSPP_P PPP_GEN           ANY      1 241
243 CODXTR   PPP_GEN           ANY      1 242
249 JUMP     NO_OPT            ANY      1 243
```

PID 211 RNXSMTAP: The purpose of this step is to clean data on the RINEX level. It makes sense to parallelize this step, which is prepared by this script. The actual processing is done in the next PID . The script deletes files that may be present from previous runs, and prepares a list of RINEX files for parallel processing.

PID 212 RNXSMT_H: Program RNXSMT is called to clean the RINEX data and to identify cycle slips. Processing is done in parallel, i.e., the program receives a list of RINEX files to be cleaned in one run. The output consists of smoothed RINEX files (extension `SMT`) containing smoothed instead of original code observations. Cycle slips and outliers are marked. In case of RINEX (version 3) files, it will select the observation types according to the priority list given in `V_OBSINF`.

The letter `H` at the end of the script indicates that the time window is adapted to the `V_HOURLY` variable for an hourly processing scheme.

PID 221 RNXGRA: Program RNXGRA produces a summary of the smoothed RINEX observation data, giving a complete overview of observed satellites, involved stations and their performance. This file appears in the processing summary and may help to identify possible data tracking problems of observing sites.

PID 231 RXOBV3AP: The script prepares several runs of the program RXOBV3 to import the observations from RINEX into the binary Bernese format. In general, this can be done in parallel for the individual stations. The only exception is if — as is the case for this BPE — the entries in the "Abbreviation table" are automatically updated, i.e., if the option "Action if station not in abbreviation list" in panel "RXOBV3 2: Input Options 1" is set to UPDATE. In that case, all stations are processed in one group to prevent interferences with this update step due to multiple file access at the same time.

PID 232 RXOBV3_H: This script calls the program RXOBV3, which creates Bernese observation files for code and phase from RINEX files according to the number of sessions indicated by `V_HOURLY` in case of an hourly processing scheme.

In addition, the station information file (indicated by the BPE variable `V_STAINF`) may be used to check the RINEX header content. The options set for this PCF are restrictive, the BPE will stop in case of inconsistencies in the station name. For inconsistencies regarding the receiver/antenna type, a message is added to the protocol file. With the default restrictive settings, the BPE will only stop in one of the subsequent steps if the receiver or antenna is unknown to the related correction tables ("Receiver information" and "Phase center eccentricities").

If a new station is encountered, the coordinate file is updated using the RINEX header information. Also, the station name abbreviation table is updated for stations that are not yet listed. The processing summary informs about these actions.

The results are zero–difference code and phase observation files for each station (extensions CZH, CZO, PZH, and PZO files in the campaign's OBS directory).

PID 233 CRDMRDAT: Compares the value in V_REFINF with all known indicators for a geodetic datum in the file ${X}/GEN/DATUM. and adjusts the option "New local geodetic datum" for the program CRDMERGE. Program CRDMERGE is executed to set the geodetic datum and coordinate reference epoch to a common value for all stations involved.

PID 241 CODSPPAP: This script prepares the parallel run of CODSPP. Due to its waiting conditions, it is not executed before all input files are available.

PID 242 CODSPP_P: This script runs CODSPP. The receiver clock synchronization is the main task of CODSPP. The program also determines approximate station coordinates by a straightforward point positioning approach, using only smoothed code observations. This is useful in case the a priori coordinates (e.g., from the RINEX header) are of doubtful quality. Outliers in the observations can be detected in this step, too.

PID 243 CODXTR: This script runs the output extraction program for CODSPP named CODXTR. It produces a summary file informing on detected outliers, missing orbits, excluded satellites due to satellite problems, and missing clocks. The extract is part of the protocol file.

PID 249 JUMP: The code and phase data are cleaned and ready for the next processing step. The section for preprocessing the observation files using MAUPRP does not need to be executed. Therefore, the processing continues to the end of this sequence. The task of this script is to jump to PID 299 of the PCF.

There are two examples for parallel scripts that contain a special section for the hourly processing. To indicate this, they are named, e.g., RNXSMT_H instead of RNXSMT_P. See the comments on the script naming in the previous two sections of the example.

```
#
# Phase preprocessing with MAUPRP
251 RNXSMTAP PPP_GE2           ANY      1 249
252 RNXSMT_H PPP_GE2           ANY      1 251
253 RNXGRA   PPP_GEN           ANY      1 252
261 RXOBV3AP PPP_GE2           ANY      1 253
262 RXOBV3_H PPP_GE2           ANY      1 261
263 CRDMRDAT PPP_GEN           ANY      1 262
271 CODSPPAP PPP_GE2           ANY      1 199 263
272 CODSPP_P PPP_GE2           ANY      1 271
273 CODXTR   PPP_GEN           ANY      1 272
281 MAUPRPAP PPP_GE2           ANY      1 199 273
282 MAUPRP_P PPP_GE2           ANY      1 281
283 MPRXTR   PPP_GE2           ANY      1 282
299 DUMMY    NO_OPT            ANY      1 201 249 283
```

PID 251 RNXSMTAP: As in PID 211, the RINEX files are divided into groups that may be processed together in parallel, which is actually done in the next script.

PID 252 RNXSMT_H: The procedure of this script is nearly the same as for PID 212 : executing the program RNXSMT. The only difference is that the original phase observations are copied into the resulting smoothed RINEX files (extension `SMT`).

PID 253 RNXGRA: As in PID 213, program RNXGRA produces a summary of the smoothed RINEX observation data, giving a complete overview of observed satellites, involved stations and their performance. This file appears in the processing summary and may help to identify possible data tracking problems of observing sites.

PID 261 RXOBV3AP: The import of the smoothed RINEX files (extension `SMT`) is prepared, where several groups of files are processed in parallel (or not, if the option "Action if station not in abbreviation list" is set to UPDATE). It is thus the same as explained in detail above for PID 231 .

PID 262 RXOBV3_H: The same functions as described in PID 232 are executed resulting in binary zero–difference code and phase observation files for each station (extensions `CZH`, `CZO`, `PZH`, and `PZO` files in the campaign's `OBS` directory).

The only difference is the sampling of the observation files: if phase observations are preprocessed on RINEX level using RNXSMT only (as done in PID 232), the data sampling of the PPP processing (as defined by the BPE server variable `V_SAMPL`) may be applied to the data import. Now the preprocessing of the phase measurements shall be performed later in the program MAUPRP and therefore the original sampling of, e.g., 30 seconds must be retained in the observation files. Consequently, the only difference between the input files for RXOBV3 that are used in the PIDs 232 and 262 is the value of the option "Sampling interval" .

PID 263 CRDMRDAT: The coordinate file is updated as described in PID 233 .

PID 271 CODSPPAP: This script prepares the parallel run of CODSPP, as in PID 241 . This script waits for completion of altogether three PIDs to ensure that all input files are available.

PID 272 CODSPP_P: As in PID 242 the program CODSPP is executed to synchronize the receiver clocks with respect to GPS system time. Since the phase measurements are not yet cleaned, outliers detected by the program CODSPP are only applied to CODE instead of BOTH observation types in option "Mark outliers in obs. files" (difference in the processing options for CODSPP between PIDs 242 and 272).

PID 273 CODXTR: A summary for the protocol file is extracted from the program output of CODSPP using the program CODXTR (same procedure as in PID 243).

PID 281 MAUPRPAP: The preprocessing of phase measurements is done by MAUPRP in this section. It may be run in parallel, which is initialized and prepared by this script.

PID 282 MAUPRP_P: This script actually runs the program MAUPRP for the selected stations to detect and correct cycle slips, to identify outliers, and to update the list of ambiguities for the phase measurements. The screening results are stored in the phase zero–difference observation files.

The screening of zero–difference observation files with MAUPRP requires well–established satellite clock corrections with the same sampling rate as in the data files (at least 30 seconds) — even if the PPP will be carried out with a lower sampling (e.g., 5 minutes). The higher sampling for screening the phase observations is required to improve the capability to distinguish between potential cycle slips and the change of the ionosphere from one epoch to the next.

PID 283 MPRXTR: This script runs the program MPRXTR to extract the essential information from the MAUPRP program output files into a summary table to be included in the protocol file later on.

PID 299 DUMMY: Does nothing, closes this section.

Further reading:

Section 4.2: “RINEX Observation Files”, page 75
Section 6.2: “Preprocessing on the RINEX Level”, page 129
Section 6.2.5: “Code Smoothing”, page 132
Section 4.2.3: “Import to Bernese”, page 79
Section 6.3: “Receiver Clock Synchronization and Preprocessing of Code Observations”, page 135
Section 6.5: “Preprocessing Phase Observations”, page 144
Section 22.5.2: “Parallel Processing”, page 490
Section 22.5.3: “Loops and Jumps”, page 491

Compute PPP Solutions Station by Station (including Data Screening)

This section of the PCF screens data and performs a PPP using code and phase measurements in a common analysis.

```
#
# Compute PPP solutions station by station (including data screening)
# -------------------------------------------------------------------
301 PPPEDTAP PPP_GEN           ANY      1 299
302 PPPEDT_P PPP_GEN           ANY      1 301
303 GPSXTR   PPP_GEN           ANY      1 302
311 RES_SUM  PPP_AUX           ANY      1 302
321 CRDMERGE PPP_AUX           ANY      1 302
322 ADDNEQ2  PPP_AUX           ANY      1 321
323 PPP_HLM  PPP_AUX           ANY      1 321
331 CCRNXC   PPP_AUX           ANY      1 321
341 ADDNEQ2  PPP_SNX           ANY      1 322
399 DUMMY    NO_OPT            ANY      1 303  311  322  323  331  341
```

PID 301 PPPEDTAP: This script prepares the parallel run of GPSEST.

PID 302 PPPEDT_P: The following processing programs are called to perform a station by station data cleaning process and to compute the PPP solution.

1. GPSEST: generate a residual file for data screening based on the ionosphere–free linear combination (L_3). Normalized residuals are written, as elevation–dependent weighting of observations is applied.
2. RESRMS: screen the residual file for outliers exceeding a certain threshold.
3. SATMRK: mark identified outliers in the observation files. The actual observation data remains in the files, the corresponding records are flagged as bad.
4. GPSEST: basically the same run as above, this time based on the cleaned observations. The results are stored in stationwise result files for further use:
    - station coordinates (extension `CRD`, in the `STA` directory of the campaign),
    - receiver clock corrections (clock RINEX format with extension `CLK`, in the `OUT` directory of the campaign), and

- station–specific troposphere parameters (extension TRP and troposphere SINEX with extension TRO, in the ATM directory of the campaign),
- normal equation file (extension NQ0, in the SOL directory of the campaign).

5. ADDNEQ2: generate PPP result files for each station in Bernese and external formats, namely:
   - station coordinates and SINEX (with extension SNX, in the SOL directory of the campaign).

Note that steps (1) to (3) run iteratively with different (decreasing) limits for outlier detection in program RESRMS. Only results from the first run of GPSEST are retained (EDFssssxxxx with ssss as session identifier and xxxx as the 4–character station ID). Files from the GPSEST run of step (4) are named EDLssssxxxx. The results of step (5) are stored in files named {V_C}ssssxxxx, where V_C is the value of the corresponding BPE variable, e.g., PPP.

Stations for which no ocean or atmospheric tidal loading entries were found in the respective file, entail warnings in the GPSEST error message file. These are extracted and reported in the processing summary.

In addition, the processing summary comprises information on receiver tracking technology as extracted from the GPSEST output files.

PID 303 GPSXTR: Output extractor for program GPSEST (step (4) in PPPEDP_P), producing an overview of the PPP solution and the data cleaning.

PID 311 RES_SUM: This script calls several processing programs sequentially:

1. RESRMS produces a residual statistics based on the files generated in step PID 302 before screening (solution ID EDF). L_3 is analyzed.
2. RESRMS produces a residual statistics based on the screened files from PID 302 (last GPSEST run in step (4); solution ID EDL). L_3 is analyzed.
3. RESCHK creates a residual screening statistics.

The summaries created by RESRMS and RESCHK are included in the processing summary.

PID 321 CRDMERGE: The station–specific coordinate files from PID 302 are merged into one coordinate file.

PID 322 ADDNEQ2: The normal equations from the final GPSEST in PID 302 are collected and the troposphere parameters from all stations are provided in one Bernese formatted file and in one troposphere SINEX file.

PID 323 PPP_HLM: Executes the program HELMR1 to compare the PPP results for the station coordinates with the content of the project–specific coordinate file propagated to the epoch of the current session. This comparison will also be included in the processing summary.

PID 331 CCRNXC: The station–specific clock RINEX files are combined to obtain one RINEX clock file.

PID 341 ADDNEQ2: ADDNEQ2 is used to write a single SINEX file and a combined normal equation file containing all stations based on the normal equation files generated in PID 302. The troposphere parameters are pre–eliminated in this step. The SINEX sections related to station information and the estimated coordinates are listed in the processing summary.

PID 399 DUMMY: Does nothing but closes this section of the PCF .

At this point, the PPP processing is basically finished. What follows now is the optional propagation of coordinates to the reference epoch using the NNR–NUVEL velocity model.

Further reading:

Chapter 7: “Parameter Estimation”, page 169
Section 7.4.4: “Real and Normalized Residuals”, page 174
Section 6.6: “Screening of Post–Fit Residuals”, page 161
Section 6.7: “Marking of Observations”, page 167
Chapter 10: “Station Coordinates and Velocities”, page 241
Section 10.6: “Precise Point Positioning”, page 265
Section 7.9.2: “The Output Extraction Program GPSXTR”, page 189
Section 10.7.2: “Helmert Transformation”, page 267
Chapter 12: “Troposphere Modeling and Estimation”, page 295
Section 14.3.4: “Verification of the Receiver Tracking Technology”, page 346
Section 15.3: “Clock RINEX Utilities in *Bernese GNSS Software*”, page 365
Chapter 9: “Combination of Solutions”, page 209
Section 9.4.9.1: “Writing SINEX Files”, page 233
Section 22.5.2: “Parallel Processing”, page 490
Section 22.5.3: “Loops and Jumps”, page 491

Take into Account NNR–NUVEL–1A Velocities (if Ddesired)

This block allows to take station velocities according to the NNR–NUVEL–1A model (DeMets et al. 1994) into account, if desired. The scripts are executed only if the BPE variable V_UPD is set to Y or if the geocenter estimation is enabled (variable V_G is not empty).

```
#
# Take into account NNR-NUVEL1-A velocities (if desired)
# -----------------------------------------------------
401 PPP_UPD  PPP_VEL           ANY      1 399
402 NUVELO   PPP_VEL           ANY      1 401
403 COOVEL   PPP_VEL           ANY      1 402
411 CRDMERGE PPP_VEL           ANY      1 403
421 CRDMERGE PPP_CRD           ANY      1 403
499 DUMMY    NO_OPT            ANY      1 411 421
```

PID 401 PPP_UPD: This script checks the values of the BPE server variables V_UPD and V_G. If V_UPD is different from Y and V_G is empty (it means no geocenter estimation), the remaining PIDs in this block are skipped and execution resumes at PID 499.

PID 402 NUVELO: The velocity field for the stations is computed according to the NNR–NUVEL–1A model and stored in a Bernese velocity file (extension VEL, in the STA directory of the campaign).

PID 403 COOVEL: The station coordinates in the file {V_C}yyssss.CRD are propagated from the session's epoch to the coordinate epoch of the reference frame file defined in the BPE variable V_REFEPO (January 1, 2005 for IGB08.CRD) using the velocities from the previous step. The resulting coordinate file is called {V_E}yyssss.CRD.

PID 411 CRDMERGE: The velocity field generated in PID 402 is merged into the velocity reference file defined in V_REFINF (e.g., IGB08_R.VEL). The merged velocity field is stored in the file defined in V_CRDINF (e.g., EXAMPLE.VEL). It contains all reference station velocities, completed by the additional sites in this PPP solution.

PID 421 CRDMERGE: The coordinate file generated in PID 403 is merged into the coordinate reference file. Its name is defined by V_CRDINF and it contains all reference station coordinates, completed by the additional sites in this PPP solution.

PID 499 DUMMY: Does nothing, closes this section.

Further reading:

Chapter 10: "Station Coordinates and Velocities", page 241
Section 10.7.8: "Computing Velocities from a Model", page 271
Section 10.7.9: "Propagating Coordinates to Specific Epochs", page 272
Section 10.7.7: "Merging Coordinate and Velocity Files", page 271
Section 22.5.3: "Loops and Jumps", page 491

Geocenter Estimation, Extended Version only

This group of scripts is responsible for the geocenter estimation in the extended version of the PPP example (PPP_DEMO.PCF). It is executed only if the BPE server variable V_G has a non–blank value.

For non–global networks this result is a quality check for the PPP and has no geophysical meaning. The sequence of scripts may be implemented in an analogous way in other BPEs, e.g., the double–difference example RNX2SNX.PCF, described in Section 23.4.2 .

```
#
# Geocenter estimation (V_G)
501 PPP_GCC  PPP_GCC           ANY      1 499
502 ADDNEQ2  PPP_GCC           ANY      1 501
503 GPSXTR   PPP_GCC           ANY      1 502
519 DUMMY    NO_OPT            ANY      1 503
```

PID 501 PPP_GCC: This script either initializes or skips the geocenter estimation. If the variable V_G is undefined, the scripts responsible for geocenter estimation are skipped and PID 519 follows.

PID 502 ADDNEQ2: The merged coordinate and velocity files (PIDs 411 and 421) together with the normal equation file (generated in PID 322) are used to estimate geocenter translation parameters. A minimum constraint solution with translation conditions only is performed.
The stations used for this datum definition are taken from the list of reference stations (extension FIX, filename according to BPE variable V_REFINF. . The geocenter coordinates are written into the file {V_G}yyssss.GCC .

PID 503 GPSXTR: An output summary of the previous ADDNEQ run is produced to be included in the processing summary.

PID 519 DUMMY: Does nothing, closes the section on geocenter estimation.

Further reading:

Section 16.4: "Estimation of Earth Orientation and Geocenter Parameters", page 385
Section 22.5.3: "Loops and Jumps", page 491

Pseudo–Kinematic Positioning, Extended Version only

The following sequence of scripts is related to the pseudo–kinematic positioning which is available in the extended version of the PPP example (`PPP_DEMO.PCF`) only. The group of scripts is only executed if the BPE server variable `V_H` is non–blank.

The pseudo–kinematic positioning is well suited for quality control of the individual stations in the network. The sequence may also be included in an analogous way in other BPEs, e.g., in the double–difference example `RNX2SNX.PCF` (description in Section 23.4.2).

```
#
# Pseudo-kinematic solution (V_H)
521 PPP_KIN  PPP_KIN           ANY      1 499
522 TIMESTAP PPP_KIN           ANY      1 521
523 TIMEST_P PPP_KIN           ANY      1 522
524 GPSXTR   PPP_KIN           ANY      1 523
525 TIMESTAP PPP_KIN           ANY      1 523
526 GPSXTR_P PPP_KIM           ANY      1 525
539 DUMMY    NO_OPT            ANY      1 524 526
```

PID 521 PPP_KIN: This script either initializes or skips the pseudo–kinematic positioning. If the variable `V_H` is undefined, the scripts responsible for pseudo–kinematic positioning are skipped and PID 539 follows.

PID 522 TIMESTAP: This is the preparatory script for the stationwise parallel run of GPSEST.

PID 523 TIMEST_P: The script runs the program GPSEST for a single station in a kinematic mode. An output file with the epochwise kinematic positions (extension `KIN` in the `STA` directory of your campaign) is generated.

PID 524 GPSXTR: The program GPSXTR compiles a summary of the output files of the GPSEST generating the pseudo–kinematic solution in the previous PID.

PID 525 TIMESTAP: The summary for the kinematic positioning can only be generated from one GPSEST program output per GPSXTR. The program GPSXTR thus needs to be run in parallel (station by station), which is prepared by this script.

PID 526 GPSXTR_P: The program GPSXTR is called by the script to create a statistics on the pseudo–kinematic positioning results of the GPSEST in PID 523 . For each station a summary file (extension `SUM` in the `OUT` directory of the campaign) is produced where a certain extract is included in the protocol file.

PID 539 DUMMY: Does nothing but closing the sequence of scripts related to the pseudo–kinematic positioning.

Further reading:

Section 10.5: "Estimating Kinematic Coordinates", page 260
Section 7.9.2: "The Output Extraction Program GPSXTR", page 189
Section 22.5.2: "Parallel Processing", page 490
Section 22.5.3: "Loops and Jumps", page 491

High–Rate Troposphere Estimation, Extended Version only

This group of scripts is related to the high–rate troposphere estimation that is only included in the extended version of the PPP example (`PPP_DEMO.PCF`). It is only executed if the BPE server variable `V_I` is defined.

The sequence of scripts can also be analogously transferred into the double–difference example RNX2SNX.PCF (see Section 23.4.2 for the detailed description).

```
#
# High-rate troposphere (V_I)
541 PPP_TRP  PPP_TRP           ANY      1 499
542 TIMESTAP PPP_TRP           ANY      1 541
543 TIMEST_P PPP_TRP           ANY      1 542
544 GPSXTR   PPP_TRP           ANY      1 543
545 ADDNEQ2  PPP_TRQ           ANY      1 543
546 GPSXTR   PPP_TRQ           ANY      1 545
559 DUMMY    NO_OPT            ANY      1 544 546
```

PID 541 PPP_TRP: This script either initializes or skips the high–rate troposphere estimation. If the variable V_I is undefined, the scripts responsible for high–rate troposphere estimation are skipped and PID 559 follows.

PID 542 TIMESTAP: This is the preparatory script for the stationwise parallel run of GPSEST.

PID 543 TIMEST_P: The script performs a GPSEST run to compute the high–rate troposphere solution for each station. The results are provided in Bernese formatted troposphere result files (extension TRP) and troposphere SINEX files (extension TRO). Both are located in the ATM directory of the campaign. In addition, normal equation files (extension NQ0 in the campaign's SOL directory) are written.

PID 544 GPSXTR: The script calls the program GPSXTR to extract a summary characterizing the stationwise high–rate troposphere estimation for the protocol file.

PID 545 ADDNEQ2: Collecting the normal equation files from all stations (as produced in PID 543) into one common solution in order to provide the estimated troposphere results for all stations of the current session in one Bernese formatted troposphere file (extension TRP) and one troposphere SINEX file (extension TRO), both in the ATM directory.

PID 546 GPSXTR: GPSXTR creates a summary of the ADDNEQ2 output from the previous PID to be included in the processing summary.

PID 559 DUMMY: Does nothing, closes the sequence related to the high–rate troposphere estimation.

Further reading:

Chapter 12: "Troposphere Modeling and Estimation", page 295
Section 22.5.2: "Parallel Processing", page 490
Section 22.5.3: "Loops and Jumps", page 491

Ionosphere Model and DCB Estimation, Extended Version only

The following sequence of scripts is related to the ionosphere model and DCB estimation (P1−P2). It is only available in the extended version of the PPP example (PPP_DEMO.PCF). The sequence of scripts is executed if the BPE server variable V_K is non–blank and if a daily processing scheme is applied.

```
#
# Station-specific ionosphere/DCB estimation (V_K)
561 PPP_ION  PPP_ION           ANY      1 499
562 TIMESTAP PPP_ION           ANY      1 561
563 TIMEST_P PPP_ION           ANY      1 562
564 GPSXTR   PPP_ION           ANY      1 563
565 ADDNEQ2  PPP_ION           ANY      1 563
#
# Regional ionosphere model/DCB estimation (V_L)
571 GPSEST   PPP_RIM           ANY      1 561
572 GPSXTR   PPP_RIM           ANY      1 571
579 DUMMY    NO_OPT            ANY      1 564 565 572
#
599 DUMMY    NO_OPT            ANY      1 519 539 559 579
```

PID 561 PPP_ION: This script either initializes or skips the ionosphere estimation. If the variable V_K is undefined or an hourly session scheme is applied, the scripts responsible for ionosphere estimation are skipped and PID 579 follows.

PID 562 TIMESTAP: This is the preparatory script for the stationwise parallel run of GPSEST.

PID 563 TIMEST_P: In this GPSEST run, station–specific ionosphere models are estimated and saved in Bernese ionosphere files (extension ION, in the ATM directory of the campaign). Receiver P1−P2 DCBs are also computed and stored in station–specific files (extension DCB). In addition, the normal equation files are saved for a combination of the receiver–specific DCBs (note that all other parameters are pre–eliminated prior to NEQ saving in this GPSEST run).

PID 564 GPSXTR: GPSXTR creates a summary of the GPSEST output of the previous PID. A summary concerning global ionosphere maps is explicitly requested. The extracts are listed in the processing summary.

PID 565 ADDNEQ2: ADDNEQ2 combines the contributions from the station–specific processing in PID 563 to the P1−P2 receiver DCBs. The DCB file additionally appears in the processing summary.

PID 571 GPSEST: In this GPSEST run, a regional ionosphere model is generated and stored in a Bernese ionosphere file and in an IONEX file (extension INX). Again, DCBs are stored, and the file is included in the processing summary.

PID 572 GPSXTR: An output summary of the previous GPSEST run is produced which is included in the processing summary.

PID 579 DUMMY: Does nothing, closes the section on ionosphere model estimation.

PID 599 DUMMY: Ends the section of optional parameter generation.

Further reading:

Chapter 13: “Ionosphere Modeling and Estimation”, page 311
Section 13.3.1.1: “Ionosphere Mapping”, page 318
Chapter 14: “GNSS Related Biases”, page 339
Section 22.5.2: “Parallel Processing”, page 490
Section 22.5.3: “Loops and Jumps”, page 491

Create Summary File, Save and Delete Files, End of BPE

This block wraps up the PPP_BAS.PCF and PPP_DEMO.PCF examples. No processing program is called. It will — in one form or another — also be part of your own PCFs.

```
#
# Create summary file and delete files
# ------------------------------------
901 PPP_SUM  PPP_GEN           ANY      1 599
902 PPP_SAV  PPP_GEN           ANY      1 901
903 OBS_SAV  PPP_GEN           ANY      1 901
904 PPP_DEL  PPP_GEN           ANY      1 902 903
991 BPE_CLN  PPP_GEN           ANY      1 904
#
# End of BPE
# ----------
999 DUMMY    NO_OPT            ANY      1 991
```

PID 901 PPP_SUM: This step produces a comprehensive summary of the previous processing steps allowing to check the quality of the obtained results. The resulting protocol file is called `PPPyyssss.PRC` and will be discussed in Section 23.4.1.7 .

PID 902 PPP_SAV: The result files from this BPE (listed in Section 23.4.1.4) are copied from the campaign into the `SAVEDISK` area.

PID 903 OBS_SAV: The observation files from this BPE are copied from the campaign into the `SAVEDISK` area, see description of BPE server variable `V_SAVOBS`.

PID 904 PPP_DEL: This script is responsible for deleting files generated by this PCF during this and previous sessions. This script is set up to prevent the accumulation of files in the campaign.

By default, the script removes all files related to 10 sessions prior to the current session. You may change this delay by adjusting the parameter PARAM1 of this script (also read the header of this script for more details).

PID 991 BPE_CLN: This step is responsible for cleaning up BPE–specific files (`LOG` and `PRT` files in the campaign's `BPE` directory) with a delay of 30 sessions.

PID 999 DUMMY: Does nothing. This script provides a well–defined PID as end point, useful, e.g., as jump address, or to check for the completion of the BPE.

If you want to archive other, or more files, you should not remove them from the list in the deletion script (`PPP_DEL`) but add them to the list of files to be copied to the `SAVEDISK` area (`PPP_SAV`).

The `DUMMY`–script with PID 999 may serve as a well–defined PID at the end in all your PCFs. The successful execution of a BPE can be tested in a generic way by checking the existence of this script's `LOG`– or `PRT`–file. In this way a script starting a BPE can react in case of errors (e.g., send an error mail).

Further reading:

Section 22.10: "BPE Output and Protocol Files", page 518
Section 22.11: "Error Handling", page 522

23.4.1.7. Description of the Protocol File, Quality Control

The processing is summarized in a so–called protocol file. It is compiled in the BPE user script `PPP_SUM` and exists first in the `OUT` directory of the campaign. If the BPE server variable V_SAV == 'Y', the protocol is copied by the `PPP_SAV` script to `${S}/{V_RESULT}/yyyy/OUT/PPPyyssss.PRC` and may be removed from the campaign area by the script `PPP_DEL` (if V_DEL == 'Y').

This protocol file starts with a section reporting all important input information for the BPE processing defined by the BPE server variables:

```
================================================================================
PPP BPE PROCESSING SUMMARY FOR YEAR-SESSION 10-2070
================================================================================

Summary file generated at 13-Dec-2012 07:27:03 by PPP_SUM

General files:
   Antenna phase center eccentricity file:   PCV.I08
   Satellite information file:               SATELLIT.I08
   Satellite problem file:                   SAT_2010.CRX
   Receiver characterization file:           RECEIVER.
   Orbit, ERP and clock products used from:  ${D}/COD

...
```

The following sections are related to the different tasks/features of this BPE:

PART 0: BLQ/ATL COMPLETENESS CHECK

If the files ${D}/{V_REFDIR}/{V_CRDINF}.BLQ and/or ${D}/{V_REFDIR}/{V_CRDINF}.ATL are available, the ocean and/or atmospheric tidal corrections are considered for the processing. These files contain tables of coefficients for individual stations. Stations used in this BPE, which are not listed in these tables, are listed here.

```
...

================================================================================
PART 0: EXAMPLE.BLQ FILE CONTAINS RECORDS FOR ALL STATIONS
================================================================================



================================================================================
PART 0: EXAMPLE.ATL FILE CONTAINS RECORDS FOR ALL STATIONS
================================================================================
...
```

Further reading:

Section 3.6.3: “Other Station Files”, page 68
Section 24.7.9: “Ocean Tidal Loading Table”, page 737
Section 24.7.10: “Atmospheric Tidal Loading File”, page 739

PART 1: RINEX PSEUDO-GRAPHICS

The content of the RINEX observation files is summarized here (summary file output from program RNXGRA). Here you may detect stations with problematic (e.g., reduced tracking performance of a station).

Further reading:

Section 4.2.5: “Utilities”, page 87

PART 2: ORBIT GENERATION SUMMARY

The summary file of ORBGEN reports the RMS of the fit between the satellite position given in the precise orbit file ${D}/{V_B}/{V_B}wwwwd.PRE with the orbit model of ORBGEN. It is expected that the RMS is on the few millimeters level (pseudo–stochastic pulses are set up every twelve hours to compensate for different orbit models in ORBGEN and the external program which was used to produce the orbits).

```
===============================================================================
PART 2: ORBIT GENERATION SUMMARY
===============================================================================

 PPP_102080: Generate GNSS standard orbit                         02-MAY-14 14:06
 TIME FROM DAY : 2 GPS WEEK: 1594
      TO   DAY : 3 GPS WEEK: 1594
 ------------------------------------------------------------------------------
 ORBIT REPEATABILITY FROM A 1-DAY FIT THROUGH DAILY ORBIT SOLUTIONS (MM)
 # ECLIPSING SATELLITES:   5 E /   0 M (  0 EM)
 ---------------------------------------------------------------------------------------- ...
 ECL  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  E.  E.  E.  ..  ..  ..  ..  ..  ..  ...
 DOY   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  ...
 ---------------------------------------------------------------------------------------- ...
 208   2   1   3   1   2   3   2   2   2   2   2   3   3   3   2   3   2   3   2   2   2  ...
 ---------------------------------------------------------------------------------------- ...
 ALL   2   1   3   1   2   3   2   2   2   2   2   3   3   3   2   3   2   3   2   2   2  ...
 ---------------------------------------------------------------------------------------- ...
```

If the RMS is larger, you should check the consistency of the Earth rotation files ${D}/{V_B}/{V_B}wwww7.ERP (or ${D}/{V_B}/{V_B}wwwwd.ERP). It is also possible to split the orbital arc into two or three arcs ("Number of arcs within the time window" in the panel "ORBGEN 5: Orbital Arc Definition" of the program input file). It is essential that the RMS of the orbit fit can be reduced to the few millimeters level since the satellite clock corrections for the PPP refer to the original positions of the satellite in the given precise orbit file.

Since CODE has improved its solar radiation pressure model in its operational solution from January 2015 onwards, an update of the software (published as B_049) is needed to allow the fit of the precise orbit files with the expected quality. After updating the software you may choose the new model by selecting "dynamical orbit parameters" as "System D2X" in panel "ORBGEN 3.1: Options".

Further reading:

> Section 4.3: "Precise Orbit Files", page 90
> Section 4.4: "IGS and IERS Pole Files", page 91
> Chapter 5: "Preparation of Earth Orientation, GNSS Orbit, and Satellite Clock Information", page 107

PART 3: SINGLE-POINT-POSITIONING SUMMARY

The output of program CODXTR gives an overview of the performance of CODSPP synchronizing the receiver clocks to GPS system time using pseudorange measurements. It is expected that the RMS is smaller than 0.50 meter. No additional exclusion periods should be reported in this part. If additional exclusion periods are reported, however, they may be caused by serious data problems of a specific station (please have a closer look to its results later in the protocol) or some problems/inconsistencies with the introduced satellite clock file (if all stations are listed for a specific interval).

```
===============================================================================
PART 3: SINGLE-POINT-POSITIONING SUMMARY
===============================================================================

...

  13 FILES, MAX. RMS:    0.34 M FOR STATION: ONSA 10402M004
            MAX. BAD:    4.54 % FOR STATION: WSRT 13506M005


----------------------------------------------------------------
>>> CPU/Real time for pgm "CODXTR": 0:00:00.005 / 0:00:00.004
>>> Program finished successfully
```

Further reading:

> Section 6.3: “Receiver Clock Synchronization and Preprocessing of Code Observations”, page 135

PART 4: DATA SCREENING SUMMARY

In the case of the extended version of the PPP (`PPP_DEMO.PCF`), there are two options for data screening implemented. They are chosen using the BPE server variable `V_HRCLK` (see Section 23.4.1.5). The title line in the protocol indicates which of the two options have been used:

```
================================================================================
PART 4: DATA SCREENING SUMMARY (phase data screening with MAUPRP)
================================================================================

 SUMMARY OF THE MAUPRP OUTPUT FILE
 *********************************

 SESS FIL OK?  ST1  ST2 L(KM)   #OBS.    RMS    DX     DY     DZ    #SL   #DL   #MA   ...
 ------------------------------------------------------------------------------------ ...
 2070   1 OK   GANP          0   24060      9   1204     395    -504     0    31    31   ...
 2070   2 OK   HERT          0   24468     23    632     137    1275     0   140    40   ...
...
```

If the phase measurements have been preprocessed by MAUPRP, this section starts with the extraction summary from MPRXTR. You should be attentive to the RMS of the epoch–difference solution which should be smaller than 20 to 30 mm. If all stations show large RMS values, a problem with the satellite clock corrections is likely the source. The improvements for the station position (DX, DY, DZ) may be larger than usual because the a priori coordinates stem from the RINEX observation headers.

The alternative way is the screening of the phase data using the program RNXSMT. In this case, no section reporting on the preprocessing by MAUPRP appears in the output. Note that the basic version of the PPP example (`PPP_BAS.PCF`) only supports this case.

Independent of the way of preprocessing the phase data, this section contains the output and summary files of the programs RESCHK and RESRMS. They provide some statistics on the residual screening (RMS before and after screening, number of removed measurements).

```
...

--------------------------------------------------------------------------------
PRN | % Observations   Difference   | # Observations   Difference   | RMS
    | before   after   abs      rel | before    after  abs      rel | bef    aft
----+-------------------------------+-------------------------------+-----------
  1 |   0.67    0.67   -0.00  -0.38 |    484     482       -2  -0.41 |  1.2   1.2
  2 |   3.72    3.72   -0.00  -0.01 |   2673    2672       -1  -0.04 |  0.9   0.9
  3 |   3.00    3.00    0.00   0.03 |   2153    2153        0   0.00 |  0.9   0.9
  4 |   3.55    3.55    0.00   0.03 |   2551    2551        0   0.00 |  0.9   0.9
  5 |   3.59    3.59    0.00   0.03 |   2581    2581        0   0.00 |  0.9   0.9
  6 |   3.00    3.00   -0.00  -0.02 |   2156    2155       -1  -0.05 |  0.9   0.9
  7 |   3.38    3.38   -0.00  -0.01 |   2427    2426       -1  -0.04 |  0.9   0.9
...
 32 |   3.00    3.00    0.00   0.03 |   2155    2155        0   0.00 |  0.8   0.8
----+-------------------------------+-------------------------------+-----------
TOT | 100.00  100.00    0.00   0.00 |  71886   71865      -21  -0.03 |  0.9   0.9
--------------------------------------------------------------------------------
```

The tables are useful to detect whether a certain problem is related to a station (large RMS along a row) or a satellite (large RMS within a column). The RMS after the screening (second table of the two residual summary tables) should be in the order of 1 mm or slightly below.

```
 BASELINE  SESS     1     2     3     4     5     6     7     8     9    10    11   ...  TOT
 -------------------------------------------------------------------------------- ... ----
 GANP      2070          0.9   0.9   1.0   1.0   0.9   1.0   1.0   1.0   1.0   0.8   ...  1.0
 GANP      2070          0.2   0.5   0.4   0.2   0.7   0.1   0.4   0.9   0.2   0.3   ...  0.5
 HERT      2070          0.9   0.7   0.9   0.9   0.8   1.1   1.1   0.8   0.8   0.8   ...  0.9
 HERT      2070          0.1   0.4   0.1   0.2   0.3   0.2   0.1   0.5   0.1   0.3   ...  0.3
...
 ZIMM      2070          1.1   1.1   1.0   1.1   1.3   1.2   1.1   0.7   0.8   0.9   ...  1.1
 ZIMM      2070          0.1   0.2   0.1   0.1   0.6   0.4   0.0   0.2   0.2   0.5   ...  0.4
 -------------------------------------------------------------------------------- ... ----
 TOTAL RMS:       1.2    0.9   0.9   0.9   0.9   0.9   0.9   0.8   0.9   0.8   0.8   ...  0.9
 -------------------------------------------------------------------------------- ... ----
 TOTAL OBS:       0.7    3.7   3.0   3.5   3.6   3.0   3.4   3.2   3.0   3.2   2.9   ...
 -------------------------------------------------------------------------------- ... ----
 TOT OBS (1E0):   484   2673  2153  2551  2581  2156  2427  2295  2143  2278  2075   ...
--------------------------------------------------------------------------------- ... ----
```

You should also verify that not more than 3% of the data are removed. Otherwise, you have to find out whether there are some consistency problems between your orbit and satellite clocks or whether a station (or several) has introduced the problem (e.g., due to tracking problems of the receiver).

Further reading:

Section 6.5: "Preprocessing Phase Observations", page 144
Section 7.4.4: "Real and Normalized Residuals", page 174
Section 6.6: "Screening of Post–Fit Residuals", page 161
Section 6.7: "Marking of Observations", page 167

PART 5: PPP SOLUTIONS STATISTICS

The output summary of GPSEST after the residual screening is provided for each station. Two observation files (code and phase measurements) should contribute (apart from the case you have selected a specific observation type via the BPE server variable V_OBSTYP). The RMS is expected to be 1 mm or slightly smaller.

```
================================================================================
PART 5: PPP SOLUTIONS STATISTICS
================================================================================

 EDL2070GANP.OUT  Rms:  0.8 , # fil.:   2 , # obs.:    5416 , # par.:   381
 (DOY: 207)

 EDL2070HERT.OUT  Rms:  0.7 , # fil.:   2 , # obs.:    5465 , # par.:   388
 (DOY: 207)

...
```

You may check the number of observations per stations. If one of the stations contains significantly less measurements you should check first the data availability in PART 1. If this cannot explain the deviation, you should check the residual screening procedure for this station carefully. The second table in this section informs about the improvement of the estimated coordinates (columns DH, DN, DE) with respect to the a priori values of CODSPP (code–only solution) due to the PPP processing.

```
...
 BASELINE  #OBS.  #AMB  RMS(MM) ...  DH(MM) +-    DN(MM) +-    DE(MM) +-    DL(MM) +-    DS(MM)
 ------------------------------ ... ----------------------------------------------------------
 GANP2070  5416    60     0.8   ... -505.5  0.9    60.9  0.3   -13.7  0.5     0.0  0.0   509.3
 HERT2070  5465    67     0.7   ... -810.3  0.8    16.9  0.3   -41.3  0.5     0.0  0.0   811.5
...
```

The number of ambiguities (#AMB) is expected to be in the order of 60 to 80 for a GPS–only solution and around 100 for a combined GPS/GLONASS solution. Exceptional high numbers of ambiguities (above 150 when processing GPS and GLONASS together) may also indicate data quality problems (see PART 1) or that the preprocessing options need to be adjusted for the particular set of observations.

Further reading:

Chapter 10: “Station Coordinates and Velocities”, page 241
Section 10.6: “Precise Point Positioning”, page 265
Section 7.9.2: “The Output Extraction Program GPSXTR”, page 189

PART 6: VERIFICATION OF RECEIVER TRACKING TECHNOLOGY

For each station the receiver tracking technology is verified by estimating P1-C1_MULTIPLIERS. These are the factors needed to apply the DCB for the tracking technology of the receiver (see Chapter 14 for more details).

```
================================================================================
PART 6: VERIFICATION OF RECEIVER TRACKING TECHNOLOGY
================================================================================

 GANP 11515M001  -G   2.826   0.021   C1/P2  13.130  85.612   TRIMBLE NETR8         C1/P2   OK
 HERT 13212M010  -G   2.503   0.019   C1/P2   2.284  80.202   LEICA GRX1200GGPRO    C1/P2   OK
 JOZ2 12204M002  -G   2.489   0.024   C1/P2   2.364  61.789   LEICA GRX1200GGPRO    C1/P2   OK
 LAMA 12209M001  -G   2.546   0.020   C1/P2   0.009  75.838   LEICA GRX1200GGPRO    C1/P2   OK
 MATE 12734M008  -G   2.454   0.025   C1/P2   3.677  58.376   LEICA GRX1200GGPRO    C1/P2   OK
 ONSA 10402M004  -G   0.317   0.023   P1/P2  14.088  30.308   JPS E_GGD             P1/P2   OK
 PTBB 14234M001  -G  -0.096   0.027   P1/P2   3.542  40.428   ASHTECH Z-XII3T       P1/P2   OK
 TLSE 10003M009  -G   2.851   0.023   C1/P2  13.374  81.085   TRIMBLE NETR5         C1/P2   OK
 WSRT 13506M005  -G  -0.091   0.022   P1/P2   4.056  48.728   AOA SNR-12 ACT        P1/P2   OK
 WTZR 14201M010  -G   2.503   0.030   C1/P2   1.443  50.905   LEICA GRX1200GGPRO    C1/P2   OK
 WTZZ 14201M014  -G   0.335   0.023   ?1/?2  14.655  29.028   TPS E_GGD             P1/P2
 ZIM2 14001M008  -G   2.891   0.025   C1/P2  13.991  76.571   TRIMBLE NETR5         C1/P2   OK
 ZIMM 14001M004  -G   2.608   0.021   C1/P2   2.956  76.534   TRIMBLE NETRS         C1/P2   OK
```

Note: this section is only available if you have included code measurements (V_OBSTYP == 'CODE' or 'BOTH').

Further reading:

Section 14.3.4: “Verification of the Receiver Tracking Technology”, page 346

PART 7: COMPILATION OF STATION INFORMATION

In this section you find the station equipment information extracted from the resulting SINEX file. It can be used to verify whether you have used the correct receiver and antenna type (eventually serial number) and the correct antenna height.

PART 8: STATION COORDINATE RESULTS

In this section the estimated coordinates from the PPP at the epoch of the observations are given (content of the file {V_C}yyssss.CRD; V_C is defined as a BPE server variable).

The second part is the comparison between the PPP coordinate solution and a priori values given in the file {V_CRDINF}.CRD. It is useful to check the deviation of the current PPP solution from the expected coordinates, e.g., obtained from PPP of previous days or from external sources, e.g., ITRF.

If the BPE server variable V_UPD == 'Y', the coordinates resulting from the PPP are propagated to the epoch given in V_REFEPO using the velocities derived from the NNR–NUVEL–1A model.

PART 9: STATION CLOCK ESTIMATION

This section contains the overview of the performance of the receiver clocks as provided by the statistics part of the CCRNXC program output.

```
================================================================================
PART 9: STATION CLOCK ESTIMATION
================================================================================

...

 REFERENCE CLOCK SELECTION FOR OUTPUT FILE
 -----------------------------------------

 Selected reference station:     WSRT 13506M005
 Valid for interval              2010-07-26 00:00:00 to 2010-07-26 23:55:00

...

 STATISTICS ON THE CLOCKS IN THE OUTPUT FILE
 -------------------------------------------
                          # per file         rms of poly. fit (ns)
 Clock name               out    ...         n = 0    n = 1    n = 2
 ------------------------------- ... ------------------------------
 WSRT 13506M005           288    ...         4.322    0.000    0.000
 WTZR 14201M010           288    ...         1.827    0.422    0.411

...
```

The first clock in the list was selected as reference clock (best performing clock in the series). If you have more external high performance clocks driving the GNSS receivers, you should find small numbers for the `rms of poly. fit` with `n=1` (linear clock model). For H–Maser clocks an RMS smaller than 0.05 ns, for Cesium clocks an RMS smaller than 0.10 ns is expected. Do not care about large or even huge RMS values for quartz or other low quality (internal) clocks in the receivers.

Further reading:

Section 15.3: “Clock RINEX Utilities in *Bernese GNSS Software*”, page 365

PART 10: GEOCENTER COORDINATES, Extended Version only

This part is only available if `V_G` is not blank.

```
================================================================================
PART 10: GEOCENTER COORDINATES
================================================================================

 GCC102070.OUT    Rms:  0.9 , # fil.:  26 , # obs.:   71868 , # par.:  5127
 (DOY: 207)


 55403.00  55404.00    -0.00128  0.00031  0.00539     0.00025  0.00023  0.00026
```

It contains the RMS (a posteriori sigma of unit weight) for the normal equation–based combined PPP solution for all stations. The RMS is expected to be close to 1 mm if code and phase measurements are used (V_OBSTYP = BOTH). At least one station with coordinates in the external reference frame (given in `V_REFINF`) should be included in the PPP station list to compute the translation of the PPP solution with respect to the external reference frame given in the line below (time interval from/to, translation in X,Y,Z and the formal uncertainty of the translations).

Further reading:

Section 16.4: “Estimation of Earth Orientation and Geocenter Parameters”, page 385

PART 11: PSEUDO-KINEMATIC SOLUTION, Extended version only

This solution is only generated if V_H contains a non–blank value.

```
===============================================================================
PART 11: PSEUDO-KINEMATIC SOLUTION
===============================================================================


 KIN2070GANP.OUT  Rms:  1.7 , # fil.:   2 , # obs.:    5416 , # par.:  1229
 (DOY: 207)

*AVG            -0.0007         0.0017         0.0017
*SIG             0.0065         0.0076         0.0127
*RMS             0.0064         0.0074         0.0126
 ...
```

The first two lines are the usual extractions of a GPSEST/ADDNEQ2 run, reporting mainly the RMS (a posteriori sigma of unit weight) and the number of observations and parameters. When looking at the resulting degree of freedom you should keep in mind that half of the observations are down–weighted pseudorange measurements (if you have selected V_OBSTYP = BOTH). This means that you have about twice as much measurements as parameters.

The subsequent lines are an extract from the kinematic processing summary provided by GPSXTR. They contain the difference of the coordinate solution for a certain epoch with respect to the given a priori coordinate which is the PPP solution with one set of coordinate parameters for the entire processing interval. From these differences three statistical indicators are derived: the arithmetic mean (*AVG), the standard deviation of the differences (*SIG), and the RMS of the differences of the coordinate solution for each epoch with respect to the average over all epochs (*RMS). For all three parameters the values are given in the three local components north, east, and up.

Further reading:

Section 10.5: “Estimating Kinematic Coordinates”, page 260
Section 7.9.2: “The Output Extraction Program GPSXTR”, page 189

PART 12: HIGH-RATE TROPOSPHERE, Extended Version only

This section is only available if V_I has a non–blank value.

```
===============================================================================
PART 12: HIGH-RATE TROPOSPHERE
===============================================================================

 TRP102070.OUT    Rms:  0.9 , # fil.:  26 , # obs.:   73737 , # par.:  6645
 (DOY: 207)


 TRP2070GANP.OUT  Rms:  0.8 , # fil.:   2 , # obs.:    5416 , # par.:   498
 (DOY: 207)

 TRP2070HERT.OUT  Rms:  0.7 , # fil.:   2 , # obs.:    5465 , # par.:   505
 (DOY: 207)

 ...
```

It contains the RMS (a posteriori sigma of unit weight) first of the collecting ADDNEQ2–run and below of all individual GPSEST runs of the station–by–station processing. Note: in both programs strong relative constraints for the troposphere parameters are set up to stabilize the system. It is preferable to have a higher sampling rate (e.g., 30 instead of 300 seconds) for this type of application.

Further reading:

Chapter 12: “Troposphere Modeling and Estimation”, page 295

PART 13: IONOSPHERE ESTIMATION AND RECEIVER DCB CALIBRATION, Extended Version only

The station–specific and regional ionosphere models are only included in this section for a daily processing scheme and if the variable `V_K` has a non–blank value.

```
===============================================================================
PART 13: IONOSPHERE ESTIMATION AND RECEIVER DCB CALIBRATION
===============================================================================

 ION2070GANP.OUT  Rms: 18.6 , # fil.:   2 , # obs.:    4788 , # par.:    90
 (DOY: 207)

 ION2070HERT.OUT  Rms: 18.2 , # fil.:   2 , # obs.:    4929 , # par.:    93
 (DOY: 207)

 ...
```

The section on the ionosphere estimation starts with the RMS (a posteriori sigma of unit weight) for the station–specific ionosphere models generated in a station–by–station PPP. Note: the RMS can be higher for the geometry–free (L_4) linear combination compared to the ionosphere–free (L_3) linear combination used for positioning purposes.

```
 MODEL / STATION      MIN LAT  MAX LAT   MEAN TEC  RMS ERR   MAX TEC  RMS ERR  LAT     LON    ...
                      (DEG)    (DEG)     (TECU)    (TECU)    (TECU)   (TECU)   (DEG)   (DEG) ...
 ----------------------------------------------------------------------------------------------- ...
 GANP 11515M001        34.39    59.62       ---       ---       8.27     0.33   59.60   97.75 ...
 HERT 13212M010        40.13    66.11       ---       ---       5.79     0.31   66.10   38.50 ...
 ...
```

Below this section you find some statistics for each of the station–specific ionosphere models and the resulting P1−P2 DCBs (which are computed individually for each involved GNSS).

```
 MODEL / STATION      MIN LAT  MAX LAT   MEAN TEC  RMS ERR   MAX TEC  RMS ERR  LAT     LON    ...
                      (DEG)    (DEG)     (TECU)    (TECU)    (TECU)   (TECU)   (DEG)   (DEG) ...
 ----------------------------------------------------------------------------------------------- ...
 2070-00               27.25    70.23       ---       ---       5.92     0.28   27.28   19.90 ...
```

Below you find the analogous information for the computation of the regional ionosphere model computed from the observation files used for the PPP so far. Remark: since for the geometry–free linear combination only the satellite orbits but not the satellite clock corrections are needed, it might be that you have included here also the measurements from an alternative GNSS which is not used for the other PPP steps because of missing satellite clock information.

Further reading:

Chapter 13: “Ionosphere Modeling and Estimation”, page 311
Section 13.3.1.1: “Ionosphere Mapping”, page 318
Chapter 14: “GNSS Related Biases”, page 339

23.4.2. Double–Difference Network Processing Example (RNX2SNX.PCF)

23.4.2.1. Purpose and Description

The BPE RNX2SNX.PCF is designed for a double–difference based analysis of RINEX GNSS observation data from a regional network. Station coordinates and troposphere parameters are estimated and stored in Bernese and SINEX format to facilitate further processing and combination. For each session, the corresponding normal equation information is saved for a subsequent multi–session solution (allowing the estimation of station velocities).

Some important features of this BPE are:

- Observation files with significant gaps or unexpectedly large residuals will automatically be removed from the processing to ensure a robust processing and a reasonable network solution.
- An advanced ambiguity resolution scheme is included:
  - Code–Based Wide–Lane (WL) Ambiguity Resolution
    For baselines shorter than V_BL_AMB, e.g., 6000 km, a Melbourne–Wübbena wide–lane (after checking the residuals of the code observations for outliers) and a narrow–lane ambiguity resolution is performed.
    Note: for this ambiguity resolution strategy good quality code measurements are required.
  - Phase–Based Wide–Lane (L_5) Ambiguity Resolution
    For baselines shorter than V_BL_L53, e.g., 200 km, the code–based wide–lane ambiguity resolution is replaced by a phase–only wide–lane with a subsequent narrow–lane ambiguity resolution.
  - Quasi–Ionosphere–Free (QIF) Ambiguity Resolution
    The QIF–strategy is applied on the remaining real–valued ambiguities for baselines shorter than V_BL_QIF, e.g., 2000 km.
  - Direct L_1/L_2 Ambiguity Resolution
    A direct L_1/L_2 ambiguity resolution is applied instead of the above mentioned sequence of strategies on very short baselines (shorter than V_BL_L12, e.g., 20 km).
- The resulting coordinates of the session are compared with the reference frame coordinates via a Helmert transformation (three translations due to the degree of freedom for the datum definition of a regional network). In addition, a comparison of all resulting coordinates with the previous solutions is done.
- The resulting SINEX data should allow for both the reconstruction of the unconstrained, free network solution and for the straightforward extraction of station coordinates of the originally computed minimum–constraint solution.

23.4.2.2. Input Files

The input files are copied from the DATAPOOL area (${D}) into campaign area in the BPE user script R2S_COP:

- **External products files** (mandatory)

  ```
  ${D}/{V_B}/{V_B}wwwwd.PRE
  ${D}/{V_B}/{V_B}wwww7.ERP or ${D}/{V_B}/{V_B}wwwwd.ERP
  ```

  The variable `V_B` (see Section 23.4.2.5) defines the product series to be considered for the double–difference processing (e.g., `COD` or `IGS`), where `wwwwd` stands for the GPS week and the day of week to be processed.

  Note: Orbit and ERPs should be consistent for the orbit integration.

- **Bernese formatted products files** (mandatory)

  ```
  ${D}/BSW52/P1C1yymm.DCB
  ```

  The DCB corrections are necessary to unify the reference for the different receiver tracking techniques. This may become important for the Melbourne–Wübbena linear combination with two receivers using different tracking technologies in one baseline (`yymm` is the 2-digit year and the 2-digit month of the processing session).

  ```
  ${D}/BSW52/CODwwwwd.ION
  ```

  The ionosphere file is needed to correct for the HOI terms and for the ambiguity resolution.

  Both files can be downloaded from `http://www.aiub.unibe.ch/download/CODE/`.

- **Reference frame files** (mandatory)
  The reference frame in the project–specific coordinate file is introduced by adding/overwriting the station coordinates with the values given in the reference frame files. The reference frame coordinate and velocity files are named:

  ```
  ${D}/{V_REFDIR}/{V_REFINF}_R.CRD
  ${D}/{V_REFDIR}/{V_REFINF}_R.VEL
  ```

- **Project–specific station–related files** (mandatory)
  For each station to be processed in the project, the following information has to be provided. Note that the information can (at least mostly) be provided by a PPP procedure (e.g., by applying the related BPE, see Section 23.4.1).

  To derive the a priori coordinate for the correct epoch of the processing, coordinates and velocities are needed:

  ```
  ${D}/{V_REFDIR}/{V_CRDINF}.CRD
  ${D}/{V_REFDIR}/{V_CRDINF}.VEL
  ```

  For each station a unique 4- and 2-character abbreviation is necessary to manage the observation file naming. This information is taken from the abbreviation table:

  ```
  ${D}/{V_REFDIR}/{V_CRDINF}.ABB
  ```

  The station equipment history is expected in the file

  ```
  ${D}/{V_REFDIR}/{V_CRDINF}.STA
  ```

  to guarantee the correct handling of antenna types (antenna phase center corrections) and receiver types (bias related to the tracking technology).

  ```
  ${D}/{V_REFDIR}/{V_CRDINF}.CLU
  ```

  is expected for cluster handling in SNGDIF, see Section 6.4.2 .

- **RINEX header inconsistency acceptance file** (optional)
  The station information file (`{V_CRDINF}.STA`) is used to verify the header information of the RINEX observation files to be processed. If there are known inconsistencies, they can be documented in this file. The program will accept RINEX files with such entries without further notice. The file with the accepted RINEX header inconsistencies is copied from

  ```
  ${D}/{V_REFDIR}/{V_CRXINF}.CRX
  ```

  as soon as the variable (`V_CRXINF`) contains a non–blank value.

- **Tidal loading corrections** (optional)

  ```
  ${D}/{V_REFDIR}/{V_BLQINF}.BLQ
  ${D}/{V_REFDIR}/{V_ATLINF}.ATL
  ```

  Ocean and atmospheric tidal loading may be corrected during the double–difference analysis if the corresponding corrections are provided in the files.

  Note: these files must also contain corresponding CMC values for the translation of the orbit positions from the Earth-fixed system in the center–of–mass system for the orbit integration.

All these files are copied from the `DATAPOOL` area into the campaign area by using the Perl method `copyRef` provided by the modules `${BPE}/bpe_util.pm`. This method only copies the files if the source is newer than the target. In this way it is possible to run several sessions in parallel without interactions due to potentially incomplete files during the copy process.

23.4.2.3. Observation Selection

The observation files are expected in RINEX format either in the directory `${D}/{V_RNXDIR}/` for version 2 and/or `${D}/{V_RX3DIR}/` for version 3 files. The station/observation file selection is realized in the BPE user script `RNX_COP`.

Three possibilities to select stations for processing are implemented (considered in the following order):

1. A file with the name `{V_OBSSEL}.CLU` exists in the `STA` directory of the current campaign. The name of the cluster file is defined by the BPE server variable `V_OBSSEL` (see Section 23.4.2.5).

   All station names from this file are translated into the 4-character IDs using the abbreviation table `{V_CRDINF}.ABB` (also expected to be in the `STA` directory of the current campaign). These 4-character IDs are used to identify corresponding RINEX files from the source directory in the data base.

2. A file named `{V_OBSSEL}` resides directly in the top–level directory of your current campaign. It contains the RINEX abbreviations of the files to be processed. They have to be separated by blank characters or may start with a new line:

   ```
     GANP
     HERT
     JOZ2
     LAMA
     ZIM2 ZIMM
   ```

   (In this example, six stations are selected for the processing).

3. If the BPE server variable V_OBSSEL is empty, all RINEX files of the specified session(s) are processed.

The selected RINEX files are copied and prepared for processing by the subroutine crz2rnx provided by the Perl module ${BPE}/bpe_util. The files are decompressed using the Hatanaka tool and/or any other usual compressing tool. For this purpose, the Hatanaka decompression tool CRX2RNX or CRX2RNX.exe is expected to be installed on your system and its location must be included in the system $PATH or %PATH% variable — the search path for executable programs. The same is expected for the decompression tool gzip or gzip.exe to handle the potential generic compression of the files.

Usually, only the RINEX files of the current session are processed. If you are going to process hourly sessions (session identifier must end with A..X — see method isHourly in ${BPE}/bpe_util.pm), all hours defined by the range of the last V_HOURS (see BPE server variables, Section 23.4.2.5) are concatenated and processed together under the label of the current session (of course the corresponding input files are copied, too).

Depending on the content of the BPE server variables V_RNXDIR and V_RX3DIR, the corresponding RINEX files are considered in the related format versions 2 or 3 or even both. In case of RINEX version 3, the conventional as well as the new long filenames are supported. If multiple versions of the same dataset are available, the selection is based on the given priority list defined in user script RNX_COP. The following preference list (in descending order) was defined:

1. long filename, RINEX version 3; generated by the receiver
   (indicated by data source flag R)
2. long filename, RINEX version 3; extracted from a data stream
   (indicated by data source flag S)
3. long filename, RINEX version 3 from an unknown source
   (indicated by data source flag U)
4. short filename, RINEX version 3 from an unknown source
5. RINEX version 2 (only short filenames available)

This priority list can of course be adapted according to your needs by modifying the value of the variable

```
my @rx3Typ = ("R","r","S","s","U","u","short","rx2");
```

in the user script RNX_COP.

23.4.2.4. Result Files

The most important result files are copied by the BPE user script R2S_SAV into the ${S}/{V_RESULT} area (V_RESULT is defined below in the BPE server variables area) if the BPE server variable V_SAV == 'Y'. The files are located in yearly subdirectories (yyyy). These files are:

- ${S}/{V_RESULT}/yyyy/OUT/R2Syyssss.PRC
  Main protocol file containing the summary of the processing steps (more details are provided in the Section 23.4.2.7).
- ${S}/{V_RESULT}/yyyy/STA/{V_E}yyssss.CRD
  Resulting station coordinates for the stations from the ambiguity resolved, minimum constraint network solution (epoch of the processed data).

- `${S}/{V_RESULT}/yyyy/ATM/{V_E}yyssss.TRP`
  `${S}/{V_RESULT}/yyyy/ATM/{V_E}yyssss.TRO`
  Troposphere estimates (Bernese troposphere format and troposphere SINEX format) as obtained from the ambiguity–fixed network solution.
- `${S}/{V_RESULT}/yyyy/SOL/{V_E}yyssss.NQ0.gz`
  Normal equation file (gnu–compressed) with the coordinate and troposphere parameters of all stations of the double difference solution.
- `${S}/{V_RESULT}/yyyy/SOL/{V_F}yyssss.NQ0.gz`
  `${S}/{V_RESULT}/yyyy/SOL/{V_E}yyssss.SNX.gz`
  Reduced version of the normal equation with pre–eliminated troposphere parameters containing, e.g., only the coordinate parameters.

Note: the BPE server variables `V_E` and `V_F` are set by default to the identifiers `F1_` and `R1_`. The other solution identification variables are also defined as variables, see Section 23.4.2.5.

23.4.2.5. BPE Server Variables

There are several variables assigned with a default value in the PCF (last section of the file). When starting a BPE manually they may be modified in the panel "RUNBPE 4: Server Variables Sent to Client".

Input and Result Filenames and Their Locations

It is assumed that all station–related files have the same name related to a certain project. They contain all station coordinates, equipment information, and so on. For the processing with the double–difference example BPE they are specified via the following BPE server variables:

V_REFDIR (default: REF52)
: Directory in the `DATAPOOL` area (`${D}`) where the basic Bernese files for the processing are located. The detailed list of files is given in Section 23.4.2.2 on input files.

V_CRDINF (default: EXAMPLE)
: The basename of the files used as basic Bernese input files for this double–difference processing example. Copying these files is managed in the BPE user script `R2S_COP`.

V_CRXINF (default: empty)
: Inconsistencies between the RINEX observation file headers and the station information file can be recorded in a file with "Accepted station information" inconsistencies in the program RXOBV3. If you need such a file for your processing, you can specify its name here. It will be copied from `${D}/REF52/{V_CRXINF}.CRX` into the campaign.

In general, it is recommended to use the same project–related filename for all input files of the processing. Nevertheless, a few files are related to specific options. Hence, separate variables are used for them:

V_BLQINF (default: EXAMPLE)
: The basename of the file containing the ocean tidal loading corrections for the stations. The file also has to contain the CMC related to ocean tidal loading. These corrections are needed for the orbit generation.

If the variable is empty, no displacement corrections to the stations and no CMC for the orbit generation are applied. The file `${D}/REF52/{V_BLQINF}.BLQ` is copied by the script `R2S_COP` into the campaign.

V_ATLINF (default: EXAMPLE)
The basename of the file containing the atmospheric tidal loading corrections for the stations. The file must also contain the CMC related to atmospheric tidal loading. These corrections are needed for orbit generation.

If the variable is empty, no displacement corrections to the stations and no CMC for the orbit generation are applied. The file `${D}/REF52/{V_ATLINF}.ATL` is copied by the script `R2S_COP` into the campaign.

V_HOIFIL (default: HOI$YSS+0)
Specify the filename used for the ionosphere model within your campaign area that is used to compute the HOI corrections. The ionosphere model is copied from `${D}/BSW52/CODwwwwd.ION` into the `ATM`–directory of the campaign using the given filename.

If the variable is empty, the HOI corrections are disabled.

The following BPE server variables are related to the reference frame to be used for the processing:

V_REFINF (default: IGB08)
Reference frame files used as master files when updating coordinate and velocity files (`${D}/{V_REFDIR}/{V_CRDINF}.CRD` and `${D}/{V_REFDIR}/{V_CRDINF}.VEL`). The corresponding files are expected as

```
${D}/{V_REFDIR}/{V_REFINF}_R.CRD
${D}/{V_REFDIR}/{V_REFINF}_R.VEL
```

The first sequence of the double–difference processing example replaces coordinates for stations given in these reference frame files in the project–related coordinate file.

Another BPE server variable defines the location of the result files:

V_RESULT (default: RNX2SNX)
Directory in the `SAVEDISK` area (`${S}`) where the result files of the double–difference processing example are collected in yearly subdirectories (saving of the result files is managed by the BPE user script `R2S_SAV`).

To write SINEX result files the header information (e.g., institution name, hardware used, contacts, and information about the project) is taken from a skeleton file.

V_SNXINF (default: SINEX.RNX2SNX)
The SINEX header file contains the general header information used when you want to create SINEX files, e.g., institution name, hardware used, contacts, and information about the project. With this variable this header file may be selected for all BPE scripts where a SINEX file (with station coordinates) or a troposphere SINEX file is generated.

Observation Selection

There are three variables related to the selection of the observation files. The procedure is described in Section 23.4.2.3.

V_RNXDIR (default: RINEX)
Directory in the `${D}` (`DATAPOOL`) area where the RINEX files in the format version 2 are expected.

V_RX3DIR (default: empty)
Directory containing RINEX files in format version 3 in the `${D}` (`DATAPOOL`) area.

V_OBSSEL (default: empty)
This variable may contain a filename for a cluster file or a file with the RINEX station abbreviations.

The following variables are responsible for the data selection:

V_OBSINF (default: OBS.SEL)
This variable contains the filename of the receiver type related observation types priority list which is used when importing `RINEX3` data files, located in the the `${X}/GEN/` directory. The observation selection is done in RNXSMT. More details are given in Section 24.3.7 .

V_SATSYS (default: GPS/GLO)
Select the system(s) (usually GPS or GPS/GLO) to be used for the double–difference processing example. The value of this variable is directly introduced in the option "Satellite system to be considered" in panel "RXOBV3 2: Input Options 1" (user script `RXOBV3_H`, PID 222).

Note: whether all selected systems are really used for the processing depends also on the availability of satellite orbits as well as on the content of the RINEX observation file. This option acts only as a filter to remove non–selected systems from the observation files.

V_SAMPL (default: 180)
Sampling interval for the final solution. The value of this variable is directly introduced as the option "Sampling interval" in panel "GPSEST 3.1: General Options 1" in the GPSEST processing program where the final solution is generated (user script `GPSCLU_P`, PID 502).

V_HOURLY (default: 8)
In case of an hourly session definition, a number of hours can be specified that are prepended to the current session to stabilize the estimation of ambiguity parameters and to improve the capability to resolve the ambiguities. It should not be much shorter than the length of a satellite path, e.g., 6 hours. This variable is obsolete if a daily session definition is given.

General Model and Standard File Selection

In this sequence the BPE server variables related to the general processing files are listed:

V_PCV (default: I08)
Selection of the antenna phase center model identifier.

V_SATINF (default: SATELLIT)
Basename of the satellite information file. The resulting filename is `${X}/GEN/{V_SATINF}.{V_PCV}` .

V_PCVINF (default: PCV)
Basename of the antenna phase center correction file. The resulting filename is `${X}/GEN/{V_PCVINF}.{V_PCV}` .

V_SATCRX (default: SAT_$Y+0)
Name of the satellite problem file.

V_RECINF (default: RECEIVER.)
This variable contains the filename of the receiver characterization file located in the the `${X}/GEN/` directory. More details are given in Section 24.3.6 .

Another capability of the example BPE is to update the antenna phase center correction file based on an ANTEX file. This feature is managed by a BPE server variable as well:

V_MYATX (default: empty)
Filename (including extension) for an ANTEX file containing receiver and possibly satellite antenna phase center corrections. The file is used to update the Bernese phase center file in `${X}/GEN/{V_PCVINF}.{V_PCV}`, e.g., in case of new antennas in the network.

It is expected that the ANTEX file is available in `${X}/GEN/` .

If you plan to use a file containing individually calibrated antennas you need to change the settings for "Set number of generic antennas to 999999" in panel "ATX2PCV 2: ANTEX Conversion" (Script 002 `ATX2PCV`):

- checkbox unmarked means "type–specific calibrations"
- checkbox marked means "antenna–specific calibrations"

See also Section 18.2.4: "Technicalities on the Receiver Antenna Handling".

There are two variables defining a range of sessions to be considered in order to compare the coordinate solution from the current session with the results from previous sessions realized in the user script `COMPARF`, PID 513 .

V_MINUS (default: -6)
Start of the range of sessions

V_PLUS (default: 0)
End of the range of sessions

Ambiguity Resolution

For the ambiguity resolution the following BPE server variables are relevant:

V_GNSSAR (default: ALL)
The content of this variable is filled into the selection of systems for the ambiguity resolution (option "Selection of GNSS for ambiguity resolution" in panel "GPSEST 3.2: General Options 2"). It may either contain a single GNSS (GPS or GLONASS) to allow the ambiguity resolution only for this particular system or it can be ALL to resolve all ambiguities.

If the variable is empty, the default value in the original program input file is kept.

According to the length of the baselines, different strategies for the ambiguity resolutions are recommended (see Table 8.1). The ranges of baseline lengths are defined via a series of BPE server variables:

V_BL_AMB (default: 6000)
The variable contains the maximum length of a baseline included in the Melbourne–Wübbena / Narrow–Lane ambiguity resolution scheme. The value of this variable is fed into the program BASLST (running in `GNSAMBAP`) to select the baselines.

V_BL_QIF (default: 2000)
The variable contains the maximum length of a baseline included in the Quasi–Ionosphere–Free (QIF) ambiguity resolution algorithm that is carried out on the ambiguities that were not resolved in the first step. The value of this variable is fed into the program BASLST (running in user script `GNSQIFAP`) for baselines selection.

V_BL_L53 (default: 200)
The variable contains the maximum length of a baseline included in the phase–based wide-/narrow–lane ambiguity resolution scheme.

All already resolved ambiguities are reset to the status “unresolved” to start the ambiguity resolution for this group of baselines from scratch. The value of this variable is fed into the program BASLST (running in `GNSL53AP`) to select the baselines.

V_BL_L12 (default: 20)
The variable contains the maximum length of a baseline included in the direct L1/L2 ambiguity resolution scheme.

All ambiguities already resolved for this baseline are reset to “unresolved” to re–initialize the ambiguity resolution by this algorithm. The value of this variable is fed into the program BASLST (running in `GNSL12AP`) for baseline selection.

Clustering for Parallel Processing

Many of the parallelization scripts (e.g., `RNXSMTAP`, `CODSPPAP`, `MAUPRPAP`) may run in two modes:

1. processing clusters (groups) of stations together in a parallel script or
2. processing station by station in parallel.

The active mode is controlled by the BPE server variable `V_CLU` which defines the maximum number of files in a cluster. If it is blank (i.e. not defined) a station by station parallel processing is initialized. Otherwise, clusters of up to `V_CLU` stations/baselines are formed, which are then processed in jobs running in parallel.

On one hand, the more CPUs (processing cores) are available, the more stations/baselines can be processed in parallel, shortening the overall processing time. On the other hand, the number of parallel jobs is limited by the architecture of your computing system (including disk access times or the number of available CPUs) or by your system administrator.

In general, you can plan one job per available CPU with a quick access to your data partition. If the number of stations/baselines per cluster exceeds 15 to 20, it is worth to divide the data into additional clusters.

In any case, it is important to have approximately the same computing time for each of the clusters because the BPE only continues with the next script when the last of the parallel running jobs has finished.

V_CLU (default: 7)
For all parallel running scripts it is a question on how many files (stations/baselines) should be processed together and how many jobs are running in parallel.

The default value was chosen to process the example (13 stations/12 baselines) on a dual–core machine. It results in two clusters that can processed in parallel at the same time (two processing cores).

V_CLUEDT (default: 5)
The residual screening step (`GPSEDT_P`) is very CPU intensive because several GPSEST runs are included. You may reduce the size of the clusters to speed up the processing.

If the variable `V_CLUEDT` is not specified or if it is empty, the value of `V_CLU` is also used to define the parallelization for the residual screening step.

V_CLUFIN (default: A)
The baselines created in SNGDIF are assigned to cluster numbers according to the file `{V_CRDINF}.CLU`. All baselines where the first of the two stations has the same cluster number are processed together in the final solution considering the correct correlations (script `GPSCLU_P`).

These clusters are transferred from script `SNGDIF` to `GPSCLUAP` by `CLB`–files in the `STA`–directory of your campaign. You may suppress the use of this clustering technique without changing the file `{V_CRDINF}.CLU` by changing the value of the variable `V_CLUFIN` to `N`. In this case all baselines are processed in one cluster.

With the default value `A` the clustering is automatically enabled as soon as more than 100 baselines (without counting the redundant ones in `yyssss00.CLB`, where `yyssss` stands again for the 2-digit year and the session) are processed. The limit may be changed if the variable gets a number behind the `A` indicator (e.g., `A 50` to set the limit to 50).

If you have asked for redundant baselines, they will be listed in the `yyssss00.CLB` file. This cluster is not affected by changing the value. It may even contain closed loops of baselines which is not allowed for the correct correlation strategy in GPSEST — the option "Correlation strategy" is adapted automatically by the script `GPSCLU_P`.

File and Product Naming

The following group of BPE server variables are consistently used as solution identifiers and for file naming within the BPE processing:

V_B (default: COD)
Name of the external product for GNSS orbit and ERP information. It is expected that the products are available in the `DATAPOOL` area (`${D}`) in a specific subdirectory `V_B`; the filenames must also contain this label: `{V_B}wwwwd` (see Section 23.4.2.2 for more details).

V_A (default: APR)
A priori station coordinates as they are extracted from the coordinate and velocity files (`{V_CRDINF}.CRD` and `{V_CRDINF}.VEL`, respectively) for the mean epoch of the session. These coordinates are not changed (updated) during the processing.

V_C (default: P1_)
Label of the program output and result files of the double–difference processing example of the intermediate network solution before resolving the ambiguities: coordinates (`{V_C}yyssss.CRD`), troposphere estimates (`{V_C}yyssss.TRP` in Bernese format and `{V_C}yyssss.TRO` in troposphere SINEX format), and normal equation files (`{V_C}yyssss.NQ0`).

These files are NOT copied to the product archive by the script `R2S_SAV`.

V_E (default: F1_)
Label of the program output and result files from the final processing step with introduced resolved integer ambiguities for the station coordinates (`{V_E}yyssss.CRD`) and troposphere parameters (`{V_E}yyssss.TRP` in Bernese format and `{V_E}yyssss.TRO` in troposphere SINEX format). In addition, Bernese formatted binary normal equation files (`{V_E}yyssss.NQ0`) are prepared. A reduced version of the normal equation file with pre–eliminated troposphere parameters is also available in the SINEX format (`{V_E}yyssss.SNX`).

These five files are copied by the script `R2S_SAV` into the product archive.

V_F (default: R1_)
Label of the program output and result files from a reduced version of the final normal equation file (NQ0) of the current session containing only station coordinates (troposphere parameters are pre–eliminated).

Management of the Dataflow

The last group of BPE server variables is related to the file handling in the campaign:

V_SAV (default: Y)
The BPE user script `R2S_SAV` saves the result files into the `${S}/{V_RESULT}` area if V_SAV == 'Y'.

V_SAVOBS (default: Y)
If V_SAVOBS == 'Y' the phase baseline observation files of the current session together with the cluster files `yyssss??.CLB` (independent of whether the clustering has been used in the current run or not) are copied into session–specific directories of the `SAVEDISK` area: `${S}/{V_RESULT}/yyyy/OBS/ssss`. It is checked whether all observation files listed in the `CLB`–files have been successfully copied to the archive. No compression or archiving tool is used because the most convenient tool depends on the operating system and on the user environment.

V_DEL (default: Y)
The user script `R2S_DEL` deletes all files of the currently processed session from the campaign if V_DEL == 'Y'.

Usually, the deletion should be enabled to keep the campaign clean. For debugging purposes and to identify causes of errors in the processing it may be useful to keep all files.

Skip Sessions in Case of Reprocessing

In case of a reprocessing it might be interesting to repeat only selected sessions for some reasons. A typical scenario could be that the BPE is executed once for all sessions.

Possibly, some sessions will end with an error or with unsatisfying results, e.g., because of some stations with inconsistent equipment records between the station information file and the RINEX header. After fixing these problems the reprocessing needs to be repeated only for a certain subset of sessions. Instead of starting the BPE manually for each of these sessions it is more convenient to start the BPE again for the full interval of the reprocessing and let the availability of a specific file in the `SAVEDISK` area indicate whether a certain session shall be repeated or not.

V_SKIP (default: R2S$YSS+0.PRC; deactivated by R2S$YSS+0.PRC_)

The very first script `SKIP_SES` (PID 000) checks whether a file `${S}/{V_RESULT}/yyyy/OUT/{V_SKIP}` exists, e.g., from a previous processing. The availability of the file indicates the successful processing of the session in a previous iteration of the reprocessing. Therefore, the BPE will directly jump to the last script (`DUMMY`; PID 999). If the indicated file does not exist, the BPE will execute all relevant scripts for the specific session.

We propose to use the protocol file in the `SAVEDISK` area as an indicator.

If a session shall be repeated even if it was executed up to the last script in a previous iteration, the indicator file (given in the `V_SKIP` variable) must be removed in the `SAVEDISK` area.

If all sessions shall be repeated independent of the existence of the indicator file the value of this variable can be left empty or at least be modified, e.g., from `R2S$YSS+0.PRC` to `R2S$YSS+0.PRC_`.

23.4.2.6. Description of the PCF

Copy Required Files, Create A priori Coordinate File, and Prepare the Hourly Processing

Before starting the real analysis of the GNSS measurements, all necessary files must be readily available for the BPE. The task of the first PCF section is to provide these files in the campaign area. In addition, some files may need to be prepared for a potential hourly processing scheme.

```
#
# Skip session (reprocessing)
# --------------------------
000 SKIP_SES R2S_GEN           ANY      1
#
# Copy required files
# -------------------
001 R2S_COP  R2S_GEN           ANY      1 000
002 ATX2PCV  R2S_GEN           ANY      1 001
003 COOVEL   R2S_GEN           ANY      1 001
004 COOVEL   R2S_GE2           ANY      1 001
005 CRDMERGE R2S_GEN           ANY      1 003 004
011 RNX_COP  R2S_GEN           ANY      1 001
021 OBSMRGAP R2S_GEN           ANY      1 011
022 OBSMRG_P R2S_GEN           ANY      1 021
031 ION_MRG  R2S_GEN           ANY      1 011
099 DUMMY    R2S_GEN           ANY      1 002 005 022 031
```

PID 000 SKIP_SES: This script checks for the existence of the file `${S}/{V_RESULT}/yyyy/OUT/{V_SKIP}`. The values for `V_RESULT` and `V_SKIP` are taken from the BPE server variables whereas `yyyy` is the 4-digit year of the current session. If the file exists the BPE directly jumps to the last script. This mechanism may be used, e.g., to manage whether a specific session shall be repeated in the frame of a reprocessing or not.

PID 001 R2S_COP: This script copies all necessary files (apart from the RINEX files) into the respective campaign directories (the list of files is given in Section 23.4.2.2). If appropriate, filenames will be changed according to the BPE variables and session–specific time variables.

The BPE will stop with an error if one of the requested files is missing.

PID 002 ATX2PCV: If the BPE server variable V_MYATX contains the name of an ANTEX file, this file is used to update the "PCV information file" specified by the BPE variables V_PCVINF and V_PCV. The antenna names together with the GNSS supported by the related receiver types are read from the "Station information file" (defined by V_STAINF).
If the resulting "Bernese phase center offsets" file differs from the file currently used for the processing in ${X}/GEN, it is copied to this location.

PID 003 COOVEL: The coordinates in the project–specific coordinate file ({V_CRDINF}.CRD in the STA–directory) may refer to any epoch (that may for instance be given by a BPE variable in the PPP examples). Using the program COOVEL the coordinates are propagated to the epoch of the current session applying the velocities from the corresponding velocity field ({V_CRDINF}.VEL in the STA–directory).

PID 004 COOVEL: Coordinates of the IGS reference stations are given for epoch January 1, 2005, 00:00:00 (cf. IGB08_R.CRD in the STA–directory, specified as {V_REFINF}_R.CRD). The program COOVEL propagates them to the current session's epoch using IGS velocities (see IGB08_R.VEL in the STA–directory).

PID 005 CRDMERGE: The program CRDMERGE merges the coordinates of the reference file stations (obtained in PID 004) and the project–specific coordinates (computed for the epoch to be processed in PID 003). The merged coordinate file ({V_A}yyssss.CRD, e.g., APRyyssss.CRD in the STA–directory) contains all reference frame stations completed by the coordinates for all additional sites of the project–specific coordinate file. The flags in the resulting coordinate file indicate where the coordinates are coming from (IGB08 for reference frame sites and PPP for all other entries — please note that only the first letter of the flag is used to identify the coordinates in the related fields of the processing programs).

PID 011 RNX_COP: The algorithm to select RINEX observation files, described in Section 23.4.2.3, is implemented in this script. It copies the selected files from the datapool into the campaign area and applies the Hatanaka decompression program, if necessary.

The script checks whether an hourly processing scheme is applied. In that case the hourly observation files need to be merged using the next group of scripts. This is not needed for a daily processing scheme where the execution may continue with PID 099 .

If no observation file was found for the corresponding session, the processing ends with PID 999 . This allows to run through all sessions regardless of whether data for all sessions are available or not.

PID 021 OBSMRGAP: This script and the following one form a unit. This first script prepares the parallelization, the actual processing is done in the next PID. The task of this step is to concatenate hourly RINEX files in case of an hourly processing scheme and/or to cut the observations according to the current session definition.

The script deletes files that may be present from previous runs and prepares a list of RINEX files for parallel processing.

PID 022 OBSMRG_P: The processing of the RINEX observation files is done by the program CCRINEXO. The observations that belong to the current session, or to the range of hourly sessions (defined by the BPE variable V_HOURLY), are copied from the input RINEX files in the ORX directory of the campaign into the output files located in the campaign's RAW directory.

PID 031 ION_MRG: In case of an hourly processing scheme the ionosphere maps of several days are simply concatenated by the initial copy script R2S_COP. If the ionosphere maps contain duplicate epochs, they are cleaned by this script.

PID 099 DUMMY: Does nothing, closes this section of scripts.

Further reading:

Section 3.1: "Overview of the Directory Structure", page 57
Section 18.3: "ANTEX Converter ATX2PCV", page 411
Section 10.7.9: "Propagating Coordinates to Specific Epochs", page 272
Section 10.7.7: "Merging Coordinate and Velocity Files", page 271
Section 22.5.2: "Parallel Processing", page 490
Section 22.5.3: "Loops and Jumps", page 491

Prepare Pole and Orbit Information

The next section of the example RNX2SNX.PCF performs some standard preparatory steps. In particular, orbit and Earth orientation files are converted from foreign to Bernese formats. It is very important that you always use orbits together with the corresponding pole information to avoid inconsistencies.

```
#
#
# Prepare the pole and orbit information
# --------------------------------------
101 POLUPDH  R2S_GEN           ANY      1 001
111 ORBMRGH  R2S_GEN           ANY      1 001
112 PRETAB   R2S_GEN           ANY      1 101 111
113 ORBGENH  R2S_GEN           ANY      1 112
199 DUMMY    NO_OPT            ANY      1 113
```

PID 101 POLUPDH: The IERS formatted pole file (IEP) provided, e.g., by the CODE or the IGS, is converted to Bernese format (ERP). This newly created pole file is used throughout all further processing steps.

The script checks the content of the input pole file and tries to find out the correct setting for option "Use ERP rates". Because of the diversity of the IERS/IGS formatted pole files, it might happen that the algorithm fails. In this case you have to define the option manually in the user script by replacing XYZ by the source of your products (value of BPE server variable V_B):

```
# Repeat the program if necessary
# ------------------------------
  if ( $v_b eq "XYZ" ) { $doRate = 1 }  # if products from "XYZ" are used
                                        # apply the polar rates.
  if ( $doRate != $isRate ) {
    $bpe->putKey("$ENV{U}/PAN/POLUPD.INP","USERATE","$doRate");

    my $PGMNAM = "POLUPD";
    $bpe->RUN_PGMS($PGMNAM);
  }
```

PID 111 ORBMRGH: Concatenates precise orbit files from several days if needed for an hourly processing scheme.

If products from the IGS shall be used (indicated by the BPE variable V_B == "IGS") and GLONASS shall be included (BPE variable V_SATSYS == "GPS/GLO") in the processing, the GPS and GLONASS orbit files are merged (IGSwwwwd and IGLwwwwd). Note that this is not needed if you use products from CODE because these precise orbit files contain the satellite positions for both systems.

PID 112 PRETAB: The precise orbit file (PRE), e.g., from CODE, is converted to a Bernese tabular orbit file (TAB).

In addition, the satellite clock corrections are extracted from the precise file and stored in a Bernese satellite clock file (using a polynomial representation).

PID 113 ORBGENH: Starting from the tabular orbit file, a standard orbit file is created by means of numerically integrating the equations of motion. The orbits are parameterized by six osculating elements and nine dynamical parameters (usually associated with radiation pressure). A summary file informs about the quality of the orbit fit. It is contained in the processing summary (R2Syyssss.PRC).

PID 199 DUMMY: The script does nothing but closing the section of orbit–related scripts in the PCF .

The EOP and orbit files created in these three steps are used throughout all further processing steps and will not be changed anymore.

Further reading:

Section 4.3: “Precise Orbit Files”, page 90
Section 4.4: “IGS and IERS Pole Files”, page 91
Chapter 5: “Preparation of Earth Orientation, GNSS Orbit, and Satellite Clock Information”, page 107

Convert and Synchronize Observation Data

This part of the PCF converts RINEX files to Bernese format, synchronizes the receiver clocks to GPS time, and produces an easy–to–read overview of available data. Together with the previous section (orbit and pole preparation), it will be part of most of your PCFs.

```
#
# Preprocess , convert , and synchronize observation data
# -------------------------------------------------------
201 RNXSMTAP R2S_GEN           ANY      1 099
202 RNXSMT_H R2S_GEN           ANY      1 201
211 RNXGRA   R2S_GEN           ANY      1 202
221 RXOBV3AP R2S_GEN           ANY      1 211
222 RXOBV3_H R2S_GEN           ANY      1 221
231 CODSPPAP R2S_GEN           ANY      1 099 199 222
232 CODSPP_P R2S_GEN           ANY      1 231
233 CODXTR   R2S_GEN           ANY      1 232
299 DUMMY    NO_OPT            ANY      1 233
```

PID 201 RNXSMTAP: The program RNXSMT is called in the next script to clean and smooth the code measurements. The program may be executed in parallel for several groups of stations. It is the task of this script to create these groups according to the BPE server variable V_CLU.

PID 202 RNXSMT_H: The script runs the program RNXSMT on a group of RINEX files. In case of RINEX (version 3) files, it will select the observation types according to the priority list given in `V_OBSINF`. The code observations are cleaned and the smoothed code is provided whereas the phase measurements are only copied from the original file to the new RINEX file (extension `SMT`). The phase measurements will be preprocessed in a later step of the BPE.

PID 211 RNXGRA: A summary of all available observation data is created, giving a complete overview of observed satellites, involved stations, and their performance. This file appears in the processing summary and may help to identify possible data tracking problems of observing sites.

Note for processing your own data: Stations showing large data gaps are detected by RNXGRA and listed in a deletion file (`GRAyyssss.DEL` in directory `OUT`), used by the script to delete the corresponding RINEX observation files from the `RAW` directory. The criteria defining an improper station may be changed in panel "RNXGRA 3: Options for RINEX File Selection" (to be displayed after checking the box for "Get a list of RINEX files"). All deleted stations are reported in the processing summary.

PID 221 RXOBV3AP: This script does not start any programs. It deletes existing Bernese zero–difference observation files for the current session and prepares the parallel execution of the next script.

PID 222 RXOBV3_P: Runs RXOBV3 in parallel based on the preparations from the previous script add(PID 201 and PID 202) to convert the smoothed RINEX files to Bernese observation files. The program compares the data records in the RINEX header with the entries in the station information file. Any detected header inconsistency is reported in the processing summary. Using the default settings, the BPE stops with an error if the station name is not in the abbreviation file or if an antenna type is not listed in the phase center offsets file (default: `${X}/GEN/{V_PCVINF}.{V_PCV}`, e.g., `${X}/GEN/PCV.I08`).

Note for processing your own data: In case of inconsistencies no Bernese observation files are written because the options for "ACTIONS IN CASE OF INCONSISTENCIES" are set to SKIP. This is the robust setting to generate a result without manual interactions. However, stations may be missing in the solution. Alternatively, you may set the options to ERROR causing the BPE to stop in case of inconsistencies.

PID 231 CODSPPAP: Prepares the parallelization of CODSPP. Please note that this script waits for altogether three scripts (PIDs 099, 199, 222) to be finished before starting. This ensures the existence of all necessary input files, namely the a priori coordinates (PID 006; PID 099 closes this section), the pole and orbit information (PID 113; PID 199 ends this section), and observation files in Bernese format (PID 222).

PID 232 CODSPP_P: CODSPP performs a code–based zero–difference point positioning, mainly to synchronize the receiver clocks to GPS time on a sub–microsecond level. A basic outlier detection is included as well. Epoch–wise clock corrections are estimated because receiver clocks may not be stable enough to be modeled as polynomials (at least not at the required accuracy level). Depending on the value of `V_CLU`, the program CODSPP runs in parallel.

Note for processing your own data: CODSPP stops if a priori station coordinates are missing.

PID 233 CODXTR: The program CODXTR creates a summary from the output files written in the CODSPP step. The file is listed in the processing summary. A high

RMS value and/or a lot of outliers indicate a site with bad code tracking performance. *Note for processing your own data:* Stations showing a large RMS are detected and listed in a deletion file (SPPyyssss.DEL in directory OUT). The RMS threshold can be adjusted in panel "CODXTR 2: Options". Both code and phase observation files will be added to that list. The script removes all observation files that are listed in the deletion file.

PID 299 DUMMY: The script does nothing, but it is beneficial to divide the PCF into sections.

RXOBV3 runs in parallel only if option "Action if station not in abbreviation list" in panel "RXOBV3 2: Input Options 1" is set to ERROR. Otherwise, all stations are processed in one RXOBV3 run. This prevents problems when different RXOBV3 runs try to update the station abbreviation table at the same time.

Further reading:

Section 4.2: "RINEX Observation Files", page 75
Section 4.2.3: "Import to Bernese", page 79
Section 6.3: "Receiver Clock Synchronization and Preprocessing of Code Observations", page 135
Section 22.5.5: "Script Parameters and BPE Variables", page 493
Section 22.5.2: "Parallel Processing", page 490

Form Baselines, Preprocess and Screen Phase Data, Compute a Solution Without Resolved Ambiguities

This PCF section constitutes the preprocessing step and is an essential part of the processing. Single–difference files are created, cycle slips are detected and repaired, and unreasonable observations are removed. A diligent preprocessing is crucial for GNSS processing and severely influences the quality of obtainable results.

At the end of the section, a first network solution without resolved ambiguities is created.

```
#
# Form baselines and pre-process phase data (incl. residual screening)
# --------------------------------------------------------------------
301 INIT_BSL R2S_GEN           ANY      1 299
302 SNGDIF   R2S_GEN           ANY      1 301
303 SNGDIF   R2S_GE2           ANY      1 302
311 MAUPRPAP R2S_GEN           ANY      1 302
312 MAUPRP_P R2S_GEN           ANY      1 311
313 MPRXTR   R2S_GEN           ANY      1 312
321 GPSEDTAP R2S_EDT           ANY      1 313
322 GPSEDT_P R2S_EDT           ANY      1 321
323 GPSXTR   R2S_EDT           ANY      1 322
331 RES_SUM  R2S_GEN           ANY      1 322
341 ADDNEQ2  R2S_GEN           ANY      1 331
342 GPSXTR   R2S_GEN           ANY      1 341
399 DUMMY    NO_OPT            ANY      1 303 323 342
```

PID 301 INIT_BSL: The script removes all existing code– and phase–baseline observation files (pattern: ????ssss.[CSH|CSO|PSH|PSO] in the OBS–directory) and cluster files (pattern: yyssss??.CLB in the STA directory). If you are going to repeat the processing of a certain session, it is essential that all observation files of any previous run are cleaned up.

PID 302 SNGDIF: Program SNGDIF selects a complete set of independent baselines and creates phase single–difference observation files. The adopted strategy for the selection process is OBS–MAX which can be considered as standard for almost all applications.

The list of created baselines is stored in the file `BSLyyssss.BSL` located in the `STA` directory of your campaign.

PID 303 SNGDIF: The list of baselines forming the optimal network configuration for the phase measurements is taken from the file `BSLyyssss.BSL` (created by SNGDIF in PID 302) and the corresponding single–difference files are also created for the code observations. This is essential for the ambiguity resolution based on the Melbourne–Wübbena linear combination.

PID 311 MAUPRPAP: Prepares the parallel run for the program MAUPRP.

PID 312 MAUPRP_P: MAUPRP preprocesses the phase single–difference files. Cycle slips are detected and corrected. If the size of a cycle slip cannot be reliably determined, a new ambiguity is set up. In addition, unpaired observations (i.e., only L_1 or L_2 at an epoch) and observations gathered at very low elevation angles are flagged as unusable.

PID 313 MPRXTR: MPRXTR creates a summary of the previous preprocessing step. This file is included in the processing summary.

PID 321 GPSEDTAP: Prepares the parallel run of the `GPSEDT_P` script.
In contrast to the other parallelization scripts it uses `V_CLUEDT` to get the number of files per group if this BPE server variable is defined (see Section 23.4.2.5 for more details).

PID 322 GPSEDT_P: This script performs a double–difference phase residual screening. Altogether four Bernese programs are started in sequence:

1. GPSEST to create residual files,
2. RESRMS to screen the residual files for outliers,
3. SATMRK to mark identified outliers, and
4. GPSEST to create final (clean) residual files and store normal equation files.

In both GPSEST runs all coordinates are loosely constrained to their a priori values. If an elevation–dependent observation weighting scheme is applied, normalized residuals should be stored. RESRMS checks the residual files for outliers based on the options in panel "RESRMS 2: Options". Detected outliers are listed in an edit information file (`EDTssssxxx.EDT`). SATMRK accordingly marks the corresponding observations.

PID 323 GPSXTR: Extracts the main information from the GPSEST program outputs of the final solution after cleaning the residuals for quality control.

PID 331 RES_SUM: Checks the screening results from the previous step and rejects data of misbehaving stations if necessary. Two programs are used:

1. RESRMS to create summaries from the first (unscreened) and final residual files.
2. RESCHK to create residual screening statistics and to detect bad stations based on their overall performance.

The summaries created by RESRMS and RESCHK are included in the processing summary. Problematic stations or satellites are indicated by large residuals and/or a very high percentage of deleted data.

PID 341 ADDNEQ2: A network solution with real–valued ambiguities is computed based on the normal equations stored in the GPSEST run after the residual screening (PID 322). Coordinates and troposphere estimates are saved for further use in the ambiguity resolution step.

PID 342 GPSXTR: Creates a short overview of the float solution. It is included in the processing summary (solution name P1_yyssss).

PID 399 DUMMY: The script does nothing. It finishes this section of the PCF.

If the program RESCHK detects a misbehaving station (depending on the options in panel "RESCHK 2.1: Detection of Bad Stations") the corresponding observation file is listed in a deletion file (EDTyyssss.DEL). The RES_SUM script then deletes the rejected file and jumps back to PID 301 (baseline creation) to create a new network of baselines without the deficient station and to repeat the preprocessing. This screening loop (PIDs 301–331) is continued until all stations are accepted. As single–difference files are screened, other (actually good) stations may be afflicted with errors propagated from the misbehaving station. Therefore, the program RESCHK is only allowed to delete one (the worst) station per iteration step to prevent these stations from being dragged along and being deleted, too.
Further reading:

Section 6.4: "Forming Baselines", page 140
Section 6.5: "Preprocessing Phase Observations", page 144
Chapter 7: "Parameter Estimation", page 169
Section 7.4.4: "Real and Normalized Residuals", page 174
Section 6.6: "Screening of Post–Fit Residuals", page 161
Section 6.7: "Marking of Observations", page 167
Section 6.6.3: "Detect Misbehaving Stations and Satellites", page 165
Section 7.9.2: "The Output Extraction Program GPSXTR", page 189
Chapter 9: "Combination of Solutions", page 209
Section 22.5.2: "Parallel Processing", page 490
Section 22.5.3: "Loops and Jumps", page 491

Resolve Phase Ambiguities

The next processing steps are dedicated to ambiguity resolution. After computing a solution with real–valued ambiguities at the end of the previous step, the strategies for resolving ambiguities to their integer numbers are applied to the GPS and/or GLONASS observations (according to the value of the BPE server variable V_GNSSAR) according to Table 8.1 .

```
#
# Resolve phase ambiguities
# ------------------------
401 SATMRK   R2S_GEN           ANY       1 399
411 GNSAMBAP R2S_AMB           ANY       1 401
412 GNSAMB_P R2S_AMB           ANY       1 411
421 GNSL53AP R2S_L53           ANY       1 412
422 GNSL53_P R2S_L53           ANY       1 421
431 GNSQIFAP R2S_QIF           ANY       1 422
432 GNSQIF_P R2S_QIF           ANY       1 431
441 GNSL12AP R2S_L12           ANY       1 432
442 GNSL12_P R2S_L12           ANY       1 441
443 AMBXTR   R2S_AMB           ANY       1 442
499 DUMMY    NO_OPT            ANY       1 443
```

PID 401 SATMRK: The program SATMRK at the beginning of this section resets all ambiguities indicated as resolved in the Bernese phase observation files.

PID 411 GNSAMBAP: The first strategy to be applied is the ambiguity resolution based on the Melbourne–Wübbena linear combination for long baselines. The baselines for this ambiguity resolution step are selected by using the program BASLST: the option "Maximum baseline length" is set to the BPE server variable V_BL_AMB and "Minimum baseline length" to V_BL_L53,[2] because shorter baselines are handled by other scripts of this series. The option "Baselines with observations from" is adjusted according to the BPE server variable V_GNSSAR.

The ambiguity resolution is performed for all baselines selected by the program BASLST baseline–by–baseline in a parallel mode by the following script.

Note when processing your own data: There are some old receivers used mainly in the 1990s that provide code measurements of limited quality. Because the Melburne–Wübbena linear combination is based on both code and phase data the success rate of any ambiguity resolution based on this data is limited. These receiver types may be indicated in the option "Receiver name pattern to be excluded" in panel "BASLST 3: Options".

PID 412 GNSAMB_P: This script executes several programs for three different tasks:

1. Screen the code measurements to remove large residuals
   a) GPSEST stores residuals of a code–only solution where the coordinate and troposphere estimates from the previous solution with real–valued ambiguities (script ADDNEQ2 in PID 341) are introduced without further improvement. The program output file has the name of the baseline extended by _C.
   b) RESRMS checks the residual file for large outliers, and
   c) SATMRK flags the outliers in the observation files.
2. Performs the resolution of the wide–lane ambiguities based on the Melbourne–Wübbena linear combination (strategy SIGMA in "Resolution strategy"). All necessary changes in the program input file for GPSEST are done within the script. The resolved wide–lane ambiguities are stored in the related observation files. The GPSEST program output consists of the name of the baseline extended by _W.
3. The resolved wide–lane ambiguities are introduced to solve also for the narrow–lane ambiguities in a subsequent GPSEST run. In this step the station coordinates of the second station of the baseline are estimated with respect to the first one; the program output starts again with the name of the baseline file and is extended by _N.

Independent from the value of the BPE server variable V_GNSSAR, this step is only performed on the GPS measurements because Version 5.2 of the *Bernese GNSS Software* does not support ambiguity resolution based on the Melbourne–Wübbena linear combination for GLONASS.

PID 421 GNSL53AP: This script selects all baselines where the phase–based wide–lane ambiguity resolution strategy is to be applied. Again, the program BASLST is used

[2] In fact 90% of this limit is used to make sure that no baseline gets lost for the ambiguity resolution due to rounding problems. A baseline that is close to the limit participates in two ambiguity resolution schemes.

where the range of length for the selected baselines is defined by the V_BL_L53 for option "Maximum baseline length" and V_BL_L12 for option "Minimum baseline length". The ambiguity resolution itself is performed by the next script in a parallel mode, baseline–by–baseline.

PID 422 GNSL53_P: This script carries out the phase–based wide–line ambiguity resolution strategy for all baselines selected in the previous script. Three programs are involved:

1. SATMRK to reset resolved ambiguities if a baseline close to the maximum length (given by the BPE server variable V_BL_L53) has already been selected for the previous strategy.
2. GPSEST to resolve the wide–lane ambiguities using SIGMA as "Resolution strategy". The resolved wide–line ambiguities are stored in the corresponding phase baseline observation files. Coordinates for the second station of the baselines are estimated whereas the coordinates of the first one are fixed. The troposphere parameters from the network solution without resolved ambiguities (PID 341, script ADDNEQ2) are introduced as known. The program output of this GPSEST is named with the filename of the baseline extended by _5.
3. GPSEST to resolve the narrow–lane ambiguities where the previously resolved wide–lane ambiguities are introduced. The options for the program GPSEST are adapted with respect to the first run automatically within the script. The filename of the resulting program output is constructed from the baseline extended by _3.

This ambiguity resolution strategy considers the BPE server variable V_GNSSAR and resolves also ambiguities for GLONASS if requested.

PID 431 GNSQIFAP: Prepares the parallel execution of the ambiguity resolution step based on the QIF strategy. It is applied to all baselines shorter than a certain value given by the BPE server variable V_BL_QIF and longer than a certain value specified by V_BL_L12. These values are introduced into the corresponding input fields for the program BASLST to select the related baselines. The ambiguity resolution itself is performed in the subsequent script where the individual baselines are processed in a parallel mode.

PID 432 GNSQIF_P: This ambiguity resolution step only needs one GPSEST run for each baseline to be processed. Troposphere estimates from the float solution (script ADDNEQ2, PID 341) are introduced and fixed whereas the coordinates for the second station are computed with respect to the coordinates of the first station. The "Frequency/linear combination" is set to L1&L2 and stochastic ionosphere parameters are estimated to absorb the impact of the ionosphere. The ambiguity resolution strategy QIF is selected and the solved integer ambiguities are stored in the observation header files. The name of the baselines extended by _Q defines the name of the program output for GPSEST.
The ambiguities resolved in the previous steps are introduced, i.e., the QIF strategy is only applied to the remaining ambiguities. In that way the success rate for the ambiguity resolution can by increased by typically 5 to 10%. If requested by the BPE server variable V_GNSSAR, also ambiguities for GLONASS are resolved in this step for baselines longer than V_BL_L53.

PID 441 GNSL12AP: For the shortest baselines (shorter than defined by the BPE server variable `V_BL_L12`) the ambiguities shall be resolved directly for L_1 and L_2 frequencies. These short baselines are again selected using the program BASLST and processed in the subsequent script baseline–wise in parallel mode.

PID 442 GNSL12_P: This script calls two programs:

1. SATMRK to reset resolved ambiguities if a baseline close to the maximum length given by the BPE server variable `V_BL_L12` has already been selected for the previous strategies.
2. GPSEST to perform the ambiguity resolution itself, where the "Frequency/linear combination" is set to L1&L2 but the SIGMA strategy is selected for "Resolution strategy" and no stochastic ionosphere parameters are estimated anymore. Whereas the coordinates for the second station of the baseline are estimated, the coordinates of the first one are fixed. The troposphere estimates from the network solution before the ambiguity resolution (script `ADDNEQ2` in PID 341) are introduced as known. The strategy is applied to GPS and GLONASS ambiguities according to the BPE server variable `V_GNSSAR`. The program output for GPSEST is stored in a file consisting of the baseline name extended by `_1`.

PID 443 AMBXTR: Creates summaries of each of the previous ambiguity resolution steps, listing, e.g., the percentage of successfully resolved ambiguities. This statistics will be included in the processing summary.

PID 499 DUMMY: This script does nothing except for closing the section of scripts related to the ambiguity resolution.

Please note that the ambiguity resolution (GPSEST) runs baseline by baseline regardless of the value of the variable `V_CLU`.

Further reading:

Chapter 7: "Parameter Estimation", page 169
Chapter 8: "Initial Phase Ambiguities and Ambiguity Resolution", page 191
Section 22.12.1.5: "Selection of Baselines for Processing", page 527
Section 13.5: "Stochastic Ionosphere Modeling Technique", page 331
Section 22.5.2: "Parallel Processing", page 490

Compute Ambiguity–Fixed Network Solution, Create Final `NEQ/SNX/TRO` Files

The observation files are cleaned and most of the ambiguities are resolved to their integer values. Now we are ready to compute the final ambiguity–fixed solution. The results are stored in Bernese and troposphere SINEX format. A special feature of this PCF part is the automatic verification of reference frame sites.

```
#
# Compute ambiguity-fixed network solution, create final NEQ/SNX/TRO files
# ----------------------------------------------------------------------
501 GPSCLUAP R2S_FIN           ANY      1 499
502 GPSCLU_P R2S_FIN           ANY      1 501
511 ADDNEQ2  R2S_FIN           ANY      1 502
512 GPSXTR   R2S_FIN           ANY      1 511
513 COMPARF  R2S_FIN           ANY      1 511
514 HELMCHK  R2S_FIN           ANY      1 511
521 ADDNEQ2  R2S_RED           ANY      1 514
522 GPSXTR   R2S_RED           ANY      1 521
599 DUMMY    NO_OPT            ANY      1 512 513 514 522
```

PID 501 GPSCLUAP: This is the final analysis of the observation files where all correlations between the different baselines are considered correctly. For that reason it is preferable to process all data together. Depending on the number of stations and on the available computing resources, you may have to split up your network into clusters for parallel processing as described in detail in Section 9.5.1.

With the BPE server variable V_CLUFIN you can decide whether all stations can be processed together in one cluster (if V_CLUFIN == 'N') or whether the baselines shall be processed in the clusters (if V_CLUFIN == 'Y') as defined by SNGDIF (PID 302) according to the file {V_CRDINF}.CLU (the clustering is stored in the files yyssss??.CLB located in the STA directory of your campaign).

The clustering may also be managed in an automated way (if V_CLUFIN == 'A'): no clustering is applied as long as less than 100 baselines are processed. If more baselines are listed in the yyssss??.CLB files the clustering is enabled. The limit of 100 baselines may even be adapted by adding the limit to the value of the variable V_CLUFIN (e.g., V_CLUFIN: 'A 50' to use 50 baselines as the limit for automatically enabling the clustering of the network).

According to the result of the above logic the subsequent script is started in one or more parallel running instances.

Note when processing your own data: If you have asked for redundant baselines in the script SNGDIF, PID 302 (enable checkbox for option "Add redundant baselines" in panel "SNGDIF 3.1: Options for Strategy OBS-MAX") a special cluster file yyssss00.CLB is created that must be treated separately in the subsequent GPSCLU_P script. In that case you will end up with at least two GPSCLU_P processes running in parallel — even if you do not have enabled the clustering.

PID 502 GPSCLU_P: All observations are processed considering the correct correlations within each cluster (with the sampling given in the BPE server variable V_SAMPL) and the resulting normal equation information is stored. The resolved ambiguities are introduced. No solution is computed (option "Stop program after NEQ saving" is enabled).

Note when processing your own data: If the special cluster file yyssss00.CLB (containing the redundant baselines, see above) is processed by GPSCLU_P the "Correlation strategy" is changed from CORRECT to BASELINE.

PID 511 ADDNEQ2: Based on the normal equations of the previous script the solution is computed. The datum definition is realized by three no–net–translation conditions imposed on a set of reference frame sites (given by V_REFINF, e.g., IGb08 reference coordinates). These stations are taken from the file REFyyssss.FIX (initially a copy of the input file {V_REFINF}.FIX).
From this ADDNEQ2 run a normal equation file (extension NQ0 in the SOL directory) is available that has collected all contributions from the sub–clusters. Furthermore, result files for the station coordinates (extension CRD in the STA directory) as well as troposphere estimates in Bernese and troposphere SINEX format (extensions TRP and TRO, respectively, both in the ATM directory), are generated. All result files as well as the program output refer to the solution ID {V_E}yyssss (resulting in the filename F1_yyssss assuming the default value for V_E).

PID 512 GPSXTR: Creates a short overview of the ambiguity–fixed network solution of ADDNEQ2 which is included in the processing summary (solution name F1_yyssss).

PID 513 COMPARF: Compares the estimated coordinate set with results from previous sessions. The number of sessions covered by this sliding window comparison can be adjusted by changing the PCF variables `V_PLUS` and `V_MINUS`. Among other information, the program provides an overview of the coordinate repeatability between subsequent sessions. The output and summary file created in this step are included in the processing summary. A bad repeatability indicates possible environmental, station, or processing problems. It may also be caused by geophysical phenomena (e.g., Earthquake).

PID 514 HELMCHK: The estimated coordinates of all involved reference sites are verified by means of a Helmert transformation including three translation parameters. The program HELMR1 is called. If discrepancies are detected — based on the options in "HELMR1 3: Outlier Rejection" — the solution is recomputed starting over from PID 511 with a reduced set of fiducial sites (maintained in the file `REFyyssss.FIX`). The output file of HELMR1 is included in the processing summary and may be helpful to identify problems concerning the reference sites and datum definition in general.

PID 521 ADDNEQ2: A normal equation file of reduced size is produced by pre–eliminating the troposphere parameters and horizontal gradients from the system. Thus, only coordinate parameters remain in the resulting normal equation file. The solution identifier for the result and program output files is `V_F` (default `R1_`).

PID 522 GPSXTR: A summary is extracted from the output of the reduced size solution. It is listed in the processing summary (solution name `R1_yyssss`).

PID 599 DUMMY: The script does nothing but closing this section of scripts.

The first GPSEST run is only used to create normal equations. No other result files are written. The final coordinate and troposphere results are computed with the program ADDNEQ2 due to its more sophisticated datum definition capabilities. The final SINEX file contains only coordinates and is therefore written during the size reduction step where the troposphere parameters are pre–eliminated. The fiducial site verification loop (PIDs 511–514) is repeated until all reference stations are accepted or until only one station remains. If you want to rely on only one station for datum definition, PID 514 may be skipped.

Further reading:

Chapter 7: "Parameter Estimation", page 169
Chapter 9: "Combination of Solutions", page 209
Chapter 10: "Station Coordinates and Velocities", page 241
Chapter 12: "Troposphere Modeling and Estimation", page 295
Section 10.7.5: "Coordinate Comparisons", page 270
Section 10.7.2: "Helmert Transformation", page 267
Section 22.12.3: "Rejecting Stations from the Definition of the Geodetic Datum", page 529
Section 10.2.3: "Reference Site Verification", page 249
Section 22.5.2: "Parallel Processing", page 490
Section 22.5.3: "Loops and Jumps", page 491

Create Summary File, Save Results, and Delete Files, end of BPE

In the last part of the PCF an analysis protocol is created, results are saved, and dispensable output files are removed. No Bernese programs are started here anymore. A similar sequence of scripts will most probably complete all your PCFs.

```
#
# Create summary file and delete files
# -----------------------------------
901 R2S_SUM  R2S_GEN           ANY      1 599
902 R2S_SAV  R2S_GEN           ANY      1 901
903 OBS_SAV  R2S_GEN           ANY      1 901
904 R2S_DEL  R2S_GEN           ANY      1 902 903
991 BPE_CLN  R2S_GEN           ANY      1 904
#
# End of BPE
# ----------
999 DUMMY    NO_OPT            ANY      1 991
```

PID 901 R2S_SUM: This script creates a processing summary giving a comprehensive overview of the most important results. The description of this summary is given in Section 23.4.2.7 .

PID 902 R2S_SAV: Used to save session result files (listed in Section 23.4.2.4) from the campaign into the SAVEDISK area.

PID 903 OBS_SAV: The observation files of this BPE are copied from the campaign into the SAVEDISK area, see description of BPE server variable V_SAVOBS.

PID 904 R2S_DEL: Files of previous RNX2SNX runs are deleted from the campaign directory in order not to accumulate files.

By default, the script removes all files related to 10 sessions prior to the current session. You may change this delay by adjusting the parameter PARAM1 for this script in the PCF (see also the header of this script).

PID 991 BPE_CLN: This step is responsible for cleaning up BPE–specific files (LOG and PRT files in the campaign's BPE directory) with a delay of 30 sessions.

PID 999 DUMMY: Does nothing.

The deletion script (PID 904) can be forced to delete all files not only of previous but also the current session (except of some important result and summary files) by setting PARAM1 to ALL.

The DUMMY–script may seem useless at a first glance. But if you have such a script running at a well–defined PID in all your PCFs the successful execution of a BPE can be tested in a generic way by checking the presence of the script's LOG– or PRT–file. In this way a script starting a BPE can react in case of errors (e.g., send an error mail).

Further reading:

Section 22.5.5: "Script Parameters and BPE Variables", page 493
Section 22.10: "BPE Output and Protocol Files", page 518
Section 22.11: "Error Handling", page 522

23.4.2.7. Description of the Protocol File, Quality Control

The processing is summarized in a so–called protocol file. It is compiled in the BPE user script R2S_SUM and exists first in the OUT directory of the campaign. If the

BPE server variable V_SAV == 'Y', the protocol is copied by the R2S_SAV script to ${S}/{V_RESULT}/yyyy/OUT/R2Syyssss.PRC and may be removed from the campaign area by the script R2S_DEL (if V_DEL == 'Y').

This protocol file starts with a section reporting all important input information for the BPE processing defined by the BPE server variables:

```
===============================================================================
RNX2SNX BPE PROCESSING SUMMARY FOR YEAR-SESSION 10-2070
===============================================================================


Summary file generated at 13-Dec-2012 07:44:38 by R2S_SUM

General files:
   Antenna phase center eccentricity file:   PCV.I08
   Satellite information file:               SATELLIT.I08
   Satellite problem file:                   SAT_2010.CRX
   Receiver characterization file:           RECEIVER.
   Orbit, ERP and clock products used from:  ${D}/COD

...
```

The protocol file is divided into parts related to the different sections of this BPE:

PART 0: RINEX INCONSISTENCIES

The equipment information in the RINEX header is compared with the expected equipment in the station information file when importing the observations from RINEX into the Bernese observation file format. Potential inconsistencies are reported in this section, the related action by the program is defined in the option "ACTIONS IN CASE OF INCONSISTENCIES" (e.g., for "Receiver/Antenna type" in panel "RXOBV3 5.1: Check Content of RINEX Header 1").

Further reading:

Section 4.2.3: "Import to Bernese", page 79
Section 4.2.3.4: "Verification of RINEX Header Information", page 83

PART 1: RINEX PSEUDO-GRAPHICS

The content of the RINEX observation files is summarized here (summary file output from program RNXGRA). Here you may detect stations with problematic data (e.g., reduced tracking performance of a station).

Further reading:

Section 4.2.5: "Utilities", page 87

PART 2: ORBIT GENERATION SUMMARY

The summary file of ORBGEN reports the RMS of the fit of the satellite positions given in the precise orbit file ${D}/{V_B}/{V_B}wwwwd.PRE with the orbit model of ORBGEN. It is expected that the RMS is on the few millimeters level (pseudo–stochastic pulses are set up every twelve hours to compensate for different orbit models in ORBGEN and the external program which was used to produce the orbits).

```
================================================================================
PART 2: ORBIT GENERATION SUMMARY
================================================================================

 RNX2SNX_102070: Generate GNSS standard orbit                    13-DEC-12 07:37
 TIME FROM DAY : 1 GPS WEEK: 1594
      TO   DAY : 2 GPS WEEK: 1594
 -------------------------------------------------------------------------------
 ORBIT REPEATABILITY FROM A 1-DAY FIT THROUGH DAILY ORBIT SOLUTIONS (MM)
 # ECLIPSING SATELLITES:   5 E /   0 M (  0 EM)
 ------------------------------------------------------------------------------------------ ...
 ECL  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  E.  E.  E.  ..  ..  ..  ..  ..  ..  ...
 DOY   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  ...
 ------------------------------------------------------------------------------------------ ...
 207   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1  ...
 ------------------------------------------------------------------------------------------ ...
 ALL   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1  ...
 ------------------------------------------------------------------------------------------ ...
```

If the RMS is larger, you should check the consistency of the Earth rotation files `${D}/{V_B}/{V_B}wwww7.ERP` (or the corresponding daily files `${D}/{V_B}/{V_B}wwwwd.ERP` if you are going to use them instead).

Since CODE has improved its solar radiation pressure model in its operational solution from January 2015 onwards, an update of the software (published as B_049) is needed to allow the fit of the precise orbit files with the expected quality. After updating the software you may choose the new model by selecting "dynamical orbit parameters" as "System D2X" in panel "ORBGEN 3.1: Options".

Further reading:

Section 4.3: "Precise Orbit Files", page 90
Section 4.4: "IGS and IERS Pole Files", page 91
Chapter 5: "Preparation of Earth Orientation, GNSS Orbit, and Satellite Clock Information", page 107

PART 3: SINGLE-POINT-POSITIONING SUMMARY

The output of the program CODXTR gives an overview of the performance of CODSPP synchronizing the receiver clocks to GPS system time using pseudorange measurements. It is expected that the RMS is smaller than 5.00 m. No additional exclusion periods should be reported in this part. If additional exclusion periods are reported, however, they may be caused by serious data problems of a specific station (please have a closer look to its results later in the protocol) or there are some problems/inconsistencies with the introduced satellite clock file (if all stations are listed for a specific interval).

```
================================================================================
PART 3: SINGLE-POINT-POSITIONING SUMMARY
================================================================================
...
  13 FILES, MAX. RMS:     0.47 M FOR STATION: JOZ2 12204M002
            MAX. BAD:    43.75 % FOR STATION: ONSA 10402M004


-----------------------------------------------------------------
>>> CPU/Real time for pgm "CODXTR": 0:00:00.004 / 0:00:00.004
>>> Program finished successfully
```

Further reading:

Section 6.3: "Receiver Clock Synchronization and Preprocessing of Code Observations", page 135

PART 4: PREPROCESS PHASE OBSERVATIONS

This section contains the summary of the MAUPRP program output generated by MPRXTR.

```
===============================================================================
PART 4: PREPROCESS PHASE OBSERVATIONS
===============================================================================

 SUMMARY OF THE MAUPRP OUTPUT FILE
 *********************************

 SESS FIL OK?  ST1  ST2 L(KM)   #OBS.    RMS    DX     DY     DZ    #SL   #DL   #MA  ...
 ----------------------------------------------------------------------------------- ...
 2070   1 OK   GANP JOZ2  344   43158      13    240     54    329    17   374    77  ...
 2070   2 OK   HERT ZIM2  685   42156      13   -278     -8   -296     5   560   115  ...
 ...
```

Pay attention to the RMS of the epoch–difference solution which should be smaller than 20 mm. If all baselines show large RMS values, a problem with the satellite orbits is likely the source. If only a limited number of baselines is affected the limit for observed minus computed in the input panel has to be revised or the constraints of the coordinate differences in the input panel do not correspond to the quality of the a priori coordinates. In the latter case, a repetition of MAUPRP with improved a priori coordinates may help.

Further reading:

Section 6.5: “Preprocessing Phase Observations”, page 144

PART 5: DATA PREPROCESSING/SCREENING SUMMARY

This section contains the output and summary files of the programs RESCHK and RESRMS. They provide some statistics on the residual screening (RMS before and after screening, number of removed measurements). The tables are useful to detect whether a certain problem is related to a baseline/station or to a satellite.

```
===============================================================================
PART 5: DATA PREPROCESSING/SCREENING SUMMARY
===============================================================================

...

-------------------------------------------------------------------------------
PRN | % Observations   Difference    | # Observations   Difference    | RMS
    | before   after   abs      rel | before     after  abs       rel | bef    aft
----+-------------------------------+--------------------------------+-----------
  1 |    0.22    0.22    0.00   0.31 |     182      182        0   0.00 |  1.5   1.4
  2 |    2.42    2.43    0.00   0.16 |    1970     1967       -3  -0.15 |  1.4   1.4
  3 |    1.95    1.95    0.00   0.06 |    1588     1584       -4  -0.25 |  1.3   1.3
  4 |    2.33    2.33   -0.00  -0.01 |    1894     1888       -6  -0.32 |  1.4   1.4
  5 |    2.33    2.33    0.00   0.10 |    1891     1887       -4  -0.21 |  1.4   1.4
  6 |    1.94    1.94    0.00   0.06 |    1574     1570       -4  -0.25 |  1.3   1.3
  7 |    2.16    2.16    0.00   0.14 |    1754     1751       -3  -0.17 |  1.3   1.2
...

121 |    2.02    2.02    0.00   0.19 |    1640     1638       -2  -0.12 |  1.4   1.3
122 |    2.07    2.07    0.00   0.13 |    1681     1678       -3  -0.18 |  1.4   1.3
123 |    1.82    1.81   -0.01  -0.50 |    1481     1469      -12  -0.81 |  1.5   1.4
124 |    1.65    1.65    0.00   0.01 |    1344     1340       -4  -0.30 |  1.6   1.4
----+-------------------------------+--------------------------------+-----------
TOT | 100.00  100.00    0.00   0.00 |   81287    81034     -253  -0.31 |  1.4   1.3
-------------------------------------------------------------------------------
```

The RMS after the screening should be in the order of 1.2 to 1.4 mm. It should also be verified that not more than 3% of the data are removed. Otherwise you have to find out whether there are some problems with your satellite orbits or whether one or several stations have introduced the problem (e.g., due to tracking problems of the receiver).

Further reading:

> Section 7.4.4: "Real and Normalized Residuals", page 174
> Section 6.6: "Screening of Post–Fit Residuals", page 161
> Section 6.7: "Marking of Observations", page 167

PART 6: AMBIGUITY RESOLUTION SUMMARY

This section provides a summary of the ambiguity resolution. You can expect the following benchmarks for the different strategies:

- Code–based wide–lane ambiguity resolution
  RMS before: < 0.5 mm / after: slightly larger; 95 to 100% resolved
- Code–based narrow–lane ambiguity resolution
  RMS: $1.0 - 1.5$ mm; about 90% resolved
- Phase–based wide–lane ambiguity resolution
  RMS: $1.0 - 1.5$ mm; 95 to 100% resolved
- Phase–based narrow–lane ambiguity resolution
  RMS: $1.0 - 1.5$ mm; 80% resolved
- Quasi–Ionosphere–Free (QIF) Ambiguity Resolution
  RMS: $1.0 - 1.5$ mm; 40% resolved
- Direct L_1/L_2 Ambiguity Resolution
  RMS: $1.0 - 1.5$ mm; close to 100% resolved

Remark: Phase-based wide- and narrow–lane and direct L_1/L_2 strategies start the ambiguity resolution after resetting already resolved ambiguities. On the other hand, the QIF–strategy resolves (at least for GPS) only the remaining real valued ambiguities.

Large RMS values before the ambiguity resolution in the QIF strategy may indicate that the real behavior of the ionosphere cannot be sufficiently represented by the a priori ionosphere model and absolute a priori sigma of the stochastic ionosphere parameters (the maximum value of which is given by the strategy itself). As a consequence, not all ambiguities are resolved.

Further reading:

> Chapter 8: "Initial Phase Ambiguities and Ambiguity Resolution", page 191

PART 7: GNSS COORDINATE/TROPOSPHERE SOLUTION STATISTICS

For each station, the output summary of the `GPSEST` run after the residual screening is provided. The solution `V_C` (default `P1_`) is the network solution prior to the ambiguity resolution. The second solution, `V_E` (default `F1_`), is the network solution introducing the resolved ambiguities. This should manifest itself in a smaller number of parameters. The RMS for the third solution, where the content of the NEQ `V_F` (default `R1_`) is reduced to station coordinates, should be identical with the `V_E` solution. The differences in the number of observations can be explained by differences in the number of necessary pseudo–observations according to the included parameters.

```
================================================================================
PART 7: GNSS COORDINATE/TROPOSPHERE SOLUTION STATISTICS
================================================================================


================================================================================
Preliminary (ambiguity-float) results
================================================================================

 P1_102070.OUT    Rms:  1.2 , # fil.:  12 , # obs.:   75472 , # par.:  1720
 (DOY: 207)


================================================================================
Final (ambiguity-fixed) results
================================================================================

Stations have been processed in clusters:

 F1_102070.OUT    Rms:  1.2 , # fil.:  12 , # obs.:   75628 , # par.:  1119
 (DOY: 207)


================================================================================
Size-reduced NEQ information
================================================================================

 R1_102070.OUT    Rms:  1.2 , # fil.:  12 , # obs.:   75290 , # par.:  1119
 (DOY: 207)
```

The RMS is expected to be 1 mm or slightly larger.

Further reading:

Chapter 7: “Parameter Estimation”, page 169
Chapter 10: “Station Coordinates and Velocities”, page 241
Section 7.9.2: “The Output Extraction Program GPSXTR”, page 189
Chapter 12: “Troposphere Modeling and Estimation”, page 295
Chapter 9: “Combination of Solutions”, page 209
Section 9.4.9.1: “Writing SINEX Files”, page 233

PART 8: VERIFICATION OF DATUM DEFINITION

The final coordinate solution (solution ID V_E) is compared with the reference frame coordinates. The datum definition in the final solution was realized by a minimum constraint solution for the translation parameters (for regional networks). For that reason, only translation parameters in the program HELMR1 have been set up, which should be estimated to zero. The residuals for the reference stations (not flagged with M in the last column) are expected below 1 cm. The values for the other stations correspond to the improvement of the current solution with respect to the a priori coordinates. The improvements should not exceed 10 cm (otherwise we recommend an update of the a priori coordinates for the particular station in the {V_CRDINF}.CRD file in the DATAPOOL area).

```
================================================================================
PART 8: VERIFICATION OF DATUM DEFINITION
================================================================================
...

 NUMBER OF PARAMETERS  :      3
 NUMBER OF COORDINATES :     24
 RMS OF TRANSFORMATION :     2.18 MM

...

 PARAMETERS:

 TRANSLATION IN  N      :           0.00     +-  0.77     MM
 TRANSLATION IN  E      :          -0.00     +-  0.77     MM
 TRANSLATION IN  U      :           0.00     +-  0.77     MM


 NUMBER OF ITERATIONS  :      1


 NO OUTLIER DETECTED
```

Outliers in the coordinate residuals (after considering the transformation parameters) may indicate weakly estimated coordinates (due to data problems, preprocessing issues, or incorrectly resolved ambiguities). If they are caused by permanent crustal deformations such as Earthquakes, the corresponding site should be removed from the list of reference sites. An equipment change may also introduce a deviation in the coordinate result series.

Stations with large residuals in the coordinate comparison may also be listed in the section REJECTED STATIONS that reports stations which have not been used for the datum definition.

Further reading:

Section 10.7.2: “Helmert Transformation”, page 267

PART 9: SLIDING 7-SESSION COMPARISON OF STATION COORDINATES

In this part, the network solution for the coordinates is compared to the coordinate solutions of the previous sessions using the program COMPAR called in the BPE user script PID 513 COMPARF.

This is a consistency check of the series of solutions that allows to detect events in the time series. The reasons are usually the same as mentioned above in PART 8: equipment changes, Earthquakes, or problems with the coordinate estimation.

This section does not appear if no coordinate files of previous sessions are available in the active campaign.

Further reading:

Section 10.7.5: “Coordinate Comparisons”, page 270

23.4.2.8. Velocity Estimation

The normal equation files generated in PID 521, which are reduced in size, can be used for multi–session combinations with ADDNEQ2. Because the files contain coordinates as the only explicit parameter type, even a multi–year combination to retrieve velocities is feasible. For this single program run, no example BPE is necessary. Simply combine the NEQs from the four example sessions using ADDNEQ2 following the instructions in Chapter 10.

Further reading:

> Section 10.3: "Coordinate and Velocity Estimation in Practice", page 249
> Chapter 9: "Combination of Solutions", page 209

23.4.3. Example for a Baseline–Wise Processing (BASTST.PCF)

23.4.3.1. Purpose and Description

The BPE example BASTST.PCF is designed to perform a quality check of single baselines. For each baseline a residual summary is created which gives you the observation data quality and the pre-processing performance.

It is assumed that all necessary input files are taken over from a previous RNX2SNX.PCF run (successfully completed to at least PID 312 and 313).

23.4.3.2. Input Files

The following input files are expected in the subdirectories of the campaign:

```
ORB/{V_ORBFIL}.STD
ORB/{V_ORBFIL}.ERP
STA/{V_APRFIL}.CRD
STA/{V_CRDINF}.ABB
```

Further input files are optional. They are used to switch options in the processing. These files must also be located the subdirectories of the campaign:

- ATM/{V_HOIFIL}.ION
  The ionosphere file is needed to correct for the higher order ionosphere terms.
- STA/{V_BLQINF}.BLQ
  STA/{V_ATLINF}.ATL
  Ocean and atmospheric tidal loading may be corrected during the baseline–wise processing if the corresponding corrections are provided in the files.

  These files must also contain corresponding CMC values for the translation of the orbit positions from the Earth-fixed system into the center–of–mass system for the orbit integration.

23.4.3.3. Observation Selection

The baseline observation files are expected in the OBS directory of the campaign. They must have been screened at least by MAUPRP before the BASTST.PCF example BPE is started.

If the baseline files contain resolved ambiguities, they are introduced. If you want to analyze the files without introducing the integer ambiguities you must change the corresponding setting in option "Introduce L1 and L2 integers" in panel "GPSEST 3.2: General Options 2" (script GPSEST_P).

23.4.3.4. Result Files

The result files from this BPE are not copied into the `SAVEDISK` area. They are only available in the campaign area.

The following files are generated:

- `OUT/BASyyssss.PRC`
  Main protocol file containing the summary of the processing steps (more details are provided in Section 23.4.3.7). Most of the summary and main output files of the BPE are collected in the `OUT` directory.

In addition, the program output and result files of the baseline–wise GPSEST processing are also available in the campaign:

```
OUT/{V_A}ssss????.OUT
OUT/{V_A}ssss????.RES
SOL/{V_A}ssss????.NQ0
ATM/{V_A}ssss????.TRP
STA/{V_A}ssss????.CRD
```

(???? stands for the baseline name)

Take care that the residual file contains REAL residuals (instead of NORMALIZED residuals as they are typically used for the data screening). Please inspect the option "Type of computed residuals" in panel "GPSEST 3.1: General Options 1" for the program GPSEST.

23.4.3.5. BPE Server Variables

Several variables have default values in the PCF (last section of the file). When starting a BPE manually they may be modified in the panel "RUNBPE 4: Server Variables Sent to Client".

Input Filenames

This BPE is designed to be run in an existing campaign with available a priori station coordinates and orbit files. These files are given in the following BPE server variables:

V_APRFIL (default: APR$YSS+0)
Name of the file with the a priori coordinates. It has to be specified as it is used in the program input panel.

V_ORBFIL (default: COD$YSS+0)
The name of the files containing the standard orbits and the ERP information must be given here.

Additional files are needed by the BPE for a successful processing. All these files are expected to be in the usual subdirectories of the campaign area. Typically, this is organized by the copy script of the BPE that has generated the baseline files.

V_CRDINF (default: EXAMPLE)
The basename of the files used as basic Bernese input files for this project. The file `{V_CRDINF}.ABB` is needed in this BPE.

V_BLQINF (default: EXAMPLE)
The basename of the file containing the ocean tidal loading corrections for the stations. The file must also contain the CMC related to ocean tidal loading. These corrections are needed if the orbit was generated using the CMC values.

If the variable is empty, no displacement corrections are applied to the station coordinates. It is assumed that the orbit was computed without CMC.

V_ATLINF (default: EXAMPLE)
The basename of the file containing the atmospheric tidal loading corrections for the stations. The file also must contain the CMC related to atmospheric tidal loading. These corrections are needed if the orbit was generated using the CMC values.

If the variable is empty, no displacement corrections are applied to the station coordinates. It is assumed that the orbit was computed without CMC.

V_HOIFIL (default: HOI$YSS+0)
The basename of the file containing the ionosphere model that is used to compute the HOI corrections. The file is expected in the `ATM` directory of the campaign.

The HOI corrections are disabled if this variable is empty.

Observation Selection

The following variables are responsible for the data selection:

V_SATSYS (default: GPS/GLO)
Selects the system(s) (usually GPS or GPS/GLO) to be used for the baseline–wise processing. The value of this variable is directly introduced in the option "Satellite system" in panel "GPSEST 3.1: General Options 1" (user script `GPSEST_P`).

NOTE: Whether all selected systems are really used for the processing depends also on the availability of satellite orbits as well as on the content of the baseline observation files. This option acts only as a filter to remove non–selected systems from the observation files.

V_SAMPL (default: 30)
Sampling interval (in seconds) for introducing the observations from baseline observation files into the processing (it should be a multiple of the data sampling in the observation files).

The value of this variable is directly introduced in the option "Sampling interval" in panel "GPSEST 3.1: General Options 1" (user script `GPSEST_P`).

General Model and Standard File Selection

BPE variables related to model selection:

V_PCV (default: I08)
Selection of the antenna phase center model identifier.

V_SATINF (default: SATELLIT)
Basename of the satellite information file. The resulting filename is `${X}/GEN/{V_SATINF}.{V_PCV}`.

V_PCVINF (default: PCV)
Basename of the antenna phase center correction file. The resulting filename is `${X}/GEN/{V_PCVINF}.{V_PCV}`.

V_SATCRX (default: SAT_$Y+0)
Name of the satellite problem file.

File and Product Naming

In this BPE there is only one BPE server variable to be used as solution identifier:

V_A (default: BAS)
Label of the program output and result files of the GPSEST process (`CRD`: coordinates, `TRP`: troposphere estimates, `NQ0`: normal equation files, and `RES`: residual files) and of the residual statistics.

23.4.3.6. Description of the PCF

Compute Baseline Solutions and Extract Relevant Information

This is the main part of the `BASTST.PCF` example BPE. Residual files, statistics, and summaries are created for each baseline.

```
#
# Compute baseline solutions and extract relevant information
# -----------------------------------------------------------
101 GPSESTAP BAS_GEN            ANY       1
102 GPSEST_P BAS_GEN            ANY       1 101
103 GPSXTR   BAS_GEN            ANY       1 102
111 RESRMS   BAS_GEN            ANY       1 102
112 RESCHK   BAS_GEN            ANY       1 111
```

PID 101 GPSESTAP: This script prepares the parallel GPSEST run. All baselines associated with the current session are processed individually.

As an essential step, the script deletes all result files for the current session that fit the file naming convention of this PCF .

PID 102 GPSEST_P: The program GPSEST creates a residual file for each baseline. The coordinates of the baseline's first station are fixed to their a priori values. Troposphere parameters for both stations and coordinates for the second station are estimated. The residuals are *not normalized* as no screening is intended. GPSEST runs in baseline–wise mode.

PID 103 GPSXTR: Extracts a short overview of the previous step. The resulting file is included in the processing summary.

PID 111 RESRMS: Creates a summary of the residuals. No screening is done. The resulting file is included in the processing summary.

PID 112 RESCHK: Another summary is created, containing various satellite–specific information such as the relative percentage of observations, the total number of observations, and the total RMS per satellite. This overview is included in the processing summary. Misbehaving satellites are indicated by a large amount of deleted data and a high RMS value.

Further reading:

Chapter 7: “Parameter Estimation”, page 169
Section 7.4.4: “Real and Normalized Residuals”, page 174
Section 6.6.2: “Generating Residual Statistic”, page 162
Section 6.6.3: “Detect Misbehaving Stations and Satellites”, page 165
Section 22.5.2: “Parallel Processing”, page 490

Create Processing Summary File

In the last part of the PCF an analysis protocol is created.

```
#
# Create processing summary file
# ------------------------------
901 BAS_SUM  BAS_GEN            ANY       1 103 112
991 BPE_CLN  PPP_GEN            ANY       1 901
```

PID 901 BAS_SUM: This script creates a processing summary giving a comprehensive overview of the most important results. It contains the summaries extracted in the previous steps (PIDs 103, 111, 112).

PID 991 BPE_CLN: This step is responsible for cleaning up BPE–specific files (LOG and PRT files in the campaign’s BPE directory) with a delay of 30 sessions.

Further reading:

Section 22.10: “BPE Output and Protocol Files”, page 518
Section 22.11: “Error Handling”, page 522

23.4.3.7. Description of the Protocol File, Quality Control

The processing is summarized in the so–called protocol file: BASyyssss.PRC. It is compiled in the BPE user script BAS_SUM and exists only in the OUT directory of the campaign.

This protocol file starts with a Section compiling the important filenames and the relevant BPE server variables. The result part is divided into two sections, each consisting of two and three parts, respectively:

PART 1.1: BASELINE-WISE PROCESSING SUMMARY

For each baseline the basic summary of the GPSEST run is provided. The number of observations mainly depends on the selected sampling and on the available/selected GNSS. The RMS is expected to be between 1.0 and 1.5 millimeter.

PART 1.2: COORDINATE SUMMARY

In this section the estimated coordinates for the second station of the baseline relative to the first one are given. The baseline vector solution is represented with respect to the a priori coordinates in the vertical and horizontal components.

PART 2.1: RESIDUAL SUMMARY PER SATELLITE

An overview of the number of observations and of the *real* residuals is given in the table. Notice that the residuals are not normalized using the observation weighting during the data processing.

PART 2.2: DETAILED RESIDUAL SUMMARY

For each baseline and satellite the RMS of the *real* residuals is provided in a comprehensive table.

PART 2.3: DETAILED RESIDUAL STATISTICS

The output of program RESRMS gives some more statistics on the residuals of each baseline.

23.4.4. Zero–Difference Network Processing Example (CLKDET.PCF)

23.4.4.1. Purpose and Description

The CLKDET.PCF BPE presents a processing scheme for the determination of station and satellite clock corrections. These corrections are computed based on GNSS (GPS and/or GLONASS) data for a regional network. In case of a combined GPS/GLONASS processing, also the relevant inter–system and inter–frequency biases (ISB, IFB) are computed. The procedure assumes dual–frequency data and is based on precise GNSS orbit information (e.g., from CODE). The result file is a clock RINEX file including both station and satellite clock corrections. A processing summary file CLKyyssss.PRC is written at the end of the BPE.

Some important features of this BPE are:

- Observation files with significant gaps or unexpectedly large residuals are automatically rejected.
- All clock solutions are computed with a zero–mean condition for all estimates of the receiver clock corrections (GPSEST in PIDs 301, 302, and 401). The reference clock for the final clock RINEX file is selected automatically (CCRNXC in PID 523).
- A project–specific coordinate file (V_CRDINF, e.g., EXAMPLE for EXAMPLE.CRD) is merged with a reference frame coordinate file (V_REFINF, e.g., IGB08 for IGB08_R.CRD) to use the latest coordinate reference frame for the analysis. The coordinates with a flag I in the merged project–specific coordinate file are constrained when clock corrections are estimated.

  If you have well–determined coordinates and troposphere corrections, you indicate the filenames in the variables V_FIXCRD and V_FIXTRP, respectively (the location of these files needs to be specified in variable V_FIXPTH). The script TIMEST_P adapts the datum definition in GPSEST so that these coordinates and troposphere estimates are fixed and only parameters for the remaining stations are estimated. The script CLKNEQ (PID 521) does the same for ADDNEQ2 and makes sure that the resulting coordinate and troposphere files contain records for all stations.

- The final solution consists of a combined processing of the phase and code measurements. The code observations of the individual stations are subject to a station–specific weighting. In the final solution the back–substitution step of the clock parameters is performed considering only phase data to prevent clock corrections based on more code than phase observations.
- Depending on the availability of precise high–rate satellite clock corrections (BPE variable V_HRCLK = 'N' or 'Y'), one of the following preprocessing schemes is selected to clean up the phase measurements:

  V_HRCLK = 'N' : using the program RNXSMT a consistency check is performed between the phase and code measurements based on the Melburne–Wübbena linear combination (requires dual frequency data).

  V_HRCLK = 'Y' : the program MAUPRP is called providing a sophisticated interpretation of epoch–difference residuals based on PPP (requires precise satellite clock corrections with the same sampling as the data has).

This PCF is prepared solely for a daily processing scheme. If you need the clock estimation procedure for an hourly processing scheme you need to adapt it in analogy to the PCFs for PPP or the double–difference example `RNX2SNX.PCF`.

The BPE provides the most important results in a comprehensive summary file `CLKyyssss.PRC` as described in Section 23.4.4.7.

23.4.4.2. Input Files

The input files are copied from the `DATAPOOL` area (`${D}`) into the campaign area in the BPE user script `CLK_COP`:

- **External products files** (mandatory)

  `${D}/{V_B}/{V_B}wwwwd.PRE`
  `${D}/{V_B}/{V_B}wwww7.ERP` or `${D}/{V_B}/{V_B}wwwwd.ERP`

  The variable `V_B` is defined below in the BPE server variable section and defines the product series to be considered for the zero–difference network processing (e.g., `COD` or `IGS`); `wwwwd` stands for the GPS week and the day of week to be processed.
  Note: orbit and ERPs should be consistent for the orbit integration.
- **External products files** (optional)

  `${D}/{V_B}/{V_B}wwwwd.CLK`

  A high–rate precise clock file becomes mandatory if the BPE server variable V_HRCLK = 'Y' indicates to use it for preprocessing the phase measurements. Otherwise such a file is not necessary.
  Note: the clock corrections must be fully consistent with the satellite orbit information.
- **Bernese formatted products files** (mandatory)

  `${D}/BSW52/P1C1yymm.DCB`

  The DCB corrections are necessary to unify the reference for the different receiver tracking techniques. This becomes important when receivers with different tracking technologies are processed in the same zero–difference network (`yymm` is the 2-digit year and the 2-digit month of the processing session).

```
${D}/BSW52/CODwwwwd.ION
```

The ionosphere file is needed to correct for the HOI and for the ambiguity resolution.

Both files can be downloaded from `http://www.aiub.unibe.ch/download/CODE/`.

- **Reference frame files** (mandatory)
  The reference frame in the project–specific coordinate file is introduced by adding/overwriting the station coordinates with the values given in the reference frame files. The reference frame coordinate and velocity files are named:

  ```
  ${D}/{V_REFDIR}/{V_REFINF}_R.CRD
  ${D}/{V_REFDIR}/{V_REFINF}_R.VEL
  ```

- **Project–specific station–related files** (mandatory)
  For each station to be processed, the following information has to be provided. Note that the information can (at least mostly) be provided by a PPP procedure (e.g., by applying the related BPE, see Section 23.4.1).

  To derive the a priori coordinates for the epoch of the processing, coordinates and velocities are needed:

  ```
  ${D}/{V_REFDIR}/{V_CRDINF}.CRD
  ${D}/{V_REFDIR}/{V_CRDINF}.VEL
  ```

  For each station a unique 4- and 2-character abbreviation is necessary to manage the observation file naming. This information is read from the abbreviation table:

  ```
  ${D}/{V_REFDIR}/{V_CRDINF}.ABB
  ```

  The station equipment history is expected in the file

  ```
  ${D}/{V_REFDIR}/{V_CRDINF}.STA
  ```

  to guarantee the correct handling of antenna types (antenna phase center corrections) and receiver types (biases related to the tracking technology).

- **RINEX header inconsistency acceptance file** (optional)
  The station information file (`{V_CRDINF}.STA`) is used to verify the header information of the RINEX observation files to be processed. If there are known inconsistencies they can be documented in this file. The program will accept RINEX files with such entries without further notice. The file with the accepted RINEX header inconsistencies is copied from

  ```
  ${D}/{V_REFDIR}/{V_CRXINF}.CRX
  ```

  as soon as the variable `V_CRXINF` contains a non–blank value.

- **Tidal loading corrections** (optional)
  Ocean and atmospheric tidal loading may be corrected during the zero–difference processing procedure if the corresponding corrections are provided in the files.

  Note: these files also have to contain the corresponding CMC values for the translation of the orbit positions from the Earth-fixed system into the center–of–mass system for the orbit integration.

  ```
  ${D}/{V_REFDIR}/{V_BLQINF}.BLQ
  ${D}/{V_REFDIR}/{V_ATLINF}.ATL
  ```

- **External coordinate/troposphere solution files** (optional)

```
{V_FIXPTH}/STA/{V_FIXCRD}.CRD
{V_FIXPTH}/ATM/{V_FIXTRP}.TRP
```

If an external coordinate and/or troposphere solution shall be introduced, the corresponding BPE server variables need to be defined. The corresponding files are copied by the script CLK_COP into the campaign area.

The variable V_FIXPTH has to contain the full path to allow an independent access to the SAVEDISK area or to processing campaigns.

23.4.4.3. Observation Selection

The observation files are expected in RINEX format in the directory ${D}/{V_RNXDIR}/ (format version 2) and/or in ${D}/{V_RX3DIR}/ (format version 3). The station/observation file selection is realized in the BPE user script RNX_COP.

Three possibilities to select stations for processing are implemented (considered in the following order):

1. A file with the name {V_OBSSEL}.CLU exists in the STA directory of the current campaign. The name of the cluster file is defined by the BPE server variable V_OBSSEL (see Section 23.4.4.5).

   All station names from this file are translated into the 4-character IDs using the abbreviation table {V_CRDINF}.ABB (also expected to be in the STA directory of the current campaign). These 4-character IDs are used to identify corresponding RINEX files from the source directory in the data base.

2. A file named {V_OBSSEL} resides directly in the top–level directory of your current campaign. It contains the RINEX abbreviations of the files to be processed. They have to be separated by blank characters or may start on a new line:

   ```
   GANP
   HERT
   JOZ2
   LAMA
   ZIM2 ZIMM
   ```

   (In this example, six stations are selected for the processing).

3. If the BPE server variable V_OBSSEL is empty all RINEX files of the specified session(s) are processed.

The selected RINEX files are copied and prepared for processing by the subroutine crz2rnx provided by the Perl module ${BPE}/bpe_util. The files are decompressed using the Hatanaka tool and/or any other usual compressing tool.

For this purpose, the Hatanaka decompression tool CRX2RNX (Unix, Mac OS) or CRX2RNX.exe (Windows) is expected to be installed on your system and its location must be included in the system $PATH or %PATH% variable — the search path for executable programs. The same is expected for the decompression tool gzip or gzip.exe to handle the potential generic compression of the files.

Depending on the content of the BPE server variables V_RNXDIR and V_RX3DIR, the corresponding RINEX files are considered in the related format versions 2 or 3 or even both. In the case of RINEX version 3 the conventional as well as the new long filenames are supported. If multiple versions of the same dataset are available, the selection is based on the given priority list defined in user script RNX_COP. The following preference list (in descending order) was defined:

1. long filename, RINEX version 3; generated by the receiver (indicated by data source flag R)
2. long filename, RINEX version 3; extracted from a data stream (indicated by data source flag S)
3. long filename, RINEX version 3 from an unknown source (indicated by data source flag U)
4. short filename, RINEX version 3 from an unknown source
5. RINEX version 2 (only short filenames available)

This priority list can of course be adapted according to your needs by modifying the value of the variable

```
  my @rx3Typ = ("R","r","S","s","U","u","short","rx2");
```

in the user script RNX_COP.

23.4.4.4. Result Files

The most important result files are copied by the BPE user script CLK_SAV into the ${S}/{V_RESULT} area (V_RESULT is defined below in the BPE server variables area) if the BPE server variable V_SAV == 'Y'. The files are located in yearly subdirectories (yyyy). These files are:

- ${S}/{V_RESULT}/yyyy/OUT/CLKyyssss.PRC
  Main protocol file containing the summary of the processing steps (more details are provided in Section 23.4.4.7).
- ${S}/{V_RESULT}/yyyy/ORB/{V_I}_yyssss.CLK
  ${S}/{V_RESULT}/yyyy/OUT/{V_I}_yyssss.CLK.gz
  Files with clock corrections from the consistent phase/code data analysis for GPS or GPS/GLONASS (according to the value of the BPE server variable V_SATSYS). In the ORB directory: GNSS satellite clock in Bernese format. In the OUT directory: satellite and receiver clock corrections in clock RINEX format.
- ${S}/{V_RESULT}/yyyy/STA/{V_I}_yyssss.CRD
  ${S}/{V_RESULT}/yyyy/ATM/{V_I}_yyssss.TRP
  Station coordinates and troposphere estimates that are consistent with the clock corrections. If records from an external solution have been introduced, they are completed with the solutions for coordinates and troposphere for all stations that were only available in the zero–difference processing.
- ${S}/{V_RESULT}/yyyy/STA/{V_C}_yyssss.SOS
  Station–specific observation weighting that was used for weighting the code measurements.
- ${S}/{V_RESULT}/yyyy/ORB/{V_F}_yyssss.DCB
  Inter–system/inter–frequency biases for a GPS/GLONASS processing together with P1−C1 DCB copied from the input file. In case of a GPS–only solution, this file contains only the DCBs.
- ${S}/{V_RESULT}/yyyy/SOL/{V_F}_yyssss.NQ0.gz
  Normal equation from which the above mentioned DCB–file was extracted. It only contains all the bias parameters and can be used for long–term bias solutions.

23.4.4.5. BPE Server Variables

Several variables have default values in the PCF (last section of the file). When starting a BPE manually they may be modified in the panel "RUNBPE 4: Server Variables Sent to Client".

Input and Result Filenames and Their Locations

It is assumed that all station–related files have the same name if they pertain to a certain project. They contain all station coordinates, equipment information, and so on. For the processing with the zero–difference example BPE they are specified via the following BPE server variables:

V_REFDIR (default: REF52)
Directory in the `DATAPOOL` area (`${D}`) where the basic Bernese files for the processing are located. The detailed list of files is given in Section 23.4.4.2 .

V_CRDINF (default: EXAMPLE)
The basename of the files used as basic Bernese input files for this zero–difference procedure. Copying these files is managed in the BPE user script `CLK_COP`.

V_CRXINF (default: empty)
Inconsistencies between the RINEX observation file headers and the station information file can be recorded in a file with "Accepted station information" inconsistencies in the program RXOBV3. If you need such a file for your processing, you can specify its name here. It will be copied from `${D}/REF52/{V_CRXINF}.CRX` into the campaign.

In general, it is recommended to use the same project–related filename for all input files of the processing. Nevertheless, a few files are related to specific options. Hence, separate variables are used for them:

V_BLQINF (default: EXAMPLE)
The basename of the file containing the ocean tidal loading corrections for the stations. The file also has to contain the CMC related to ocean tidal loading. These corrections are needed for the orbit generation.

If the variable is empty, no displacement corrections to the stations and no CMC for the orbit generation are applied. The file `${D}/REF52/{V_BLQINF}.BLQ` is copied into the campaign by the script `CLK_COP`.

V_ATLINF (default: EXAMPLE)
The basename of the file containing the atmospheric tidal loading corrections for the stations. The file must also contain the CMC related to atmospheric tidal loading. These corrections are needed for orbit generation.

If the variable is empty, no displacement corrections to the stations and no CMC for the orbit generation are applied. The file `${D}/REF52/{V_ATLINF}.ATL` is copied into the campaign by the script `CLK_COP`.

V_HOIFIL (default: HOI$YSS+0)
Specify the filename used for the ionosphere model within your campaign area that is used to compute the HOI corrections. The ionosphere model is copied from `${D}/BSW52/CODwwwwd.ION` into the `ATM`–directory of the campaign using the given filename.

If the variable is empty, the HOI corrections are disabled.

The following BPE server variables are related to the reference frame to be used for the processing:

V_REFINF (default: IGB08)
Reference frame files, used as master files when updating coordinate and velocity files. The corresponding files are expected as

```
${D}/{V_REFDIR}/{V_REFINF}_R.CRD
${D}/{V_REFDIR}/{V_REFINF}_R.VEL
```

The first sequence of the zero–difference processing example replaces coordinates for stations given in these reference frame files in the project–related coordinate file.

Another BPE server variable defines the location of the result files:

V_RESULT (default: CLKDET)
Directory in the `SAVEDISK` area (`${S}`) where the result files of the zero–difference processing example are collected in yearly subdirectories (saving of the result files is managed by the BPE user script `CLK_SAV`).

It is possible to introduce the results of an external solution to compute a clock solution consistent to those results:

V_FIXCRD (default: F1_$YSS+0)
Name of the coordinate solution file to be introduced.

V_FIXTRP (default: F1_$YSS+0)
Name of the troposphere solution file to be introduced.

V_FIXPTH (default: ${S}/RNX2SNX/$Y+0)
Path where the external coordinate/troposphere solution files are copied from.

The variable `V_FIXPTH` has to contain the full path to allow an independent access to the `SAVEDISK` area or to processing campaigns.

Observation Selection

There are three variables related to the selection of the observation files. The procedure is described in Section 23.4.4.3.

V_RNXDIR (default: RINEX)
Directory in the `${D}` (`DATAPOOL`) area where the RINEX files in the format version 2 are expected.

V_RX3DIR (default: empty)
Directory in the `${D}` (`DATAPOOL`) area containing RINEX files in format version 3 .

V_OBSSEL (default: empty)
This variable may contain a filename for a cluster file or a file with the RINEX station abbreviations.

The following variables are responsible for the data selection:

V_OBSINF (default: OBS.SEL)
This variable contains the filename of the receiver type related observation types priority list which is used when importing `RINEX3` data files, located in the the `${X}/GEN/` directory. The observation selection is done in RNXSMT. More details are given in Section 24.3.7 .

V_SATSYS (default: GPS/GLO)
Select the system(s) (usually GPS or GPS/GLO) to be used for the zero–difference processing example. The value of this variable is directly introduced in the option "Satellite system to be considered" in panel "RXOBV3 2: Input Options 1" (user script RXOBV3_P).

Note: whether all selected systems are really used for the processing depends also on the availability of satellite orbits as well as on the content of the RINEX observation files. This option acts only as a filter to remove non–selected systems from the observation files.

V_SAMPL (default: 300)
Sampling interval for the final solution. The value of this variable is directly introduced as the option "Sampling interval" in panel "GPSEST 3.1: General Options 1" (user script TIMEST_P). If no advanced pre-processing of the phase observations is enabled (if V_HRCLK /= 'Y') already the observation data imported into the Bernese observation files are reduced by applying this option also to "Sampling interval" in panel "RXOBV3 2: Input Options 1" (user script RXOBV3_H, PID 232).

General Model and Standard File Selection

In this sequence the BPE server variables related to the general processing files are listed:

V_PCV (default: I08)
Selection of the antenna phase center model identifier.

V_SATINF (default: SATELLIT)
Basename of the satellite information file. The resulting filename is ${X}/GEN/{V_SATINF}.{V_PCV}.

V_PCVINF (default: PCV)
Basename of the antenna phase center correction file. The resulting filename is ${X}/GEN/{V_PCVINF}.{V_PCV}.

V_SATCRX (default: SAT_$Y+0)
Name of the satellite problem file.

V_RECINF (default: RECEIVER.)
This variable contains the filename of the receiver characterization file located in the the ${X}/GEN/ directory. More details are given in Section 24.3.6 .

Another capability of the example BPE is to update the antenna phase center correction file based on an ANTEX file. This feature is managed by a BPE server variable as well:

V_MYATX (default: empty)
Filename (including extension) for an ANTEX file containing receiver and possibly satellite antenna phase center corrections. The file is used to update the Bernese phase center file in ${X}/GEN/{V_PCVINF}.{V_PCV}, e.g., in case of new antennas in the network.

It is expected that the ANTEX file is available in ${X}/GEN/ .

If you plan to use a file containing individually calibrated antennas you need to change the settings for "Set number of generic antennas to 999999" in panel "ATX2PCV 2: ANTEX Conversion" (Script 002 ATX2PCV):

- checkbox unmarked means “type–specific calibrations”
- checkbox marked means “antenna–specific calibrations”

See also Section 18.2.4: “Technicalities on the Receiver Antenna Handling”.

Selection of Preprocessing Mode

You may select how the phase measurements will be preprocessed. You have two options, depending on the availability of satellite clock corrections:

V_HRCLK (default: YES, if possible)

NO checks only the consistency between code and phase observations which limits the level of screening to the noise level of the code data.
YES uses a more involved algorithm but requires high–rate (30 seconds) precise satellite clock corrections. Due to the lack of GLONASS satellite clock corrections, `CLKDET.PCF` is provided with the value NO to allow the generation of a combined GPS+GLONASS solution.

Clustering for Parallel Processing

Many of the parallelization scripts (e.g., `RNXSMTAP`, `CODSPPAP`, `MAUPRPAP`) may run in two modes:

1. processing groups of stations together in a parallel script or
2. processing station by station in parallel.

For processing the stations in groups, a BPE server variable `V_CLU` is used: if it is defined the stations are grouped accordingly. If it is not defined a station–by–station processing is initialized.

On the one hand, the more CPUs (processing cores) are available the more stations can be processed in parallel to shorten the overall processing time. On the other hand, the number of parallel jobs is limited by the architecture of your computing system (including disk access times or the number of available CPUs) or by your system administrator.

In general you can plan one job per available CPU with a quick access to your data partition. If this number exceeds 15 to 20 stations/baselines per cluster it is worth to divide the jobs into additional clusters.

In any case, it is important to have approximately the same computing time for each of the clusters because the BPE only continues with the next script when the last of the parallel running jobs has finished.

V_CLU (default: 7)

For all parallel running scripts it is a question on how many files (stations/baselines) should be processed together and how many jobs are running in parallel.

The default value was defined to process the example (13 stations) on a dual–core machine. It results in two clusters that can be processed in parallel at the same time (two processing cores).

V_CLUCOD (default: 1)
Number of clusters for preprocessing the code measurements in regional clusters for residual screening. It is, of course, preferable to process all stations in one cluster. However, if the number of stations becomes larger this might not be efficient anymore. In that case a clustering becomes an option or even a necessity. Please note that in case of clustering a certain redundancy is requested to ensure a reliable computation of the residuals.

V_CLUPRP (default: 1)
The common processing of code and phase data is more CPU–time consuming than the code–only network solution. Not only more observations are processed, also more parameters (ambiguities) have to be solved for. For that reason a specific BPE server variable is available to define the number of clusters for these steps (`MKCLUSAP`/`TIMEST_P` with options `CLK_EDO` and `CLK_EDT`).

V_CLUSOL (default: 1)
The final solution (`MKCLUSAP`/`TIMEST_P` with options `CLK_RES`) may also be computed in clusters if too many stations are included.

V_MAXSOL (default: 40)
For the final solution a maximum number of stations per cluster may be specified. This option limits the total number of stations in the final solution in conjunction with the number of clusters (MAXSTA = {V_MAXSOL} * {V_CLUSOL}). It can, thus, happen that not all stations are included in the final solution. Most probably this does not harm the solution (satellite clocks may be computed from any set of well distributed ground stations). In case the solution if of limited quality, however, the variables `V_CLUSOL` and/or `V_MAXSOL` have to be increased.

File and Product Naming

The following group of BPE server variables are consistently used as solution identifiers and for file naming within the BPE processing:

V_B (default: COD)
Name of the external product for GNSS orbit, ERPs and potential satellite clock information (see description of variable `V_HRCLK`). It is expected that the products are available in the `DATAPOOL` area (`${D}`) in a specific subdirectory `V_B`; the filenames must also contain this label: `{V_B}wwwwd` (see also Section 23.4.4.2 on input files).

V_A (default: APR)
Starting station coordinates set for the processing.

V_C (default: CW)
Program output and result files of the processing steps related to the computation of the weights for the code observations.

V_E (default: CR)
Files related to the residual screening of the code observation files in a zero–difference network solution.

V_F (default: CB)
Program output and result files related to the computation of inter–system/inter–frequency biases.

V_G (default: T0)
If the BPE server variable `V_HRCLK` indicates that no precise high–rate satellite clocks are available, the phase observation files are only screened by RNXSMT. An additional step with a residual screening of a zero–difference network solution based on code and phase measurements in regional clusters (see BPE server variable `V_CLUPRP`) is conducted.

The output and result files of this step are named by the label in `V_G`.

V_H (default: TR)
Output and result files of a zero–difference network solution using code and phase observations processed together in regional clusters (see BPE server variable `V_CLUPRP`).

V_I (default: TP)
Final results and the related program output files of the clock solution based on code and phase measurements in global clusters (see BPE server variable `V_CLURES`).

The variables `V_C` to `V_I` contain only two characters. A third character is hard–wired in the panels and user scripts. The main output and result files use "_" as the third character, whereas for minor output and summary files other letters are used.

Management of the Dataflow

The last group of BPE server variables is related to the file handling in the campaign:

V_SAV (default: Y)
Save result file into the `${S}/{V_RESULT}` area if V_SAV == 'Y'.

V_SAVOBS (default: Y)
If V_SAVOBS == 'Y' the code and phase zero–difference observation files of the current session are copied into session–specific directories of the `SAVEDISK` area: `${S}/{V_RESULT}/yyyy/OBS/ssss`.

V_DEL (default: Y)
Delete all files of the currently processed session from the campaign if V_DEL == 'Y'. Usually, the deletion should be enabled to keep the campaign clean. For debugging purposes and to identify sources for errors in the processing, it may be useful to keep all files.

Skip Sessions in Case of Reprocessing

In case of a reprocessing it might be interesting to repeat only selected sessions for some reasons. A typical scenario could be that the BPE is executed once for all sessions. Possibly, some sessions will end with an error or with unsatisfying results, e.g., because of some stations with inconsistent equipment records between the station information file and the RINEX header. After fixing these problems, the reprocessing needs to be repeated only for a certain subset of sessions. Instead of starting the BPE manually for each of these sessions, it is more convenient to start the BPE again for the full interval of the reprocessing and let the availability of a specific file in the `SAVEDISK` area indicate whether a certain session shall be repeated or not.

V_SKIP (default: CLK$YSS+0.PRC; deactivated by CLK$YSS+0.PRC_)

The very first script SKIP_SES (PID 000) checks whether a file ${S}/{V_RESULT}/yyyy/OUT/{V_SKIP} exists, e.g., from a previous processing. The availability of the file indicates the successful processing of the session in a previous iteration of the reprocessing. Therefore, the BPE will directly jump to the last script (DUMMY; PID 999). If the indicated file does not exist, the BPE will execute all relevant scripts for the specific session.

We propose to use the protocol file in the SAVEDISK area as an indicator.

If a session shall be repeated even if it was executed up to the last script in a previous iteration, the indicator file (given in the V_SKIP variable) must be removed in the SAVEDISK area.

If all sessions shall be repeated regardless of the existence of the indicator file, the value of this variable can be left empty or at least be modified, e.g., from CLK$YSS+0.PRC to CLK$YSS+0.PRC_ .

23.4.4.6. Description of the PCF

Copy Required Files and Create A Priori Coordinate File

Before any program is started, all necessary files must be copied from the datapool into the respective campaign directories. In addition, the a priori coordinates must be prepared.

```
#
# Skip session (reprocessing)
# ---------------------------
000 SKIP_SES CLK_GEN           ANY      1
#
# Copy required files
# -------------------
001 CLK_COP  CLK_GEN           ANY      1 000
002 ATX2PCV  CLK_GEN           ANY      1 001
003 COOVEL   CLK_GEN           ANY      1 001
004 COOVEL   CLK_GE2           ANY      1 001
005 CRDMERGE CLK_GEN           ANY      1 003 004
011 RNX_COP  CLK_GEN           ANY      1 001
099 DUMMY    CLK_GEN           ANY      1 002 005 011
```

PID 000 SKIP_SES: This script checks whether the file ${S}/{V_RESULT}/yyyy/OUT/{V_SKIP} exists. The values for V_RESULT and V_SKIP are taken from the BPE server variables whereas yyyy is the 4-digit year of the current session. If the file exists, the BPE directly jumps to the last script. This mechanism may be used, e.g., to manage whether a specific session shall be repeated in the frame of a reprocessing or not.

PID 001 CLK_COP: This script copies all necessary files (apart from the RINEX files) into the respective campaign directories (the list of files is given in Section 23.4.4.2). If appropriate, file names will be changed according to the BPE server variables and session–specific time variables.

The BPE will stop with an error if one of the requested files is missing.

PID 002 ATX2PCV: If the BPE server variable V_MYATX contains the name of an ANTEX file, this file is used to update the "PCV information file" specified by the BPE variables V_PCVINF and V_PCV. The antenna names together with the GNSS supported

by the related receiver types are read from the "Station information file" (defined by V_STAINF).
If the resulting "Bernese phase center offsets" file differs from the file currently used for the processing in ${X}/GEN, it is copied to this location.

PID 003 COOVEL: The coordinates in the project–specific coordinate file ({V_CRDINF}.CRD in the STA–directory) may refer to any epoch (that may for instance be given by a BPE variable in the PPP examples). Using the program COOVEL, the coordinates are propagated to the epoch of the current session applying the velocities from the corresponding velocity field ({V_CRDINF}.VEL in the STA–directory).

PID 004 COOVEL: Coordinates of the IGS reference stations are given for epoch January 1, 2005, 00:00:00 (cf. IGB08_R.CRD in the STA–directory, specified as {V_REFINF}_R.CRD). The program COOVEL propagates them to the current session's epoch using IGS velocities (see IGB08_R.VEL in the STA–directory).

PID 005 CRDMERGE: The program CRDMERGE merges the coordinates of the reference frame stations (obtained in PID 004) and the project–specific coordinates (computed for the epoch to be processed in PID 003). The merged coordinate file ({V_A}yyssss.CRD, e.g., APRyyssss.CRD in the STA–directory) contains all reference frame stations completed by the coordinates for all additional sites of the project–specific coordinate file. The flags in the resulting coordinate file indicate where the coordinates are coming from (IGB08 for reference frame sites and PPP for all other entries — please note that only the first letter of the flag is used to identify the coordinates in the related fields of the processing programs).

PID 011 RNX_COP: The algorithm to select RINEX observation files, described in Section 23.4.2.3, is implemented in this script. It copies the selected files from the datapool into the campaign area and applies the Hatanaka decompression program if necessary.

The script checks whether an hourly processing scheme is applied. In that case the hourly observation files need to be merged using the next group of scripts. This is not needed for a daily processing scheme where the execution may continue with PID 099 .

If no observation file was found for the corresponding session, the processing ends with PID 999 . This allows to run through all sessions independent of whether data for all sessions are available or not.

PID 099 DUMMY: Does nothing, closes this section of scripts.

Further reading:

Section 3.1: "Overview of the Directory Structure", page 57
Section 18.3: "ANTEX Converter ATX2PCV", page 411
Section 10.7.9: "Propagating Coordinates to Specific Epochs", page 272
Section 22.5.3: "Loops and Jumps", page 491

Prepare Pole and Orbit Information

The next section of the CLKDET.PCF performs some standard preparatory steps. In particular, orbit and Earth orientation files are converted from foreign to Bernese formats. It is very important that you always use orbits together with the corresponding pole information to avoid inconsistencies.

```
#
# Prepare the pole and orbit information
# -------------------------------------
101 POLUPD   CLK_GEN           ANY      1 001
111 ORBMRG   CLK_GEN           ANY      1 001
112 PRETAB   CLK_GEN           ANY      1 111
113 ORBGEN   CLK_GEN           ANY      1 112
```

PID 101 POLUPD: The program POLUPD converts the pole file from IGS/IERS format to Bernese format. This pole file will be used for all following processing steps where EOPs are necessary.

The script checks the content of the input pole file and tries to find out the correct setting for option "Use ERP rates". Because of the diversity of the IERS/IGS formatted pole files it might happen that the algorithm fails. In this case you have to define the option manually in the user script by replacing XYZ by the source of your products (value of BPE server variable `V_B`):

```
# Repeat the program if necessary
# ------------------------------
  if ( $v_b eq "XYZ" ) { $doRate = 1 }  # if products from "XYZ" are used
                                        # apply the polar rates.
  if ( $doRate != $isRate ) {
    $bpe->putKey("$ENV{U}/PAN/POLUPD.INP","USERATE","$doRate");

    my $PGMNAM = "POLUPD";
    $bpe->RUN_PGMS($PGMNAM);
  }
```

PID 111 PRETAB: The orbit files are available in the precise orbit file format (SP3c) giving a table of satellite positions in an Earth-fixed frame (e.g., IGB 08). These positions have to be converted into the inertial frame using the program PRETAB.

PID 112 ORBGEN: The tabular orbit file is used as input for the final orbit generation step with ORBGEN. The result is a standard orbit file and a summary providing the quality of the orbit fit. It is contained in the processing summary (`CLKyyssss.PRC`).

The orbit and Earth orientation files from this step are used throughout all further processing steps. They will not be changed anymore.

Further reading:

Section 4.3: "Precise Orbit Files", page 90
Section 4.4: "IGS and IERS Pole Files", page 91
Chapter 5: "Preparation of Earth Orientation, GNSS Orbit, and Satellite Clock Information", page 107

Extract Broadcast Clocks from Navigation Files

The starting point for the clock estimation in this example are the satellite broadcast clocks. In order to use them within the *Bernese GNSS Software* they have to be extracted from the RINEX navigation files. In addition, the broadcast information must pass some initial tests. These steps are integrated in this part of the PCF .

```
#
# Extract broadcast clocks from navigation files
# ---------------------------------------------
121 CCRINEXN CLK_GE2           ANY      1 099
122 CCRINEXG CLK_GE2           ANY      1 099
123 RXNPRE   CLK_GE2           ANY      1 121 122
124 PRETAB   CLK_GE2           ANY      1 123
199 DUMMY    NO_OPT            ANY      1 113 124
```

PID 121 CCRINEXN: The available RINEX navigation files for GPS from the different stations (`ORX` directory) are merged and concatenated to one file (`RAW` directory).

PID 122 CCRINEXG: Provides the same functionality as the previous script/program but for GLONASS .

PID 123 RXNPRE: Because the GLONASS satellites broadcast their positions and velocities (as well as their accelerations) in the Earth-fixed frame (instead of orbital elements like in case of GPS), they can easily be converted to the precise orbit format. By evaluating the orbital elements broadcasted by the GPS satellites, the corresponding information may also be transferred into precise orbit format.

The program RXNPRE reads the navigation RINEX files of different systems, checks them for plausibility, and merges them into one file in the precise orbit format. The same is done with the satellite clock records obtained from the navigation messages.

PID 124 PRETAB: This call of the program PRETAB reads the satellite clock values from the precise orbit files and fits them with a simple clock model (resulting file `APRyyssss.CLK` in the `ORB` directory, where `APR` is the default value for the BPE variable `V_A`). The satellite positions extracted from the navigation messages are not used in the further processing.

PID 199 DUMMY: The script does nothing but closing the section of orbit and satellite clock extraction–related scripts in the PCF .

Because GNSS does not allow direct access to any timescale, only differences between the estimated clock corrections (e.g., with respect to a reference clock) can be interpreted. Nevertheless, all receiver clocks (including the reference clock) have to be synchronized to GPS broadcast time. This can be realized by using satellite broadcast clocks as basis for the estimation of receiver clock corrections. Alternatively, you may extract the satellite clocks from the CODE precise file using a polynomial representation (program PRETAB in PID 112).

Further reading:

Section 4.10: “RINEX Navigation Files”, page 100
Section 4.10.3: “Utilities”, page 101
Section 4.10.1: “Import to Bernese”, page 100
Chapter 5: “Preparation of Earth Orientation, GNSS Orbit, and Satellite Clock Information”, page 107
Section 5.3: “Preparation of GNSS Broadcast Information”, page 111
Section 5.6: “Preparation of Satellite Clock Corrections”, page 125

Preprocess, Convert, and Synchronize Observation Data

This part of the PCF prepares the observation data. An ASCII graphic of observations is created and a first data screening based on the RINEX level is performed. After converting RINEX files to Bernese format, receiver clocks are synchronized.

Depending on the availability of high–rate satellite clocks, the zero–difference processing example offers two versions for preprocessing the phase observation files:

- using the program MAUPRP if high–rate satellite clock corrections are available (indicated by the BPE server variable V_HRCLK == 'Y') or

- using the program RNXSMT; here, only the consistency between code and phase data of the two frequencies is checked.

```
#
# Preprocess, convert, and synchronize observation data
# ----------------------------------------------------
201 PPP_PRP  CLK_GEN           ANY      1 099
```

PID 201 PPP_PRP: Reads the value of the BPE server variable `V_HRCLK` and continues the processing at the corresponding PID:

PID 211 for preprocessing code and phase data based on RNXSMT or
PID 251 for using the program MAUPRP to clean the phase observations.

```
#
# Phase preprocessing with RNXSMT
211 RNXSMTAP CLK_GE2           ANY      1 201
212 RNXSMT_P CLK_GE2           ANY      1 211
221 RNXGRA   CLK_GEN           ANY      1 212
231 RXOBV3AP CLK_GE2           ANY      1 221
232 RXOBV3_P CLK_GE2           ANY      1 231
241 CODSPPAP CLK_GE2           ANY      1 232
242 CODSPP_P CLK_GE2           ANY      1 241
243 CODXTR   CLK_GE2           ANY      1 242
249 JUMP     NO_OPT            ANY      1 243
```

PID 211 RNXSMTAP: The purpose of this step is to clean the data on the RINEX level. It makes sense to parallelize this step, which is prepared by this script. The actual processing is done in the next PID. The script deletes files that may be present from previous runs, and prepares a list of RINEX files for parallel processing.

PID 212 RNXSMT_P: The program RNXSMT is called to clean the RINEX data and to identify cycle slips. Processing is done in parallel, i.e., the program receives a list of RINEX files to be cleaned in one run. The output consists of smoothed RINEX files (extension `SMT`) containing smoothed instead of original code observations. Cycle slips and outliers are marked. In case of RINEX (version 3) files, it will select the observation types according to the priority list given in `V_OBSINF`.

PID 221 RNXGRA: A summary of all available observation data is created giving a complete overview of observed satellites, involved stations and their performance. This file appears in the processing summary and may help to identify possible data tracking problems of observing sites.
Note for processing your own data: This script also removes stations with data problems from the processing. Stations showing large data gaps are detected by RNXGRA and listed in a deletion file (`GRAyyssss.DEL` in the `OUT` directory). This list is used by the BPE script to delete the corresponding RINEX observation files from the `RAW` directory. All deleted stations are reported in the processing summary, too.

PID 231 RXOBV3AP: The script prepares several runs of the program RXOBV3 to import the observations from RINEX into the binary Bernese format. In general, this can be done in parallel for the individual stations.

PID 232 RXOBV3_P: This script calls the program RXOBV3, which creates Bernese observation files for code and phase from the smoothed RINEX files.

The program compares the data records in the RINEX header with the entries in the station information file. Any detected header inconsistency is reported in the processing summary. Using the default settings, the BPE stops with an error if the station name is not in the abbreviation file or if an antenna type is not listed in the phase center offsets file (default: ${X}/GEN/{V_PCVINF}.{V_PCV}, e.g., ${X}/GEN/PCV.I08).

The results are zero–difference code and phase observation files for each station (extensions CZH, CZO, PZH, PZO files in the campaign's OBS directory).

Note for processing your own data: In case of inconsistencies, no Bernese observation files are written because the options for "ACTIONS IN CASE OF INCONSISTENCIES" are set to SKIP. This is the robust setting to generate a result without manual interactions, however, stations may be missing in the solution. Alternatively, you may set the options to ERROR causing the BPE to stop in case of inconsistencies.

PID 241 CODSPPAP: This script prepares the parallel run of CODSPP.

PID 242 CODSPP_P: This script runs CODSPP, the main task of which is the receiver clock synchronization. The program also determines approximate station coordinates by a straightforward point positioning approach using only smoothed code observations. This is useful if the a priori coordinates (e.g., from the RINEX header) are of doubtful quality. Outliers in the observations can be detected in this step, too.

PID 243 CODXTR: This script runs the output extraction program for CODSPP named CODXTR. It produces a summary file informing of detected outliers, missing orbits, excluded satellites due to satellite problems, and missing clocks. The extract will be a part of the protocol file.

PID 249 JUMP: The code and phase data are cleaned and ready for the next processing step. The section for preprocessing the observation files using MAUPRP does not have to be executed. Therefore, the processing continues to the end of this sequence. The task of this script is to jump to PID 299 of the PCF.

RXOBV3 runs in parallel only if the option "Action if station not in abbreviation list" in panel "RXOBV3 2: Input Options 1" is set to ERROR. Otherwise, all stations are processed in one RXOBV3 run. This prevents problems when different RXOBV3 runs try to update the station abbreviation table at the same time.

```
#
# Extract precise satellite clocks for MAUPRP-based pre-processing
251 CCRNXC   CLK_GE2           ANY      1 249
252 RNXCLK   CLK_GE2           ANY      1 251
```

PID 251 CCRNXC: The precise satellite clock information in clock RINEX format, which is needed for the preprocessing of the phase measurements in the program MAUPRP, is adjusted according to the time window of the current session.

PID 252 RNXCLK: The precise GNSS satellite clock information is extracted from the external clock RINEX file into the Bernese formatted file that is located in the ORB directory of the campaign.

```
#
# Phase preprocessing with MAUPRP
261 RNXSMTAP CLK_GEN           ANY      1 249
262 RNXSMT_P CLK_GEN           ANY      1 261
263 RNXGRA   CLK_GEN           ANY      1 262
271 RXOBV3AP CLK_GEN           ANY      1 263
272 RXOBV3_P CLK_GEN           ANY      1 271
281 CODSPPAP CLK_GEN           ANY      1 272 252
282 CODSPP_P CLK_GEN           ANY      1 281
283 CODXTR   CLK_GEN           ANY      1 282
291 MAUPRPAP CLK_GEN           ANY      1 283
292 MAUPRP_P CLK_GEN           ANY      1 291
293 MPRXTR   CLK_GEN           ANY      1 292
299 DUMMY    NO_OPT            ANY      1 201 249 293
```

PID 261 RNXSMTAP: As in PID 211, the RINEX files are divided into groups that may be processed together in parallel, which is actually performed in the next script.

PID 262 RNXSMT_H: The procedure of this script is nearly the same as in PID 222: executing the program RNXSMT. The only difference is that the original phase observations are copied into the resulting RINEX files (extension `SMT`).

PID 263 RNXGRA: As in PID 221, program RNXGRA produces a summary of the smoothed RINEX observation data, giving a complete overview of observed satellites, involved stations and their performance. This file appears in the processing summary and may help to identify possible data tracking problems of observing sites.

PID 271 RXOBV3AP: The import of the smoothed RINEX files (extension `SMT`) is prepared, where several groups of files are processed in parallel.

PID 272 RXOBV3_P: The same functions as described in PID 232 are executed resulting in binary zero–difference code and phase observation files for each station (extensions `CZH`, `CZO`, `PZH`, `PZO` files in the campaign's `OBS` directory).

The only difference is the sampling of the observation files: if phase observations are preprocessed on RINEX level using solely RNXSMT (as done in PID 222), the data sampling (as defined by the BPE server variable `V_SAMPL`) may be applied to the data import. The preprocessing of the phase measurements shall be performed later in the program MAUPRP and therefore the original sampling of, e.g., 30 seconds, must be retained in the observation files. Consequently, the only difference between the input files for RXOBV3 that are used in the PIDs 232 and 262 is the value of the option "Sampling interval" .

PID 281 CODSPPAP: This script prepares the parallel run of CODSPP; similar to PID 241 .

PID 282 CODSPP_P: As in PID 242, the program CODSPP is executed to synchronize the receiver clocks with respect to GPS system time. Since the phase measurements are not yet cleaned, outliers detected by the program CODSPP are only applied to CODE instead of BOTH observation types in option "Mark outliers in obs. files" (difference in the processing options for CODSPP between PIDs 242 and 282).

PID 283 CODXTR: From the program output of CODSPP a summary for the protocol file is extracted using the program CODXTR (same procedure as in PID 243).

PID 291 MAUPRPAP: The preprocessing of phase measurements is done by MAUPRP in this section. It may be run in parallel, which is initialized and prepared by this script.

PID 292 MAUPRP_P: This script actually runs the program MAUPRP for the selected stations to detect and correct cycle slips, to identify outliers, and to update the list of ambiguities for the phase measurements. The screening results are stored in the phase zero–difference observation files.

The screening of zero–difference observation files with MAUPRP requires well–established satellite clock corrections with the same sampling as the data files (at least 30 seconds) — even if the final zero–difference solution shall be carried out with a lower sampling (e.g., 5 minutes according to BPE server variable `V_SAMPL`). This higher sampling for screening the phase observations is required to improve the capability to distinguish between potential cycle slips and the change of the ionosphere from one epoch to the next one.

PID 293 MPRXTR: The script runs the program MPRXTR to extract the essential information from the MAUPRP program output files into a summary table that will be included in the protocol file later on.

PID 299 DUMMY: Does nothing, closes this section.

Further reading:

Section 4.2: “RINEX Observation Files”, page 75
Section 6.2: “Preprocessing on the RINEX Level”, page 129
Section 6.2.5: “Code Smoothing”, page 132
Section 4.2.3: “Import to Bernese”, page 79
Section 6.3: “Receiver Clock Synchronization and Preprocessing of Code Observations”, page 135
Section 6.5: “Preprocessing Phase Observations”, page 144
Section 22.5.2: “Parallel Processing”, page 490
Section 22.5.3: “Loops and Jumps”, page 491

Compute Weights for the Code

The noise level of the code observation differs between different receiver types. In addition, multipath effects may have a significantly different impact on the code measurements for different tracking stations. These are the main reasons why a station–dependent weighting for the code data is helpful. In this section of scripts the station–dependent weights for the code measurements in the zero–difference network is derived from the residuals of a GPS–only solution.

```
#
# Compute weights for the code
# ----------------------------
301 MKCLUSAC CLK_WGT           ANY      1 299
302 TIMEST_C CLK_WGT           ANY      1 301
303 GPSXTR   CLK_WGT           ANY      1 302
304 CLK_CHK  CLK_WGT           ANY      1 303
305 RESRMS   CLK_WGT           ANY      1 304
```

PID 301 MKCLUSAC: Prepares a parallel processing of stations of the zero–difference network using only the code measurements (indicated by the last letter `C` of the script name). The stations are automatically divided into regional clusters using the program MKCLUS. The number of clusters is given by the BPE server variable `V_CLUCOD`. By default, this is set to `1`, i.e., all stations are processed in one cluster.

PID 302 TIMEST_C: A full zero–difference network solution is performed. Station coordinates, troposphere parameters, receiver and satellite clock corrections are estimated for each of the clusters defined by the previous script. The residuals are stored in corresponding binary files (`{V_C}ssssnnn.RES`, where `V_C` is a BPE server variable with the default value `CW`, `ssss` is the session to be processed, and `nnn` denotes the SUB_PID of the related parallel BPE job).

The script calls the program GPSEST but selects only the code zero–difference observation files assigned by the preparatory script to the corresponding parallel job. This is indicated by the letter `C` at the end of the script name (instead of the typically used letter `P` for parallel scripts).

PID 303 GPSXTR: Creates an extract of the GPSEST program output for the processing summary.

PID 304 CLK_CHK: This script runs a sequence of the programs RESRMS and RESCHK on the residuals obtained in PID 302 . The task is to identify misbehaving stations in the cluster solutions that may — in the extreme case — disturb the least squares solution in PID 302, during which the residuals are generated. The observation files for such stations (code and phase observation files) are deleted. This deletion is reported in the processing summary and the processing continues with PID 301 to create a new set of residual files without these stations.

PID 305 RESRMS: Evaluates the residuals from PID 302 and computes the station–dependent weighting factors. They are stored in the file `{V_C}_yyssss.SOS` in the `STA` directory. The program output of this step is a component of the processing summary.

Further reading:

Chapter 7: “Parameter Estimation”, page 169
Section 7.7.1: “Pre–Elimination of Parameters”, page 182
Section 7.7.2: “Back–Substitution of Epoch–Parameters”, page 183
Section 7.4.4: “Real and Normalized Residuals”, page 174
Section 6.6: “Screening of Post–Fit Residuals”, page 161
Section 6.6.3: “Detect Misbehaving Stations and Satellites”, page 165
Section 6.6.2: “Generating Residual Statistic”, page 162
Section 24.7.14: “Station Observation Sigma Factor File”, page 743
Section 22.5.2: “Parallel Processing”, page 490
Section 22.5.3: “Loops and Jumps”, page 491

Clean Code Residuals

```
#
# Clean code residuals
# --------------------
321 MKCLUSAC CLK_COD           ANY      1 305
322 TIMEST_C CLK_COD           ANY      1 321
323 GPSXTR   CLK_COD           ANY      1 322
324 CLK_CHK  CLK_COD           ANY      1 323
325 RESRMS   CLK_COD           ANY      1 324
331 MRKOBSAP CLK_COD           ANY      1 325
332 MRKOBS_P CLK_COD           ANY      1 331
```

PID 321 MKCLUSAC: This script prepares a parallel processing of the code measurements, similar to the script PID 301 . The only difference is that there is a redundancy requirement in option “Minimum number of observations per satellite at each epoch” (panel “MKCLUS 3: Regional Cluster Definition Options (Zero Differences)”). In this way it is assured that each observation is confirmed by at least another one in the same cluster to guarantee a reliable residual computation even at the edge of a regional network. This requirement may result in the fact that a few stations are processed in more than one cluster.

PID 322 TIMEST_C: The code measurements of the stations assigned to each of the clusters are processed together by the program GPSEST, solving for station coordinates, troposphere parameters, and receiver and satellite clock corrections. All GNSS indicated in the BPE server variable `V_SATSYS` are considered. The observations are weighted according to the factors in the file `{V_C}_yyssss.SOS` . The only results stored are the residuals (filename `{V_E}ssssnnn.RES` with `CR` as the default value for `V_E`).

PID 323 GPSXTR: Creates a summary of the GPSEST program output for the processing protocol.

PID 324 CLK_CHK: Based on a sequence of RESRMS and RESCHK calls, stations with bad observations are detected and their observation files (code and phase) are removed from the further processing. Such measurements may disturb the least squares adjustment if they contain too large or too many outliers. The processing summary reports whether observation files have been deleted or not. In the case of deletions the processing continues at PID 321 to recompute the residual files without the misbehaving stations.

PID 325 RESRMS: Analyzes all residuals from PID 322 to detect observations related to residuals exceeding a certain threshold (given in option "Detect large residuals: code limit"). These measurements are listed in an edit information file named `{V_E}yyssss.EDT` with `CR` as the default value for `V_E`.

PID 331 MRKOBSAP: The problematic measurements indicated in the edit information file are now marked in the Bernese observation files. This can be done in parallel mode. The script supports — as most of the other parallelization scripts — two strategies: either the BPE variable `V_CLU` is defined and provides the number of clusters or a station–by–station processing is initialized. In the `CLKDET.PCF` the first method is applied.

PID 332 MRKOBS_P: The script runs the program SATMRK to apply the corrections to the selected Bernese observation files indicated by the edit information file from PID 325.

Further reading:

Chapter 7: "Parameter Estimation", page 169
Section 7.7.1: "Pre–Elimination of Parameters", page 182
Section 7.7.2: "Back–Substitution of Epoch–Parameters", page 183
Section 15.2: "Precise Clock Estimation", page 355
Section 7.4.4: "Real and Normalized Residuals", page 174
Section 6.6: "Screening of Post–Fit Residuals", page 161
Section 6.6.3: "Detect Misbehaving Stations and Satellites", page 165
Section 6.7: "Marking of Observations", page 167
Section 22.5.2: "Parallel Processing", page 490
Section 22.5.3: "Loops and Jumps", page 491

Unified Inter–System/Inter–Frequency Code Biases

In case of a multi–GNSS processing (e.g., considering GPS and GLONASS) an inter–system bias (ISB) between the the different systems needs to be estimated for each receiver. A zero–mean condition over all receivers in the network is established, which means in the very end that the the reference for the satellite clock corrections within each GNSS is affected. In case of GLONASS on top of the ISB also the inter–frequency bias (IFB) becomes relevant.

These biases need to be unified and introduced when the observations are processed in different clusters for the final solution. This computation of biases is performed in the following sequence of scripts. The analysis of code data is sufficient because the related biases are absorbed by the ambiguity parameters in case of processing phase measurements.

```
#
# Unified inter-system/inter-frequency code biases
# -----------------------------------------------
341 MKCLUSAC CLK_IFB           ANY      1 332
342 TIMEST_C CLK_IFB           ANY      1 341
343 IFBNEQ   CLK_IFB           ANY      1 342
390 COD_SUM  NO_OPT            ANY      1 343
399 DUMMY    NO_OPT            ANY      1 390
```

PID 341 MKCLUSAC: Provides the same functionality as PID 321 .

PID 342 TIMEST_C: Calls GPSEST to analyze the code measurements in order to set up a NEQ file containing station coordinates, troposphere parameters, IFB– and DCB–parameters as well as receiver and satellite clock parameters. The clock parameters are only implicitly included (pre–eliminated before saving the NEQ file). Only one set of satellite clock parameters remains in the NEQ files every hour (see option "Sampling rate for preelimination").

Note that there is no solution computed here, only the NEQ is constructed from the observations ("Stop program after NEQ saving" is enabled).

PID 343 IFBNEQ: Using the program ADDNEQ2 the NEQ files from the previous PID are combined, the coordinates and troposphere parameters are pre–eliminated and a solution is generated. The results for the ISB/IFB parameters are stored together with the DCB parameters in the file `{V_F}_yyssss.DCB` (with `CB` as the default value for `V_F`) to be used for further processing. Note that the DCB values are constrained to the values introduced by the input file `P1C1yymm.DCB` .

ADDNEQ2 also checks for exceptionally large IFB estimates (see option "Inter-frequency bias, alarming limit"). In that case the particular satellite is excluded from the further processing.

PID 390 COD_SUM: Collects some information from the code–data related steps for the processing summary.

PID 399 DUMMY: Does nothing but closing the section on processing the code measurements.

Further reading:

Chapter 7: "Parameter Estimation", page 169
Section 7.7.1: "Pre–Elimination of Parameters", page 182
Section 7.6.2: "Epoch–Parameters", page 182
Section 15.2.2: "GLONASS Clock Estimation", page 358

Preprocessing: Residual Check (only if no precise clocks are available)

If the preprocessing of the phase–measurements has only been performed by RNXSMT (BPE server variable V_HRCLK == 'N'; running through PID 221 to 249), only a consistency check between the code and phase measurements on both frequencies has been done so far. Hence, so far the screening of the phase data depends on the noise level of the code observations. For that reason an additional screening of the residuals with a threshold is needed before the next step.

If the preprocessing of the phase observations has been done with the sophisticated algorithms of MAUPRP (BPE server variable V_HRCLK == 'Y'; running through PID 251 to 293), this sequence of residual screening can be skipped.

```
#
# Preprocessing: residual check, from RNXSMT
# -----------------------------------------
401 PPP_PRP  CLK_GEN          ANY      1 399
#
411 MKCLUSAP CLK_EDO          ANY      1 401
412 TIMEST_P CLK_EDO          ANY      1 411
413 GPSXTR   CLK_EDO          ANY      1 412
414 CLK_CHK  CLK_EDO          ANY      1 413
415 RESRMS   CLK_EDO          ANY      1 414
421 MRKOBSAP CLK_EDO          ANY      1 415
422 MRKOBS_P CLK_EDO          ANY      1 421
429 GOTO     NO_OPT           ANY      1 422
```

PID 401 PPP_PRP: As in PID 211 the script reads the BPE server variable V_HRCLK and decides at which PID the processing continues:

PID 451: section is skipped (if V_HRCLK == 'Y') or
PID 411: section is executed (if V_HRCLK == 'N').

PID 411 MKCLUSAP: Distributes the stations into regional clusters to speed up the residual computation. As in PID 321 for screening the code measurements, a certain redundancy is requested for each estimated (satellite) clock correction to ensure a reliable residual computation even on the boundary of a regional network.

Because now code and phase measurements are processed together, more parameters need to be estimated (in particular phase ambiguities). For that reason the BPE server variable V_CLUPRP is now used to define the maximum number of clusters (to be filled into option "Number of clusters").

As usual, the result files of a potential previous run of the program MKCLUS generated within this section are removed from the campaign directory structure.

PID 412 TIMEST_P: Computes a zero–difference network solution based on code– and phase–measurements.

Coordinates, troposphere parameters, and receiver and satellite clock corrections are estimated. The previously computed IFB/ISB biases are introduced from the file {V_F}_yyssss.DCB (default value CB for variable V_F). A zero–mean condition for all corrections estimated for the station clock parameters is set up instead of selecting one specific station as the reference clock. Of each cluster the program output and residual files are stored in files named {V_G}ssssnnn, with TO as the default value for V_G.

PID 413 GPSXTR: Collects essential information from the GPSEST program output for the processing summary.

PID 414 CLK_CHK: Within a sequence of the programs RESRMS and RESCHK the residuals are analyzed for stations producing huge residuals that may disturb the least squares adjustment. If such stations are detected, a notice appears in the processing summary. The corresponding (code and phase) observation files are deleted, and the processing is continued at PID 411 in order to recompute the residuals without these stations.

PID 415 RESRMS: A residual summary for the processing summary is generated and residuals exceeding a threshold of 0.020 m (code measurements 2 m) are indicated in an edit information file for removal.

PID 421 MRKOBSAP: Preparing the removal of observations related to large residuals in a parallel mode for groups of stations using the BPE server variable V_CLU as the maximum number of stations per group.

PID 422 MRKOBS_P: Marks the data with residuals exceeding the threshold and those identified by RESRMS (PID 415) in the Bernese code and phase observation files.

PID 429 GOTO: This script is included to allow iterations in this sequence of residual screening, if needed. In that case you have to add the PID 411 as PARAM2 (or even PARAM3 in the PCF behind the NEXTJOB indicator related to this PID in order to run a second (or third) iteration.

Further reading:

Chapter 7: "Parameter Estimation", page 169
Section 15.2: "Precise Clock Estimation", page 355
Section 7.7.1: "Pre–Elimination of Parameters", page 182
Section 7.7.2: "Back–Substitution of Epoch–Parameters", page 183
Section 7.4.4: "Real and Normalized Residuals", page 174
Section 6.6: "Screening of Post–Fit Residuals", page 161
Section 6.6.3: "Detect Misbehaving Stations and Satellites", page 165
Section 6.7: "Marking of Observations", page 167
Section 22.5.2: "Parallel Processing", page 490
Section 22.5.3: "Loops and Jumps", page 491

Preprocessing: Residual Check, last iteration

This section is similar to the previous section but the residual screening is performed with a reduced threshold of 6 mm . If the preprocessing for the phase data is done by RNXSMT it is the final residual screening. In case of the MAUPRP–based preprocessing it is not only a residual check but the first combined processing of code and phase measurements. Potential inconsistencies between both observation types are detected here and potentially affected measurements are excluded or — in the extreme case — the station is completely excluded from the further processing.

```
#
# Preprocessing: residual check, last iteration
# ---------------------------------------------
451 MKCLUSAP CLK_EDT           ANY      1 429
452 TIMEST_P CLK_EDT           ANY      1 451
453 GPSXTR   CLK_EDT           ANY      1 452
454 CLK_CHK  CLK_EDT           ANY      1 453
455 RESRMS   CLK_EDT           ANY      1 454
461 MRKOBSAP CLK_EDT           ANY      1 455
462 MRKOBS_P CLK_EDT           ANY      1 461
471 CCRNXC   CLK_EDT           ANY      1 455
489 GOTO     NO_OPT            ANY      1 462 471
#
499 DUMMY    NO_OPT            ANY      1 489
```

PID 451 MKCLUSAP: Same as in PID 411 .

PID 452 TIMEST_P: Same as in PID 412 but the filenames are based on the solution ID V_H (default value TR) instead of V_G . In addition to the residuals, the resulting receiver and satellite clock corrections from each cluster are stored in a clock RINEX file.

PID 453 GPSXTR: A summary for the processing protocol is extracted from the GPSEST program output in PID 452 .

PID 454 CLK_CHK: Same as PID 414 but with reduced thresholds for excluding stations. In case of station exclusion, the process continues with PID 451 to recompute the residuals without the misbehaving station.

PID 455 RESRMS: Same as PID 415 but with a reduced threshold of 6 mm for the residual screening.

PID 461 MRKOBSAP: Same as PID 421 .

PID 462 MRKOBS_P: Same as PID 422 .

PID 471 CCRNXC: The receiver and satellite clock corrections computed in PID 452 within the clusters are collected and combined to one clock RINEX file `{V_H}yyssss.CLK` .

PID 489 GOTO: As with PID 429, an iteration of this section is prepared if it is necessary when processing your own data. In that case for each additional iteration the PID 451 should be added as `PARAM2` for this script after the `NEXTJOB` keyword in the PCF .

PID 499 DUMMY: Does nothing but ends the section of residual screening.

Further reading:

Chapter 7: "Parameter Estimation", page 169
Section 15.2: "Precise Clock Estimation", page 355
Section 7.7.1: "Pre–Elimination of Parameters", page 182
Section 7.7.2: "Back–Substitution of Epoch–Parameters", page 183
Section 7.4.4: "Real and Normalized Residuals", page 174
Section 6.6: "Screening of Post–Fit Residuals", page 161
Section 6.6.3: "Detect Misbehaving Stations and Satellites", page 165
Section 6.7: "Marking of Observations", page 167
Section 15.3: "Clock RINEX Utilities in *Bernese GNSS Software*", page 365
Section 22.5.2: "Parallel Processing", page 490
Section 22.5.3: "Loops and Jumps", page 491

Combined Code and Phase Solution

Now that the data screening is completed, a final solution can be computed.

```
#
# Combined code and phase solution
# --------------------------------
501 MKCLUSAP CLK_RES           ANY      1 499
502 TIMEST_P CLK_RES           ANY      1 501
503 GPSXTR   CLK_RES           ANY      1 502
504 RESRMS   CLK_RES           ANY      1 502
521 CLKNEQ   CLK_RE2           ANY      1 503
522 GPSXTR   CLK_RE2           ANY      1 521
523 CCRNXC   CLK_RES           ANY      1 521
#
599 DUMMY    NO_OPT            ANY      1 522 523
```

PID 501 MKCLUSAP: For the computation of the final solution, it is preferable not to apply any clustering. If necessary, a global clustering is preferable to compute all satellite clocks from different network configurations. For this clustering the two BPE variables `V_CLUSOL`, defining the number of clusters, and `V_MAXSOL`, defining the maximum number of stations per cluster, are responsible. They are inserted into the program MKCLUS if the value of the option "Strategy for zero difference observations" is GLOBAL .

Before the processing itself is started, all result files of a potential previous execution of this section are deleted.

PID 502 TIMEST_P: The script runs GPSEST to generate the final solution. The DCB together with the IFB/ISB values of PID 343 are introduced as known. Even if the clock parameters are linear the a priori values should not be too far from the results to prevent numerical problems. For that reason the a priori receiver and satellite clock values are taken from the last preprocessing step (clock RINEX files `{V_H}yyssss.CLK`, generated in PID 471). Station coordinates and troposphere parameters are estimated along with the receiver and satellite clock corrections. As usual in the script `TIMEST_P`: if a coordinate and/or troposphere file is specified by the BPE server variables `V_FIXCRD` and `V_FIXTRP`, respectively, only those coordinates or troposphere parameters are estimated that are not introduced by these files. Apart from the program output the normal equation with coordinate and troposphere parameters, the clock corrections (clock RINEX format) and the residuals are stored in files with the name `{V_I}ssssnnn` with corresponding extensions (the default value for `V_I` is `TP`).

PID 503 GPSXTR: A summary of the GPSEST run is extracted for the processing protocol file.

PID 504 RESRMS: A residual statistics of the final solution is created with RESRMS. This overview is contained in the processing summary.

PID 521 CLKNEQ: Coordinate and troposphere parameters included in the NEQ files of the clock estimation (PID 502) are introduced to the programADDNEQ2. The coordinate and troposphere result files of this solution are merged with values introduced into the GPSEST run of PID 502 to one coordinate file and one Bernese troposphere file named `{V_I}yyssss.CRD` and `{V_I}yyssss.TRP`, respectively.

PID 522 GPSXTR: Extract statistical information from the previous ADDNEQ2 run for the processing summary.

PID 523 CCRNXC: The program CCRNXC combines the estimated clock corrections from all cluster sub–solutions of PID 502, identifies the best reference clock among all station clocks and generates the final clock solution file (clock RINEX and Bernese clock file). The processing summary contains the resulting CCRNXC output providing information on the selected reference clock and the performance of a polynomial fit of the estimated clock corrections.

PID 599 DUMMY: Does nothing but closing this section.

Further reading:

Chapter 7: "Parameter Estimation", page 169
Section 7.7.1: "Pre–Elimination of Parameters", page 182
Section 7.7.2: "Back–Substitution of Epoch–Parameters", page 183
Chapter 15: "Clock Estimation", page 353
Section 15.2: "Precise Clock Estimation", page 355
Section 15.3: "Clock RINEX Utilities in *Bernese GNSS Software*", page 365
Section 4.9: "Clock RINEX File", page 99
Section 22.5.2: "Parallel Processing", page 490

Create Summary File and Delete Files

The last part of the PCF creates a protocol file (summarizing the clock generation procedure) and deletes dispensable files generated during the run of the PCF . The scripts do not run any Bernese programs.

```
#
# Create summary file and delete files
# -----------------------------------
901 CLK_SUM  CLK_GEN           ANY      1 599
902 CLK_SAV  CLK_GEN           ANY      1 901
903 OBS_SAV  CLK_GEN           ANY      1 901
904 CLK_DEL  CLK_GEN           ANY      1 902 903
991 BPE_CLN  CLK_GEN           ANY      1 904
#
# End of BPE
# ----------
999 DUMMY    NO_OPT            ANY      1 991
```

PID 901 CLK_SUM: All old protocol files of the processed session are deleted. The comprehensive summary is composed of several parts describing the different steps. The protocol file is discussed in Section 23.4.4.7 .

PID 902 CLK_SAV: The result files of this BPE (listed in Section 23.4.4.4) are copied from the campaign into the SAVEDISK area.

PID 903 OBS_SAV: The observation files of this BPE are copied from the campaign into the SAVEDISK area, see description of BPE server variable V_SAVOBS.

PID 904 CLK_DEL: This script deletes all files that are no longer needed after the end of the BPE. This prevents unimportant files from accumulating.

PID 991 BPE_CLN: Deletes files from the campaign's BPE subdirectory with a delay of 30 sessions. Thus, all BPE protocol and log files of the last 30 sessions will remain in the directory.

PID 999 DUMMY: Does nothing. This script provides a well–defined PID as end point, useful, e.g., as jump address, or to check for the completion of the BPE.

If you want to archive other or more files you should not remove these files from the list in the deletion script (CLK_DEL) but add them to the list of files to be copied to the SAVEDISK area (CLK_SAV).

The DUMMY–script may seem useless at a first glance. But if you have such a script running at a well defined PID (e.g., PID 999) in all your PCFs the successful execution of a BPE can be tested in a generic way by checking the existence of the DUMMY–script's LOG– or PRT–file. In this way a script starting a BPE can check the successful completion of the BPE and react in case of errors (e.g., send an error mail).

Further reading:

Section 22.10: "BPE Output and Protocol Files", page 518
Section 22.11: "Error Handling", page 522

23.4.4.7. Description of the Protocol File, Quality Control

The processing is summarized in a so–called protocol file. It is compiled in the BPE user script CLK_SUM and exists first in the OUT directory of the campaign. If the BPE variable V_SAV == 'Y' the protocol is copied by the CLK_SAV script to ${S}/{V_RESULT}/yyyy/OUT/CLKyyssss.PRC and is removed from the campaign area by the script CLK_DEL (if V_DEL == 'Y').

The processing summary starts with reporting all important input information defined by BPE server variables used for the current processing:

```
===============================================================================
CLKDET BPE PROCESSING SUMMARY FOR YEAR-SESSION 10-2070
===============================================================================


Summary file generated at 13-Dec-2012 08:11:18 by CLK_SUM

General files:
   Antenna phase center eccentricity file:   PCV.I08
   Satellite information file:               SATELLIT.I08
   Satellite problem file:                   SAT_2010.CRX
   Receiver characterization file:           RECEIVER.
   Orbit, ERP and clock products used from:  ${D}/COD

...
```

The processing report is divided into sections related to the different tasks/features of this BPE:

PART 1: RINEX PSEUDO-GRAPHICS

The content of the RINEX observation files is summarized here (summary file output from program RNXGRA). Here you may detect stations with problematic data (e.g., reduced tracking performance of a station).

Further reading:

Section 4.2.5: “Utilities”, page 87

PART 2: ORBIT GENERATION SUMMARY

The summary file of ORBGEN reports the RMS of the fit of the satellite position given in the precise orbit file ${D}/{V_B}/{V_B}wwwwd.PRE with the orbit model of ORBGEN. It is expected that the RMS is on the few millimeter level (pseudo–stochastic pulses are set up every twelve hours to compensate for different orbit models in ORBGEN and the external program which was used to generate the orbits).

```
===============================================================================
PART 2: ORBIT GENERATION SUMMARY
===============================================================================

 RNX2SNX_102070: Generate GNSS standard orbit                    13-DEC-12 07:37
 TIME FROM DAY : 1 GPS WEEK: 1594
      TO   DAY : 2 GPS WEEK: 1594
 ------------------------------------------------------------------------------
 ORBIT REPEATABILITY FROM A 1-DAY FIT THROUGH DAILY ORBIT SOLUTIONS (MM)
 # ECLIPSING SATELLITES:   5 E /   0 M (  0 EM)
 ----------------------------------------------------------------------------------------- ...
 ECL   ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  E.  E.  E.  ..  ..  ..  ..  ..  ..  ...
 DOY    1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  ...
 ----------------------------------------------------------------------------------------- ...
 207    1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1  ...
 ----------------------------------------------------------------------------------------- ...
 ALL    1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1  ...
 ----------------------------------------------------------------------------------------- ...
```

If the RMS is larger, you should check the consistency of the Earth rotation files ${D}/{V_B}/{V_B}wwww7.ERP or the corresponding daily files ${D}/{V_B}/{V_B}wwwwd.ERP if you were using them instead.

Since CODE has improved its solar radiation pressure model in its operational solution from January 2015 onwards, an update of the software (published as B_049) is needed to allow the fit of the precise orbit files with the expected quality. After updating the software you may choose the new model by selecting “dynamical orbit parameters” as “System D2X” in panel “ORBGEN 3.1: Options”.

Further reading:

Section 4.3: “Precise Orbit Files”, page 90
Section 4.4: “IGS and IERS Pole Files”, page 91
Chapter 5: “Preparation of Earth Orientation, GNSS Orbit, and Satellite Clock Information”, page 107

PART 3: SINGLE-POINT-POSITIONING SUMMARY

The output of program CODXTR, which gives an overview of the performance of CODSPP, synchronizes the receiver clocks to GPS system time using pseudorange measurements. It is expected that the RMS is smaller than 5.00 m. No additional exclusion periods should be reported in this part.

```
================================================================================
PART 3: SINGLE-POINT-POSITIONING SUMMARY
================================================================================
...
  13 FILES, MAX. RMS:     2.80 M FOR STATION: GANP 11515M001
            MAX. BAD:     4.56 % FOR STATION: WSRT 13506M005


-----------------------------------------------------------------
>>> CPU/Real time for pgm "CODXTR": 0:00:00.004 / 0:00:00.004
>>> Program finished successfully
```

Additional exclusion periods may be caused by serious data problems for a specific station (please have a closer look to its results later in the protocol) or by some problems/inconsistencies with the introduced satellite clock file (if all stations are listed for a specific interval).

Further reading:

Section 6.3: “Receiver Clock Synchronization and Preprocessing of Code Observations”, page 135

PART 4: WEIGHTING OF CODE OBSERVATIONS

First, a GPS–only network solution based solely on the code measurements is estimated. Satellite as well station clocks are estimated. The real (not normalized) residuals are stored and analyzed to divide the stations into classes according to the noise level of the code measurements. Typically, (nearly) all stations are in the upper half of the list below:

```
================================================================================
PART 4: WEIGHTING OF THE CODE OBSERVATIONS
================================================================================
...

 STATION OBSERVATION SIGMA FACTOR SUMMARY:       ${P}/EXAMPLE/STA/CW_102070.SOS
 ----------------------------------------

 Sigma factor                                               Number of stations
 -------------------------------------------------------------------------------

     1.000                                                       6      46.2 %
     1.414   (measurement noise larger than      0.0010 m)       6      46.2 %
     1.732   (measurement noise larger than      0.0015 m)       1       7.7 %
     2.000   (measurement noise larger than      0.0020 m)       0       0.0 %
     2.236   (measurement noise larger than      0.0025 m)       0       0.0 %
     2.449   (measurement noise larger than      0.0030 m)       0       0.0 %

 -------------------------------------------------------------------------------
```

If this is not the case, please verify whether the algorithm to detect misbehaving stations made a proper decision.

Further reading:

Section 6.6.2: "Generating Residual Statistic", page 162
Section 24.7.14: "Station Observation Sigma Factor File", page 743

PART 5: SCREENING OF CODE OBSERVATIONS

The screening of code observations takes place in a zero–difference network solution considering only the code measurements. The residuals of this solution are checked. According to the value of the BPE server variable V_CLUCOD, the stations are processed in a single cluster or in several clusters.

First, the extraction from the parameter estimation is given. The RMS is expected to be a few tenths of millimeters because we use smoothed code observations. A certain redundancy is requested to compute reliable residuals. Therefore, the total number of files given in the individual output files may exceed the number of stations if you process more than one cluster.

The residuals are analyzed for misbehaving stations or satellites (e.g., due to exceptionally huge satellite clock biases). Any detected event is reported in the RESCHK program output. You can verify the overall picture in this step by inspecting the table of residual RMS for the stations and satellites.

Further reading:

Section 7: "Parameter Estimation", page 169
Section 15: "Clock Estimation", page 353
Section 7.4.4: "Real and Normalized Residuals", page 174
Section 6.6: "Screening of Post–Fit Residuals", page 161
Section 6.7: "Marking of Observations", page 167

PART 6: INTER-SATELLITE/P1-C1 CODE BIASES

In the next step the code data are processed to obtain the relevant biases: ISB/IFB. For this purpose, the code measurements are processed in a zero–difference network solution. If the BPE server variable V_CLUCOD is larger than one, the data files are divided into clusters. The network solution is computed in two steps:

- setup normal equations in GPSEST without computing a solution (number of clusters according to V_CLUCOD) and
- a subsequent ADDNEQ2 run to collect the normal equations of the clusters and to compute the solution.

Again, the RMS is expected to be a few tenths of a millimeter. Note: for each GLONASS satellite an individual bias is computed to be on the safe side in case the IFB is different for the two GLONASS satellites using the same frequency number (which can happen from time to time).

The P1−C1 DCBs introduced into this step are copied together with the ISB/IFB into an output file.

Further reading:

Chapter 7: “Parameter Estimation”, page 169
Section 7.7.1: “Pre–Elimination of Parameters”, page 182
Section 7.6.2: “Epoch–Parameters”, page 182
Chapter 15: “Clock Estimation”, page 353
Section 15.2.2: “GLONASS Clock Estimation”, page 358

PART 7: DATA SCREENING SUMMARY

If the phase measurements have been preprocessed by MAUPRP, this section starts with the extraction summary from MPRXTR. You should be attentive to the RMS of the epoch–difference solution which should be smaller than 20 to 30 mm. If all stations show large RMS values, a problem with the satellite clock corrections is likely the source.

If you have no high–rate precise clocks and have to go through RNXSMT to preprocess the phase data, a combined code+phase residual screening is performed. It is the same procedure as described in PART 8 but with higher thresholds for the residual screening. It may happen that large residuals appear for individual satellites/stations stemming from the limitations of the combined code and phase screening in RNXSMT.

Further reading:

Section 6.5: “Preprocessing Phase Observations”, page 144
Section 7.4.4: “Real and Normalized Residuals”, page 174
Section 6.6: “Screening of Post–Fit Residuals”, page 161
Section 6.7: “Marking of Observations”, page 167

PART 8: CODE+PHASE RESIDUAL SCREENING SUMMARY

Independent from the way of preprocessing the phase data, this section contains the output and summary files of a combined code and phase zero–difference network solution. The RMS of the combined code+phase network solution should be at the 1 mm level or slightly smaller. The number of clusters depends on the value of the BPE server variable `V_CLUPRP`. If more than one cluster is computed, the stations may appear in several clusters to guarantee a certain redundancy to generate reliable residuals for screening. The summary of the GPSEST to compute the residuals consists of statistics on the residual screening, (RMS before and after the screening, and the number of removed measurements) as provided by RESCHK. The tables are useful to detect whether a specific problem is related to a station or to a satellite. The RMS after the screening should be in the order of 1 mm or slightly smaller. It should also be verified that no more than 3% of the data is removed. Otherwise you have to find out whether there are some consistency problems between your orbit and satellite clocks or whether one or a few stations have introduced the problem (e.g., due to tracking problems of the receiver).

Further reading:

Section 7: “Parameter Estimation”, page 169
Section 15: “Clock Estimation”, page 353
Section 7.4.4: “Real and Normalized Residuals”, page 174
Section 6.6: “Screening of Post–Fit Residuals”, page 161
Section 6.7: “Marking of Observations”, page 167

PART 9: STATISICS ON THE SOLUTION

The statistics on the solution is provided in three parts.

PART 9.1: CODE+PHASE CLUSTER SOLUTION SUMMARY

The final solution is computed as a zero–difference network solution where the stations may be processed in a certain number of global clusters. The number of clusters is given by the BPE server variable V_CLUSOL. The summary of the GPSEST of each cluster is given together with the output of an ADDNEQ2 run to combine the coordinate and troposphere results from the different clusters. The RMS is expected to be at the one millimeter level or slightly smaller.

```
================================================================================
PART 9.1: CODE+PHASE CLUSTER SOLUTION SUMMARY
================================================================================

 TP2070001.OUT    Rms:  0.8 , # fil.:  26 , # obs.:  108513 , # par.: 10901
 (DOY: 207)


 TPA102070.OUT    Rms:  0.8 , # fil.:  26 , # obs.:  108513 , # par.: 10901
 (DOY: 207)
```

Further reading:

Chapter 7: “Parameter Estimation”, page 169
Chapter 15: “Clock Estimation”, page 353
Section 7.9.2: “The Output Extraction Program GPSXTR”, page 189

PART 9.2: CODE+PHASE CLUSTER SOLUTION RESIDUALS

The back–substitution step to compute clocks and residuals in GPSEST is only performed for phase measurements (the code data is only used to determine the parameters that depend on several epochs such as ambiguities). For that reason, only a residual statistics for the phase measurements is provided here. This feature has the advantage that no clock parameters are computed from code observations only.

Further reading:

Section 7.4.4: “Real and Normalized Residuals”, page 174
Section 6.6.2: “Generating Residual Statistic”, page 162

PART 9.3: CODE+PHASE CLUSTER SOLUTION CLOCK STATISTICS

This section contains an overview of the performance of the receiver clocks as provided by the statistics part of the CCRNXC program output.

```
================================================================================
PART 9.1: CODE+PHASE CLUSTER SOLUTION SUMMARY
================================================================================

...
 REFERENCE CLOCK SELECTION FOR OUTPUT FILE
 ----------------------------------------

 Selected reference station:     WTZR 14201M010
 Valid for interval              2010-07-26 00:00:00 to 2010-07-26 23:55:00
...
```

```
 STATISTICS ON THE CLOCKS IN THE OUTPUT FILE
 -------------------------------------------
                         # per file         rms of poly. fit (ns)
 Clock name              out    001         n = 0    n = 1    n = 2
 ------------------------------------------------------------------
 WTZR 14201M010          288    288          1.160    0.000    0.000
 WSRT 13506M005          288    288          4.832    0.030    0.030
 PTBB 14234M001          288    288          0.098    0.056    0.041
 MATE 12734M008          288    288         28.853    0.057    0.055
 ONSA 10402M004          284    284          5.941    0.097    0.090
...
```

The first clock in the list was selected as reference clock (best performing clock in the series). If you have more external high performance clocks driving the GNSS receivers, the numbers for the `rms of poly. fit` with `n=1` (linear clock model) should be smaller. For H–Maser and Caesium clocks RMS values below 0.05 and 0.10 ns are expected, respectively. Do not care about large or even huge RMS values for quartz or other low quality (internal) clocks in the receivers.

Further reading:

Section 15.3: "Clock RINEX Utilities in *Bernese GNSS Software*", page 365

23.4.5. Example for LEO Precise Orbit Determination (LEOPOD.PCF)

23.4.5.1. Purpose and Description

This is an example for a BPE to perform a reduced–dynamic and kinematic orbit determination (see Sections 17.2 and 17.3) for one Low Earth Orbiter (LEO), based on zero–difference GNSS data. It starts with screening the carrier phase observation files, computes a reduced–dynamic orbit solution with piecewise constant accelerations as empirical parameters as well as a kinematic orbit solution. At the end both orbit solutions are compared to get an indicator of the orbit quality.

The BPE is prepared for GRACE–A and GRACE–B. After adapting the data sources and importing mission–specific files (see Section 17.4), it should be usable also for other LEOs.

23.4.5.2. Input Files

The input files are copied from the `DATAPOOL` area (`${D}`) into the campaign area by the BPE user script `LEO_COP`. Many of these files are defined using BPE server variables described in Section 23.4.5.5 :

- **External GNSS satellite orbit, ERP, and clock products files** (mandatory)

  ```
  ${D}/{V_B}/{V_B}wwwwd.PRE
  ${D}/{V_B}/{V_B}wwww7.ERP
  ${D}/{V_B}/{V_B}wwwwd.CLK
  ```

  The variable `V_B` defines the product series to be considered for the PPP–based precise orbit determination (e.g., `COD` or `IGS`), and `wwwwd` stands for the GPS week and the day of week to be processed. For a successful PPP it is crucial to use *fully consistent* GNSS satellite orbit, clock corrections, and ERP products.

- **Bernese formatted products files** (mandatory)

  ```
  ${D}/BSW52/P1C1yymm.DCB
  ```

  The DCB corrections are necessary to unify the reference for the GNSS satellite clock corrections if the LEO receiver does not track P–Code on both frequencies; `yymm` is the 2-digit year and the 2-digit month of the processing session.

  The files can be downloaded from `http://www.aiub.unibe.ch/download/CODE/`.

- **General files, which are necessary for the processing**

  ```
  ${D}/{V_REFDIR}/{V_CRDINF}.CRD
  ${D}/{V_REFDIR}/{V_CRDINF}.ABB
  ```

  Even if the a priori positions of the LEO are not taken from the coordinate file, they must be listed there for technical reasons with any coordinate values (e.g., `(0,0,0)`). To create the Bernese observation filenames, the LEO must be listed in the abbreviation table together with a unique 4– and 2–character abbreviation.

- **Station naming/equipment checking** (mandatory)
  The station information file

  ```
  ${D}/{V_REFDIR}/{V_CRDINF}.STA
  ```

  is mandatory for LEO processing, because the entry `SPACEBORNE` as the `MARKER TYPE` in section `TYPE 005:  HANDLING STATION TYPES` is needed for the processing programs.

- **Tidal loading corrections** (optional)
  Ocean and atmospheric tidal loading files contain corresponding CMC values for the transformation of the orbit positions from the Earth-fixed system into the center–of–mass system for the orbit integration:

  ```
  ${D}/{V_REFDIR}/{V_BLQINF}.BLQ
  ${D}/{V_REFDIR}/{V_BLQINF}.ATL
  ```

23.4.5.3. Observation Selection

The GPS observation files are expected in RINEX format (Hatanaka compressed is allowed if the decompression program is available in `${X}/EXE/CRX2RNX` — see variable `$c2rPgm` definition in `${BPE}/bpe_util.pm`) in the directory

```
${D}/{V_LEODIR}/RINEX/
```

The expected name for the RINEX files is `{V_LEO}ssss.yyO` (or with the extension `.yyD` for Hatanaka compressed files), where `ssss` denotes the 4-character session identifier and `yy` the 2-digit year.

The corresponding attitude files of the LEO are stored in the directory

```
${D}/{V_LEODIR}/ATTIT/
```

The attitude files are assumed following the naming convention `{V_ATTINF}.ATT`. If no attitude file is available, `V_ATTINF` may also be empty. The attitude files are used by the programs MAUPRP, GPSEST, and CODSPP.

23.4.5.4. Result Files

The most important result files are copied by the BPE user script LEO_SAV into the ${S}/{V_RESULT} area (V_RESULT is defined as BPE server variable, see Section 23.4.5.5) if the BPE server variable V_SAV == 'Y'. The files are located in yearly subdirectories (yyyy). These files are:

- ${S}/{V_RESULT}/yyyy/OUT/{V_LEO}{V_F}yyssss.PRC
  Main protocol file containing the summary of the processing steps (more details are provided in Section 23.4.5.7).
- ${S}/{V_RESULT}/yyyy/OUT/RD{V_F}yyssss.RES
  ${S}/{V_RESULT}/yyyy/OUT/KN{V_F}yyssss.RES
  Phase observation residuals from the reduced–dynamic (RD) and kinematic (KN) orbit determination.
- ${S}/{V_RESULT}/yyyy/OUT/KN{V_F}yyssss.CLK.gz
  Clock RINEX file (gzip compressed) with the receiver clock corrections for the LEO from the kinematic positioning together with the used satellite clock corrections.
- ${S}/{V_RESULT}/yyyy/STA/KN{V_F}yyssss.KIN
  Resulting kinematic positions of the LEO .
- ${S}/{V_RESULT}/yyyy/ORB/RD{V_F}yyssss.PRE
  ${S}/{V_RESULT}/yyyy/ORB/RD{V_F}yyssss.ELE
  ${S}/{V_RESULT}/yyyy/ORB/RD{V_F}yyssss.STD
  Result files of the reduced–dynamic orbit determination, PRE file with Earth-fixed positions and velocities (SP3c format), ELE file with orbital elements and values of the empirical parameters, STD file with orbit in inertial system (binary file).
- ${S}/{V_RESULT}/yyyy/ORB/KN{V_F}yyssss.PRE
  Result file of the kinematic orbit determination, PRE file with Earth-fixed positions (SP3c format).

23.4.5.5. BPE Server Variables

There exist several variables having a default value in the PCF (last section of the file). When starting a BPE manually they may be modified in the fourth panel "RUNBPE 4: Server Variables Sent to Client".

Selecting the LEO to be Processed

The LEO satellite to be processed is selected by a BPE server variable:

V_LEO (default: GRCA)
Name (4–character ID) of the LEO to be processed. The RINEX and attitude files must have this 4–character ID at the beginning of the name. This name is also contained in the processing summary filename of the BPE run.

Input and Result Filenames and Their Locations

The basic files in Bernese formats and their location in the datapool needed for the data analysis are defined using the following BPE server variables:

V_REFDIR (default: REF52)
Directory in the `DATAPOOL` area (`${D}`) where the basic Bernese files for the processing are located. The detailed list of files is given in Section 23.4.5.2 .

V_CRDINF (default: EXP_LEO)
The basename of the files used as basic Bernese input files (`CRD`, `ABB`, and `STA`) for this LEO POD procedure. Copying these files is managed in the BPE user script `LEO_COP` .

V_BLQINF (default: EXAMPLE)
Name for the file containing CMC values related to the ocean tidal loading. These corrections are used in the orbit generation.

V_ATLINF (default: EXAMPLE)
Name for the file containing CMC values related to the atmospheric tidal loading. These corrections are used in the orbit generation.

V_LEODIR (default: LEO)
Directory with LEO data files in the `DATAPOOL` area (`${D}`). A detailed description of the structure and the necessary files is given in the Section 23.4.5.3. Copying these files is managed in the BPE user script `LEO_RNX` .

V_ATTINF (default: GRCA$YSS+0)
Name of the corresponding attitude file (file name: (`{V_ATTINF}.ATT`). If empty, no external attitude information will be used.

V_RESULT (default: LEOPOD)
Directory in the `SAVEDISK` area (`${S}`) where the result files from the LEO POD processing are collected in yearly subdirectories. It is recommended to include the name of the processed LEO in the archive name. Saving of the result files is managed by the BPE user script `LEO_SAV`.

Observation Selection

The following variables are responsible for the data selection during the processing:

V_OBSINF (default: OBS.SEL)
This variable contains the filename of the receiver type related observation types priority list which is used when importing `RINEX3` data files, located in the the `${X}/GEN/` directory. The observation selection is done in RNXSMT. More details are given in Section 24.3.7 .

V_SAMPL (default: 30)
Sampling of the LEO GPS observations in seconds.
Many LEO missions provide a higher sampling than 30 sec. If you would like to use a higher sampling, e.g., 10 sec, you should have GPS clock corrections available with this sampling (e.g., with a sampling of 5 seconds from the CODE final processing). You may also interpolate the clock corrections (corresponding option in CODSPP,

MAUPRP, and GPSEST), but the gain in the number of observations does not help then, because the observation noise grows significantly due to the interpolation of the clocks (see Bock et al. 2009).

V_MINEL (default: 5)
Elevation cut–off angle in degrees for observations.
Many LEO missions provide observations down to 0 degree elevation, but you have to be sure that these observations are not of degraded quality. You should only use the observations down to the elevation angle where phase center variations (PCV) corrections are available (if you have any) for the LEO antenna.

General Model and Standard File Selection

In this sequence the BPE server variables related to the general processing files are listed:

V_PCV (default: I08)
Selection of the antenna phase center model identifier.

V_SATINF (default: SATELLIT)
Basename of the satellite information file. The resulting filename is `${X}/GEN/{V_SATINF}.{V_PCV}`

V_PCVINF (default: PCV)
Basename of the antenna phase center correction file. The resulting filename is `${X}/GEN/{V_PCVINF}.{V_PCV}`

V_SATCRX (default: SAT_$Y+0)
Name of the satellite problem file.

V_RECINF (default: RECEIVER.)
This variable contains the filename of the receiver characterization file located in the the `${X}/GEN/` directory. More details are given in Section 24.3.6 .

File and Product Naming

The following group of BPE server variables are used as product and solution identifiers and are consequently used for file naming within the BPE processing:

V_B (default: COD)
Name of the external product for GNSS orbit, satellite clock, and ERP information. It is expected that the products are available in the `DATAPOOL` area in a specific subdirectory `V_B`; the filenames must also contain this label: `{V_B}wwwwd` and `{V_B}wwww7` (see Section 23.4.5.2 for more details).

V_A (default: A)
Identifier for the a priori orbit and data screening files.

V_F (default: AF)
Identifier for the final orbit solution of the LEO. The result files of the reduced–dynamic and kinematic orbit generation use this identifier.

Management of the Dataflow

The last group is related to the file handling in the campaign

V_SAV (default: Y)
Save result file into the `${S}/{V_RESULT}` area if V_SAV == 'Y'. This variable is managed by the BPE user script `LEO_SAV`.

V_SAVOBS (default: Y)
If V_SAVOBS == 'Y' the code and phase zero–difference observation files from the current session are copied into session–specific directories of the `SAVEDISK` area: `${S}/{V_RESULT}/yyyy/OBS/yyssss`.

This variable is managed by the BPE user script `OBS_SAV`.

V_DEL (default: Y)
Delete all files of the currently processed session from the campaign if V_DEL == 'Y'; managed in `LEO_DEL` user script. Usually the deletion should be enabled to keep the campaign clean. For debugging purposes and to identify sources of errors in the processing it may be useful to keep all files.

Skip Sessions in Case of Reprocessing

In case of a reprocessing it might be interesting to repeat only selected sessions for some reasons. A typical scenario could be that the BPE is executed once for all sessions. Possibly, some sessions will end with an error or with unsatisfying results. After fixing these problems, the reprocessing needs to be repeated only for a certain subset of sessions. Instead of starting the BPE manually for each of these sessions, it is more convenient to start the BPE again for the full interval of the reprocessing and let the availability of a specific file in the `SAVEDISK` area indicate whether a certain session shall be repeated or not.

V_SKIP (default: $(LEO)$(F)$YSS+0.PRC; deactivated $(LEO)$(F)$YSS+0.PRC_)
The very first script `SKIP_SES` (PID 000) checks whether the file `${S}/{V_RESULT}/yyyy/OUT/{V_SKIP}` exists, e.g., from a previous iteration for processing the sessions of a certain interval. The availability of the file indicates the successful processing of the session in a previous iteration of the reprocessing. That's why the BPE will directly jump to the last script (`DUMMY`; PID 999). If the indicated file does not exist, the BPE will process all relevant scripts for the specific session.

We propose to use the protocol file in the `SAVEDISK` area as an indicator.

If a session shall be repeated even if it was executed up to the last script in a previous iteration, the indicator file (given in the `V_SKIP` variable) must be removed in the `SAVEDISK` area.

If all sessions shall be repeated independent from the existence of the indicator file the value for this variable can be deleted or the filename must at least be modified, e.g., from `$(LEO)$(F)$YSS+0.PRC` to `$(LEO)$(F)$YSS+0.PRC_`.

23.4.5.6. Description of the PCF

Copy Required Files

```
#
# Skip session (reprocessing)
# ---------------------------
000 SKIP_SES NO_OPT            ANY      1
#
# Copy required files
# -------------------
001 LEO_COP  NO_OPT            ANY      1 000
```

PID 000 SKIP_SES: This script checks whether the file `${S}/{V_RESULT}/yyyy/OUT/{V_SKIP}` exists. The values for `V_RESULT` and `V_SKIP` are taken from the BPE server variables whereas `yyyy` is related to the year of the current session. If the file exists the script directly jumps to the last script of the BPE. The mechanism may be used, e.g., to manage whether a specific session shall be repeated in the frame of a reprocessing or not.

PID 001 LEO_COP: This script copies all necessary files into the respective campaign directories (the list of files is given in Section 23.4.5.2). If appropriate, file names will be changed according to the BPE variables and/or a session–specific date.

The BPE will stop with an error if one of the requested files is missing.

Further reading:

Section 3.1: "Overview of the Directory Structure", page 57
Section 22.5.3: "Loops and Jumps", page 491

Prepare Pole, Orbit, and Clock Information

```
#
# Prepare pole, orbit, and clock information
# ------------------------------------------
051 POLUPD   LPD_GEN           ANY      1 001
061 PRETAB   LPD_GEN           ANY      1 051
062 ORBGEN   LPD_GEN           ANY      1 061
071 RNXCLK   LPD_GEN           ANY      1 002
```

PID 051 POLUPD: Program POLUPD converts the pole file from IGS/IERS format to Bernese format. This pole file will be used for all following processing steps where EOPs are necessary.

The script checks the content of the input pole file and tries to find out the correct setting for option "Use ERP rates". Because of the diversity of IERS/IGS formatted pole files it might happen that the algorithm fails. In this case you have to define the option manually in the user script by replacing XYZ by the source of your products (value of BPE server variable `V_B`):

```
# Repeat the program if necessary
# ------------------------------
  if ( $v_b eq "XYZ" ) { $doRate = 1 }  # if products from "XYZ" are used
                                        # apply the polar rates.
  if ( $doRate != $isRate ) {
    $bpe->putKey("$ENV{U}/PAN/POLUPD.INP","USERATE","$doRate");

    my $PGMNAM = "POLUPD";
    $bpe->RUN_PGMS($PGMNAM);
  }
```

PID 061 PRETAB: The GNSS orbit files are available in the precise orbit file format (SP3c), containing a table of satellite positions in an Earth-fixed frame (e.g., IGb 08). These positions have to be converted into the inertial frame using the program PRETAB.

PID 062 ORBGEN: The tabular orbit file is used as input for the final GNSS orbit generation step with ORBGEN. The result is a standard orbit file and a summary providing the quality of the orbit fit (`ORByyssss.LST`). It is contained in the processing summary.

PID 071 RNXCLK: The precise GNSS satellite clock information is converted from the external clock RINEX file into the Bernese format that is located in the `ORB` directory of the campaign.

Further reading:

Section 4.3: "Precise Orbit Files", page 90
Section 4.4: "IGS and IERS Pole Files", page 91
Chapter 5: "Preparation of Earth Orientation, GNSS Orbit, and Satellite Clock Information", page 107

LEO RINEX Import

```
#
# LEO RINEX import
# -----------------------------------------
080 LEO_RNX  NO_OPT            ANY      1 000
090 RNXSMT   LPD_APR           ANY      1 001 080
100 RXOBV3   LPD_APR           ANY      1 090
```

PID 080 LEO_RNX: This script copies the RINEX and attitude files from the `RINEX` and `ATTIT` subdirectories of `${D}/{V_LEODIR}` into the campaign. Hatanaka decompression of the RINEX files is performed if necessary.

The file names are labeled by the BPE variable `V_LEO`. If no RINEX files are found for the current session, the script directly jumps to PID 999. This allows to run through a range of sessions independent of whether data for all sessions are available or not.

PID 090 RNXSMT: Program RNXSMT is called to prepare potential RINEX (version 3) data, according to the observation priority list given in `V_OBSINF`, to be converted into Bernese files in the next step PID 100.

PID 100 RXOBV3: The program RXOBV3 creates Bernese observation files for code and phase from the smoothed RINEX files.

Further reading:

Section 22.5.3: "Loops and Jumps", page 491
Section 4.2.3: "Import to Bernese", page 79

First A Priori Orbit Generation for LEO

```
#
# First a priori orbit generation for LEO
# -----------------------------------------
110 CODSPP   LPD_APR           ANY      1 062 071 100
120 KINPRE   LPD_APR           ANY      1 110
130 ORBAPR   LPD_APR           ANY      1 120
```

PID 110 CODSPP: Based on the code observations, the program CODSPP computes here a first coarse kinematic orbit. Furthermore, the receiver clock is synchronized with GPS time. At this stage, no attitude information is used. Notice that no a priori orbit is needed for the estimation of this kinematic orbit.

PID 120 KINPRE: The program KINPRE converts the kinematic positions and the clock corrections (computed by CODSPP of PID 110) into SP3c format. The resulting file is called `{V_A}yyssss.PRE`.

PID 130 ORBAPR: The program ORBGEN numerically integrates the equations of motion to fit the kinematic LEO positions in `{V_A}yyssss.PRE` into a standard orbit (`{V_A}yyssss.STD`). The orbit is parameterized by six osculating elements and constant and once–per–revolution empirical accelerations in radial, along–track, and cross–track directions. Apart from the equations of motion, ORBGEN also numerically integrates the variational equations to provide the partial derivatives of the orbit w.r.t. the orbit parameters (written into the file `{V_A}yyssss.RPR`). They are needed for the orbit improvement during the data screening.

Further reading:

Section 6.3: "Receiver Clock Synchronization and Preprocessing of Code Observations", page 135
Section 5.4.3: "Generation of Standard Orbit Information", page 115
Section 16.3.1: "The Procedure for Orbit Improvement in the *Bernese GNSS Software*", page 378

Data Screening

```
#
# Data screening
# -------------------------------------------
140 MPGPUP   LPD_MPG           ANY      1 130
150 GPSEST   LPD_GRS           ANY      1 140
160 RESRMS   LPD_GRS           ANY      1 150
170 SATMRK   LPD_GRS           ANY      1 160
200 CODSPP   LPD_INT           ANY      1 170
```

PID 140 MPGPUP: This script calls the programs MAUPRP, GPSEST and ORBGEN successively and in a loop of three iterations:

- MAUPRP: The phase observations are screened with screening parameters (a priori coordinate sigmas and the sigmas for the L_1– and L_2–observations in the no cycle slip hypothesis) which become more and more stringent in each screening iteration. In the first iteration, the a priori standard orbit of PID 130 is introduced. For the further iterations, the standard orbit resulting from the previous screening iteration is used.
- GPSEST: An orbit is estimated from the screened phase observations, using the ionosphere–free linear combination. The orbit resulting from PID 130 is used as a priori orbit (in each iteration). Apart from the six osculating elements the orbit is parameterized by constant and once–per–revolution empirical accelerations in radial, along–track, and cross–track directions, as well as constrained pseudo–stochastic pulses every 15 minutes and in the same three directions. The spacing and the constraints of the pseudo–stochastic pulses can be adjusted in panel "GPSEST 6.4.2: LEO Orbit Determination 2". The result of this GPSEST run

is an ELE file containing the improved orbit parameters. After the GPSEST run the program GPSXTR is executed in order to create a summary of the GPSEST output file. This summary is then displayed in the main processing protocol.

Note when processing your own data: The pseudo–stochastic pulses in "GPSEST 6.4.2: LEO Orbit Determination 2" are only set up for the LEO satellites GRACE–A and GRACE–B . If you are processing data from another LEO satellite, you have to adjust the corresponding option.

- ORBGEN: Using the orbit parameters in the ELE file of the previous GPSEST run, a standard orbit is numerically integrated.

This iterative screening is necessary because the first a priori orbit is not yet good enough for a reliable data screening in MAUPRP. With each iteration step the quality of the improved orbit increases and the screening parameters can be set tighter.

PID 150 GPSEST: The program GPSEST is called to compute ionosphere–free phase residuals when using the screened observation files and the standard orbit of the last screening iteration. They are used for the subsequent residual screening.

PID 160 RESRMS: The program RESRMS reads the residual file of the previous GPSEST run and writes observations with residuals larger than 4 cm into an edit information file. This threshold is specified in panel "RESRMS 2: Options" and may need to be adapted depending on the GPS data quality.

PID 170 SATMRK: The program SATMRK marks all the observations in the phase observation file which were screened out and written into the edit information file by RESRMS of PID 160. The marked observations will not be used for the final orbit determination.

PID 200 CODSPP: Based on the better standard orbit of the last screening iteration and introducing the LEO attitude information, the receiver clocks are synchronized with GPS time a second time using program CODSPP.

Further reading:

Section 6.5: "Preprocessing Phase Observations", page 144
Section 6.5.6: "Screening of LEO Data", page 151
Chapter 7: "Parameter Estimation", page 169
Section 7.9.2: "The Output Extraction Program GPSXTR", page 189
Section 5.4.3: "Generation of Standard Orbit Information", page 115
Section 6.6.2: "Generating Residual Statistic", page 162
Section 6.6: "Screening of Post–Fit Residuals", page 161
Section 6.7: "Marking of Observations", page 167
Section 6.3: "Receiver Clock Synchronization and Preprocessing of Code Observations", page 135

A Priori Orbit for Final Orbit Determination

```
#
# A priori orbit for final orbit determination
# -----------------------------------------
400 STDPRE   LPD_INT          ANY      1 140
410 ORBAPF   LPD_INT          ANY      1 400
```

PID 400 STDPRE: The program STDPRE converts the standard orbit of the last screening iteration into a precise orbit file in SP3c format.

PID 410 ORBAPF: Using program ORBGEN, the positions in the SP3c file created by program STDPRE of PID 400 are fit into a standard orbit again (`{V_A}_yyssss.STD`). Apart from the six osculating elements only constant empirical accelerations in radial, along–track, and cross–track directions are used as orbit parameters. This allows to parameterize the final reduced–dynamic orbit completely independently from the orbit parametrization used for the data screening in PID 140. The partial derivatives of the orbit w.r.t. the orbit parameters are computed and written into the file `{V_A}_yyssss.RPR` as well. The orbit and the partial derivatives will be used as a priori information for the following final reduced–dynamic orbit determination.

Further reading:

Section 16.2.3: "Conversion of Standard Orbit Files to Precise Orbit Files", page 377
Section 5.4.3: "Generation of Standard Orbit Information", page 115
Section 16.3.1: "The Procedure for Orbit Improvement in the *Bernese GNSS Software*", page 378

Final LEO POD

```
#
# Final LEO POD
# -------------
# Reduced-dynamic orbit
# -----------------------------------------
500 GPSEST   LPD_FIN           ANY      1 200 410
501 GPSXTR   LPD_FIN           ANY      1 500
502 ORBGEN   LPD_FIN           ANY      1 501
503 STDPRE   LPD_FIN           ANY      1 502
511 GPSEST   LPD_RES           ANY      1 503
512 RESRMS   LPD_RES           ANY      1 511
```

PID 500 GPSEST: At this step the program GPSEST estimates the reduced–dynamic orbit using the screened phase observations. The a priori standard orbit and the partial derivatives of PID 410 are introduced. The six osculating elements, constant empirical accelerations in radial, along–track, and cross–track directions, as well as constrained 6 minutes piecewise constant empirical accelerations in the same directions are set up. The once–per–revolution empirical accelerations are not set up, since, together with the piecewise constant accelerations, they would lead to an overparametrization. The settings for the piecewise constant accelerations can be adjusted in panel "GPSEST 6.4.2: LEO Orbit Determination 2". It is important to pay attention that the interval between subsequent stochastic parameters is a multiple of the integration step size for the eqation of motion specified in ORBGEN of PID 410. The result of this GPSEST run is the file `RD{V_F}yyssss.ELE`, which contains the improved orbital parameters. Since carrier phase ambiguities are pre–eliminated as soon as possible, no observation residuals can be computed at this step. To get the residuals there is an additional GPSEST run in PID 511.

Note when processing your own data: The pseudo–stochastic pulses in "GPSEST 6.4.2: LEO Orbit Determination 2" are only set up for the LEO satellites GRACE–A and GRACE–B . If you are processing data from another LEO satellite, you have to adjust the corresponding option.

PID 501 GPSXTR: The program GPSXTR reads the output file of the previous GPSEST and writes a summary. This will be included in the main processing protocol.

PID 502 ORBGEN: Using the improved orbit parameters in the ELE file of the GPSEST run of PID 500, the final reduced–dynamic standard orbit is numerically integrated. The resulting file is named RD{V_F}yyssss.STD.

PID 503 STDPRE: The program STDPRE converts the standard orbit into the precise orbit format, containing Earth-fixed positions of the LEO. The resulting precise orbit file is named RD{V_F}yyssss.PRE.

PID 511 GPSEST: Using the screened observation files and the final reduced–dynamic standard orbit, this GPSEST computes the phase residuals of the orbit. They are written into the residual file RD{V_F}yyssss.RES.

PID 512 RESRMS: The program RESRMS reads the residual file generated by GPSEST of PID 511 and writes a summary which will be included in the final processing protocol.

```
#
# Kinematic orbit
# -------------------------------------------
600 GPSEST   LPD_KIN           ANY      1 503
601 GPSXTR   LPD_KIN           ANY      1 600
602 KINPRE   LPD_KIN           ANY      1 600
612 RESRMS   LPD_KIN           ANY      1 601 602
```

PID 600 GPSEST: In this step the program GPSEST estimates the kinematic orbit using the screened observation files and the final reduced–dynamic orbit as a prioi information. Kinematic coordinates and receiver clock corrections are estimated and pre–eliminated every epoch. The coordinates are written into the file KN{V_F}yyssss.KIN and the clock corrections into the clock RINEX file KN{V_F}yyssss.CLK. In addition, the phase residuals are written into the residual file KN{V_F}yyssss.RES (no parameters apart from the epoch–specific parameters are pre–eliminated). The minimum numbers of observations required for a clock and coordinate estimation are specified in the panels "GPSEST 6.8.1: Clock Estimation 1" and "GPSEST 6.9: Kinematic Coordinates". By default, they are both set to 5. If you change them, make sure that they are set to the same value, because otherwise it may happen that for a given epoch, kinematic positions but no receiver clock correction are available, or vice–versa.

PID 601 GPSXTR: The program GPSXTR reads the output file of the previous GPSEST run and writes a summary which will be included in the final processing protocol.

PID 602 KINPRE: The program KINPRE converts the kinematic coordinates and the receiver clock corrections of the kinematic orbit into SP3c format. The resulting precise orbit file is called KN{V_F}yyssss.PRE.

PID 612 RESRMS: The program RESRMS reads the residual file generated by GPSEST of PID 600 and writes a summary which will be included in the final processing protocol.

```
#
# Comparison of orbits
# -------------------------------------------
700 ORBCMP   LPD_CMP           ANY      1 600 503 602
```

PID 700 ORBCMP: As the last processing step, the program ORBCMP computes the differences between the reduced–dynamic orbit (RD{V_F}yyssss.PRE) and the kinematic orbit (KN{V_F}yyssss.PRE). The coordinate system for the comparison is specified in panel "ORBCMP 2: General Options" and is by default set to RSW (radial, along–track, cross–track). The velocity information needed for the definition of this system is read from the reduced–dynamic standard orbit (RD{V_F}yyssss.STD), which is read by ORBCMP as well. The orbit differences are written to the file CP{V_F}yyssss.PLT.

Further reading:

Section 16.3.1: "The Procedure for Orbit Improvement in the *Bernese GNSS Software*", page 378
Section 7.7.1: "Pre–Elimination of Parameters", page 182
Section 17.5: "LEO Data Processing", page 394
Section 7.9.2: "The Output Extraction Program GPSXTR", page 189
Section 16.2.3: "Conversion of Standard Orbit Files to Precise Orbit Files", page 377
Section 6.6.2: "Generating Residual Statistic", page 162
Section 5.5.2: "Comparison of Satellite Orbits", page 124

Create Summary File, Save and Delete Files

```
#
# Create summary file, save and delete files
# ------------------------------------------
901 LEO_SUM  LPD_GEN           ANY      1 700 512 612
902 LEO_SAV  LPD_GEN           ANY      1 901
903 OBS_SAV  LPD_GEN           ANY      1 901
904 LEO_DEL  LPD_GEN           ANY      1 902 903
991 BPE_CLN  LPD_GEN           ANY      1 904
#
# End of BPE
# ----------
999 DUMMY    NO_OPT            ANY      1 991
```

PID 901 LEO_SUM: All old protocol files of the processed session are deleted. The comprehensive summary is composed of several parts describing the different steps. The protocol file is discussed in Section 23.4.5.7 .

PID 902 LEO_SAV: The result files from this BPE (listed in Section 23.4.5.4) are copied from the campaign into the SAVEDISK area, see description of BPE server variable V_RESULT.

PID 903 OBS_SAV: The observation files from this BPE are copied from the campaign into the SAVEDISK area, see description of BPE server variable V_SAVOBS.

PID 904 LEO_DEL: This script deletes all files which are no longer needed after the end of the BPE. This prevents unimportant files from accumulating.

PID 991 BPE_CLN: Deletes files from the campaign's BPE–subdirectory with a delay of 30 sessions. Thus, all BPE protocol and log files of the last 30 sessions will remain in the directory.

PID 999 DUMMY: Does nothing. This script provides a well defined PID as end point, useful, e.g., as jump address, or to check for the completion of the BPE.

If you want to archive other or more files you should not remove these files from the list in the deletion script (LEO_DEL), but add them to the list of files to be copied to the SAVEDISK area (LEO_SAV).

The DUMMY–script may seem useless at a first glance. But if you have such a script running at a well defined PID (e.g., PID 999) in all your PCFs the successful execution of a BPE can be tested in a generic way by checking the existence of the DUMMY–script's LOG– or PRT–file. In this way a script starting a BPE can check the successful completion of the BPE and react in case of errors (e.g., send an error mail).

Further reading:

Section 22.10: "BPE Output and Protocol Files", page 518
Section 22.11: "Error Handling", page 522

23.4.5.7. Description of the Protocol File, Quality Control

The processing is summarized in a so–called protocol file. It is compiled in the BPE user script LEO_SUM and exists first in the OUT directory of the campaign. Depending on the settings for the BPE server variables, this protocol file is copied by the LEO_SAV script (if V_SAV == 'Y') into ${S}/{V_RESULT}/yyyy/OUT/{V_LEO}{V_F}yyssss.PRC and is removed from the campaign area by the script LEO_DEL (if V_DEL == 'Y').

At the beginning of the processing summary the files that have been used for the data analysis are reported:

```
===============================================================================
LEOPOD BPE PROCESSING SUMMARY FOR GRCA YEAR-SESSION 10-2070
===============================================================================


Summary file generated at 13-Dec-2012 08:38:55 by LEO_SUM

General files:
   Antenna phase center eccentricity file:   PCV.I08
   Satellite information file:               SATELLIT.I08
   Satellite problem file:                   SAT_2010.CRX
   Receiver characterization file:           RECEIVER.
   Orbit, ERP and clock products used from:  ${D}/COD

...
```

Then, the sampling rate, the elevation cut–off angle and information about the empirical parameters, as well as a list of stored result files are listed.

The part reporting the processing results is divided into sections related to the different tasks/features of the BPE:

PART 0: GNSS ORBIT GENERATION SUMMARY

The summary file of ORBGEN reports the RMS of the fit of the satellite position given in the precise orbit file ${D}/{V_B}/{V_B}wwwwd.PRE with the orbit model from ORBGEN. It is expected that the RMS is on the few millimeters level. If the RMS is larger, you should check the consistency of the Earth rotation files ${D}/{V_B}/{V_B}wwww7.ERP.

```
===============================================================================
PART 0: GNSS ORBIT GENERATION SUMMARY
===============================================================================

 LEOPOD_102070: Generate GNSS standard orbit                      13-DEC-12 08:27
 TIME FROM DAY : 1 GPS WEEK: 1594
      TO   DAY : 2 GPS WEEK: 1594
 ------------------------------------------------------------------------------
 ORBIT REPEATABILITY FROM A 1-DAY FIT THROUGH DAILY ORBIT SOLUTIONS (MM)
 # ECLIPSING SATELLITES:   5 E /   0 M (  0 EM)
 -------------------------------------------------------------------------------------- ...
 ECL   ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  E.  E.  E.  ..  ..  ..  ..  ..  ..   ...
 DOY    1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21   ...
 -------------------------------------------------------------------------------------- ...
 207    1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   ...
 -------------------------------------------------------------------------------------- ...
 ALL    1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   ...
 -------------------------------------------------------------------------------------- ...
```

It might also be necessary to split the orbital arc into two or three arcs (option "Number of arcs within the time window" in panel "ORBGEN 5: Orbital Arc Definition" of the ORBGEN input file). It is essential that the RMS of the orbit fit can be reduced to the few millimeters level since the satellite clock corrections for the PPP (in this case the special application on LEO POD) refer to the original position of the satellite in the given precise orbit file.

Since CODE has improved its solar radiation pressure model for the GNSS satellites in its operational solution from January 2015 onwards, an update of the software (published as B_049) is needed to allow the fit of the precise orbit files with the expected quality. After updating the software you may choose the new model by selecting "dynamical orbit parameters" as "System D2X" in panel "ORBGEN 3.1: Options".

Further reading:

Section 4.3: "Precise Orbit Files", page 90
Chapter 5: "Preparation of Earth Orientation, GNSS Orbit, and Satellite Clock Information", page 107

PART 1: RINEX CONSISTENCY CHECK

Inconsistencies between the RINEX file from the LEO and the station information file are listed here. Unfortunately, this list is quite long for most of the LEOs . The reason is that no convention exists on how the LEO receivers and antennas should be named. Additionally, in most cases, the `MARKER TYPE` entry `SPACEBORNE` is missing. This description is, however, mandatory for the LEO processing in the *Bernese GNSS Software* to distinguish between spaceborne and terrestrial receivers.

Further reading:

Section 4.2.3: "Import to Bernese", page 79
Section 4.2.3.4: "Verification of RINEX Header Information", page 83

PART 2: A PRIORI ORBIT GENERATION SUMMARY

The summary file of ORBGEN reports the RMS of the fit of the satellite position given in the precise orbit file `{V_A}yyssss.PRE` with the orbit model from ORBGEN. The positions in the precise orbit file are kinematic positions derived from code observations of the LEO .

```
===============================================================================
PART 2: GRCA A PRIORI ORBIT GENERATION SUMMARY
===============================================================================


 ------------------------------------------------------------------------------
 ORBIT REPEATABILITY FROM A 1-DAY FIT THROUGH DAILY ORBIT SOLUTIONS (CM)
 ------- ----------------------------------------------------------------------
     909E
 ------------------------------------------------------------------------------
 207 188
 ------------------------------------------------------------------------------
 ALL 188
 ------------------------------------------------------------------------------
```

It is expected that the RMS is on the few meters level. This depends on the code observation quality and whether the orbit can be fitted well without any empirical accelerations.

Further reading:

Section 5.4.3: "Generation of Standard Orbit Information", page 115

PART 3: DATA SCREENING SUMMARY (A PRIORI ORBIT IMPROVEMENT WITH GPSEST)

The data screening is done iteratively, because the first a priori orbit is not yet good enough for a reliable screening result in MAUPRP. Therefore, an orbit improvement step including GPSEST and ORBGEN is done. This loop is iterated three times to be sure that the screening is sufficient. In this part of the protocol, the GPSEST summaries of these three iterations are listed.

```
================================================================================
PART 3: DATA SCREENING SUMMARY (A PRIORI ORBIT IMPROVEMENT WITH GPSEST)
================================================================================

 A1102070.OUT     Rms:  2.3 , # fil.:   1 , # obs.:   23297 , # par.:  3601

 A2102070.OUT     Rms:  2.2 , # fil.:   1 , # obs.:   23267 , # par.:  3600

 A3102070.OUT     Rms:  2.2 , # fil.:   1 , # obs.:   23268 , # par.:  3600
```

If the RMS of the individual iterations are much higher, it may be caused by missing pseudo–stochastic pulses in GPSEST ("GPSEST 6.4.2: LEO Orbit Determination 2") in PID 140. Only pulses for GRACE–A and GRACE–B are by default defined.

Further reading:

Section 7.9.2: "The Output Extraction Program GPSXTR", page 189

PART 4: A PRIORI ORBIT SUMMARY BEFORE FINAL ORBIT DETERMINATION

The summary file of ORBGEN reports the RMS of the fit of the satellite position given in the precise orbit file {V_A}_yyssss.PRE with the orbit model from ORBGEN. The positions in the precise orbit file are extracted from the standard orbit file (extension STD), which was derived from the iterative data screening plus orbit improvement step. This standard orbit contains pseudo–stochastic pulses every 15 minutes. The final reduced–dynamic orbit will be generated with piecewise constant accelerations every 6 minutes. Due to the mixture of empirical parameters and the different sampling, you have to generate a new standard orbit file without pulses with ORBGEN.

```
================================================================================
PART 4: GRCA A PRIORI ORBIT SUMMARY BEFORE FINAL ORBIT DETERMINATION
================================================================================

 -------------------------------------------------------------------------------
 ORBIT REPEATABILITY FROM A 1-DAY FIT THROUGH DAILY ORBIT SOLUTIONS (CM)
 -------------------------------------------------------------------------------
     909E
 -------------------------------------------------------------------------------
 207  91
 -------------------------------------------------------------------------------
 ALL  91
 -------------------------------------------------------------------------------
```

It is expected that the RMS is on the meter level. This depends on how well the orbit of the particular LEO can be fitted without setting up any stochastic pulses.

Further reading:

Section 16.2.3: “Conversion of Standard Orbit Files to Precise Orbit Files”, page 377
Section 5.4.3: “Generation of Standard Orbit Information”, page 115

PART 5: REDUCED-DYNAMIC ORBIT DETERMINATION

Part of the GPSEST output is listed in the processing summary file. In addition, an RESRMS summary of the reduced–dynamic orbit generation is listed to validate the quality of the observations. One may detect GPS satellites with problems here.

```
================================================================================
PART 5: REDUCED-DYNAMIC ORBIT DETERMINATION
================================================================================

 PARAMETER TYPE              #PARAMETERS  #PRE-ELIMINATED      #SET-UP  #NO-OBS  #REF #SINGULAR
 -----------------------------------------------------------------------------------------------

 ORBITAL ELEMENTS                      9          0                   9        0      0      0
 AMBIGUITIES                         420        420  (BEFORE INV)   421        1      0      0
 STOCHASTIC ORBIT PARAMETERS         720          0                 720        0      0      0
 EPOCH WISE STATION CLOCKS          2880       2880  (EPOCH-WISE)  2880        0      0      0

 -----------------------------------------------------------------------------------------------
 TOTAL NUMBER OF PARAMETERS         4029       3300                4030        1      0      0
 -----------------------------------------------------------------------------------------------


 NUMBER OF OBSERVATIONS (PART 1):
 -------------------------------

 TYPE       FREQUENCY      FILE/PAR      #OBSERVATIONS
 -----------------------------------------------------------------------------------------------

 PHASE         L3           ALL               23252
 PSEUDO                     ALL                   1        (CURRENTLY ONLY REFERENCE AMBIGUITIES)

 -----------------------------------------------------------------------------------------------
 TOTAL NUMBER OF OBSERVATIONS                 23253
 -----------------------------------------------------------------------------------------------
```

If the stochastic orbit parameters are missing, then the pseudo–stochastic pulses might not be set up in GPSEST “GPSEST 6.4.2: LEO Orbit Determination 2” of PID 500. Only pulses for GRACE–A and GRACE–B are by default defined.

```
================================================================================
PART 5.1: REDUCED-DYNAMIC ORBIT DETERMINATION - RESIDUAL SUMMARY
================================================================================

 BASELINE  SESS     2     3     4     5     6     7     8     9    10    11    12   ...  TOT
 -------------------------------------------------------------------------------- ... ----
 GRCA      2070   5.5   5.4   6.2   5.4   5.6   6.1   6.3   4.9   5.4   7.0   5.4   ...  5.7
 -------------------------------------------------------------------------------- ... ----
 TOTAL RMS:       5.5   5.4   6.2   5.4   5.6   6.1   6.3   4.9   5.4   7.0   5.4   ...  5.7
 -------------------------------------------------------------------------------- ... ----
 TOTAL OBS:       2.7   3.7   3.0   3.6   4.3   2.8   2.9   2.6   3.8   3.1   3.3   ...
 -------------------------------------------------------------------------------- ... ----
 TOT OBS (1E0):   617   868   689   831   993   640   677   603   877   722   760
 -------------------------------------------------------------------------------- ... ----
```

Further reading:

Chapter 7: “Parameter Estimation”, page 169
Section 6.6.2: “Generating Residual Statistic”, page 162

PART 6: KINEMATIC ORBIT DETERMINATION

Part of the GPSEST output is listed in the processing summary file. In addition, an RESRMS summary of the kinematic orbit generation is listed to validate the quality of the observations. One may detect GPS satellites with problems here.

Further reading:

Chapter 7: “Parameter Estimation”, page 169
Section 6.6.2: “Generating Residual Statistic”, page 162

PART 7: ORBIT COMPARISON

The reduced–dynamic and kinematic orbit may be compared by the program ORBCMP. The comparison is done for the radial, along–track and out–of–plane directions. The summary of this comparison in terms of RMS, mean, minimum and maximum values is listed in this file. Normally, this comparison shows values in the range of a few centimeters.

```
================================================================================
PART 7: ORBIT COMPARISON
================================================================================

 Comparison of reduced-dynamic and kinematic orbit:
              radial   along-track  out-of-plane
 RMS          0.0234        0.0176        0.0149 m
 Mean        -0.0003       -0.0024        0.0086 m
 Min         -0.1099       -0.0664       -0.0316 m
 Max          0.1175        0.0511        0.0418 m
```

Further reading:

Section 5.5.2: “Comparison of Satellite Orbits”, page 124

23.4.6. Example for SLR Orbit Validation (SLRVAL.PCF)

23.4.6.1. Purpose and Description

This BPE performs a GNSS orbit validation using SLR tracking data. To obtain Bernese formatted observation files, the SLR NP files are first converted to the RINEX format (see Section 19.3.2). Then, the observations are converted from the RINEX format to the Bernese format (see Section 19.3.3). Based on the available GNSS orbit and ERP files, a standard obit file is created. GPSEST is used to compute the discrepancy between the GNSS orbits and the laser observations (note that no parameters are estimated). Residual statistics are created using the program QLRSUM; various outlier criteria can be defined.

23.4.6.2. Input Files

The input files are copied from the DATAPOOL area (${D}) into the campaign area in the BPE user script SLR_COP:

- **Orbit products files to be validated** (mandatory)

  ${D}/{V_O1}/{V_O1}wwwwd.PRE or ${D}/{V_O1}/{V_O1}wwwwd.STD
  ${D}/{V_O1}/{V_O1}wwww7.ERP

The variable `V_O1` defines the orbit to be validated with SLR observations (e.g., `COD`); `wwwwd` stands for the GPS week and the day of week to be processed. It is important to use fully consistent GNSS satellite orbit and ERP products.

- **Station–related reference files** (mandatory)
  The station coordinates for the SLR tracking stations are extracted from the files

  ```
  ${D}/{V_REFDIR}/{V_CRDINF}.CRD
  ${D}/{V_REFDIR}/{V_CRDINF}.VEL
  ```

  They are also used to propagate the coordinates to the epoch of measurements.

  The abbreviation table is needed for the creation of the observation files in the Bernese format:

  ```
  ${D}/{V_REFDIR}/{V_CRDINF}.ABB
  ```

  The station information file is only used to adjust the names of the SLR stations:

  ```
  ${D}/{V_REFDIR}/{V_CRDINF}.STA
  ```

- **Tidal loading corrections** (optional)
  Ocean tidal loading may be corrected during the SLR–based obit validation procedure if the corresponding CMC corrections are provided in the file

  ```
  ${D}/{V_REFDIR}/{V_BLQINF}.BLQ
  ```

- **SLR observation file** (mandatory)

  ```
  ${D}/{V_NPDIR}/{V_NP}.yymmdd
  ```

For the 4 days of the example (2 days in 2010, 2 days in 2011) the SLR NP files to GNSS satellites are available in the CSTG format. The string `yymmdd` holds the 2-digit year, the 2-digit month, and the 2-digit day.

23.4.6.3. Result Files

The only result file is the processing summary described in Section 23.4.6.6. It is copied by the BPE user script `SLR_SAV` into the `${S}/{V_RESULT}` area (`V_RESULT` is defined as BPE server variable, see Section 23.4.6.4) if the BPE server variable V_SAV == 'Y'. The files are located in yearly subdirectories (`yyyy`):

```
${S}/{V_RESULT}/yyyy/SLRVALyyddd.PRC
```

where `yy` and `ddd` are the 2-digit year and the 3-digit day of the year, respectively.

23.4.6.4. BPE Server Variables

There are several variables, assigned with a default value in the PCF (last section of the file). When starting a BPE manually, they may be modified in the fourth panel "RUNBPE 4: Server Variables Sent to Client".

Input and Result Filenames and Their Locations

The basic station–related files in the Bernese format and their location in the `DATAPOOL` area are defined using the following set of BPE server variables:

V_REFDIR (default: REF52)
Directory in the DATAPOOL area (${D}/{V_REFDIR}) containing the basic Bernese files that are copied within the BPE script SLR_COP.

V_CRDINF (default: EX52_SLR)
Station–related reference files copied from the DATAPOOL area (${D}) into the processing campaign. Copying these files is managed by the BPE user script SLR_COP. The station positions used as reference for computing the SLR residuals are extrapolated from {V_CRDINF}.CRD and {V_CRDINF}.VEL to the processing epoch by the BPE user script COOVEL.

V_BLQINF (default: EX52_SLR)
The basename of the file containing the ocean tidal loading corrections for the stations. The file also has to contain the CMC related to ocean tidal loading. These corrections are needed for orbit generation. If the variable is empty, no displacement corrections to the stations and no CMC for the orbit generation are applied. The file ${D}/{V_REFDIR}/{V_BLQINF}.BLQ is copied by the script SLR_COP into the campaign.

The location of the result files in the archive is defined using the BPE server variable:

V_RESULT (default: SLRVAL)
The result file is saved into the ${S}/{V_RESULT} area if V_SAV == 'Y'. The saving is managed by the BPE user script SLR_SAV.

Selecting Orbits to be Validated

The orbit to be validated with this procedure is defined by the following variables:

V_O1 (default: COD)
Orbit to be validated with SLR observations. The variable is used for the directory in the DATAPOOL area as well as for the naming of the orbit: ${D}/{V_O1}/{V_O1}wwwwd

V_ORBEXT (default: PRE)
Specify the type of orbit to be copied from the DATAPOOL area. You have two possibilities:

STD: the SLR validation starts directly with a Bernese formatted standard orbit. This is usually the case if you validate orbits that you estimated by yourself.

PRE: you use precise orbits for the SLR validation. In this case, a standard orbit has to be generated first, which is done in PID 110. This type of orbit is available if you validate orbits from external sources such as from CODE or from the IGS.

Satellite Selection

The satellites for the orbit validation are selected with the following BPE server variables:

V_SATSEL (default: SATSEL)
Specify the name of the satellite selection file (default extension FIX, description in Section 24.7.12). The file contains the PRNs of the satellites to be included in

the SLR orbit validation. It must be prepared beforehand and stored in the STA directory of the processing campaign. If no satellite selection file is found in the STA directory, the default selection file distributed with the software release will be used (${D}/MSC/SATSEL.FIX).

According to the specified FIX file, the BPE scripts QLRNX_1 and QLRSUM_1 generate the list of satellites to fill in the option "LIST OF SATELLITES" for the programs QLRINEXO and QLRSUM, respectively.

V_LEO (default: N)
Activate LEO satellites for SLR validation. Different settings are needed in GPSEST when validating the orbit of a LEO satellite.

Observation Selection

The location of the SLR observations in the datapool are given by the following BPE server variables:

V_NPDIR (default: SLR_NP)
Directory with normal point files.

V_NP (default: gnss)
The SLR normal point files are copied from the DATAPOOL area (${D}/{V_NPDIR}/{V_NP}.yymmdd) to the ORX directory of the processing campaign. The copying of daily files is done within the BPE script SLR_COP.
Note: data in both formats CSTG and CRD can be handled. The BPE user script QLRNX_1 automatically switches on May 2, 2012 from the CSTG to the CRD format.

General Model and Standard File Selection

In this section the BPE server variables related to the general processing files are introduced:

V_PCV (default: I08)
Selection of the antenna phase center model identifier.

V_SATINF (default: SATELLIT)
Basename of the satellite information file. The resulting filename is ${X}/GEN/{V_SATINF}.{V_PCV}.

V_PCVINF (default: PCV)
Basename of the antenna phase center correction file. The resulting filename is ${X}/GEN/{V_PCVINF}.{V_PCV}.

V_SATCRX (default: SAT_$Y+0)
Name of the satellite problem file.

File and Product Naming

This group of BPE server variables is used as solution identifiers and defines the names of the files during the BPE processing:

V_A (default: APR)
Station coordinates extrapolated to the processing epoch.

V_G (default: R1_)
Label of the GPSEST output (RES, OUT).

V_Q (default: S1)
Label of the QLRSUM output (SUM, PLT, EDT, OUT).

Management of the Dataflow

The last group of user scripts is related to the file handling within the campaign:

V_SAV (default: Y)
Save the result file into the ${S}/{V_RESULT} area if V_SAV == 'Y'. The saving is managed by the BPE user script SLR_SAV.

V_DEL (default: Y)
If V_DEL == 'Y', delete all files from the campaign that belong to the session 7 days prior to the currently processed one. The file deletion is managed in the SLR_DEL user script. Usually, the deletion should be enabled to keep the campaign clean. For debugging purposes and to identify error sources in the processing, it may be useful to keep all files.

23.4.6.5. Description of the PCF

Copy Required Files, Prepare Standard Orbit

```
#
# Copy required files
# -------------------
001 SLR_COP  NO_OPT            ANY      1
#
010 COOVEL   SLR_VAL           ANY      1 001
020 POLUPDH  SLR_VAL           ANY      1 001
#
# Prepare STD orbit if necessary
# ------------------------------
100 CHK_ORB  NO_OPT            ANY      1 001
110 ORBGEN   SLR_VAL           ANY      1 020 100
```

PID 001 SLR_COP: This script copies all necessary files into the respective campaign directories (see Sections 23.4.6.2 and 23.4.6.4). The BPE will stop with an error if no observation file or no FIX file could be located.

PID 010 COOVEL: The station coordinates are propagated from the epoch of the reference frame file defined in the BPE variable V_CRDINF (January 1, 2005 for EX52_SLR.CRD) to the currently processed session epoch, using the velocities from EX52_SLR.VEL. The resulting coordinate file is called {V_A}yyssss.CRD.

PID 020 POLUPDH: Program POLUPD extracts ERP information from an IERS formatted pole file (extension IEP) into a Bernese formatted pole file (extension ERP).

PID 100 CHK_ORB: This script checks whether V_ORBEXT is set to PRE or STD. In case a PRE file is used, the next script (PID 110) is executed. In case a STD file is used, PID 110 is skipped (cf. values of PARAM1 and PARAM2).

PID 110 ORBGEN: This script is only executed if the orbit is available as a PRE file. In this case a standard orbit file (STD) is generated.

Further reading:

Section 10.7.9: “Propagating Coordinates to Specific Epochs”, page 272
Section 4.3: “Precise Orbit Files”, page 90
Section 4.4: “IGS and IERS Pole Files”, page 91

Import Observations Including Meteo Data

```
#
# Import observations (including meteo data)
# -----------------------------------------
200 QLRNX_1  SLR_VAL            ANY      1 100
#
300 RXMBV_1  SLR_VAL            ANY      1 200
310 RXOBV_1  SLR_VAL            ANY      1 010 200
#
350 CHK_OBS  NO_OPT             ANY      1 300 310
```

PID 200 QLRNX_1: This script converts SLR NP files in the CSTG and CRD format to RINEX observation files using the programs QLRINEXO and CRD2RNXO, respectively. The NP files before May 2, 2012 are supposed to be given in the CSTG format and those after May 2, 2012 are supposed to be given in the CRD format. For each station, a RINEX observation file (`ccccssss.yyO` with `cccc` being the 4-character station code, `ssss` being the session identifier, and `yy` being the 2-digit year) and a RINEX meteo file (`ccccssss.yyM`) is created.

PID 300 RXMBV_1: This script converts RINEX meteo files into Bernese formatted meteo files (`ccccssss.MET` with `cccc` being the 4-character station code and `ssss` being the session identifier).

PID 310 RXOBV_1: This script converts RINEX observation files into Bernese formatted observation files (`ccccssss.RZH` and `ccccsss.RZO` with `cccc` being the 4-character station code and `ssss` being the session identifier).

PID 350 CHK_OBS: This script checks whether Bernese formatted observation files are available for the currently processed session. If observation files are available, the next script (PID 400) is executed. If no observation file is available, however, the next script that is executed is PID 901 (cf. values of `PARAM1` and `PARAM2`). This check ensures that the BPE ends successfully, even if there might be no observations available for one session.

Further reading:

Section 19.3: “Importing SLR Observations”, page 432

Compute SLR Residuals

```
#
# Compute SLR residuals
# ---------------------
400 GPSEST_S SLR_VAL            ANY      1 020 110 350
500 QLRSUM_1 SLR_VAL            ANY      1 400
```

PID 400 GPSEST_S: This GPSEST run compares the computed range, i.e. the range between a laser station and the GNSS orbit (`{V_O1}wwwwd.STD` with `wwww` being the GPS week and `d` being the day of week) with the observed range. No parameters are estimated. The only output files are the binary residual file `{V_G}wwwwd.RES` and the general GPSEST output file `{V_G}wwwwd.OUT`.

PID 500 QLRSUM_1: This script runs QLRSUM for an analysis of the SLR residuals. Three output files are generated: `{V_Q}yyddd.SUM`, which contains information such as station ID, satellite PRN, and residual for each SLR observation, `{V_Q}yyddd.PLT`, which contains the residual output for graphical tools, and `{V_Q}yyddd.EDT` giving the range observations during shadow passages. Moreover, the general QLRSUM output file `{V_Q}yyddd.OUT` is created. Note that `yyddd` specifies the 2-digit year and the day of year.

Further reading:

Chapter 7: "Parameter Estimation", page 169
Chapter 19: "Satellite Laser Ranging (SLR)", page 423
Section 19.5: "Analyzing SLR Data", page 435

Create Summary File, Save and Delete Files

```
#
# Create summary file and delete files
# ------------------------------------
901 SLR_SUM  NO_OPT            ANY      1 500
902 SLR_SAV  NO_OPT            ANY      1 901
903 SLR_DEL  NO_OPT            ANY      1 902
991 BPE_CLN  NO_OPT            ANY      1 903
#
# End of BPE
# ----------
999 DUMMY    NO_OPT            ANY      1 991
```

PID 901 SLR_SUM: All old protocol files of the processed session are deleted. The comprehensive summary is composed of several parts describing the different steps. The protocol file is discussed in Section 23.4.6.6 .

PID 902 SLR_SAV: The result file from this BPE, i.e. `SLRVALyyddd.PRC`, is copied from the campaign into the `SAVEDISK` area.

PID 903 SLR_DEL: This script deletes all files which are no longer needed after the end of the BPE. This prevents unimportant files from accumulating.

PID 991 BPE_CLN: Deletes files from the campaign's `BPE`–subdirectory with a delay of 30 sessions. Thus, all BPE protocol and log files of the last 30 sessions will remain in the directory.

PID 999 DUMMY: Does nothing. This script provides a well defined PID as end point, useful, e.g., as jump address, or to check for the completion of the BPE.

If you want to archive other or more files you should not remove these files from the list in the deletion script (`SLR_DEL`) but add them to the list of files to be copied to the `SAVEDISK` area (`SLR_SAV`).

Further reading:

Section 22.10: "BPE Output and Protocol Files", page 518
Section 22.11: "Error Handling", page 522

23.4.6.6. Description of the Protocol File, Quality Control

The processing is summarized in a so–called protocol file. It is compiled in the BPE user script `SLR_SUM` and exists first in the `OUT` directory of the campaign. Depending on the settings for the BPE server variables, the protocol is copied by the SLR_SAV script (if V_SAV == 'Y') to ${S}/{V_RESULT}/yyyy/SLRVALyyddd.PRC and may be removed from the campaign area by the script SLR_DEL (if V_DEL == 'Y').

This protocol file starts with a section reporting all important input information for the BPE processing defined by the BPE server variables:

```
===============================================================================
SLRVAL BPE PROCESSING SUMMARY FOR YEAR-SESSION 10-207
===============================================================================


Summary file generated at 13-Dec-2012 08:51:15 by SLR_SUM

General files:
   Antenna phase center eccentricity file:   PCV.I08
   Satellite information file:               SATELLIT.I08
   Satellite problem file:                   SAT_2010.CRX

...
```

PART 1: SLR OBSERVATION IMPORT (QL)

The first part of the processing summary contains the program output from the program QLRINEXO or CRD2RNXO (see script QLRNX_1 at PID 200) which converts the SLR normal point files to RINEX format. This part gives the list of included satellites ({V_SATESEL}.FIX), the observation window, and a list of RINEX observation and meteo files.

Further reading:

Section 19.3.2: “Import of SLR Observations to RINEX Format”, page 432

PART 2: SLR RESIDUAL STATISTICS SUMMARY

The program output from QLRSUM is included as the main part of the processing summary. It provides the statistics on the residuals of the SLR measurements as computed by the GPSEST run executed in script GPSEST_S (PID 400).

Further reading:

Section 19.5: “Analyzing SLR Data”, page 435

23.5. Processing Own Data With Example BPEs

The example PCFs provided in Version 5.2 of the *Bernese GNSS Software* are designed to be used in real life applications. The options are set for a daily analysis of a regional network. Depending on your requirements, some steps may be left out, but the intention of these examples is to provide an optimal processing tool for the given applications. This section describes how to use these PCFss on your own campaign data and how to process kinematic stations.

23.5.1. Preliminaries

It is assumed that your campaign has been set up, basic files are located in the correct directories in the DATAPOOL area (${D}), and that the session is correctly defined. For further information see Chapter 3. Now, you have to check whether the structure of your datapool still corresponds to the assumptions implemented in the copy scripts of the PCF in question (PPP_COP, R2S_COP, CLK_COP, LEO_COP, SLR_COP, all located in the ${U}/SCRIPT directory). The observation files are copied using the script RNX_COP (LEO_RNX and SLR_COP when processing LEO or SLR data) following the rules explained in the related section of the detailed description of the corresponding BPE.

Make sure that all result files you want to store in a long–term archive are copied at the end of the PCF by the save scripts (PPP_SUM, R2S_SUM, CLK_SUM, LEO_SUM, SLR_SUM, all located in the ${U}/SCRIPT directory). The observation files are copied on request by the script OBS_SAV (see also the description for the related PCF variable V_SAVOBS in the related section describing the related BPE).

If you are going to process a larger number of stations the input option "Maximum number of parameters in combined NEQ" in the program ADDNEQ2 may become too small. This results in a message like:

```
 *** SR NEQCKDIM: DIMENSION TOO SMALL
                  Requested num. of parameters:        3542
                  Maximum size of the array   :        3000
```

Increasing the values for the expected size of the normal equation in the corresponding input panels will solve this issue.

Another aspect that might improve the performance is the distribution of the jobs between the available CPUs . Please inspect the sections related to the set of V_CLU variables in the related BPE.

After these preparatory steps, you are ready to run a PCF from the examples with your own campaign/data.

You may have to make more changes in the program options or in the user scripts depending on your application (e.g., another troposphere setup according to the recommendations in Section 12.6 if you analyze a local instead of regional network). Be sure to keep the program input panels in the OPT directories consistent with your changes! Surely, by now, you also have realized the value of using menu or BPE variables as much as possible in order to keep modifications in scripts and panels simple.

If you have changed some of the user scripts or program input options and you want to go back to the original examples you can copy them from the directories ${X}/PCF, ${X}/OPT, and ${X}/USERSCPT into your user environment ${U}/PCF, ${U}/OPT, and ${U}/SCRIPT.

23.5.2. Adding New Stations

To include new stations into the processing with the examples, the following steps are necessary:

- Enter the equipment information into the station information file ${D}/{V_REFDIR}/{V_STAINF}.STA: Sections TYPE 001 (if needed) and TYPE 002 .

  It is recommended that you also add the history (if there is any) to be prepared for a potential reprocessing. Possibly, you may also need to enter the station to the file for "Accepted station information" to indicate potential inconsistencies in the RINEX header with respect to the equipment information to be used for the processing.

- Add the station to the cluster ${D}/{V_REFDIR}/{V_STAINF}.CLU and to the tectonic plate assignment file ${D}/{V_REFDIR}/{V_STAINF}.PLD.

- If you want to define the 4– and 2–character abbreviations manually, you need to put them into the abbreviation table ${D}/{V_REFDIR}/{V_STAINF}.ABB — otherwise it is done automatically in the subsequent step.
- Run the PPP_BAS.PCF example BPE on the RINEX observation files of the new stations. It automatically updates the coordinate and velocity files (${D}/{V_REFDIR}/{V_STAINF}.CRD and ${D}/{V_REFDIR}/{V_STAINF}.VEL) as well as the abbreviation table (${D}/{V_REFDIR}/{V_STAINF}.ABB) if it was not done manually before. Note that the automated update procedure depends on the BPE variable V_UPD.

  The list of input files for the PPP processing example BPE is provided in Section 23.4.1.2.
- The ocean tidal loading corrections (${D}/{V_REFDIR}/{V_BLQINF}.BLQ) have to be provided for the new station. Please inspect the instructions in $X/DOC/README_BLQ.TXT.

  At the same time you are asked to provide atmospheric tidal loading corrections in the file ${D}/{V_REFDIR}/{V_ATLINF}.ATL. The corresponding instructions are provided in $X/DOC/README_ATL.TXT.

23.5.3. Change of the Reference Frame

The examples use the IGb 08 realization of the ITRF 2008 reference frame. To change this reference frame to a new realization of the ITRF, only the PCF variable V_REFINF needs to be changed. In the directory REF52 of the DATAPOOL area, the following four files are expected:

{V_REFINF}.CRD coordinates as given in the reference frame in the Bernese coordinate file format.

{V_REFINF}.VEL velocities as given in the reference frame in the Bernese velocity file format.

{V_REFINF}.FIX list of stations that are given in the reference frame as a Bernese station selection file.

{V_REFINF}.SIG list of stations that are given in the reference frame as a Bernese station sigma file (three columns).

for reference frames relevant for an IGS–related processing, all these files can be typically downloaded from http://www.aiub.unibe.ch/download/BSWUSER52/STA. These files are copied into the campaign and merged with the project–specific coordinate and velocity files. This guarantees that you have a complete list of a priori coordinates for your GNSS data processing after a change of the reference frame.

Of course, the same procedure can be done to switch to any other set of reference frame files. This may, e.g., be useful for a local densification of a previously analyzed regional network. Anyhow, if you change from the official IGS realization of the ITRF to another reference frame, make sure that the station positions are consistent with the reference frame of the satellite orbits.

If the reference epoch of a new reference frame is different from January 1, 2005, you have to adapt the reference epoch in the PPP example by adjusting the BPE variable V_REFEPO in the PCFs PPP_BAS.PCF and PPP_DEMO.PCF.

Please note that the consistency between the reference frame and the antenna model is expected. If you want to process historic data, we recommend to use orbit and clock products from a reprocessing effort (e.g., available at ftp://ftp.unibe.ch/aiub/REPRO_yyyy/

BSWUSER52/) instead of switching back to an old reference frame to keep the consistency with the historic orbit products.

23.5.4. Change of the Antenna Models

As stated above, the antenna model has to be consistent with the used reference frame. To exchange the antenna model to be used for the processing, the corresponding satellite information file ({V_SATINF}.{V_PCV}) and phase center correction file ({V_PCVINF}.{V_PCV}) must be available in the ${X}/GEN directory of your software installation. It is recommended to regularly download and update these files from http://www.aiub.unibe.ch/download/BSWUSER52/GEN (in particular, if a new satellite has been launched). The antenna model that is used is indicated by the BPE variable V_PCV.

The receiver antenna offsets, which are used in the program GPSEST for the different GNSS, are included in the normal equation file to allow a consistency check when combining the normal equations in the program ADDNEQ2. As a matter of fact, the offsets for all systems are required that are supported by the receivers connected to the corresponding receiver antenna. This is automatically ensured if the phase center correction file is updated by the program ATX2PCV using the related station information file as proposed by the example BPEs.

You may want to add antennas with your own calibration values to the processing. The calibrations are expected in the ANTEX format and the file must be located in the ${X}/GEN directory. You may specify the filename in the BPE variable V_MYATX of the PCF to update the phase center correction file with these values if needed. If you have several ANTEX files for the update you can introduce a loop in the user script ${U}/SCRIPT/ATX2PCV:

```
...
# Loop all your ATX files:
# ------------------------
  foreach $myAtx ( ... list of your files ... ) {

# Was an ATX specified?
# ---------------------
    if ( $myAtx eq "" || ! -e "${dirGen}/${myAtx}" ) {
      prtMess($bpe,"No␣ANTEX␣file␣for␣update");
    } else {
      prtMess($bpe,"Use␣file␣$myAtx␣for␣update");
      copy("${dirGen}/${myAtx}","${dirOut}");

# Run program
# -----------
      my $PGMNAM = "ATX2PCV";
      $bpe->RUN_PGMS($PGMNAM);


# Update the file in $X/GEN if requested
# -------------------------------------
      my $pcvOld = "${dirGen}/${pcvInf}.${pcv}";
      my $pcvNew = $bpe->getKey("$ENV{U}/INP/ATX2PCV.INP","PHASRSG");


      if (compare("$pcvOld","$pcvNew",    # Copare but remove the creation date
          sub {my $a = $_[0]; my $b = $_[1];
               $a =~ s/[0-9][0-9]-[A-Z][A-Z][A-Z]-[0-9][0-9]  [0-9][0-9]:[0-9][0-9]$//;
               $b =~ s/[0-9][0-9]-[A-Z][A-Z][A-Z]-[0-9][0-9]  [0-9][0-9]:[0-9][0-9]$//;
               $a ne $b} ) != 0) {
        copy("$pcvNew","$pcvOld");
        prtMess($bpe,"Copy␣new␣Bernese␣phase␣eccenter␣file:␣${pcvOld}");
      }
    }
  }  # end the loop over the ATX files
...
```

Note that this script udpates the phase center correction file in ${X}/GEN that is possibly used by several BPE processes. The more updates are needed (because of a long list of ANTEX files to be considered),the higher the risk that another BPE process crashes because of an incomplete file during the copy process. Hence, it is recommended that this update step of the phase center correction file is done only once at the beginning of the processing.

To switch to individually calibrated antennas you have to check the option "Set number of generic antennas to 999999" in panel "ATX2PCV 2: ANTEX Conversion". Please note that the indicator for type–specific calibrations is different for a phase center correction file, containing also individually calibrated antennas, compared to a file with only type–specific calibrations. Please inspect Section 18.2.4 ("Technicalities on the Receiver Antenna Handling") for more details.

23.5.5. Change Troposphere Modeling

The processing examples are distributed in the configuration for using the Global Mapping Function (GMF, Böhm et al. 2006a). If you want to use the Vienna Mapping Function (VMF1, Böhm and Schuh 2004), instead, you have to go through the following modifications in the example BPEs:

1. Download the VMF1 grid files from the TU Vienna server http://ggosatm.hg.tuwien.ac.at/DELAY/GRID/VMFG/yyyy to your DATAPOOL area.
2. Depending on which BPE you want to use, modify the copy scripts (PPP_COP, R2S_COP, and/or CLK_COP in the ${U}/SCRIPT directory to copy the VMF1)grid files from the datapool into your processing campaign. You need five files to cover a full day.
3. The difference between the VMF1 and GMF mapping function is small enough that the VMF1 only needs to be applied for the generation of the final solution. All preprocessing steps may be carried out using the GMF troposphere model.

   In the case of the processing examples the final solution is computed in the following scripts:

PPP_BAS.PCF	302	PPPEDT_P	PPP_GEN
PPP_DEMO.PCF	302	PPPEDT_P	PPP_GEN
	523	TIMEST_P	PPP_KIN
	543	TIMEST_P	PPP_TRP
	563	TIMEST_P	PPP_ION
	571	GPSEST	PPP_RIM
RNX2SNX.PCF	502	GPSCLU_P	R2S_FIN
CLKDET.PCF	502	TIMEST_P	CLK_RES

   Via "Menu>BPE>Edit PCF program input files" you have to change the following options in the GPSEST.INP file of the above listed option directories:

   - add the file name of the input field "Gridded VMF1 coefficients" in panel "GPSEST 1.1: Input Files 1"
   - select DRY_VMF for the option "ZPD model and mapping function (GNSS)" in panel "GPSEST 3.2: General Options 2"
   - select WET_VMF for the option "Tropo: mapping function" in panel "GPSEST 6.1.1: Site-Specific Troposphere Parameters 1"

4. In the clock estimation example (CLKDET.PCF), you may want to introduce a troposphere solution file which was generated using VMF1 troposphere corrections. In that case you have to include the "Gridded VMF1 coefficients" in panel "GPSEST 1.1: Input Files 1" to all GPSEST runs of CLKDET.PCF where this file is introduced:

CLKDET.PCF	412	TIMEST_P	CLK_ED0
	452	TIMEST_P	CLK_EDT
	502	TIMEST_P	CLK_RES

General information on how to use the VMF1 within the *Bernese GNSS Software* is given in the ${X}/DOC/README_VMF.TXT file (accessible via "Menu>Help>Readme": VMF correction from TU Vienna).

23.5.6. Orbit and Clock Products

The example BPEs are prepared using the CODE Earth rotation and satellite orbit and clock products. They result from a rigorously combined processing of GPS and GLONASS data in three day long–arc solutions at the CODE Analysis Center of the IGS. Using these orbit products you will obtain the best possible consistency with the models implemented in the *Bernese GNSS Software* Version 5.2 (even if the development of the orbit and processing models continues at CODE).

Alternatively, you may also use the combined products from the IGS, where GPS– and GLONASS–related products are combined independently from different sets of solutions. If you deposit not only igswwwwd.sp3 but also iglwwwwd.sp3 orbit files in the IGS subdirectory of the DATAPOOL area, both GPS and GLONASS is considered. The two files are merged in one of the BPE processing steps. In order to switch the provider, you have to change the value of the BPE variable V_B from COD to IGS (or to any other provider of the information as far as the corresponding product files are available in the DATAPOOL area).

23.5.7. Using RINEX Observation Files, Version 3

When the example BPEs have been developed, the version 3 of the RINEX files was not yet widely used. Meanwhile, version 3 becomes more and more popular within the scientific GNSS community.

To use RINEX 3 observation files, the BPE server variable V_RX3DIR must point to the location of the RINEX 3 files in the DATAPOOL area. The BPE script RNX_COP will copy the RINEX 3 files into the RAW–directory of your campaign. If the RINEX 3 files have long filenames in the DATAPOOL area (see Section 4.2.2), they will be renamed (assuming that the first four characters are still unique in agreement with the IGS naming convention).

To successfully import RINEX 3 observation files for processing into the *Bernese GNSS Software* Version 5.2 , it is essential to have the observation types listed in the priority list of the observation type selection file (description in Section 24.3.7). The file must be specified as "Receiver-specific observation type priority" in panel "RNXSMT 1.1: General Files" .

It is essential that RINEX 3 observation files are read first by the program RNXSMT because it is the only program that handles the priority list in Version 5.2 of the *Bernese GNSS Software* . The program writes a smoothed RINEX file with a specific COMMENT section in the header reporting the RINEX 3 style observation types for each individual satellite that will be included later on into the binary Bernese formatted observation header files.

If you get a message like

```
 *** PG RNXGRA: No RINEX3 files are permitted.
                They need first to be processed by PGM RNXSMT.
```

from any program reading RINEX files, you should revisit your BPE to make sure that the program RNXSMT is executed prior to the program issuing this message. Another potential source of problems is that the observations get lost: if the measurement type from the RINEX 3 header is not listed in the observation type selection file, the corresponding measurements are not considered by the software.

23.5.8. Near–Real Time Applications

Both example BPEs for PPP (PPP_BAS.PCF and PPP_DEMO.PCF) as well as the double–difference example (RNX2SNX.PCF) are prepared to process the data in an hourly processing scheme. Exchange the session table (see Fig. 3.3 on page 64 in Chapter 3) and provide the BPE with hourly instead of daily RINEX observation files. This is sufficient to process the data of the most recent few hours (according to the value of V_HOURLY in the BPE variable).

Depending on the application, it might be interesting to fix the station coordinates in the near–real time (NRT) processing. In that case they need to be updated regularly, e.g., by processing the data in an independent daily processing scheme. Such a procedure was, e.g., established and described in Arnold et al. (2015a) for an even higher update rate in the solutions.

In addition, it is worth to optimize the distribution of the processing steps to the available CPUs (see the description related to the V_CLU variable series in the individual examples). The ambiguity resolution scheme in the RNX2SNX.PCF double–difference example is by default optimized to resolve as many ambiguities as possible. The processing time can be reduced if you limit yourself to the QIF ambiguity resolution method (scripts GNSQIFAP and GNSQIF_P at PIDs 431 and 432) either by adapting the PCF or by redefining the BPE variables related to the other ambiguity resolution strategies(V_BL_AMB, V_BL_L53, and V_BL_L12) to zero. Please note that the call of the SNGDIF program in PID 303 only forms the baseline files for the code measurements to be used for the Melbourne–Wübbena ambiguity resolution. If this step is disabled, this script can be skipped also to reduce processing time.

It is also worth to define the maximum time that is allowed for a BPE to run on a specific session. This can easily be done by defining a value in the option "Maximum Time (in Seconds) for BPE Run" in panel "RUNBPE 1: Client Environment/Session Selection" when launching the BPE. The same effect can be achieved if you define the keyword BPE_MAXTIME when starting the BPE from a Perl script: $$bpe{BPE_MAXTIME} = 2400, to allow, e.g., a maximum of 2400 seconds (40 minutes) to run the BPE. If this limit is reached, the BPE stops with an error. In this way it can be prevented that several BPEs run at the same time.

23.5.9. How to Include Kinematic Stations in the BPEs

23.5.9.1. Precise Point Positioning (PPP_BAS.PCF)

The following steps are necessary to estimate kinematic coordinates for a station using the PPP_BAS.PCF BPE:

PID 232 CODSPP_P:

- Turn on the kinematic mode by setting option "Estimate coordinates" in panel "CODSPP 2: Input Options" to KINEMATIC.
- Save the results in a kinematic coordinate file (option "Kinematic Coordinates" in panel "CODSPP 1.3: Output Files").

PID 302 PPPEDT_P:

- Introduce the kinematic solution from the previous step in PID 323 using option "Kinematic coordinates" in panel "GPSEST 1.2: Input Files 2".
- Save the results in a "Kinematic coordinates" file.
- Set the "Type of computed residuals" in panel "GPSEST 3.1: General Options 1" to NORM_APRIORI.
- Enable the kinematic positioning with checkbox "Kinematic coordinates" in Section "EPOCH PARAMETERS" of panel "GPSEST 5.1: Setup of Parameters and Pre-Elimination 1" and set the "Pre-elimination: Kinematic coordinates" to EVERY_EPOCH.
- Select the kinematic station in panel "GPSEST 6.9: Kinematic Coordinates".
- Constrain the solution to the a priori positions in the kinematic coordinates file according to the already achieved accuracy (depends primarily on the data quality — according to our experience data quality decreases with increasing rover speed). This value may be adapted during the iteration in this script (e.g., by using the $bpe->putKey(); mechanism, see Section 22.6.5.1).

PID 303 GPSXTR: A filename for the "Kinematic summary" in panel "GPSXTR 2: Output Files" may be added.

PID 311 . . . PID 499: These scripts are not necessary in kinematic mode and can be skipped.

PID 901 PPP_SUM . . . PID 903 BPE_CLN: Adapt these scripts to your needs.

The kinematic solutions of several stations may be merged by simply concatenating the positions in the single kinematic coordinate files. A header section (e.g., from the first file) must be included.

The clock RINEX files that you introduce into the PPP procedure limit the sampling for your kinematic solution. If you use products from CODE (BPE variable V_B = COD) you are provided with a 30 seconds sampling of the satellite clocks. If you want to process your data with a higher sampling you have to use the 5 seconds clock solution provided in dedicated files following the naming convention CODyyddd.CLK_05S.Z, available at the server http://www.aiub.unibe.ch/download/CODE/.

Other standard products, like for instance from the IGS, have typically a sampling of 300 seconds; dedicated densifications to 30 seconds may be provided in alternative files as well.

The following BPE server variables and scripts may need modifications to densify the solution below 300 or even 30 seconds:

V_B: is the BPE server variable to choose the consistent set of precise orbits, Earth orientation information and clock RINEX files. The script PID 001 PPP_COP copies the main clock RINEX file with the name of `{V_B}wwwwd.CLK` and, if available, also `V_Bwwwwd.CLK_30S`. The file with the higher sampling is used for the processing. The file with the 5 seconds sampling from CODE is not considered here. If you want to use it you have to adapt the script accordingly.

V_SAMPL: is the BPE server variable where the sampling of the PPP solution is specified. It is considered in the corresponding input fields for data sampling:

- in PID 222 RXOBV3_H to import the observations from RINEX into the internal Bernese format and
- in PID 302 PPPEDT_P for the option "Sampling interval" in the program GPSEST to process the data.

PID 212 RNXSMT_P:

- Adapt option "Sampling interval for RINEX data" to generate a kinematic solution with a sampling higher than 30 seconds.
- Adapt the number of epochs in the RINEX observation file in the user input field "Maximum number of records in a RINEX file" if needed.

The ultra–high rate clock products from CODE offer precise satellite clock corrections with a sampling of 5 seconds. If you need to process data with a higher sampling you have to interpolate between these clock corrections. According to Bock et al. (2009) this interpolation does not degrade the solution significantly. It may be enabled by specifying an interval of 5 seconds in the option "Clock interpolation allowed over" in all program input panels.

23.5.9.2. Double–Difference Solution (`RNX2SNX.PCF`)

As prerequisite we assume that you have station positions flagged with `K` for all epochs in the kinematic coordinate file (e.g., from a previous PPP solution using the `PPP_BAS.PCF` BPE). Follow the steps below to perform a double–difference solution including kinematic stations:

V_SAMPL: is the BPE server variable where the sampling of the kinematic solution is specified. It is considered in the corresponding input fields for data sampling for the GPSEST program, in particular in the PIDs 322 and 501. The option "Sampling interval of residual files" in program RESRMS is adjusted as well in PID 322.

PID 212 RNXSMT_H: needs to be adapted only if a sampling of higher than 30 seconds is targeted:

- Increase the sampling of the data to be processed from 30 seconds to the value that you need in option "Sampling interval for RINEX data".
- Adjust the number of epochs in the option "Maximum number of records in a RINEX file".

PID 222 RXOBV3_H: assumes in the option "Sampling interval" a sampling interval of 30 seconds that must be increased if a higher sampling for the kinematic solution is needed. Please do not reduce it because the subsequent preprocessing step in the program MAUPRP is requesting this data sampling to reliably distinguish between changes in the ionospheric conditions and real problems in the observations.

PID 232 CODSPP_P:

- Introduce a priori kinematic coordinates (e.g., from a PPP).
- Disable the coordinate estimation by setting "Estimate coordinates" in panel "CODSPP 2: Input Options" to NONE.

PID 302 SNGDIF: Parameters for kinematic coordinates are estimated from a quite low number of observations per parameter. It makes sense to ensure that as few observations as possible of the kinematic station are lost during baseline creation.
If you have, e.g., only one kinematic station in your network you can achieve this by choosing the "Processing strategy" STAR with the kinematic station as "Reference station for STAR (P)".

The network configuration of the SNGDIF script in PID 303 is automatically adapted by the second call of the script. There is no need to make any changes to the call of the program SNGDIF in PID 303 .

PID 312 MAUPRP_P:

- Introduce the a priori kinematic coordinates file from a PPP.
- It is recommended to activate "Do not accept cycle slip corrections" in panel "MAUPRP 8: Cycle Slip Detection/Correction". New phase ambiguities are set up instead.
- Adapt the "Sigma for L1 observations" and "Sigma for L2 observations" to the accuracy of the kinematic coordinates from a PPP.
- Kinematic coordinate estimation in MAUPRP is not necessary if you have excellent kinematic coordinate results from a PPP.
  You should activate the "Kinematic coordinate estimation" in panel "MAUPRP 6: Epoch-Difference Solution" if you have to increase the "Sigma for L1 observations" and "Sigma for L2 observations" significantly (e.g., 0.005 m allows to pass slips of up to three cycles). In that case the "A priori coordinate/baseline vector sigmas" should be as strong as possible.

PID 321 GPSEDTAP: It is preferable to have all baselines containing one kinematic station in the same cluster by setting up the corresponding configuration in the `{V_CRDINF}.CLU` file. Otherwise, the kinematic positions are independently estimated in GPSEST.

PID 322 GPSEDT_P:

- Introduce the a priori kinematic coordinates file from a PPP.
- Enable the estimation of "Kinematic coordinates" and set the "Pre-elimination: Kinematic coordinates" to EVERY_EPOCH.
- Kinematic coordinate estimation is not necessary in the first GPSEST run if the a priori values from the PPP analysis are already very good. But kinematic positioning must be enabled at least in the second run (using the $bpe->putKey(); mechanism).

- Set the "Type of computed residuals" in panel "GPSEST 3.1: General Options 1" to NORM_APRIORI.
  (Variance/covariance information for non–epoch parameters is not available when pre–eliminating kinematic positions every epoch since the option "Var-covar wrt epoch parameters" in panel "GPSEST 6.7: General Options for Epoch Parameters" is set to SIMPLIFIED).
- Select the kinematic station in panel "GPSEST 6.9: Kinematic Coordinates".
- Constrain the solution to the a priori kinematic coordinates according to the already achieved accuracy by specifying "A PRIORI SIGMAS" in panel "GPSEST 6.9: Kinematic Coordinates".

PID 323 GPSXTR: If you have enabled the kinematic coordinate estimation in the previous GPSEST it might be interesting to extract a statistics on the performance of the results. For this purpose the summary file "Kinematic summary" could be specified.

PID 341 ADDNEQ2: The normal equation files written by the second run of the program GPSEST do not contain kinematic position parameters (due to the huge number), but only static coordinate parameters for all remaining sites. In addition, the troposphere parameters for all stations are included. ADDNEQ2 will generate coordinate and troposphere result files but cannot compute the kinematic coordinates.
Add another GPSEST (e.g., as PID 343) to update the kinematic coordinate file.

PID 343 GPSEST:

- Introduce the coordinate and troposphere result files from ADDNEQ2. Set option "Coordinates fixed" in panel "GPSEST 4: Datum Definition for Station Coordinates" to WITH_FLAG. In the next panel, select # (any non–blank flag) for "Stations with specific flags in CRD file". Now all static stations are fixed to their a priori coordinates. Set up the troposphere as in PID 322 and set option "STATIONS TO BE EXCLUDED FROM TROPOSPHERE ESTIMATION" "Tropo: list of stations with no troposphere estimation" to WITH_TROPO.
- Use the variable V_SAMPL to adapt the "Sampling interval".
- Enable the estimation of "Kinematic coordinates" and set the "Pre-elimination: Kinematic coordinates" to EVERY_EPOCH.
- Select the kinematic station in panel "GPSEST 6.9: Kinematic Coordinates".
- Constrain the solution to the a priori kinematic coordinates according to the already achieved accuracy by specifying corresponding "A PRIORI SIGMAS".
- Save the results in a kinematic coordinates file.

PID 412 GNSAMB_P:
PID 422 GNSL53_P:
PID 432 GNSQIF_P:
PID 442 GNSL12_P:
For all ambiguity resolution steps you should adapt the settings in GPSEST as follows:

- Introduce "Kinematic coordinates".
- The station positions are taken from the kinematic coordinates file. No further kinematic setup has to be done. Nevertheless, (static) coordinates are estimated for one of the two stations of the baseline: option "Coordinates fixed" in panel "GPSEST 4: Datum Definition for Station Coordinates" is set to FIRST.

If you are using a sampling different from the usual 30 seconds, the formal error of the ambiguity parameters might change. This has an effect on the statistical evaluation in the frame of the ambiguity resolution algorithm. Please inspect Chapter 8 on "Initial Phase Ambiguities and Ambiguity Resolution" for more details.

PID 501 GPSCLUAP: It is preferable to apply no clustering in this step (V_CLUFIN = 'N').

PID 502 GPSCLU_P:

- Introduce the latest "Kinematic coordinates".
- The "Sampling interval" is taken from the BPE server variable V_SAMPL .
- Enable the estimation of "Kinematic coordinates" and set the "Pre-elimination: Kinematic coordinates" to EVERY_EPOCH.
- Select the kinematic station in panel "GPSEST 6.9: Kinematic Coordinates".
- Constrain the solution to the a priori kinematic coordinates according to the already achieved accuracy by specifying corresponding "A PRIORI SIGMAS".

PID 511 ADDNEQ2: Proceed in analogy to PID 341. ADDNEQ2 provides a coordinate result file with the solution for all static sites and a troposphere result file for all stations. The datum definition is verified in the sequence of script up to PID 514 HELMCHK . Introduce the coordinates generated with the verified datum definition and the related troposphere estimates into an additional GPSEST run (e.g., PID 515) and store the final kinematic coordinate solution. At this step also a GPSXTR script should be added to get some statistics on the kinematic coordinate estimation in a summary file "Kinematic summary" .

PID 901 R2S_SUM . . . PID 904 R2S_DEL: Adapt these scripts according to your needs.

23.5.9.3. Zero–Difference Network Solution (CLKDET.PCF)

As prerequisite we assume that you have station positions flagged with K for all epochs in the kinematic coordinate file (e.g., from a previous PPP solution). Follow the steps below to perform a zero–difference solution including kinematic stations:

The CLKDET.PCF processing example contains two ways for preprocessing zero–difference observation files. The first one assumes that no precise high–rate satellite clock corrections are available:

PID 222 RNXSMT_P:

- Adapt the "Sampling interval for RINEX data" (panel "RNXSMT 2.1: Options") to generate a kinematic solution with a sampling higher than 30 seconds.
- Occasionally, the option "Maximum number of records in a RINEX file" must also be adjusted.

PID 232 RXOBV3_P: The sampling of the observations to be imported into the binary Bernese observation file format is defined by the BPE server variable V_SAMPL .

PID 242 CODSPP_P:

- Introduce the a priori kinematic coordinates file from a PPP.

- Disable coordinate estimation by setting option "Estimate coordinates" to NONE.

The alternative way to preprocess the zero–difference observation files requires precise high–rate satellite clock corrections:

PID 261 RNXSMT_P:

- Adapt the "Sampling interval for RINEX data" (panel "RNXSMT 2.1: Options") to generate a kinematic solution with a sampling higher than 30 seconds.
- Occasionally, the option "Maximum number of records in a RINEX file" must also be adjusted.

PID 272 RXOBV3_P: If you are going to generate a kinematic solution with a sampling of higher than 30 seconds, you have to adapt the value for "Sampling interval" (panel "RXOBV3 2: Input Options 1").

PID 282 CODSPP_P:

- The a priori kinematic coordinate file from a PPP has to be introduced in the input field "Kin. input coordinates".
- The coordinate estimation needs to be disabled by setting option "Estimate coordinates" to NONE.

PID 292 MAUPRP_P:

- In the input option "Kinematic input coordinates" the a priori kinematic coordinate file from a PPP needs to be introduced.
- Adapt the "Sigma for L1 observations" and "Sigma for L2 observations" to the accuracy of the kinematic coordinates from a PPP.
- Kinematic coordinate estimation in MAUPRP is not necessary if you have excellent kinematic coordinate results from a PPP.
  You should activate the "Kinematic coordinate estimation" in panel "MAUPRP 6: Epoch-Difference Solution" if you have to increase the "Sigma for L1 observations" and "Sigma for L2 observations" significantly (e.g., 0.005 m allows to pass slips of up to three cycles). In that case the "A priori coordinate/baseline vector sigmas" should be as strong as possible.

The subsequent scripts are relevant regardless of the selected preprocessing scheme above:

PID 302 TIMEST_C:

- Introduce the a priori kinematic coordinates file from a PPP.
- As in all subsequent processing steps, the data sampling is controlled using the BPE server variable V_SAMPL.

Because in this step only code observations are analyzed, it is typically sufficient to introduce the positions of the kinematic station(s). If you have doubts about the quality, you might also enable the kinematic coordinate estimation in this processing step:

- Enable the estimation of "Kinematic coordinates" and set the "Pre-elimination: Kinematic coordinates" to EVERY_EPOCH.
- Select the kinematic station in panel "GPSEST 6.9: Kinematic Coordinates".

- Constrain the solution to the a priori kinematic coordinates according to the already achieved accuracy by specifying corresponding "A PRIORI SIGMAS".
- Save the results in a kinematic coordinate file.

This kinematic coordinate file has to be used as a priori file for the subsequent processing step.

Note that if you are estimating kinematic coordinates, no parallelization in PID 301 should take place: V_CLUCOD = 1 . Otherwise, independent kinematic solutions in different clusters may result.

PID 303 GPSXTR: If you have enabled the kinematic coordinate estimation in the previous GPSEST run, it might be interesting to extract a statistics on the performance of the results. For this purpose, the summary file "Kinematic summary" could be specified.

PID 321 MKCLUSAC:
PID 322 TIMEST_C:
PID 323 GPSXTR:
PID 341 MKCLUSAC:
PID 342 TIMEST_C:
These steps should be adapted in analogy to the description of scripts PID 302 and 303 above.

PID 411 MKCLUSAP: It is recommended to process all stations together in one cluster if kinematic coordinates are estimated: V_CLUPRP = 1 .

PID 412 TIMEST_P: If you have selected the preprocessing without precise high–rate satellite clock corrections available (V_HRCLK == 'N'), you have to apply here the same changes as described for the subsequent processing step (PID 452).

PID 451 MKCLUSAP: Disable the clustering for kinematic coordinate estimation: V_CLUPRP = 1 .

PID 452 TIMEST_P:

- Introduce the a priori kinematic coordinate file with option "Kinematic coordinates" (panel "GPSEST 1.2: Input Files 2").
- The kinematic coordinate estimation is enabled by checking the checkbox "Kinematic coordinates" (panel "GPSEST 5.1: Setup of Parameters and Pre-Elimination 1"); the option "Pre-elimination: Kinematic coordinates" should be set to EVERY_EPOCH .
- Select the kinematic station(s) in the input option "Station selection" .
- Constrain the kinematic positions as much as possible in the "A PRIORI SIGMAS" input fields.
- Store the results from the kinematic coordinate estimation.

PID 501 MKCLUSAP: All stations should be processed together without any clustering. This is achieved by setting the BPE server variable V_CLUSOL = 1 . If the maximum number of stations given in the BPE server variable V_MAXSOL is smaller than the number of available stations, you should increase this number — otherwise the available stations cannot support the kinematic positioning. If you have any reasons to keep this number small, you should at least take care that your kinematic station(s) are selected. List them in a station selection file that is specified in the option "List of core sites" (panel "MKCLUS 1: Filenames").

PID 502 TIMEST_P:

- Change options in PID 452 .
- Introduce the latest kinematic solution.
- Save the results in a kinematic coordinate file.

PID 421 . . . PID 423: These scripts are not necessary in kinematic mode and can be skipped.

PID 901 CLK_SUM . . . PID 904 CLK_DEL: Adapt these scripts to your needs.

In contrast to PPP, satellite clocks are estimated together with the other parameters for all epochs. Therefore, the broadcast clock corrections can be used as a priori information. They do not limit the sampling of the analysis because they are given as polynomials.

It might be helpful to establish iterations within the BPE after a significant improvement of the kinematic trajectory that allows, for instance, to reduce the assumed noise of the measurements in PID 292 MAUPRP_P .

The `CLKDET.PCF` example BPE can also be adapted to introduce a kinematic coordinate file obtained in the double–difference network processing example `RNX2SNX.PCF` and to compute the corresponding receiver clocks. In that case, you have to copy this file into the campaign (follow the example to copy the coordinate and troposphere solution in PID 001 CLK_COP). This file must be introduced in all further processing steps as described above, but kinematic coordinate estimation must not be be enabled.

24. Data Structure

24.1. Introduction

This chapter gives an introduction into the data structure of the *Bernese GNSS Software* Version 5.2 . This includes a detailed description of the data files and formats used in the software. We start with general input files used by most of the programs and conclude with a variety of output files.

24.2. Overview of the Data Files

In the following sections the content of the files is explained in more detail. The focus is put on some specific directories: the *general files* of the program–specific area and all important files of the *campaign–specific* area.

According to "Menu>Configure>Paths and extensions" we may also divide the file types into the following categories (see Chapter 3).

General files:

Description: Section 24.3
Location: `${X}/GEN` (or `%X%\GEN` for Windows) directory in the program area
Panel: "MENU_EXT 1: General Files"

Observation files:

RINEX files:

Description: Section 24.4, see also Chapter 4
Location: `ORX` and `RAW` directories in the campaign area
Panels: "MENU_EXT 3: Campaign Data, RINEX Files"

Orbit related files:

Description: Section 24.6
Location: `ORB` directory in the campaign area (one exception in `STA`)
Panels: "MENU_EXT 2.1: Campaign Data, Orbit and EOP Files"
"MENU_EXT 2.2: Campaign Data, Orbit Files for LEOs"

Bernese observation files:

Description: Section 24.5
Location: `OBS` directory in the campaign area
Panels: "MENU_EXT 4: Campaign Data, Observation Files"

Station related files:

Description: Section 24.7
Location: `STA` directory in the campaign area
Panels: "MENU_EXT 5: Campaign Data, Station Files"

Atmosphere related files:

Description:	Section 24.8
Location:	ATM directory in the campaign area
Panels:	"MENU_EXT 6: Campaign Data, Atmosphere Modeling Files"

Station and atmosphere related grid files:

Description:	Section 24.9
Location:	GRD directory in the campaign area
Panels:	"MENU_EXT 6: Campaign Data, Atmosphere Modeling Files"

Solution related files:

Description:	Section 24.10
Location:	SOL directory in the campaign area
Panels:	"MENU_EXT 8: Campaign Data, Solution Files"

Output and miscellaneous files:

Description:	Section 24.11
Location:	mostly OUT directory in the campaign area
Panels:	"MENU_EXT 7: Campaign Data, Output Files" "MENU_EXT 9: Campaign Data, Bernese Files (ASCII)"

BPE related files:

Description:	Section 24.12, see also Chapter 22
Location:	configuration in the user area (${U}) protocol and log in BPE directory in the campaign area
Panels:	"MENU_EXT 10: Bernese Processing Engine (BPE)"

24.3. General Files

The general files are important for the processing part. They are independent from the user and from the campaign to be processed. They are stored in the ${X}/GEN directory. Table 24.1 shows the list of general files available in the distribution and indicates which files need to be updated from time to time. Up–to–date files can be downloaded from the anonymous ftp–server at the AIUB (http://www.aiub.unibe.ch/download/BSWUSER52/GEN, see Section 4.13.1).

Table 24.1.: List of general files.

Filename	Content	Updating needed	Description in Section	Download
CONST.	All basic constants used in the *Bernese GNSS Software*	No	24.3.1	BSW aftp
DATUM.	Definition of geodetic datum	Introducing new reference ellipsoid	24.3.2	BSW aftp
GPSUTC.	Leap seconds	When a new leap second is announced by the IERS	24.3.9	BSW aftp

continued on next page

continued from previous page				
Filename	Content	Updating needed	Description in Section	Download
`PCV.I08`[a b]	Phase center eccentricities variations	New antenna included or updating the complete antenna correction model	24.3.3	BSW aftp
`SATELLIT.I08`[a b]	Satellite information	Satellite constellation changes or updating the complete antenna correction model	24.3.4	BSW aftp
`SAT_yyyy.CRX`[b]	Satellite problems	Satellite maneuvers, bad data, ...	24.3.5	BSW aftp
`RECEIVER.`	Receiver information	Introducing new receiver type	24.3.6	BSW aftp
`OBS.SEL`	Receiver type related observation types priority list	If a receiver type with a non–default priority list is introduced	24.3.7	—
`FREQINFO.FRQ`	Frequency information file	Introducing a new SLR site or if the wavelength of a laser is changed	24.3.8	BSW aftp
`POLOFF.`	Pole offset coefficients	Introducing new values from IERS annual report (until 1997)	24.3.10	—
`SINEX.` `SINEX.TRO`	SINEX header information	Adapt SINEX header for your institution	24.3.18	—
`IONEX.`	IONEX header information	Adapt IONEX header for your institution	24.3.19	—
`IAU2000R06.NUT`[c]	Nutation model coefficients	No	24.3.11	BSW aftp
`IERS2000XY.SUB`[c]	Subdaily pole model coefficients	No	24.3.12	BSW aftp
`EGM2008_SMALL.`[c]	Earth potential coefficients	No	24.3.13	BSW aftp
`OT_FES2004.TID`[c]	Ocean tides coefficients	No	24.3.14	BSW aftp
`TIDE2000.TPO`[c]	Solid Earth tide coefficients	No	24.3.15	BSW aftp
`DE405.EPH`	Planetary and Lunar Ephemerides	No	24.3.16	—
`s1_s2_def_ce.dat`	Atmospheric tide deformations	No	24.3.17	—

[a] Pay attention to the consistency of these files, see Chapter 18 for more details.

[b] If you are going to use reprocessed product series from CODE, you should use the consistent set of these files.

[c] Alternatives to these default models are available at `http://www.aiub.unibe.ch/download/BSWUSER52/GEN/ALTERNATIVE_MODELS/`.

24.3.1. Constants File

Type: ASCII
Directory: `${X}/GEN` (UNIX) or `%X%\GEN` (Windows).
Extension: —
Content: All physical constants used in the program system.
Created by: Does not change. Available from BSW aftp server.
Used by: Most of the programs.
Example: Figure 24.1 and `${X}/GEN/CONST.`.

Remarks:

- The constants refer to the WGS–84 system of constants. Exception is GM, where the value from Laser ranging is used and is consistent with the Terrestrial Time (TT) timescale.
- The particular values are read from the file with the subroutine `${LG}/DEFCON.f` and are introduced into programs and subroutines using a command like USE d_const, ONLY : GM.

```
GENERAL CONSTANTS FOR BERNESE GNSS SOFTWARE 5.2
-----------------------------------------------

C      =  299792458.D0     VELOCITY OF LIGHT                M/SEC
FREQ1  = 1575420000.D0     L1-CARRIER FREQUENCY   GPS       1/SEC
FREQ2  = 1227600000.D0     L2-CARRIER FREQUENCY   GPS       1/SEC
FREQ5  = 1176450000.D0     L5-CARRIER FREQUENCY   GPS       1/SEC
FREQP  =   10230000.D0     P-CODE     FREQUENCY   GPS       1/SEC
FREQG1 = 1602000000.D0     L1-CARRIER FREQUENCY   GLONASS   1/SEC
FREQG2 = 1246000000.D0     L2-CARRIER FREQUENCY   GLONASS   1/SEC
DFRQG1 =     562500.D0     L1-CARRIER FREQ. DIFF. GLONASS   1/SEC
DFRQG2 =     437500.D0     L2-CARRIER FREQ. DIFF. GLONASS   1/SEC
FREQGP =    5110000.D0     P-CODE     FREQUENCY   GLONASS   1/SEC
FRQE1  = 1575420000.D0     L1-CARRIER FREQUENCY   GALILEO   1/SEC
FRQE5  = 1191795000.D0     L5-CARRIER FREQUENCY   GALILEO   1/SEC
FRQE5a = 1176450000.D0     L5a-CARRIER FREQUENCY  GALILEO   1/SEC
FRQE5b = 1207140000.D0     L5b-CARRIER FREQUENCY  GALILEO   1/SEC
FRQE6  = 1278750000.D0     L6-CARRIER FREQUENCY   GALILEO   1/SEC
FRQS1  = 1575420000.D0     L1-CARRIER FREQUENCY   SBAS      1/SEC
FRQS5  = 1176450000.D0     L5-CARRIER FREQUENCY   SBAS      1/SEC
FRQC1  = 1589740000.D0     L1-CARRIER FREQUENCY   COMPASS   1/SEC
FRQC2  = 1561098000.D0     L2-CARRIER FREQUENCY   COMPASS   1/SEC
FRQC5b = 1207140000.D0     L5b-CARRIER FREQUENCY  COMPASS   1/SEC
FRQC6  = 1268520000.D0     L6-CARRIER FREQUENCY   COMPASS   1/SEC
FRQJ1  = 1575420000.D0     L1-CARRIER FREQUENCY   QZSS      1/SEC
FRQJ2  = 1227600000.D0     L2-CARRIER FREQUENCY   QZSS      1/SEC
FRQJ5  = 1176450000.D0     L5-CARRIER FREQUENCY   QZSS      1/SEC
FRQJ6  = 1278750000.D0     L6-CARRIER FREQUENCY   QZSS      1/SEC
GM     = 398.6004415D12    GRAVITY CONSTANT*EARTH MASS      M**3/SEC**2
GMS    = 1.3271250D20      GRAVITY CONSTANT*SOLAR MASS      M**3/SEC**2
GMM    = 4.9027890D12      GRAVITY CONSTANT*LUNAR MASS      M**3/SEC**2
AU     = 149597870691      ASTRONOMICAL UNIT                M
AE     =    6378137.D0     EQUATORIAL RADIUS OF EARTH       M
CONRE  =    6371000.D0     MEAN RADIUS OF THE EARTH         M
J2     =   1.0826359D-3    DYNAMICAL FORM-FACTOR IERS(2003) 1
FACTEC = 40.3D16           IONOSPHERIC FACTOR               M/SEC**2/TECU
P0     =          -.94D-7  NOMINAL RAD.PR. ACCELERAT.       M/SEC**2
OMEGA  = 7292115.1467D-11  ANGULAR VELOCITY OF EARTH        RAD/SEC
ET-UTC =         55.       EPH. TIME (ET) MINUS UTC         SEC
WGTPHA =          1.D0     WEIGHT FOR PHASE OBSERVATIONS    1
WGTCOD =          1.D-4    WEIGHT FOR CODE OBSERVATIONS     1
HREF   =         0.        REFERENCE HEIGHT FOR METEO MODEL M
PREF   =       1013.25     PRESSURE AT HREF                 MBAR
TREF   =         18.       TEMPERATURE AT HREF              DEG. CELSIUS
HUMREF =         50.       HUMIDITY AT HREF                 %
```

Figure 24.1.: File of all physical constants `CONST.`.

- The values for WGTPHA and WGTCOD are used to specify the relative weights between phase and code observations (if you use both observation types simultaneously in the parameter estimation program GPSEST).
- Carrier frequencies for various GNSSs and frequency differences for GLONASS are included.
- HREF, PREF, TREF, and HUMREF are used for the definition of the standard troposphere models.
- The major constants contained in this file should *never* be changed by the user.

24.3.2. Geodetic Datum Information

Type: ASCII
Directory: ${X}/GEN (UNIX) or %X%\GEN (Windows).
Extension: —
Content: Information concerning different geodetic datum definitions for transforming geocentric cartesian coordinates into ellipsoidal coordinates.
Created by: User–defined. Available at BSW aftp server.
Used by: All programs accessing coordinate files. RXNPRE ("Menu>RINEX>Import RINEX to Bernese format>Navigation files to SP3") for conversion of PZ–90 into ITRF.
Example: Figure 24.2 and ${X}/GEN/DATUM..

Users may add more geodetic data. Each coordinate file refers to one of the datum information specified in this list. The datum information is only used to compute ellipsoidal coordinates of the sites and has no influence on the estimated coordinates. The entry for

```
DATUM DEFINITION FILE FOR BERNESE GNSS SOFTWARE 5.2
---------------------------------------------------
DATUM                ELLIPSOID          SHIFTS (M)          ROTATIONS (")

ITRF2008             AE = 6378137.000   DX =      0.0000    RX =    0.0000
    GLOBAL           1/F= 298.2572221   DY =      0.0000    RY =    0.0000
                     SC = 0.00000D+00   DZ =      0.0000    RZ =    0.0000

IGb08                AE = 6378137.000   DX =      0.0000    RX =    0.0000
    GLOBAL           1/F= 298.2572221   DY =      0.0000    RY =    0.0000
                     SC = 0.00000D+00   DZ =      0.0000    RZ =    0.0000

IGS08                AE = 6378137.000   DX =      0.0000    RX =    0.0000
    GLOBAL           1/F= 298.2572221   DY =      0.0000    RY =    0.0000
                     SC = 0.00000D+00   DZ =      0.0000    RZ =    0.0000

...

ETRS89               AE = 6378137.000   DX =      0.0000    RX =    0.0000
    REGIONAL         1/F= 298.2572221   DY =      0.0000    RY =    0.0000
                     SC = 0.00000D+00   DZ =      0.0000    RZ =    0.0000

WGS - 84             AE = 6378137.000   DX =      0.0000    RX =    0.0000
    GLOBAL           1/F= 298.2572236   DY =      0.0000    RY =    0.0000
                     SC = 0.00000D+00   DZ =      0.0000    RZ =    0.0000

CH - 1903            AE = 6377397.200   DX =    679.0000    RX =    0.0000
    LOCAL            1/F= 299.1528000   DY =     -2.0000    RY =    0.0000
                     SC = 0.00000D+00   DZ =    404.0000    RZ =    0.0000

PZ - 90              AE = 6378137.000   DX =      0.0000    RX =    0.0000
    REGIONAL         1/F= 298.2572236   DY =      0.0000    RY =    0.0000
                     SC = 0.00000D+00   DZ =      0.0000    RZ =   -0.3345

PZ - 90.02           AE = 6378137.000   DX =     -0.3600    RX =    0.0000
    REGIONAL         1/F= 298.2572236   DY =      0.0800    RY =    0.0000
                     SC = 0.00000D+00   DZ =      0.1800    RZ =    0.0000
```

Figure 24.2.: File of the geodetic datum definitions DATUM..

PZ–90 is required by the program RXNPRE for the transformation of GLONASS broadcast information into ITRF.

The additional keywords `GLOBAL`, `REGIONAL`, or `LOCAL` characterize the validity of the reference ellipsoid. Only `GLOBAL` ones can be used as the basis for the GNSS analysis itself to establish the consistency with the satellite orbits and related global parameters. After the GNSS analysis the coordinate solution may be transformed in any `REGIONAL`, or `LOCAL` frame, using, e.g., one of the following programs ETRS89 ("Menu>Service>Coordinate tools>Transform to ETRS89"), COOSYS ("Menu>Service>Coordinate tools>Coordinate transformation"), or HELMR1 ("Menu>Service>Coordinate tools>Helmert transformation").

24.3.3. Antenna Phase Center Offsets and Patterns

Type:	ASCII
Directory:	`${X}/GEN` (UNIX) or `%X%\GEN` (Windows) for the input file, campaign–specific directory `OUT` for output files.
Extension:	—
Content:	Antenna phase center offsets and variations for receiver and satellite antennas.
Created by:	User–defined for input, GPSEST ("Menu>Processing>Parameter estimation") and ADDNEQ2 ("Menu>Processing>Combine normal equation systems") for estimation, ATX2PCV ("Menu>Conversion>ANTEX to Bernese format") for output. Available from BSW aftp server.
Used by:	All programs processing GNSS and SLR observations.
Example:	Figures 24.3 and 24.4 show an example. An up–to–date version of the file `PCV.I08` is available at the anonymous BSW ftp area (see Section 4.13).

Remarks on the format:

- Below the title line three keywords provide some reference information on the file. A antenna phase center correction file may contain as many entries for antennas as needed. The record related to a specific antenna starts with `ANTENNA/RADOME TYPE`.
- The first part of each contains the antenna name, radome indicator, and antenna number. For each GNSS the resolution and source of information is indicated separately. A specific field defines the `TYP` of the antenna correction model:
  `TYP = 0:` No antenna phase center variations are given.
  `TYP = 1:` Phase center variations are stored as a grid depending on zenith/nadir angle (with a resolution of `D(Z)` up to the maximum, defined by `M(Z)` in degrees) and azimuth (with a resolution of `D(A)` degrees). A resolution in azimuth `D(A)` of 360° means that no azimuth dependency is considered.
  `TYP = 2-4:` Phase center variations are stored in the form of spherical harmonic coefficients of degree `N(Z)` and order `M(A)` (TYP = 2: unnormalized; TYP = 3: normalized; TYP = 4: normalized with upper hemisphere only).
- In the second part of each antenna block (indicated with the keywords `NORTH MM`, `EAST MM`, and `UP MM`) the mean phase center offsets are given (north, east, and up components) for each available frequency of each available GNSS.
- In the third part of each antenna block (indicated by the keyword “A\Z”) the phase center variations are stored according to the type–flag and resolution in the first part. For each system and frequency a new block is expected.

```
ANTENNA PHASE CENTER VARIATIONS DERIVED FROM ANTEX FILE          09-OCT-12 16:33
--------------------------------------------------------------------------------
FORMAT VERSION:  1.00
ANTENNA MODEL: IGS08
LAST INPUT FILE: I08.ATX

PHASE CENTER OFFSETS AND MAPS AND/OR COEFFICIENTS OF SPHERICAL HARMONICS IN MM:
-------------------------------------------------------------------------------

TYPE 1 :  ELEVATION (RESP. NADIR)/AZIMUTH GRID
TYPE 2 :  SPHERICAL HARMONICS COEFFICIENTS (UNNORMALIZED)
TYPE 3 :  SPHERICAL HARMONICS COEFFICIENTS (NORMALIZED)
TYPE 4 :  SPHERICAL HARMONICS COEFFICIENTS, UPPER HEMISPHERE ONLY (NORMALIZED)

D(Z)   :  ZENITH (RESP. NADIR) TABULAR INTERVAL (DEGREES)
D(A)   :  AZIMUTH TABULAR INTERVAL (DEGREES)
N(Z)   :  DEGREE OF SPHERICAL HARMONICS DEVELOPMENT
M(A)   :  ORDER  OF SPHERICAL HARMONICS DEVELOPMENT
M(Z)   :  MAXIMUM ZENITH (RESP. NADIR) ANGLE

...

ANTENNA/RADOME TYPE  NUMBER SYS FRQ TYP D(O) D(Z) D(A) M(Z)  SINEX      METHOD               DATE       REMARK
******************** ****** *   *** ***  ***  ***  ***  ***  ********** ******************** ********** ************************************
MW  BLOCK IIA    035      1       2   1    0    1  360   17  IGS08_1708                      25-MAR-11

      NORTH MM  EAST MM   UP MM   FACTOR
      *****.** *****.** *****.**  **********
 01 0     0.00     0.00     0.00   0.100E+01
 02 0     0.00     0.00     0.00   0.100E+01

 01 0  A\Z     0      1      2      3      4      5      6      7      8      9     10     11     12     13     14     15     16     17
 01 0   0  -0.80  -0.90  -0.90  -0.80  -0.40   0.20   0.80   1.30   1.40   1.20   0.70   0.00  -0.40  -0.70  -0.90  -0.90  -0.90  -0.90
 02 0  A\Z     0      1      2      3      4      5      6      7      8      9     10     11     12     13     14     15     16     17
 02 0   0  -0.80  -0.90  -0.90  -0.80  -0.40   0.20   0.80   1.30   1.40   1.20   0.70   0.00  -0.40  -0.70  -0.90  -0.90  -0.90  -0.90

...

ANTENNA/RADOME TYPE  NUMBER SYS FRQ TYP D(O) D(Z) D(A) M(Z)  SINEX      METHOD               DATE       REMARK
******************** ****** *   *** ***  ***  ***  ***  ***  ********** ******************** ********** ************************************
SLR REFL.  GPS                0       2   0    0    0    0   90

      NORTH MM  EAST MM   UP MM   FACTOR
      *****.** *****.** *****.**  **********
 01 0     0.00     0.00     0.00   0.100E+01
 02 0     0.00     0.00     0.00   0.100E+01

...
```

Figure 24.3.: Antenna phase center correction model IGS08 (file PCV.I08, example for satellite antennas).

```
...

ANTENNA/RADOME TYPE  NUMBER SYS FRQ TYP D(O) D(Z) D(A) M(Z)  SINEX      METHOD               DATE       REMARK
******************** ****** *   *** ***  ***  ***  ***  ***  ********** ******************** ********** ****************************************
LEIAR25.R3       NONE      0 G     2   1    0    5    5   90  IGS08_1708 ROBOT                25-MAR-11
                             R     2   1    0    5    5   90  IGS08_1708 ROBOT                25-MAR-11

      NORTH MM  EAST MM   UP MM    FACTOR
      *****.** *****.** *****.**  **********
G01 0     0.41     0.11   162.23   0.100E+01
G02 0     0.18    -0.51   159.51   0.100E+01
R01 0     0.41     0.11   162.23   0.100E+01
R02 0     0.18    -0.51   159.51   0.100E+01

G01 0  A\Z      0      5     10     15     20     25     30     35     40     45     50     55     60     65     70     75     80     85     90
G01 0    0   0.00   0.22   0.88   1.81   2.76   3.46   3.72   3.46   2.75   1.76   0.68  -0.30  -1.08  -1.65  -1.93  -1.82  -1.09   0.49   2.91
G01 0    5   0.00   0.22   0.87   1.80   2.75   3.46   3.72   3.46   2.74   1.72   0.63  -0.37  -1.16  -1.71  -2.00  -1.89  -1.19   0.35   2.77
...
G01 0  355   0.00   0.22   0.89   1.81   2.76   3.45   3.71   3.46   2.77   1.79   0.74  -0.23  -1.01  -1.58  -1.87  -1.74  -0.97   0.66   3.12
G01 0  360   0.00   0.22   0.88   1.81   2.76   3.46   3.72   3.46   2.75   1.76   0.68  -0.30  -1.08  -1.65  -1.93  -1.82  -1.09   0.49   2.91
G02 0  A\Z      0      5     10     15     20     25     30     35     40     45     50     55     60     65     70     75     80     85     90
G02 0    0   0.00  -0.14  -0.49  -0.99  -1.55  -2.14  -2.74  -3.38  -4.03  -4.60  -4.92  -4.84  -4.25  -3.15  -1.64   0.19   2.31   4.87   8.07
G02 0    5   0.00  -0.14  -0.50  -0.99  -1.56  -2.16  -2.77  -3.42  -4.07  -4.65  -4.99  -4.93  -4.36  -3.29  -1.81   0.00   2.11   4.67   7.82
...
G02 0  355   0.00  -0.14  -0.49  -0.98  -1.54  -2.12  -2.72  -3.35  -3.99  -4.54  -4.86  -4.76  -4.14  -3.01  -1.46   0.38   2.51   5.09   8.34
G02 0  360   0.00  -0.14  -0.49  -0.99  -1.55  -2.14  -2.74  -3.38  -4.03  -4.60  -4.92  -4.84  -4.25  -3.15  -1.64   0.19   2.31   4.87   8.07
R01 0  A\Z      0      5     10     15     20     25     30     35     40     45     50     55     60     65     70     75     80     85     90
R01 0    0   0.00   0.01   0.37   1.04   1.91   2.71   3.19   3.16   2.60   1.66   0.58  -0.36  -1.05  -1.48  -1.71  -1.72  -1.28   0.04   2.55
R01 0    5   0.00   0.01   0.36   1.03   1.89   2.69   3.17   3.13   2.57   1.61   0.52  -0.44  -1.14  -1.57  -1.82  -1.82  -1.37  -0.06   2.39
...
R01 0  355   0.00   0.01   0.38   1.06   1.92   2.72   3.21   3.19   2.65   1.72   0.67  -0.27  -0.92  -1.35  -1.58  -1.61  -1.17   0.18   2.80
R01 0  360   0.00   0.01   0.37   1.04   1.91   2.71   3.19   3.16   2.60   1.66   0.58  -0.36  -1.05  -1.48  -1.71  -1.72  -1.28   0.04   2.55
R02 0  A\Z      0      5     10     15     20     25     30     35     40     45     50     55     60     65     70     75     80     85     90
R02 0    0   0.00  -0.26  -0.75  -1.37  -2.00  -2.60  -3.20  -3.86  -4.60  -5.37  -5.97  -6.23  -5.96  -5.10  -3.68  -1.78   0.54   3.32   6.55
R02 0    5   0.00  -0.27  -0.77  -1.38  -2.02  -2.64  -3.25  -3.92  -4.66  -5.42  -6.04  -6.30  -6.05  -5.23  -3.86  -1.98   0.33   3.11   6.28
...
R02 0  355   0.00  -0.25  -0.74  -1.35  -1.97  -2.57  -3.15  -3.81  -4.54  -5.30  -5.91  -6.15  -5.85  -4.94  -3.48  -1.57   0.74   3.52   6.86
R02 0  360   0.00  -0.26  -0.75  -1.37  -2.00  -2.60  -3.20  -3.86  -4.60  -5.37  -5.97  -6.23  -5.96  -5.10  -3.68  -1.78   0.54   3.32   6.55

ANTENNA/RADOME TYPE  NUMBER SYS FRQ TYP D(O) D(Z) D(A) M(Z)  SINEX      METHOD               DATE       REMARK
******************** ****** *   *** ***  ***  ***  ***  ***  ********** ******************** ********** ****************************************
LEIAR25.R4       LEIT      0 G     2   1    0    5    5   90  IGS08_1708 ROBOT                25-MAR-11
                             R     2   1    0    5    5   90  IGS08_1708 ROBOT                25-MAR-11
...
```

Figure 24.4.: Antenna phase center correction model IGS08 (file PCV.I08, example for receiver antennas).

- To ensure the full consistency, satellite and receiver antenna phase center corrections are maintained in the same file. Because the satellite antennas are used more often in the processing they should be listed above the receiver antenna corrections.

Remarks on usage:

- Use IGS values in order to assure consistency with the IGS products.
- It is highly recommended to import values from ANTEX using program ATX2PCV ("Menu>Conversion>ANTEX to Bernese format").
- The IGS maintains the file `igs08.atx` which contains absolute IGS phase center corrections for satellite and receiver antennas, which shall be used with the IGS08/IGb08 terrestrial reference frame. These values were obtained from robot calibrations.
- The IGS has investigated the effect of using absolute antenna phase patterns. For large networks such patterns require the use of block–specific satellite antenna phase patterns and satellite specific antenna offsets (Schmid and Rothacher 2003). On GPS week 1400 the IGS has switched to the absolute antenna model `IGS05` (`igs05.atx`). Since GPS week 1632, the IGS is using the absolute antenna model `IGS08` (`igs08.atx`); the full history of reference frame related updates is given in Table 10.2 (page 246).
- Relative phase center corrections are not supported anymore. With the exchange of the reference frame (e.g., IGS 05 to IGS 08) the antenna phase center corrections referring to the old frame (e.g., `igs05.atx`) become obsolete at the same time and are not maintained anymore (e.g., regarding satellite launches).

Further remarks:

- The Bernese PCV file contains patterns for the GNSS satellites (which are zero for relative phase patterns). The patterns have to be named with the sensor name that is specified in the second section in the satellite information file (see Section 24.3.4), e.g., `MW  BLOCK IIF    062` for the Block IIF satellite with PRN 25 (as of July 2014). If a satellite is carrying several sensors (e.g., microwave emitter and SLR reflector), two entries are needed for this particular satellite (see for instance the example in Figure 24.3).
- Starting from GPS week 1017 (July 1999), a new IGS convention for receiver and antenna names should be used as the basis for the receiver antenna naming to remain compatible with the naming in the ANTEX files of the IGS. The list of the currently valid receiver and antenna names may be found in `ftp://ftp.igs.org/pub/station/general/rcvr_ant.tab`. These antenna names have also to be used in the header of the Bernese observation files (see Section 4.2.3.3 for importing the observations).
- The antenna number is a mandatory entry in the antenna phase center correction files. It can be either `NUMBER == 0` which means that the first entry matching the antenna name (including the radome) is used. In all other cases not only a match of the `ANTENNA/RADOME TYPE` but also in the field `NUMBER` is needed to select the related record. For more details on the usage of the antenna numbers in the context of individually or type–specific calibration please be referred to Table 18.1 and the related Section 18.2.4 .
- The "antenna type" `NULLANTENNA` is coming from the real–time streaming and indicates that the PCV corrections are already applied to the measurements (see IGS Mail 6821). The processing with *Bernese GNSS Software* will stop if a RINEX file is containing this entry in the header, unless the user explicitly confirms the

compatibility of the applied corrections to the corrections for the other antennas by adding this particular antenna into the file.

- Receiver antenna phase patterns can be estimated in GPSEST and ADDNEQ2. The output files have the same format as the input file. The extensions of these output files are `PHG` or `PHH`, depending on the strategy used for the estimation of the patterns. For satellite patterns only grids are supported.
- For details concerning the estimation of elevation– (and azimuth–) dependent antenna phase center variations for ground (and satellite) antennas see Chapter 18.

24.3.4. Satellite Information File

Type: ASCII
Directory: `${X}/GEN` (UNIX) or `%X%\GEN` (Windows).
Extension: Variable.
Content: Satellite information (block, sensor offsets, masses, etc.) for GPS (PRN < 100), GLONASS (100 < PRN < 200), geostationary (300 < PRN < 400), and LEOs (PRN > 900). For Galileo 200 < PRN < 300 is reserved.
Created by: User–defined. Available from BSW aftp server.
Used by: Most orbit and processing programs.
Example: Figure 24.5. An up–to–date version of the file `SATELLIT.I08` is available at the anonymous BSW ftp area (see Section 4.13).

Remarks:

- Please be aware that the information contained in the satellite information file for *Bernese GNSS Software* Version 5.2 is different than from Version 5.0 .
- The header provides information on the format version, the antenna model and the radiation pressure model name. As of July 2014, those are `1.00`, `IGS08`, and `C061001m` respectively.
- In `PART 1: PHYSICAL SATELLITE PARAMETERS` the satellites are listed with block numbers, COSPAR numbers, attitude flag, time window (launch and decommissioning epochs), masses, area–to–mass ratio, a priori radiation pressure coefficient, air drag model specification, air drag coefficient, and the orbital plane.
- `PART 2: ON-BOARD SENSORS` contains for each satellite the sensor name(s), time window (which needs to be the same as in the first section!), sensor offsets (for microwave antenna or SLR reflector), sensor boresight and azimuth unit vectors w.r.t. a satellite–fixed coordinate system, the ANTEX sensor name, the frequency number (GLONASS), and the list of transmitted signals. The sensor name has to be defined in the antenna phase center file.
- `PART 3: RPR PARAMETERS` contains the information concerning the radiation pressure model to be used.
- A fourth section (`PART 4: REMARKS`) gives the information on the different flags and parameters, provides internet addresses of related sites, and contains a change log. This section is not considered by any of the processing programs.
- If a new satellite is launched, the information for this new satellite has to be included into the file. A new line has to be added in the different parts and time window, block number, antenna offsets, etc. has to be introduced correctly. You may copy the lines from another satellite and adapt it accordingly. In particular pay attention to the block and the antenna offsets. Check that no time window of an inactive satellite with the same PRN number is overlapping with that of the new satellite. It is also essential that the time windows in all three parts are fully consistent.

```
SATELLITE-SPECIFIC INFO FOR GPS/GLONASS/GEO/LEO/SLR, BSW5.2          04-JUL-2013
--------------------------------------------------------------------------------

FORMAT VERSION:  1.00
ANTENNA MODEL: IGS08
RADIATION PRESSURE MODEL : C061001    (CODE MODEL COD9801, SPRINGER ET AL, 98)


PART 1: PHYSICAL SATELLITE PARAMETERS
-------------------------------------

         BLOCK    COSPAR  ATTITUDE      START TIME           END TIME           MASS    AREA/MASS RAD.PRESS AIR DRAG AIR DRAG  PLN SLT
PRN  SVN  NO.       ID      FLAG   YYYY MM DD HH MM SS  YYYY MM DD HH MM SS     (KG)    (M2/KG)       CO     MODEL     CO

  1   32   3   1992-079A     0     1992 11 22 00 00 00  2008 10 17 00 00 00     975.0     0.0000     0.0000     0     0.0000     6
  1   37   3   1993-032A     0     2008 10 23 00 00 00  2009 01 07 00 00 00     975.0     0.0000     0.0000     0     0.0000     3
  1   49   7   2009-014A     0     2009 03 24 00 00 00  2011 05 07 00 00 00    1100.0     0.0000     0.0000     0     0.0000     2
  1   35   3   1993-054A     0     2011 06 02 00 00 00  2011 07 13 00 00 00     975.0     0.0000     0.0000     0     0.0000     2
  1   63   8   2011-036A     0     2011 07 16 00 00 00                         1555.3     0.0000     0.0000     0     0.0000     4
...
101  771 101   1992-005C     0     1996 01 01 00 00 00  1996 12 21 23 59 59     900.0     0.0000     0.0000     0     0.0000
101  779 101   1998-077A     0     1999 01 01 00 00 00  2002 07 08 23 59 59     900.0     0.0000     0.0000     0     0.0000
101  796 101   2004-053A     0     2004 12 26 00 00 00  2009 12 13 23 59 59     900.0     0.0000     0.0000     0     0.0000   101
101  730 102   2009-070A     0     2009 12 14 00 00 00                         1415.0     0.0000     0.0000     0     0.0000   101
...

PART 2: ON-BOARD SENSORS
------------------------
                                              START TIME           END TIME                 SENSOR OFFSETS (M)       SENSOR BORESIGHT VECTOR (U)  ...
PRN  TYPE  SENSOR NAME______SVN  NUMBER  YYYY MM DD HH MM SS  YYYY MM DD HH MM SS         DX        DY        DZ         X       Y       Z        ...

  1  MW    MW  BLOCK IIA    032       1  1992 11 22 00 00 00  2008 10 17 00 00 00       0.2790    0.0000    2.3808     0.0000  0.0000  1.0000     ...
  1  MW    MW  BLOCK IIA    037       2  2008 10 23 00 00 00  2009 01 07 00 00 00       0.2790    0.0000    2.3522     0.0000  0.0000  1.0000     ...
  1  MW    MW  BLOCK IIR-M  049       1  2009 03 24 00 00 00  2011 05 07 00 00 00       0.0000    0.0000    0.9656     0.0000  0.0000  1.0000     ...
  1  MW    MW  BLOCK IIA    035       3  2011 06 02 00 00 00  2011 07 13 00 00 00       0.2790    0.0000    2.6220     0.0000  0.0000  1.0000     ...
  1  SLR   SLR REFL.  GPS                2011 06 02 00 00 00  2011 07 13 00 00 00       0.8626   -0.5245    0.6695     0.0000  0.0000  1.0000     ...
  1  MW    MW  BLOCK IIF    063       1  2011 07 16 00 00 00                            0.3940    0.0000    1.5613     0.0000  0.0000  1.0000     ...
...

PART 3: RPR PARAMETERS
----------------------

# GPS SATELLITES
# --------------

        START TIME           END TIME
PRN  YYYY MM DD HH MM SS  YYYY MM DD HH MM SS   M    #     DO        YO        BO        DOC2      DOC4      YOC2      BOC2      BOC4   ...

  1  1992 11 22 00 00 00  2008 10 17 00 00 00   8   19  -91.6720    0.8190   -0.6598   -0.7192    0.5139   -0.0651    0.2771    0.1292  ...
  1  2008 10 23 00 00 00  2009 01 07 00 00 00   8   19  -90.9540    1.1482   -0.3390   -0.7192    0.5139   -0.0651    0.2771    0.1292  ...
  1  2009 03 24 00 00 00  2011 05 07 00 00 00   0    0
  1  2011 06 02 00 00 00  2011 07 13 00 00 00   8   19  -91.0040    0.6907   -0.3394   -0.7192    0.5139   -0.0651    0.2771    0.1292  ...
  1  2011 07 16 00 00 00                        0    0
...
```

Figure 24.5.: Satellite information file SATELLIT.I08.

- An up–to–date `SATELLIT.xxx` file may be downloaded from the anonymous BSWUSER ftp area (`http://www.aiub.unibe.ch/download/BSWUSER52/`, see Section 4.13).
- The entries for the `SENSOR NAME` is used to identify a satellite in the antenna phase center correction file (see Section 24.3.3). Together with the entry `NUMBER` it must be unique: if one and the same satellite vehicle is active using different PRNs or different frequency numbers (in case of GLONASS, for each period a new entry in the `PART 2: ON-BOARD SENSORS` is needed using the same `SENSOR NAME` but different entries for `NUMBER` as demonstrated in the example of Figure 24.5 .
- Observations without satellite information for the specific epochs are removed by programs RXOBV3 and RNXSMT. A corresponding warning message is issued.
- The model number (column `M`) in `PART 3: RPR PARAMETERS` refers to specific solar radiation pressure models with a given number and order of coefficients behind.

24.3.5. Satellite Problem File

Type:	ASCII
Directory:	`${X}/GEN` (UNIX) or `%X%\GEN` (Windows).
Extension:	`CRX`
Content:	Problems with satellites (maneuvers, bad data).
Created by:	User–defined. Available from BSW aftp server. Updated by DEFXTR ("Menu>Orbits/EOP>Extract ORBGEN program output") for orbit splittings, RESCHK ("Menu>Service>Automated processing>Detect misbehaving stations/satellites") for bad satellite data.
Used by:	Most orbit and processing programs.
Example:	Figure 24.6 and `${X}/GEN/SAT_2015.CRX`.

Remark:

- We recommend to use the updated files (filename characterized by the year) from the anonymous BSWUSER ftp area (`http://www.aiub.unibe.ch/download/BSWUSER52/`, see Section 4.13). Using this file you avoid many troubles related to problematic satellites.
- At the end of the file, a list of supported `PROBLEM`/`ACTION` number combinations with the related explanation is given.
- The first problem type (`PROBLEM=0`) is used to indicate satellite *maneuvers*. The processing programs read this file and use the orbit of satellite with number PRN before and PRN+50 after the maneuver epoch. The satellite with number PRN+50 is only present in the orbit files, but not in the observation files. ORBGEN treats the "new" satellite as any other satellite. For maneuvers the action number is always `0`.
- If the maneuver epoch is accurately determined, the data before and after the maneuver are usable without problems. Only a few observations around the maneuver epoch (from few to several minutes) may have to be deleted using problem type `3` (the residuals of MAUPRP may be used for a refinement of this time interval).
- The problem type *satellite modeling* (`PROBLEM=4`) is used for long–arc computations using program ADDNEQ2 ("Menu>Processing>Combine normal equation systems"). This problem type indicates to set up a new arc (`ACTION=0`) for the specified satellite at the specified time (only NEQ boundaries are allowed).
- Another action also indicated with `PROBLEM=4`, is to add stochastic pulses for specific satellites and epochs in ORBGEN. This action is related to `ACTION=4`. This kind of

defining pulses is equivalent to the definition in the input panel "ORBGEN 4: Parameter Selection" (see Section 16.3.1.1).

- If an orbit is not well supported by observations, it is preferable not to estimate stochastic pulses for this particular satellite. For this purpose the combination of PROBLEM=4 and ACTION=5 numbers indicates an interval where the program ADDNEQ2 removes all stochastic pulses for the specified satellite.
- The problem type *bad satellite data* (PROBLEMS=1, 2, or 3) is used to exclude data of a particular satellite from the processing. If you specify this file in RXOBV3 ("Menu >RINEX>Import RINEX to Bernese format>Observation files") you have the possibility to transfer them to the Bernese formats as marked observations (ACTION=1) or to remove them (ACTION=2; not transferred to the Bernese–specific format). The remove action is supported only in program RXOBV3. If observations for a satellite are not removed, orbit information has to be available for the processing.
- All further processing programs exclude the satellites indicated as bad (PROBLEMS=1, 2, or 3, independent from the ACTION number) from the processing.

  If the pre–processing program MAUPRP ("Menu>Processing>Phase preprocessing") is forced not to check the observations for a specific satellite during a certain interval, the related measurements are marked and excluded from further processing because no cycle slip and outlier detection has been performed.

```
SATELLITE PROBLEMS: MANEUVERS OR BAD OBSERVATION INTERVALS, BSW5.2  SAT_2013.CRX
--------------------------------------------------------------------------------

SATELLITE  PROBLEM    ACTION           FROM                    TO
  ***         *          *       YYYY MM DD HH MM SS  YYYY MM DD HH MM SS

   27         3          2       2013 03 14 00 00 00  2013 03 27 23 59 59
   27         3          2       2013 05 09 23 59 59  2013 05 30 23 59 59
   30         3          2       2013 03 29 00 00 00  2013 03 31 23 59 59
   30         3          2       2013 05 07 00 00 00  2013 05 10 23 59 59
   37         3          2       2011 07 13 00 00 00  2099 12 31 23 59 59   ??
  102         3          2       2013 07 01 00 00 00  2013 07 01 09 59 59
  108         3          2       2013 01 26 00 00 00  2013 01 29 23 59 59
...
  -26         4          0       2013 07 11 00 00 00
   26         0          0       2013 07 11 18 03 28
  -26         4          0       2013 07 12 00 00 00
...



PROBLEM DESCRIPTION    PROBLEM          ACTION DESCRIPTION          ACTION
-------------------    -------          ------------------          ------

SATELLITE MANEUVER        0             SET UP SAT. WITH SVN=SVN+50    0
SATELLITE MODELING        4             SPLIT UP ARCS IN ADDNEQ2       0
SATELLITE MODELING        4             ADD PULSES IN ORBGEN           4
SATELLITE MODELING        4             DELETE ALL PULSES IN ADDNEQ2   5
BAD PHASE DATA            1             MARK PHASE OBSERVATIONS        1
BAD PHASE DATA            1             REMOVE PHASE OBSERVATIONS      2
BAD CODE DATA             2             MARK CODE OBSERVATIONS         1
BAD CODE DATA             2             REMOVE CODE OBSERVATIONS       2
BAD PHASE AND CODE DATA   3             MARK BOTH OBSERVATION TYPES    1
BAD PHASE AND CODE DATA   3             REMOVE BOTH OBSERVATION TYPES  2

SIGNS ARE ONLY RELEVANT FOR LONGARC COMPUTATION USING ADDNEQ2:
+ SIGN: FORCE ADDNEQ2 TO PERFORM AN ARCSPLIT (=> NO LONGARC)
- SIGN: PREVENT ADDNEQ2 FROM DOING AN ARCSPLIT (=> FORCE LONGARC)
```

Figure 24.6.: Satellite problem file (part of example file SAT_2013.CRX). The files SAT_yyyy.CRX are available in the anonymous BSW ftp area (see Section 4.13).

- When reading precise orbit files, satellites indicated as bad (PROBLEMS=1, 2, or 3, independent from the ACTION number) are also not considered. In this way satellites may be excluded that shall not be processed at all.
- The file may be updated by DEFXTR (inserting arc splittings) and RESCHK (adding intervals for bad satellite data). If an arc split event with a negative satellite number occurs, no event is added (e.g., in case of weakly observed satellites or potential arc splittings close to a maneuver event).

24.3.6. Receiver Information File

Type:	ASCII
Directory:	${X}/GEN (UNIX) or %X%\GEN (Windows).
Extension:	—
Content:	Characteristics of different receiver types.
Created by:	User–defined. Available from BSW aftp server.
Used by:	Programs ATX2PCV ("Menu>Conversion>ANTEX to Bernese format"), CODSPP ("Menu>Processing>Code-based clock synchronization"), GPSEST ("Menu>Processing >Parameter estimation"), CLKEST ("Menu>Service> *Clock* tools>Epoch-wise clock interpolation"), GPSSIM ("Menu>Service>Generate simulated observation data"), BV3RXO ("Menu>RINEX>Export RINEX from Bernese format>Observation files"), ADDNEQ2 ("Menu>Processing>Combine normal equation systems"), NEQ2ASC ("Menu>Conversion >Normal equations (binary/ASCII)"), and SNX2NQ0 ("Menu>Conversion>SINEX to normal equations").
Example:	Figure 24.7 and ${X}/GEN/RECEIVER..

The receiver characterization file is used to correctly apply DCBs and to list which GNSS is supported by a certain receiver type.

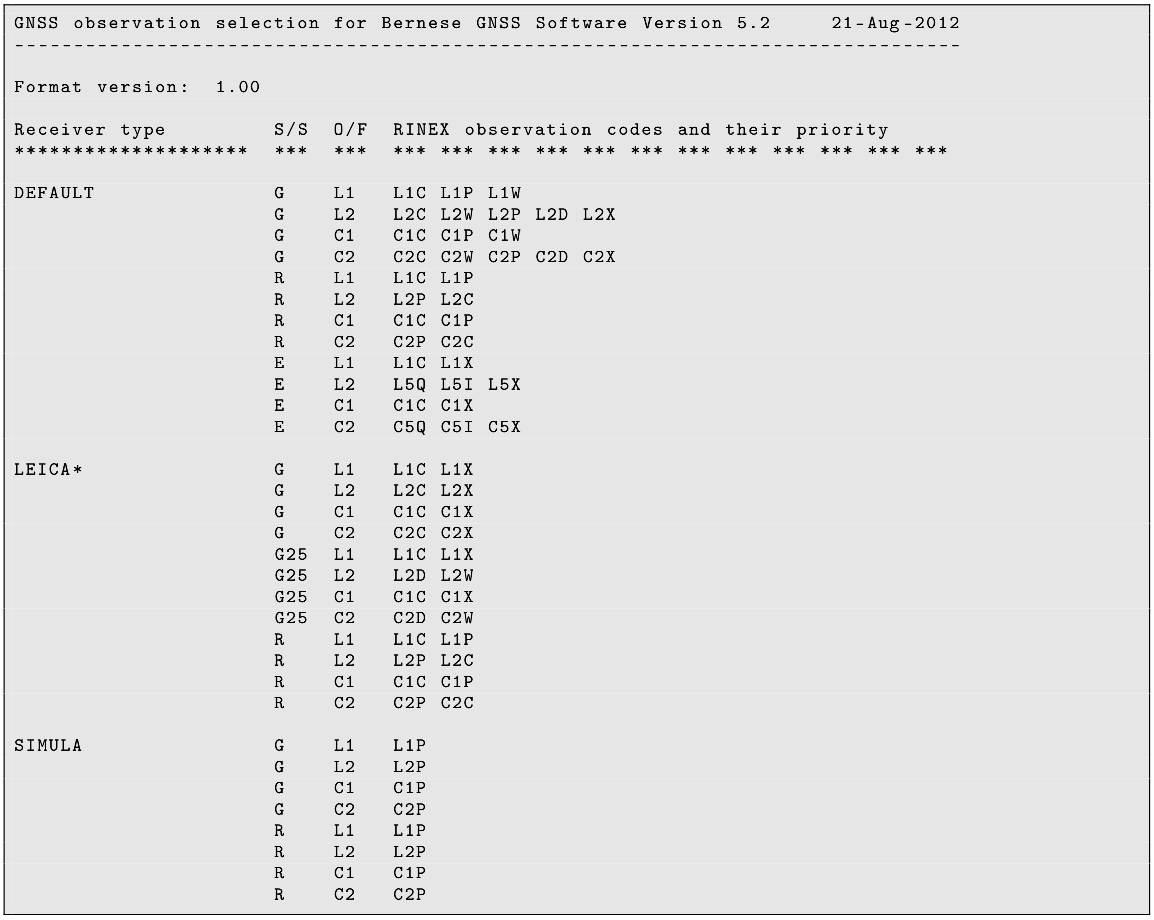
```
RECEIVER INFORMATION FILE, BERNESE GNSS SOFTWARE 5.2                 06-DEC-2013
--------------------------------------------------------------------------------

RECEIVER TYPE        #FREQ  CODE  FREQ  WAVE.F.  SYST
********************   *     **    L*:     *     ****
...

TOPCON GP-DX1          2     C1    L1:     1     G
                             X2    L2:     1

TOPCON GP-R1DY         2     P1    L1:     1     G
                             P2    L2:     1

TOPCON TT4000SSI       2     C1    L1:     1     G
                             X2    L2:     1

TPS EGGDT              2     P1    L1:     1     GR
                             P2    L2:     1

TPS EUROCARD           2     P1    L1:     1     GR
                             P2    L2:     1

...

REMARK:

  X2=C1+(P2-P1)

  G: GPS
  R: GLONASS
  E: GALILEO
  S: SBAS
```

Figure 24.7.: Receiver characterization file RECEIVER..

24.3.7. Observation Type Selection File

Type:	ASCII
Directory:	`${X}/GEN` (UNIX) or `%X%\GEN` (Windows).
Extension:	—
Content:	Priority list with RINEX3 observation codes for each of the both code and phase frequencies.
Created by:	User–defined. Available with the distribution of the *Bernese GNSS Software* Version 5.2 .
Used by:	Program RNXSMT ("Menu>RINEX>RINEX utilities>Clean/smooth observation files").
Example:	Figure 24.8 and `${X}/GEN/OBS.SEL`.

In column `Receiver type` in the observation type selection file, the receiver type is given with a 20–character string. The receiver type in the RINEX header must match with one of the receiver types in the observation type selection file. Otherwise, the settings of the `DEFAULT`–receiver are applied. Thus, it is recommended to define at least the `DEFAULT`–receiver in the file. If other observables or observations of other GNSS shall be used for certain receiver types, you may add special rules for these receivers. You may also use wildcards in order to define a common rule for a group of receivers, e.g., for all receivers of a certain manufacturer.



```
GNSS observation selection for Bernese GNSS Software Version 5.2     21-Aug-2012
--------------------------------------------------------------------------------

Format version:  1.00

Receiver type         S/S  O/F  RINEX observation codes and their priority
********************  ***  ***  *** *** *** *** *** *** *** *** *** *** *** ***

DEFAULT               G    L1   L1C L1P L1W
                      G    L2   L2C L2W L2P L2D L2X
                      G    C1   C1C C1P C1W
                      G    C2   C2C C2W C2P C2D C2X
                      R    L1   L1C L1P
                      R    L2   L2P L2C
                      R    C1   C1C C1P
                      R    C2   C2P C2C
                      E    L1   L1C L1X
                      E    L2   L5Q L5I L5X
                      E    C1   C1C C1X
                      E    C2   C5Q C5I C5X

LEICA*                G    L1   L1C L1X
                      G    L2   L2C L2X
                      G    C1   C1C C1X
                      G    C2   C2C C2X
                      G25  L1   L1C L1X
                      G25  L2   L2D L2W
                      G25  C1   C1C C1X
                      G25  C2   C2D C2W
                      R    L1   L1C L1P
                      R    L2   L2P L2C
                      R    C1   C1C C1P
                      R    C2   C2P C2C

SIMULA                G    L1   L1P
                      G    L2   L2P
                      G    C1   C1P
                      G    C2   C2P
                      R    L1   L1P
                      R    L2   L2P
                      R    C1   C1P
                      R    C2   C2P
```

Figure 24.8.: Observation type selection file `OBS.SEL`.

The column S/S contains the satellite or system designator. It might contain the GNSS system character according to the RINEX format description (e.g., G for GPS or R for GLONASS). The observation type priority might also be defined in a satellite–specific way. In this case the satellite PRN number must be given. Observations of satellites or GNSS not specified in this column are excluded from the further processing. Thus the OBS.SEL file might be used for satellite system selection.

The column O/F designates the four signal “channels” used within the *Bernese GNSS Software* (two phase channels L1 and L2 and two code channels C1 and C2).

The remaining columns contain the observation types in RINEX3–notation (three characters). Only observation types listed in one line can be selected for the corresponding receiver type, satellite (–system) and observation channel. The priority is descending to the right side of the table. Signals of different frequencies are not permitted to be mixed within the same line (e.g., L5Q and L7Q). Observation type designators describing the RINEX2 observation types (e.g., L1W for L1 or C1W for P1) may be used if RINEX2 files should be processed in the same way as the RINEX3 files (e.g., in a common campaign).

24.3.8. Frequency Information File

Type:	ASCII
Directory:	${X}/GEN (UNIX) or %X%\GEN (Windows).
Extension:	FRQ
Content:	Provides the wavelength/frequencies of the laser instrument for SLR tracking stations.
Created by:	User–defined. Available from BSW aftp server. Updated by RXOBV3 ("Menu>RINEX>Import RINEX to Bernese format>Observation files").
Used by:	Used by programs RXOBV3, CODSPP ("Menu>Processing>Code-based clock synchronization"), GPSEST ("Menu>Processing>Parameter estimation"), and GPSSIM ("Menu>Service>Generate simulated observation data").
Example:	Figure 24.9 and ${X}/GEN/FREQINFO.FRQ.

The frequency information file contains information such as observation type (for the time being only RANGE), station name, and wavelength.

```
Frequencies of observation types                                               28-AUG-14 16:30
-----------------------------------------------------------------------------------------------

Type   Satellite/Station    Name     Value      Unit  Valid from           to
*****  ********************  **  ***********.*  ****  **** ** ** ** ** **  **** ** ** ** ** **
RANGE  1824 12356S001        R1           532.0  nm
RANGE  1831 12368S001        R1           532.0  nm
RANGE  1863 12340S001        R1           532.0  nm
RANGE  1864 12340S002        R1           532.0  nm
...
RANGE  7105 40451M105        R1           532.0  nm
RANGE  7105 40451M105        R2          1064.0  nm
...
RANGE  7918 40451M120        R1           532.0  nm
RANGE  7920 40451M117        R1           532.0  nm
RANGE  7939 12734S001        R2           694.3  nm
RANGE  7941 12734S008        R1           532.0  nm
RANGE  8833 13504M002        R1           532.1  nm
RANGE  8834 14201S018        R1           532.0  nm
RANGE  8834 14201S018        R2          1064.0  nm
```

Figure 24.9.: Frequency information file for SLR processing.

24.3.9. Leap Seconds

Type: ASCII
Directory: `${X}/GEN` (UNIX) or `%X%\GEN` (Windows).
Extension: —
Content: List of leap seconds.
Created by: User–defined. Available from BSW aftp server.
Used by: Diverse orbit and processing programs.
Example: Figure 24.10 and `${X}/GEN/GPSUTC.`.

Remarks:

- The file has to be updated each time when a leap second is introduced.
- Most programs take the leap second information from the pole file.
- The information from the file is essential when processing SLR data together with GNSS measurements because SLR observations are taken in UTC . In related output files (e.g., orbit files) the time system for the data records may be selected. For the conversion the information from this file is necessary.
- If external information (e.g., LEO attitude) is given in UTC, a conversion into GPS time is needed to process the GNSS measurements.

```
DIFFERENCE OF GPS-TIME MINUS UTC-TIME
-------------------------------------
GPS-UTC     VALID SINCE
 (SEC)      YYYY MM DD HH MM SS.SS

...
   6.       1990 01 01 00 00 00.00
   7.       1991 01 01 00 00 00.00
   8.       1992 07 01 00 00 00.00
   9.       1993 07 01 00 00 00.00
  10.       1994 07 01 00 00 00.00
  11.       1996 01 01 00 00 00.00
  12.       1997 07 01 00 00 00.00
  13.       1999 01 01 00 00 00.00
  14.       2006 01 01 00 00 00.00
  15.       2009 01 01 00 00 00.00
  16.       2012 07 01 00 00 00.00
```

Figure 24.10.: The leap second file `GPSUTC.`.

24.3.10. Pole Offsets for the C04 and Rapid Pole Series

Type: ASCII
Directory: `${X}/GEN` (UNIX) or `%X%\GEN` (Windows).
Extension: —
Content: Pole offsets for the C04 combined pole series (the C04 pole series is based on a reference system different from the ITRF realizations).
Created by: User–defined. Transformation parameters obtained from the IERS annual reports (usually Table II-3).
Used by: POLUPD ("Menu>Orbits/EOP>Handle EOP files>Convert IERS to Bernese Format").
Example: Figure 24.11.

This file contains the pole offset information that is used to transform C04 pole and Rapid pole information to the actual epoch. Until the revision of the generation of C04 pole and Rapid pole information by the IERS in 1997, this file had to be update every year to introduce the new "constants" given in the annual report of the IERS. Today

```
POLE OFFSET COEFFICIENTS IERS EOP SERIES -- IERS REFERENCE FRAMES     17-MAR-94
--------------------------------------------------------------------------------
REFERENCE TIME (YYYY MM DD HH.H): 1988 00 00 00.0

      X     RMSX     Y     RMSY    UT1    RMSU   DPSI    RMS    DEPSI   RMS
     MAS    MAS     MAS    MAS    0.1MS  0.1MS   MAS    MAS     MAS    MAS
**  *****   ****   *****   ****   *****   ****  *****   ****   *****   ****
A    -0.10  0.40    0.70  0.40   -0.60  0.20   -0.20  0.01    0.30  0.01
A'    0.08  0.04    0.15  0.04    0.04  0.03    0.00  0.00    0.00  0.00

REF: 1992 IERS ANNUAL REPORT, Table II-3.
```

Figure 24.11.: Pole offset file in Bernese format. The values are valid for the transformation of the C04 pole to the ITRF 94 realization of the terrestrial reference frame.

the pole offsets are *no longer required* because the IERS pole series are kept consistent with the ITRF in near–realtime based on measurements provided by the space geodetic techniques.

If you are using IGS pole information or the C04 or the Rapid pole obatined from the anonymous ftp account of AIUB (`http://www.aiub.unibe.ch/download/`) (see Section 4.13) you do not need this file.

24.3.11. Nutation Model

Type:	ASCII
Directory:	`${X}/GEN` (UNIX) or `%X%\GEN` (Windows).
Extension:	`NUT`
Content:	Nutation model coefficients.
Created by:	User–defined.
Used by:	All programs writing and using EOP and/or orbit information.
Example:	Figure 24.12 and `${X}/GEN/IAU2000R06.NUT`.

The nutation file `IAU2000R06.NUT` contains the coefficients for luni–solar and planetary nutation terms as well as for the computation of the nutation arguments in accordance with the IERS Conventions 2010 (Petit and Luzum 2010).

In order to increase efficiency, the nutation matrix is buffered in a table with tabular interval of 1 hour. Values in this interval are interpolated linearly. The nutation model file has to be consequently used for all programs starting with program PRETAB.

For compatibility reasons the coefficients of older models namely `IAU2000.NUT` and `IAU80.NUT` are also distributed with the *Bernese GNSS Software* Version 5.2. It is essential to use the same nutation model file together with a certain Bernese pole file all the time (in case of an inconsistency a message is issued).

```
IAU 2000 NUTATION MODEL INCLUDING PLANETARY NUTATION
--------------------------------------------------------------------------------

Nutation model name: IAU2000R06


Precession corrections to IAU 1976 precession model:
---------------------------------------------------

Offset in longitude (")      :   -0.0417750           at J2000.0
Offset in obliquity (")      :   -0.0068192
Correction in longitude ("/c):   -0.29965
Correction in obliquity ("/c):   -0.02524


Fundamental arguments:
---------------------

ARG          A0                  A1                  A2                  A3                  A4                R
            (")                 ("/c)               ("/c**2)            ("/c**3)            ("/c**4)

L     485868.2490360000   715923.2177999020       31.8792000000        0.0516350000       -0.0002447000     1325.
L'   1287104.7930500000  1292581.0480999947       -0.5532000000        0.0001360000       -0.0000114900       99.
F     335779.5262320000   295262.8478000164      -12.7512000000       -0.0010370000        0.0000041700     1342.
D    1072260.7036900001  1105601.2090001106       -6.3706000000        0.0065930000       -0.0000316900     1236.
O     450160.3980360000  -482890.5431000004        7.4722000000        0.0077020000       -0.0000593900       -5.

LQ    908103.2597768833   261628.6889778376        0.0000000000        0.0000000000        0.0000000000      415.
LV    655127.2830690601   712136.4335481822        0.0000000000        0.0000000000        0.0000000000      162.
    .
    .
PA           0.0000000000     5029.0969397151        1.1111299474        0.0000000000        0.0000000000        0.


     MULTIPLIERS OF FUNDAMENTAL ARGUMENTS                       PERIOD                    LONGITUDE                                   OBLIQ...
                                                                (days)      LS        LS'         LC         LC'        OC        OC'      ...
   L   L'  F   D   O  LQ  LV  LE  LM  LJ  LS  LU  LN  PA                   (mas)      (mas/c)    (mas)      (mas/c)    (mas)      (mas/c)  ...

   0   0   0   0   1   0   0   0   0   0   0   0   0   0     -6798.383 -17206.4161   -17.4666     3.3386     0.0029  9205.2331     0.9086  ...
   0   0   2  -2   2   0   0   0   0   0   0   0   0   0       182.621  -1317.0906    -0.1675    -1.3696     0.0012   573.0336    -0.3015  ...
   0   0   2   0   2   0   0   0   0   0   0   0   0   0        13.661   -227.6413    -0.0234     0.2796     0.0002    97.8459    -0.0485  ...
   0   0   0   0   2   0   0   0   0   0   0   0   0   0     -3399.192    207.4554     0.0207    -0.0698     0.0000   -89.7492     0.0470  ...
```

Figure 24.12.: Nutation model file `IAU2000R06.NUT`.

24.3.12. Subdaily Pole Model

Type:	ASCII
Directory:	`${X}/GEN` (UNIX) or `%X%\GEN` (Windows).
Extension:	`SUB`
Content:	Subdaily pole model coefficients.
Created by:	User–defined.
Used by:	All programs writing and using orbit information.
Example:	Figure 24.13 and `${X}/GEN/IERS2010XY.SUB`.

The periodic variations of the Earth's rotation due to tidal deformation are computed in subroutine `${LG}/UT1RED.f` for the periods between 5 and 35 days. The coefficients conform with IERS Conventions 2010 (Petit and Luzum 2010). Coefficients for tidal variations in rotation and pole coordinates with daily and subdaily periods caused by ocean tides are specified in the subdaily pole model files. They are provided by the file `IERS2010XY.SUB`.

The efficiency of the computation of subdaily corrections is increased by buffering and linear interpolation within 5 minute time intervals. The subdaily pole model file has to be used consequently for all programs starting with program PRETAB.

In *Bernese GNSS Software* Version 5.2 , alternative subdaily pole model files `RAY_96.SUB` (McCarthy 1996) and `IERS2000.SUB` (McCarthy and Petit 2004) are distributed as well for compatibility reasons. It is essential to use the same subdaily pole model file all the time together with a certain Bernese pole file (in case of an inconsistency a message is issued).

```
IERS Conventions 2010, based on Ray 1994, XY-values
--------------------------------------------------------------------------------
FORMAT VERSION: 2
SUBDAILY ERP MODEL NAME: IERS2010


FUNDAMENTAL ARGUMENTS:
---------------------

ARG           A0                   A1                  A2                  A3               ...
             (")                 ("/C)              ("/C**2)            ("/C**3)            ...

L      485868.2490360000   715923.2178000000       31.8792000000        0.0516350000        ...
L'    1287104.7930500000  1292581.0481000000       -0.5532000000        0.0001360000        ...
F      335779.5262320000   295262.8478000000      -12.7512000000       -0.0010370000        ...
D     1072260.7036900001  1105601.2090000000       -6.3706000000        0.0065930000        ...
O      450160.3980360000  -482890.5431000000        7.4722000000        0.0077020000        ...
TP    1657658.2261500000     2772.1929900000        1.3965600000       -0.0000930000        ...


   FUNDAMENTAL ARGUMENTS    PERIOD             X                   Y                   UT
                            (DAYS)       (0.001 MAS)         (0.001 MAS)          (0.001 MS)
   L   L'  F   D   O   T                 XCOS       XSIN       YCOS       YSIN       UTCOS     UTSIN

  -1   0  -2  -2  -2   1    1.2113611    0.9        0.0       -0.1       -0.9       -0.08      0.40
  -2   0  -2   0  -1   1    1.1671262    0.6        0.1        0.1       -0.6       -0.06      0.19
  -2   0  -2   0  -2   1    1.1669259    3.4        0.3        0.3       -3.4       -0.31      1.03
   0   0  -2  -2  -1   1    1.1605476    0.8        0.1        0.1       -0.8       -0.07      0.22
   0   0  -2  -2  -2   1    1.1603495    4.2        0.5        0.5       -4.1       -0.39      1.19
  -1   0  -2   0  -1   1    1.1196993    5.0        1.2        1.2       -5.0       -0.47      0.97
  ...
```

Figure 24.13.: Subdaily pole model file `IERS2010XY.SUB` .

24.3.13. Geopotential Coefficients

Type:	ASCII
Directory:	`${X}/GEN` (UNIX) or `%X%\GEN` (Windows).
Extension:	—
Content:	Earth potential coefficients.
Created by:	User–defined.
Used by:	Program ORBGEN ("Menu>Orbits/EOP>Create/update standard orbits").
Example:	Figure 24.14 and `${X}/GEN/EGM2008_SMALL.`.

The model EGM2008 is the one recommended in the latest IERS Conventions 2010. It replaces the EGM96 model, as recommended in the previous IERS Conventions 2003. Other (older) models are distributed in Version 5.2 , such as GEMT3, GEM10N, JGM3, EIGEN2 and TEG4. Geopotential model files are read by the subroutine ${LG}/GETPOT.f with their respective format. For each model a flag indicating "tide–free" or "zero–tide" is defined.

Note that EGM2008 is provided in a reduced form, up to degree and order 200 only, as opposed to 2160 for the full model, hence the name EGM2008_SMALL..

```
Pavlis, N.K., S.A. Holmes, S.C. Kenyon, and J.K. Factor:
An Earth Gravitational Model to Degree 2160: EGM2008,
presented at the 2008 General Assembly of the European Geosciences Union,
Vienna, Austria, April 13-18, 2008.


product_type                gravity_field
modelname                   EGM2008
earth_gravity_constant      0.3986004415E+15
radius                      0.63781363E+07
max_degree                  2190
errors                      calibrated
norm                        fully_normalized
tide_system                 tide_free

url                   http://earth-info.nima.mil/GandG/



key     L    M             C                       S                    sigma C            s..
end_of_head =================================================================================..
gfc     0    0    1.0d0                    0.0d0                    0.0d0               0.0d0
gfc     2    0   -0.484165143790815D-03    0.000000000000000D+00    0.7481239490D-11    0.000..
gfc     2    1   -0.206615509074176D-09    0.138441389137979D-08    0.7063781502D-11    0.734..
gfc     2    2    0.243938357328313D-05   -0.140027370385934D-05    0.7230231722D-11    0.742..
gfc     3    0    0.957161207093473D-06    0.000000000000000D+00    0.5731430751D-11    0.000..
gfc     3    1    0.203046201047864D-05    0.248200415856872D-06    0.5726633183D-11    0.597..
gfc     3    2    0.904787894809528D-06   -0.619005475177618D-06    0.6374776928D-11    0.640..
gfc     3    3    0.721321757121568D-06    0.141434926192941D-05    0.6029131793D-11    0.602..
gfc     4    0    0.539965866638991D-06    0.000000000000000D+00    0.4431111968D-11    0.000..
gfc     4    1   -0.536157389388867D-06   -0.473567346518086D-06    0.4568074333D-11    0.468..
gfc     4    2    0.350501623962649D-06    0.662480026275829D-06    0.5307840320D-11    0.518..
gfc     4    3    0.990856766672321D-06   -0.200956723567452D-06    0.5631952953D-11    0.562..
gfc     4    4   -0.188519633023033D-06    0.308803882149194D-06    0.5372877167D-11    0.538..
...
gfc   200  193    0.471563246223772D-09    0.498912413509514D-10    0.3571400271D-10    0.357..
gfc   200  194   -0.249556689781556D-09    0.267412268963774D-10    0.3582760079D-10    0.358..
gfc   200  195    0.548444296204805D-09   -0.481449840771276D-09    0.3594254403D-10    0.359..
gfc   200  196   -0.306132475736424D-09    0.664215532623157D-09    0.3605884930D-10    0.360..
gfc   200  197    0.223114768845739D-09    0.173848103788711D-09    0.3617653364D-10    0.361..
gfc   200  198   -0.552938163626523D-09    0.508102987826649D-09    0.3629561441D-10    0.362..
gfc   200  199    0.751012006100548D-09   -0.105951851512426D-08    0.3641610920D-10    0.364..
gfc   200  200    0.590824206838993D-10   -0.128589429556450D-09    0.3653803588D-10    0.365..
```

Figure 24.14.: The geopotential file EGM2008_SMALL..

24.3.14. Ocean Tides Model File

Type:	ASCII
Directory:	`${X}/GEN` (UNIX) or `%X%\GEN` (Windows).
Extension:	`TID`
Content:	Ocean tides model coefficients.
Created by:	User–defined.
Used by:	ORBGEN ("Menu>Orbits/EOP>Create/update standard orbits").
Example:	Figure 24.15 and `${X}/GEN/OT_FES2004.TID`.

There is a series of ocean tidal models distributed with the *Bernese GNSS Software* Version 5.2 .

```
 UTCSR OCEAN TIDE MODEL FROM SCHWIDERSKI AND INTERPOLATION/EXTRAPOLATION (PLUS TEG-2B)
 117 767  20  20
 0
 0.63781450000000E+07 0.10250000000000E+04 0.59742300000000E+25 0.43440000000000E+03-0.12300..
 0.00000000000000E+00-0.30750000000000E+00-0.19500000000000E+00-0.13200000000000E+00-0.10320..
-0.81400000000000E-01-0.75500000000000E-01-0.71400000000000E-01-0.68200000000000E-01-0.65100..
-0.60300000000000E-01-0.58400000000000E-01-0.56700000000000E-01-0.55300000000000E-01-0.54000..
-0.51200000000000E-01-0.49700000000000E-01
  55.565LP    20     0.27930000000000E-01 0.14709399847621E-03 0.18000000000000E+03 0.299000..
  55.575LP A   9    -0.28000000000000E-03 0.29418799695241E-03 0.00000000000000E+00 0.299000..
  55.765LP B   1     0.40000000000000E-04 0.76600359860474E-03 0.18000000000000E+03 0.299000..
  56.544SA A   1    -0.40000000000000E-04 0.25906843923209E-02 0.00000000000000E+00 0.299000..
  56.554SA    11    -0.49200000000000E-02 0.27377783907971E-02 0.00000000000000E+00 0.299000..
  56.556SA B   2     0.26000000000000E-03 0.27380399907945E-02 0.18000000000000E+03 0.299000..
  56.564SA C   1     0.50000000000000E-04 0.28848723892733E-02 0.18000000000000E+03 0.299000..
  57.355SSAB   8    -0.32000000000000E-03 0.48569087814630E-02 0.00000000000000E+00 0.299000..
  57.553SSAD   1    -0.12000000000000E-03 0.54755567815941E-02 0.00000000000000E+00 0.299000..
 ....
```

Figure 24.15.: Ocean tides file (`OT_FES2004.TID`).

24.3.15. Solid Earth Tide File

Type:	ASCII
Directory:	`${X}/GEN` (UNIX) or `%X%\GEN` (Windows).
Extension:	`TPO`
Content:	Parameters for the impact of solid Earth tides on the gravity potential.
Created by:	
Used by:	ORBGEN ("Menu>Orbits/EOP>Create/update standard orbits").
Example:	Figure 24.16 and `${X}/GEN/TIDE2000.TPO`.

```
# Frequency Dependence of Solid Earth Tidal Potential (Step1+2)
# ------------------------------------------------------------------------------
#
  SOLID EARTH TIDE MODEL: IERS2000
  ELASTICITY            : ELASTIC
#
  LOVE NUMBERS:
# ------------------------------------------------------------------------------
#  n  m   Re(knm)   Im(knm)    knm+
# ------------------------------------------------------------------------------
   2  0   0.29525   0.00000   -0.00087
   2  1   0.29470   0.00000   -0.00079
   2  2   0.29801   0.00000   -0.00057
   3  0   0.09300   0.00000    0.00000
   3  1   0.09300   0.00000    0.00000
   3  2   0.09300   0.00000    0.00000
   3  3   0.09400   0.00000    0.00000
# ------------------------------------------------------------------------------
  AMPLITUDES:
# ------------------------------------------------------------------------------
# Name  | t  s  h  p  N  ps     ip      op
# A6    |                     1e-12   1e-12
# ------------------------------------------------------------------------------
  2Q1     1 -3  0  2  0  0     -0.1     0.0
  sigma1  1 -3  2  0  0  0     -0.1     0.0
          1 -2  0  1 -1  0     -0.1     0.0
  Q1      1 -2  0  1  0  0     -0.7     0.1
  rho1    1 -2  2 -1  0  0     -0.1     0.0
          1 -1  0  0 -1  0     -1.3     0.1
  O1      1 -1  0  0  0  0     -6.8     0.6
  tau1    1 -1  2  0  0  0      0.1     0.0
  Ntau1   1  0 -2  1  0  0      0.1     0.0
          1  0  0 -1 -1  0      0.1     0.0
  LK1     1  0  0 -1  0  0      0.4     0.0
  NO1     1  0  0  1  0  0      1.3    -0.1
          1  0  0  1  1  0      0.3     0.0
  chi1    1  0  2 -1  0  0      0.3     0.0
          1  0  2 -1  1  0      0.1     0.0
  pi1     1  1 -3  0  0  1     -1.9     0.1
          1  1 -2  0 -1  0      0.5     0.0
  P1      1  1 -2  0  0  0    -43.4     2.9
          1  1 -1  0  0 -1      0.6     0.0
  S1      1  1 -1  0  0  1      1.6    -0.1
          1  1  0 -2 -1  0      0.1     0.0
          1  1  0  0 -2  0      0.1     0.0
          1  1  0  0 -1  0     -8.8     0.5
...
  Mstm    0  3 -2  1  0  0      0.1     0.2
  Mtm     0  3  0 -1  0  0      0.4     1.1
          0  3  0 -1  1  0      0.2     0.5
  Msqm    0  4 -2  0  0  0      0.1     0.2
  Mqm     0  4  0 -2  0  0      0.1     0.1
  N2      2 -1  0  1  0  0     -0.3     0.0
  M2      2  0  0  0  0  0     -1.2     0.0
# ------------------------------------------------------------------------------
```

Figure 24.16.: Example of solid Earth tide file.

24.3.16. Planetary and Lunar Ephemerides

Type:	binary
Directory:	`${X}/GEN` (UNIX) or `%X%\GEN` (Windows).
Extension:	`EPH`
Content:	Planetary and Lunar ephemerides.
Created by:	User–defined.
Used by:	ORBGEN ("Menu>Orbits/EOP>Create/update standard orbits").
Example:	—

The Development Ephemerides DE405 are available from JPL and have to be converted to binary format. For a description on how to obtain and prepare the ephemeris file see the file `${X}/DOC/README_JPL_EPH.TXT` available in the *Bernese GNSS Software* distribution.

```
$$ calculated using atmospheric tidal model Ray and
$$ Ponte, Annales Geophysicae (2003) v21; 1897-1910
$$ Farrell elastic Greens functions have been used
    0.000   90.000  0.0183  0.0761 -0.0787  0.0139  0.0077 -0.0309 -0.0001 -0.0056  0.0320 ...
    0.000   89.000  0.0247  0.0856 -0.0837  0.0160  0.0079 -0.0304 -0.0013 -0.0055  0.0317 ...
    0.000   88.000  0.0295  0.0938 -0.0847  0.0180  0.0078 -0.0299 -0.0014 -0.0057  0.0315 ...
    0.000   87.000  0.0349  0.1025 -0.0845  0.0201  0.0079 -0.0291 -0.0016 -0.0059  0.0316 ...
    0.000   86.000  0.0410  0.1123 -0.0839  0.0221  0.0077 -0.0284 -0.0019 -0.0063  0.0319 ...
    0.000   85.000  0.0468  0.1227 -0.0837  0.0238  0.0069 -0.0282 -0.0024 -0.0068  0.0326 ...
    0.000   84.000  0.0522  0.1338 -0.0839  0.0250  0.0055 -0.0284 -0.0033 -0.0074  0.0334 ...
    0.000   83.000  0.0556  0.1442 -0.0850  0.0253  0.0037 -0.0291 -0.0043 -0.0079  0.0342 ...
    0.000   82.000  0.0564  0.1532 -0.0875  0.0246  0.0015 -0.0302 -0.0054 -0.0080  0.0349 ...
  ...
```

Figure 24.17.: Example of a grid with atmospheric tide deformations.

24.3.17. Atmospheric Tide Deformations

Type: ASCII
Directory: `${X}/GEN` (UNIX) or `%X%\GEN` (Windows).
Extension: —
Content: Grid with atmospheric tidal loading corrections.
Created by: User–defined.
Used by: GRDS1S2 ("Menu>Service>Coordinate tools>Extract atmospheric tidal loading coefficients").
Example: Figure 24.17 and `${X}/GEN/s1_s2_def_ce.dat`.

The file contains gridded components for deformation from atmospheric tidal deformations, e.g., Ray and Ponte (2003) . They are evaluated by the program GRDS1S2 which extracts the corrections for each station by bi–linear interpolation. This information is then provided to the processing programs in an atmospheric tidal loading file (located in the `STA` directory of the campaign area, see Section 24.7.10).

The grid file may start with comment lines that have to be indicated by a `$` at the beginning of the line. The columns in the data part are defined as follows:

1. longitude of the grid point in degree
2. latitude of the grid point in degree
3. cosine term for S1, radial component in millimeter
4. sine term for S1, radial component in millimeter
5. cosine term for S2, radial component in millimeter
6. sine term for S2, radial component in millimeter
7. cosine term for S1, north component in millimeter
8. sine term for S1, north component in millimeter
9. cosine term for S2, north component in millimeter
10. sine term for S2, north component in millimeter
11. cosine term for S1, east component in millimeter
12. sine term for S1, east component in millimeter
13. cosine term for S2, east component in millimeter
14. sine term for S2, east component in millimeter

24.3.18. SINEX General Information File

Type: ASCII
Directory: `${X}/GEN` (UNIX) or `%X%\GEN` (Windows).
Extension: —

```
SINEX HEADER INFORMATION FILE                                   10-MAR-2004
------------------------------------------------------------------------
(REMARK: YES=1,NO=0 ; 2 EMPTY LINES AFTER EVERY INPUT GROUP)

AGENCY:
-------

       ***
--> :  XYZ


DATA:
-----

       ***
--> :  IGS


HEADER INFORMATION TO BE INCLUDED:
----------------------------------

+FILE/REFERENCE
*INFO_TYPE_________ INFO________________________________________________________
 DESCRIPTION        My agency/institute
 OUTPUT             One-session solution generated by RNX2SNX BPE
 CONTACT            My e-mail address
 SOFTWARE           Bernese GNSS Software Version 5.2
 HARDWARE           My computer
 INPUT              IGS/MGEX GNSS tracking data
-FILE/REFERENCE

+FILE/COMMENT
-FILE/COMMENT

+INPUT/ACKNOWLEDGMENTS
*AGY DESCRIPTION________________________________________________________________
 XYZ My agency/institute and its address
 IGS International GNSS Service
-INPUT/ACKNOWLEDGMENTS
```

Figure 24.18.: General SINEX information file `SINEX.`.

Content: General information to be included in SINEX and Troposphere SINEX output files (see Sections 4.5 and 9.4.9.1).

Created by: User–defined.

Used by: ADDNEQ2 ("Menu>Processing>Combine normal equation systems") and GPSEST ("Menu >Processing>Parameter estimation") to include general information into the SINEX file and Troposphere SINEX file.

Example: Figure 24.18 and `${X}/GEN/SINEX.`.

If no file is specified in "GPSEST 1.3: General Files" or "ADDNEQ2 1.1: General Files" no information concerning your institution is included in the SINEX file (not–given fields (`---`) are used instead). If you want to share your SINEX result files with others, modify this file by adding information regarding your institution. See Section 4.5 for more information on SINEX.

24.3.19. IONEX General Information File

Type: ASCII

Directory: `${X}/GEN` (UNIX) or `%X%\GEN` (Windows).

Extension: —

Content: General information file to be included in IONEX output files (see Chapter 13).

Created by: User–defined.

```
IONEX CONTROL FILE                                           21-FEB-2004
------------------------------------------------------------------------
(REMARK: YES=1,NO=0 ; 2 EMPTY LINES AFTER EVERY INPUT GROUP)

SATELLITE SYSTEM:
----------------
      ********************
--> : GNSS


AGENCY:
------
      ********************
--> : My agency/institute


MULTI-LINE DESCRIPTION (UP TO 30 LINES):
---------------------------------------
      ************************************************************
--> : My description text


OBSERVABLES USED:
----------------
      ************************************************************
--> : One-way carrier phase leveled to code


MULTI-LINE COMMENT (UP TO 30 LINES):
-----------------------------------
      ************************************************************
--> : TEC/RMS values in 0.1 TECU; 9999, if no value available


DCB COMMENT:
-----------
      ************************************************************
--> : DCB values in ns; zero-mean condition wrt satellite values


INFORMATION TO BE SAVED:
-----------------------
...

REQUESTED SNAPSHOTS:
-------------------
...

DATA GRID:
---------
...

ADDITIONAL OPTIONS:
------------------
...
```

Figure 24.19.: General IONEX information file `IONEX.`.

Used by: GPSEST ("Menu>Processing>Parameter estimation") and ADDNEQ2 ("Menu >Processing>Combine normal equation systems") to include general information into the IONEX files.

Example: Figure 24.19 and `${X}/GEN/IONEX.`.

You have to adjust and complete this file as soon as you want to create ionosphere map files in IONEX format (see Section 24.8.5). We recommend you to copy the example file `${X}/GEN/IONEX.` to, e. g., `${X}/GEN/IONEX.USR` and to edit the last–mentioned file. This file does not only contain "auxiliary" text information to be included in IONEX files but also important specifications to be defined by the user.

Let us briefly highlight all entries:

1. `SATELLITE SYSTEM`, the satellite system used, e. g., GPS, GLONASS, or GNSS.
2. `AGENCY`, the agency creating the IONEX file, e. g., AIUB.

3. MULTI-LINE DESCRIPTION is intended for a brief description of the technique used to derive the TEC data provided. A contact address is desirable.
4. OBSERVABLES USED, one–line specification of the observable(s) used in the TEC computation.
5. MULTI-LINE COMMENT, additional comment lines.
6. DCB COMMENT, DCB–related comment line (one line only).
7. INFORMATION TO BE SAVED: Here you may specify whether TEC MAPS (TEC values), RMS MAPS (RMS errors), DIFFERENTIAL CODE BIASES (DCB estimates for the GPS/GLONASS satellites and receivers) are to be included in the IONEX file. The entry DEFAULT EXPONENT defines in which unit the TEC and RMS values are given in the IONEX file. With -1, the *recommended* value, these values are given in units of 0.1 TECU.
8. At REQUESTED SNAPSHOTS, you may define for which times TEC maps (snapshots) should be computed and written to the IONEX file. Using 12 00 00, 12 00 00, 86400, you would get a single snapshot at 12:00:00 UT. Another setting might be: 01 00 00, 23 00 00, 7200. This would lead to 12 2–hourly snapshots, namely at 01:00:00, 03:00:00, 05:00:00, ..., 23:00:00 UT.
9. At DATA GRID, you have to define the geographical area and the spatial resolution of the grid where TEC and RMS values are reported. Note that the current setting is suited for global TEC maps, but *not* for regional maps.
10. Under ADDITIONAL OPTIONS, you might define (a) a "minimum TEC value" (TEC values below this threshold are set to this value), (b) a "maximum rms value" (TEC values with larger RMS errors are considered as unavailable), and (c) a "scaling factor for rms values" to scale (multiply) the RMS errors obtained by the least–squares adjustment. Since *negative* TEC is physically impossible, a threshold value of 0 TECU would make sense at (a).

Nevertheless, we advise the interested user to study the IONEX format specifications (Schaer et al. 1998) and to have a look at CODE's ionosphere–related web site http://www.aiub.unibe.ch/ionosphere.html (Schaer 1998).

24.3.20. Panel Update File List

Type: ASCII
Directory: ${X}/SKL (UNIX) or %X%\SKL (Windows).
Extension: UPD
Content: List of directories (wildcards allowed) for panels to be updated.
Created by: User–defined.
Used by: Menu for updating program panels ("Menu>Configure>Update program input files") and for changing the general keyword values ("Menu>Configure>Change general options").

```
/home/aiub/igsauto/GPSUSER/PAN/
/home/aiub/igsauto/GPSUSER/OPT/*
/home/aiub/fridez/GPSUSER/PAN/
/home/aiub/fridez/GPSUSER/OPT/PPP_*/
/home/aiub/bernese/GPSUSER/PAN/
/home/aiub/bernese/GPSUSER/OPT/*
/home/aiub/pppuser/GPSUSER/PAN/
/home/aiub/pppuser/GPSUSER/OPT/*
```

Figure 24.20.: Example for panel update file.

Example: Figure 24.20 and `${X}/DOC/EXAMPLE.UPD`.

Please have a closer look at the usage of the wildcards in the example of Figure 24.20: you may either specify all option directories of a certain user or a subset (like for the user `fridez`).

24.4. RINEX Data Files

In the *Bernese GNSS Software* the RINEX files are considered as raw data files. A full description of the RINEX files and the file naming convention is given in Section 4.2 . To import the RINEX files into Bernese binary observation, broadcast orbit, or meteo file format, they are expected in the `RAW` directory of the campaign area. One RINEX observation file per session and station is expected.

If you have to produce them first, the program CCRINEXO ("Menu>RINEX>Cut/concatenate RINEX files>Observation files", description in Section 4.2.5) can be used to concatenate a set of RINEX observation files first. In that case the original RINEX files have to be located in the `ORX` directory of the campaign. The merged files are put to the `RAW` directory by the program. Analogue procedures can be established to merge the navigation messages from different stations by the programs CCRINEXN ("Menu>RINEX>Cut/concatenate RINEX files >Navigation files (GPS)") for GPS and CCRINEXG ("Menu>RINEX>Cut/concatenate RINEX files>Navigation files (GLONASS)") for GLONASS (description in Section 4.10.3).

24.5. Bernese Observation Files

24.5.1. General Remarks

The binary observation information used in the *Bernese GNSS Software* is split into two parts and stored in independent files:

- a header file containing all the relevant information to identify and process the observations (station, receiver, satellites, etc.).
- an observation file with all the code, phase, or range observations.

The type of file (code/phase/range, zero/single, header/observations) is indicated by the extension, see Table 24.2. The file format is the same for all types. This separation of information in two files makes it easier and more efficient to, e.g., update resolved ambiguities, since in this case only the header files have to be read and written and not all the observation files, which are much larger than the header files.

SLR range observations are converted into the same binary format as GNSS microwave measurements.

No examples are supplied for the binary observation and header files. The ASCII image of an observation file consisting of header and observations contains, however, the full information of the binary version and will be described in detail below.

There are small differences between the formats of the observation files for the different versions of the *Bernese GNSS Software*. Backward compatibility is guaranteed, which means that you may, e.g., process Version 5.0 observation files with Version 5.2 , but not vice versa.

Table 24.2.: Extensions of Bernese observation files.

File Description	Ext
Code/ range header (binary)	CZH
Code/ range observations (binary)	CZO
Phase header (binary)	PZH
Phase observations (binary)	PZO
SLR range header (binary)	RZH
SLR range observations (binary)	RZO
Code single difference header (binary)	CSH
Code single difference observations (binary)	CSO
Phase single difference header (binary)	PSH
Phase single difference observations (binary)	PSO
Code range observations (ASCII)	FCZ
Phase ASCII observations (ASCII)	FPZ
SLR range observations (ASCII)	FRZ
Code single difference observations (ASCII)	FCS
Phase single difference observations (ASCII)	FPS

Program CHGHED ("Menu>Service>Bernese observation files>Change header") may be used to change the content of the header, while program SATMRK ("Menu>Service>Bernese observation files>Mark/delete observations") may modify observation flags and reset ambiguities.

24.5.2. Header and Observation Files

Because of the identical format of all the header and observation file types, these files are described together in one section.

Type: Binary
Directory: Campaign–specific directory `OBS`.
Extension: `CZH`/`CZO`, `PZH`/`PZO`, `RZH`/`RZO`, `CSH`/`CSO`, or `PSH`/`PSO`.
Content: Code/phase/range zero/single–difference observation and associated important information in the header files.
Created by: Programs RXOBV3 ("Menu>RINEX>Import RINEX to Bernese format>Observation files"), GPSSIM ("Menu>Service>Generate simulated observation data"), SNGDIF ("Menu>Processing>Create baseline files"), and OBSSPL ("Menu>Service>Bernese observation files>Split observation files").
Updated by preprocessing programs CODSPP ("Menu>Processing>Code-based clock synchronization"), MAUPRP ("Menu>Processing>Phase preprocessing"), and SATMRK ("Menu>Service>Bernese observation files>Mark/delete observations"); as well as GPSEST ("Menu>Processing>Parameter estimation") for resolved ambiguities.
Used by: All processing programs dealing with observation/header information and some of the service programs (e.g., service and conversion programs in "Menu>Service>Bernese observation files" and "Menu>Conversion>Observation files").
Example: ASCII image of a header file (first part) and observation file (second part) in Figure 24.21 for single–difference phase observations.

All observations are stored in a binary format. The program OBSFMT ("Menu>Conversion>Observation files>Binary to ASCII") and the edit and browse utilities in "Menu>Service>Bernese

```
 1  INTRO           :  RNX2SNX_102070
 2
 3  MEASUREMENT TYPE:  PHASE                      CREATED :  28-FEB-14 09:47
 4  REFERENCE EPOCH :  2010-07-26  0:00:00 (207)  MODIFIED:  28-FEB-14 09:55
 5
 6  # DIFFERENCES   :      1           FORMAT NUMBER      :      6
 7  # FREQUENCIES   :      2           SESSION IDENTIFIER :   2070
 8  # SATELLITES    :     51           SUBSESSION IDENTIF.:      0
 9  # EPOCHS        :   2880           OBS. INTERVAL (S)  :     30.000000
10  # FLAGGED EPOCHS:      0           REMARK NUMBER      :      0
11
12  STATION NAME    :  WTZR 14201M010              ZIM2 14001M008
13  OPERATOR NAME   :  AUTOMATIC                   GPSBASE
14  RECEIVER TYPE   :  LEICA GRX1200GGPRO          TRIMBLE NETR5
15  ANTENNA TYPE    :  LEIAR25.R3      LEIT        TRM59800.00     NONE
16  RECEIVER/ANTENNA:  999999  /  20031            999999  /  60369
17
18  CLOCK CORRECTION:  OFFSET PER EPOCH            OFFSET PER EPOCH
19
20  POS.ECCENTR. (M):   0.0000  0.0000  0.0710      0.0000  0.0000  0.0000
21                                                              Station1            Station2
22  SAT    #L1-OBS OK  #L1-OBS BAD  #L2-OBS OK  #L2-OBS BAD  Obstype1  Obstype2  Obstype1  ...
23    3         762          150          762          115
24   11         751           43          751           32
25   14         984           94          984           84
26   17        1018           73         1018           70
27  ...
28
29   AMB SAT EPOCH  WLF    L1-AMBIG.   CLUS    L2-AMBIG.   CLUS    L5-AMBIG.   CLUS
30     1   3      1  1/1       -753897.    1       -587457.    1       -166424.  148
31     2   3   2160  1/1       -753750.  153       -587438.  153       -166295.   25
32     3  11      1  1/1       -654971.  148       -510371.  148       -144583.  152
33     4  11   2581  1/1       -655239.  153       -510334.  153       -144888.  153
34     5  14      1  1/1       1871636.  148       1458413.  148        413240.  148
35     6  14   1427  1/1       1871402.  148       1458417.  148        413002.  148
36     7  14   2553  1/1             0.    7             0.    7             0.    7
37     8  14   2571  1/1       1871540.  153       1458369.  153        413188.   25
38     9  17      1  1/1       -367271.   10       -286190.   10        -81064.   10
39    10  17   1093  1/1       -367431.   10       -286197.   10        -81217.   10
40    11  17   1923  1/1             0.   11             0.   11             0.   11
41    12  17   2841  1/1       -367344.   12       -286197.   12        -81129.   12
42    13  19      1  1/1       -750109.  148       -584504.  148       -165588.  152
43    14  19    973  1/1             0.   14             0.   14             0.   14
44  ...
45
46  L1,L2 OBSERVATIONS:
47     OBS.N    TIME   F #S   PHASE (M)  FFS SA       ... DATE, FR+CL 1 (S), FR+CL 2 (S)
48         1   0:00:00   18         0.095S 2   3      ... 10-07-26  0.000000009  0.000000031
49                                  0.024S 2   3      ...
50         2   0:00:30   18       113.653  2   3      ... 10-07-26  0.000000009  0.000000031
51                                113.589  2   3      ...
52         3   0:01:00   18       223.186  2   3      ... 10-07-26  0.000000009  0.000000031
53                                223.116  2   3      ...
54         4   0:01:30   18       328.731  2   3      ... 10-07-26  0.000000009  0.000000031
55                                328.659  2   3      ...
56         5   0:02:00   18       430.278  2   3      ... 10-07-26  0.000000009  0.000000030
57                                430.211  2   3      ...
58         6   0:02:30   18       527.811  2   3      ... 10-07-26  0.000000009  0.000000031
59                                527.746  2   3      ...
60  ...
```

Figure 24.21.: Example of a single difference observation file (ASCII image), line numbers added.

observation files>Edit observation header file" and "Menu>Service>Bernese observation files>Browse observation data files" create an ASCII image of a binary file. In the binary format the header and the observations are stored in two different files. The program OBSFMT merges these two files into one ASCII image. Real values are rounded to millimeters and nanoseconds which should be below the measurement noise. Using program FMTOBS ("Menu >Conversion>Observation files>ASCII to binary") ASCII observation files can be transformed back to binary.

Description of the ASCII file (Figure 24.21, for a better explanation, line numbers are added):
Line Comment

1 Campaign name (ch*16); title (ch*53).

3 Measurement type: PHASE, CODE or RANGE; file creation date and time.

4 The reference epoch is the full second part of the first observation epoch in the file; file modification date and time (updated by programs changing the file).

6 Number of differences: 0 = zero–difference file, 1 = single–difference file; file format number (6 for version Version 5.2).

7 Number of frequencies: 1 or 2; session identifier (used in program SNGDIF to be sure that the files are from the same session, in program IONEST to arrange files in sessions, and in program GPSEST to know which files have to be correlated).

8 Total number of satellites in the file; the session file number is the last character of the session string.

9 The number of epochs may be different from the number of the observation epochs. The epoch of the last observation record in the file may be computed from the reference epoch, the number of epochs, and the observation interval.
Observation interval in seconds (sampling rates below 1 sec are not supported, at present).

10 Number of occurrences of an epoch flag (given in the RINEX format, e.g., in case of power failure).
Remark number: it may be used to mark a file. The remark number is not used in any program so far, but it is printed by GPSEST.

12 Station name(s) (ch*16).

13 Operator name(s) (ch*20).

14 Receiver type(s) (ch*20).

15 Antenna type(s) (ch*20).

16 Receiver and antenna serial number(s) (6 digits integer).

18 Type of clock correction computed for the receiver clock(s): POLYNOMIAL DEG n, OFFSET PER EPOCH, or NONE.

20 Position eccentricities in a local coordinate frame (north, east, up). The transfer programs store the vector from the marker on the ground to the antenna reference point (ARP). Usually the eccentricities in north and east direction are zero and the third eccentricity (up) is equal to the antenna height.

22 Listing (lines 23–26) for each satellite: satellite number (PRN), the number of good and marked observations for each frequency (L_1 or L_1&L_2), and tracking types if the observations are imported from RINEX3 .

29 List (lines 30–43) of ambiguities (only present in the case of a phase header file) containing the ambiguity number, the satellite number, the start epoch of the ambiguity, the wavelength factor(s) for one or both frequencies, the value of the ambiguities within the corresponding cluster, and the ambiguity cluster number. If two frequencies are available the wide–lane ambiguity (L_5) is stored (together with its cluster number), independently of the L_1 and L_2 ambiguities. If two ambiguities of the same frequency belong to the same ambiguity cluster (i.e., they have the same cluster number), they are *resolved* relative to each other (as double–difference ambiguity). See Chapter 8 for more details concerning ambiguities.

46 List of all observations with the following information:

- Internal observation number (with respect to the reference epoch)
- Observation epoch (full seconds only)
- Epoch flag (provided by the RINEX format, e.g., for power failure) (`F`)
- Number of satellites observed at this epoch (`#S`)
- For each satellite observed:
  - observation (in meters)

  - two flags possible (FF): S = cycle slip and/or M = marked observation
  - signal/noise ratio (S)
  - indicator for Selective Availability (SA)
  - satellite number (PRN)
- Date of observation
- Clock parameter for station 1 in seconds:
  - Zero–difference files: fractional second of observation epoch
  - Single–difference files: sum of fractional second of observation epoch + clock correction to GPS time for station 1
- Clock parameter for station 2 in seconds:
  - Zero–difference files: clock correction to GPS time for current epoch
  - Single–difference files: sum of fractional second of observation epoch + clock correction to GPS time for station 2

If available, the L_2–observations will follow in the next line.

59 A blank line is inserted into the formatted file whenever one or more observation epochs are missing or if the satellite constellation changes. Blank lines between the observations will be removed when the file will be transformed back into a binary file.

24.6. Orbit Related Files

This section describes file formats containing orbit information and orbit related files. Some of them are specific to SLR or LEO data processing. The file types are listed in Table 24.3.

24.6.1. Satellite Broadcast Messages

Type: ASCII
Directory: Campaign–specific directory ORB.
Extension: BRD
Content: Satellite broadcast messages.
Created by: Transformation program RXNBV3 ("Menu>RINEX>Import RINEX to Bernese format>Navigation files") and the checking program BRDTST ("Menu>Orbits/EOP >Broadcast orbits>Check broadcast orbits").
Used by: BRDTST, BRDTAB ("Menu>Orbits/EOP>Broadcast orbits>Create tabular orbits"), SATCLK ("Menu>Orbits/EOP>Broadcast orbits>Extract satellite clocks"), CODSPP ("Menu>Processing>Code-based clock synchronization"), and BV3RXN ("Menu>RINEX >Export RINEX from Bernese format>Navigation files").
Example: Figure 24.22 and ${X}/DOC/EXAMPLE.BRD.

The example file is truncated in the middle of the first message. The first record is a title line. Each message (containing 2·20 parameters: 20 for ephemerides and 20 for satellite clocks) is preceded by a record in which the satellite number and a sequence number for the messages of a satellite is contained. This sequence number is never used by the accessing programs, which means that different broadcast files may be merged into one file (by removing the title line of the following file). The values in the broadcast messages are explained in the header of the subroutine ${LG}/GTBRDC.f. A description of the message may also be found in van Dierendonck et al. (1978).

Table 24.3.: Orbit related files.

File Description	Extension	Description in Section
Files with orbit information		
Broadcast orbits	BRD	24.6.1
Precise orbits	PRE	24.6.2
Tabular orbits	TAB	24.6.3
Standard orbits (binary)	STD	24.6.4
Standard orbits (ASCII)	FSO	24.6.4
Derivatives w.r.t. orbit parameters (binary)	RPR	24.6.5
Derivatives w.r.t. orbit parameters (ASCII)	FPR	24.6.5
Improved orbit elements	ELE	24.6.6
Orbit related files		
EOPs in IGS/IERS format	IEP	24.6.7
EOPs in Bernese format	ERP	24.6.8
Geocenter coordinates	GCC	24.6.9
Satellite clock corrections	CLK	24.6.10
Receiver clock corrections	CLK	24.6.11
Differential Code Bias for satellites and receivers	DCB	24.6.12
Inter–system biases	ISB	24.6.13
Files for SLR data processing		
Inter–range vector file	IRV	24.6.14
Precise orbit in CPF orbit format	CPF	24.6.15
State vector file	STV	24.6.16
Files for processing LEO data		
Attitude data	ATT	24.6.17
LEO external aux. information	AUX	24.6.18
Accelerometer data	ACC	24.6.19
LEO maneuvers	MAN	24.6.20
LEO velocities (epoch–wise)	KVL	24.7.5

```
                                                          29-OCT-15 19:47
SVN-NUMBER=  1  MESSAGE-NR=  1
    0.186300000000000000D+04
    0.518400000000000000D+06
    0.265602528955664374D+08
    0.470826553646000023D-02
    0.962846828994999959D+00
    0.222425730697999979D+01
    0.480619513218000005D+00
...

    0.000000000000000000D+00
SVN-NUMBER=  1  MESSAGE-NR=  2
    0.186300000000000000D+04
    0.525600000000000000D+06
    0.265602524236969538D+08
...
    0.000000000000000000D+00
SVN-NUMBER=  2  MESSAGE-NR=  1
    0.186300000000000000D+04
    0.518400000000000000D+06
    0.265599995026732683D+08
...
```

Figure 24.22.: Broadcast messages (BRD File). 40 lines of information per message.

24.6.2. Precise Ephemerides in IGS Format

Type: ASCII
Directory: Campaign–specific directory `ORB`.
Extension: `PRE`
Content: Precise ephemerides.
Created by: CCPREORB ("Menu>Orbits/EOP>Concatenate/merge precise orbit files"), RXNPRE ("Menu>RINEX>Import RINEX to Bernese format>Navigation files to SP3"), STDPRE ("Menu>Orbits/EOP>Convert standard to precise orbits"), PREWEI ("Menu>Orbits/EOP >Set accuracy codes in precise orbits"), KINPRE ("Menu>Service>LEO utilities>Convert kinematic pos. to precise orbits"), and CPFSP3 ("Menu>Service>SLR utilities>Convert CPF to precise orbit files"). Official exchange format within IGS.
Used by: CCPREORB), PRETAB ("Menu>Orbits/EOP>Create tabular orbits"), ORBGEN ("Menu>Orbits/EOP>Create/update standard orbits"), PREWEI, ORBCMP ("Menu >Orbits/EOP>Compare orbits>Precise orbits"), SP3CPF ("Menu>Service>SLR utilities >Convert precise orbits to CPF files"), and IRV2STV ("Menu>Service>SLR utilities>Convert IRV/PRE to state vectors").
Example: Figure 4.8.

Files from external sources may also have the extensions `SP3`, `SP3d`, `EPH`. Apart from SP3c, older precise orbit file formats (SP1 and SP3) may be read and written by the *Bernese GNSS Software*. All the satellite positions in the precise files are given in an Earth–fixed reference frame.

SP1 positions of the satellites (obsolete).
SP3 positions in km (and optionally velocities in dm/s) and clock information of the satellites in microseconds.
SP3c positions in km (and optionally velocities) and clock information of the satellites in microseconds (the new format allows, e.g., for maneuver and prediction flags).

The recommended format is SP3c. Detailed information may be found in Remondi (1989) and at `ftp://ftp.igs.org/pub/data/format/sp3c.txt`.
The satellite–specific RMS values are given in lines 8–9, specifying the RMS in 2^x mm, $x = 5 - 14$ (accuracy codes). Satellites with accuracy codes zero or exceeding a user specified limit may be excluded in the program PRETAB ("Menu>Orbits/EOP>Create tabular orbits", "PRETAB 2: General Options"). Accuracy codes can also be set or changed by program PREWEI ("Menu>Orbits/EOP>Set accuracy codes in precise orbits") using a LST file written by ORBGEN. For more details on the satellite clock values (last column, `999999.999999` if not available) and their use, refer to Section 5.6.

24.6.3. Tabular Orbits

Type: ASCII
Directory: Campaign–specific directory `ORB`.
Extension: `TAB`
Content: Tabular satellite positions in the inertial frame B1950.0 or J2000.0.
Created by: PRETAB ("Menu>Orbits/EOP>Create tabular orbits"), BRDTAB ("Menu>Orbits/EOP >Broadcast orbits>Create tabular orbits").
Used by: ORBGEN ("Menu>Orbits/EOP>Create/update standard orbits").
Example: Figure 24.23 and `EXAMPLE.TAB`.

```
TABULAR EPHEMERIS DERIVED FROM PREC. EPH:  PR2010.207  SYSTEM J2000.0
CELESTIAL POLE OFFSET: IAU2000R06      BIAS     SUBDAILY POLE MODEL: IERS2010
LOADING CMC - OTLOAD: FES2004         / HARLOAD: 342 tides   - Y     ATLOAD: Ray_Ponte    ...
   7   26 2010  0.000000000000D+00  0.000000000000D+00  0.000000000000D+00
   7   26 2010  0.230000000000D+02  0.450000000000D+02  0.209540641867D-06
 0.900000000000D+03     96
     -0.506241737459D-01       0.126124410087D+00       0.474095536349D+00



   52
SVN       1

SVN       2

...

SVN     124

    1    0.154612953355279D+05    0.334405163828652D+04   -0.214316097394082D+05
    2   -0.237620218222412D+05    0.659559741330191D+03   -0.123182125462358D+05
    3    0.215967251888043D+05   -0.142150668086936D+05   -0.719292036591851D+04
    4   -0.247435763058587D+05   -0.906133034350745D+04   -0.100950179284864D+04
    5   -0.133163380933714D+05    0.993264736081350D+04   -0.207543501389673D+05
    6    0.225058195020617D+05   -0.970735689587141D+04   -0.102587020896925D+05
    7   -0.118981938013204D+05   -0.169474541908490D+05   -0.165399013362784D+05
...
```

Figure 24.23.: Tabular orbit information (TAB file).

The orbit source is specified in the title line from column 44 to 53 (PR2010.207 in Figure 24.23). This information is transferred to the standard orbit file by program ORBGEN. The following line contains information about the subdaily pole and nutation model used. The third line contains information about the center of mass corrections (CMC) settings. The next two lines contain start/end times. They are followed by the tabular interval and the (nominal) number of ephemeris points. The next line contains pole information (which is not used by the program system). In the following lines the number of satellites and the satellite numbers (PRN–numbers) are defined. Finally the satellite positions (in system B1950.0 or J2000.0) are given in km. Satellite and epoch belonging to a specific record are reconstructed from the record number which is the first item of each record. Records of satellites for which no positions exist for a certain time interval are not contained in the file.

24.6.4. Standard Orbits

Type: Binary
Directory: Campaign–specific directory ORB.
Extension: STD
Content: Standard orbit (Bernese orbit representation using sets of polynomials).
Created by: ORBGEN ("Menu>Orbits/EOP>Create/update standard orbits").
Used by: Some orbit programs for comparison (STDDIF ("Menu>Orbits/EOP>Compare orbits>Standard orbits") or STDELE ("Menu>Orbits/EOP>Compare orbits>Osculating elements")), orbit format conversion tools (STDPRE ("Menu>Orbits/EOP>Convert standard to precise orbits")), and most of the processing programs ("Menu >Processing"). Other programs like for instance ORBCMP ("Menu>Orbits/EOP >Compare orbits>Precise orbits"), KINPRE ("Menu>Service>LEO utilities>Convert kinematic pos. to precise orbits"), or QLRSUM ("Menu>Service>SLR utilities>Create quick-look residual statistics") use the orbit to provide additional information to orbit differences, satellite positions, or SLR residuals.

```
  -1
   2   1
  17
TITLE : RNX2SNX_102070: Generate GNSS standard orbit
CREATE: 28-FEB-14 11:33
INTEG2:      3600   10    21600   12
NUTSUB: IAU2000R06       IERS2010         BIAS
GRAVIT: EGM2008            12          MEANPOLE IERS2010
TIDPOT: IERS2000    ELAS STEP_1+2      POLTID   IERS2010 K20=0.30
OTIDES: ICGEM fes2004    IERS2003      XMIN     0.00000 DEG   8
JPLEPH: DE405
PLANET: JUPITER VENUS MARS
RELATV: PPN IERS1996 P
EMPIRI: DYX ONCE-PER-REV C061001  3
SHADOW: STEP SPHERE MOON
OTLOAD: FES2004          CMC: Y     HARLOAD: 342 tides
ATLOAD: Ray_Ponte        CMC: Y
ERPMOD: NONE
ANTTHR: NO
TIMSYS: GPS
     52     24     10
  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
 25 26 27 28 29 30 31 32101102103104105107108110111113114115117118119120
121122123124
PR2010.207
 55403. 2.
 55403.
 55403.041666666664
 55403.083333333336
 55403.125
 ...
 55403.916666666664
 55403.958333333336
 55404.
 2.655798358883985E+7
 0.004478823922334417
 0.9666313680525306
 1.4217973969882007
 0.9456761362752834
 4.924284926457779
 15752.758488378156
 2.6560358315524142E+7
 0.009616015091552266
 0.9389122918705551
 -2.7802764704382232
 -3.168337303537692
 5.676450465764465
 -17490.101010140977
 ...
```

Figure 24.24.: Standard orbits (STD/FSO file).

Example: Figure 24.24 (ASCII example).

A standard orbit contains all information needed for computation of position, velocity, and higher time derivatives for each satellite. The orbit is stored in form of polynomial coefficients (one set of coefficients for typically 1 hour in case of GNSS satellites). One standard orbit file may contain several arcs per satellite. For additional information we refer to Chapter 5. The format is binary. To transform it to ASCII and back to binary (e.g., to allow a transfer to a different computer platform) use the programs STDFMT and FMTSTD ("Menu>Conversion>Orbit files>Binary to ASCII" and "Menu>Conversion>Orbit files>ASCII to binary", default extension for the ASCII files is FSO). The latest version contains certain information about the models used to generate the standard orbit, thereby allowing for consistency checks.

24.6.5. Radiation Pressure Coefficient File

Type: Binary
Directory: Campaign–specific directory ORB.

Extension: RPR
Content: Radiation pressure coefficients and partial derivatives of the satellite positions with respect to the radiation pressure coefficients. The partial derivatives of the satellite positions with respect to the Keplerian elements referring to the beginning of the arc are contained in this file, as well.
Created by: ORBGEN ("Menu>Orbits/EOP>Create/update standard orbits").
Used by: Program GPSEST ("Menu>Processing>Parameter estimation") for orbit improvements, program ADDNEQ2 ("Menu>Processing>Combine normal equation systems") for long arc computation.
Example: Not given. The structure of this file is similar to the standard orbit file (e.g., more than one arc per satellite possible).

It is only necessary to generate an RPR file with the program ORBGEN if you want to proceed to an orbit improvement. In all other cases the STD files are sufficient for the orbit representation. For additional information see Chapter 16. The format is binary. To transform it to ASCII and back to binary (e.g., to allow a transfer to a different computer platform) use the programs STDFMT and FMTSTD ("Menu>Conversion>Orbit files >Binary to ASCII" and "Menu>Conversion>Orbit files>ASCII to binary", default extension for the ASCII files: FRP).

24.6.6. Osculating Orbital Elements

Type: ASCII
Directory: Campaign–specific directory ORB.
Extension: ELE
Content: A priori and estimated orbit parameters with formal uncertainties.
Created by: GPSEST ("Menu>Processing>Parameter estimation") and ADDNEQ2 ("Menu >Processing>Combine normal equation systems") in case of an orbit improvement. IRV2STV ("Menu>Service>SLR utilities>Convert IRV/PRE to state vectors") allows to write the osculating elements from a inter–range vector file (see Section 24.6.14).
Used by: ORBGEN ("Menu>Orbits/EOP>Create/update standard orbits") in the UPDATE mode and STDPRE ("Menu>Orbits/EOP>Convert standard to precise orbits") to derive approximated RMS values of the orbit quality for each satellite.
Example: Figure 24.25: In the example file, each arc is characterized by 18 orbital elements (6 Keplerian elements, 9 radiation pressure coefficient parameters, and 3 pseudo–stochastic parameters at the middle of the arc). ${X}/DOC/EXAMPLE.ELE is also available.

The orbit model description from the standard orbit used as a priori orbit in the current orbit improvement process is repeated at the beginning of the file. It is used to verify the setting in the program ORBGEN when updating the osculating elements in the file to a full standard orbit. More information are given in Chapter 16 .

The data part of the file contains the following information: each line starts with the characterization of the orbital parameter. The first column contains the a priori orbital parameters, column 2 and 3 the improved values and their RMS. An asterisk * indicates that a parameter was not estimated, ORBSYS 2 in the last column defines the orbital system (ORBSYS 1: B1950.0, ORBSYS 2: J2000.0), and the string C061001B characterizes, if present, the a priori model for radiation pressure and the orbit model

```
CODE RAPID 3-DAY SOLUTION, DAY 104, YEAR 2010
--------------------------------------------------------------------------------
FORMAT:      1    13
TITLE : Create Tabular Orbit File using Precise Ephemeris
CREATE: 15-APR-10 06:45
INTEGR:   1.00000   10  6.00000   12
NUTSUB: IAU2000          IERS2000         BIAS
GRAVIT: JGM-3              12          MEANPOLE IERS2003
TIDPOT: IERS2000    ELAS STEP_1+2      POLTID   IERS2003 K20=0.30
OTIDES: CSR3.0           IERS2003      XMIN     0.05000
JPLEPH: DE405
PLANET: JUPITER VENUS MARS
RELATV: PPN IERS1996 P
EMPIRI: DYX ONCE-PER-REV C061001  3
SHADOW: STEP SPHERE MOON
OTLOAD: FES2004          CMC: Y
--------------------------------------------------------------------------------
ARC-NUMBER          =  1 SATELLITE          =  1 TOSC=  55300.000000000000
---------------------------------------------------------------------------
A                   =  26570557.28940  26570557.11397 +-       0.004    ORBSYS 2
E                   =    0.0045259656    0.0045259661 +- 0.000000000    ORBSYS 2
I                   =    55.287196617    55.287196920 +- 0.000000007    ORBSYS 2
NODE                =    85.532182598    85.532182502 +- 0.000000012    ORBSYS 2
PERIGEE             =    48.855474798    48.855448222 +- 0.000002146    ORBSYS 2
ARG. OF LAT (START) =   111.384180744   111.384180777 +- 0.000000012    ORBSYS 2
D0                  =-0.985845120D-07-0.978499895D-07 +- 0.32546D-10    C061001B
Y0                  =-0.445458220D-09-0.486586855D-09 +- 0.11381D-10    C061001B
X0                  =-0.207717453D-09-0.139381326D-08 +- 0.19273D-10    C061001B
DC                  = 0.329890359D-13-0.475787930D-13 +- 0.13940D-11    C061001B
YC                  = 0.737523040D-14-0.602613561D-13 +- 0.13971D-11    C061001B
XC                  = 0.223539942D-08 0.732369306D-09 +- 0.67328D-10    C061001B
DS                  = 0.318221686D-13 0.231724166D-13 +- 0.13929D-11    C061001B
YS                  =-0.148625171D-13-0.145908604D-12 +- 0.13971D-11    C061001B
XS                  =-0.266977616D-08-0.104811997D-08 +- 0.59256D-10    C061001B
ARC-NUMBER          =  1 SATELLITE          =  2 TOSC=  55300.000000000000
---------------------------------------------------------------------------
A                   =  26557891.47688  26557891.45164 +-       0.002    ORBSYS 2
...
*** STOCHASTIC ORBIT PARAMETERS ***
--------------------------------------------------------------------------------
ARC-NUMBER          =  1 SATELLITE          =  1 TOSC=  55300.000000000000
---------------------------------------------------------------------------
 1 3  13 55300.50000  0.000000000D+00 0.486213259D-06    0.72917D-06
 2 3  13 55300.50000  0.000000000D+00-0.505589817D-05    0.45236D-06
 3 3  13 55300.50000  0.000000000D+00 0.120632225D-09    0.13971D-07
ARC-NUMBER          =  1 SATELLITE          =  2 TOSC=  55300.000000000000
---------------------------------------------------------------------------
 1 3  13 55300.50000  0.000000000D+00 0.161516380D-05    0.57757D-06
...
```

Figure 24.25.: File of a priori and estimated orbit parameters (ELE file).

used (orbit model flag B). Pseudo–stochastic orbit parameters are listed at the end of the file.

Note that ELE files generated with *Bernese GNSS Software* Version 5.0 cannot be integrated using Version 5.2 .

24.6.7. Pole File in IGS/IERS Format

Type: ASCII
Directory: Campaign–specific directory ORB.
Extension: IEP
Content: Pole coordinates, UT1−UTC, and nutation offsets together with statistical information.
Created by: Parameter estimation program GPSEST ("Menu>Processing>Parameter estimation") or ADDNEQ2 ("Menu>Processing>Combine normal equation systems") or by other IGS ACs.

```
VERSION 2
IGSFIN3D: C3 3-DAY SOLUTION DAY: 10099                                     14-APR-10 01:07
------------------------------------------------------------------------------------------..
NUTATION MODEL       : IAU2000                    SUBDAILY POLE MODEL: IERS2000
  MJD          X-P       Y-P    UT1UTC     LOD   S-X   S-Y  S-UT  S-LD  NR  NF  NT    X-RT   Y-RT ..
              E-6"      E-6"     E-7S  E-7S/D  E-6"  E-6" E-7S E-7S/D                    E-6"/D   ..
55294.00    -67807    334959    91140   12208     8     7     1     6 248   0  24    -435   2171 ..
55294.50    -68025    336044    85036   12896     6     5     3    10 248   0  24    -435   2171 ..
55295.00    -68242    337130    78587   13303     6     6     6    18 248   0  24    -238   2574 ..
55295.50    -68361    338417    71936   13915     5     5     7    20 248   0  24    -238   2574 ..
55296.00    -68480    339704    64978   14220     6     6     9    26 248   0  24     100   2441 ..
55296.50    -68430    340925    57868   14654     6     5    10    30 248   0  24     100   2441 ..
55297.00    -68380    342145    50541   14654     8     7    12    30 248   0  24     100   2441 ..
```

Figure 24.26.: Pole file (IEP) in IGS/IERS format.

Used by: Extraction programs POLUPD ("Menu>Orbits/EOP>Handle EOP files>Convert IERS to Bernese Format"), POLXTR ("Menu>Orbits/EOP>Handle EOP files>Concatenate IERS pole files"), exchange format for ERPs within the IERS and IGS.

Example: Figure 24.26 shows Earth rotation estimates in the IEP format stemming from a 3–days arc. The right–most columns listing the correlations and nutation estimates are not reproduced. ${X}/DOC/EXAMPLE.IEP is also available.

This file is is used to exchange polar motion, UT1−UTC, and nutation information from global ERP estimations with the IERS Bureau and within the IGS. Currently the version 2 is in use for all official IGS products (see IGS Mail 1943). The reading routine in the *Bernese GNSS Software* ${LG}/RDPOL.f supports 13 different pole file formats.

Files from external sources have the extension ERP. *In order to avoid confusion with Bernese ERP files, rename external files to names with extension IEP.*

24.6.8. Earth Rotation Parameters or Pole Coordinates in Bernese Format

Type: ASCII

Directory: Campaign–specific directory ORB.

Extension: ERP

Content: Pole coordinates, UT1−UTC, UTC−GPS, nutation offsets.

Created by: Download e.g. the file C04_yyyy.ERP or BULLET_A.ERP in Bernese ERP format from our anonymous ftp server (http://www.aiub.unibe.ch/

```
IERS C04 POLE                                                            15-JUL-14 12:55
------------------------------------------------------------------------------------..
NUTATION MODEL       : IAU2000R06                 SUBDAILY POLE MODEL: IERS2010
    DATE    TIME    X-POLE   Y-POLE    UT1-UTC  GPS-UTC     RMS XP   RMS YP    RMS DT    DE-CP..
YYYY MM DD HH MM      (")      (")       (S)      (S) REM      (")      (")      (S)       (")..

2014  1  1  0  0  0.03864  0.31887 -0.097051   16. C04   0.00006 0.00007 0.000006   0.000..
2014  1  2  0  0  0.03839  0.31956 -0.098242   16. C04   0.00007 0.00007 0.000010   0.000..
2014  1  3  0  0  0.03778  0.32036 -0.099504   16. C04   0.00007 0.00007 0.000004   0.000..
2014  1  4  0  0  0.03712  0.32076 -0.100853   16. C04   0.00008 0.00008 0.000009   0.000..
2014  1  5  0  0  0.03669  0.32119 -0.102259   16. C04   0.00006 0.00007 0.000008   0.000..
2014  1  6  0  0  0.03611  0.32188 -0.103667   16. C04   0.00005 0.00006 0.000006   0.000..
2014  1  7  0  0  0.03528  0.32279 -0.104994   16. C04   0.00006 0.00006 0.000002   0.000..
```

Figure 24.27.: Pole file (ERP) in Bernese Format.

download/BSWUSER52/ORB/). External ERP files may be imported by the program POLUPD ("Menu>Orbits/EOP>Handle EOP files>Convert IERS to Bernese Format"). The pole file may also be created as a result of a parameter estimation using programs GPSEST ("Menu>Processing>Parameter estimation") or ADDNEQ2 ("Menu>Processing>Combine normal equation systems").

Used by: All orbit programs and all processing programs.
Example: Figure 24.27 and ${X}/DOC/EXAMPLE.ERP.

ERP files are input files for most of the programs but they may also be created as output files from parameter estimation using GPSEST or ADDNEQ2. The first line below the title reports the used nutation and subdaily Earth rotation models. Always use the same pole file together with one and the same orbit file (standard orbit or tabular).

The pole file is accessed by the subroutine ${LG}/GETPOL.f. It is not important that the pole positions are given at equidistant time intervals. ${LG}/GETPOL.f checks for each request, however, that the spacing between the two data points used for interpolation is smaller than 10 days. The table values are linearly interpolated and a warning is given if a leap second occurred in the interpolation interval.

24.6.9. Geocenter Coordinates

Type: ASCII
Directory: Campaign–specific directory ORB.
Extension: GCC
Content: Geocenter coordinates.
Created by: ADDNEQ2 ("Menu>Processing>Combine normal equation systems").
Used by: ADDNEQ2.
Example: Figure 24.28 and ${X}/DOC/EXAMPLE.GCC.

The file contains estimated geocenter coordinates in X, Y, Z (columns 3–5) together with formal errors (columns 6–8) in meters for time intervals (Modified Julian Day (MJD), columns 1–2) as estimated by ADDNEQ2 (see Section 16.4.3). In general, it makes only sense to estimate geocenter coordinates when improving orbits in a global solution and applying a no–net rotation and no–net translation constraint. The results then give the coordinates of the geocenter as sensed by the satellite orbits with respect to the origin of the terrestrial reference frame in which the fiducial stations are defined.

```
 55274.00  55277.00       0.0101   0.0016   0.0032       0.0004   0.0004   0.0006
 55275.00  55278.00       0.0102   0.0021   0.0015       0.0004   0.0004   0.0005
 55276.00  55279.00       0.0103   0.0017  -0.0000       0.0004   0.0004   0.0005
```

Figure 24.28.: Geocenter coordinates (GCC file).

24.6.10. Satellite Clock Coefficients

Type: ASCII
Directory: Campaign–specific directory ORB.
Extension: CLK
Content: GNSS satellite clock parameters (extracted from broadcast messages, precise orbit files, clock RINEX files, or estimated).

```
SATELLITE CLOCKS FROM BROADCAST FILE        (ADAPTED BY CODXTR)  15-APR-10 10:36
--------------------------------------------------------------------------------

SAT WEEK   TOC    #PAR     A0 (SEC)       A1 (SEC/SEC)    A2 (SEC/SEC**2)

  2 1579  345600.  3   0.246495008D-03  0.363797881D-11  0.000000000D+00
  2 1579  352800.  3   0.246521086D-03  0.363797881D-11  0.000000000D+00
  2 1579  360000.  3   0.246547628D-03  0.363797881D-11  0.000000000D+00
  2 1579  367200.  3   0.246573705D-03  0.363797881D-11  0.000000000D+00
  2 1579  374400.  3   0.246599782D-03  0.363797881D-11  0.000000000D+00
  2 1579  381600.  3   0.246625859D-03  0.363797881D-11  0.000000000D+00
  3 1579  345600.  3   0.541965477D-03  0.511590770D-11  0.000000000D+00
  3 1579  352800.  3   0.542001799D-03  0.511590770D-11  0.000000000D+00
  3 1579  360000.  3   0.542038586D-03  0.511590770D-11  0.000000000D+00
  3 1579  367200.  3   0.542074908D-03  0.511590770D-11  0.000000000D+00
  3 1579  374400.  3   0.542111695D-03  0.511590770D-11  0.000000000D+00
...
 30 1579  352800.  3   0.233786181D-03  0.341060513D-11  0.000000000D+00
 30 1579  360000.  3   0.233810395D-03  0.341060513D-11  0.000000000D+00
 30 1579  367200.  3   0.233835075D-03  0.341060513D-11  0.000000000D+00
 30 1579  374400.  3   0.233859755D-03  0.341060513D-11  0.000000000D+00
```

Figure 24.29.: Satellite clock (CLK) file.

Created by: SATCLK ("Menu>Orbits/EOP>Broadcast orbits>Extract satellite clocks") for extraction from broadcast file, RNXCLK ("Menu>RINEX>RINEX utilities>Extract satellite clock") for extraction from clock RINEX, PRETAB("Menu>Orbits/EOP>Create tabular orbits") for extraction from precise ephemerides, GPSEST ("Menu>Processing >Parameter estimation"), ADDNEQ2 ("Menu>Processing>Combine normal equation systems"), and CLKEST ("Menu>Service> *Clock* tools>Epoch-wise clock interpolation") for reporting estimated clock corrections, CCRNXC ("Menu>Service> *Clock* tools >Combine/manipulate clock RINEX files") for providing the GNSS satellite clock corrections from the manipulation of clock RINEX files, CODSPP ("Menu >Processing>Code-based clock synchronization") for correcting GLONASS satellite clock offsets to GPS time.

Used by: Processing programs CODSPP ("Menu>Processing>Code-based clock synchronization"), MAUPRP ("Menu>Processing>Phase preprocessing"), GPSEST, ADDNEQ2, and CLKEST; Simulation program GPSSIM ("Menu>Service>Generate simulated observation data") as well as STDPRE ("Menu>Orbits/EOP>Convert standard to precise orbits") to get the satellite clocks for the resulting precise orbit file.

Example: Figure 24.29 and ${X}/DOC/EXAMPLE1.CLK.

Remarks:

- In CODSPP, a satellite clock file has to be specified if standard orbits are used as orbit information (no clock information stored in standard orbits).
- Precise high rate clocks are required if MAUPRP is used in the zero–difference mode.
- In GPSEST the GNSS satellite clock corrections are needed in the zero–difference mode.
- It is possible to store the clocks of several sessions in one file.
- If the number of polynomial coefficients is one, (#PAR=1) the clock reading routine does not interpolate for epochs with missing clock values. If an interpolation is necessary, it must explicitly be enabled by an input option (e.g., "Clock interpolation allowed over" in panel "GPSEST 3.3: General Options 3").
- Clock RINEX files may contain also receiver clock corrections. This file type is stored in the OUT directory of the campaign (see Section 24.11.8).
- When CODXTR ("Menu>Processing>Program output extraction>Code-based clock synchronization") is evaluating the program output from CODSPP, it may detect satellites involved in

maneuver or clock maintenance events. These satellite clock may be removed from a satellite clock file.

See Section 5.6 for more information.

24.6.11. Receiver Clock Coefficients

Type: ASCII
Directory: Campaign–specific directory `ORB`.
Extension: `CLK`
ATTENTION: same extension and directory as the satellite clock file.
Content: Receiver clock corrections (for simulation purposes only).
Created by: User–defined.
Used by: GPSSIM ("Menu>Service>Generate simulated observation data").
Example: Figure 24.30 and `${X}/DOC/EXAMPLE2.CLK`.

```
RECEIVER CLOCKS AS POLYNOMIALS OF DEGREE N-1
POLYNOMIAL COEFFICIENTS IN SEC, SEC/DAY, SEC/DAY**2,...
 STATION        N     A0       A1       A2       A3       A4       A5
CHASSERAL       6 -0.003002+0.001234-0.000000+0.010000-0.034556-0.023476
GENEROSO        6 +0.020000-0.010000+0.010000-0.005045-0.000345-0.123456
TITLIS          6 -0.013000+0.020000-0.001234+0.001000-0.006000-0.500000
ZIMMERWALD      6 +0.016017-0.000123+0.000222-0.002000-0.012345+0.000555
```

Figure 24.30.: Receiver clock corrections (for simulation only).

24.6.12. Differential Code Biases for Satellites and Receivers

Type: ASCII
Directory: Campaign–specific directory `ORB`.
Extension: `DCB`
Content: DCB values P1−P2 and P1−C1 for GPS/GLONASS satellites and receivers.
Created by: RXNBV3 ("Menu>RINEX>Import RINEX to Bernese format>Navigation files") to extract DCBs from navigation messages, RNXSMT ("Menu>RINEX>RINEX utilities >Clean/smooth observation files") to directly compute DCBs from RINEX observation files, CODSPP ("Menu>Processing>Code-based clock synchronization") to report the estimated ISB, GPSEST ("Menu>Processing>Parameter estimation"), and ADDNEQ2 ("Menu>Processing>Combine normal equation systems") to provide the results from a parameter estimation.
Used by: Programs CODSPP ("Menu>Processing>Code-based clock synchronization"), GPSEST, CLKEST ("Menu>Service> *Clock* tools>Epoch-wise clock interpolation"), and GPSSIM ("Menu>Service>Generate simulated observation data") to correct observations for the biases as well as ADDNEQ2 to modify a priori values in case of parameter estimation.
Example: Figure 24.31 and `${X}/DOC/EXAMPLE.DCB`.

Remarks:

- Five types of code biases are supported and are indicated by keywords above each data section:
    1. `P1-P2`: Differential Code Bias,

```
CODE'S 30-DAY GNSS P1-P2 DCB SOLUTION, ENDING DAY 191, 2014      15-JUL-14 07:46
--------------------------------------------------------------------------------

DIFFERENTIAL (P1-P2) CODE BIASES FOR SATELLITES AND RECEIVERS:

PRN / STATION NAME        VALUE (NS)  RMS (NS)
***   ****************    *****.***   *****.***
G01                          -8.540       0.012
G02                           8.018       0.007
...
G30                          -7.088       0.008
G31                           3.714       0.007
G32                          -2.889       0.006
R01                          -5.605       0.016
R02                          -0.036       0.008
...
R23                          -3.870       0.008
R24                           4.974       0.009
G     ABMF 97103M001        -10.423       0.093
R     ABMF 97103M001         -9.109       0.103
G     ABPO 33302M001         -6.201       0.038
G     ADIS 31502M001         -0.700       0.137
R     ADIS 31502M001        -10.118       0.173
...
```

Figure 24.31.: Differential P1−P2 code biases for GNSS satellites (DCB file).

2. P1-C1: Differential Code Bias,
3. LC: inter–system bias corresponding to the L_3 linear combination,
4. P2-C2: Differential Code Bias,
5. INTER-FREQ: inter–frequency bias when processing GLONASS measurements.

Combinations between these types are possible even within one file.

- GPS–related DCB values are indicated by a leading G, GLONASS–related values by a leading R. If such an indicator is missing (e. g., in *old* CODE P1−P2 DCB files), the DCB information is assumed to be GPS–related.
- In case of GPS/GLONASS–combined receivers, *two* receiver–specific bias values are provided, one related to GPS and one related to GLONASS. Note that a bias value specific to a receiver is addressed with the corresponding station name. The type of code biases is indicated by the string P1-P2 or P1-C1 in the header of the file.

More information concerning DCBs may be found in Chapter 14 or in Schaer (1999).

24.6.13. Inter–System Biases

Type: ASCII
Directory: Campaign–specific directory ORB.
Extension: ISB
Content: Inter–system biases between GPS and GLONASS with a variation in time.
Created by: Program GPSEST ("Menu>Processing>Parameter estimation" and ADDNEQ2 ("Menu >Processing>Combine normal equation systems".
Used by: GPSEST and ADDNEQ2.
Example: Figure 24.32 and ${X}/DOC/EXAMPLE.ISB.

24.6.14. Tuned Inter–Range Vector Format

Type: ASCII
Directory: Campaign–specific directory ORB.

```
ADD_Q1D: Q1 1-DAY SOLUTION 04122, AMB. FIXED                    22-APR-10 23:17
--------------------------------------------------------------------------------

 Station name           Epoch                    ISBias        RMS   Frq
 ****************       ********** ********    **.****    **.****    **
 LEIJ 14267M001         2004 05 01 00 00 00     0.1110     0.0235    L3
 LEIJ 14267M001         2004 05 01 02 00 00     0.4117     0.0216    L3
 LEIJ 14267M001         2004 05 01 04 00 00    -0.0870     0.0211    L3
 LEIJ 14267M001         2004 05 01 06 00 00    -0.2005     0.0210    L3
 LEIJ 14267M001         2004 05 01 08 00 00     0.0098     0.0208    L3
 LEIJ 14267M001         2004 05 01 10 00 00    -0.0711     0.0208    L3
 LEIJ 14267M001         2004 05 01 12 00 00    -0.2017     0.0210    L3
 LEIJ 14267M001         2004 05 01 14 00 00    -0.0907     0.0209    L3
 LEIJ 14267M001         2004 05 01 16 00 00    -0.1305     0.0212    L3
 LEIJ 14267M001         2004 05 01 18 00 00    -0.0820     0.0224    L3
 LEIJ 14267M001         2004 05 01 20 00 00     0.2300     0.0572    L3
 LEIJ 14267M001         2004 05 01 22 00 00     0.0781     0.0872    L3
 LEIJ 14267M001         2004 05 02 00 00 00     0.0061     0.0852    L3

 LHAZ 21613M002         2004 05 01 00 00 00     0.2014     0.0544    L3
 LHAZ 21613M002         2004 05 01 02 00 00    -0.4448     0.0528    L3
 ...
```

Figure 24.32.: Time variable inter–system biases between GPS and GLONASS (ISB file).

Extension: IRV
Content: Predicted satellite positions.
Created by: Downloaded from ftp://cddis.gsfc.nasa.gov/pub/slr/predicts/. The file has to be copied to a filename with the extension IRV.
Used by: IRV2STV ("Menu>Service>SLR utilities>Convert IRV/PRE to state vectors").
Example: Figure 24.33.

The Tuned Inter-Range Vector (IRV) format was used by the ILRS as a satellite prediction format until it was phased out in 2006 and replaced with predictions in the CPF format. More information on this format is available at http://ilrs.gsfc.nasa.gov/data_and_products/predictions/index.html.

24.6.15. Consolidated Prediction Format

Type: ASCII
Directory: Campaign-specific directory ORB.
Extension: CPF
Content: Predicted satellite positions.
Created by: SP3CPF ("Menu>Service>SLR utilities>Convert precise orbits to CPF files") or download via ftp from the data centers (CDDIS, EDC) or from the prediction centers.
Used by: CPFSP3 ("Menu>Service>SLR utilities>Convert CPF to precise orbit files").
Example: Figure 24.34.

The CPF is the file format for predicted positions for a large number of laser ranging targets as it is currently used be the ILRS. It is in use since June 30, 2006 and replaces the previously used Tuned IRV format. Predictions are provided in daily tables containing the geocentric positions for each target which can then be used to interpolate for precise predictions. Furthermore, the format is expandable for extended capabilities. More information is available at http://ilrs.gsfc.nasa.gov/data_and_products/predictions/index.html.

```
IRVSATSC     TUNED  IRVS
 1997  1  1  0  0  0.0    12022204.411872     -563174.866066   -2444781.505362
 1155     29      0          901.977360384    -2896.433617469     5241.120916809
    -44      23       0             3162.0     9014248.040444     3246.664659724
IRVSATSC     TUNED  IRVS
 1997  1  2  0  0  0.0    -7841935.206664    -2277903.476820    9098959.612579
 1155     29      0        -4509.776474283     2338.773680561    -3275.158060377
    -48      23       0             3159.0    -1020879.070905    -5446.160854099
IRVSATSC     TUNED  IRVS
 1997  1  3  0  0  0.0     -193679.111339     4250852.314719  -11574166.882065
 1155     29      0         5928.333870965     -667.469088560     -344.056495080
    -53      24       0             3156.0    -7516993.678685     4916.808287325
IRVSATSC     TUNED  IRVS
 1997  1  4  0  0  0.0     8106092.681370    -4224616.862241    8118630.077824
 1155     29      0        -4522.599089241    -1324.652773364     3797.470005155
    -57      25       0             3154.0    12000105.896953    -2049.781857450
IRVSATSC     TUNED  IRVS
 1997  1  5  0  0  0.0   -12014540.467295     2357293.755309    -970589.154413
 1155     29      0          933.545356845     2703.570464351    -5339.944231097
    -62      26       0             3151.0   -10627835.866399    -1702.828409902
IRVSATSC     TUNED  IRVS
 1997  1  6  0  0  0.0    10087312.697364      771991.118021   -7003478.237279
 1155     29      0         3154.291642824    -2853.110730247     4269.185575413
    -66      28       0             3150.0     3855825.578106     4570.366487991
IRVSATSC     TUNED  IRVS
 1997  1  7  0  0  0.0    -3328681.017752    -3492892.864552   11219575.439742
 1155     29      0        -5673.188439228     1692.469413366    -1146.824174394
    -71      29       0             3147.0     4398001.557438    -5127.543200255
IRVSATSC     TUNED  IRVS
 1997  1  8  0  0  0.0    -5054191.119463     4769259.959605  -10181417.464786
 1155     29      0         5408.863434471      292.241143631    -2561.025397686
    -75      30       0             3145.0   -10466348.624645     3140.079180416
IRVSATSC     TUNED  IRVS
 1997  1  9  0  0  0.0    10987543.108616    -3772980.401955    3875215.157968
 1155     29      0        -2558.485237991    -2178.807734308     5050.589688965
    -80      31       0             3142.0    11089777.864629      313.296716666
IRVSATSC     TUNED  IRVS
 1997  1 10  0  0  0.0   -11531752.993705     1112815.873728    4000236.346855
 1155     29      0        -1478.177055338     3050.948562414    -5025.066116169
    -84      33       0             3141.0    -6418700.773122    -3452.294609093
IRVSATSC     TUNED  IRVS
```

Figure 24.33.: Example of a tuned `IRV` file for LAGEOS–1 .

```
H1 CPF  1  COD 2015 08 10 01  7221 glonass124
H2  1004101 9124     37137 2015 08 09 23 59 43 2015 08 14 23 44 43   900 1 1  0 0 0
H5  1.8735
H9
10 0 57243  86383.000000  0   16993434.505  10340487.105  15992299.918
10 0 57244    883.000000  0   14915059.477   9953790.568  18152058.344
10 0 57244   1783.000000  0   12569425.247   9707262.498  19959551.682
10 0 57244   2683.000000  0   10012516.737   9635907.897  21379343.692
10 0 57244   3583.000000  0    7307519.058   9764250.827  22383468.284
10 0 57244   4483.000000  0    4522418.380  10105166.793  22952001.379
10 0 57244   5383.000000  0    1727401.523  10659219.373  23073479.354
10 0 57244   6283.000000  0   -1007838.530  11414532.327  22745153.702
10 0 57244   7183.000000  0   -3616779.054  12347205.818  21973074.590
10 0 57244   8083.000000  0   -6038613.737  13422262.132  20771999.363
10 0 57244   8983.000000  0   -8220561.044  14595083.380  19165125.612
10 0 57244   9883.000000  0  -10119836.587  15813281.944  17183652.276
10 0 57244  10783.000000  0  -11705205.226  17018925.217  14866176.075
10 0 57244  11683.000000  0  -12958046.287  18151020.072  12257934.548
.. . .....  ............  .  .............  ............  ............
99
```

Figure 24.34.: Example of an orbit prediction file in the `CPF` format.

24.6.16. State Vector File

Type: ASCII
Directory: Campaign-specific directory `ORB`.
Extension: `STV`
Content: State vector in inertial frame.

Created by: IRV2STV ("Menu>Service>SLR utilities>Convert IRV/PRE to state vectors").
Used by: —
Example: Figure 24.35.

The state vector file STV contains satellite state vectors (i.e., position and velocity) in the inertial reference frame.

```
# State vector in J2000.0
# TOSC        X                Y                Z             XDOT      ... COSPARID OBJID
# [MJD]      [m]              [m]              [m]            [m/s]     ...
56848.00  6148067.00654  -2152860.88813  10343418.75023   2299.41297 ... 76039A   L51_LEO
56848.00  5866549.46295   4823284.54917  -9720523.48186  -4129.45641 ... 92070B   L52_LEO
```

Figure 24.35.: Example of a STV for LAGEOS–1 and –2 .

24.6.17. Satellite Attitude for LEOs

Type: ASCII
Directory: Campaign–specific directory ORB.
Extension: ATT
Content: Satellite attitude information.
Created by: Program LEOAUX ("Menu>Service>LEO utilities>Prepare LEO auxiliary information") or original CHAMP, JASON or GRACE attitude files.
Used by: CODSPP ("Menu>Processing>Code-based clock synchronization"), MAUPRP ("Menu >Processing>Phase preprocessing"), GPSEST ("Menu>Processing>Parameter estimation"), and GPSSIM ("Menu>Service>Generate simulated observation data").
Example: Figure 24.36.

The attitude information provides the transformation from the LEO satellite body fixed coordinate system to inertial (J2000). The reading routine ${LG}/READATT.f90 may read original attitude files from CHAMP, JASON and GRACE, or this LEO satellite information may be converted into Bernese format using program LEOAUX. The program has to be executed with the command RUNGPS (see Section 21.9). The Bernese file is composed of a header including the integer MJD of the first epoch and of a data section containing the fractional day (column 1) and the attitude rotation matrix elements ordered by row (columns 2–9). Attitude for GPS and GLONASS satellites is hard–wired in the code.

```
* ATTITUDE FILE OF CHAMP 0003902 FOR DOY 150-2001           24-JUN-03 17:01
* -------------------------------------------------------------------------------------------
* Data derived from ${K}/JASON/ORB/CHA01150_12.AUX
* MJD                                                 attitude rotation matrix
*    Part of MJD                        first row
* -------------------------------------------------------------------------------------------
 52059.
  0.00000000000000  0.04616420851114 -0.48525053457856  0.87315564740210 -0.04845498091011  ...
  0.00011574073869  0.05591995226978 -0.48442487681705  0.87304381199357 -0.04250661535403  ...
  0.00023148148466  0.06569138988084 -0.48360788229990  0.87281616478788 -0.03654388796433  ...
  0.00034722222335  0.07547242410888 -0.48279728443165  0.87247389379643 -0.03057827702533  ...
  0.00046296296205  0.08520095043148 -0.48199448618883  0.87202185370906 -0.02459677408861  ...
  ...               ...               ...               ...               ...                ...
```

Figure 24.36.: Satellite attitude file from program LEOAUX .

24.6.18. Satellite Auxiliary Information for LEOs

Type: ASCII
Directory: Campaign–specific directory ORB.
Extension: AUX
Content: Auxiliary information for LEOs (attitude information, accelerometer data, maneuver information).
Created by: Mission–specific software to convert binary data into ASCII format.
Used by: LEOAUX ("Menu>Service>LEO utilities>Prepare LEO auxiliary information") to extract attitude information.
Example: Figure 24.37.

The program LEOAUX can handle the auxiliary information files of the missions CHAMP and GRACE. If data of other missions is to be converted, the subroutine ${LG}/SVLAUX.f90 needs to be extended.

```
PRODUCER AGENCY               : NASA
PRODUCER INSTITUTION          : JPL
FILE TYPE ipSCA1BF            : 6
FILE FORMAT 0=BINARY 1=ASCII  : 1
NUMBER OF HEADER RECORDS      : 23
SOFTWARE VERSION              : $Id: dat_fil_aux.tex,v 1.1 2015/11/25 15:40:12 dach Exp $
SOFTWARE LINK TIME            : @(#) 2011-09-14 11:54:29 rl02  g3
REFERENCE DOCUMENTATION       : GRACE Level 1 Software Handbook
SATELLITE NAME                : GRACE B
SENSOR NAME                   : IPU 4
TIME EPOCH (GPS TIME)         : 2000-01-01 12:00:00
TIME FIRST OBS(SEC PAST EPOCH): 221486400.000000 (2007-01-08 00:00:00.00)
TIME LAST OBS(SEC PAST EPOCH) : 221572795.000000 (2007-01-08 23:59:55.00)
NUMBER OF DATA RECORDS        : 17280
PRODUCT CREATE START TIME(UTC): 2011-11-29 01:10:33 by rl02
PRODUCT CREATE END TIME(UTC)  : 2011-11-29 01:10:33 by rl02
FILESIZE (BYTES)              : 2390648
FILENAME                      : GB-OG-1B-SCAATT+JPL-SCA1B_2007-01-08_B_02.asc
PROCESS LEVEL (1A OR 1B)      : 1B
INPUT FILE NAME               : SCA1A<-SCA1A_2007-01-08_B_02.dat
INPUT FILE TIME TAG (UTC)     : SCA1A<-2011-05-09 19:07:35 by rl02
INPUT FILE NAME               : CLK1B<-CLK1B_2007-01-08_B_02.dat
INPUT FILE TIME TAG (UTC)     : CLK1B<-2011-11-29 01:08:44 by rl02
END OF HEADER
221486400  B  4  4.4053164706145420e-01  4.2223746302294851e-01  7.5768528979749061e-01 ...
221486405  B  4  4.3844188638959447e-01  4.2286539020797420e-01  7.5890799539129117e-01 ...
221486410  B  4  4.3635324333397907e-01  4.2348157335350289e-01  7.6012985872348715e-01 ...
...
```

Figure 24.37.: Satellite auxiliary information for LEOs.

24.6.19. Satellite Accelerations for LEOs

Type: ASCII
Directory: Campaign–specific directory ORB.
Extension: ACC
Content: Non–gravitational accelerations of LEO, measured by accelerometers.
Created by: Program LEOAUX ("Menu>Service>LEO utilities>Prepare LEO auxiliary information").
Used by: For internal use only.
Example: Figure 24.38.

```
* ACCELERATION FILE OF GRACE 0201202 FOR DOY 343-2012       09-JAN-14 13:05
* ------------------------------------------------------------------------- ...
* Data derived from ${G}/GRRL02/ORB/GRCB12343.AUX
* SYSTEM
* MJD                                 linear accelerations                  ...
*    Part of MJD              R                  S                  W       ...
* ------------------------------------------------------------------------- ...
RSW
 56269.
   0.00000000000000  -0.00071577420059   0.00071980247488   0.01321447917747 ...
   0.00001157407678  -0.00071577908724   0.00071970007653   0.01321452417322 ...
   0.00002314814628  -0.00071578337109   0.00071959429914   0.01321456621839 ...
   0.00003472222306  -0.00071578692752   0.00071948558631   0.01321460268355 ...
...
```

Figure 24.38.: Satellite acceleration file from program LEOAUX.

24.6.20. Satellite Maneuvers for LEOs

Type: ASCII
Directory: Campaign–specific directory ORB.
Extension: MAN
Content: Maneuvers of LEOs.
Created by: Program LEOAUX ("Menu>Service>LEO utilities>Prepare LEO auxiliary information").
Used by: For internal use only.
Example: Figure 24.39.

```
  "920" "2011 02 03" "17 14 00" "5.0D-0" "5.0D-0" "5.0D-0"
  "920" "2011 02 03" "17 16 00" "5.0D-0" "5.0D-0" "5.0D-0"
  "920" "2011 02 03" "18 02 00" "5.0D-0" "5.0D-0" "5.0D-0"
  "920" "2011 02 03" "18 04 00" "5.0D-0" "5.0D-0" "5.0D-0"
```

Figure 24.39.: Satellite maneuver file for LEO.

24.7. Station Related Files

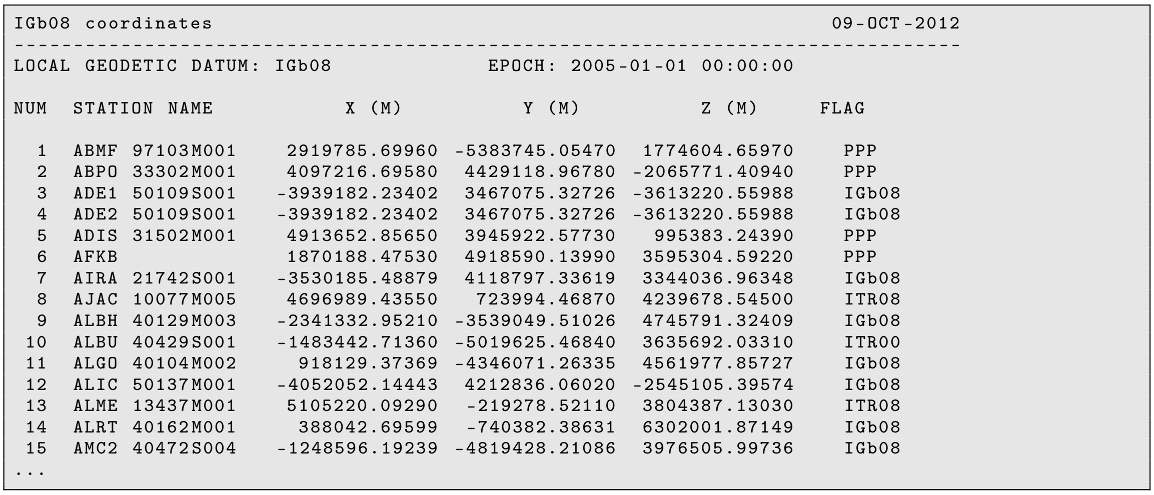
The station related file types used by *Bernese GNSS Software* Version 5.2 are listed in Table 24.4. They are explained in this section.

24.7.1. Station Coordinates

Type: ASCII
Directory: Campaign–specific directory STA.
Extension: CRD
Content: Geocentric station coordinates.
Created by: User–defined ("Menu>Campaign>Edit station files>Station coordinates") or as result file of the processing programs ("Menu>Processing") and numerous service programs ("Menu>Service", in particular in "Menu>Service>Coordinate tools").
Used by: Processing programs as a priori coordinates and many service programs as input files.
Example: Figure 24.40 and ${X}/DOC/EXAMPLE.CRD. Also, ITRF coordinate files (such as ITRF2008_R.CRD and IGB08_R.CRD) for most of the permanent global IGS sites are available from the anonymous CODE ftp area (http://www.aiub.unibe.ch/download/BSWUSER52/STA/).

Table 24.4.: Station related files.

File Description	Extension	Description in Section
Files with station coordinates and velocities		
Station coordinates	`CRD`	24.7.1
Station eccentricities	`ECC`	24.7.2
Station velocities	`VEL`	24.7.3
Kinematic coordinates	`KIN`	24.7.4
Kinematic velocities	`VEL`	24.7.5
Files with equipment configuration		
Station information file	`STA`	24.7.6
Station problem file	`CRX`	24.7.7
Receiver antenna orientation file	`AZI`	24.7.8
Files with station related models and corrections		
Ocean tidal loading table	`BLQ`	24.7.9
Atmospheric tidal loading table	`ATL`	24.7.10
SLR data handling (range bias, CoM)	`RGB`	24.7.11
Files for station selection		
Station selection file	`FIX`	24.7.12
Station sigma file	`SIG`	24.7.13
Station observation sigma factors	`SOS`	24.7.14
Miscellaneous station related files		
Station abbreviation table	`ABB`	24.7.15
Tectonic plate assignment	`PLD`	24.7.16
Baseline definition file	`BSL`	24.7.17
Cluster definition file (input)	`CLU`	24.7.18
Cluster definition file (output)	`CLB`	24.7.19
Files for time series analysis		
Time series list of events	`EVL`	24.7.20
Earthquake list	`ERQ`	24.7.21
Session table		
Session table	`SES`	24.7.22



```
IGb08 coordinates                                                    09-OCT-2012
--------------------------------------------------------------------------------
LOCAL GEODETIC DATUM: IGb08             EPOCH: 2005-01-01 00:00:00

NUM  STATION NAME           X (M)          Y (M)          Z (M)     FLAG

  1  ABMF 97103M001    2919785.69960 -5383745.05470  1774604.65970    PPP
  2  ABPO 33302M001    4097216.69580  4429118.96780 -2065771.40940    PPP
  3  ADE1 50109S001   -3939182.23402  3467075.32726 -3613220.55988    IGb08
  4  ADE2 50109S001   -3939182.23402  3467075.32726 -3613220.55988    IGb08
  5  ADIS 31502M001    4913652.85650  3945922.57730   995383.24390    PPP
  6  AFKB              1870188.47530  4918590.13990  3595304.59220    PPP
  7  AIRA 21742S001   -3530185.48879  4118797.33619  3344036.96348    IGb08
  8  AJAC 10077M005    4696989.43550   723994.46870  4239678.54500    ITR08
  9  ALBH 40129M003   -2341332.95210 -3539049.51026  4745791.32409    IGb08
 10  ALBU 40429S001   -1483442.71360 -5019625.46840  3635692.03310    ITR00
 11  ALGO 40104M002     918129.37369 -4346071.26335  4561977.85727    IGb08
 12  ALIC 50137M001   -4052052.14443  4212836.06020 -2545105.39574    IGb08
 13  ALME 13437M001    5105220.09290  -219278.52110  3804387.13030    ITR08
 14  ALRT 40162M001     388042.69599  -740382.38631  6302001.87149    IGb08
 15  AMC2 40472S004   -1248596.19239 -4819428.21086  3976505.99736    IGb08
...
```

Figure 24.40.: Coordinate (`CRD`) file.

Remarks:

- Each coordinate file has to contain in line 3 the name of a geodetic datum specified in the datum file (see Section 24.3.2). Geodetic datum does not mean that the coordinates below are referring to this datum. The geodetic datum is only used to compute ellipsoidal coordinates in the processing programs if necessary. Whenever possible the geocentric coordinates should refer to the ITRF (i.e., to the same reference frame as the precise orbits used).
- The coordinate epoch is written as the middle of the interval of the processed data by all programs except RXOBV3. It is used in the programs COOVEL ("Menu>Service >Coordinate tools>Extrapolate coordinates", description in Section 10.7.9), HELMR1 ("Menu >Service>Coordinate tools>Helmert transformation", description in Section 10.7.2), COMPAR ("Menu>Service>Coordinate tools>Coordinate comparison", description in Section 10.7.5), and ADDNEQ2 ("Menu>Processing>Combine normal equation systems") to apply the station velocities in order to correct the epoch for the coordinates.
- Data records start in line 7. The end of the input file is indicated by a blank line. Lines below the blank line are ignored.
- With Version 5.2 , station numbers do *not* have to be unique for stations processed simultaneously, i.e., double use of numbers is allowed.
- It is then possible to have coordinate files with more that 1 000 stations.
- The reading routine stops if it finds more than one entry for a station.
- Programs which compute station coordinates always update an input coordinate file leaving coordinates unchanged for stations which were not processed.
- Different programs mark the estimated coordinates with different flags. The flags of coordinates which were unchanged by a program are set to blank. The flags are used by program CRDMERGE in order to establish a priority order when merging coordinate files (see Section 10.7.7) and by programs HELMR1 and COMPAR which compares only coordinates with flags. GPSEST and ADDNEQ2 allow to select stations (e.g., for datum definition) based on their flag. The defined flags are:

R	Coordinates extracted from RINEX files, from program RXOBV3.
C	Coordinates from a code single point positioning, from program CODSPP.
U	Coordinates from an epoch–difference solution in the zero–difference phase screening, from program MAUPRP.
T	Coordinates from a triple–difference solution in the baseline screening, from program MAUPRP.
P	Coordinates estimated with or without constraints in the main parameter estimation step, from program GPSEST.
M	Mean coordinates from several solutions, from program COMPAR.
A	Coordinates estimated without constraints in the normal equation combination step, from program ADDNEQ2.
W	Coordinates estimated with constraints in the normal equation combination step, from program ADDNEQ2.
N	Estimated coordinates for a station used to define a free network condition in the normal equation combination step, from program ADDNEQ2.
I	ITRF coordinates.
E	Eccentric site coordinates, when using a station eccentricity file.
F	Coordinates of the station were fixed in the solution, from programs MAUPRP and GPSEST.
others	Not estimated, not used, or unknown.

- The format of reading the rectangular coordinates in the geocentric system is: FORMAT(I3,2X,A16,3F15.5,4X,A1). It means that only the first character of the flag is read in even if they can be extended to five characters by the program CRDMERGE.
- Geodetic coordinates (latitude, longitude, height) of the stations may be found in various program outputs, e.g., from GPSEST or ADDNEQ2 (see for instance Section 10.3.1).

24.7.2. Station Eccentricities

Type:	ASCII
Directory:	Campaign–specific directory STA.
Extension:	ECC
Content:	Station eccentricities.
Created by:	User–defined, assistance using "Menu>Campaign>Edit station files>Station eccentricities".
Used by:	The processing programs CODSPP ("Menu>Processing>Code-based clock synchronization"), SNGDIF ("Menu>Processing>Create baseline files"), MAUPRP ("Menu >Processing>Phase preprocessing"), GPSEST ("Menu>Processing>Parameter estimation"), and IONEST ("Menu>Service>Ionosphere tools>Local ionosphere model estimation").
Example:	Figure 24.41 and ${X}/DOC/EXAMPLE.ECC.

With an eccentricity file it is possible to have receivers at eccentric points with a known position relative to the center station. Introducing the eccentricity elements as given, only the coordinates of the center station will be estimated in GPSEST, CODSPP, and MAUPRP. In some cases it is easier to estimate the eccentric coordinates (where the GNSS receiver/antenna is located) and to handle the eccentricity problem outside of the Bernese programs (especially if the eccentric values are not precisely known). In that case no eccentricity file is needed.

The eccentricity file may also be used to estimate one set of coordinates for two receivers by introducing the known vector between the two sites into an eccentricity file (local ties).

The file contains:

- A number for the eccentric station[1].
- The name of the eccentric station.
- Name of center station belonging to the eccentric station.
- Eccentricities (DN, DE, DH) in the local geodetic datum specified in the third line of the file if you set SYSTEM to L (L:LOCAL) or the (DX, DY, DZ) eccentricities in the geocentric system (G:GEOCENTRIC).

[1] In contrast to Version 5.0 this must not be unique anymore over all stations processed together

```
Example for an eccentricity file
--------------------------------------------------------------------------------
LOCAL GEODETIC DATUM: IGS00              SYSTEM : L (G: GEOCENTRIC, L: LOCAL)
                                                 CENTER --> STATION
NUM  STATION NAME      CENTER NAME           DN (M)     DE (M)     DH (M)

  1  AUCK 50209S001    AUCK 50209M001       -0.0030    -0.0050     1.3490
  2  FAIR 40408S001    FAIR 40408M001        0.0070     0.0040     0.5320
```

Figure 24.41.: Station eccentricity (ECC) file.

- The datum identifier in the third line must be equal to the datum in the coordinate file used. The eccentricities are added to the coordinates of the center station to obtain the eccentric station coordinates.

The end of the list is indicated by a blank line.

The following strategy is used to obtain the coordinates of a given list of station names:

1. If there is no eccentricity file specified as input in the menu panel, all the stations in the list are assumed to be center stations and the coordinates are directly taken from the coordinate file used.
2. If an eccentricity file is specified, this file is first screened for station names in the list:
   - If a station name is found in the eccentricity file, the station is assumed to be an eccenter and the eccentricities (DN, DE, DH) or (DX, DY, DZ) are added to the coordinates of the corresponding center stations which in turn, must be included in the coordinate file.
   - If a station name is not found in the eccentricity file, the station is assumed to be a center station and the coordinates will be taken from the coordinate file.

Eccenters may also be handled using the antenna eccentricity information in the station information file. Stacking of eccenters of the same center is then possible in ADDNEQ2. In order to include variance–covariance information eccenters should be introduced as local ties on the normal equation level, e.g., imported from SINEX.

24.7.3. Station Velocities

Type: ASCII

Directory: Campaign–specific directory `STA`.

Extension: `VEL`

Content: Station velocity information.

Created by: User–defined, created by ADDNEQ2 ("Menu>Processing>Combine normal equation systems"), FODITS ("Menu>Service>Coordinate tools>Analysis of time series"), NUVELO ("Menu>Service>Coordinate tools>Compute NUVEL velocities"), SNX2NQ0 ("Menu >Conversion>SINEX to normal equations"), CRDMERGE ("Menu>Service>Coordinate tools >Merge coordinate/velocity files"), or VELDIF ("Menu>Service>Coordinate tools>Velocity comparison". You may use "Menu>Campaign>Edit station files>Station velocities" as assistance for editing the file.

Used by: Program ADDNEQ2: as a priori velocity information or as output file for the velocity estimates. Program FODITS: to be updated according to the dicontinuity, velocity changes decisions of the program. Programs COOVEL ("Menu>Service>Coordinate tools>Extrapolate coordinates"), COMPAR ("Menu>Service>Coordinate tools>Coordinate comparison"), and HELMR1 ("Menu>Service >Coordinate tools>Helmert transformation") to adjust the epochs for station coordinates. Program CRDMERGE for maintenance and VELDIF for comparisons.

Example: Figure 24.42 and `${X}/DOC/EXAMPLE.VEL`. Also, ITRF velocity files (such as `ITRF2008_R.VEL` and `IGB08_R.VEL`) for most of the permanent global IGS sites are available from the anonymous CODE ftp area (`http://www.aiub.unibe.ch/download/BSWUSER52/STA/`).

```
IGb08 velocities                                                      09-OCT-2012
--------------------------------------------------------------------------------
LOCAL GEODETIC DATUM: IGb08

NUM  STATION NAME           VX (M/Y)       VY (M/Y)       VZ (M/Y)  FLAG   PLATE

  1  ABMF 97103M001          0.00250        0.00490        0.01090    NNR  CARB
  2  ABPO 33302M001         -0.01100        0.01790        0.01660    NNR  AFRC
  3  ADE1 50109S001         -0.04085        0.00298        0.04851    Ib08 AUST
  4  ADE2 50109S001         -0.04085        0.00298        0.04851    Ib08 AUST
  5  ADIS 31502M001         -0.01850        0.01840        0.01870    NNR  AFRC
  6  AFKB                   -0.02410        0.00940       -0.00040    NNR  EURA
  7  AIRA 21742S001         -0.02581       -0.00734       -0.01507    Ib08
  8  AJAC 10077M005         -0.01360        0.01920        0.01210    IT08 EURA
  9  ALBH 40129M003         -0.01016       -0.00077       -0.00477    Ib08 NOAM
 10  ALBU 40429S001          0.01450        0.02100        0.00180    IT00 EURA
 11  ALGO 40104M002         -0.01578       -0.00407        0.00425    Ib08 NOAM
 12  ALIC 50137M001         -0.03955       -0.00505        0.05410    Ib08 AUST
 13  ALME 13437M001         -0.00790        0.01910        0.01340    IT08 EURA
 14  ALRT 40162M001         -0.02089       -0.00541        0.00612    Ib08 NOAM
 15  AMC2 40472S004         -0.01463        0.00089       -0.00492    Ib08 NOAM
...
```

Figure 24.42.: Station velocity (VEL) file.

Remarks:

- Station names need to be identical to the station names of the coordinate files (or center name of the eccentricity file).
- The information concerning the local geodetic datum has to be identical to the one in the coordinate file used.
- Velocity information (VX, VY, VZ in meter per year) has to be given in the geocentric coordinate system.
- Velocity flags are almost identical to the coordinate flags:

V   Velocities from NNR–NUVEL1 or NNR–NUVEL1A model, from program NUVELO.

A   Velocities estimated without constraints in the normal equation combination step, from program ADDNEQ2.

W   Velocities estimated with constraints in the normal equation combination step, from program ADDNEQ2.

N   Estimated velocities for a station used to define a free network condition in the normal equation combination step, from program ADDNEQ2.

I   ITRF velocities.

others   Not estimated, not used, or unknown.

- The tectonic plate information may be used to derive a new NNR–NUVEL1 or NNR–NUVEL1A velocity file for NUVELO after copying the file to a name with extension PLD (see Section 24.7.16).
- The format to read the data record (apart from the tectonic plate information) is the same as for the station coordinate file.
- If a station has two entries in a velocity file the program stops with an error message.
- If a station is not listed in the velocity but used in the processing, a zero–velocity field is assumed and a related message is issued.

24.7.4. Kinematic Coordinates

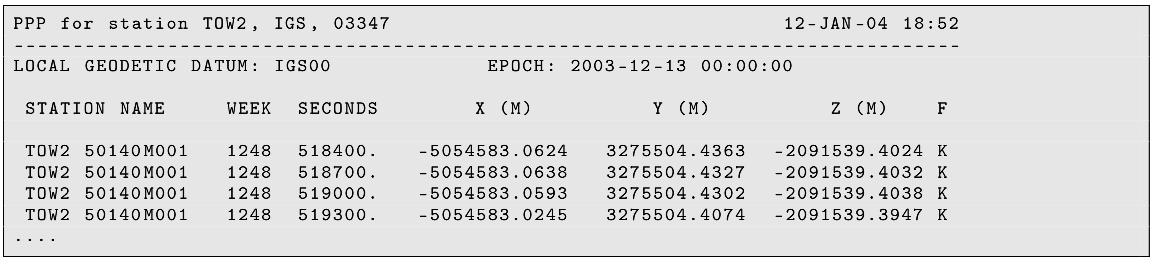
Type:	ASCII
Directory:	Campaign–specific directory STA.
Extension:	KIN
Content:	Kinematic coordinates of a moving receiver (e.g., rover or LEO).
Created by:	Result files of the processing programs CODSPP ("Menu>Processing>Code-based clock synchronization"), or GPSEST ("Menu>Processing>Parameter estimation").
Used by:	CODSPP, MAUPRP ("Menu>Processing>Phase preprocessing"), GPSEST, KINPRE ("Menu>Service>LEO utilities>Convert kinematic pos. to precise orbits"), and GPSSIM ("Menu>Service>Generate simulated observation data").
Example:	Figure 24.43 and ${X}/DOC/EXAMPLE.KIN.

The epoch in the file is given by the GPS week and seconds within the week.

Flags in the kinematic coordinate file:

K Epoch coordinate estimate is reliable.
S Epoch coordinate estimated with "few" observations (specified by "kin. Pos.:min. number of obs. per epoch" in panel "GPSEST 6.9: Kinematic Coordinates") or if a singularity is detected to compute one of the components.
X Estimation of epoch coordinates was singular. Interpolated coordinates are given.

If a kinematic coordinate file is introduced without estimating new kinematic coordinates, only epochs flagged with K are used. If the input file serves to provide only the a priori information for a subsequent kinematic coordinate estimation, all epochs (independently of the flag) are considered.



```
PPP for station TOW2, IGS, 03347                                 12-JAN-04 18:52
--------------------------------------------------------------------------------
LOCAL GEODETIC DATUM: IGS00             EPOCH: 2003-12-13 00:00:00

 STATION NAME     WEEK  SECONDS        X (M)          Y (M)          Z (M)    F

 TOW2 50140M001   1248  518400.   -5054583.0624   3275504.4363  -2091539.4024 K
 TOW2 50140M001   1248  518700.   -5054583.0638   3275504.4327  -2091539.4032 K
 TOW2 50140M001   1248  519000.   -5054583.0593   3275504.4302  -2091539.4038 K
 TOW2 50140M001   1248  519300.   -5054583.0245   3275504.4074  -2091539.3947 K
....
```

Figure 24.43.: Kinematic coordinate (KIN) file.

24.7.5. Kinematic Velocities

Type:	ASCII
Directory:	Campaign–specific directory STA.
Extension:	VEL
Content:	Velocities in kinematic file format. Only used for LEOs if no standard orbit is available.
Created by:	User–defined.
Used by:	GPSEST ("Menu>Processing>Parameter estimation"), GPSSIM ("Menu>Service>Generate simulated observation data"), MAUPRP ("Menu>Processing>Phase preprocessing") if LEO data is processed with kinematic trajectory as input.
Example:	No example.

The file format is identical to the format of the kinematic coordinate file, except that the units are dm/s such that the resolution of the values is 0.01 mm/s. The flags have the same meaning as for the kinematic coordinate file (see Section 24.7.4).

The file is used, e.g., to define the local LEO orbit system (e.g., for the computation of the satellite attitude) if only kinematic coordinates are given and velocities cannot be obtained from a standard orbit. Directory and extension are the same as for station velocity files.

24.7.6. Station Information File

Type: ASCII
Directory: Campaign–specific directory `STA`.
Extension: `STA`
Content: Station related meta information (e.g., station name, equipment, intervals with problematic data).
Created by: User–defined with support of "Menu>Campaign>Edit station files>Station information file" or by utilities available at "Menu>Service>Station information files".
Used by: Numerous programs for various purposes, see Table 24.5 .
Example: Figure 24.44 and `${X}/DOC/EXAMPLE.STA`.

The station information file is the central database for meta information in the *Bernese GNSS Software* and is accessed by several programs. It is divided into the following sections:

1. `TYPE 001:  RENAMING OF STATIONS`
   Using a time window you may rename stations to a different names name.
2. `TYPE 002:  STATION INFORMATION`
   With a time window for each station, the receiver name, the receiver serial number, the antenna name, the antenna serial number, the antenna eccentricity, and a station description to be inserted in the SINEX header may be specified.
3. `TYPE 003:  HANDLING OF STATION PROBLEMS`
   Names and time intervals of stations that shall be excluded on different levels from the processing.
4. `TYPE 004:  STATION COORDINATES AND VELOCITIES (ADDNEQ)`
   This part allows you to constrain the estimates of the station coordinates and/or station velocities of two stations relatively to each other. To get identical velocities for two stations, the a priori velocities must be identical. It is possible to constrain north, east, and up components independently.
5. `TYPE 005:  HANDLING STATION TYPES`
   In this section the `MARKER TYPE` of an antenna may be specified. This is mandatory for spaceborne receivers. Programs check the marker type in order to distinguish ground and spaceborne receivers.

Remarks:

- The file contains the station related meta information in one centralized file. This database is accessed by many programs in the *Bernese GNSS Software* for different purposes. An overview is given in Table 24.5 .
- There are several specific sections describing the usage of the file for specific purposes:
  Section 4.2.3: verify the RINEX header information when importing observations by program RXOBV3,

```
Example for a station information file                           16-OCT-09 15:55
--------------------------------------------------------------------------------

FORMAT VERSION: 1.01
TECHNIQUE:      GNSS

TYPE 001: RENAMING OF STATIONS
------------------------------

STATION NAME          FLG          FROM                   TO         OLD STATION NAME      REMARK
****************      ***  YYYY MM DD HH MM SS  YYYY MM DD HH MM SS  ****************      ************************
ABMF 97103M001        001  2008 07 15 00 00 00                       ABMF*                 IGS.SNX
QUI2 42003S003        003                                            U062 42003S003        NGA RENAMING
WDC2 40451S005        003                       2006 07 16 23 59 59  U072 40451S005        NGA RENAMING
WDC4 40451S008        003  2006 07 17 00 00 00                       U072 40451S005        NGA RENAMING
BAHR 24901M002        001                                            BAHR*                 STN
...

TYPE 002: STATION INFORMATION
-----------------------------

STATION NAME          FLG          FROM                   TO         RECEIVER TYPE         RECEIVER SERIAL NBR   REC #   ANTENNA TYPE         ...
****************      ***  YYYY MM DD HH MM SS  YYYY MM DD HH MM SS  ********************  ********************  ******  ******************** ...
ABMF 97103M001        001  2008 07 15 00 00 00  2009 10 15 20 00 00  ASHTECH UZ-12         UC220                 999999  AERAT2775_43    SPKE ...
ABMF 97103M001        001  2009 10 15 20 00 00                       TRIMBLE NETR5         4917K                 999999  TRM55971.00     NONE ...
BARH 49927S001        001                       2007 03 26 23 59 59  TRIMBLE 4000SSI                             999999  TRM29659.00     UNAV ...
...

TYPE 003: HANDLING OF STATION PROBLEMS
--------------------------------------

STATION NAME          FLG          FROM                   TO         REMARK
****************      ***  YYYY MM DD HH MM SS  YYYY MM DD HH MM SS  ************************************************************
ZIMM 14001M004        001  2003 08 12 00 00 00  2003 08 12 23 59 59  IGSMAIL.4541
...

TYPE 004: STATION COORDINATES AND VELOCITIES (ADDNEQ)
-----------------------------------------------------
                                            RELATIVE CONSTR. POSITION     RELATIVE CONSTR. VELOCITY
STATION NAME 1        STATION NAME 2        NORTH     EAST      UP        NORTH     EAST      UP
****************      ****************      **.*****  **.*****  **.*****  **.*****  **.*****  **.*****
ZIMM 14001M004        ZIMJ 14001M006        99.99999  99.99999  99.99999   0.00010   0.00010   0.00010


TYPE 005: HANDLING STATION TYPES
--------------------------------

STATION NAME          FLG  FROM                 TO                   MARKER TYPE           REMARK
****************      ***  YYYY MM DD HH MM SS  YYYY MM DD HH MM SS  ********************  ************************
JASO L08              001                                            SPACEBORNE
...
```

Figure 24.44.: Station information file (example file `CODE.STA`).

Section 9.4.6: the usage in the context of NEQ manipulations by program ADDNEQ2,

Section 10.3.5: the usage in the context of coordinate time series generation (see also Chapter 11 on the program FODITS), and

Section 18.3: when updating the antenna phase center correction file with the program ATX2PCV.

- Wildcards are only allowed in the column `OLD STATION NAME` when importing observations.
- Flags available in the file are currently not used.
- Empty epoch fields are considered as open time interval boundaries.

Table 24.5.: Programs using the different problem types of the station information file.

		TYPE				
Program name		001	002	003	004	005
Utility programs from "Menu>Service>Station information files"						
RNX2STA	"Menu>Service>Station information files>Extract information from RINEX"	X	X			
SNX2STA	"Menu>Service>Station information files>Extract information from SINEX"		X			
STA2STA	"Menu>Service>Station information files>Convert station information files"	X	X	X	X	
STAMERGE	"Menu>Service>Station information files>Merge station information files"	X	X	X	X	
Programs that check meta data when importing measurements						
RXOBV3	"Menu>RINEX>Import RINEX to Bernese format>Observation files"	X	X	X		X
RNXGRA	"Menu>RINEX>RINEX utilities>Create observation statistics"	X				
RXMBV3	"Menu>RINEX>Import RINEX to Bernese format>Meteo files"	X				
QLRINEXO	"Menu>Service>SLR utilities>Convert normal point files (old format) to RINEX"	X				
CRD2RNXO	"Menu>Service>SLR utilities>Convert normal point files (new format) to RINEX"	X				
Various purposes, see Section 9.4.6 and Chapter 11						
ADDNEQ2	"Menu>Processing>Combine normal equation systems"	X	X	X	X	
FODITS	"Menu>Service>Coordinate tools>Analysis of time series"	X	X	X	X	
Programs that identify LEOs						
CODSPP	"Menu>Processing>Code-based clock synchronization"					X
MAUPRP	"Menu>Processing>Phase preprocessing"					X
GPSEST	"Menu>Processing>Parameter estimation"					X
GPSSIM	"Menu>Service>Generate simulated observation data"					X
Programs that access the equipment information						
ATX2PCV	"Menu>Conversion>ANTEX to Bernese format"		X			
CHGHED	"Menu>Service>Bernese observation files>Change header"		X			
Programs which are deleting records for problematic stations						
SNX2SLR	"Menu>Service>SLR utilities>Prepare SLR data handling file"			X		
TROTRO	"Menu>Conversion>Manipulate troposphere SINEX files"			X		

24.7.7. Station Problem File

Type: ASCII
Directory: Campaign–specific directory STA.
Extension: CRX
Content: Known station inconsistencies, i.e., known wrong RINEX header entries for receiver name/number, antenna name/number, or antenna eccentricity.
Created by: User–defined. STA2STA ("Menu>Service>Station information files>Convert station information files") converts Version 5.0 –formatted tables into the new format.
Used by: RXOBV3 ("Menu>RINEX>Import RINEX to Bernese format>Observation files").
Example: Figure 24.45 and ${X}/DOC/EXAMPLE.CRX.

The structure of the file is similar as the TYPE 002 section of the station information file.

The file may be used to suppress error handling for known station inconsistencies in program RXOBV3 (see Section 4.2.3.4). Enter the expected (wrong) RINEX header entry in the corresponding column. The other columns may be specified as unused by " *** UNDEFINED *** " for string values (receiver and antenna type and the serial numbers), "999999" for the integer values (internal receiver and antenna number), and "999.9999" for antenna eccentricity.

```
POSSIBLE RINEX INCONSISTENCIES                                      11-Apr-2010
-------------------------------------------------------------------------------

STATION NAME                  FROM                   TO         RECEIVER TYPE        ...
****************      YYYY MM DD HH MM SS  YYYY MM DD HH MM SS  ******************** ...
ALIC 50137M001        1990 01 01 00 00 00  2099 12 31 00 00 00   *** UNDEFINED ***   ...
AMUN 66040M001        1990 01 01 00 00 00  2099 12 31 00 00 00   *** UNDEFINED ***   ...
ANKR 20805M002        1990 01 01 00 00 00  2099 12 31 00 00 00   *** UNDEFINED ***   ...
AUCK 50209M001B       2002 12 02 00 00 00  2003 01 08 23 59 59  ASHTECH MICROZ       ...
BAKO 23101M002        1990 01 01 00 00 00  2099 12 31 00 00 00  TRIMBLE 4000SSE      ...
BJFS 21601M001        1990 01 01 00 00 00  2099 12 31 00 00 00  *Z-XII*              ...

Undefined values:     1990 01 01 00 00 00  2099 12 31 00 00 00   *** UNDEFINED ***   ...
```

Figure 24.45.: Station problem file for known station inconsistencies.

24.7.8. Receiver Antenna Orientation File

Type: ASCII
Directory: Campaign–specific directory STA.
Extension: AZI
Content: Receiver antenna orientations.
Created by: User–defined.
Used by: GPSEST ("Menu>Processing>Parameter estimation") and GPSSIM ("Menu>Service >Generate simulated observation data").
Example: Figure 24.46 and ${X}/DOC/EXAMPLE.AZI.

The antenna orientation file specifies the azimuth of the antenna zero–azimuth. Each session should be listed where any of the antennas was not oriented to the north in order. The misorientation of the antenna is considered when applying or estimating phase center patterns. This file is of special interest if you have to process antenna calibration campaigns, where the antennas were oriented differently from session to session. The azimuthal antenna phase center corrections are also correctly considered in that way if the antenna is not oriented to north.

```
RECEIVER ANTENNA ORIENTATIONS                                          15-NOV-03
--------------------------------------------------------------------------------

RECEIVER TYPE          ANTENNA S/N   SESS  AZIMUTH
ANTENNA  TYPE          FROM    TO            DEG
********************  ****** ******  ****    ***

AOA SNR-8000 ACT        2104   2104  0011    270
AOAD/M_T

AOA SNR-8000 ACT        3104   3104  0011    270
AOAD/M_T

AOA SNR-8000 ACT        4104   4104  0011    270
AOAD/M_T

AOA SNR-8000 ACT        5104   5104  0011    270
AOAD/M_T

AOA SNR-8000 ACT         104    104  0011    270
AOAD/M_T
```

Figure 24.46.: Antenna orientation (AZI) file.

The file contains receiver and antenna names, antenna number, session number, and the orientation of the antenna (azimuth in degrees). Note that an entry is only considered if the specified session number corresponds to the session number given in the header of the observation file. If no antenna orientation file is specified, a default orientation of all antennas to the north (azimuth of 0) is assumed.

The receiver name contained in the file is not checked in Version 5.2 . You may specify the antenna name in the field foreseen for the receiver name and set the field foreseen for the antenna name blank. Note that in this case each entry is separated by two blank lines.

24.7.9. Ocean Tidal Loading Table

Type:	ASCII
Directory:	Campaign–specific directory STA.
Extension:	BLQ
Content:	Ocean tidal loading amplitudes and phases.
Created by:	User–defined, following the instructions in ${X}/DOC/README_BLQ.TXT .
Used by:	Orbit programs to consider the center of mass corrections for ocean tidal loading, namely BRDTAB ("Menu>Orbits/EOP>Broadcast orbits>Create tabular orbits"), PRETAB ("Menu>Orbits/EOP>Create tabular orbits"), ORBGEN ("Menu >Orbits/EOP>Create/update standard orbits"), STDPRE ("Menu>Orbits/EOP>Convert standard to precise orbits"), and IRV2STV ("Menu>Service>SLR utilities>Convert IRV/PRE to state vectors") as well as the processing programs GPSEST ("Menu>Processing >Parameter estimation"), GPSSIM ("Menu>Service>Generate simulated observation data"), and CLKEST ("Menu>Service> *Clock* tools>Epoch-wise clock interpolation") to consider the CMC for the backward transformation and the ocean tidal loading deformation for the tracking stations.
Example:	Block for site ABMF in Figure 24.47. An example is also available in ${X}/ DOC/EXAMPLE.BLQ and in the anonymous BSW ftp area (see Section 4.13).

This table may optionally be used in program GPSEST, GPSSIM, and CLKEST in order to take into account the effects ocean tide loading on site coordinates. It contains station–specific amplitudes and phases of the eleven largest tidal constituents for the vertical

```
$$ Ocean loading displacement
$$
$$ Calculated on froste using olfg/olmpp of H.-G. Scherneck
$$
$$ COLUMN ORDER:  M2  S2  N2  K2  K1  O1  P1  Q1  MF  MM SSA
$$
$$ ROW ORDER:
$$ AMPLITUDES (m)
$$   RADIAL
$$   TANGENTL    EW
$$   TANGENTL    NS
$$ PHASES (degrees)
$$   RADIAL
$$   TANGENTL    EW
$$   TANGENTL    NS
$$
$$ Displacement is defined positive in upwards, South and West direction.
$$ The phase lag is relative to Greenwich and lags positive. The
$$ Gutenberg-Bullen Greens function is used. In the ocean tide model the
$$ deficit of tidal water mass has been corrected by subtracting a uniform
$$ layer of water with a certain phase lag globally.
$$
$$ Complete <model name> : No interpolation of ocean model was necessary
$$ <model name>_PP       : Ocean model has been interpolated near the station
$$                         (PP = Post-Processing)
$$
$$ CMC:  NO   (corr.tide centre of mass)
$$
$$ Ocean tide model: FES2004
$$
$$ CMC start : center of mass coefficient file content
$$ CMC format: (a,1p,t42,3(2x,2e12.4))
$$ CMC frequ : M2   NCDF_FES2004                           -1.2661E-03 -1.4298E-03   -1.3724...
$$ CMC frequ : S2   NCDF_FES2004                           -1.7763E-04 -5.7273E-04   -5.3350...
$$ CMC frequ : N2   NCDF_FES2004                           -3.2372E-04 -2.8986E-04   -2.7121...
$$ CMC frequ : K2   NCDF_FES2004                           -1.1814E-04 -1.5250E-04   -1.1223...
$$ CMC frequ : K1   NCDF_FES2004                           -1.1370E-03  4.4839E-03   -1.8539...
$$ CMC frequ : O1   NCDF_FES2004                           -1.6802E-04  2.9702E-03   -1.3985...
$$ CMC frequ : P1   NCDF_FES2004                           -3.6495E-04  1.4941E-03   -6.1436...
$$ CMC frequ : Q1   NCDF_FES2004                            3.0709E-05  4.5472E-04   -2.7831...
$$ CMC frequ : Mf   NCDF_FES2004                           -5.0643E-04 -7.3040E-05   -2.2065...
$$ CMC frequ : Mm   NCDF_FES2004                           -2.7885E-04  2.0596E-05    4.6882...
$$ CMC frequ : Ssa  NCDF_FES2004                           -1.4899E-04  2.6146E-06    1.3687...
$$ CMC end   :
$$
$$ END HEADER
$$
  ABMF
$$ FES2004_PP ID: 2012-08-02 03:03:48
$$ Computed by OLMPP by H G Scherneck, Onsala Space Observatory, 2012
$$ ABMF,                      RADI TANG  lon/lat:  298.4725   16.2623  -25.579
  .01090 .00347 .00252 .00089 .00373 .00392 .00123 .00092 .00114 .00057 .00043
  .00214 .00079 .00044 .00023 .00036 .00037 .00012 .00007 .00007 .00002 .00000
  .00410 .00109 .00090 .00027 .00030 .00020 .00010 .00006 .00003 .00002 .00001
    70.7   96.1   50.0   91.9   32.9   32.8   32.9   10.0  178.8  178.0  179.3
    57.9   79.8   48.1   74.0   81.7   56.5   82.5   25.4  145.5  126.1   53.2
  -156.7 -127.3 -175.5 -126.0  -63.0 -115.5  -62.6 -126.0 -174.8  176.2  179.3
$$
  GANP
$$ Complete FES2004
$$ Computed by OLFG, H.-G. Scherneck, Onsala Space Observatory 2011-Oct-27
$$ GANP,                      RADI TANG  lon/lat:   20.3229   49.0347  746.013
  .00400 .00141 .00087 .00036 .00159 .00094 .00053 .00011 .00047 .00026 .00021
  .00144 .00031 .00034 .00009 .00038 .00029 .00013 .00005 .00005 .00003 .00003
  .00040 .00014 .00006 .00004 .00036 .00013 .00012 .00001 .00002 .00001 .00000
   -72.3  -40.2  -91.7  -50.9  -63.8 -101.8  -64.6 -142.8    8.0    5.2    0.9
    66.8  100.9   41.6   91.0  107.4   44.3  105.8    5.6 -169.3 -165.6 -177.1
   -81.1  -39.7 -109.2  -58.7   39.3    8.7   38.9   49.5 -166.4  174.3 -175.7
...
```

Figure 24.47.: Ocean tidal loading (BLQ) file. Excerpt of one station–specific block.

as well as for the horizontal station components. From these main coefficients 342 constituents are derived using the routine ${LG}/HARSUB.f according to the IERS conventions (Petit and Luzum 2010). For reasons of efficiency, the pre–processing programs CODSPP ("Menu>Processing>Code-based clock synchronization"), MAUPRP ("Menu>Processing>Phase preprocessing"), and IONEST ("Menu>Service>Ionosphere tools>Local ionosphere model estimation") do not apply these corrections.

Use the web–service at http://holt.oso.chalmers.se/loading/ to get a table of the ocean loading coefficients for your stations. Copy the coordinates to the input field of the web–page. For the GNSS analysis you need the vertical and horizontal displacements, no corrections for the center of mass motion have to be applied. After submitting the job you will get the ocean loading file by e–mail. Only approximate site coordinates are required. Compute a new set of coefficients for stations that are separated by more than 10 km.

You have to save this information in a file with the extension BLQ in your campaign's STA–directory or append it to an already existing file (for efficiency compute coefficients only for stations for which you do not already have the information).

The reading routine ${LG}/GTOCNL.f checks only the first four characters of the station name to find the coefficients in the file if the station 4–character abbreviation in the first and the fourth line of a station block in the file are equal (see Figure 24.47). If the name entries are different, the two names are concatenated (assuming that the second entry contains the station domes number) and compared to the full name of the station for which the coefficients are requested. The routine is called once per station in a program run. The coefficients are buffered for each requested station.

The file used for the processing at CODE is available at http://www.aiub.unibe.ch/download/BSWUSER52/STA/FES2004.BLQ.

The header of the ocean tidal loading table in Figure 24.47 provides the center of mass corrections (CMC). They may be applied for orbit programs when transferring satellite positions from the Earth–fixed to the celestial frame in order to ensure that the satellite orbits refer to the center-of-mass. It is obvious that the CMC need also be applied again when transforming the satellite positions back to the Earth–fixed frame in the processing programs. See more details on the CMC correction in Section 5.4.1 .

24.7.10. Atmospheric Tidal Loading File

Type: ASCII

Directory: Campaign–specific directory STA.

Extension: ATL

Content: Atmospheric tidal loading corrections.

Created by: User defined or GRDS1S2 ("Menu>Service>Coordinate tools>Extract atmospheric tidal loading coefficients").

Used by: Orbit programs to consider the center of mass corrections for ocean tidal loading, namely BRDTAB ("Menu>Orbits/EOP>Broadcast orbits>Create tabular orbits"), PRETAB ("Menu>Orbits/EOP>Create tabular orbits"), ORBGEN ("Menu>Orbits/EOP>Create/update standard orbits"), STDPRE ("Menu>Orbits/EOP>Convert standard to precise orbits"), and IRV2STV ("Menu>Service>SLR utilities>Convert IRV/PRE to state vectors") as well as the processing programs GPSEST ("Menu>Processing>Parameter estimation"), GPSSIM ("Menu>Service>Generate simulated observation data"), and CLKEST ("Menu>Service> *Clock* tools>Epoch-wise clock interpolation") to consider the CMC for the backward transformation and the ocean tidal loading deformation for the tracking stations.

Example: Figure 24.48 and ${X}/DOC/EXAMPLE.ATL.

```
 $$ Atmospheric Tidal loading displacememt
 $$
 $$ calculated using grdinterp.f (t. van Dam)
 $$ calculated using atmospheric tidal model Ray and
 $$ Ponte, Annales Geophysicae (2003) v21; 1897-1910
 $$ Farrell elastic Greens functions have been used
 $$
 $$ Displacement is defined positive up, north
 $$ and east directions
 $$
 $$ Column order coss1 sins1 coss2 sins2
 $$
 $$ Row order:
 $$ RADIAL
 $$ TANGENTIAL NS
 $$ TANGENTIAL EW
 $$
 $$
 $$
 $$
 $$ station ABPO; coord.(long,lat) 47.22921 -19.018307
 $$
ABPO 33302M001
       -0.3414     -0.3933      1.1520      0.3189
       -0.0707     -0.0794      0.0310      0.0201
       -0.2743      0.2426      0.0497     -0.1709
 $$
 $$ station ADE1; coord.(long,lat) 138.64734 -34.729
 $$
ADE1 50109S001
       -0.3320      0.4477     -0.8302     -0.1207
       -0.0555      0.0782     -0.0948     -0.0104
        0.2952      0.2096     -0.0194      0.1892
 $$
 $$ station ADE2; coord.(long,lat) 138.64734 -34.729
 $$
...
```

Figure 24.48.: Example of an atmospheric tidal loading ATL file.

The structure and usage of the file on atmospheric tidal loading is analogue to the one of the ocean tidal loading. The effect is quite small (the unit is changed from meters to millimeters) but when applying for non–tidal atmospheric loading, it is important to correct *consistently* for the tidal effect. The principle how to generate the file is described in ${X}/DOC/README_ATL.TXT.

24.7.11. SLR Correction File

Type: ASCII
Directory: Campaign–specific directory STA.
Extension: SLR
Content: Biases and center of mass corrections for SLR processing.
Created by: User defined or created by SNX2SLR ("Menu>Service>SLR utilities>Prepare SLR data handling file").
Used by: GPSEST ("Menu>Processing>Parameter estimation") and ADDNEQ2 ("Menu >Processing>Combine normal equation systems").
Example: Figure 24.49 and ${X}/DOC/EXAMPLE.SLR.
Created by: User–defined or created by SNX2SLR ("Menu>Service>SLR utilities>Prepare SLR data handling file"; only entries from the type RGB, TIM, P but not for COM are supported).
Used by: GPSEST ("Menu>Processing>Parameter estimation"), ADDNEQ2 ("Menu>Processing >Combine normal equation systems"), QLRSUM ("Menu>Service>SLR utilities>Create quick-

```
Extract SLR data handling information from SINEX                 18-NOV-15 12:39
--------------------------------------------------------------------------------------------

Format Version:  1.00

TYP  Station name      Sat  WL  Value (m)   ... Valid from           to                  ...
***  **************** ****  **  ****.****** ... ******************   ****************** ...
RGB  7810 14001S001         R1      0.050000 ... 1988-05-24 00:00:00  1989-09-30 00:00:00 ...
RGB  7810 14001S007         R1     -0.026000 ... 1998-01-01 00:00:00  2002-05-29 00:00:00 ...
RGB  7810 14001S007         R1     -0.015000 ... 2002-05-29 00:00:00  2003-03-11 00:00:00 ...
RGB  7810 14001S007         R1     -0.022000 ... 2003-03-11 00:00:00  2004-12-28 00:00:00 ...
RGB  7810 14001S007         R1     -0.026000 ... 2004-12-28 00:00:00  2006-02-06 00:00:00 ...
TIM  7810 14001S001    951  R1     64.500000 ... 1997-10-04 23:18:57  1997-10-04 23:24:47 ...
RGB  1884 12302S002         R1     -0.075000 ... 1994-01-01 00:00:00  1997-09-02 00:00:00 ...
P    7210 40445M001                 0.000000 ... 1996-06-18 00:00:00  1996-11-25 00:00:00 ...
TIM  7090 50107M001         R1      0.387500 ... 1998-12-21 00:00:00  1998-12-22 00:00:00 ...
TIM  7105 40451M105         R1      0.201500 ... 1999-03-04 20:15:00  1999-03-08 11:35:20 ...
RGB  7124 92201M007         R1      0.163910 ... 2004-05-01 00:00:00  2004-07-01 17:34:00 ...
P    1864 12340S002                -6.800000 ... 1990-09-27 00:00:00  2000-12-24 00:00:00 ...
RGB  1868 12341S001         R1      1.500000 ... 1999-11-15 10:23:00  1999-11-15 19:11:00 ...
RGB  1884 12302S002    951  R1     -5.650000 ... 1999-08-02 19:38:00  1999-08-02 20:38:00 ...
RGB  1884 12302S002    951  R1     -4.150000 ... 1999-08-03 21:53:00  1999-08-05 01:01:00 ...
RGB  1884 12302S002    951  R1     -5.650000 ... 1999-08-05 19:09:00  1999-08-05 23:39:00 ...
RGB  1884 12302S002    951  R1     -4.150000 ... 1999-08-07 19:52:00  1999-08-07 20:52:00 ...
RGB  1884 12302S002    951  R1     -5.650000 ... 1999-08-07 23:27:00  1999-08-08 00:27:00 ...
RGB  7105 40451M105    951  R1     -9.221000 ... 1999-12-05 06:31:45  1999-12-05 07:31:45 ...
COM  7810 14001S001    951          0.251000 ... 1980-01-01 00:00:00  1995-04-30 00:00:00 ...
COM  7810 14001S001    952          0.251000 ... 1980-01-01 00:00:00  1995-04-30 00:00:00 ...
COM  7810 14001S007    951          0.245000 ... 1996-01-02 00:00:00  2001-03-09 00:00:00 ...
COM  7810 14001S007    952          0.245000 ... 1996-01-02 00:00:00  2001-03-09 00:00:00 ...
COM  7810 14001S007    951          0.248000 ... 2001-03-09 00:00:00  2008-02-18 00:00:00 ...
COM  7810 14001S007    952          0.248000 ... 2001-03-09 00:00:00  2008-02-18 00:00:00 ...
COM  7810 14001S007    951          0.249000 ... 2008-03-04 00:00:00  2050-12-31 00:00:00 ...
COM  7810 14001S007    952          0.249000 ... 2008-03-04 00:00:00  2050-12-31 00:00:00 ...
COM  7811 12205S001    951          0.253000 ... 1988-05-13 00:00:00  2050-12-31 00:00:00 ...
COM  7811 12205S001    952          0.253000 ... 1988-05-13 00:00:00  2050-12-31 00:00:00 ...
COM  7820 21609S002    951          0.251000 ... 1998-05-20 00:00:00  2050-12-31 00:00:00 ...
COM  7820 21609S002    952          0.251000 ... 1998-05-20 00:00:00  2050-12-31 00:00:00 ...
COM                    951          0.251000 ...                                          ...
COM                    952          0.251000 ...                                          ...
...  .... .........    ...  ..      ........ ... ..................   .................. ...
```

Figure 24.49.: Example of an SLR correction file.

look residual statistics"), and SNX2SLR ("Menu>Service>SLR utilities>Prepare SLR data handling file").

Example: Figure 24.49 and `${X}/DOC/EXAMPLE.SLR`.

The SINEX-file, which is the input file for SNX2SLR, can be downloaded from `http://ilrs.dgfi.tum.de/fileadmin/data_handling/ILRS_Data_Handling_File.snx`. The resulting `SLR` file contains station– and satellite–specific SLR observation corrections such as:

`RGB`: range biases,
`TIM`: time biases, and
`P`: pressure corrections.

Note that the SINEX file does not contain information about station-specific center-of-mass corrections (`COM`). In order to process SLR data to LAGEOS, the `COM` values in the `SLR` file have to be manually introduced. To use station-specific `COM` values, the information given in the LAGEOS data handling file (`http://ilrs.dgfi.tum.de/fileadmin/data_handling/com_lageos.txt`) has to be converted to the format as shown in Figure 24.49. Note that time and pressure biases are not yet supported by the current *Bernese GNSS Software*.

24.7.12. Station Selection File

Type: ASCII
Directory: Campaign–specific directory STA.
Extension: FIX
Content: Selection list of station names for automated processing, e.g., for datum definition, selection of potential reference clock, etc.
Created by: User–defined. Assistance using "Menu>Campaign>Edit station files>Station selection list". Several programs may also construct or update station selection files CRDMERGE ("Menu>Service>Coordinate tools>Merge coordinate/velocity files"), FODITS ("Menu>Service>Coordinate tools>Analysis of time series"), HELMR1 ("Menu >Service>Coordinate tools>Helmert transformation"), and SNX2NQ0 ("Menu>Conversion >SINEX to normal equations").
Used by: Programs for different purposes ADDNEQ2 ("Menu>Processing>Combine normal equation systems"), CCRNXC ("Menu>Service> *Clock* tools>Combine/manipulate clock RINEX files"), FODITS, GPSEST ("Menu>Processing>Parameter estimation"), HELMR1, MKCLUS ("Menu>Service>Automated processing>Form clusters"), SNX2STA ("Menu>Service>Station information files>Extract information from SINEX"), and STAMERGE ("Menu>Service>Station information files>Merge station information files").
Example: Figure 24.50 and ${X}/DOC/EXAMPLE.FIX.

The file contains a listing of station names. The program stops reading the file after a blank line.

```
Example for a station selection file                        15-APR-10 07:33
-------------------------------------------------------------------------------

Station name
****************
AUCK 50209M001
FAIR 40408M001
OHIG 66008M001
THU1 43001M001
WTZR 14201M010
```

Figure 24.50.: Station selection (FIX) file.

24.7.13. Station Sigma File

Type: ASCII
Directory: Campaign–specific directory STA.
Extension: SIG
Content: Absolute and relative constraints for the estimation of coordinates and of site–specific troposphere parameters. It is also used report and introduce the RMS of linear fits of clock values.
Created by: User–defined. Assistance using "Menu>Campaign>Edit station files>Station selection list". CCRNXC ("Menu>Service> *Clock* tools>Combine/manipulate clock RINEX files") and FODITS ("Menu>Service>Coordinate tools>Analysis of time series").
Used by: Programs for different purposes ADDNEQ2 ("Menu>Processing>Combine normal equation systems"), CCRNXC ("Menu>Service> *Clock* tools>Combine/manipulate clock RINEX files"), CLKEST ("Menu>Service> *Clock* tools>Epoch-wise clock interpolation"), FODITS, GPSEST ("Menu>Processing>Parameter estimation"), and MKCLUS ("Menu >Service>Automated processing>Form clusters").

```
Station sigma file for contraining of site coordinates           25-OCT-09 15:46
--------------------------------------------------------------------------------

Station name            sigma1    sigma2    sigma3
****************       **.****   **.****   **.****
AUCK 50209M001          0.0001    0.0001    0.0001
FAIR 40408M001          0.0001    0.0001    0.0001
OHIG 66008M001          0.0001    0.0001    0.0001
THU1 43001M001          0.0001    0.0001    0.0001
WTZR 14201M010          0.0001    0.0001    0.0001
```

Figure 24.51.: Station sigma (SIG) file for the constraining of site coordinates.

Example: Figure 24.51 and ${X}/DOC/EXAMPLE.SIG.

The station sigma file has a generic format consisting of a header (lines 1 to 5), a column containing the station name, and a number of columns containing sigma values. A sigma value of zero is interpreted as no sigma, i.e., no constraint. Programs accept only files containing the correct number of sigma values. The number of sigma values per station depends on the application:

One column: Contains clock RMS values in microseconds used in program MKCLUS for the optimum creation of station clusters for clock estimation. Such a file may be written by CCRNXC.

Two columns: Contains absolute (first column) and relative (second column) sigmas in meters for the constraining of zenith troposphere parameters (no gradient estimation). Used by program GPSEST.

Three columns: Contains sigma values for the constraining of station coordinates (in meters) or station velocities (in meters/year) in the north, east, and up components (first, second, third column). Used by programs GPSEST and ADDNEQ2.

Four columns: Contains sigmas in meters for the absolute (first column) and relative (second column) constraining of zenith troposphere parameters and for the absolute (third column) and relative (fourth column) constraining of tropospheric gradient parameters. Used by program GPSEST.

Six columns: Contains sigmas in meters for the absolute (first column) and relative (second column) constraining of zenith troposphere parameters, and for the absolute (third column) and relative (fourth column) constraining of tropospheric gradient parameters in north direction, and for absolute (fifth column) and relative (sixth column) constraining of gradient parameters in east direction. Used by program GPSEST.

24.7.14. Station Observation Sigma Factor File

Type: ASCII

Directory: Campaign–specific directory STA.

Extension: SOS

Content: Station observation sigma factors for a list of stations and for specific measurement types.

Created by: Program RESRMS ("Menu>Service>Residual files>Create residual statistics") or using "Menu>Campaign>Edit station files>Observation sigma factors".

Used by: CODSPP ("Menu>Processing>Code-based clock synchronization") and GPSEST ("Menu >Processing>Parameter estimation").

```
RRESIDUAL CHECK 14195                                             15-JUL-14 17:52
----------------------------------------------------------------------------------
                                           VALID:  from                  to
STATION NAME          MEATYP  SIG-FACT  yyyy mm dd hh mm ss  yyyy mm dd hh mm ss
****************      *****   ****.***  **** ** ** ** ** **  **** ** ** ** ** **
ALIC 50137M001        CODE       1.000
ASPA 50503S006        CODE       3.500
AUCK 50209M001        CODE       3.500
BNDY 50185M001        CODE       8.500
...
```

Figure 24.52.: Station observation sigma factor (SOS) file.

Example: Figure 24.52 and ${X}/DOC/EXAMPLE.SOS.

This file provides station observation sigma factors for a list of stations and for specific measurement types. The factors may be used by program CODSPP for the scaling of the outlier rejection threshold, see Section 6.3.2. Program GPSEST may make use of the observation sigma factors for applying a station–specific weighting of the observations or for a rescaling of the edit level for code observations in the zero–difference mode, see Section 7.4.2. In particular, weighting of observations may be useful for pseudorange observations. Station observation sigma factors may be determined with program RESRMS based on residual statistics, see Section 6.6.

The end of the station observation sigma factor file is indicated by a blank line. Lines below the blank line are ignored.

24.7.15. Station Abbreviation Table

Type: ASCII
Directory: Campaign–specific directory STA.
Extension: ABB
Content: 4– and 2–character station abbreviations.
Created by: User–defined, edit the file using "Menu>Campaign>Edit station files>Abbreviation table". Program RXOBV3 ("Menu>RINEX>Import RINEX to Bernese format>Observation files" can update the table.
Used by: RXOBV3, SNGDIF ("Menu>Processing>Create baseline files"), QLRINEXO ("Menu >Service>SLR utilities>Convert normal point files (old format) to RINEX"), CRD2RNXO ("Menu>Service>SLR utilities>Convert normal point files (new format) to RINEX"), GPSSIM ("Menu>Service>Generate simulated observation data") to generate observation filenames as well as BASLST ("Menu>Service>Automated processing>Select baselines"), MPRXTR ("Menu>Processing>Program output extraction>Phase preprocessing"), RESCHK ("Menu>Service>Automated processing>Detect misbehaving stations/satellites"), RESRMS ("Menu>Service>Residual files>Create residual statistics"), and STA2ID to extract the full station name from an abbreviation.
Example: Figure 24.53 and ${X}/DOC/EXAMPLE.ABB.

Station abbreviations are used for the automatic generation of filenames by programs RXOBV3, SNGDIF, GPSSIM, QLRINEXO, and CRD2RNXO. Program RXOBV3 may automatically add unique abbreviations to the table for new stations (see Section 4.2.3.6). Abbreviations have to be unique. Otherwise observation files may be overwritten by the observation files of other stations. When editing the table with the menu, a warning message is issued if two abbreviations are the same.

```
Example for station abbreviation table                           05-AUG-11 12:40
--------------------------------------------------------------------------------

Station name            4-ID    2-ID    Remark
****************        ****     **     ***************************************
AUCK 50209M001          AUCK     AU
FAIR 40408M001          FAIR     FA
OHIG 66008M001          OHIG     OH
THU1 43001M001          THU1     TH
WTZR 14201M010          WTZR     WT
CHAM L06                CHAM     L6     LEO
JASO L08                JASO     L8     LEO
GRCA L09                GRCA     L9     LEO
GRCB L10                GRCB     L0     LEO
7501 30302M003          7501            SLR
7810 14001S001          7810            SLR
```

Figure 24.53.: Station abbreviation table (ABB).

24.7.16. Tectonic Plate Assignment

Type:	ASCII
Directory:	Campaign–specific directory STA.
Extension:	PLD
Content:	Tectonic plate assignment of stations.
Created by:	User–defined. You may use "Menu>Campaign>Edit station files>Tectonic plate assignment" as assistance for editing the file.
Used by:	NUVELO ("Menu>Service>Coordinate tools>Compute NUVEL velocities").
Example:	Figure 24.54 and ${X}/DOC/EXAMPLE.PLD.

The format of the plate definition file is the same as for to the station velocity file. It may even contain velocities. Addressing routines use only station name and plate code. See Table 24.6 for the tectonic plates available in program NUVELO of the *Bernese GNSS Software* Version 5.2 .

Table 24.6.: List of supported tectonic plates.

Plate ID	Description
PCFC	Pacific
AFRC	African
ANTA	Antarctic
ARAB	Arabian
AUST	Australian
CARB	Carribean
COCO	Cocos (north of NAZC, south of NOAM, east of CARB)
EURA	Eurasian
INDI	Indian
NAZC	Nazca (west of SOAM, east of PCFC)
NOAM	North American
SOAM	South American
JUFU	Juan de Fuca Plate (inbetween NOAM and PCFC, northern NOAM)
PHIL	Philippine

```
Example for a plate definition file                            09-OCT-12 13:07
--------------------------------------------------------------------------------
LOCAL GEODETIC DATUM: IGb08

NUM  STATION NAME           VX (M/Y)       VY (M/Y)       VZ (M/Y)  FLAG   PLATE

  1  ABMF 97103M001                                                        CARB
  2  ABPO 33302M001                                                        AFRC
  3  ADE1 50109S001                                                        AUST
  4  ADE2 50109S001                                                        AUST
  5  ADIS 31502M001                                                        AFRC
  8  AJAC 10077M005                                                        EURA
  9  ALBH 40129M003                                                        NOAM
 10  ALBU 40429S001                                                        EURA
 11  ALGO 40104M002                                                        NOAM
 12  ALIC 50137M001                                                        AUST
 ...
```

Figure 24.54.: Tectonic plate assignment (PLD) file.

24.7.17. Baseline Definition File

Type: ASCII

Directory: Campaign–specific directory STA.

Extension: BSL

Content: Pre–defined baselines.

Created by: User–defined. Assistance using "Menu>Campaign>Edit station files>Baseline definition file", or written by SNGDIF ("Menu>Processing>Create baseline files") or MPRXTR ("Menu>Processing>Program output extraction>Phase preprocessing").

Used by: SNGDIF to select baselines to be created as well as by the programs COMPAR ("Menu>Service>Coordinate tools>Coordinate comparison") and GPSXTR ("Menu>Processing>Program output extraction>Parameter estimation/stacking") to select for which baselines specific program output sections shall be generated.

Example: Figure 24.55 and ${X}/DOC/EXAMPLE.BSL.

The baseline definition file contain pairs of station names and may be used in the following cases:

1. Pre–define baselines in the program SNGDIF.
   This option is useful to force program SNGDIF to form specific baselines (e.g., baselines with identical antenna types) independently of the strategy used. Possible strategies are explained in Section 6.4. It is in addition possible to store the baseline definitions created by SNGDIF. That helps in the case you have, e.g., to form the same baselines for the pseudorange observations that you created for the phase observations (Melbourne–Wübbena combination in GPSEST). Another application is a combined half automatic, half manual baseline selection (e.g., to store the baselines

```
ADIS 31502M001   ANKR 20805M002
ADIS 31502M001   MAL2 33201M003
ADIS 31502M001   URUM 21612M001
ALIC 50137M001   ASPA 50503S006
ALIC 50137M001   CEDU 50138M001
ANKR 20805M002   GLSV 12356M001
AREQ 42202M005   CRAR 66001M004
AREQ 42202M005   FRDN 40146M001
AREQ 42202M005   GLPS 42005M002
AREQ 42202M005   KOUR 97301M210
...
```

Figure 24.55.: Baseline definition (BSL) file.

of SNGDIF using the criterion of a maximum number of observations in a first step, to change the baselines in the file according to your wishes, and to run SNGDIF in a second iteration specifying your baseline definitions). Finally, program MPRXTR may write a baseline definition file to complete an existing baseline network after the deletion of a bad baseline.

2. Select baselines for writing repeatability values in the output file of COMPAR (description in Section 10.7.5).
3. To select pairs of clocks that shall be included in the summary files on clock statistics extracted from zero–difference network solutions by the program GPSXTR (description in Section 7.9.2).

24.7.18. Cluster Definitions (Input)

Type: ASCII

Directory: Campaign–specific directory `STA`.

Extension: `CLU`

Content: Cluster definitions to group stations.

Created by: User–defined, supported by menu "Menu>Campaign>Edit station files>Cluster definition file".

Used by: SNGDIF ("Menu>Processing>Create baseline files") to group baselines into clusters or GPSEST ("Menu>Processing>Parameter estimation") and ADDNEQ2 ("Menu >Processing>Combine normal equation systems") to stack scaling factor parameters of groups of stations.

Example: Figure 24.56 and `${X}/DOC/EXAMPLE.CLU`.

The maximum number of clusters allowed is 100 (00–99).

Application of cluster files:

- Scaling factor parameters in GPSEST and ADDNEQ2:
  When estimating scaling factors for geophysical deformation models (see Section 10.4) stations may be grouped to estimate only one scaling factor per group (e.g., several stations at one location). When applying this file, only one parameter is setup for all stations with the same cluster number. This feature is available in GPSEST when the parameters are defined accordingly, but also in ADDNEQ2 where the parameters are stacked considering the cluster definitions.

```
SNGDIF: CLUSTER DEFINITION FILE
--------------------------------------------------------------------------------
STATION NAME      CLU
****************  ***

ABMF 97103M001      2
ABPO 33302M001      3
ADE1 50109S001      3
ADE2 50109S001      3
ADIS 31502M001      3
AFKB                1
AIRA 21742S001      3
AJAC 10077M005      1
ALBH 40129M003      2
ALGO 40104M002      2
ALIC 50137M001      3
ALME 13437M001      1
```

Figure 24.56.: Example for cluster definition input (`CLU`) file.

- Creation of baselines by SNGDIF:
  By using such a cluster definition file it is possible to define clusters of sites. The intention is to handle all sites within a cluster with correct correlations in program GPSEST. If the correct handling of the correlations is not required (not the highest accuracy requirements) a subdivision into clusters is not necessary. Due to memory and computer time limitations it is not recommended to handle more than about 40 sites with correct correlations. If your network consists of considerably more sites, you may use the cluster definition file to form clusters of baselines with the program SNGDIF, then process the individual clusters (with correct correlations), and finally combine the cluster normal equation using ADDNEQ2 to a complete network solution. It only makes sense to specify a cluster definition file, if you also specify a cluster definition output file in SNGDIF (see next section).

24.7.19. Cluster Definitions (Output)

Type:	ASCII
Directory:	Campaign–specific directory `STA`.
Extension:	`CLB`
Content:	Cluster definitions to be used in connection with GPSEST.
Created by:	SNGDIF ("Menu>Processing>Create baseline files"), MKCLUS ("Menu>Service >Automated processing>Form clusters").
Used by:	In an automatic processing scheme for the selection of the observation files (e.g., GPSEST).
Example:	Figure 24.57 and `${X}/DOC/EXAMPLE.CLB`.

If you specify a cluster definition input (`CLU`) file in SNGDIF (see previous section) containing, e.g., n clusters, you should also specify a cluster definition output (`CLB`) filename (e.g., `CLUST_`). SNGDIF then creates, according to the cluster numbers `cc` specified in the `CLU` file, n files with the filenames `CLUST_cc.CLB`. In each of these files the baselines belonging to the cluster number `cc` are stored. Each baseline is assigned to the cluster the first of the two stations belongs to.

A cluster definition output file may be used for a cluster–wise parallelization of the processing in the BPE. In each parallel run of program GPSEST the observation file selection may be specified by inserting the content of the file to the corresponding keyword in the program input file using a putkey–command (see Section 22.6.4).

The analogue procedure is available in program MKCLUS (description in Section 22.12.1.4), e.g., to assign zero–difference observation files into regional or global clusters. The file-

```
DIOF1960
ATZU1960
AAAU1960
B2CN1960
B2WH1960
BLKK1960
BLK41960
BLM11960
BLMK1960
FTPD1960
BRFS1960
...
```

Figure 24.57.: Cluster definition output (`CLB`) file for one particular cluster (from SNGDIF).

names are listed in the corresponding cluster output files and may be introduced into parallel running instances of GPSEST for the observation file selection.

Cluster definition file entries written by program MKCLUS include path and extension, while entries written by program SNGDIF contains the file basenames. The menu system handles both cases.

24.7.20. Event List File

Type:	ASCII
Directory:	Campaign–specific directory `STA`.
Extension:	`EVL`
Content:	Events in time series.
Created by:	User and FODITS ("Menu>Service>Coordinate tools>Analysis of time series").
Used by:	FODITS
Example:	Figure 24.58 and `${X}/DOC/EXAMPLE.EVL`.

The event list files may contain both a priori events for program FODITS and new events as detected by FODITS. See the header of the example file in Figure 24.58 for more details. The usage of the file is explained in Chapter 11 .

24.7.21. Earthquake List File

Type:	ASCII
Directory:	Campaign–specific directory `STA`.
Extension:	`ERQ`
Content:	Earthquake database.
Created by:	to be downloaded by the user from U. S. Geological Survey (USGS) Earthquake Hazards Program `http://neic.usgs.gov/neis/epic/epic_global.html` .
Used by:	FODITS ("Menu>Service>Coordinate tools>Analysis of time series").
Example:	Figure 24.59 and `${X}/DOC/EXAMPLE.ERQ`.

The usage of the file is explained in Chapter 11 .

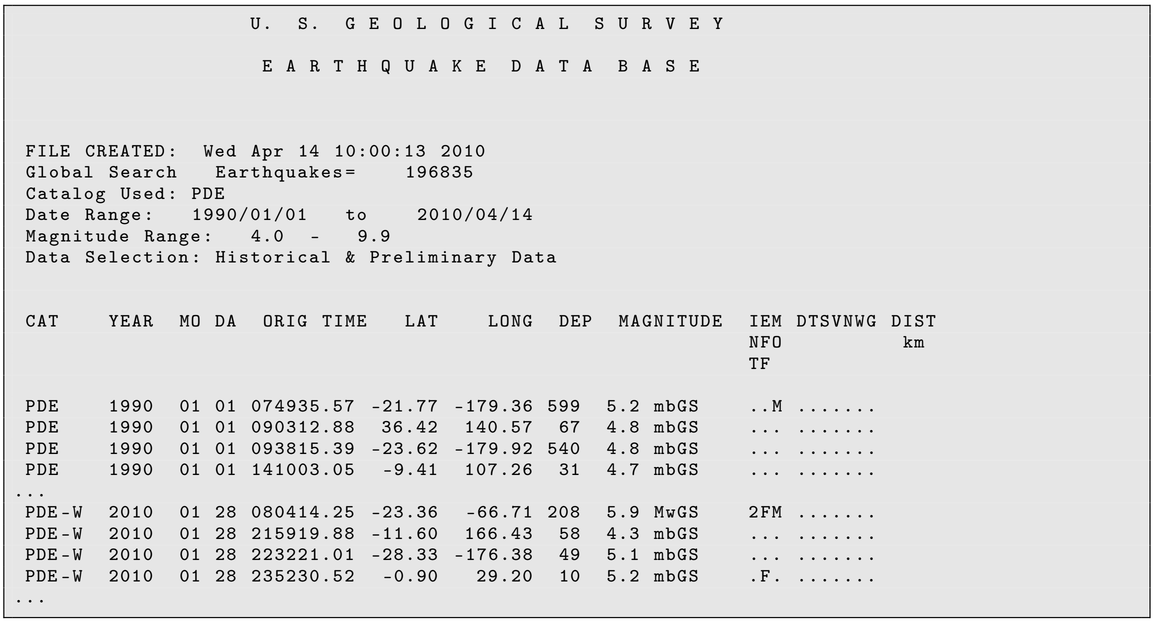
24.7.22. Session Table

Type:	ASCII, menu input file format
Directory:	Campaign–specific directory `STA`.
Extension:	`SES`
Content:	Time interval covering all observations which should be processed together. Program input panel format.
Created by:	The file is copied to the campaign during the campaign setup ("Menu >Campaign>Create new campaign"). Edit the file using "Menu>Campaign>Edit session table".
Used by:	All programs.
Example:	Figure 3.3 and `${X}/SKL/SESSIONS.SES`.

Two types of session tables exist: *fixed* session tables with explicitly defined sessions, and *open* session tables, with the doy in the session name specified using wildcards, so that it

```
FODITS - TIME SERIES EVENT LIST FILE
------------------------------------
If created by FODITS this file contains all parameters of the functional model.
If created by the user this file allows it to test, set, or remove parameters in the functional model.

The first 50 lines may contain an header. The file begins with EVL at the beginning of the line.
Empty lines are skipped. No alphabetical order of stations mandatory.
Blanks in 'From Epoch', 'To Epoch' means open. Wrong blanks are reported as error and the program FODITS stops.
Blanks in 'Periods' means all functional period defined in the PANEL.
Blanks in 'Amplitude' means estimate it. Non-blanks in 'Amplitude' means set exactly the given values.

Description:
------------
Flg :   - NEW, tested by FODITS, removed if non-significant.
        - SET, tested by FODITS, but always significant.
        - NOT, interval of no events (Evnt).

Evnt:   - JUMP, discontinuity (use 'Form Epoch').
        - VELO, velocity change (use 'From Epoch').
        - OUTL, outlier (use 'From Epoch' for one outlier, use both 'From Epoch' and 'To Epoch' for a sequence).
        - PERI, functional period ('From Epoch' and 'To Epoch' are optional, 'Period' must be defined).

-----------------------------------------------------------------------------------------------------------------
EVL                        From Epoch           To Epoch             Period      Amplitude [m]           Remark
STATION NAME     Flg Evnt  YYYY-MM-DD HH:MM:SS  YYYY-MM-DD HH:MM:SS  DDDD.DDDD   North   East    Up      RRRRRRRR
-----------------------------------------------------------------------------------------------------------------
ACOR 13434M001   NEW JUMP  2006-11-01 06:00:00                                   0.0098 -0.0050  0.0211  EVL-JUMP
ACOR 13434M001   NEW OUTL  2006-11-01 06:00:00                                                           EVL-OUTL
ACOR 13434M001   NEW OUTL  2007-03-21 18:00:00                                                           SNOW ANT
ACOR 13434M001   NEW OUTL  2007-03-28 12:00:00                                   0.0060  0.0026  0.0192  EVL-OUTL
ACOR 13434M001   SET PERI                                             365.2500   0.0004  0.0011  0.0016  EVL-PERI
ACOR 13434M001   NEW PERI                                             182.6250   0.0008  0.0004  0.0008  EVL-PERI

AJAC 10077M005   NEW JUMP  2006-11-01 06:00:00                                                           EVL-JUMP
AJAC 10077M005   NEW OUTL  2005-11-15 06:00:00                                  -0.0005 -0.0020  0.0122  EVL-OUTL
AJAC 10077M005   NEW OUTL  2006-11-01                                                                    RAIN
AJAC 10077M005   SET PERI                                             365.2500   0.0011  0.0004  0.0008  EVL-PERI
AJAC 10077M005   NEW PERI                                             182.6250   0.0006  0.0003  0.0001  EVL-PERI
AJAC 10077M005   NOT PERI  2006-11-15 06:00:00 2007-11-15 06:00:00    182.6250                           EVL-PERI

ALAC 13433M001   NEW JUMP  2006-02-22 06:00:00                                                           EVL-JUMP
ALAC 13433M001   NOT PERI                      2007-11-15 06:00:00                                       MULTIPAT
...
```

Figure 24.58.: Example of an event list (EVL) file.



```
                   U.  S.  G E O L O G I C A L  S U R V E Y

                    E A R T H Q U A K E  D A T A  B A S E


 FILE CREATED:  Wed Apr 14 10:00:13 2010
 Global Search   Earthquakes=    196835
 Catalog Used: PDE
 Date Range:   1990/01/01   to    2010/04/14
 Magnitude Range:   4.0  -   9.9
 Data Selection: Historical & Preliminary Data


 CAT    YEAR  MO DA  ORIG TIME   LAT    LONG  DEP  MAGNITUDE  IEM DTSVNWG DIST
                                                              NFO          km
                                                              TF

 PDE    1990  01 01 074935.57 -21.77 -179.36 599  5.2 mbGS    ..M .......
 PDE    1990  01 01 090312.88  36.42  140.57  67  4.8 mbGS    ... .......
 PDE    1990  01 01 093815.39 -23.62 -179.92 540  4.8 mbGS    ... .......
 PDE    1990  01 01 141003.05  -9.41  107.26  31  4.7 mbGS    ... .......
...
 PDE-W  2010  01 28 080414.25 -23.36  -66.71 208  5.9 MwGS    2FM .......
 PDE-W  2010  01 28 215919.88 -11.60  166.43  58  4.3 mbGS    ... .......
 PDE-W  2010  01 28 223221.01 -28.33 -176.38  49  5.1 mbGS    ... .......
 PDE-W  2010  01 28 235230.52  -0.90   29.20  10  5.2 mbGS    .F. .......
...
```

Figure 24.59.: Example of an Earthquake list (ERQ) file.

can be used indifferently for any date. The two types may not be mixed. For details see Section 3.4 and online help.

24.8. Atmosphere Related Files

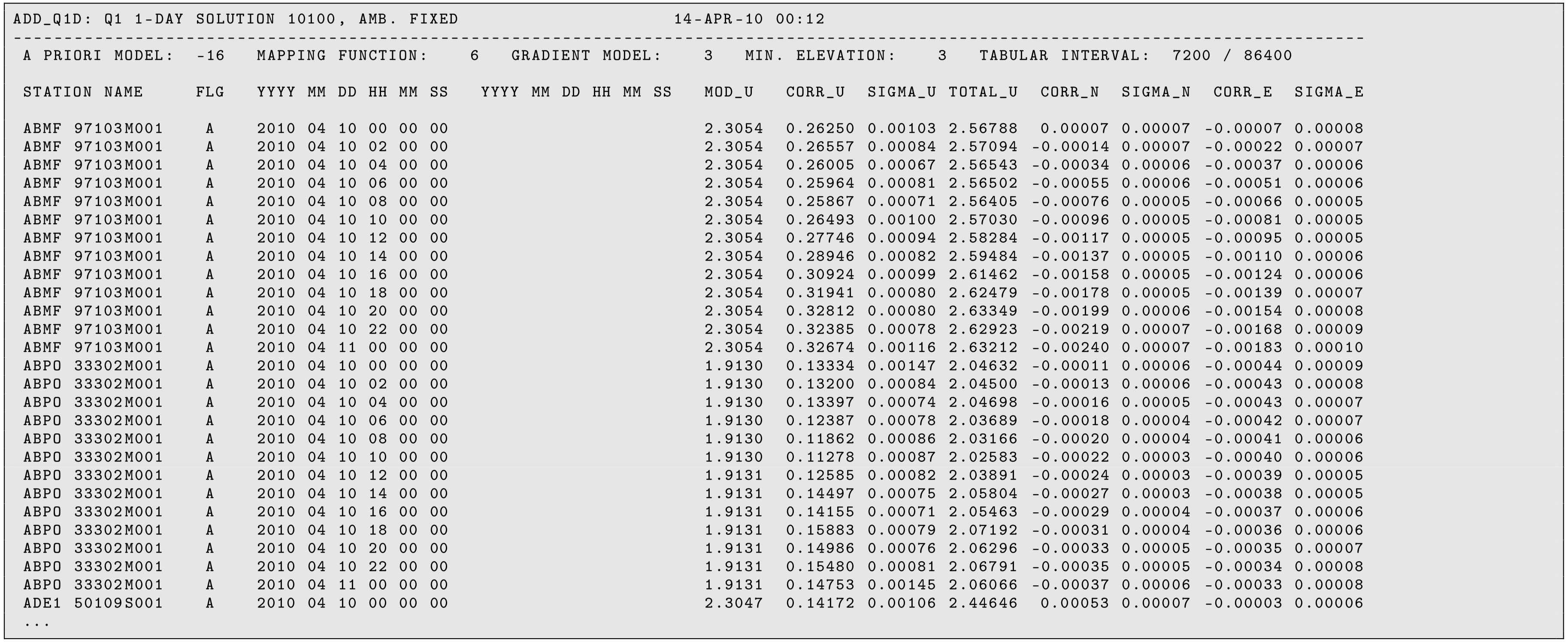
The atmosphere related file types, explained in this section, are listed in Table 24.7.

Table 24.7.: Atmosphere related files.

File Description	Extension	Description in Section
Troposphere related files		
Troposphere parameter file	TRP	24.8.1
Tropospheric SINEX file	TRO	24.8.2
Meteo and water vapor radiometer data	MET	24.8.3
Ionosphere related files		
Ionosphere models	ION	24.8.4
Ionosphere IONEX maps	INX	24.8.5

24.8.1. Troposphere Parameter File

Type: ASCII
Directory: Campaign–specific directory ATM.
Extension: TRP
Content: Tropospheric zenith delays and gradients (estimated).
Created by: GPSEST ("Menu>Processing>Parameter estimation") or ADDNEQ2 ("Menu>Processing >Combine normal equation systems").



```
ADD_Q1D: Q1 1-DAY SOLUTION 10100, AMB. FIXED                     14-APR-10 00:12
-------------------------------------------------------------------------------------------------------------------------------------
 A PRIORI MODEL:  -16   MAPPING FUNCTION:    6   GRADIENT MODEL:    3   MIN. ELEVATION:    3   TABULAR INTERVAL:  7200 / 86400

 STATION NAME     FLG   YYYY MM DD HH MM SS   YYYY MM DD HH MM SS   MOD_U   CORR_U  SIGMA_U TOTAL_U  CORR_N  SIGMA_N  CORR_E  SIGMA_E

 ABMF 97103M001    A    2010 04 10 00 00 00                         2.3054  0.26250 0.00103 2.56788  0.00007 0.00007 -0.00007 0.00008
 ABMF 97103M001    A    2010 04 10 02 00 00                         2.3054  0.26557 0.00084 2.57094 -0.00014 0.00007 -0.00022 0.00007
 ABMF 97103M001    A    2010 04 10 04 00 00                         2.3054  0.26005 0.00067 2.56543 -0.00034 0.00006 -0.00037 0.00006
 ABMF 97103M001    A    2010 04 10 06 00 00                         2.3054  0.25964 0.00081 2.56502 -0.00055 0.00006 -0.00051 0.00006
 ABMF 97103M001    A    2010 04 10 08 00 00                         2.3054  0.25867 0.00071 2.56405 -0.00076 0.00005 -0.00066 0.00005
 ABMF 97103M001    A    2010 04 10 10 00 00                         2.3054  0.26493 0.00100 2.57030 -0.00096 0.00005 -0.00081 0.00005
 ABMF 97103M001    A    2010 04 10 12 00 00                         2.3054  0.27746 0.00094 2.58284 -0.00117 0.00005 -0.00095 0.00005
 ABMF 97103M001    A    2010 04 10 14 00 00                         2.3054  0.28946 0.00082 2.59484 -0.00137 0.00005 -0.00110 0.00006
 ABMF 97103M001    A    2010 04 10 16 00 00                         2.3054  0.30924 0.00099 2.61462 -0.00158 0.00005 -0.00124 0.00006
 ABMF 97103M001    A    2010 04 10 18 00 00                         2.3054  0.31941 0.00080 2.62479 -0.00178 0.00005 -0.00139 0.00007
 ABMF 97103M001    A    2010 04 10 20 00 00                         2.3054  0.32812 0.00080 2.63349 -0.00199 0.00006 -0.00154 0.00008
 ABMF 97103M001    A    2010 04 10 22 00 00                         2.3054  0.32385 0.00078 2.62923 -0.00219 0.00007 -0.00168 0.00009
 ABMF 97103M001    A    2010 04 11 00 00 00                         2.3054  0.32674 0.00116 2.63212 -0.00240 0.00007 -0.00183 0.00010
 ABPO 33302M001    A    2010 04 10 00 00 00                         1.9130  0.13334 0.00147 2.04632 -0.00011 0.00006 -0.00044 0.00009
 ABPO 33302M001    A    2010 04 10 02 00 00                         1.9130  0.13200 0.00084 2.04500 -0.00013 0.00006 -0.00043 0.00008
 ABPO 33302M001    A    2010 04 10 04 00 00                         1.9130  0.13397 0.00074 2.04698 -0.00016 0.00005 -0.00043 0.00007
 ABPO 33302M001    A    2010 04 10 06 00 00                         1.9130  0.12387 0.00078 2.03689 -0.00018 0.00004 -0.00042 0.00007
 ABPO 33302M001    A    2010 04 10 08 00 00                         1.9130  0.11862 0.00086 2.03166 -0.00020 0.00004 -0.00041 0.00006
 ABPO 33302M001    A    2010 04 10 10 00 00                         1.9130  0.11278 0.00087 2.02583 -0.00022 0.00003 -0.00040 0.00006
 ABPO 33302M001    A    2010 04 10 12 00 00                         1.9131  0.12585 0.00082 2.03891 -0.00024 0.00003 -0.00039 0.00005
 ABPO 33302M001    A    2010 04 10 14 00 00                         1.9131  0.14497 0.00075 2.05804 -0.00027 0.00003 -0.00038 0.00005
 ABPO 33302M001    A    2010 04 10 16 00 00                         1.9131  0.14155 0.00071 2.05463 -0.00029 0.00004 -0.00037 0.00006
 ABPO 33302M001    A    2010 04 10 18 00 00                         1.9131  0.15883 0.00079 2.07192 -0.00031 0.00004 -0.00036 0.00006
 ABPO 33302M001    A    2010 04 10 20 00 00                         1.9131  0.14986 0.00076 2.06296 -0.00033 0.00005 -0.00035 0.00007
 ABPO 33302M001    A    2010 04 10 22 00 00                         1.9131  0.15480 0.00081 2.06791 -0.00035 0.00005 -0.00034 0.00008
 ABPO 33302M001    A    2010 04 11 00 00 00                         1.9131  0.14753 0.00145 2.06066 -0.00037 0.00006 -0.00033 0.00008
 ADE1 50109S001    A    2010 04 10 00 00 00                         2.3047  0.14172 0.00106 2.44646  0.00053 0.00007 -0.00003 0.00006
 ...
```

Figure 24.60.: Troposphere estimates in TRP file format.

Used by: The files may be introduced in GPSEST as a priori troposphere information. In CODSPP ("Menu>Processing>Code-based clock synchronization"), MAUPRP ("Menu>Processing>Phase preprocessing"), CLKEST ("Menu>Service> *Clock* tools>Epoch-wise clock interpolation"), GPSSIM ("Menu>Service>Generate simulated observation data") they may be introduced as known. In ADDNEQ2 the files may be introduced to fix the troposphere parameters to their a priori values).

Example: Figure 24.60 and `${X}/DOC/EXAMPLE.TRP`.

The troposphere files contain the following information:

1. Title.
2. Header line:
   - `A PRIORI MODEL`:

Saastamoinen	= -1,
Hopfield (Remondi)	= -2,
Essen and Froome	= -3,
Marini-Murray (SLR)	= -4,
Saastamoinen with Niell dry mapping	= -5,
GPT with GMF dry+wet mapping	= -6,
ECMWF with VMF1 dry+wet mapping	= -7,
Mendes-Pavlis (SLR)	= -8,
Saastamoinen dry part only	= -11,
Hopfield dry part only	= -12,
Simplified Hopfield dry part only	= -13,
Saastamoinen dry with Niell dry mapping	= -15,
GPT dry with GMF dry mapping	= -16,
ECMWF dry with VMF1 dry mapping	= -17.

The specified a priori troposphere model is used to correct for the main effect of the tropospheric delay (see also the standard atmosphere model definition in the file `${X}/GEN/CONST.` of Section 24.3.1).
The positive model numbers (1 up to 17) correspond to the same models but with observed meteorological values used for the computation of the a priori troposphere zenith delay.

   - `MAPPING FUNCTION`:

$1/\cos(z)$	= 1,
Hopfield	= 2,
Dry Niell	= 3,
Wet Niell	= 4,
Dry GMF	= 5,
Wet GMF	= 6,
Dry VMF1	= 7,
Wet VMF1	= 8.

   - `GRADIENT MODEL`:

No estimation	= 0,
Tilting	= 1,
Linear	= 2,
TANZ from MacMillan (1995)	= 3,
CHENHER from Chen and Herring (1997)	= 4.

   - `MIN. ELEVATION`: cut–off angle, in degrees

- TABULAR INTERVAL: for zenith parameters and for gradient parameters, in seconds.

3. Data part:
    - Station name.
    - Station flag (see Section 24.7.1).
    - Epoch for piecewise linear parameter representation (second epoch is blank, it was used in older versions of the software for piecewise constant representation).
    - Zenith delay (total effect) from the a priori model (zero, if observed meteo used), in m.
    - Estimated value (correction to the a priori zenith value), in m.
    - Estimated formal RMS of the correction, in m.
    - Total zenith delay (sum of a priori value and estimated correction), in m.
    - Gradient parameters and formal RMS in north and east directions, in m.

If a troposphere is introduced, only the estimated values (CORR_U, CORR_N, CORR_E) are used. The a priori zenith delay is reconstructed using the model specified in the header line. For that reason the input field where a troposphere model can be selected becomes inactive if a troposphere estimate file is introduced.

24.8.2. Tropospheric SINEX File

Type: ASCII
Directory: Campaign–specific directory ATM.
Extension: TRO
Content: Station–specific total zenith path delay and gradient estimates together with the station coordinates (optional record).
Created by: GPSEST ("Menu>Processing>Parameter estimation") and ADDNEQ2 ("Menu >Processing>Combine normal equation systems").
Used by: Exchange format internationally adopted. The utility TROTRO ("Menu >Conversion>Manipulate troposphere SINEX files") allows to remove the records indicated by section TYPE 003: HANDLING OF STATION PROBLEMS of a station information file.
Example: Figure 24.61.

The interested user is referred to Section 12.5.3. Details on the tropospheric SINEX format may be gathered from ftp://ftp.igs.org/pub/data/format/sinex_tropo.txt, the current version of the format specifications. Header information is gathered from file ${X}/GEN/SINEX., see Section 24.3.18.

24.8.3. Meteo and Water Vapor Radiometer Data

Type: ASCII
Directory: Campaign–specific directory ATM.
Extension: MET
Content: Station surface meteorological data or water vapor radiometer data.
Created by: RXMBV3 ("Menu>RINEX>Import RINEX to Bernese format>Meteo files").
Used by: GPSEST ("Menu>Processing>Parameter estimation"), GPSSIM ("Menu>Service>Generate simulated observation data"), and MAUPRP ("Menu>Processing>Phase preprocessing"), as a priori troposphere information.
Example: Figures 24.62 (type 1). More example files are available in the distribution in ${X}/DOC.

```
%=TRO 0.01 COD 14:198:36405 IGS 14:198:00000 14:198:28800 P  MIX
*-------------------------------------------------------------------------------
+FILE/REFERENCE
*INFO_TYPE_________ INFO________________________________________________________
 DESCRIPTION        CODE, Astronomical Institute, University of Bern
 OUTPUT             Troposphere zenith path delay estimates
 CONTACT            code@aiub.unibe.ch
 SOFTWARE           Bernese GNSS Software Version 5.2
 HARDWARE           UBELIX: Linux, x86_64
-FILE/REFERENCE
*-------------------------------------------------------------------------------
+TROP/DESCRIPTION
*_________KEYWORD_____________ __VALUE(S)_______________________________________
 ELEVATION CUTOFF ANGLE                             3
 SAMPLING INTERVAL                                180
 SAMPLING TROP                                   3600
 TROP MAPPING FUNCTION         WET VMF
 SOLUTION_FIELDS_1             TROTOT STDDEV TGNTOT STDDEV TGETOT STDDEV
-TROP/DESCRIPTION
*-------------------------------------------------------------------------------
+TROP/STA_COORDINATES
*SITE PT SOLN T __STA_X_____ __STA_Y_____ __STA_Z_____ SYSTEM REMRK
 ADIS  A    1 P  4913652.675  3945922.745   995383.418 IGb08  COD
 ALIC  A    1 P -4052052.524  4212836.008 -2545104.878 IGb08  COD
...
 ZIM2  A    1 P  4331299.745   567537.491  4633133.830 IGb08  COD
 ZIMM  A    1 P  4331296.932   567556.049  4633134.045 IGb08  COD
-TROP/STA_COORDINATES
*-------------------------------------------------------------------------------
+TROP/SOLUTION
*SITE ____EPOCH___ TROTOT STDDEV  TGNTOT STDDEV  TGETOT STDDEV
 ADIS 14:198:01800 1880.0    0.7   0.254  0.066  -0.275  0.084
 ADIS 14:198:05400 1882.8    0.6   0.474  0.059  -0.091  0.075
 ADIS 14:198:09000 1882.2    0.6   0.693  0.061   0.093  0.078
 ADIS 14:198:12600 1878.2    0.6   0.912  0.072   0.277  0.091
 ADIS 14:198:16200 1876.1    0.6   1.131  0.088   0.461  0.112
 ADIS 14:198:19800 1876.0    0.8   1.351  0.108   0.645  0.136
 ADIS 14:198:23400 1879.1    1.0   1.570  0.129   0.829  0.163
 ADIS 14:198:27000 1885.2    2.9   1.789  0.152   1.013  0.190
 ALIC 14:198:01800 2228.0    0.6  -0.164  0.047  -0.071  0.050
 ALIC 14:198:05400 2222.6    0.5  -0.173  0.047  -0.106  0.051
 ALIC 14:198:09000 2221.8    0.5  -0.183  0.060  -0.141  0.066
...
 ZIMM 14:198:19800 2198.7    0.9  -0.303  0.149   0.982  0.124
 ZIMM 14:198:23400 2196.5    1.0  -0.262  0.177   1.134  0.148
 ZIMM 14:198:27000 2190.2    3.3  -0.221  0.206   1.286  0.173
-TROP/SOLUTION
%=ENDTRO
```

Figure 24.61.: Tropospheric SINEX (TRO) file.

```
EXAMPLE FOR METEO FILE OF TYPE 1 (BERNESE GNSS SOFTWARE VERSION 5.2)
STATION : 7810 ZIMM          UTC-LOCAL TIME(HOURS) =  0 TYP= 1  #VALUES=  3  MOD=  0
  JJ MM DD HH MM SS  PPP.PP  TT.TT  HH.HH
   3 11  8 17 14 38  914.20   2.50  69.00
   3 11  8 17 43 28  914.20   2.50  69.00
   3 11  9  3  8 36  918.00   2.70  67.00
   3 11  9  3 17  6  918.00   2.70  67.00
   3 11  9 18 23 27  919.70   5.80  62.00
   3 11  9 19  3 43  919.70   5.80  62.00
   3 11  9 19 27 37  919.70   5.80  62.00
   3 11 11 16 53 56  916.20   5.90  69.00
   3 11 11 17 11 49  916.20   5.90  69.00
   3 11 11 21 17 28  916.30   5.50  72.00
...
```

Figure 24.62.: Meteo (MET) file of type 1.

Remarks:

- There is one meteo file per station (session–independent). The time difference between subsequent epochs is not essential. If the subroutine `${LG}/METEO.f` gets a request to calculate tropospheric refraction at time t, this value is calculated by linear interpolation of the table values in this file, where the two nearest times of recorded meteo data are used.

- File structure:
  The first record characterizes the campaign. The second record defines the station name, the difference local time − UTC (meteorological data may be recorded in local time), and the data type. The following types of meteo files are allowed:

Type 1	Pressure (mbar), temperature (Celsius), humidity (%).
Type 2	Pressure (mbar), dry temperature, wet temperature (both in Celsius).
Type 3	dr (m), where dr is the total tropospheric zenith delay.
Type 4	Obsolete, replaced by troposphere parameter file (see Section 24.8.1).
Type 5	Pressure (mbar), temperature (Celsius), humidity (%), and zenith wet delay.
Type 6	Pressure (mbar), temperature (Celsius), humidity (%), and precipitable water vapor.

  The following models (mapping functions) are allowed for type 3:

Model 0	$1/\cos(z)$ mapping
Model 1	simplified Herring mapping

- The meteo file has to end with a blank line (or a line starting with −1).

24.8.4. Ionosphere Models

Type: ASCII

Directory: Campaign–specific directory `ATM`.

Extension: `ION`

Content: Ionosphere models (represented by sets of TEC parameters).

Created by: IONEST ("Menu>Service>Ionosphere tools>Local ionosphere model estimation", model type 1 only), GPSEST ("Menu>Processing>Parameter estimation", model types 2 and 3), ADDNEQ2 ("Menu>Processing>Combine normal equation systems", model type 2 and 3).

Used by: Programs ADDNEQ2 to define a priori values for the ionosphere model estimation, as well as MAUPRP ("Menu>Processing>Phase preprocessing"), GPSEST, and GPSSIM ("Menu>Service>Generate simulated observation data").

Example: Figure 24.63 (model type 2). Example files available for the three model types in `${X}/DOC`.

We support three types of ionosphere models:

Type 1 *Local* ionosphere models (see Section 13.3.1.2) may be generated and used in the following way: estimated by program IONEST (processing zero–difference L_4 observations) and introduced into programs MAUPRP and GPSEST.

Type 2 *Global/regional* ionosphere models (see Section 13.3.1.3) may be created using GPSEST.

Type 3 *Station–specific* ionosphere models (see Section 13.3.1.4) are treated exactly in the same way as global/regional models. One full set of ionosphere parameters is set up with respect to each station involved, however.

Ionosphere models may be introduced for the following purposes:

- MAUPRP and GPSEST to correct for the ionosphere effect when processing single–frequency data.

```
CODE'S RAPID IONOSPHERE MAPS FOR DAY 104, 2010                   15-APR-10 07:09
--------------------------------------------------------------------------------
MODEL NUMBER / STATION NAME                        : 1040-01
MODEL TYPE (1=LOCAL,2=GLOBAL,3=STATION)            :    2
MAXIMUM DEGREE OF SPHERICAL HARMONICS              :   15
MAXIMUM ORDER                                      :   15
DEVELOPMENT WITH RESPECT TO
  GEOGRAPHICAL (=1) OR GEOMAGNETIC (=2) FRAME      :    2
  MEAN (=1) OR TRUE (=2) POSITION OF THE SUN       :    1
MAPPING FUNCTION (0=NONE,1=COSZ,2=MSLM,3=ESM)      :    2
HEIGHT OF SINGLE LAYER AND ITS RMS ERROR (KM)      :  450.00     0.00
COORDINATES OF EARTH-CENTERED DIPOLE AXIS
  LATITUDE OF NORTH GEOMAGNETIC POLE (DEGREES)     :   79.93
  EAST LONGITUDE (DEGREES)                         :  -71.96
PERIOD OF VALIDITY
  FROM EPOCH / REFERENCE EPOCH (Y,M,D,H,M,S)       : 2010 04 14 00 00 00
  TO EPOCH                                         :
LATITUDE BAND COVERED
  MINIMUM LATITUDE (DEGREES)                       :  -88.10
  MAXIMUM LATITUDE (DEGREES)                       :   89.11
ADDITIONAL INFORMATION
  NUMBER OF CONTRIBUTING STATIONS                  :  120
  NUMBER OF CONTRIBUTING SATELLITES                :   51
  ELEVATION CUT-OFF ANGLE (DEGREES)                :   10
  MAXIMUM TEC AND ITS RMS ERROR (TECU)             :   50.32     0.56
COMMENT / WARNING                                  :
COEFFICIENTS
DEGREE  ORDER    VALUE (TECU)   RMS (TECU)
   0       0      13.14128103     0.0471
   1       0       2.03069152     0.0437
   1       1       6.12370400     0.0447
   1      -1       4.01780782     0.0411
   2       0      -5.15873430     0.0430
   2       1       0.05442838     0.0429
   2      -1       0.88610159     0.0403
   2       2       1.56289796     0.0404
   2      -2       0.81094286     0.0405
   3       0      -1.17727869     0.0402
   3       1      -1.97485592     0.0428
...
  15      15      -0.06579750     0.0340
  15     -15       0.02577385     0.0340
```

Figure 24.63.: Ionosphere (ION) file of model type 2.

- MAUPRP, GPSEST, and GPSSIM to enable the corrections for higher order ionosphere (HOI) effects.
- GPSSIM as the basis for the simulation of the ionospheric effect on the measurements (see Section 20.6).

More information concerning ionosphere models is given in Chapter 13.

24.8.5. Ionosphere IONEX Maps

Type:	ASCII
Directory:	Campaign–specific directory ATM.
Extension:	INX
Content:	Earth–fixed grid maps (snapshots) of Total Electron Content (TEC) values (and of associated RMS errors, optionally).
Created by:	GPSEST ("Menu>Processing>Parameter estimation") and ADDNEQ2 ("Menu >Processing>Combine normal equation systems").
Used by:	Exchange format internationally adopted.
Example:	Figure 4.12.

More information concerning the Ionosphere model EXchange format (IONEX) format may be found in Sections 4.8, 24.3.19 and in Chapter 13. The interested user is finally referred to the IONEX format specifications (Schaer et al. 1998) or `ftp://ftp.igs.org/pub/data/format/ionex1.pdf`. Header information is gathered from file `${X}/GEN/IONEX.`, see Section 24.3.19.

24.9. Station and Atmosphere Related Grid Files

Type: ASCII
Directory: Campaign–specific directory `GRD`.
Extension: `GRD`
Content: VMF1 coefficients and station displacement models.
Created by: User defined.
Used by: CLKEST ("Menu>Service> Clock tools>Epoch-wise clock interpolation"), CODSPP ("Menu>Processing>Code-based clock synchronization"), MAUPRP ("Menu>Processing >Phase preprocessing"), GPSEST ("Menu>Processing>Parameter estimation") and GPSSIM ("Menu>Service>Generate simulated observation data") for VMF1 coefficients and GPSEST as well as GPSSIM also for station displacement models.
Examples: Figures 24.64 and 24.65, and files `${X}/DOC/EXAMP_VMF1.GRD` and `${X}/DOC/EXAMP_ALOAD.GRD`.

In *Bernese GNSS Software* Version 5.2, grid files can be used in following applications:

1. Grid files can be used for any gridded loading parameters.
   The content of the file is described in the header, see e.g. Figure 24.65. Geometrical deformations of the Earth crust from any source can be introduced as corrections for the station coordinates (as solid Earth tides). Three different sets of corrections may be introduced, where the keyword "`!  Data_types:`" is used to identify which type of correction is expected in the file. The following keywords correspond to the three types of corrections:

   `ATM-NT-LOAD:` for atmospheric pressure loading,

```
! Version:            1.0
! Source:             J. Boehm, TU Vienna (created: 2012-01-15)
! Data_types:         VMF1 (lat lon ah aw zhd zwd)
! Epoch:              2012 01 14 12 00  0.0
! Scale_factor:       1.e+00
! Range/resolution:   -90 90 0 360 2 2.5
! Comment:            http://ggosatm.hg.tuwien.ac.at/DELAY/GRID/VMFG/
 90.0   0.0 0.00116032  0.00055520  2.2935  0.0181
 90.0   2.5 0.00116032  0.00055520  2.2935  0.0181
 90.0   5.0 0.00116032  0.00055520  2.2935  0.0181
 90.0   7.5 0.00116032  0.00055520  2.2935  0.0181
 90.0  10.0 0.00116032  0.00055520  2.2935  0.0181
...
  0.0 327.5 0.00127622  0.00058690  2.3098  0.2503
  0.0 330.0 0.00127663  0.00056200  2.3102  0.2173
  0.0 332.5 0.00127606  0.00056046  2.3104  0.2329
  0.0 335.0 0.00127670  0.00054595  2.3119  0.2193
  0.0 337.5 0.00127665  0.00054182  2.3114  0.2165
...
-90.0 347.5 0.00119609  0.00047353  1.5336  0.0044
-90.0 350.0 0.00119609  0.00047353  1.5336  0.0044
-90.0 352.5 0.00119609  0.00047353  1.5336  0.0044
-90.0 355.0 0.00119609  0.00047353  1.5336  0.0044
-90.0 357.5 0.00119609  0.00047353  1.5336  0.0044
```

Figure 24.64.: Example of `GRD` file for VMF1 coefficients.

```
! Version:             1.0
! Source:              ATM Loading by L. Petrov (via J. Boehm: 2011-03-16)
! Data_types:          ATM-NT-LOAD (lat lon up north east)
! Epoch:               2011 02 28 06 00  0.0
! Scale_factor:        1.e+00
! Range/resolution:    -90 90 0 360 2.5 2.5
! Comment:             www.hg.tuwien.ac.at/~ecmwf1
 90.0   0.0 -0.00217 -0.00113 -0.00078
 90.0   2.5 -0.00217 -0.00113 -0.00078
 90.0   5.0 -0.00217 -0.00113 -0.00078
 90.0   7.5 -0.00217 -0.00113 -0.00078
 90.0  10.0 -0.00217 -0.00113 -0.00078
...
 57.5 350.0 -0.00322 -0.00124 -0.00102
 57.5 352.5 -0.00392 -0.00121 -0.00085
 57.5 355.0 -0.00621 -0.00126 -0.00073
 57.5 357.5 -0.00633 -0.00123 -0.00124
 55.0   0.0 -0.00529 -0.00085 -0.00132
 55.0   2.5 -0.00525 -0.00069 -0.00102
 55.0   5.0 -0.00569 -0.00056 -0.00075
 55.0   7.5 -0.00655 -0.00041 -0.00046
...
-90.0 350.0  0.00155  0.00203 -0.00055
-90.0 352.5  0.00155  0.00203 -0.00055
-90.0 355.0  0.00155  0.00203 -0.00055
-90.0 357.5  0.00155  0.00203 -0.00055
```

Figure 24.65.: Example of GRD file for atmospheric pressure loading.

OCN-NT-LOAD: for non–tidal ocean loading, and
HYDR-LOAD: for hydrostatic pressure loading.

The program only accepts input files with the expected keywords in the input field.

2. The VMF1 grid files can be downloaded from http://ggosatm.hg.tuwien.ac.at/DELAY/GRID/VMFG/.

Remark for advanced users: the values from the grid files are simply applied to the coordinates. It means you can put any displacement corrections in one of these files.

24.10. Solution Related Files

The solution related file types, explained in this section, are listed in Table 24.8.

Table 24.8.: Solution related files.

File Description	Extension	Description in Section
Normal equation file (binary)	NQ0	24.10.1
Normal equation file (ASCII)	FN0	24.10.1
SINEX file	SNX	24.10.2
Parameter estimates	EST	24.10.3

24.10.1. Normal Equation Files

Type: Binary
Directory: Campaign–specific directory SOL.
Extension: NQ0

Content: Normal equations and important a priori information.
Created by: GPSEST ("Menu>Processing>Parameter estimation"), ADDNEQ2 ("Menu>Processing >Combine normal equation systems"), SNX2NQ0 ("Menu>Conversion>SINEX to normal equations").
Used by: ADDNEQ2 to combine sequential solutions.
Example: —

Normal equation files contain important information concerning the parameter characterization, the a priori values used, as well as the normal equations including solution vector computed by programs GPSEST and ADDNEQ2. For more information on normal equations we refer to Chapter 9.

To transform binary normal equation files to ASCII files and vice versa (necessary if you have to change the computer platform), you can use the program NEQ2ASC in "Menu >Conversion>Normal equations (binary/ASCII)".

Note that the old normal equation file format from the old ADDNEQ program (extension `NEQ`) is not supported anymore.

24.10.2. SINEX File

Type: ASCII
Directory: Campaign–specific directory `SOL`.
Extension: `SNX`
Content: Coordinates, velocities, ERPs and other parameters in the Solution INdependent EXchange format (SINEX) Version 2.01 .
Created by: ADDNEQ2 ("Menu>Processing>Combine normal equation systems").
Used by: SNX2NQ0 ("Menu>Conversion>SINEX to normal equations"), SNX2STA ("Menu>Service >Station information files>Extract information from SINEX"), and SNX2SLR ("Menu>Service >SLR utilities>Prepare SLR data handling file"). Also the official format for the coordinate and velocity submissions to IERS.
Example: Example files are available in the anonymous CODE ftp `http://www.aiub.unibe.ch/download/`.

See Section 4.5 for more details and Kouba et al. (1996) or `http://www.iers.org` for a format definition.

General information is included in the SINEX files with help of the general file `${X}/GEN/SINEX.` (see Section 24.3.18).

24.10.3. File with Parameter Estimates

Type: ASCII
Directory: Campaign–specific directory `SOL`.
Extension: `EST`
Content: Parameter list with the corresponding estimate.
Created by: ADDNEQ2 ("Menu>Processing>Combine normal equation systems").
Used by: ADDNEQ2, only considered for parameter type "GNSS–specific parameters".
Example: Figures 24.66 and file `${X}/DOC/EXAMPLE.EST`.

```
...
ZIM2 14001M008        55688.000000000  55738.999652778            1         742           1      ...           0    4331299.78927         0.00011  #CRD
ZIM2 14001M008        55688.000000000  55738.999652778            1         743           2      ...           0     567537.43982         0.00014  #CRD
ZIM2 14001M008        55688.000000000  55738.999652778            1         744           3      ...           0    4633133.80299         0.00011  #CRD
ZIMM 14001M004        55688.000000000  55738.999652778            1         745           1      ...           0    4331296.97863         0.00013  #CRD
ZIMM 14001M004        55688.000000000  55738.999652778            1         746           2      ...           0     567555.99452         0.00014  #CRD
ZIMM 14001M004        55688.000000000  55738.999652778            1         747           3      ...           0    4633134.01416         0.00012  #CRD
...
ZIM2 14001M008        55688.000000000  55738.999652778           30           0           1      ...           0          0.00128         0.00009  #GSP
ZIM2 14001M008        55688.000000000  55738.999652778           30           0           2      ...           0          0.00011         0.00009  #GSP
ZIM2 14001M008        55688.000000000  55738.999652778           30           0           3      ...           0         -0.00118         0.00008  #GSP
ZIM2 14001M008        55688.000000000  55738.999652778           30           0           4      ...           0         -0.00053         0.00003  #GSP
...
                      55688.000000000  55738.999652778           27           1           1      ...           0          0.99995         0.00014  #HOI
                      55688.000000000  55738.999652778           27           2           1      ...           0          1.00000         0.00014  #HOI
                      55688.000000000  55738.999652778           27           3           1      ...           0          1.00000         0.00014  #HOI
...
MW  BLOCK IIA    035  55714.000000000  55738.999652778           12           1           1      ...           0          0.27900         0.00001  #SAO
MW  BLOCK IIA    035  55714.000000000  55738.999652778           12           1           2      ...           0          0.00001         0.00001  #SAO
MW  BLOCK IIA    035  55714.000000000  55738.999652778           12           1           3      ...           0          2.62200         0.00001  #SAO
MW  BLOCK IIR-B  061  55688.000000000  55738.999652778           12           1           1      ...           0          0.00010         0.00001  #SAO
...
IERS2000              55688.000000000  55688.000000000           10           1           1      ...          24        -41.54990         0.00707  #ERP
IERS2000              55688.000000000  55688.000000000           10           1           1      ...          24        355.54278         0.00726  #ERP
IERS2000              55688.000000000  55688.000000000           10           1           1      ...          24       -251.30574         0.00014  #ERP
IERS2000              55689.000000000  55689.000000000           10           1           2      ...          24        -41.63920         0.00552  #ERP
IERS2000              55689.000000000  55689.000000000           10           1           2      ...          24        357.35933         0.00561  #ERP
...
                      55688.000000000  55738.999652778           16           1           0      ...           1          0.00013         0.00001  #GCC
                      55688.000000000  55738.999652778           16           2           0      ...           2          0.00008         0.00001  #GCC
                      55688.000000000  55738.999652778           16           3           0      ...           3          0.00059         0.00001  #GCC
```

Figure 24.66.: Parameter estimates (`EST`) file.

The parameter estimate file reports the internal parameter description as it is displayed in Figure 24.66 with the following meaning of the columns:

Column 1 Parameter name (character with the length 20).
Columns 2–3 Validity interval for the parameter (two values in MJD).
In case of piecewise linear representation or epoch parameters the two values are identical.
Columns 4–10 Parameter description (seven integer numbers, three of them are removed in Figure 24.66 for plotting); the first number specifies the parameter type (see Table 24.9).
Column 11 Estimated parameter (a priori plus improvement).
Column 12 Formal error for the estimated parameter.
Column 13 A parameter type identifier key (see Table 24.9).

Table 24.9.: List of parameters with their parameter identifier numbers.

Number	Description	key
1	Station coordinates	#CRD
2	Receiver clocks	#RCO
3	Orbital elements	#ORB
4	Ambiguities[a]	
5	Receiver antenna offset parameters	#RAO
6	Site–specific troposphere parameters	#TRP
7	Local ionosphere models[b]	
8	Differential code biases	#DCB
9	Local troposphere models[b]	
10	Earth rotation parameters	#ERP
11	Stochastic orbit parameters	#SORB
12	Satellite antenna offset parameters	#SAO
13	Earth potential parameters[b]	
14	Resonance terms (hill theory)[b]	
15	Albedo parameters[b]	
16	Center of mass	#GCC
17	Stochastic ionosphere parameters[a]	
18	Receiver phase center variations	#RAP
19	Global ionosphere model parameters	#GIM
20	Not used	
21	Kinematic coordinates[a]	
22	Scaling factors for vienna grid files	#GRD
23	Epoch wise clocks station clocks	#RCK
24	Epoch wise clocks satellite clocks	#SCK
25	Satellite phase center variations	#SAP
26	Range biases	#RGB
27	Higher–order iono scaling parameters	#HOI
28	Helmert transformation parameters	#HLM
29	Not used	
30	GNSS–specific parameters	#GSP

[a] Parameter type not supported by ADDNEQ2.
[b] Experimental parameter type for internal use only, not supported by ADDNEQ2.

When introducing this file into ADDNEQ2, only the a priori values for parameter type 30 (GNSS–specific parameters) are set to the estimated values. For other parameter types the corresponding specific filetypes have to be used.

24.11. Output and Miscellaneous Files

The miscellaneous file types, explained in this section, are listed in Table 24.10.

Table 24.10.: Miscellaneous files.

File Description	Extension	Description in Section
Program output files		
Program output file	OUT, L??	24.11.1
Error message file	MSG	24.11.2
Summary output file	SUM	24.11.3
Listing output file	LST	24.11.4
Plot file	PLT	24.11.5
Program specific summary files		
Single point positioning file (XYZ)	SMC	24.11.6
Single point positioning file (BLH)	SME	24.11.6
RINEX pseudo graphics file	SMC	24.11.7
Further in– and output files		
Clock corrections, RINEX format	CLK	24.11.8
Residual file (binary)	RES	24.11.9
Residual file (ASCII)	FRS	24.11.9
Variance–covariance file	COV	24.11.10
Normal equation rescaling file	WGT	24.11.11
Receiver antenna PCV estimates (grid)	PHG	24.3.3
Receiver antenna PCV estimates (harmonic)	PHH	24.3.3
Observation editing file	EDT	24.11.12
Deletion list	DEL	24.11.13

24.11.1. Program Output Files

Type: ASCII
Directory: Campaign–specific directory OUT.
Extension: OUT, Lnn or nnn.
Content: Job output of all programs.
Created by: Each program.
Used by: User for documentation of the program run.
DEFXTR ("Menu>Orbits/EOP>Extract ORBGEN program output"), CODXTR ("Menu>Processing>Program output extraction>Code-based clock synchronization"), MPRXTR ("Menu>Processing>Program output extraction>Phase preprocessing"), and GPSXTR ("Menu>Processing>Program output extraction>Parameter estimation/stacking") to extract information from the output files.
Example: —

Each program generates a program output file containing at least a header with a title line, a list of the input files, and the settings of the most important options. The output file name may be generated automatically or specified by the user. In the first case the filename is `pgmname.Lnn` with a number `nn` (or `nnn` for numbers larger than 99) which is incremented for each program run. Errors and warnings may be included in the program output file or written in a special error message file (see Section 24.11.2). For details on output handling see Section 21.7.

The extraction programs are well suited for the generation of a summary file in an automated processing using the BPE.

24.11.2. Error Message Files

Type: ASCII
Directory: User–specific directory `${U}/WORK`.
Extension: `MSG`.
Content: Error and warning messages.
Created by: Each program.
Used by: User and BPE for error handling.
Example: —

Each program and subroutine writes error and warning messages to the error message file. Alternatively, these messages may be routed to the program output file (see Section 24.11.1). For details on error output handling see Section 21.7.

Errors are indicated with a string `***`, warnings with a string `###`. Usually the occurrence of an error causes the program to interrupt processing.

24.11.3. Summary Files

Type: ASCII
Directory: Campaign–specific directory `OUT`.
Extension: `SUM`
Content: Summaries of program output.
Created by: Several programs, e.g.:
ADDNEQ2 ("Menu>Processing>Combine normal equation systems") and COMPAR ("Menu>Service>Coordinate tools>Coordinate comparison"): weekly summary file, DEFXTR ("Menu>Orbits/EOP>Extract ORBGEN program output"): summary file of ORBGEN output file, MPRXTR ("Menu>Processing>Program output extraction>Phase preprocessing"): summary file of MAUPRP output file(s), GPSXTR ("Menu >Processing>Program output extraction>Parameter estimation/stacking"): divers summary file of GPSEST and ADDNEQ2 output file(s), RESRMS ("Menu>Service>Residual files>Create residual statistics"): summary file of residuals per station/baseline and satellite, RESCHK ("Menu>Service>Automated processing>Detect misbehaving stations/satellites"): summary file of satellite–wise residual statistics, QLRSUM ("Menu>Service>SLR utilities>Create quick-look residual statistics"): summary files with residual statistics, SATGRA ("Menu>Service>Bernese observation files>Create pseudographics"): summary of available observations during a repositioning event, and AMBCHK: summary of ambiguity check.

Used by: The user for quality control, to have available short summary files with the most important information from different programs. RESCHK further evaluates the residual summary file from program RESRMS and BASLST ("Menu>Service>Automated processing>Select baselines") reads ambiguity resolution summary file from program GPSXTR.

Example: —

24.11.4. List Files

Type: ASCII

Directory: Campaign–specific directory `OUT`.

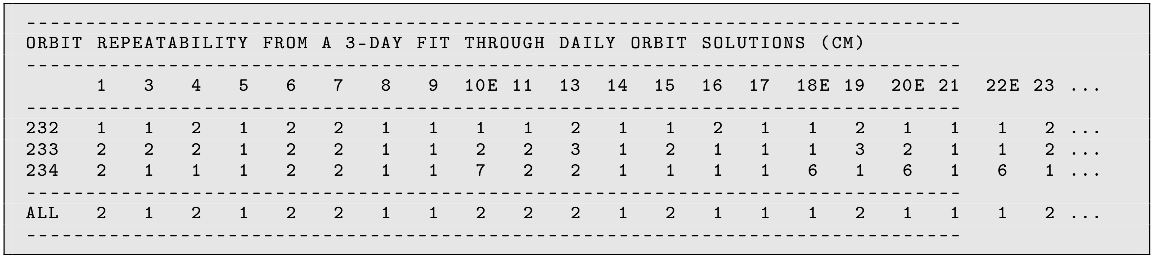
Extension: `LST`

Content: Contains special output information.

Created by: Several programs, e.g.:
ORBGEN ("Menu>Orbits/EOP>Create/update standard orbits"): list of fit RMS values, RNXGRA ("Menu>RINEX>RINEX utilities>Create observation statistics"): list of selected files, CODXTR ("Menu>Processing>Program output extraction>Code-based clock synchronization"): list of selected files, HELMR1 ("Menu>Service>Coordinate tools>Helmert transformation"): list of outliers, and RESRMS ("Menu>Service>Residual files>Create residual statistics"): histogram of residuals.

Used by: Operator to save important job output information. PREWEI ("Menu >Orbits/EOP>Set accuracy codes in precise orbits") read list file from ORBGEN to derive the accuracy codes for precise orbit files.

Example: Figure 24.67 (ORBGEN list file), `${X}/DOC/EXAMPLE1.LST` (ORBGEN list file), `${X}/DOC/EXAMPLE2.LST` (HELMR1 list file).



```
-------------------------------------------------------------------------------
ORBIT REPEATABILITY FROM A 3-DAY FIT THROUGH DAILY ORBIT SOLUTIONS (CM)
-------------------------------------------------------------------------------
      1   3   4   5   6   7   8   9  10E 11  13  14  15  16  17  18E 19  20E 21  22E 23 ...
-------------------------------------------------------------------------------
232   1   1   2   1   2   2   1   1   1   1   2   1   1   2   1   1   2   1   1   1   2 ...
233   2   2   2   1   2   2   1   1   2   2   3   1   2   1   1   1   3   2   1   1   2 ...
234   2   1   1   1   2   2   1   1   7   2   2   1   1   1   1   6   1   6   1   6   1 ...
-------------------------------------------------------------------------------
ALL   2   1   2   1   2   2   1   1   2   2   2   1   2   1   1   1   2   1   1   1   2 ...
-------------------------------------------------------------------------------
```

Figure 24.67.: ORBGEN list file.

24.11.5. Plot File

Type: ASCII

Directory: Campaign–specific directory `OUT`.

Extension: `PLT`

Content: Plot information.

Created by: ADDNEQ2 ("Menu>Processing>Combine normal equation systems") and COMPAR ("Menu>Service>Coordinate tools>Coordinate comparison"): to store coordinate residuals (N, E, U), ORBGEN ("Menu>Orbits/EOP>Create/update standard orbits"): to store residuals from the orbit fit, POLXTR ("Menu>Orbits/EOP>Handle EOP files >Concatenate IERS pole files"): to store ERP estimates, STDDIF ("Menu>Orbits/EOP

>Compare orbits>Standard orbits"): to store differences between orbits (in the radial, along–track, and out of plane directions), and QLRSUM ("Menu>Service >SLR utilities>Create quick-look residual statistics"): to store the SLR residuals.

Used by: The operator to produce plots (using an arbitrary plotting program). FODITS ("Menu>Service>Coordinate tools>Analysis of time series") to read the repeatability of the station coordinate series from program ADDNEQ2 .

Example: —

The plot files only contain the data to be plotted. You may import this information into any plotting program.

24.11.6. Single Point Positioning File

Type: ASCII
Directory: Campaign–specific directory OUT.
Extension: SMC, SME
Content: Contains the CODSPP summary of geocentric coordinates (SMC) or of ellipsoidal coordinates (SME).
Created by: CODSPP ("Menu>Processing>Code-based clock synchronization").
Used by: May be useful as a history of coordinate estimations using pseudorange observations.
Example: Figure 24.68.

The advantage of the SMC (and SME) files compared to the usual coordinate files (see Section 24.7.1) has to be seen in the circumstance that the RMS information is also stored. The information may be used to check estimated satellite clock corrections, see Section 15.4.

This file type cannot be re–introduced into any other Bernese program, however. If you specify always the same file in all CODSPP runs, the new results are appended to all previous results, instead of overwriting the previous results. The maximum number of lines for this file is 10 000 lines.

```
SINGLE POINT POSITIONING SOLUTIONS: GEOCENTRIC COORDINATES IN WGS-84
--------------------------------------------------------------------

STATION NAME      SESS F FR C T I EL NS  DT   #OBS   RMS(M)        X (M)      SX(M)  ...
------------------------------------------------------------------------------------...

ALGO 40104M002    0050 0 L3 2 E N 15 10  30   1991    0.11       918129.4128  0.0000  ...
AMC2 40472S004    0050 0 L3 2 E N 15 10  30   1879    0.12     -1248596.1568  0.0000  ...
BRUS 13101M004B   0050 0 L3 2 E N 15 10  30   1802    0.13      4027893.7971  0.0000  ...
NRC1 40114M001    0050 0 L3 2 E N 15 10  30   1974    0.09      1112777.2352  0.0000  ...
NYA1 10317M003    0050 0 L3 2 E N 15 10  30   1558    1.24      1202433.8810  0.0000  ...
ONSA 10402M004    0050 0 L3 2 E N 15 10  30   1909    0.15      3370658.5904  0.0000  ...
PIE1 40456M001    0050 0 L3 2 E N 15 10  30   1775    0.18     -1640916.8706  0.0000  ...
PTBB 14234M001B   0050 0 L3 2 E N 15 10  30   1809    0.13      3844060.0093  0.0000  ...
QAQ1 43007M001B   0050 0 L3 2 E N 15 10  30   1202    3.05      2170942.1550  0.0000  ...
REYK 10202M001    0050 0 L3 2 E N 15 10  30   2110    3.34      2587384.3566  0.0000  ...
THU3 43001M002    0050 0 L3 2 E N 15 10  30   2425    2.06       538093.5938  0.0000  ...
USNO 40451S003    0050 0 L3 2 E N 15 10  30   1842    0.11      1112189.8126  0.0000  ...
WAB1 14014M001    0050 0 L3 2 E N 15 10  30   2017    0.35      4327317.6576  0.0000  ...
WSRT 13506M005    0050 0 L3 2 E N 15 10  30   2051    0.09      3828735.9129  0.0000  ...
WTZR 14201M010    0050 0 L3 2 E N 15 10  30   1934    0.13      4075580.6005  0.0000  ...
YELL 40127M003B   0050 0 L3 2 E N 15 10  30   2142    0.17     -1224452.6187  0.0000  ...

------------------------------------------------------------------------------------...
```

Figure 24.68.: CODSPP summary (SMC) file.

Do not write single point positioning files with the same name in parallel CODSPP runs. The writing of the programs to the same file is not synchronized and may end in files which do not contain the complete information from all runs.

24.11.7. RINEX Pseudo Graphics

Type: ASCII
Directory: Campaign–specific directory OUT.
Extension: SMC
Content: Pseudo graphics of satellite observations contained in RINEX observation file.
Created by: RNXGRA ("Menu>RINEX>RINEX utilities>Create observation statistics").
Used by: Useful to identify tracking problems.
Example: Figure 4.7.

The file type has the same extension as the single point positioning file (see Section 24.11.6). For more information on the content of the RINEX pseudo graphics file see Section 4.2.5.

24.11.8. Clock Corrections, RINEX Format

Type: ASCII
Directory: Campaign–specific directory OUT.
Extension: CLK
Content: Satellite and station clock parameters in the official Clock RINEX format.
Created by: ADDNEQ2 ("Menu>Processing>Combine normal equation systems"), CCRNXC ("Menu >Service> *Clock* tools>Combine/manipulate clock RINEX files"), GPSEST ("Menu >Processing>Parameter estimation"), CODSPP ("Menu>Processing>Code-based clock synchronization"), and CLKEST ("Menu>Service> *Clock* tools>Epoch-wise clock interpolation").
Used by: CCRNXC — the clock RINEX utility (description in Section 15.3), RNXCLK ("Menu>RINEX>RINEX utilities>Extract satellite clock") to extract the satellite clock corrections, GPSEST, ADDNEQ2 to adjust the a priori values, and KINPRE ("Menu>Service>LEO utilities>Convert kinematic pos. to precise orbits") to provide the satellite clock corrections included in the resulting precise orbit file.
Example: Figure 4.13.

Note that clock RINEX files have the same extension as Bernese satellite clock files but reside in a different directory (OUT instead of ORB).

The processing programs indicate periods when the satellites are in eclipse by the letter E behind the clock record. This is not conforming to the international format description and must be removed using the option "Remove satellite clocks with eclipse flag" (panel "CCRNXC 2: Clock/Epoch Selection for Processing") before distributing the files.

For more details see Section 4.9.

```
IGSF_G: SOLUTION 14147001, 2. ITERATION                          07-JUL-14 09:14
--------------------------------------------------------------------------------

GENERAL INFORMATION:
--------------------

   Type of residual file:                   1
   Format of residual records:              1
   Elevation/Azimuth flag:                  0
   Program created the file:                GPSEST
   Difference level of observations:        1 sta.       1 sat.       0 epo.
   Number of parameters:                  493
   Total number of residual files:         14

Num  Station 1         Station 2         Reference epoch       Session     # Freq.  Type dt ...

  1  LHAZ 21613M002    SGOC 23501M003    2014-05-27 06:00:00   1470 0      1 3 0 0   1   30 ...
  2  LHAZ 21613M002    POLV 12336M001    2014-05-27 06:00:00   1470 0      1 3 0 0   1   30 ...
...

Num Epoch  Frq    Sat.         Value                    Flg      Elev     Azi

  1     1   3      3    7      0.418085578401927727D-02          0.00     0.00
  1     1   3      7   13      0.631839953638724250D-02          0.00     0.00
  1     1   3     13   16     -0.116839853742478628D-02          0.00     0.00
  1     1   3     16   19      0.416116792795218071D-02          0.00     0.00
  1     1   3     19   20      0.315030655485160072D-03          0.00     0.00
  1     1   3     20   23     -0.545191547657730702D-02          0.00     0.00
  1     1   3     23   27      0.157854712159410937D-02          0.00     0.00
  1     1   3     27   32      0.715862323084348606D-02          0.00     0.00
  1    11   3      3    7      0.201919783981645562D-01          0.00     0.00
  1    11   3      7   13     -0.210302566709358318D-01          0.00     0.00
  1    11   3     13   16      0.854776821347491556D-03          0.00     0.00
...
  1  1431   3     25   29      0.719095162841354441D-03          0.00     0.00
  1  1431   3     29   15     -0.359457023084910204D-02          0.00     0.00
  1  1431   3     15   21     -0.922058160150739103D-02          0.00     0.00
  2     1   3      7   13     -0.137904448526096693D-01          0.00     0.00
  2     1   3     13   16      0.225425334428896867D-01          0.00     0.00
  2     1   3     16   20     -0.151118534115338293D-01          0.00     0.00
  2     1   3     20   23     -0.375556988463471991D-02          0.00     0.00
  2    11   3      7   13      0.111582191569614131D-01          0.00     0.00
...
```

Figure 24.69.: ASCII version of residual (FRS) file.

24.11.9. Residual Files

Type: Binary
Directory: Campaign–specific directory OUT.
Extension: RES
Content: Residuals stemming from the processing programs.
Created by: CODSPP ("Menu>Processing>Code-based clock synchronization"), MAUPRP ("Menu >Processing>Phase preprocessing"), GPSEST ("Menu>Processing>Parameter estimation"), IONEST ("Menu>Service>Ionosphere tools>Local ionosphere model estimation"), and ORBGEN ("Menu>Orbits/EOP>Create/update standard orbits").
Used by: REDISP ("Menu>Service>Residual files>Display residual file") to browse residuals, RESRMS ("Menu>Service>Residual files>Create residual statistics") to check residuals for outliers, QLRSUM ("Menu>Service>SLR utilities>Create quick-look residual statistics") to create statistics on residuals of SLR observations.
Example: Figure 24.69 (ASCII example for single–difference phase residual file).

ASCII versions of a residual file may be created using RESFMT ("Menu>Conversion>Residual files>Binary to ASCII"). The header of the ASCII file specifies the type and format version, the program that created the residual file, the differencing level, number of parameters and observation files. A list for each station gives the reference epoch, frequencies, sampling, and list of satellites. The data part gives the residual in meters for every link and frequency.

For zero–difference residuals, in addition, azimuth and elevation are given. For the creation of binary from ASCII versions use FMTRES ("Menu>Conversion>Residual files>ASCII to binary"). For details see Section 6.6.

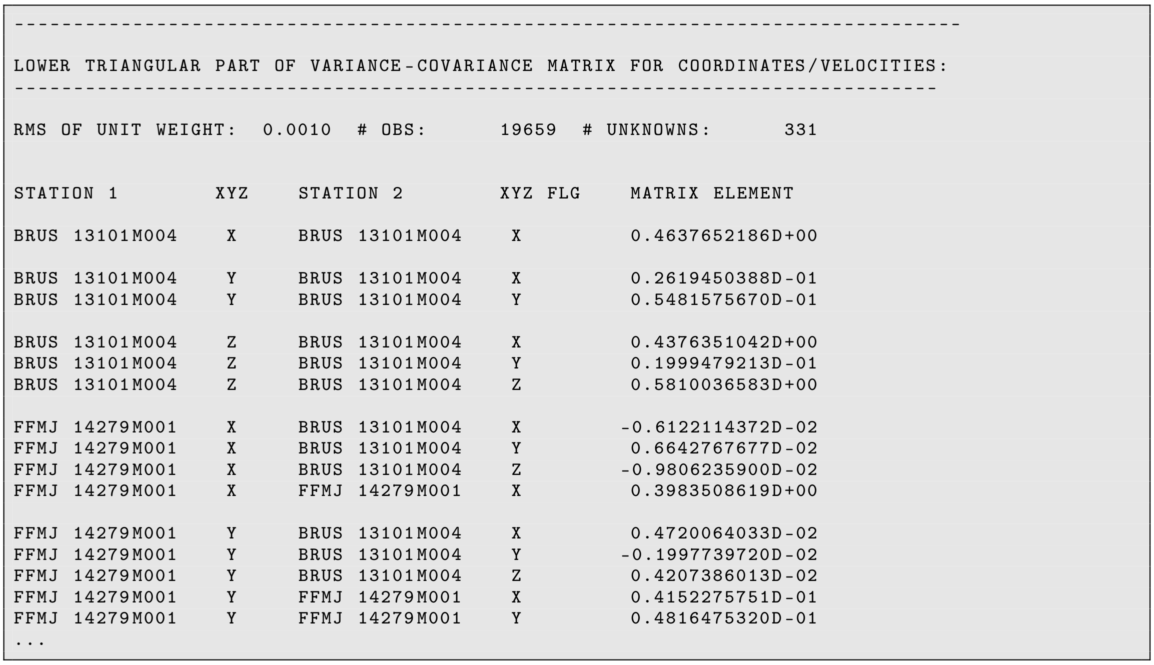
24.11.10. Variance–Covariance Matrix

Type: ASCII
Directory: Campaign–specific directory `OUT`.
Extension: `COV`
Content: Variance–covariance matrix of the least square adjustment.
Created by: GPSEST ("Menu>Processing>Parameter estimation"), ADDNEQ2 ("Menu>Processing >Combine normal equation systems"), and COMPAR ("Menu>Service>Coordinate tools >Coordinate comparison", only type 1).
Used by: COMPAR for combination of solutions[2].
Example: Figures 24.70 (type 1) and Figures 24.71 (type 2), `${X}/DOC/EXAMPLE.COV` (type 2).

There are two different file types that may be created.

1. Variance–covariance matrix of the station coordinates and velocities only (may be used as input together with the coordinate files in the program COMPAR).
2. The full variance–covariance matrix including a parameter characterization list at the beginning of the file. We do not recommend to specify this type of output, because this file cannot be used in any other Bernese programs and a big file may be created. The file may be useful for external investigations. An example file is given in the distribution.

[2] This feature should not be used anymore because ADDNEQ2 provides a much more complete tool for the same purpose.



```
--------------------------------------------------------------------------------

LOWER TRIANGULAR PART OF VARIANCE-COVARIANCE MATRIX FOR COORDINATES/VELOCITIES:
------------------------------------------------------------------------------

RMS OF UNIT WEIGHT:  0.0010  # OBS:      19659  # UNKNOWNS:      331


STATION 1         XYZ    STATION 2        XYZ FLG    MATRIX ELEMENT

BRUS 13101M004     X     BRUS 13101M004    X         0.4637652186D+00

BRUS 13101M004     Y     BRUS 13101M004    X         0.2619450388D-01
BRUS 13101M004     Y     BRUS 13101M004    Y         0.5481575670D-01

BRUS 13101M004     Z     BRUS 13101M004    X         0.4376351042D+00
BRUS 13101M004     Z     BRUS 13101M004    Y         0.1999479213D-01
BRUS 13101M004     Z     BRUS 13101M004    Z         0.5810036583D+00

FFMJ 14279M001     X     BRUS 13101M004    X        -0.6122114372D-02
FFMJ 14279M001     X     BRUS 13101M004    Y         0.6642767677D-02
FFMJ 14279M001     X     BRUS 13101M004    Z        -0.9806235900D-02
FFMJ 14279M001     X     FFMJ 14279M001    X         0.3983508619D+00

FFMJ 14279M001     Y     BRUS 13101M004    X         0.4720064033D-02
FFMJ 14279M001     Y     BRUS 13101M004    Y        -0.1997739720D-02
FFMJ 14279M001     Y     BRUS 13101M004    Z         0.4207386013D-02
FFMJ 14279M001     Y     FFMJ 14279M001    X         0.4152275751D-01
FFMJ 14279M001     Y     FFMJ 14279M001    Y         0.4816475320D-01
...
```

Figure 24.70.: Variance–covariance (`COV`) file of type 1.

```
EXAMPLE                                                                  25-AUG-15 09:55
--------------------------------------------------------------------------------

PARAMETERS:
----------

# STATION COORDINATES      :      24
# TROPOSPHERE FOR STATIONS :      56

TOTAL # OF PARAMETERS      :      80


PARAMETER CHARACTERIZATION LIST:
-------------------------------

PARAM  TYPE  STATION NAME     COORDINATE

   1     1   ZIMJ 14001M006       X
   2     1   ZIMJ 14001M006       Y
   3     1   ZIMJ 14001M006       Z
...
  22     1   VILL 13406M001       X
  23     1   VILL 13406M001       Y
  24     1   VILL 13406M001       Z

PARAM  TYPE  REQUEST  STATION NAME     COMPONENT

  25     6      1     ZIMJ 14001M006        U
  26     6      2     ZIMJ 14001M006        U
  27     6      3     ZIMJ 14001M006        U
...
  78     6     54     VILL 13406M001        U
  79     6     55     VILL 13406M001        U
  80     6     56     VILL 13406M001        U


LOWER TRIANGULAR PART OF VARIANCE-COVARIANCE MATRIX:
---------------------------------------------------

RMS OF UNIT WEIGHT:  0.0011  # OBS:      25041  # UNKNOWNS:      553


  ROW  COL      MATRIX ELEMENT

    1    1      0.3347281860D+00
    2    1     -0.1217930301D-02
    2    2      0.3228655815D+00
    3    1      0.1393063499D+00
    3    2      0.2529192220D-01
    3    3      0.3318400365D+00
    4    1      0.1593018178D+00
    4    2     -0.2838543280D-02
    4    3      0.3340358226D-02
    4    4      0.3856043525D+00
    5    1      0.5451020275D-03
...
   80   78      0.1071729115D+00
   80   79     -0.4187054495D+00
   80   80      0.1860370000D+01
```

Figure 24.71.: Variance–covariance (COV) file of type 2.

This file (type 1) may, e.g., be used to combine GNSS solutions with terrestrial geodetic network solutions using different adjustment software tools. The combination of different (GNSS) solutions may be performed using either the program COMPAR (coordinates only, no transformation of the datum is possible) or preferably using the program ADDNEQ2 (all parameter types supported, based on normal equations, see Section 9). Furthermore, it is possible to extract from these files the necessary information for plotting error ellipses.

24.11.11. Normal Equation Rescaling File

Type:	ASCII
Directory:	Campaign–specific directory `OUT`.
Extension:	`WGT`
Content:	Normal equations rescaling information and management of Helmert transformation parameters.
Created by:	User–defined and ADDNEQ2 ("Menu>Processing>Combine normal equation systems").
Used by:	ADDNEQ2 to rescale normal equations for the computation of combined solutions.
Example:	Figure 24.72 and `${X}/DOC/EXAMPLE.WGT`.

Rescaling of normal equations for the combination of solutions is usually *not* necessary, if you combine your own normal equations. However, if you are combining solutions from different processing centers (using different program systems) rescaling the associated variance–covariance matrices may be mandatory. The example in Figure 24.72 shows the rescaling values for different European ACs (all using Bernese, Brockmann and Gurtner 1996). The value can either be defined by the user according to certain criteria or may be filled by the program ADDNEQ2 by a variance component estimation.

```
COVARIANCE COMPONENTS
COVARIANCE COMPONENT ESTIMATION AND HELMERT PARAMETERS
---------------------------------------------------------------------------------
NUM  VALUE                 FILENAME                           GRP HELMERT PAR.   HELMERT ...
***  *******************   ********************************   ***  * * * * * * * ********...

  1  1.0000000000000E+00                                      BEK  0 0 0 0 0 0 0   0.0000...
  2  1.0000000000000E+00                                      COE  0 0 0 0 0 0 0   0.0000...
  3  0.8000000000000E+00                                      DEO  0 0 0 0 0 0 0   0.0000...
  4  1.0000000000000E+00                                      ROB  0 0 0 0 0 0 0   0.0000...
  5  1.0000000000000E+00                                      WUT  0 0 0 0 0 0 0   0.0000...
```

Figure 24.72.: Normal equation rescaling (`WGT`) file.

The columns of the file may need a few comments:

- The mentioned rescaling values refer to the normal equations (<1 means down–weighting, 0.25, e.g., means down–weighting by a factor of 2: $1/2^2 = 1/4 = 0.25$).
- You may specify a rescaling value for each file separately by specifying `FILE NAME`s.
- You may also specify a `GRP` name (please leave the `FILE NAME` blank in this case). The `GRP` name specified here is compared to the normal equation input filenames (first 3 characters disregarding the directory path). This makes it possible to handle files in groups.
- The `NUM` column is used to indicate whether a certain component of the Helmert transformation shall be applied (value is `1`) or whether the component shall be estimated (value is `2`) when combining the normal equations (see Section 9.4.5).
- The information of the `HELMERT PAR` may be used to apply a priori Helmert transformations for coordinates in individual normal equations before stacking.

24.11.12. Observation Editing File

Type:	ASCII
Directory:	Campaign–specific directory `OUT`.

```
--------------------------------------------------------------------------------
RESIDUAL CHECK CLUSTER 03320015                                  16-NOV-09 20:02
--------------------------------------------------------------------------------

FILE INFORMATION:
----------------

FILE  SESS SESF  REFERENCE EPOCH   DTOBS TYPE STATION1         STATION2
--------------------------------------------------------------------------------

  1   3200  0    03-11-16 00:00:00   30   P   KOKB 40424M004B  PIMO 22003M001
  2   3200  0    03-11-16 00:00:00   30   P   LAE1 51002M001   PIMO 22003M001
  3   3200  0    03-11-16 00:00:00   30   P   LAE1 51002M001   TOW2 50140M001
  4   3200  0    03-11-16 00:00:00   30   P   MAS1 31303M002   ZIMJ 14001M006

--------------------------------------------------------------------------------


 RESRMS INPUT OPTIONS
--------------------------------------------------------------------------------

 OUTLIER DETECTION LEVEL (PHASE)       :    0.004  M
 OUTLIER DETECTION LEVEL (CODE)        :    0.004  M
 OUTLIER DETECTION LEVEL (RANGE)       :    0.004  M
 SAMPLING RATE OF RESIDUALS            :      180  S
 MINIMUM TIME INTERVAL FOR SMALL PIECES:      361  S
 MINIMUM NUMBER OF OBS. PER AMBIGUITY  :        0
 SAMPLING RATE FOR COUNTING OBS.       :        0  S

--------------------------------------------------------------------------------


EDITING INFORMATION:
--------------------
(EDITING TYPES: MARK=1, RESET=-1, ELIM.=2, SLIP=3, NEW AMB.=4, RESET AMB.=-4,
                SET CYC.SLIP FLAG=5, RESET CYC.SLIP FLAG=-5)
                     EPOCH NUMBERS
FILE  SAT  TYPE  FRQ  START  END    SLIP SIZE  REASON  #EPOCHS
--------------------------------------------------------------------------------

  1    14    1    3     158   168                  1       11
  1    16    1    3     686   696                  1       11
  1    11    1    3    1100  1110                  1       11
  1     2    1    3    1184  1194                  1       11
  1    31    1    3    1286  1296                  1       11
  1     7    1    3    1562  1572                  1       11
  2    31    1    3     596   606                  1       11
  2    28    1    3    1292  1302                  1       11
  2    20    1    3    1460  1470                  1       11
  2     7    1    3    1778  1788                  1       11
  2     4    1    3    1778  1788                  1       11
  3     6    1    3     386   396                  1       11
  3    27    1    3    1034  1050                  1       17
  3    31    1    3    1070  1080                  1       11
  3    27    1    3    1076  1086                  1       11
  3    31    1    3    1436  1446                  1       11
  3    11    1    3    1724  1740                  1       17
  4    28    1    3    1154  1164                  1       11

--------------------------------------------------------------------------------
```

Figure 24.73.: Observation editing (EDT) file.

Extension:	EDT
Content:	Editing requests for Bernese observation files (remove observations, add/remove marks and cycle slip flags, add/remove/reset ambiguities).
Created by:	User–defined or as output file from RESRMS ("Menu>Service>Residual files>Create residual statistics").
Used by:	SATMRK ("Menu>Service>Bernese observation files>Mark/delete observations").
Example:	Figure 24.73 and ${X}/DOC/EXAMPLE.EDT.

It is recommended to use the program RESRMS to detect outliers in the residuals of GPSEST. The outliers found by RESRMS may then be marked in the observation files

Table 24.11.: Actions (REASON) in observation editing file.

Action	Index
Mark / reset / delete observation	1 / −1 / 2
Apply cycle slip correction	3
Add / remove ambiguity	4 / −4
set / remove cycle slip flag	5 / −5

using program SATMRK (see Sections 6.6 and 6.7). Table 24.11 lists the possible actions (REASON) supported by the observation editing file.

24.11.13. Delete Files

Type: ASCII
Directory: Campaign–specific directory OUT.
Extension: DEL
Content: List of files (e.g., observation files) which should be deleted.
Created by: CODXTR ("Menu>Processing>Program output extraction>Code-based clock synchronization"), MPRXTR ("Menu>Processing>Program output extraction>Phase preprocessing"), RNXGRA ("Menu>RINEX>RINEX utilities>Create observation statistics"), RESCHK ("Menu>Service>Automated processing>Detect misbehaving stations/satellites"), and MKCLUS ("Menu>Service>Automated processing>Form clusters").
Used by: Especially well–suited for the automated processing using the BPE, but rarely used in an interactive processing mode.
Example: Figure 24.74.

The delete file contains a list of files not fulfilling specific requirements and which should be deleted. The filenames are listed with the full path (disk drive, campaign name, campaign–specific directory, e.g., OBS) and extension.

```
/aiub/EXAMPLE/OBS/OSYS2330.PSH
/aiub/EXAMPLE/OBS/OSYS2330.PSO
/aiub/EXAMPLE/OBS/IVWH2330.PSH
/aiub/EXAMPLE/OBS/IVWH2330.PSO
```

Figure 24.74.: Example for deletion file.

24.12. BPE Related Files

24.12.1. Process Control File

Type: ASCII
Directory: ${U}/PCF in the user area.
Extension: PCF
Content: Management of BPE user scripts.
Created by: User defined, with support by "Menu>BPE>Edit process control file (PCF)".
Used by: BPE.
Example: —

See Section 22.5 for more details.

24.12.2. CPU Control File

Type:	ASCII
Directory:	`${U}/PAN` in the user area.
Extension:	`CPU`
Content:	CPU management when running a BPE.
Created by:	User defined, with support by "Menu>BPE>Edit CPU file" .
Used by:	BPE.
Example:	—

See Section 22.4 for more details.

24.12.3. BPE Protocol File

Type:	ASCII
Directory:	Campaign–specific directory `BPE`.
Extension:	`PRT`
Content:	Execution protocol of BPE user script.
Created by:	BPE
Used by:	The operator to check execution of BPE.
Example:	—

See Section 22.10.2 for more details.

24.12.4. BPE Log File

Type:	ASCII
Directory:	Campaign–specific directory `BPE`.
Extension:	`LOG`
Content:	Execution log of BPE user script.
Created by:	BPE
Used by:	The operator to check execution of BPE.
Example:	—

See Section 22.10.2 for more details.

25. Installation Guide

25.1. Installation Guide for Windows Platforms

Do not install any files of Version 5.2 into existing Version 5.0 –directories. You will loose the existing version by doing this.

25.1.1. System Requirements

- Windows version:
  All current Windows versions are supported. A detailed list of successfully tested Windows versions is available on our website (`http://www.bernese.unibe.ch/support`).
- Perl 5: recent version (check your version with `perl -v` on a command line)
  It is freely available from `http://www.activestate.com`. The Perl installation requires about 80 MB of disk space.
- A modern PC with a dual– or quad–core processor allows the parallel processing of several BPE jobs. A fast access to the data on the hard disk is important for an efficient processing.
- Disk space:
  The installation itself only requires about 500 MB. Provide sufficient space for your own data later on (`GPSDATA`). Temporary scratch files in `GPSTEMP` generated during processing can reach several GB depending on the corresponding task.

25.1.2. Contents of the Distribution

The *Bernese GNSS Software* distribution media contains the following files:

`content.txt`	Content of the CD–ROM
`SETUP\BERN52.EXE`	Setup for the main software tree
`SETUP\GPSUSER52.EXE`	Setup for the user environment
`SETUP\GPSTEMP.EXE`	Setup for the temporary user environment
`SETUP\CAMPAIGN52.EXE`	Setup for the example campaign
`SETUP\DATAPOOL.EXE`	Setup for the datapool area
`SETUP\SAVEDISK.EXE`	Setup for the savedisk area
`ZIPEXE\exe_aiub_32.zip`	Executables for 32–bit systems (default)
`ZIPEXE\exe_aiub_64.zip`	Executables for 64–bit systems

`README\INSTALL.TXT`	Description of the installation procedure
`README\JPL_EPH.TXT`	Instructions to generate the JPL ephem. file
`README\FIRST_STEPS.TXT`	Remarks on the first steps with the software
`README\V50_TO_V52.TXT`	Important remarks for *Bernese GNSS Software* Version 5.0 users
`README\UPDATE.TXT`	Description of the software update procedure
`README\DOCU52.pdf`	*Bernese GNSS Software* Version 5.2 user manual
`README\TUTORIAL.pdf`	Tutorial from the Introductory Course

In the following, we assume that you install from the CD–ROM or from the directory into which you have downloaded the distribution files.

25.1.3. Installation of the Software

Privileges to Install the Software

Depending on your Windows version the installation of the *Bernese GNSS Software* setup wizards will need administrator privileges. In this case we recommend to proceed as follows:

0. Let us assume that the user *bsw* shall use the *Bernese GNSS Software* on this Windows computer.
1. The system administrator gives administrator privileges to user *bsw*.
2. All setup wizards for the *Bernese GNSS Software* are executed by user *bsw*.
3. The system administrator removes the administrator privileges for user *bsw*.

Run the setup wizards as the user *Administrator* only if you will really work with this account with the *Bernese GNSS Software* (which is of course not recommended).

Note: Only the user *bsw* can run the *Bernese GNSS Software* after this installation procedure even if on some Windows versions the icons may appear also for other users. You can either remove these icons from the desktop or establish the user environment variables from user *bsw* also to the other users.

For the installation of the *Bernese GNSS Software* on a Windows Vista, Windows 7, Windows 8, or Windows 10 operating system a User Account Control message (UAC) may appear. This is not critical and may be ignored.

Overview of the Installation Procedure

The setup wizards for Version 5.2 in the `SETUP` directory of your CD–ROM should be installed in the following order:

1. `BERN52`
2. `GPSUSER52` (also runs `GPSTEMP` installation)
   2.1 `GPSTEMP`
3. `CAMPAIGN52` (also runs `DATAPOOL` and `SAVEDISK` installation)
   3.1 `DATAPOOL`
   3.2 `SAVEDISK`

A detailed description of each step is given in the following subsections. Each of the above setup wizards can also be installed individually.

Each of these setup wizards copies the corresponding directory tree to your hard-disk and establishes environment variables for the user (HKEY_CURRENT_USER\Environment).

Even if the setup wizard proposes drive C:\ for the installation, we strongly recommend to install the software and the other components on another partition (drive) of your computer. Drive C:\ is set as default only because it is the only drive that must exist as a hard disk in all Windows systems.

Remarks for a Multi–User Environment

Typically, the *Bernese GNSS Software* is used only by one user per computer on Windows platforms. Nevertheless, a multi–user environment can be established based on network drives provided by a Windows server to the desktop PCs of the users. In that case the BERN52 and GPSDATA (with DATAPOOL, CAMPAIGN52, and SAVEDISK) should be installed on a network drive, whereas GPSUSER52 and GPSTEMP are recommended to be located on the local desktop PCs of the users. Note, that the environment variables have to be defined on all computers on which the software is to be used (take care not to generate any interferences when running the setup wizards from different PCs to the same target area; we think it is safer to copy the Bernese–related environment variables manually instead).

25.1.3.1. Installation of the Main Program Tree

Change into the SETUP–directory of your CD–ROM. Start the installation by double–clicking on BERN52.exe .

This setup wizard will allow you to select the destination for the following *Bernese GNSS Software* main program tree:

...\BERN52\BPE	Main BPE Perl modules and scripts
\GPS	General files, help, documentation, master input panels, scripts, etc.
\INC	Fortran include files and modules
\LIB	Fortran subroutines and modules
\MENU	User interface (menu program)
\PGM	Fortran main processing programs

The installation sets up new user–specific environment variables in the windows registry. These variables are:

VERSION	Bernese release version number
C	Location of BERN52, according to your choice during installation
X	Location of general files and scripts
XG	Location of the main program executables BERN52\PGM\EXE_AIUB (to be adapted if other executables shall be used)
XQ	Location of the menu executable BERN52\MENU
BPE	Location of BERN52\BPE, according to your choice during installation
I	Location of Fortran include modules BERN52\INC\FOR
LG	Location of Fortran library module routines BERN52\LIB\FOR
FG	Location of main program sources BERN52\PGM\FOR

USER	Value of the Windows USER_NAME variable (it will be displayed in the header of the program output files)
BERNESE_VARIABLES	List of variables to be exported to run BPE user scripts: "P T X U USER"
BPE_SERVER_HOST	"localhost" (name of host for BPE, i.e., this machine)
F_VERS	"AIUB" (change to your compiler name, if you want to compile the source by your own)
F_VERS_LIST	List of additional Fortran compilers
JPLEPH	Identifier of the JPL ephemerides file
OS_NAME	"WIN32" (operating system name for compilation)
QTBERN	Path to QT library (only used for compiling the menu)
CGROUP	Only for testing purposes at AIUB

These variables are written to the HKEY_CURRENT_USER\Environment registry key and are, therefore, valid for the installing user, only. If you wish to change any of these variables after the installation (particularly the X variable), go to Windows "Control Panel>System>Advanced >Environment Variables" and adapt the settings in the "User variables" part.

After a successful installation, a new "Bernese 5.2" icon should appear on your desktop.

Remark for 64–bit systems:
The setup wizard has installed 32–bit executables on your system so far. executables for a 64–bit system are distributed as a zip–archive exe_aiub_64.zip on the CD–ROM in the directory ZIPEXE. You can copy these files into the %XG% (BERN52\PGM\EXE_AIUB)–directory and replace the 32–bit executables.

25.1.3.2. Installation of the GPSUSER52 Tree

Change into the SETUP–directory of your CD–ROM. Start the installation by double–clicking on GPSUSER52.EXE.

This setup wizard will allow you to select the destination directory for the user–specific configuration of the *Bernese GNSS Software*, the GPSUSER52 area. It is not necessary to install GPSUSER52 as a subdirectory of BERN52. It will contain all files that are needed to run the example BPE.

The installation defines the user–specific environment variable %U%in the windows registry for the location of the user–specific directory GPSUSER52.

After successful installation, a new "GPSUSER52" icon should appear on your desktop.

25.1.3.3. Installation of the GPSTEMP Tree

By default the setup wizard for installing the temporary user environment of the *Bernese GNSS Software* is started automatically by the GPSUSER52.EXE wizard. But it can be launched manually as well by double–clicking on GPSTEMP.EXE in the SETUP–directory of the CD–ROM.

The setup wizard asks you to select the location of the GPSTEMP directory. The installation defines the user–specific environment variable %T% for the location of the temporary directory.

25.1.3.4. Installation of the CAMPAIGN52 Tree

Change into the SETUP–directory of your CD–ROM. Start the installation by double–clicking on CAMPAIGN52.EXE.

This wizard will allow you to select the destination directory for the Bernese GPSDATA areas:

CAMPAIGN52	Processing area of the GNSS data
DATAPOOL	Location of all external input files
SAVEDISK	Archive of the result files

This setup wizard is designed to establish the CAMPAIGN52 directory that initially contains the campaign ${P}/EXAMPLE. (Please note the UNIX–syntax that is used for all variables in the Bernese environment also on Windows platforms.)

The installation defines the user–specific environment variable %P% for the processing campaign location.

After successful installation, a new "CAMPAIGN52" icon should appear on your desktop. Over time, this area can take up a lot of disk space. Make sure to install it on a suitable disk/partition. In particular the GPSDATA area should not be located on the C:\ partition.

If you do not want to install the data for the example campaign for any reason you can create the directory CAMPAIGN52 manually and set the environment variable %P% to the location of CAMPAIGN52 instead.

25.1.3.5. Installation of the DATAPOOL Tree

By default the DATAPOOL tree will be installed at the end of the CAMPAIGN52.EXE wizard. But it can be installed manually as well by double–clicking on DATAPOOL.EXE available in the SETUP–directory of your CD–ROM.

The installation defines the user–specific environment variable %D% for the location of the DATAPOOL area. After the installation, it contains all files that are needed to run through the example BPE distributed together with the software.

After successful installation, a new "DATAPOOL" icon should appear on your desktop.

The content of the DATAPOOL tree is described in a separate README file. If you open the folder by double–click on the "DATAPOOL" icon on your desktop you can find the README_DATAPOOL.TXT file.

25.1.3.6. Installation of the SAVEDISK Tree

By default the SAVEDISK tree will be installed at the end of the CAMPAIGN52.EXE wizard. But it can be installed manually as well by double–clicking on SAVEDISK.EXE available in the SETUP–directory of your CD–ROM.

The installation defines the user–specific environment variable %S% for the location of the SAVEDISK area. After the installation it contains the files from the reference solution of the example BPEs.

After successful installation, a new “SAVEDISK” icon should appear on your desktop.

The content of the SAVEDISK tree is described in a separate README file. If you open the folder by double–click on the “SAVEDISK” icon on your desktop you can find the README_SAVEDISK.TXT file.

25.1.3.7. Finishing the Installation

At this point, in order to complete the installation of the *Bernese GNSS Software*, a reboot of your system is required to make the newly defined environment variables available. Afterwards, the *Bernese GNSS Software* will be ready to run.

25.1.4. Additional Remarks

25.1.4.1. Installing the Ephemeris from JPL

The planetary and lunar ephemeris from JPL are used to compute the gravitational attraction of Sun, Moon and planets for satellite orbit determination. You find the installation description in the README/JPL_EPH.TXT file on the CD–ROM or at %X%\DOC\README_JPL_EPH.TXT on your computer.

Please be aware that you cannot use the DE200.EPH file (in the %X%\GEN directory) from a previous version of the *Bernese GNSS Software* because the DE number changed to 405.

25.1.4.2. Additional Software Tools

The following tools are needed to manage RINEX data from IGS related sources. Note, that they are also used to run the example BPEs:

- file compression/decompression tool that can be used from the command window, e.g., gzip from www.gzip.org
- the Hatanaka RINEX file compression/decompression utilities
  from http://terras.gsi.go.jp/ja/crx2rnx.html

Note, that these programs must be included in the %PATH% variable to become executable without the full path.

25.1.4.3. Add a new Campaign Directory

To add another campaign directory you have to define an additional variable (e.g., %Q%) in the Windows registry (Windows "Control Panel>System>Advanced>Environment Variables" part "User variables"). Furthermore, it has to be added to the list of variables in BERNESE_VARIABLES registry entry.

In addition, each user has to add this variable in his list of “Environment variables” in the first panel of "Menu>Configure>Menu variables". To process data from this new campaign directory with a BPE you have to add this variable to the MENU_VAR.INP files in all %U%\OPT directories.

25.1.4.4. Update your Installation of the Software

Online updates are downloaded and installed until no further update is available. If this cannot be done automatically, the corresponding information for a manual update is prompted to the screen.

Please read the Section 25.3 for further instructions.

25.1.5. Compilation of Modules and Programs

The installation files contain the fully compiled source code in the `%C%\PGM\EXE_AIUB` directory. Therefore, for Windows, it is not necessary to have a Fortran compiler installed on your machine. However, if you want to adapt some declarations, or to change the Fortran source code for any other reason, you will need to recompile the source code.

We have used the Portland Fortran 95 Compiler pgf90 version 12.10–0 to compile the Fortran part of the *Bernese GNSS Software* Version 5.2 , on a 64–bit Windows 7 system.

Preparation of Compilation

For the compilation of the Fortran source you need:

- a Fortran compiler.
  Use either the Portland compiler or the G95 (`http://www.g95.org`) compiler. Other compilers have not been tested by AIUB but you may inspect `http://www.bernese.unibe.ch/support/cmp_all_list.php` whether users have reported the successful installation with other compilers. More information about the use of other compilers is given in "Unknown Fortran Compiler" later in this section.
- the "make"–tool (e.g., from `http://gnuwin32.sourceforge.net/packages/make.htm`)

Note, that these programs must be included in the `%PATH%` variable to become executable without the full path. Depending on your Fortran compiler possibly some more environment variables are necessary.

Generating the Makefile

Starting with Version 5.2 the compilation of the Fortran processing programs is based on "Makefile". The Makefile(s) are generated by the tool

```
perl %X%\EXE\makemake.pl -r %C%
```

to be started from the command line.

The resulting Makefile is located in the directory `%C%\LIB\OBJ_xxx` where `xxx` stands for the compiler name from variable `%F_VERS%` (e.g., "`G95`"). You can go into this directory and directly execute the "make"–command.

Compilation of Individual Modules and Programs

For your convenience we have also provided a script to compile the source code (it executes the make command internally with the correct options):

1. `perl %X%\EXE\cbern.pl LIB`
   This command compiles all modified Fortran modules and creates/updates the Bernese core library. Note, that the programs are not compiled!
2. `perl %X%\EXE\cbern.pl ALL`
   This command also compiles the modified Fortran programs and links all programs to the Bernese core library.
3. `perl %X%\EXE\cbern.pl COMPLINK`
   This command re–compiles all Fortran source files, creates the Bernese core library and links all programs.
4. `perl %X%\EXE\cbern.pl  <SRNAME>`
   This command compiles only the component `SRNAME`. If other components needed by `SRNAME` have also been changed, they are compiled as well.

Note, the compilation process is locked by the existence of the file `%C%\BERN.lock`. If you have interrupted the compilation process for any reasons you have also to delete this file before you can start a new attempt to compile any routine.

Using of the Configure Tool as Alternative

The configuration tool `configure.pm` allows you to recompile the full source code in one step. You may start it on the command line by typing

```
perl %X%\EXE\configure.pm
```

The following menu is displayed

```

==========================================
CONFIGURATION OF THE BERNESE GNSS SOFTWARE
==========================================
 1 ...   ---
 2 ... Install online updates
 3 ...   ---
 4 ...   ---
 5 ... Compile the programs
 6 ...   ---
 7 ... Remove BSW keys from environment

 x ... Exit

Enter option:
```

With step 5 you can recompile all fortran programs. Please read Section 25.3 for further instructions.

Activating the Recompiled Executables

Your compiled executables will be stored in the directory `%C%\PGM\EXE_xxx` where `xxx` stands again for the compiler name from variable `%F_VERS%` (e.g., “`G95`”). The environment variable pointing to the executables from the Fortran programs is `%XG%`. In the default installation it points to the directory `%C%\PGM\EXE_AIUB` where the executables from the installation CD–ROM are located. To use your own set of executables you must redefine the variable %XG% in the registry and enable the new value by rebooting your system.

Remember, the processing will always be done taking the executables from the path in ${XG} or the default program path set in the panel of "Menu>Configure>Program names" .

Unknown Fortran Compiler

A list of tested Fortran compilers can be found at http://www.bernese.unibe.ch/support/cmp_all_list.php (note the restrictions regarding the Lahey compiler).

If you wish to implement a new Fortran compiler you will need to adapt the file %X%/EXE/Makefile.template where the compilers and their options are defined. You can use the settings of the compiler OTHER as a template. You should have your compiler manual ready to specify the individual options.

Procedure for unknown compiler:

- When you are asked by the configuration tool to give the compiler name, select OTHER.
- Adapt the file Makefile.template in the main software tree BERN52\GPS\EXE for your compiler by filling out the options for the compiler OTHER (part ifeq ($(F_VERS),OTHER)).
- After the compilation, the path %XG% to the program executables will end with EXE_OTHER.

25.1.6. Uninstall the Bernese GNSS Software

There is no wizard to uninstall the *Bernese GNSS Software*. If you want to uninstall the software from your system, you first have to remove the environment variables from the registry. For this purpose, type

```
perl %X%\EXE\configure.pm
```

in a command window. Select step 7 ... Remove BSW keys from environment from the menu (displayed on page 782).

After that you can remove all the individual Bernese–related directory trees:

%U%	user environment
%T%	temporary user environment

Note: if more than one user has been installed, %U% and %T% have to be removed for all users.

%P%	campaign area
%D%	DATAPOOL area
%S%	SAVEDISK area
%C%	main software tree

Take care that you have archived all relevant files before you delete the directory trees.

Remove the Bernese desktop shortcuts and the Bernese start menu from the "Start Menu>All Programs" list.

25.2. Installation Guide for UNIX/Linux/Mac Platforms

Do not install any files of Version 5.2 into existing Version 5.0–directories. You will loose the existing version by doing this.

25.2.1. System Requirements

- Perl 5: recent version (check your version with `perl -v` on a command line) including the standard modules.
- Basic Unix tools:
  - tar (tape archiver)
  - gzip (file compression/decompression tool)
  - make
- Fortran 90 compiler: a list of Fortran compilers we have used to test the software can be found at `http://www.bernese.unibe.ch/support/cmp_all_list.php`.
- C++ compiler (e.g., gcc–c++)
- Qt 4 library: it is expected that the variable `${QTDIR}` is set before the installation is started to indicate the location of the Qt libraries to be used to compile the Bernese menu (e.g., `QTDIR="/usr/lib/qt"`). If you use another version of the Qt libraries for other applications, the variable `${QTDIR}` has probably a different value. Redefine the variable at the command line. This value will then be saved in the Bernese environment variable `${QTBERN}` and the Bernese menu will find the correct version of the Qt libraries after the installation in any case, even if `${QTDIR}` points again to another version of the Qt libraries.

  Remarks on installing the Qt libraries:

  - In most cases it is not enough to install the Qt library package from your Linux distribution.
  - On some Linux distributions not all libraries or C++ header files needed to establish the Qt library from the source code were installed with the default configuration. In all cases this could be fixed by installing additional software package from the Linux distribution (e.g., libXext–devel, zlib–devel).
  - A sequence of commands typically needed to install the Qt libraries is included in the `README/INSTALL.TXT` on your installation medium.
  - For Mac users only: After the installation of the Qt libraries, copy the directory `$SRCDIR/qt-everywhere-opensource-src-$VERS/src/gui/mac/qt_menu.nib` to `$QTDIR`:

    ```
     > cp -r $SCRDIR/qt-everywhere-opensource-src-$VERS/src/gui/mac/qt_menu.nib $QTDIR
    ```

25.2.2. Contents of the Distribution

The *Bernese GNSS Software* distribution media contains the following files:

`content.txt`	Content of the CD–ROM
`setup.sh`	Installation script
`BERN52.tgz`	Main software tree
`CAMPAIGN52.tgz`	Example campaign
`DATAPOOL.tgz`	Datapool area
`SAVEDISK.tgz`	Savedisk area
`ICONS.tgz`	Icons for X–Window System
`README/INSTALL.TXT`	Description of the installation procedure
`README/JPL_EPH.TXT`	Instructions to generate the JPL ephem. file
`README/FIRST_STEPS.TXT`	Remarks on the first steps with the software
`README/V50_TO_V52.TXT`	Important remarks for *Bernese GNSS Software* Version 5.0 users
`README/UPDATE.TXT`	Description of the software update procedure
`README/DOCU52.pdf`	*Bernese GNSS Software* Version 5.2 user manual
`README/TUTORIAL.pdf`	Tutorial from the Introductory Course

In the following, we assume that you install from the CD–ROM or from the directory into which you have downloaded the distribution files.

25.2.3. Installation of the Software

25.2.3.1. Questionnaire to Prepare the Installation

The default proposal from the installation/configuration tool is intended for single–user and single–computer environments (all is installed in the user's home directory). If you want to establish a multi–user or multi–computer environment you should start with establishing a concept for the installation. Here are some useful comments for this step:

- The directory for the source code is proposed with `${HOME}/BERN52`. It must be accessible for reading and execution by all users who will work with the software and from all hosts where the BPE will run. So it might either be a home directory of a specific *Bernese GNSS Software* administrator user or you relocate it on a central location in your system (e.g., `/usr/local/BERN52`).

  The campaign list `MENU_CMP.INP` is a file common to all users. It sits in the software directory `${X}/PAN` (e.g., `/usr/local/BERN52/GPS/PAN`). In order to edit this file for adding a new campaign to the list, the users need also write permission for this file.

  The disk space requirement for the complete installation is at least 500 MB.

- The directory for the user files is specific to each user of the software. It must be accessible from all hosts where the BPE will run. The best location of this directory is the home directory of each user (e.g., `${HOME}/GPSUSER52`).

- The directory for temporary files is necessary to run the BPE and is user–specific. This directory needs to exist on all hosts accessed by the BPE (but it can be on physically different disks). A fast disk with enough space is preferable. If you have a local scratch disk this directory should be located there, otherwise the home directory is a good choice (e.g., `/scratch/local/${USER}/GPSTEMP` or `${HOME}/GPSTEMP`). (Each script running simultaneously in the BPE may produce a scratch file!)
- Finally, a data area is needed. It is divided into three parts:

  `CAMPAIGN52`: Processing area of the GNSS data
  `DATAPOOL`: Location of all external input files
  `SAVEDISK`: Archive of the result files

  These directories must be accessible from all hosts where the BPE will run. You should consider to select a disk area with a big amount of storage (e.g., `/bigdisk/CAMPAIGN52`).

25.2.3.2. Running the Installation Procedure

Extracting the Files

Now you can start the installation. Copy all installation files from the CD–ROM or ftp them to an installation directory (e.g., `${HOME}/INST52`).

Remark to users with older versions of the *Bernese GNSS Software*:
To avoid potential interferences with default settings we recommend that you have NOT loaded the Bernese variables from an older version when you install Version 5.2 .

Run the installation script `setup.sh` to extract the source code and related files to your system:

```
> cd ${HOME}/INST52
> sh ./setup.sh
```

A line like

```
> Full path where the BERN52 software tree will be installed
  [ PATH_TO_YOUR_HOME ]:
```

appears and you are asked for the directory where the software should be installed. Either you accept the proposed default directory (press “ENTER”) or write your choice to the command line. You have to confirm your selection or the default directory.

After the successful extraction of the archive file you will be asked for the path to the Perl interpreter:

```
> Perl program to be used [ /usr/bin/perl ]:
```

The default path is the location your system returns when you type `which perl` . Normally you can confirm the proposed path except if you have different Perl installations on your system and the proposed path is not the correct one. The path to the Perl interpreter will be adapted in all scripts of the software which use Perl.

Configuration and Compilation of the Software

The installation script `setup.sh` automatically starts the configuration tool `configure.pm`. After the installation, this tool can be called manually to add, e.g., the user environment for another user or to install the Example campaign. It will be located at `${X}/EXE`.

The main menu of the configuration tool looks like:

```
==========================================
CONFIGURATION OF THE BERNESE GNSS SOFTWARE
==========================================
 0 ... Complete installation (Steps 1 to 5)
 1 ... Update LOADGPS.setvar
 2 ... Install online updates
 3 ... Add a new user environment
 4 ... Compile the menu
 5 ... Compile the programs
 6 ... Install the Example campaign
 7 ...   ---

 x ... Exit

Enter option:
```

We recommend to select option “0” in order to make a full installation. Nevertheless, we will explain the installation according to the particular steps “1” to “5”. The only difference is that for selection “0” the installation runs automatically through steps “1” to “5”.

Configure Menu: `1 ...  Update LOADGPS.setvar`

First you get a list of supported/tested Fortran compilers where you have to select the one which is installed on your system:

```
Select Compiler for OS LINUX:
 1: G95
 2: GNU -> /usr/bin/gfortran
 3: IFC
 4: PG_F90
 5: SUNF90
 6: OTHER

Select  [2]:
```

In the example above, the GNU–Fortran compiler has been found on your system (it is indicated by the path to the compiler). Confirm the proposal from the `configure.pm` tool or choose another compiler. If another compiler is chosen it possibly still needs to be installed before you are going to compile the Fortran programs in menu option “5” of the configuration tool. If your compiler is not listed, read Section 25.2.5.2: “Unknown Fortran Compiler”.

A list with your selections and default settings for the campaign directories (in `${HOME}/GPSDATA`) and the user–specific directories (`${HOME}/GPSUSER52` and `${HOME}/GPSTEMP`) is given:

```
Current Values:
--------------
    VARIABLE DESCRIPTION           VARIABLE NAME      VARIABLE VALUE
 1: Path to the software           C               => /home/user/BERN52
 2: Path to Qt lib for Bernese     QTBERN          => /usr/local/qt_4.7.4
 3: Operating system group         OS              => UNIX
 4: Name of the operating system   OS_NAME         => LINUX
 5: Fortran compiler name          F_VERS          => G95
 6: List of additional compilers   F_VERS_LIST     =>
 7: Host of the BPE server         BPE_SERVER_HOST => myComputer
 8: Path to user environment       U                => ${HOME}/GPSUSER52
 9: Path to temp. user environment T                => ${HOME}/GPSTEMP
10: Path to campaign directory     P                => ${HOME}/GPSDATA/CAMPAIGN52
11: Path to datapool area          D                => ${HOME}/GPSDATA/DATAPOOL
12: Path to savedisk area          S                => ${HOME}/GPSDATA/SAVEDISK

Accept the values (y/n):
```

The screen shows the list of variables with their values and a description. If the Bernese environment has been loaded before the configure tool was started (without the argument `--init`), the corresponding values from the current file `${X}/EXE/LOADGPS.setvar` are displayed here.

You can accept the values as they are or you may modify the settings after typing "n" (not accept) with entering the item number followed by the new value, e.g.,

```
  > 10 /bigdisk/GPSDATA/CAMPAIGN52
```

If you redefine the variables for the user–specific directories for any reasons, make sure that they are really user dependent by using the system variables ${HOME} or ${USER} (e.g., `/scratch/local/${USER}`) and not the full path (e.g. `/scratch/local/user`).

The `${X}/EXE/LOADGPS.setvar` file is generated automatically using these entries. The syntax for bourne–shell or c–shell is used according to the value of the ${SHELL} system variable.

Additional remarks:

- If you want to change `1: Path to the software` you have to start the installation again with `setup.sh`.
- The variable in `7: Host of the BPE server` must contain the name of the host where the BPE server will run. Usually, it is detected automatically. If you are not permanently connected to the Internet and the hostname changes each time for new connections, you have to adapt the value in a reasonable manner (e.g., using `$(hostname)` in bash notation or `‘hostname‘` in tcsh notation). If you use the BPE locally only, you can use `localhost` in any case.
- Any changes to the `${X}/EXE/LOADGPS.setvar` file you make outside the configuration tool, e.g., adding a second campaign directory, will be lost if you update the Bernese environment file again with this tool.

Configure Menu: `2 ... Install online updates`

Online updates are downloaded and installed until no further update is available. If this cannot be done automatically, the corresponding information for a manual update is prompted to the screen.

Please read Section 25.3 for further instructions.

Configure Menu: `3 ... Add a new user environment`

The user area is added. At first, the directory structure for the Bernese user environment is created at the locations defined by the variables ${U} and ${T}. They were previously set when creating the ${X}/EXE/LOADGPS.setvar file during the installation:

${U}/PAN Program–specific input option files (initially a copy of ${X}/PAN)
/OUT Directory for campaign–independent output files
/OPT Your set of option directories (initially a copy of ${X}/OPT containing the option directories for the Example campaign)
/PCF Directory of user–specific Process Control Files (PCFs) for the BPE (initially a copy of ${X}/PCF for the example BPEs)
/SCRIPT Contains the user–specific scripts necessary for the PCFs (initially a copy of ${X}/USERSCPT for the example BPEs)
/WORK User–specific files generated during interactive work
${T} directory for temporary files used by the BPE

If you need to create a user environment for a new user after the installation, the Bernese environment must be loaded by this new user before starting the configuration tool. The variables ${U} and ${T} should always point to the user–specific directory of the current user.

```
> source /home/user/BERN52/GPS/EXE/LOADGPS.setvar
> echo ${U}
/home/new_user/GPSUSER52
> ${X}/EXE/configure.pm
```

Adapt the commands for this operation depending on the location of the software and your shell.

Configure Menu: 4 ... Compile the menu

The menu will be compiled. The compilation can take a while. After this step, the executable of the menu will be located in ${XQ}.

If the compilation failed inspect the output log file. If you cannot find the reason for the failure you may find an answer on the Frequently Asked Questions website http://www.bernese.unibe.ch/faq or you can contact the support at bernese@aiub.unibe.ch. Add the output log file, the ${X}/EXE/LOADGPS.setvar file and all relevant information about your system to the e–mail.

Configure Menu: 5 ... Compile the programs

The Fortran source will be compiled. The compilation can take a while — depending on the compiler and the performance of your system.

After this step, the executables of the *Bernese GNSS Software* programs are located in ${XG}, where the directory name ends with EXE_${F_VERS} (e.g., EXE_G95 if g95 has been used as Fortran compiler). A second set of executables is available in ${XG}c. These executables are compiled with the debugging options defined in ${X}/EXE/Makefile.template.

If the compilation failed inspect the output log file. If you cannot find the reason for the failure you may find an answer on the Frequently Asked Questions website http://www.bernese.unibe.ch/faq or you can contact the support at bernese@aiub.unibe.ch. Add the output log file, the ${X}/EXE/LOADGPS.setvar file and all relevant information about your system to the e–mail.

Configure Menu: 6 ... Install the Example campaign

You have to execute this step also if you have selected “0” for a complete installation.

The Example campaign will be installed on your system. The location of the campaign is given by the variables ${P}, ${D} and ${S} which were previously set when creating the ${X}/EXE/LOADGPS.setvar file:

${P}/EXAMPLE/ATM	empty directory
/BPE	empty directory
/GRD	empty directory
/MSC	empty directory
/OBS	empty directory
/ORB	empty directory
/ORX	empty directory
/OUT	empty directory
/RAW	empty directory
/SOL	empty directory
/STA	campaign’s session table (SESSIONS.SES)
${D}	see ${D}/DATAPOOL.README
${S}	see ${S}/SAVEDISK.README

If you extract the Example campaign after the installation, the Bernese environment must be loaded before starting the configuration tool.

Note: to run the example BPEs, you also need to have installed on your system:

- the Hatanaka RINEX file compression/decompression utilities
  from http://terras.gsi.go.jp/ja/crx2rnx.html
- the planetary and lunar ephemeris from JPL
  (see Section 25.2.4.1)

Configure Menu: 7 ... --

This is a dummy menu item that is only relevant for windows platforms. It is kept to keep the configure.pm tool consistent for both platforms.

Configure Menu: x ... Exit

The installation is now finished and you can exit the configuration tool.

25.2.4. Additional Remarks

25.2.4.1. Installing the Ephemeris from JPL

The planetary and lunar ephemeris from JPL are used to compute the gravitational attraction of Sun, Moon and planets for satellite orbit determination.

You find the installation description in the README/JPL_EPH.TXT file on the installation media or ${X}/DOC/README_JPL_EPH.TXT.

Please be aware that you cannot use the DE200.EPH file from a previous version of the *Bernese GNSS Software* (${X}/GEN directory) because the DE number changed to 405.

25.2.4.2. Include the Software in X–Window Desktop Environment

After the installation of the software you can find a directory ICONS in ${X}/DOC. With the files in this directory you have the possibility to add an icon for the *Bernese GNSS Software* on your desktop.

Please consult the file ${X}/DOC/ICONS/README.icons for further instructions.

25.2.4.3. Add a New Campaign Directory

To add another campaign directory you have to define an additional variable (e.g., ${Q}) in the ${X}/EXE/LOADGPS.setvar file (do not forget to source the file to activate the new variable).

In addition each user has to add this variable in his list of "Environment variables" in the first panel of "Menu>Configure>Menu variables". To process data from this new campaign directory with a BPE you have to add this variable to the MENU_VAR.INP files in all ${U}/OPT directories.

25.2.5. Compilation of Individual Modules and Programs

The compilation of the Fortran processing programs is based on "Makefile" since Version 5.2. The Makefile(s) are generated by the tool

```
makemake.pl -r $C
```

The Makefile is located in the directory ${C}/LIB/OBJ_xxx where xxx stands for the compiler name from variable ${F_VERS} (e.g., "G95"). You can go into this directory and directly execute the make–command.

For your convenience we have also established a script to compile the source code (it executes the make command internally with the correct options):

1. CBERN LIB
   This command compiles all modified Fortran modules and creates/updates the Bernese core library. Note, that the programs are not compiled!
2. CBERN ALL
   This command also compiles the modified Fortran programs and links all programs to the Bernese core library.
3. CBERN COMPLINK
   This command re–compiles all Fortran source files, creates the Bernese core library and links all programs. It is the same as step 5 ... Compile the programs of the configuration tool configure.pm.

4. `CBERN <SRNAME>`
   This command compiles only the component `SRNAME`. If other components that are needed by `SRNAME` have been also changed they are also compiled.

Note, that the compilation process is locked by the existence of the file `${C}/BERN.lock`. If you have interrupted the compilation process for any reasons you have also to delete this file before you can start a new attempt to compile any routine.

25.2.5.1. Changing F_VERS and Multi–Compilers Support

The environment variable F_VERS is the identifier for the main Fortran compiler used in `${X}/EXE/Makefile.template` for the compilation, as well as a part of the path to the executables of the *Bernese GNSS Software* (variable `${XG}` which is `${C}/PGM/EXE_${F_VERS}`).

To change the default Fortran compiler, the value for F_VERS has to be changed either with the configure utility `configure.pm`, step "1" (do not accept the values displayed), or by editing `${X}/EXE/LOADGPS.setvar` manually. After loading the new environment the compilation will be done by the new compiler and the *Bernese GNSS Software* will consider the executables from the new directory `${XG}`.

If you have more than one Fortran compiler available (e.g., for comparison purposes or for software development), it is possible to add additional compilers to the list F_VERS_LIST in the Bernese environment. In this way, the compiler scripts `makemake.pl` and `CBERN` will consider all compilers. This is useful for consistency and reliability checks of the Fortran source code.

The processing will always be done by taking the executables from the path in `${XG}` or the default program path set in the panel of "Menu>Configure>Program names".

25.2.5.2. Unknown Fortran Compiler

A list of tested Fortran compilers can be found at `http://www.bernese.unibe.ch/support/cmp_all_list.php`.

If you wish to implement a new Fortran compiler, you will need to adapt the file `${X}/EXE/Makefile.template` where the compilers and their options are defined. You can use the settings of compiler `OTHER` as a template. You should have your compiler manual ready to specify the individual options.

Procedure for unknown compiler:

- When you are asked by the configuration tool to give the compiler name select `OTHER`.
- Adapt the file `Makefile.template` in the main software tree `BERN52/GPS/EXE` for your compiler by filling out the options for the compiler `OTHER` (part `ifeq ($(F_VERS),OTHER)`).
- After the compilation the path `${XG}` to the program executables will end with `EXE_OTHER`.

25.2.6. Uninstall the Bernese GNSS Software

There is no tool to uninstall the *Bernese GNSS Software*. If you want to uninstall the software on your system you only have to remove the following directories:

`$U`	user environment
`$T`	temporary user environment

Note: if more than one user has been installed, `${U}` and `${T}` have to be removed for all users.

`$P`	campaign area
`$D`	`DATAPOOL` area
`$S`	`SAVEDISK` area
`$C`	main software tree including `LOADGPS.setvar`

Take care that you have archived all relevant files before you delete the directory trees.

25.3. Updating Your Installation of the Software

General Remarks

- Keep your source code, executables, and supporting files consistent at all times. Remember that program input files are also located in the user environment in the home directory of *each* individual user of your installation.
- We strongly recommend to update your version as soon as a new release has been published, or a new bug–fix, correction, or improvement is announced on the website.
- Please be careful if you made some modifications in the software! If this is the case, you should extract the update tar–file into a temporary directory different to the default directory and merge your modifications with the updated files.
- A release contains all individually listed bug–fixes, corrections, and improvements.
- For users of Version 5.0: The update package for your release contains now all changes and brings you directly to the latest release. In contrast to Version 5.0 you need to install only one update package to get to the latest release level.

25.3.1. Update Procedure with the Update Utility

To update your release version, please proceed as follows:

- Start the Perl script `configure.pm`. On UNIX/Linux platforms it is found in `${X}/EXE`. For Windows platforms it is started from a command line with the command:

  ```
          perl %X%\EXE\configure.pm
  ```

  Select the option `2) Install online updates`. The utility will check your installation for completeness and will find your current release version.

- If the program `wget` is installed on your operating system (which is normally the case for UNIX/Linux), the update utility looks automatically for an online update for your release version. The corresponding archive will be downloaded and you will be asked if the files should really be extracted. If yes, the update is done automatically.

  If `wget` is not available, you will be asked to download the appropriate update archive together with access information to the website where these files are located. Then you can download the update file and save it in the Bernese top directory `${C}` (UNIX/Linux) or `%C%` (Windows), respectively.

  Windows:
  The update utility supports several tools for the extraction of the ZIP–archives from the command line. If one of these is available on your system, the utility will extract the files to a directory of your choice (default: the Bernese top directory). If none of the supported tools is installed on your computer, the update utility will ask you to extract the files manually by using winzip or any other unzip utility you have available.

- The extraction of the archives updates only the files in the main tree of the *Bernese GNSS Software* (`${C}` or `%C%`),. If a program input panel has changed, users have to update the input files in their user environment. If this is the case, the utility starts the *Bernese GNSS Software* menu with the panel of the program UPDPAN. Select the directories (`${U}/PAN`, `${U}/OPT/*`, or a list of directories from an `UPD`–file) and run the program. After updating all program input panels in the user directories you may quit the Bernese menu. For more information on the UPDPAN program, please refer to the online help.

The update of the source and supporting files is now complete. All users on UNIX/Linux platforms must recompile the source code in order to make the update active. The same applies to users on Windows platforms using their own compiler (and not the distributed binaries, see below). This is achieved with the menu item `5) Compile the programs` in the `configure.pm` script.

Windows users without a compiler can download the executables from the most recent release of the bsw together with the update archive and extract it into `%XG%`–directory (default: `%C%\PGM\EXE_AIUB`).

25.3.2. Updating for Individual Bugs/Corrections/Improvements

We will publish from time to time bug–fixes, corrections, or improvements on the website:

`http://www.bernese.unibe.ch/support`

and select your release. There you find instructions on how to download the affected files, where to place them in your environment, and how to recompile the source code.

Note: these updates are cumulative. You have to consider all of them. It may damage your installation if you try to establish only selected bug–fixes, corrections, or improvements.

The bug–fixes, corrections, or improvements are listed for different releases of the *Bernese GNSS Software*. The release date of your installation can easily be checked in the menu "Menu>Help>About" .

25.4. Programming Standards and Conventions

25.4.1. General Remarks

All programs and subroutines are written in Fortran 77 or Fortran 90. Fortran 90 is the standard programming language for the *Bernese GNSS Software* Version Version 5.2 and all newly developed programs are written in Fortran 90. Users who would like to make changes in the programs need to consider the following aspects:

- For obvious reasons, the AIUB *cannot provide the full support anymore if you have modified your source code.*
- All variables need to be explicitly defined.
  All modules contain the IMPLICIT NONE statement.
- The naming convention for modules located in the directory `${I}` (or `%I%` for Windows) is as follows:
  - Basic *standard modules* are:
    `M_BERN.f90` defines basic parameters (constants) and variables with logical file numbers (`LFNxxx`)
    `M_GLOBAL.f90` provides variables and parameters to characterize GNSS and observation types
    `M_MAXDIM` defines several maximum dimension parameters
    `M_TIME.f90` defines structures, operators and functions for handling time intervals
    `M_EPOCH.f90` defines structures, operators and functions for handling of precise epochs
    `M_CPU.f90` provides routines to evaluate the processing time
  - *Data type–related modules* start with `D_` . They provide at least the structures needed to map the content of the file and a routine to initialize the structure. In some cases they also contain methods to read and convert data structures between different versions of a data file.
  - *Program–related modules* start with the identifier `P_` . They provide parameters and structures related to specific programs. Routines to initialize the structures or simple access routines may be contained as well.
- All subroutines and functions are declared as MODULE .
  The related modules start with `s_` or `f_` respectively. They are located in the library directory `${LG}` (`%LG%` for Windows). In most cases there are separate files for each subroutine and function. Exceptions are modules containing private structures that need specific access routines.
  In each place where the subroutine or function is called it must be declared by a USE–statement.
- The *Bernese GNSS Software* contains numerous routines for standard tasks. The most important ones are listed in Table 25.1 .
  Please look into the header of the related files to learn how they should be used. You may also search for examples within the software on the usage of these routines.

Table 25.1.: Routines for standard tasks (selection of the most important ones).

Filename	Purpose
General subroutines:	
RPLENVAR.f90	to replace environment variables according to the content of the keyword ENVIRONMENT (description in Section 21.6.3)
INQUIRE.f90	an overlay to the instinct INQUIRE routine considering the replacement of environment variables using RPLENVAR
MYGETARG.f90	to return the command line or reads the standard input (maps different functions provided by different compilers into a standard structure)
DEFCON.f	to define and read constants from constant file (see Section 24.3.1)
MAXTST.f	to check local maximum dimensions
Opening and reading files:	
FPARSE.f	to parse a file specification to get directory, filename, and extension
OPNFIL.f	to open file with a sophisticated error handling
OPNERR.f	to evaluate and report the error codes from subroutine OPNFIL
OPNSYS.f90	to open the general program files SYSINP, SYSOUT (Program output), and SYSERR (for error and warning messages) and assigns the logical file numbers lfnkbd, lfnprt, and lfnerr .
LINCOUNT.f90	to count lines of a formatted file name
NEXTLINE.f90	to read one line from an opened file
FINDLN.f90	to read through an opened file until a specific line is found
Accessing options from input file:	
READ1KEY.f90	to read one key from a given input file
READINPF.f90	to read the content of an input panel into a buffer
READKEYS.f90	to return the value from a given keyword
CKOPTB.f90	to extract the setting of a checkbox or a series of radiobuttons
CKOPTC.f90	to extract the setting of a combobox
CKOPTI.f90	to evaluate the input option as an integer value
CKOPTR.f90	to evaluate the input option as a real value
CKOPTL.f90	to evaluate the input option for a character value (checking for the length)
CKOPTD.f90	to evaluate the input option as a date value
CKOPTT.f90	to evaluate the input option as a time value
CKOPTZ.f90	to evaluate the input option as a combined date/time value
CKOPTU.f90	to extract substrings from uniline record
CKOPTN.f90	to check the size of keyValue with respect to a maximum
GTFLNA.f90	to return a filename from a given keyword with a dedicated error handling
GTFILE.f90	to get file names assuming a list of filenames using a given keyword, returning a vectorized array
GTFILE2.f90	to get file names assuming a list of filenames using a given keyword
GTFILE3.f90	same as GTFILE2 but with a longer string for the filenames
TSTKEY.f90	to check the list of keywords in the input file for a given keyword
GTTIMWIN.f90	to read time window information in the related standard panel
OBSFILLST.f90	to create a list of observation files from a given group of keywords
RDPWIN.f90	to get the parameter time window specification
SESTIM.f90	to return the time window related to a given session ID
NUMKEYS.f90	to return the number of values assigned to a keyword
GTKEYW.f	to get value of a keyword directly from data panels
PTKEYW.f	to store values of keywords into data panel
WRITEKEY.f90	to write keywords and their values into the inputfile

continued on next page

continued from previous page

Filename	Purpose
Print commands:	
PRFLNA.f90	to print input and output filenames indicated in the program input file by DESCR_
PRITIT.f90	to print the standard header section for the program output (including program name and program title)
PRFILE.f90	to print a list of filenames into the program output
PRIWIN.f90	to print window to output file
Conversion of time expressions:	
DATTIM.f	to access the system date and time
DSTIME.f	to computate the daylight saving period
IYEAR4.f	to expand 2–digit year to 4–digit year
DJUL.f	to compute the Modified Julian Day (MJD) from year, month, and (real–valued) day
JMT.f	to compute year, month, day of month from Modified Julian Day
GPSMJD.f	to compute the MJD from the GPS week and the number of seconds within the week (starting at Sunday 00:00)
MJDGPS.f	to compute the GPS week and the number of seconds within the week from the Modified Julian Day
WEEKDY.f	to compute the GPS week, the day of the week, and the year from a Modified Julian Day
RADGMS.f	to transform an angle in radians into degrees/hours, minutes and seconds
GMSRAD.f	to transform degrees/hours, minutes, and seconds into an angle (in radian)
DGPSUT.f	to get the difference GPS–time − UTC time in seconds from the leap second file (description in Section 24.3.9)
TIMST2.f90	to write one or two MJD values into a string
TIMSTR.f	same as TIMST2 but for two–digit year
ST2TIM.f90	to read one or two date and time from a string and return the related MJD value(s)
STRTIM.f	same as ST2TIM but for two–digit year
SESWIN.f90	to return the session boundaries to a given time interval
OSAMPL.f90	to check whether an epoch coincides with a sampling epoch and return previous and next sampling epoch
String related subroutines:	
SPLARG.f90	to split line into a list of arguments (blanks or double quotes)
SPLSTR.f	to split strings into sub–strings
WILDCD.f	to test string for wild card
STRIPDIR.f90	to strip the path (directory) from the string
STRIPEXT.f90	to strip the path (directory) and extension from the string
FINDWD.f	to find the n$^{\text{th}}$ word in a string
CHRCOUNT.f90	to count number of a specific character sequence in a string
LOWERC.f	to convert string to lower case
UPPERC.f	to convert string to upper case
SJUSTL.f	left justify a character string
LENGT0.f	to determine the length of a string
LENGT1.f	to determine the length of a string (or 1 for length zero)

continued on next page

continued from previous page

Filename	Purpose
Array allocation and handling:	
ALCERR.f90	to display an error message if an array could not be allocated
CKSIZEC1.f90	to check the size of the string–pointer array and to extend it if needed
CKSIZEI1.f90	to check the size of the integer–pointer array and to extend it if needed
CKSIZER1.f90	to check the size of the real–pointer array and to extend it if needed
DIMTST.f	to check the maximum dimensions
IORDUP.f	to provide an index array sorting a vector of integer vector in ascending order
CORDUP.f	to provide an index array sorting a character vector in ascending order
DORDUP.f	to provide an index array sorting a vector of real values in ascending order
DORDDN.f	to provide an index array sorting a vector of real values in descending order
SORT.f	to reorder the elements of a vector of real values in ascending order
LISTC1.f90	to search item in a character list, update list if desired
LISTI4.f90	to search item in an integer list, update list if desired
MAJOR.f	majority voting: maximum number of elements withing tolerance
MAJOR1.f	to find the array element which has the smallest sum of distances
MAXLR8.f	to search greatest item in a list of real values
MINLR8.f	to search smallest item in a list of real values
Various mathematical functions:	
FAKULT.f	to compute the facrotial
IOR.f	bit–wise logical inclusive or
RANDU.f	to generate a random integer number and a float random number from an input variable
NORMAL.f	to generate a normally distributed random number
NORRND.f90	to generate pairs of a normally distributed random numbers
Satellite related subroutines:	
PRN2SVN.f90	conversion of PRN number to SVN (e.g., “110” to “717”)
LEOPRN.f90	to search the structure of the satellite information file (description in Section 24.3.4) for LEO PRN
SVN2PRN.f90	conversion of SVN to PRN number
CHR2SVN.f90	to convert the system ID and satellite number the full Bernese satellite number (e.g., satellite number “10” of system “R” to “110”)
SVN2CHR.f90	to return the satellite number modulo 100 and the character
TYP2SVN.f	to apply the `CHR2SVN` function to a list of satellites
SVN2TYP.f	to apply the `SVN2CHR` function to a list of satellites
PRN2COS.f90	conversion of PRN number to COSPAR number
COS2PRN.f90	conversion of COSPAR number to PRN number
COS2SLR.f90	conversion of COSPAR number to ILRS number
SLR2COS.f90	to convert of ILRS number to COSPAR number
PRN2PRN.f90	return true PRN number in case of anomalies or inconsistentcies
SVNSYS.f90	to check a list, whether satellites from a certain GNSS or a LEO is contained
MIXSVN.f	to return whether a list contains only satellites from a specific GNSS or LEO, or which combination of them
SATBLK.f90	to return the satellite Block ID from satellite information file (see Section 24.3.4) for a specific satellite
ASMODE.f90	to return the Anti-Spoofing (AS) mode of a GPS satellite
MODSVN.f	to return original SVN for satellites after a repositioning event (removing +50 indicator)
DEFREQ.f	to define frequencies for each indivaidual satellite

continued on next page

continued from previous page	
Filename	Purpose
Station and coordinate handling:	
`XYZELL.f90`	to compute ellipsoidal from cartesian coordinates
`ELLXYZ.f`	to compute cartesian from ellipsoidal coordinates
`ECCELL.f`	to transform a local eccentricity vector from the global cartesian to the local horizont system
`ELLECC.f`	to transform a local eccentricity vector from the local horizont to the global cartesian system
`ERR3D.f`	to compute covariance matrix (3x3) of cartesian coordinates from the covariance matrix of ellipsoidal coord.
`ERR3DM.f90`	to transform the variance–covariance information (from polar to cartesian coordinate system)
Flag handling:	
`CLRFLG.f`	to clear bit number in a character flag
`SETFLG.f`	to set bit number in a character flag
`TSTFLG.f`	to test bit number in a character flag
Exit routines:	
`EXITRC.f`	to end a program, the routine calls the three subsequent modules
`EXIT_DEL.f90`	to delete all files listed in `DELETE_FILES`
`EXIT_MEN.f90`	to print the return code to keyword `MENUAUX_IRCODE`
`EXIT_PGM.f`	to exit the program

If you add a new module (subroutine or function) or add a call to an additional module, you must update the `Makefile` before compilation. On UNIX/Linux/Mac, you update the `Makefile` by running the script

```
${X}/EXE/makemake.pl -r $C
```

and then compile the source code by

```
CBERN ALL
```

On a Windows platform you update the `Makefile` by

```
perl %X%\EXE\makemake.pl -r %C%
```

and then compile the source code by

```
perl %X%\EXE\CBERN ALL
```

Without changing the dependencies between the modules it is sufficient only to compile the modified modules and programs.

25.4.2. Maximum Dimensions

Although the memory of a number of arrays is allocated dynamically, still many dimensions are predefined in module files. This is likely the kind of source code modification that a typical user might need to do.

Most of the main programs have the following line in the declaration part:

```
USE m_maxdim, ONLY: max...
```

```

! -------------------------------------------------------------------------
! Bernese GNSS Software Version 5.2
! -------------------------------------------------------------------------

MODULE m_maxdim

! -------------------------------------------------------------------------
! Purpose:     This module defines several maximum dimension parameters
!
! Author:      L. Mervart
!
! Created:     12-Feb-2003
!
! Changes:     .....
!
! Copyright:  Astronomical Institute
!              University of Bern
!              Switzerland
! -------------------------------------------------------------------------

! MAXAMB: Maximum number of ambiguities
! MAXFLS: Maximum number of files in a session
! MAXGIM: Maximum number of global/local ionosphere models
! MAXGIT: Maximum number of terms per global/local ionosphere models
! MAXPOT: Maximum number of geo-potential terms
! MAXREC: Maximum number of receivers that are processed
! MAXSAA: Maximum number of satellites in satellite information file
! MAXSAC: Maximum number of satellite clock parameters (polynomial degree + 1)
! MAXSAS: Maximum number of satellites at one epoch
! MAXSAT: Maximum number of satellites that are processed
! MAXSTA: Maximum number of stations allowed for processing and in neqs
! MAXCRD: Maximum number of stations allowed in coordinate file
! MAXINT: Maximum number of integration intervals
! MAXSHD: Maximum number of shadow transits of satellites

  USE m_bern,   ONLY: i4b

  IMPLICIT NONE

  INTEGER(i4b),PARAMETER  :: maxvar=21
  INTEGER(i4b),PARAMETER  :: maxbad=500
  INTEGER(i4b),PARAMETER  :: maxgit=300
  INTEGER(i4b),PARAMETER  :: maxocn=11
  INTEGER(i4b),PARAMETER  :: maxatm=2
  INTEGER(i4b),PARAMETER  :: maxrec=800
  INTEGER(i4b),PARAMETER  :: maxsaa=550
  INTEGER(i4b),PARAMETER  :: maxsac=5
  INTEGER(i4b),PARAMETER  :: maxint=5100
  INTEGER(i4b),PARAMETER  :: maxshd=100

  INTEGER(i4b),PARAMETER  :: maxamb=1500
  INTEGER(i4b),PARAMETER  :: maxfls=90
  INTEGER(i4b),PARAMETER  :: maxgim=200
  INTEGER(i4b),PARAMETER  :: maxpot=140
  INTEGER(i4b),PARAMETER  :: maxsas=70
  INTEGER(i4b),PARAMETER  :: maxsat=95
  INTEGER(i4b),PARAMETER  :: maxsta=3000
  INTEGER(i4b),PARAMETER  :: maxcrd=1000
  INTEGER(i4b),PARAMETER  :: maxstc=60

END MODULE m_maxdim
```

Figure 25.1.: Module file ${I}/M_MAXDIM.f90.

The module file ${I}/M_MAXDIM.f90 (Figure 25.1) can be found in the include directory ${C}/INC/FOR. To increase a specific parameter (e.g., maxrec), you may change the corresponding entry (e.g., change maxrec=800 to maxrec=1200).

If you have changed one of the maximum dimensions in ${I}/M_MAXDIM.f90 it is recommended to recompile the source code according to the instructions for your operating system (Section 25.1.5 for Windows and Section 25.2.5 for UNIX/Linux).

26. The Step from Version 5.0 to Version 5.2

If you start to use an existing campaign originally processed with Version 5.0 of the *Bernese GNSS Software* with the new Version 5.2 you have to be aware that you cannot switch back to Version 5.0. Note also, that the results from Version 5.0 cannot be reproduced one-by-one with the Version 5.2 due to some changes in the analysis models, e.g., solid Earth tides. Therefore, it is recommended to duplicate campaigns in the campaign tree of the new Version 5.2 and leave the ones from Version 5.0 unchanged.

26.1. Convert a Campaign from Version 5.0 to Version 5.2

26.1.1. Preparation of the Campaign

Add the name of the campaign to the list of campaigns ("Menu>Campaign>Edit list of campaigns") and select it as the active campaign ("Menu>Campaign>Select active campaign"). You will get a message that no session table is available in your actual campaign. Do not worry about it at this point, the new session table will be created in the next step; when creating the directory structure for the campaign in the CAMPAIGN52-area using "Menu>Campaign>Create new campaign" .

The new campaign directory will contain one subdirectory more than your structure for Version 5.0:

GRD-directory: containing the grid files for VMF1 and/or geophysical crustal deformations.

26.1.2. File Conversions

Not all files from your old campaign-tree can be copied (or moved) into the new directory structure of the campaign: a number of files need to be converted and adapted before you can start processing the data from an existing campaign with Version 5.2 of the *Bernese GNSS Software* (the full list of files with changed formats is given in Section 26.3):

Station information files: The format of the station information file has been changed from Version 5.0 to Version 5.2. You need to run the converter tool STA2STA ("Menu>Service>Station information files>Convert station information files") to transfer your information into the new format.
There are new columns in the station information file related to individually calibrated receiver antennas. If you have such antennas in your processing and if you want to apply the corresponding corrections we recommend to fill the ANTENNA SERIAL NBR fields with the ANT # entries from the RINEX headers.

Antenna phase center corrections file: There is a completely new file format for the antenna phase center corrections. It has to be generated based on antenna calibration values in ANTEX format for all potential antennas and GNSS (according to the station and receiver information file) using the program ATX2PCV ("Menu>Conversion >ANTEX to Bernese format"). For more information on this program we refer to Section 18.3 .

Coordinate and velocity files: The coordinate and velocity files in Version 5.2 contain more digits than in the previous versions of the *Bernese GNSS Software* . To benefit from this development we recommend to use the extended format at least for stations to be used for datum definition. You may download files containing the coordinates and velocities from the most recent realizations of the global terrestrial reference frame (e.g., IGb 08) from `http://www.aiub.unibe.ch/download/BSWUSER52/STA` and use the CRDMERGE program ("Menu>Service>Coordinate tools>Merge coordinate/velocity files") to update the coordinate and velocity files (enter the downloaded files as "Coordinates master file" and your old file as "Merge files").

Atmospheric tidal loading file: Based on the coordinate file you may generate a file containing atmospheric tidal loading corrections for each station if you want to apply such corrections when processing your data. You can use the program GRDS1S2 ("Menu>Service>Coordinate tools>Extract atmospheric tidal loading coefficients") to extract the coefficients for each of your stations from a global grid.

Now you are ready to continue working on your data in the new campaign area of Version 5.2 .

26.2. BPE-Processing for a Converted Campaign

The example BPEs distributed with Version 5.2 follow a different dataflow than in the example distributed with Version 5.0 . The input files are not located in the campaign directories but in a separate `DATAPOOL` area (abbreviated by `${D}`). You have to provide the input files at least according to the structure of the `DATAPOOL` area, see Section 26.2.1 .

If have already used a `DATAPOOL` area for Version 5.0 it is not necessary to change the structure of your `DATAPOOL` but simply to adjust the BPE scripts copying the files into the campaign (typically named `xxx_COP`, where `xxx` indicates the related BPE).

If you have constructed your own complex BPEs you do not want to start developing them for Version 5.2 from scratch. In that case you only want to adapt the existing BPEs to the new environment. Note, that you have to update the models manually to the available standard of Version 5.2 . Natively we cannot provide you with a fully-detailed and tailored description here because your own BPEs may be very different. Nevertheless, we will provide you with some hints fo this procedure in Section 26.2.2.

26.2.1. Adapting the Dataflow for the BPE Examples

According to the locations of input files as it is realized in the BPE examples distributed with Version 5.2 of the *Bernese GNSS Software* you need to prepare your input data in the `DATAPOOL`-area as follows:

- The RINEX subdirectory contains all input RINEX files. Note that these files can also be Hatanaka compressed if you use the scripts to import the observation files from the DATAPOOL into the CAMPAIGN52-area as distributed with Version 5.2 of the *Bernese GNSS Software*.
- Satellite clock, orbit, and EOP input files are expected in a directory indicating their source, e.g., COD or IGS.
- The subdirectory BSW52 contains all information in a format specific to the *Bernese GNSS Software*, e.g., iononsphere models for QIF ambiguity resolution or higher-order ionosphere corrections and DCB corrections.
- Project–specific files are expected in the REF52 subdirectory. These are in particular:
  - reference frame related files (i.e., IGb08_R.CRD, IGb08_R.VEL, IGb08.FIX)
  - station-related files containing the necessary entries for all stations in your project:

XYZ.CRD	a priori coordinate file
XYZ.VEL	a priori velocity file
XYZ.STA	station information file
XYZ.CRX	station information problem file
	both *after* format conversion using the STA2STA program, see above.
XYZ.ABB	abbreviation table
XYZ.PLD	tectonic plate assignment
XYZ.CLU	cluster definition file
XYZ.BLQ	table with ocean tidal loading corrections
XYZ.ATL	table with atmospheric tidal pressure loading corrections
	This file is new in Version 5.2 of *Bernese GNSS Software* and needs to be created based on the coordinate file, see above.

    XYZ denotes an indicator to be used for all project-related filenames.

    We recommend to generate these files in a campaign area, as described in Section 26.1.2 and copy them afterwards into the DATAPOOL area.

Note that if you have already worked with a DATAPOOL-area and a SAVDISK-area you only have to adapt the xxx_COP and xxx_SAV scripts at the beginning and the end of your BPE – where xxx indicates the related BPE.

26.2.2. Adapting Your Existing BPEs for Version 5.2

The first step is to copy the files you need from the GPSUSER-environment of Version 5.0 to the new one related to the current version. You need to copy the content of three directories:

- ${U}/PCF: the PCF for your BPEs
- ${U}/SCRIPT: the user scripts of your BPEs
- ${U}/OPT: (including the subdirectories) containing the input files for the Bernese processing programs

Make sure in case of common filenames between the distributed processing examples and your own BPEs that you clearly decide whether you want to overwrite the distributed files with your own ones, use the distributed ones but completed by your own ones, or merge the content of your files with the distributed ones. A good practice for this purpose might be:

- PCFs with the same name have to be renamed.
- The user scripts (in ${U}/SCRIPT) which have the same name in Version 5.0 and Version 5.2 used for the example campaign can be skipped from copying, i.e. the scripts from Version 5.2 should be used (with exeptions of the xxx_COP and the xxx_SUM scripts.
- If you overwrite existing program input files in option subdirectories, the behaviour of existing BPEs may change (even after updating program input files, see next section).
  If you want to avoid this, you may rename option subdirectories and adapt your PCF accordingly.

Update of the Program Input Files

The program input files have to be updated with the tool UPDPAN ("Menu>Configure>Update program input files"). It is recommended to select all INP files from ${X}/PAN (using * for option "Master program input files") and update the existing files in all option directories (select ${U}/OPT/* in option "Panel directory") to be able to run the old BPEs with the new software version. To keep the old values in your program input files, use the option UPDATE and EXISTING as "UPDATE OPTIONS".

The output file from UPDPAN as well as the updated program input panels which changed should be studied carefully.

Afterwards it is absolutely necessary to check each panel of you BPE because some keywords have been renamed, removed or added to the program input file in order to support new features of the Version 5.2 of the *Bernese GNSS Software*. In some cases also the type of the element may have changed. For this purpose the program CHNGEN ("Menu>Configure >Change general options") might be useful. Note, that such changes may also affect user scripts if you are evaluating or manipulating keywords here using the getKey or putKey functions.

Example: RNX2SNX.PCF from Version 5.0 with New Software

A full list of affected keywords cannot be given here because we cannot know the features you are using in your BPEs. As an example we show here what you need to adapt if you want to run the example BPE RNX2SNX.PCF from Version 5.0 with the current Version 5.2 of the *Bernese GNSS Software*:

Select the panel directories ${U}/OPT/* and set a new value for the two names in panel "CHNGEN 2: Change General Input Files"

"Satellite info"	→	SATELLIT.I01
"Phase center ecc."	→	PCV.I01

Activate the panel for changing individual keywords by checking the box for "Change any keywords". Apply the following changes to the listed keywords for option "Change any keywords" in panel "CHNGEN 2: Change General Input Files":

Keyword	Old value		New value
JPLEPH:	DE200	→	DE405
ABBREV:	ABBREV	→	EXAMPLE
LENINTER:	1.0	→	01 00 00
VARLNINT:	6.0	→	06 00 00
PRINT:	NO	→	0
PRINT:	YES	→	1
BLKRET:	0	→	NO
BLKRET:	1	→	YES

One keyword for an option has been changed in the program input file of RESRMS that has to be set manually by opening the correspondig panels. Using the "Menu>BPE>Edit PCF program input files" you may select the PCF (e.g., RNX2SNX.PCF). The menu presents all processing programs, the name of the user scripts and the corresponding option directories. Select all calls of the RESRMS program (e.g., in GPSEDT_P and RESCHK steps) and introduce the value 180 to the option "Sampling interval of residual files" in panel "RESRMS 2: Options".

This is already enough for the RNX2SNX.PCF example of the old *Bernese GNSS Software*. For other BPEs there might be additional keywords to change.

Update the Models

After verification of the results, you can now update to the latest models, e.g., reference frame IGb 08 and antenna model igs08. For further instructions on this matter, see Sections "Change of the Reference Frame" and "Change of the Antenna Models".

26.3. Changed File Formats

26.3.1. Changed Formats

Here we list all files with changed formats and we indicate how you can convert your information:

- Station information/problem file
  (extensions STA and CRX in the STA directory of the campaign)
  Use program STA2STA ("Menu>Service>Station information files>Convert station information files") to convert these files from Version 5.0 (format version 0.00) to the new format (1.01 or higher).
- Antenna phase center correction file
  Use the new program ATX2PCV ("Menu>Conversion>ANTEX to Bernese format") to read the corrections from ANTEX for all the stations contained in the station information file.
- Satellite information file
  The new formatted file is included in the distribution, but you can also download it from http://www.aiub.unibe.ch/download/BSWUSER52/GEN .
- Constant and datum information files
  Files extended with any additional records are included in the distribution, but you

can also download them from `http://www.aiub.unibe.ch/download/BSWUSER52/GEN`.

- Program input files
  (extension `INP` in `${U}/PAN` or `%U%\PAN` directory)
  Use the new input files distributed with the software. You may use "Menu>Configure >Update program input files" to update your old input files to the new version.

  Be aware that the new input files cannot be displayed with your menu from Version 5.0 anymore.

26.3.2. New Formats without Converters

There are several more file formats that have changed from Version 5.0 to Version 5.2 . The old formats can still be read by the programs but files stored by the program from Version 5.2 cannot be handled by the Version 5.0 programs. These file types are:

- Coordinate and velocity files
  (extensions `CRD` and `VEL` in the `STA` directory of the campaign)
- Normal equation files
  (extension `NQ0` in the `SOL` directory of the campaign)
- Standard orbit files
  (extension `STD` in the `ORB` directory of the campaign)
- Observation (header) files
  (extensions `CZH`, `PZH`, `RZH`, `CSH`, and `PSH` in the `OBS` directory of the campaign)
- `DCB` files may contain new record types
  (extension `DCB` in the `ORB` directory of the campaign)

26.3.3. New File Types

Due to new functionality of the *Bernese GNSS Software* Version 5.2 with respect to Version 5.0 some new file formats have been introduced:

- Grid files
  (Extension `GRD` in the `GRD` directory of the campaign)
  VMF1 coefficients and geophysical models
- Atmospheric tidal loading
  (Extension `ATL` in the `STA` directory of the campaign)
  Extraction (done by program GRDS1S2, "Menu>Service>Coordinate tools>Extract atmospheric tidal loading coefficients") from a global grid for the stations to be processed.
- Gravitational effect of the solid Earth tides on orbits
  (Extension TPO in `${X}/GEN` - or `%X%\GEN` - directory)
  To be used by ORBGEN in the same way as the corrections for the ocean tides
- Satellite-/receiver-specific observation type selection
  (in `${X}/GEN` - or `%X%\GEN` - directory)
  To select the measurement types from a RINEX 2 or a RINEX 3 observation file to be transferred as the first and second frequency into the Bernese formatted observation file

- Bias exchange file
  (extension BIA in the ORB directory of the campaign)
  International exchange format for receiver/satellite code and phase biases
- Event list file and Earthquake file
  (extensions EVL and ERQ in STA directory of the campaign)
  To be used to define a priori events for FODITS ("Menu>Service>Coordinate tools>Analysis of time series") and report significant events from a FODITS analysis (EVL only)
- SLR corrections
  (extension SLR in the STA directory of the campaign)
  To provide biases and center of mass corrections for SLR processing

26.3.4. File Types Not Supported Anymore

Not all file types prior to Version 5.0 are supported anymore, in particular old normal equation files (extension NEQ), station renaming tables (extension STN), receiver/antenna renaming tables (extension TRN), antenna height translation tables (HTR), old abbreviation tables, etc.

Orbital element files (extention ELE) generated with Version 5.0 cannot be integrated by ORBGEN of Version 5.2 .

Bibliography

Argus, D. F. and R. G. Gordon. No–net–rotation model of current plate velocities incorporating plate motion model Nuvel–1. *Geophysical Research Letters*, 18(11):2039–2042, 1991.

Arnold, D., S. Lutz, R. Dach, A. Jäggi, and J. Steinborn. Near real–time coordinate estimation from double–difference GNSS data. In *International Association of Geodesy Symposia*, pages 1–7. Springer Berlin Heidelberg, Potsdam (Germany), 2015a. doi: 10.1007/1345_2015_173.

Arnold, D., M. Meindl, G. Beutler, R. Dach, S. Schaer, S. Lutz, L. Prange, K. Sośnica, L. Mervart, and A. Jäggi. CODE's new solar radiation pressure model for GNSS orbit determination. *Journal of Geodesy*, 89(8):775–791, Aug. 2015b. doi: 10.1007/s00190-015-0814-4.

Bar-Sever, Y. E. A new model for GPS yaw attitude. *Journal of Geodesy*, 70(11):714–723, 1996. doi: 10.1007/BF00867149.

Bassiri, S. and G. Hajj. Higher-order ionospheric effects on the Global Positioning System observables and means of modeling them. *Manuscripta Geodaetica*, 18:280–289, 1993.

Bauerŝíma, I. NAVSTAR/Global Positioning System (GPS), I. In *Mitteilungen der Satelliten–Beobachtungsstation Zimmerwald*, number 7. Astronomical Institute, University of Bern, 1982.

Bauerŝíma, I. NAVSTAR/Global Positioning System (GPS), II. In *Mitteilungen der Satelliten–Beobachtungsstation Zimmerwald*, number 10. Astronomical Institute, University of Bern, 1983.

Berg, H. *Allgemeine Meteorologie*. Dümmler's Verlag, Bonn, 1948.

Beutler, G. Himmelsmechanik II: Der erdnahe Raum. In *Mitteilungen der Satelliten–Beobachtungsstation Zimmerwald*, number 28. Astronomical Institute, University of Bern, 1991.

Beutler, G. The impact of the International GPS Service for Geodynamics on the surveying and mapping community. In *International Union of Surveying and Mapping (IUSM) Presented Papers of the Working Group Sessions*, pages 84–94, Washington, Aug. 1992.

Beutler, G. *Methods of Celestial Mechanics*. Springer–Verlag, Berlin, Heidelberg, New York, 2005. ISBN 3-211-82364-6.

Beutler, G., I. Bauerŝíma, W. Gurtner, M. Rothacher, T. Schildknecht, and A. Geiger. Atmospheric refraction and other important biases in GPS carrier phase observations. In *Atmospheric Effects on Geodetic Space Measurements, Monograph 12*, pages 15–43. School of Surveying, University of New South Wales, Kensington, Australia, 1988.

Beutler, G., E. Brockmann, W. Gurtner, U. Hugentobler, L. Mervart, M. Rothacher, and A. Verdun. Extended orbit modeling techniques at the CODE processing center of the International GPS Service for Geodynamics (IGS): Theory and initial results. *Manuscripta Geodaetica*, 19 (6):367–386, April 1994.

Beutler, G., A. Geiger, M. Rothacher, S. Schaer, D. Schneider, and A. Wiget. Dreidimensionales Testnetz Turtmann 1985–1993, Teil II (GPS-Netz). In *Geodätisch-geophysikalische Arbeiten in der Schweiz*, volume 51. Schweizerische Geodätische Kommission, Institut für Geodäsie und Photogrammetrie, Eidg. Technische Hochschule Zürich, Zürich, 1995.

Beutler, G., E. Brockmann, U. Hugentobler, L. Mervart, M. Rothacher, and R. Weber. Combining consecutive short arcs into long arcs for precise and efficient GPS orbit determination. *Journal of Geodesy*, 70(5):287–299, 1996. doi: 10.1007/BF00867349.

Beutler, G., M. Rothacher, S. Schaer, T. A. Springer, J. Kouba, and R. E. Neilan. The International GPS Service (IGS): An interdisciplinary service in support of earth sciences. *Advances in Space Research*, 23(4):631–653, 1999. doi: 10.1016/S0273-1177(99)00160-X.

Blewitt, G., Y. Bock, and J. Kouba. Constructing the IGS polyhedron by distributed processing. In Zumberge, J. F. and R. Liu, editors, *IGS workshop on Densification of the IERS Terrestrial Reference Frame through regional GPS Networks*, pages 21–41, Jet Propulsion Laboratory, Pasadena, California, USA, November 1994. IGS Central Bureau.

Bock, H., R. Dach, A. Jäggi, and G. Beutler. High-rate GPS clock corrections from CODE: support of 1 Hz applications. *Journal of Geodesy*, 83(11):1083–1094, 2009. doi: 10.1007/s00190-009-0326-1.

Bock, H., A. Jäggi, U. Meyer, P. Visser, J. van den IJssel, T. van Helleputte, M. Heinze, and U. Hugentobler. GPS–derived orbits for the GOCE satellite. *Journal of Geodesy*, 85(11):807–818, 2011. doi: 10.1007/s00190-011-0484-9. Issue "GOCE — The Gravity and Steady-state Ocean Circulation Explorer".

Böhm, J. and H. Schuh. Vienna mapping function in VLBI analyses. *Geophysical Research Letters*, 31(1):L01603, 2004. doi: 10.1029/2003GL018984.

Böhm, J., A. Niell, P. Tregoning, and H. Schuh. Global Mapping Function (GMF): A new empirical mapping function based on numerical weather model data. *Geophysical Research Letters*, 33 (7):L07304, 2006a. doi: 10.1029/2005GL025546.

Böhm, J., B. Werl, and H. Schuh. Troposphere mapping functions for GPS and VLBI from ECMWF operationals analysis data. *Journal of Geophysical Research*, 111(B2):B02406, 2006b. doi: 10.1029/2005JB003629.

Böhm, J., R. Heinkelmann, and H. Schuh. Short Note: A global model of pressure and temperature for geodetic applications. *Journal of Geodesy*, 81(10):679–683, 2007. doi: 10.1007/s00190-007-0135-3.

Brockmann, E. Combination of solutions for geodetic and geodynamic applications of the Global Positioning System (GPS). In *Geodätisch-geophysikalische Arbeiten in der Schweiz*, volume 55. Schweizerische Geodätische Kommission, Institut für Geodäsie und Photogrammetrie, Eidg. Technische Hochschule Zürich, Zürich, 1997.

Brockmann, E. and W. Gurtner. Combination of GPS solutions for densification of the European network: Concepts and results derived from 5 European associated analysis centers of the IGS. In *EUREF workshop, Ankara, May 1996*, 1996.

Brunner, F. K. and M. Gu. An improved model for dual frequency ionospheric correction of GPS observations. *Manuscripta Geodaetica*, 16(3):205–214, 1991.

Caissy, M., L. Agrotis, G. Weber, M. Hernandez-Pajares, and U. Hugentobler. The International GNSS real–time Service. *GPS World*, pages 52–58, June 2012.

Castrique, L. Earth rotation and reference frames submitted to the IERS. Annual report 1995, Central Bureau of IERS — Observatoire de Paris, 1996.

Chen, G. and T. A. Herring. Effects of atmospheric azimuthal asymmetry on the analysis of space geodetic data. *Journal of Geophysical Research*, 102(B9):20489–20502, 1997. doi: 10.1029/97JB01739.

Dach, R. Comparison of IGS AC GLONASS clock correction results. In *IGS Workshop on GNSS Biases 2012*, Bern, Switzerland, January 2012. Presentation.

Dach, R., G. Beutler, U. Hugentobler, S. Schaer, T. Schildknecht, T. A. Springer, G. Dudle, and L. Prost. Time transfer using GPS carrier phase: Error propagation and results. *Journal of Geodesy*, 77(1–2):1–14, 2003. doi: 10.1007/s00190-002-0296-z.

Dach, R., S. Schaer, U. Hugentobler, and T. Schildknecht. Combined multi–system GNSS analysis for time and frequency transfer. In *Proceedings of the* 20^{th} *European Frequency and Time Forum EFTF 2006*, Braunschweig, Germany, 2006a.

Dach, R., T. Schildknecht, U. Hugentobler, L.-G. Bernier, and G. Dudle. Continuous geodetic time transfer analysis method. *Ultrasonics, Ferroelectrics, and Frequency Control, IEEE Transactions on*, 53(7):1250–1259, 2006b. doi: 10.1109/TUFFC.2006.1665073.

Dach, R., U. Hugentobler, P. Fridez, and M. Meindl, editors. *Bernese GPS Software, Version 5.0.* Astronomical Institute, University of Bern, Bern, Switzerland, Jan. 2007. URL `ftp://ftp.unibe.ch/aiub/BERN50/DOCU/DOCU50.pdf`. User manual.

Dach, R., E. Brockmann, S. Schaer, G. Beutler, M. Meindl, L. Prange, H. Bock, A. Jäggi, and L. Ostini. GNSS processing at CODE: Status report. *Journal of Geodesy*, 83(3–4):353–365, 2009. doi: 10.1007/s00190-008-0281-2.

Dach, R., J. Böhm, S. Lutz, P. Steigenberger, and G. Beutler. Evaluation of the impact of atmospheric pressure loading modeling on GNSS data analysis. *Journal of Geodesy*, 85(2): 75–91, 2011a. doi: 10.1007/s00190-010-0417-z.

Dach, R., M. Meindl, S. Schaer, S. Lutz, R. Schmid, and G. Beutler. System dependence of GNSS receiver antenna calibrations. In *EGU General Assembly*, Vienna, Austria, April 2011b. Poster.

Dach, R., R. Schmid, M. Schmitz, D. Thaller, S. Schaer, S. Lutz, P. Steigenberger, G. Wübbena, and G. Beutler. Improved antenna phase center models for GLONASS. *GPS Solutions*, 15 (1):49–65, 2011c. doi: 10.1007/s10291-010-0169-5.

Dach, R., S. Schaer, S. Lutz, D. Arnold, H. Bock, E. Orliac, L. Prange, A. Villiger, L. Mervart, A. Jäggi, G. Beutler, E. Brockmann, D. Ineichen, A. Wiget, A. Rülke, D. Thaller, H. Habrich, W. Söhne, J. Ihde, P. Steigenberger, and U. Hugentobler. Center for Orbit Determination In Europe: IGS Technical Report 2014. In Dach, R. and Y. Jean, editors, *International GNSS Service: Technical Report 2014*, pages 21–34. IGS Central Bureau, May 2015.

Delle Donne, D., A. J. L. Harris, M. Ripepe, and R. Wright. Earthquake-induced thermal anomalies at active volcanoes. *Geology*, 38(9):771–774, 2010. doi: 10.1130/G30984.1.

DeMets, C., R. G. Gordon, D. F. Argus, and S. Stein. Effect of recent revisions to the geomagnetic reversal time scale on estimates of current plate motions. *Geophysical Research Letters*, 21 (20):2191–2194, 1994.

Dick, W. R. and B. Richter. IERS annual report 2000. Annual report 2000, International Earth Rotation Service, Central Bureau, Frankfurt am Main: Bundesamts für Kartographie und Geodäsie, 2001.

Dong, P., P. Fang, Y. Bock, M. K. Cheng, and S. Miyazaki. Anatomy of apparent seasonal variations from GPS-derived site position time series. *Journal of Geophysical Research*, 107(B4), 2002. doi: 10.1029/2001JB000573.

Dow, J. M., R. E. Neilan, and C. Rizos. The International GNSS Service in a changing landscape of Global Navigation Satellite Systems. *Journal of Geodesy*, 83(3–4):191–198, 2009. doi: 10.1007/s00190-008-0300-3.

Drinkwater, M., R. Haagmans, D. Muzi, A. Popescu, R. Floberghagen, M. Kern, and M. Fehringer. The GOCE gravity mission: ESA's first core explorer. In *Proceedings of the 3rd GOCE User Workshop, 6-8 November 2006, Frascati, Italy, ESA SP-627*, pages 1–7, 2006.

Ekman, M. The Permanent Problem of the Permanent Tide: What to do with it in Geodetic Reference Sytems? *Bulletin d'Informations Marées Terrestres*, 125:9508–9513, 1995. Obs. Royal de Belgique, Brussels.

Eriksson, D., D. S. MacMillan, and J. M. Gipson. Tropospheric delay ray tracing applied in VLBI analysis. *Journal of Geophysical Research*, 119(12):9156–9170, 2014. doi: 10.1002/2014JB011552.

Essen, L. and K. D. Froome. The refractive indices and dielectric constants of air and its principal constituents at 24 000 mc/s. *Proceedings of Physical Society*, 64(B):862–875, 1951.

Fliegel, H. F., T. E. Gallini, and E. R. Swift. Global Positioning System radiation force model for geodetic applications. *Geophysical Research Letters*, 97(B1):559–568, 1992. doi: 10.1029/91JB02564.

Frei, E. Rapid differential positioning with the Global Positioning System (GPS). In *Geodätisch-geophysikalische Arbeiten in der Schweiz*, volume 44. Schweizerische Geodätische Kommission, Institut für Geodäsie und Photogrammetrie, Eidg. Technische Hochschule Zürich, Zürich, 1991.

Frei, E. and G. Beutler. Rapid static positioning based on the Fast Ambiguity Resolution Approach "FARA": Theory and first results. *Manuscripta Geodaetica*, 15(6):325–356, 1990.

Fritsche, M., R. Dietrich, C. Knöfel, A. Rülke, S. Vey, M. Rothacher, and P. Steigenberger. Impact of higher-order ionospheric terms on GPS estimates. *Geophysical Research Letters*, 32(23), 2005. doi: 10.1029/2005GL024342.

Fritsche, M., K. Sośnica, C. Rodriguez-Solano, P. Steigenberger, K. Wang, R. Dietrich, R. Dach, U. Hugentobler, and M. Rothacher. Homogeneous reprocessing of GPS, GLONASS and SLR observations. *Journal of Geodesy*, 88(7):625–642, 2014. doi: 10.1007/s00190-014-0710-3.

Gao, Y., F. Lahaye, P. Heroux, X. Liao, N. Beck, and M. Olynik. Modeling and estimation of C1–P1 bias in GPS receivers. *Journal of Geodesy*, 74(9):621–626, 2001. doi: 10.1007/s001900000117.

GLONASS-ICD. *GLONASS Interface Control Document, Version 5.1.* Coordination Scientific Information Center, Moscow, Russia, 2008.

Goad, C. C. and L. Goodman. A modified hopfield tropospheric refraction correction model. In *Proceedings of the Fall Annual Meeting of the American Geophysical Union*, San Francisco, California, December 12–17, 1974.

GPS-ICD. *GPS Interface Control Document, Revision C (ICD-GPS-200C.* US Departement of Defense (DoD), 1993. URL `http://www.navcen.uscg.gov/pubs/gps/icd200/`.

GPS-IS. *GPS Interface Specification Document, Revision H (IS-GPS-200H).* GPS Directorate Systems Engineering and Integration, 2013. URL `http://www.gps.gov/technical/icwg/IS-GPS-200H.pdf`.

GPS-NAV. *GPS Modernized Civil Navigation Live-Sky Broadcast Test Plan, Revision H (IS-GPS-200H.* GPS Directorate, 2013. URL `http://www.navcen.uscg.gov/Content/GPS/CNAVTesting/L2C_L5_CNAV_Test_Plan.pdf`.

Habrich, H. *Geodetic applications of the Global Navigation Satellite System (GLONASS) and of GLONASS/GPS combinations.* PhD thesis, Astronomical Institute, University of Bern, Bern, Switzerland, 1999.

Hefty, J., M. Rothacher, T. A. Springer, R. Weber, and G. Beutler. Analysis of the first year of earth orientation parameters with a sub–daily resolution gained at the CODE processing center of the IGS. *Journal of Geodesy*, 74(6):479–487, 2000. doi: 10.1007/s001900000108.

Helmert, F. R. *Die Ausgleichsrechnung nach der Methode der kleinsten Quadrate.* Teubner, Leipzig, 1872.

Hilla, S. Extending the Standard Product 3 (SP3) orbit format. In Tetreault, P., editor, *Proceedings of the IGS Network, Data, and Analysis Center Workshop, Towards Real Time*, Ottawa, Canada, April 8–11, 2002.

Hilla, S. The extended Standard Product 3 orbit format (SP3-c). Format description, August 2010. URL `ftp://igscb.jpl.nasa.gov/igscb/data/format/sp3c.txt`.

Hofmann-Wellenhof, B., H. Lichtenegger, and J. Collins. *GPS: Theory and Practice.* Springer, 1992. ISBN 3-211-82364-6.

Hopfield, H. S. Two-quadratic tropospheric refractivity profile for correcting satellite data. *Journal of Geophysical Research*, 74:4487–4499, 1969.

Hugentobler, U., S. Schaer, and P. Fridez, editors. *The Bernese GPS Software Version 4.2.* Astronomical Institute, University of Bern, Bern, Switzerland, Feb. 2001. URL `ftp://ftp.unibe.ch/aiub/BERN42/DOCU/docu42.pdf`. User manual.

Hugentobler, U., M. Meindl, G. Beutler, H. Bock, R. Dach, A. Jäggi, C. Urschl, L. Mervart, M. Rothacher, S. Schaer, E. Brockmann, D. Ineichen, A. Wiget, U. Wild, G. Weber, H. Habrich, and C. Boucher. CODE IGS analysis center technical report 2003/2004. In Gowey, K., R. E. Neilan, and A. W. Moore, editors, *IGS 2004 Technical Reports*, pages 40–51, Jet Propulsion Laboratory, Pasadena, California, USA, 2008. IGS Central Bureau.

Ineichen, D., M. Rothacher, T. A. Springer, and G. Beutler. Computation of precise GLONASS orbits for IGEX-98. In Schwarz, K.-P., editor, *Geodesy Beyond 2000, The Challenges of the First Decade*, volume 121 of *International Association of Geodesy Symposia*, pages 26–31. Springer Berlin Heidelberg, 2000. doi: 10.1007/978-3-642-59742-8_5.

Ineichen, D., T. A. Springer, and G. Beutler. Combined processing of the IGS and the IGEX network. *Journal of Geodesy*, 75(11):575–586, 2001. doi: 10.1007/s001900000152.

Ineichen, D., G. Beutler, and U. Hugentobler. Sensitivity of GPS and GLONASS orbits with respect to resonant geopotential parameters. *Journal of Geodesy*, 77(7–8):478–486, 2003. doi: 10.1007/s00190-003-0348-z.

Jäggi, A. Pseudo-stochastic orbit modeling of low Earth satellites using the Global Positioning System. In *Geodätisch-geophysikalische Arbeiten in der Schweiz*, volume 73. Schweizerische Geodätische Kommission, Institut für Geodäsie und Photogrammetrie, Eidg. Technische Hochschule Zürich, Zürich, 2007.

Jäggi, A., U. Hugentobler, and G. Beutler. Pseudo-stochastic orbit modeling techniques for low-Earth orbiters. *Journal of Geodesy*, 80(1):47–60, 2006. doi: 10.1007/s00190--006--0029--9.

Jäggi, A., U. Hugentobler, H. Bock, and G. Beutler. Precise orbit determination for GRACE using undifferenced or doubly differenced GPS data. *Advances in Space Research*, 39(10): 1612–1619, 2007. doi: 10.1016/j.asr.2007.03.012.

Jäggi, A., H. Bock, R. König, Y. Moon, M. Wermuth, and O. Montenbruck. Inter-agency comparison of TerraSAR-X/TanDEM-X baseline solutions. In *AGU General Assembly*, Dec. 2011a. Poster.

Jäggi, A., H. Bock, L. Prange, U. Meyer, and G. Beutler. GPS-only gravity field recovery with GOCE, CHAMP, and GRACE. *Advances in Space Research*, 47(6):1020–1028, 2011b. doi: 10.1016/j.asr.2010.11.008.

Janes, H. W., R. B. Langley, and S. P. Newby. A comparison of several models for the prediction of tropospheric propagation delay. In *Proceedings 5th International Geodetic Symposium on Satellite Positioning*, pages 777–788, Las Cruces, New Mexico, USA, 1989.

Jefferson, D. C., M. B. Heflin, and R. J. Muellerschoen. Examining the C1-P1 Pseudorange Bias. *GPS Solutions*, 4(4):25–30, 2001. doi: 10.1007/PL00012862.

Koch, K. R. *Parameter estimation and hypothesis testing in linear models.* Springer, Berlin Heidelberg New York, 1988.

Kouba, J. Improved relativistic transformations in GPS. *GPS Solutions*, 8(4):170–180, 2004. doi: 10.1007/s10291-004-102-x.

Kouba, J., J. Ray, and M. M. Watkins. IGS reference frame realization. In Dow, J. A., J. Kouba, and T. A. Springer, editors, *Proceedings of the 1998 IGS Analysis Center Workshop*, Darmstadt, Germany, February 1998. ESA/ESOC.

Kouba, J. et al. SINEX—Solution-Independent Exchange Format Version 1.00. In Neilan, R. E., P. A. Van Scoy, and J. F. Zumberge, editors, *Proceedings of the IGS Analysis Center Workshop, Silver Spring, Maryland, USA*, pages 233–276, JPL, Pasadena, California, USA, March 19–21, 1996. IGS Central Bureau. URL `ftp://igscb.jpl.nasa.gov/igscb/data/format/sinex.txt`.

Langley, R. B. Propagation of the GPS signals. In *Lecture Notes International School "GPS for Geodesy"*. Springer-Verlag, Delft, The Netherlands, 1996.

Lee-Lueng, F., E. J. Christensen, C. A. Y. Jr., M. Lefebvre, Y. Ménard, M. Dorrer, and P. Escudier. TOPEX/POSEIDON mission overview. *Journal of Geophysical Research*, 99(C12):24369–24381, 1994.

Leick, A. *GPS Satellite Surveying.* Wiley, 1995. ISBN 0-471-30626-6.

Lutz, S., S. Schaer, M. Meindl, R. Dach, and P. Steigenberger. Higher-order ionosphere modeling for CODE's next reprocessing activities. In *AGU Fall Meeting*, San Francisco, California, Dec. 2009. Poster.

Lutz, S., S. Schaer, M. Meindl, R. Dach, and P. Steigenberger. Higher-order ionosphere modeling for CODE's next reprocessing activities. In *IGS Workshop*, Newcastle upon Tyne, 2010. International GNSS Service. Poster.

MacMillan, D. S. Atmospheric gradients from Very Long Baseline Interferometry observations. *Geophysical Research Letters*, 22(9):1041–1044, 1995. doi: 10.1029/95GL00887.

Marini, J. W. and C. W. Murray. Correction of laser range tracking data for atmospheric refraction at elevations above 10 degrees. Technical Report X-591-73-351, NASA GSFC, 1973.

McCarthy, D. D. IERS conventions (1996). IERS Technical Note 21, Observatoire de Paris, Paris, July 1996. URL `http://www.iers.org/SharedDocs/Publikationen/EN/IERS/Publications/tn/TechnNote21/tn21.pdf`.

McCarthy, D. D. and G. Petit. IERS conventions (2003). IERS Technical Note 32, Bundesamt für Kartographie und Geodäsie, Frankfurt am Main, 2004. URL `http://www.iers.org/iers/publications/tn/tn32/`.

Meindl, M., S. Schaer, U. Hugentobler, and G. Beutler. Tropospheric gradient estimation at code: Results from global solutions. In al., R. A. A. et, editor, *Applications of GPS Remote Sensing to Meteorology and Related Fields*, volume 82(1B) of *Journal of the Meteorological Society of Japan*, pages 331–338. Meteorological Society of Japan, 2004.

Meindl, M., R. Dach, S. Schaer, U. Hugentobler, and G. Beutler. Developing a multi-GNSS analysis software for scientific purposes. In *2nd International Colloquium — Scientific and Fundamental Aspects of the Galileo Programme*, Padua, Italy, Oct. 2009.

Meindl, M., G. Beutler, D. Thaller, R. Dach, and A. Jäggi. Geocenter coordinates estimated from GNSS data as viewed by perturbation theory. *Advances in Space Research*, 51(7):1047–1064, 2013. doi: 10.1016/j.asr.2012.10.026.

Melbourne, W. G. The case for ranging in GPS based geodetic systems. In Goad, C., editor, *Proceedings of the 1st International Symposium on Precise Positioning with the Global Positioning System*, pages 373–386, US Department of Commerce, Rockville, Maryland, 1985.

Mendes, V. and E. Pavlis. High-accuracy zenith delay prediction at optical wavelengths. *Geophysical Research Letters*, 31:L14602, 2004. doi: doi:10.1029/2004GL020308.

Mervart, L. Ambiguity resolution techniques in geodetic and geodynamic applications of the Global Positioning System. volume 53 of *Geodätisch-geophysikalische Arbeiten in der Schweiz*. Schweizerische Geodätische Kommission, Institut für Geodäsie und Photogrammetrie, Eidg. Technische Hochschule Zürich, Zürich, 1995.

Mervart, L., G. Beutler, and U. Wild. Ambiguity resolution strategies using the results of the International GPS Geodynamics Service (IGS). *Bulletin Géodésique*, 68(1):29–38, 1994. doi: 10.1007/BF00806750.

Mervart, L., G. Beutler, M. Rothacher, and S. Schaer. The impact of ambiguity resolution on GPS orbit determination and on global geodynamics studies. presented at the XXI. General Assembly of the International Union of Geodesy and Geophysics, Boulder, Colorado, July 2–14, 1995.

Montenbruck, O., E. Gill, and R. Kroes. Rapid orbit determination of LEO satellites using IGS clock and ephemeris products. *GPS Solutions*, 9(3):226–235, 2005. doi: 10.1007/s10291-005-0131-0.

Mueller, I. I. and G. Beutler. The International GPS Service for Geodynamics — development and current structure. In *Proceedings 6th International Geodetic Symposium on Satellite Positioning, Vol. 2*, pages 823–835, Ohio State University, Columbus, Ohio, USA, 1992.

Niell, A. E. Global mapping functions for the atmosphere delay at radio wavelengths. *Journal of Geophysical Research*, 101(B2):3227–3246, 1996. doi: 10.1029/95JB03048.

Ostini, L. *Analysis and Quality Assessment of GNSS–Derived Paramter Time Series*. PhD thesis, Astronomical Institute, University of Bern, Bern, Switzerland, Feb. 2012.

Ostini, L., R. Dach, M. Meindl, S. Schaer, and U. Hugentobler. FODITS: A new tool of the Bernese GPS Software. In Torres, J. A. and H. Hornik, editors, *Subcommission for the European Reference Frame (EUREF)*, 2008.

Ostini, L., R. Dach, S. Schaer, U. Hugentobler, M. Meindl, and G. Beutler. Time series analysis using FODITS. In *COST ES0701 WG 1-3 meeting*, Nottingham, UK, Mar. 2010. URL `U:\Groups\gps\PosterNottiingham2010\PosterNottingham2010Ati`. Poster.

Petit, G. and B. Luzum. IERS Conventions (2010). IERS Technical Note 36, Bundesamt für Kartographie und Geodäsie, Frankfurt am Main, 2010. URL `http://www.iers.org/IERS/EN/Publications/TechnicalNotes/tn36.html`.

Petrie, E. J., M. A. King, P. Moore, and D. A. Lavallée. Higher-order ionospheric effects on the gps reference frame and velocities. *Journal of Geophysical Research*, 115(B3):B03417, 2010. doi: 10.1029/2009JB006677.

Povalyaev, A. New Structure for GLONASS Navigation Message. *GPS World*, 24(11):36, November 2013.

Ray, J. New pseudorange bias convention. IGS Mail No. 2744, IGS Central Bureau Information System, 2000.

Ray, J. Updated P1–C1 pseudorange bias corrections. IGS Mail No. 3160, IGS Central Bureau Information System, 2001.

Ray, R. D. and R. M. Ponte. Barometric tides from ECMWF operational analyses. *Annales Geophysicae*, 21(8):1897–1910, 2003. doi: 10.5194/angeo-21-1897-2003.

Ray, R. D., D. J. Steinberg, B. F. Chao, and D. E. Cartwright. Diurnal and semidiurnal variations in the Earth's rotation rate induced by oceanic tides. *Science*, 264(5160):830–832, 1994. doi: 10.1126/science.264.5160.830.

Reigber, C., H. Lühr, and P. Schwintzer. CHAMP mission status. *Advances in Space Research*, 30(2):129–134, 2002. doi: 10.1016/S0273-1177(02)00276-4.

Remondi, B. W. Global Positioning System carrier phase, description and use. *Bulletin Géodésique*, 59(4):361–377, 1985. doi: 10.1007/BF02521069.

Remondi, B. W. Extending the National Geodetic Survey Standard GPS Orbit Formats. Technical Report NOS 133 NGS 46, NOAA, USA, 1989.

Remondi, B. W. NGS second generation ASCII and binary orbit formats and associated interpolated studies. presented at the XX. General Assembly of the International Union of Geodesy and Geophysics, Vienna, Austria, Aug. 1991.

Rodriguez-Solano, C. J., U. Hugentobler, P. Steigenberger, and S. Lutz. Impact of Earth radiation pressure on GPS position estimates. *Journal of Geodesy*, 86(5):309–317, 2012. doi: 10.1007/s00190-011-0517-4.

Rothacher, M. Orbits of satellite systems in space geodesy. volume 46 of *Geodätisch-geophysikalische Arbeiten in der Schweiz*. Schweizerische Geodätische Kommission, Institut für Geodäsie und Photogrammetrie, Eidg. Technische Hochschule Zürich, Zürich, 1992.

Rothacher, M. and G. Beutler. The role of GPS in the study of global change. *Physics and Chemistry of the Earth*, 23(9-10):1029–1040, 1998. doi: 10.1016/S0079-1946(98)00143-8.

Rothacher, M. and L. Mervart. *The Bernese GPS Software Version 4.0.* Astronomical Institute, University of Bern, University of Bern, Sept. 1996. URL `ftp://ftp.unibe.ch/aiub/BERN40/DOCU/`. User manual.

Rothacher, M., G. Beutler, W. Gurtner, A. Geiger, H. G. Kahle, and D. Schneider. The Swiss 1985 GPS campaign. In *Proceedings of the Fourth International Symposium on Satellite Positioning*, volume 2, pages 979–991, Austin, Texas, 1986.

Rothacher, M., G. Beutler, W. Gurtner, E. Brockmann, and L. Mervart. *The Bernese GPS Software, Version 3.4.* Druckerei der Universität Bern, Astronomical Institute, University of Bern, 1993.

Rothacher, M., G. Beutler, and L. Mervart. The perturbation of the orbital elements of GPS satellites through direct radiation pressure. In Gendt, G. and G. Dick, editors, *IGS Workshop Proceedings on Special Topics and New Directions*, pages 152–166, Potsdam, Germany, May 15–18, 1995a. GeoForschungsZentrum.

Rothacher, M., S. Schaer, L. Mervart, and G. Beutler. Determination of antenna phase center variations using GPS data. In Gendt, G. and G. Dick, editors, *IGS Workshop Proceedings on Special Topics and New Directions*, pages 77–92, Potsdam, Germany, May 15–18, 1995b. GeoForschungsZentrum.

Rothacher, M., W. Gurtner, S. Schaer, R. Weber, and H. O. Hase. Azimuth- and elevation-dependent phase center corrections for geodetic GPS antennas estimated from GPS calibration campaigns. In Beutler, G., W. G. Melbourne, G. W. Hein, and G. Seeber, editors, *GPS Trends in Precise Terrestrial, Airborne, and Spaceborne Applications*, volume 115 of *International Association of Geodesy Symposia*, pages 333–338. Springer Berlin Heidelberg, 1996. doi: 10.1007/978-3-642-80133-4_56.

Rothacher, M., T. A. Springer, S. Schaer, and G. Beutler. Processing strategies for regional GPS networks. In Brunner, F. K., editor, *Advances in Positioning and Reference Frames*, volume 118 of *International Association of Geodesy Symposia*, pages 93–100. Springer Berlin Heidelberg, 1998. doi: 10.1007/978-3-662-03714-0_14.

Rothacher, M., G. Beutler, T. A. Herring, and R. Weber. Estimation of nutation using the Global Positioning System. *Journal of Geophysical Research*, 104(B3):4835–4859, 1999. doi: 10.1029/1998JB900078.

Saastamoinen, I. I. Contribution to the theory of atmospheric refraction. *Bulletin Géodésique*, 105(1):279–298, 1972. doi: 10.1007/BF02521844.

Santerre, R. Impact of GPS satellite sky distribution. *Manuscripta Geodaetica*, 16:28–53, 1991.

Schaer, S. Stochastische Ionosphärenmodellierung beim Rapid Static Positioning mit GPS. Diplomarbeit, Astronomisches Institut, Universität Bern, 1994.

Schaer, S. CODE's Global Ionosphere Maps (GIMs), 1998. URL `http://www.aiub.unibe.ch/ionosphere/`. automatically updated web site.

Schaer, S. Mapping and predicting the Earth's ionosphere using the Global Positioning System. volume 59 of *Geodätisch-geophysikalische Arbeiten in der Schweiz*. Schweizerische Geodätische Kommission, Institut für Geodäsie und Photogrammetrie, Eidg. Technische Hochschule Zürich, Zürich, Switzerland, 1999.

Schaer, S. Monitoring P1–C1 code biases. IGS Mail No. 2827, IGS Central Bureau Information System, 2000.

Schaer, S. TRIMBLE 4700. IGS Mail No. 3887, IGS Central Bureau Information System, 2002.

Schaer, S., G. Beutler, L. Mervart, M. Rothacher, and U. Wild. Global and regional ionosphere models using the GPS double difference phase observable. In Gendt, G. and G. Dick, editors, *IGS Workshop Proceedings on Special Topics and New Directions*, pages 77–92, Potsdam, Germany, May 15–18, 1995. GeoForschungsZentrum.

Schaer, S., G. Beutler, M. Rothacher, and T. A. Springer. Daily global ionosphere maps based on GPS carrier phase data routinely produced by the CODE analysis center. In Neilan, R. E., P. A. Van Scoy, and J. F. Zumberge, editors, *Proceedings of the IGS Analysis Center Workshop, Silver Spring, Maryland, USA*, pages 181–192, JPL, Pasadena, California, USA, March 19–21, 1996. IGS Central Bureau.

Schaer, S., W. Gurtner, and J. Feltens. IONEX: The IONosphere Map EXchange Format Version 1. In Dow, J. M., J. Kouba, and T. A. Springer, editors, *Proceedings of the IGS Analysis Center Workshop*, pages 233–247, Darmstadt, Germany, February 9–11, 1998. ESA/ESOC. URL `ftp://igscb.jpl.nasa.gov/igscb/data/format/ionex1.pdf`.

Schaer, S., G. Beutler, M. Rothacher, E. Brockmann, A. Wiget, and U. Wild. The impact of the atmosphere and other systematic errors on permanent GPS networks. In Schwarz, K.-P., editor, *Geodesy Beyond 2000, The Challenges of the First Decade*, volume 121 of *International Association of Geodesy Symposia*, pages 373–380. Springer Berlin Heidelberg, 2000. doi: 10.1007/978-3-642-59742-8_61.

Schaer, S. C., E. Brockmann, G. Beutler, and M. Meindl. Rapid static positioning using GPS and GLONASS. *Bulletin of Geodesy and Geomatics*, LXIX(2–3):179–194, May/December 2010.

Schmid, R. and M. Rothacher. Estimation of elevation-dependent satellite antenna phase center variations of GPS satellites. *Journal of Geodesy*, 77(7–8):440–446, 2003. doi: 10.1007/s00190-003-0339-0.

Seidelmann, P. K. *Explanatory Supplement to the Astronomical Almanac*. University Science Books, ISBN 0-935702-68-7, Mill Valley, California, 1992.

Slater, J. A., P. Willis, G. Beutler, W. Gurtner, W. Lewandowski, C. Noll, R. Weber, R. E. Neilan, and G. Hein. The International GLONASS Experiment (IGEX-98): Organization, preliminary results and future plans. In *Prodeedings of the ION GPS-99*, Nashville, Tennessee, Sept. 1999.

Sleewaegen, J. M., A. Simsky, W. de Wilde, F. Boon, and T. Willems. Demystifying GLONASS inter-frequency carrier phase biases. 7(3):57–61, 2012.

Sośnica, K. Determination of precise satellite orbits and geodetic parameters using Satellite Laser Ranging. volume 93 of *Geodätisch-geophysikalische Arbeiten in der Schweiz*. Schweizerische Geodätische Kommission, Institut für Geodäsie und Photogrammetrie, Eidg. Technische Hochschule Zürich, Zürich, 2015.

Sośnica, K., D. Thaller, A. Jäggi, R. Dach, and G. Beutler. Sensitivity of Lageos orbits to global gravity field models. *Artificial Satelllites*, 47(2):47–65, 2012. doi: 10.2478/v10018-012-0013-y.

Sośnica, K., D. Thaller, R. Dach, A. Jäggi, and G. Beutler. Impact of loading displacements on SLR-derived parameters and on the consistency between GNSS and SLR results. *Journal of Geodesy*, 87(8):751–769, 2013. doi: 10.1007/s00190-013-0644-1.

Sośnica, K., A. Jäggi, D. Thaller, G. Beutler, and R. Dach. Contribution of Starlette, Stella, and AJISAI to the SLR-derived global reference frame. *Journal of Geodesy*, 88(8):789–804, 2014. doi: 10.1007/s00190-014-0722-z.

Springer, T. A. Modeling and validating orbits and clocks using the Global Positioning System. volume 60 of *Geodätisch-geophysikalische Arbeiten in der Schweiz*. Schweizerische Geodätische Kommission, Institut für Geodäsie und Photogrammetrie, Eidg. Technische Hochschule Zürich, Zürich, 2000.

Springer, T. A., G. Beutler, and M. Rothacher. A new solar radiation pressure model for the GPS satellites. *GPS Solutions*, 2(3):50–62, 1999. doi: 10.1007/PL00012757.

Standish, E. M. The observational basis for JPL's DE200, the planetary ephemerides of the astronomical almanac. *Astronomy and Astrophysics*, 233:252–271, 1990.

Steigenberger, P., J. Böhm, and V. Tesmer. Comparison of GMF/GPT with VMF1/ECMWF and implications for atmospheric loading. *Journal of Geodesy*, 83(10):943–951, 2009. doi: 10.1007/s00190-009-0311-8.

Steigenberger, P., S. Lutz, R. Dach, and U. Hugentobler. CODE contribution to the first IGS reprocessing campaign. Technical report, Technische Universität München, 2011.

Švehla, D. and M. Rothacher. Kinematic precise orbit determination for gravity field determination. In Sansò, F., editor, *A Windowon the Future of Geodesy*, volume 128 of *International Association of Geodesy Symposia*, pages 181–188. Springer Berlin Heidelberg, 2005. doi: 10.1007/3-540-27432-4_32.

Tapley, B. D., S. Bettadpur, J. C. Ries, and M. M. Watkins. GRACE measurements of mass variability in the Earth system. *Science*, 305(5683):503–505, 2004. doi: 10.1126/science.1099192.

Tatevian, S. and S. Kuzin. A Development of the Russian Geodetic Reference Network. In *Reference Frames for Applications in Geodetic Science, IAG Commission I Symposium, Luxembourg, October 13–17, 2014*, IAG Symposia. Springer, 2015. doi: 10.1007/1345_2015_181.

Teunissen, P. J. G. and A. Kleusberg, editors. *GPS for Geodesy*. Springer, 1998a. ISBN 3-540-63661-7.

Teunissen, P. J. G. and A. Kleusberg, editors. *GPS for Geodesy*, chapter Quality Control and GPS. Springer-Verlag, Berlin, Heidelberg, New York, 1998b. ISBN 3-540-63661-7.

Thaller, D., K. Sośnica, G. Beutler, R. Dach, and A. Jäggi. *Combining SLR and GNSS measurements*, chapter Commission 1: Reference Frames, pages 12–14. Swiss National Report on the Geodetic Activities in the years 2007-2011 to the XXV General Assembly of the IUGG, Melbourne, Australia, 28 June - 7 July 2011. Swiss Geodetic Commission, 2011a.

Thaller, D., K. Sośnica, R. Dach, A. Jäggi, M. Mareyen, B. Richter, and G. Beutler. GNSS satellites as co-locations for a combined GNSS and SLR analysis. In *XXV IUGG General Assembly, Earth on the Edge: Science for a Sustainable Planet*, Melbourne, Australia, July 2011b. Presentation.

Thaller, D., K. Sośnica, R. Dach, A. Jäggi, and G. Beutler. LAGEOS–ETALON solutions using the Bernese GNSS Software. In *Proceedings of the 17^{th} International Workshop on Laser Ranging, Extending the Range*, volume 48 of *Mitteilungen des Bundesamtes für Kartographie und Geodäsie*, pages 333–336, 2012.

Urschl, C., G. Beutler, W. Gurtner, U. Hugentobler, and S. Schaer. Contribution of SLR tracking data to GNSS orbit determination. *Advances in Space Research*, 39(10):1515–1523, 2007. doi: 10.1016/j.asr.2007.01.038.

van Dierendonck, A., S. Russel, E. Kopitzke, and M. Birnbaum. The GPS navigation message. *Navigation: Journal of the Institute of Navigation*, 25(2):147–165, 1978.

Vaníček, P. and E. J. Krakiwsky. *Geodesy: The Concepts*. North Holland, Amsterdam, 1982.

Wild, U. Ionosphere and geodetic satellite systems, permanent GPS tracking data for modelling and monitoring. volume 48 of *Geodätisch-geophysikalische Arbeiten in der Schweiz*. Institut für Geodäsie und Photogrammetrie, Eidg. Technische Hochschule Zürich, Zürich, 1994.

Willis, P., J. Slater, G. Beutler, W. Gurtner, C. Noll, R. Weber, R. E. Neilan, and G. Hein. The IGEX-98 campaign: Highlights and perspective. In Schwarz, K.-P., editor, *Geodesy Beyond 2000, The Challenges of the First Decade*, volume 121 of *International Association of Geodesy Symposia*, pages 22–25. Springer Berlin Heidelberg, 2000. doi: 10.1007/978-3-642-59742-8_4.

Wilson, B. C., C. H. Yinger, W. A. Feess, and C. C. Shank. The broadcast interfrequency biases. *GPS World*, 10(9):56–66, Sept. 1999.

Wooden, W. H. NAVSTAR Global Positioning System: 1985. In Goad, C., editor, *Proceedings 1st International Symposium on Precise Positioning with the Global Positioning System, Vol. 1*, pages 403–412, US Department of Commerce, Rockville, Maryland, 1985.

Wu, J. T., C. Wu, G. A. Hajj, W. I. Bertiger, and S. M. Lichten. Effects of antenna orientation on GPS carrier phase. *Manuscripta Geodaetica*, 18:91–98, 1993.

Wübbena, G. Software developments for geodetic positioning with GPS using TI 4100 code and carrier measurements. In Goad, C., editor, *Proceedings First International Symposium on Precise Positioning with the Global Positioning System*, pages 403–412, US Department of Commerce, Rockville, Maryland, 1985.

Wübbena, G., M. Schmitz, F. Menge, V. Böder, and G. Seeber. Automated absolute field calibration of GPS antennas. In *Prodeedings of the ION GPS-00*, Salt Lake City, Utah, 2000.

Zielinski, J. B. Covariance in 3D networks resulting from orbital errors. In *Lecture Notes in Earth Sciences, GPS-Techniques applied to Geodesy and Surveying*, pages 504–514, Berlin, 1988. Springer-Verlag.

Zumberge, J. F., D. C. Jefferson, M. B. Heflin, and F. H. Webb. Earth orientation results from the Jet Propulsion Laboratory using GPS. IERS Technical Note 17, Sept. 1994.

List of Abbreviations

AC	analysis center
ACC	analysis center coordinator
AIUB	Astronomical Institute of the University of Bern
ANTEX	ANTenna EXchange format
ARP	antenna reference point
AS	Anti-Spoofing
AWG	Analysis Working Group
BeiDou	Chinese GNSS
BGS	British Geological Survey
BKG	Bundesamt für Kartographie und Geodäsie
BPE	Bernese Processing Engine
BSW	Bernese GNSS Software
C/A-code	Coarse-Acquisition or Clear-Access code
CB	central bureau
CBIS	Central Bureau Information System
CC	cross-correlation
CHAMP	CHAllenging Minisatellite Payload
CDDIS	Crustal Dynamics Data Information System
CDMA	Code Division Multiple Access
CIGNET	Cooperative International GPS Network
CIP	Celestial Intermediate Pole
CMC	center of mass corrections
CODE	Center for Orbit Determination in Europe
CoM	center-of-mass
CoF	center-of-figure
COSPAR	Committee on Space Research
CPF	Consolidated Prediction Format
CRD	Consolidated Laser Ranging Data Format
CSTG	Coordination of Space Techniques for Geodesy and Geodynamics
DCB	Differential Code Bias
DCPB	Differential Code-Phase Bias
DLR	Deutsches Zentrum für Luft- und Raumfahrt
DOMES	Directory of MERIT Sites

DORIS	Doppler Orbitography and Radiopositioning Integrated by Satellite
DoY	day of year
ECMWF	European Centre for Medium-Range Weather Forecasts
ECOM	Empirical CODE Orbit Model
EDC	EUROLAS Data Center
ERP	Earth rotation parameter
EOP	Earth orientation parameter
EPN	EUREF Permanent Network
ESM	extended slab model
EUREF	IAG Reference Frame Sub-Commission for Europe
ETH	Swiss Federal Institute of Technology, Zurich
FARA	Fast Ambiguity Resolution Approach
FDMA	Frequency Division Multiple Access
GFZ	Helmholtz-Zentrum Potsdam Deutsches GeoForschungsZentrum
Galileo	European GNSS
GLONASS	Глобальная навигационная спутниковая система: Global Navigation Satellite System
GHOST	GPS High-precision Orbit Determination Software Tools
GIM	Global Ionosphere Model
GMF	Global Mapping Function
GNSS	Global Navigation Satellite Systems
GOCE	Gravity and steady-state Ocean Circulation Explorer
GOP	Pecný Geodetic Observatory
GPT	Global Pressure Temperature model
GPS	Global Positioning System
GRACE	Gravity Recovery And Climate Experiment
GUI	Graphical User Interface
HOI	higher order ionosphere
IAGA	International Association of Geomagnetism and Aeronomy
IAPG	Institut für Astronomische und Physikalische Geodäsie, Technische Universität München
IAU	International Astronomical Union
ICRF	Inertial Celestial Reference Frame
IERS	International Earth Rotation and Reference Systems Service
IFB	inter–frequency bias
IGEX	International Glonass EXperiment
IGLOS-PP	International GLONASS Service Pilot Project
IGN	Institut national de l'information géographique et forestière
IGRF	International Geomagnetic Reference Field

IGS	International GNSS Service
ILRS	International Laser Ranging Service
IONEX	Ionosphere model EXchange format
IPG	Institut für Planetare Geodäsie, Technische Universität Dresden
ISB	inter–system bias
IRV	Inter-Range Vector
ITRF	International Terrestrial Reference Frame
ITRS	IERS Terrestrial Reference System
JPL	Jet Propulsion Laboratory
JPL DE	Jet Propulsion Laboratory Development Ephemeris
KASI	Korean Astronomy and Space Science Institute
L2C-code	L_2 Civil code
LAGEOS	LAser GEOdynamic Satellite
LARES	LAser RElativistic Satellite
LC	linear combination
LEO	Low Earth Orbiter
LLR	Lunar Laser Ranging
LRA	Laser Retroreflector Array
LOD	length of day
MGEX	Multi-GNSS EXperiment
MJD	Modified Julian Day
MSTID	Medium-Scale Traveling Ionospheric Disturbance
NAVSTAR GPS	NAVigation Satellite Timing And Ranging Global Positioning System
NEQ	normal equation
NGS	National Geodetic Survey
NP	Normal Point
NRT	near–real time
PCF	Process Control File
P-code	Precise or Precision code
PCV	phase center variations
POD	precise orbit determination
ppm	parts per million
PID	process ID in the Bernese Processing Engine
POD	precise orbit determination
PPP	Precise Point Positioning
PRN	pseudo-random noise
QIF	Quasi–Ionosphere–Free
QZSS	Quasi-Zenith Satellite System

RGB	range bias
RINEX	Receiver INdependent EXchange format
RMS	root mean squares
ROB	Royal Observatory of Belgium
RRE	Residual Range Error
RTCM	Radio Technical Commission for Maritime Services
SA	Selective Availability
SLR	Satellite Laser Ranging
SINEX	Solution INdependent EXchange format
SIO	Scripps Institution of Oceanography
SIP	Stochastic Ionosphere Parameter
SLM	Single-Layer Model
SVN	satellite vehicle number
swisstopo	Swiss Federal Office of Topography
TEC	Total Electron Content
TECU	TEC Units
TID	Traveling Ionospheric Disturbance
TT	Terrestrial Time
TUD	Technische Universität Dresden
TUM	Technische Universität München
USGS	U. S. Geological Survey
UCAR	University Corporation for Atmospheric Research
VLBI	Very Long Baseline Interferometry
VMF1	Vienna Mapping Function
Y-code	Encrypted P-code
ZPD	zenith path delay

Index of Programs

Index of Program Panels

Bold printed page numbers indicate a figure of the panel.

Index of Keywords