The BIFROST User's Handbook

A description of the BIFROST observing system

12 June 2020

Mikael Lerner

The BIFROST system

The crown jewel of the BIFROST package is the observing system that is used to provide a graphical interface and top-level control software for controlling a telescope or another observing system. However, the package itself is much bigger including software to collect, distribute, monitor and log engineering data from many different subsystems. It also contains a number of auxiliary programs that can be used to display engineering values, examine logs, plot diagrams, prepare source catalogs, download ephemeris data, edit command files, create observation diaries and more.

At Onsala Space Observatory (OSO) there are five different versions of the observing system used to control the 20m telescope, the 25m telescope, the two OTTs (a single system since the two telescopes are identical) and two different radiometers: the CO/O_3 radiometer as well as the H_2O radiometer that are part of the ozone project. There are many commonalities between the five systems, but each of them also have functions that are unique for that particular system.

BIFROST is based on the CIMA control system used at the Arecibo Observatory. Although it has gone through a lot of development, observers at the two observatories would immediately recognize the heritage between the two systems.

The name CIMA could not be kept though since it either is interpreted as an acronym for "*Control Interface Module for Arecibo*" or as the Spanish word for *peak*, which wouldn't make sense in Sweden. BIFROST was chosen in line with the tradition of using names from the Nordic mythology, where Bifrost is the bridge spanning between the Earth and the Heaven. It is thought to represent either the rainbow or the Milky Way. BIFROST is a backronym and stands for *B-something Interface For Radioastronomical Observatory Systems and Telescopes*.

BIFROST has replaced the PEGASUS control system which originated at CFHT and was introduced at OSO in the early 90s.

The purpose of this handbook is to provide a user who is going to observe with BIFROST with an overview of the system and an understanding of the major features of the system and how to utilize them. The 20m telescope system will be taken as an example in most cases since it is the most complicated of the systems and also since that is the system that most external observers will come in contact with. This handbook will not dwell too much in details — for that, you should sit down with BIFROST and use the built-in help functions. In fact, the entire BIFROST system contains almost 2500 help texts explaining each and every feature.

BIFROST features

BIFROST offers a lot of advanced features. This section will give a quick overview of some of the more important ones. For further details, you should read the more detailed descriptions in the following sections.

- *Extensive on-line help:* Each window has a **Help** button which gives a one-page summary of what that window does. You can also right-click on any feature in a window and get a help message for that particular feature. The BIFROST system contains almost 2500 help texts in total.
- *Both interactive and automatic:* BIFROST can be used in fully interactive mode or in fully automatic mode via *command files*. The two modes are fully intermixable and you can switch between these two modes as you wish during your observations. You can do all the things you do interactively with a command file — and some more things like pause due to bad weather. The interactive windows are updated automatically during command file observing to always show the current configuration used.
- *Flexible scripting language:* Command files are written in a simple custom-designed scripting language which offers loops ("`REPEAT`") and conditional checks ("`IF`"). Command files can thus be everything from a few simple lines to perform a routine task to complicated scripts that can run observations completely automatically for days.
- *Customized command file editor:* Sorry emacs-fans, it is highly recommended to use BIFROST's own editor to write and modify command files. The editor does not offer the same advanced editor functions as your favorite editor, but it does offer on-the-fly syntax checking, syntax explanations, descriptive help and examples, which usually are much more valuable features.
- *Off-line and on-line mode:* BIFROST can be run in off-line mode with simulated observations. This can be used for training and testing. The off-line mode can also be used to prepare catalog files, configurations and command files to be used during the real observations.
- *Automatic logging of configurations:* Each configuration used for observation (receiver, IF/LO set-up, backend configuration and so on) is automatically saved in a configuration file. The name of the this file is written to the FITS-header, so you can always load this file into BIFROST and see exactly what configuration was used.
- *Configurable logging:* You have the freedom to select both how much information you want to see in the observation log display as well as the format it is going to be presented in. You can change the information level and the format whenever you want during your observations, and you can also ask for the changes to be applied retroactively, if you want to go back and check something that happened earlier. For the log files produced you have less freedom since they are supposed to always contain all relevant information, but you still have the choice of if you want to include debug messages and to what extent.

- *Error log display:* Instead of scrolling through a long log searching for any errors or warnings that may have appeared during the night, you can open the BIFROST error log display which will just show you those most important messages.
- *Retroactive debug messages:* Enabling logging of debug messages will quickly produce enormously large log files and is not recommended. However, when something goes wrong in a strange way, having those debug messages could be very useful. BIFROST is thus keeping track of all debug messages and if a task fails, then the debug messages generated before the failure will be retroactively added to the log. You thus get debug messages when you need them but not otherwise.
- *Email any error messages:* You can set up one or more email addresses to which BIFROST will forward any error or warning message. You can thus leave observations unattended and still be alerted if something goes wrong.
- *Automatic diaries:* BIFROST has a tool to grab information about FITS-scans, sources, Tsys-values, pointing and focusing results and so on from the logs and to write it up in neat PDF-file known as a *diary*. This is especially handy for getting a summary of what actually happened during a command file observation.

Differences between BIFROST and PEGASUS

This section is only of interest for users who have experience with PEGASUS, the previous control system used for more than a quarter of a century at Onsala. It highlights some of the important differences between the two systems.

- *No more colour accounts:* All BIFROST observations are done from one single account, the bifrost account. there is thus no need to log out and log in to another account when changing project. You just need to restart BIFROST and enter the new project code. Each project instead has its own area under */projects/*`telescope`*/*`project-code`. Note that the button to open a terminal will open a terminal window which starts in that project area.
- *Select your version:* There is usually several versions of BIFROST installed in parallel. There will typically be (at least) three versions: the normal version, an older version to fall back to in case some bug is found in the normal version and a future version which contains new features but probably also new bugs.
- *Prepare yourself off-line:* BIFROST can be installed and run off-line on any Linux computer simulating all aspects of an observation including producing fake spectra. Introductions can thus be performed in the AoD's office and an observer can familiarize himself/herself with BIFROST before getting on the telescope and even set up catalog files, test out command files and save configuration files which then can be used when the real observations start.
- *Use the help:* In PEGASUS most help pages were either non-existent or not very helpful, so you really had to consult the AoD or the written instructions. The BIFROST system contains almost 2500 help texts. Each window has a good one-page summary and then you can right-click on any feature in any window and get more specific help for that particular feature — so if in doubt, right-click!
- *Looking for the right button:* The PEGASUS menu was a bit hard to navigate since it contained a large set of buttons. In BIFROST, only the most important ones are kept in the main menu. The rest has been moved to a secondary menu called **More utilities**. A few windows are also accessed through another window; for example, the antenna limits window is accessed via a button in the antenna control window.
- *Nothing happens until you press* **Accept***:* This philosophy applies to many windows in BIFROST as well (the ones that have **Accept** and **Dismiss** buttons), but there are also windows with more direct control, for example, antenna control that only has buttons for direct action. Notice also that error checking of parameters is done on the fly and buttons like **Accept** are disabled as soon as you type a bad option. The **Cancel** button is in BIFROST called **Dismiss**.
- *No* **Save values** *or* **Defaults** *buttons:* The **Save values** in PEGASUS is a potential disaster since it can fool you to believe that the system is in a different configuration

than it actually is or even affect an ongoing observation without you knowing it! There are no **Save values** buttons in BIFROST, since you are supposed to save your values to "*configuration files*".

- *No ".par-files":* BIFROST does not have any "*.par-files*", so how do BIFROST remember what you selected between sessions? *It doesn't!* BIFROST always starts up with fixed default values (well, it actually does remember the last used frequency for each receiver and it can be told to remember the position of the windows on the screen), but the idea is that you should save each set-up you do as a "*configuration file*". The next time you want the same configuration you just load that configuration file instead of doing manual set-ups.
- *One thing at a time:* While PEGASUS allows you to do several things in parallel (and screw up things by sending conflicting commands), BIFROST insists on doing one thing at a time. You can thus not send a new command to BIFROST until it has finished the previous one, with a few exceptions: antenna movement is a non-blocking command.
- *Oops, I forgot to configure the backend:* In PEGASUS you could easily start an observation having forgotten some important step in the set-up. That is not possible in BIFROST since you can't even open the observing menu until you have done a proper set-up.
- *Rats, the VLBI people forgot to switch off the phase cal:* There is no longer a risk that equipment you may not be aware of is left on. In spectral line mode, BIFROST won't show you any button for the phase cal, but each time you configure the receiver IF/LO set up, commands will be sent to switch off all phase cal units.
- *No error detection:* There are lots of error conditions that PEGASUS just reports and then keeps on going. It has happened that observations have been running for hours after a serious error with the system just recording crap and the observers believing that everything is fine. BIFROST, on the other hand, is very picky and will abort as soon as it encounters anything that is not behaving as it expects.
- *One observing menu:* In PEGASUS there are different menus for single observations, maps and for pointings/focusings. In BIFROST there is just one menu: the BIFROST observing menu. Within this menu you then select which *observing scheme* (calibration, pointing, single) you want to use. Note that the observing schemes don't share variables, so even if several observing schemes have parameters with the same name, BIFROST will remember (during the session) the different values used for different observing schemes.
- *Only one permanent status window:* In PEGASUS one or several small status windows pop up (and grab focus) and go away while you are observing. In BIFROST there is only one status window (the observation status window) and it stays there permanently. It is not used just for observations but also for other activities such as configurations or power adjustments.
- *Automatic alarms:* It is not possible to forget to start the alarm window in BIFROST, since the alarms are an integrated part of the system. There is no alarm window, but there is a window where you can modify the action BIFROST should take upon

encountering different kinds of alarms. That window is opened from the **More utilities** menu.

- *Automatic VLBI daemon:* The VLBI-daemon is an integrated part of the BIFROST executive, so there is no need to explicitly switch it on as in Pegasus — BIFROST will automatically start to listen for connections from FS's as soon as you start up in VLBI-mode.
- *No flooded home directory:* Pegasus puts all the FITS-files in the project home directory, which becomes a bit cumbersome when you start having 50 000 spectra. In the BIFROST project area, you have a directory called *Data* in which BIFROST creates a subdirectory for each UTC-day. Each daily directory then has a set of subdirectories (e.g. *Calibrations*, *Data*, *Pointings*) where the different FITS-files are written.
- *No preallocated FITS-scans:* Pegasus preallocates FITS-numbers and opens new FITS-files when an observation is started, which means that these FITS-numbers are lost if the observation is aborted or crashes and you will have some incomplete FITS-files among your data. BIFROST performs the observation first and doesn't allocate FITS-numbers until it has a complete data set ready for writing. Intermediate results are still available for viewing in the quick-look facility but they will be stored in temporary FITS-files on */tmp*.
- *New FITS-file codes:* While Pegasus uses a single letter attached to the FITS-number to indicate what type of spectrum it contains, BIFROST uses a wider selection of letters as well as combinations of several letters. The idea is to be able to uniquely identify all types of spectra from the filename. A "*pd*"-FITS-file is thus the unique code for a pointing spectrum in the down position, while an "*rc*"-file is a reference spectrum for the center position.
- *Raw data both for signal and reference:* While Pegasus saves an unprocessed reference spectrum in an "*r*"-file, the signal spectrum is processed and calibrated before being saved to an "*s*"-file thus making it impossible to reprocess the raw data. If asked to save raw data (that is a user option), BIFROST will save the two raw spectra to an "*r*"-file and an "*s*"-file (or an "*R*"-file and an "*S*"-file for calibrations) while the processed spectrum will be saved in another file, typically called "*d*" (for data).
- *Clip the spectra:* In BIFROST you don't need to save the entire spectra. If the edges are horrible, you can configure the backend to automatically clip as many channels as you want from either end of the spectrum. This can also be used to keep down the size of narrow-line spectra like spectra used for pointing which don't need to contain vast amounts of bandwidth which is never going to be used — clipping pointing spectra also reduces the need to fiddle around with zooming in the quick-look facility.
- *Keep adding data to your averages:* In Pegasus you can form averages by adding together individual spectra from each observing loop. However, as soon as you change to another source or another set-up, you have to remember to clear the averages, otherwise you start to add together pieces of data that have nothing to do with each other, and there is no way you can go back and add more data to the same average after that. Don't worry about that in BIFROST — the system keeps track of sources

and set-ups, and makes sure that only data that should be added together is averaged, so you can switch back and forth between sources and still get all data taken for each particular source written to its particular average file.

- *Flexible jobs:* While the PEGASUS job scripts were fairly inflexible and had to be set up for just one type of observation: single, map or pointing/focusing, BIFROST command files have no such limitations. It is a straight-forward scripting language where you can mix different observing modes with different source catalogs and different receiver or frequency settings as you wish.
- *Jobs are for everyone:* Traditionally PEGASUS jobs have only been used for pointing runs and by some very experienced observers. It is expected that it is going to be more common that also inexperienced observers will use command files for parts of their observations. "`IF`"-statements and "`REPEAT`"-statements offer flexibility to write efficient command files.
- *Pointing and focusing:* While PEGASUS allows you to set up pointing and focusing as one single observation, BIFROST considers them as two independent observing schemes. For interactive observations that shouldn't be a problem, but maybe we want to be able to combine them in command file observing.
- *Quick-look instead of grapher:* The PEGASUS **grapher** is replaced with the quick-look facility (which also can be run in stand-alone mode to look at old spectra). The two programs are not similar. For example, the function to calculate RMS-values is not yet implemented in the quick-look facility. However, the quick-look do offer much more flexibility in designing your own lay-out with up to 3×3 plot windows and it automatically switches between lay-outs depending on the data type (calibration, single spectra, pointing).
- *Skip the paper log sheets:* Many PEGASUS observers use preprinted log sheets to note objects observed, scan numbers, Tsys-values, pointing results and breaks due to bad weather. BIFROST comes with a tool that can extract that kind of information from the logs and turn it into nicely formatted PDF-documents (called *diaries*).

BIFROST fundamentals

This section describes some fundamental things like the software parts used, the different observing methods and how the BIFROST windows are designed.

BIFROST software parts

When observing with BIFROST you are actually running five different programs (there could be even more running if you open some auxiliary programs), three of which you will interact with and two which are running in the shadows. Those five components are listed here:

- **Frontend:** This is the program that is responsible for getting the instructions of what to do from you. This program runs the main menu and all other windows where you set up parameters or send commands to do things. However, it doesn't perform any actions — instead they are all sent to the **executive**.
- **Executive:** This is the program that performs all the action. It runs in the background and receives all its instructions from the **frontend**.
- **Feedback:** This program handles the feedback in the form of the observing log window, the observation status window as well as other monitoring windows.
- **Logger:** This is a small program that handles the log messages. All log messages are sent to the logger (the vast majority coming from the **executive**), which formats them and decides which ones should be written to the log file, which ones should be sent to the **feedback** for display in the observing log window and which ones should be ignored.
- **Quick-look:** The quick-look facility is a program which shows you the spectra that have been taken. It is a separate program since it can also be launched in stand-alone mode to look at data in old FITS-files. (This program is not available in the OTT-version of BIFROST since the OTTs don't produce any spectra).

As an observer, you don't need to understand this system, but it's good to know that the different tasks are divided up like that when you see messages like "`the executive is busy`".

The main reason for splitting up the tasks in different programs is to ensure that the main menus are always responsive and don't *hang* because BIFROST is busy doing something.

Command execution

You issue commands to BIFROST by clicking on action buttons in the graphic windows belonging to the **frontend**. The frontend doesn't do anything itself and it will just pass on the command to the **executive** for execution. The **executive** only processes one command at a time, so you will have to wait for one command to be finished before you can send the next one. (The one exception to this rule is that you can send a command to move the telescope and then continue doing something else while the telescope is moving.)

While the **executive** is busy, it will become unresponsive and all action buttons in the **frontend** will be disabled. If you in some way still manage to send another command to the **executive**, it will reject the command with the "`the executive is busy`" message. The unresponsiveness is not 100% though since there are still a few commands that the **executive** will accept, for example, commands to abort or stop what it is doing or status requests will still be processed.

BIFROST versions

The BIFROST observing system as well as all other BIFROST programs are usually available in several different versions, which each have a unique version code. A version code has the format "`N.N.NN`" where "`N.N`" is the actual code and the final "`NN`" is a two-digit number that is incremented each time there is a software modification. Official versions always have fully numeric version codes, while test and development version usually use letters as well (for example "`3.2.xx`").

Each version is also assigned a "*version name*" which is a single word. It is important to note that the version name won't stay the same — it changes during the life-time of a version. A new version with new features may initially be called "`future`" while it is being commissioned, then being renamed to "`normal`" when it becomes the new standard version, and later being renamed to "`old`" when it gets replaced by a newer version.

The idea is that the version name "`normal`" always should refer to the currently recommended version and "`old`" should refer to the previous version, which you may want to use if you discover that there has been some bugs introduced in the newer version. You use the version name when you select which BIFROST version to use.

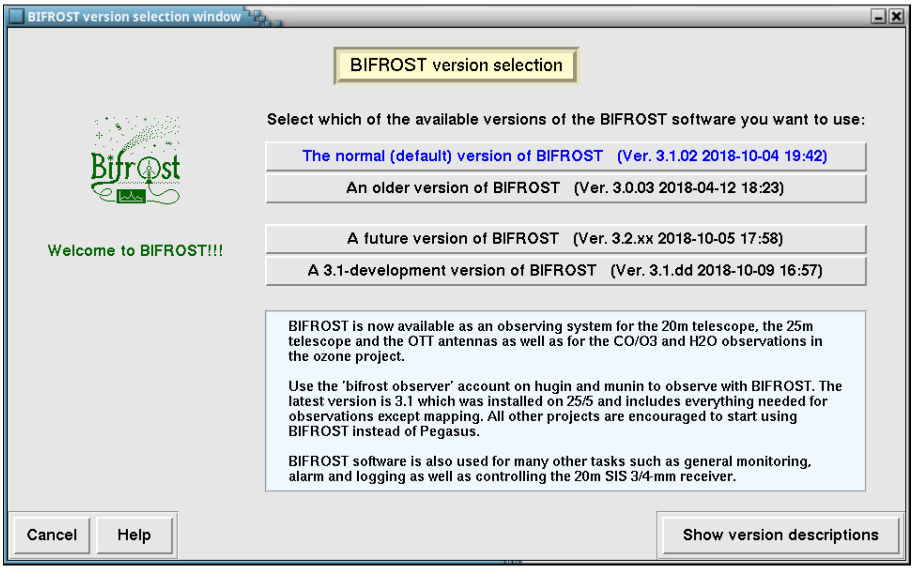
Version selection

When starting a BIFROST observing session of any other BIFROST program, you actually first pass through the "*BIFROST selector*" which selects which version to start for you. If nothing is specified, the selector always tries to give you the "`normal`" version. You can override this by sending an instruction to the selector via an argument — such arguments should be preceded with double "`-`" signs, since arguments with a single "`-`" sign will be passed on to the program that will be started.

The "`--X`" argument is the most useful, since it asks the selector to open the graphic BIFROST version selection menu, where you can click on the version you want to use. The "`--V`" option will also present all the different versions for you to choose from, but it will just do it with text in the terminal window. When selecting a version, you only need to type as many characters in the version name as needed to make it unique (which usually is just one). If you already know which version you want to use, you can give the version name directly as as argument with "`--v` *`version`*". Again, you only need to type as many characters as needed to remove any ambiguity.

Observing methods

BIFROST can be run both in interactive mode or in automatic mode using *command files*. BIFROST offers a fully graphic user interface where the user can control



The version selection window where you select which version of BIFROST to use. This menu shows up when you start any BIFROST command if you specify the "`--X`*" option.*

and perform observations in a simple way. For routine observations, BIFROST can run "*command files*" which are scripted observations using a simple, BIFROST-specific script language.

The two modes are fully intermixable and a set-up can be made interactively followed by command file observations or vice versa. The typical usage would be to use interactive mode for testing out things and command files for routine observations.

Command files are simple text-based files containing BIFROST commands with a simple syntax. Command files allow several elements of flow control: there are command structures for conditional execution (IF-statements) and for loops (REPEAT-statements), which both offer flexibility. There are mechanisms for handling errors and restart an observation if possible, as well as keep an observer informed by sending email alerts.

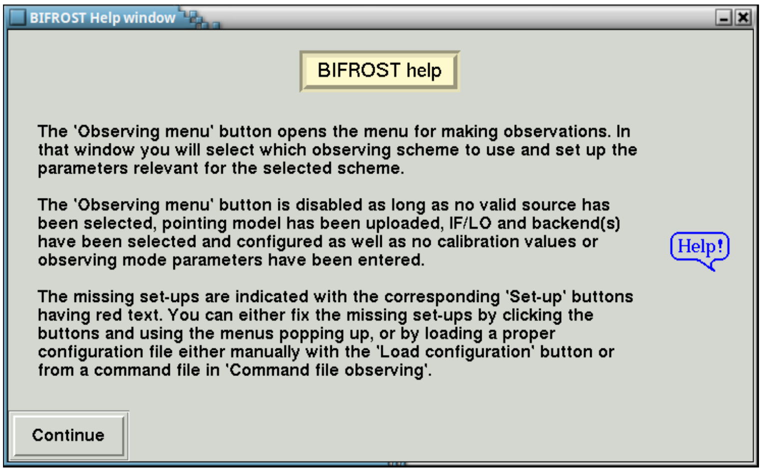
In VLBI-mode, BIFROST can be controlled from a field system computer running a VLBI schedule. This computer will send telescope, configuration and calibration commands via a socket to the BIFROST executive.

In VLBI-mode, it is also possible to control the telescope movements directly from a VLBI schedule in the form of a SNAP file, which BIFROST can execute.

General window design

Almost all BIFROST windows follow a certain design scheme. They have a steel-grey background and a yellowish box at the top with the window title. There are almost always some action buttons at the bottom left and many windows also have some action buttons at the bottom right.

Most windows have a fixed size, but windows which contain some graphic box can often be resized. Some windows that contain list boxes may be resizable only in one



An observer wonders why the observing menu button is disabled. Right-clicking on the button provides the explanation.

direction. Windows that support resizing can be made larger but not smaller than their minimum size.

Getting help

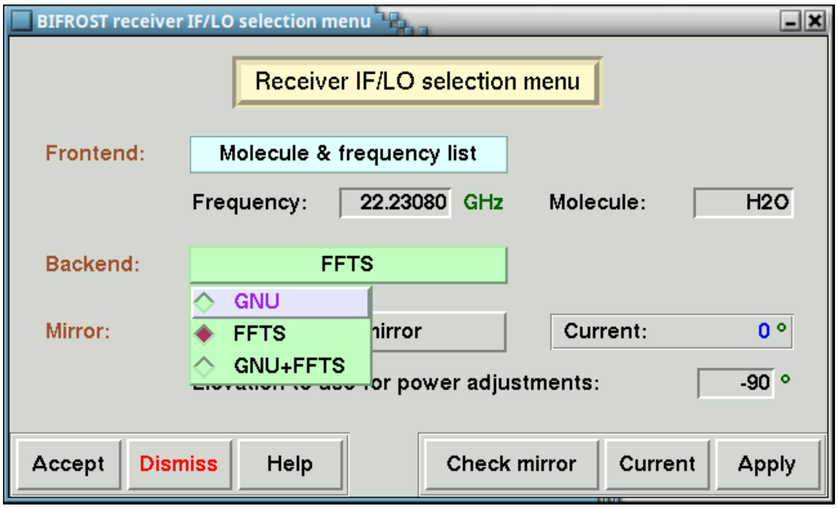
BIFROST comes with extensive on-line documentation. The entire BIFROST system contains almost 2500 help texts explaining each and every feature, and the user is encouraged to use this system to find out about the details of the system.

Almost all BIFROST windows have a **Help** button in the lower left corner which will bring up a one-page help window giving a description of that window. You can also use the right mouse button to right-click on almost any feature in any window and get a pop-up help page with specific help on that feature. For those few features that don't have a specific help page, you will instead get the one-page help, when you right-click on them, so it is always worth right-clicking.

If it is a small and simple window, the one-page help may give you all the information you need but, but if it is a large and complex one, you will only get a broad overview of the window and you will have to get the details by right-clicking on individual features in the window.

To avoid filling the screen with help windows, the help facility only keeps one help window open at a time. If you already have a help window open when you click to get help on something else, the first window will be automatically closed when the new help window is opened.

There is a preference which you can set in the BIFROST observing session preferences set-up window, which controls how the pop-up help windows should behave. The default is that they have a **Continue** button which you will have to click when you want to close the window, but you can change this behaviour so that the help window just stays open as long as you keep the right-mouse button pressed, and disappears automatically when you release the mouse button.



An example of a BIFROST window. In this case, the observer is changing the backend selection with a pull-down menu. The text on the **Dismiss** *button has turned red to indicate that the new selection has not been applied, and the* **Current** *button is enabled to allow the observer to restore the window to show the currently applied configuration.*

Setting options and parameters

Options and parameters can be set by typing in values in entry fields, by selecting an option from a pull-down menu, by marking different options in a multiple option pull-down menu, by selecting something from a selection box, by selecting one of several radio buttons or by pressing a push-button. There may be options and parameters that are not available because of other choices you have made — those ones will still be shown but the are greyed out and can't be set.

When entering something into an entry field, BIFROST will apply on-the-fly error checking. The background will turn red instantaneously if you type something bad. It may also happen that BIFROST automatically removes some illegal character you have tried to type.

Many options are presented in the form of pull-down menus. Those are shown as a *mint-green* box with the selected value. Clicking on the box will open the pull-down menu and you can select a different value. Since the available options may depend on some other option it may happen that you have a value selected that is no longer allowed due to a change of another parameter. In that case the box will turn *red* to indicate that you have to select another value.

Multiple option pull-down menus are used very rarely in BIFROST. They have a menu title in a *light-blue* box. When clicking on the box, a pull-down menu opens, where each option has a check box where you can mark whether that option should be selected or not.

An even rarer form of pull-down menu is the action menu where the different options trigger different actions when clicked on. This menu is using a *light-orange* colour.

A selection box is a box where a (maybe large) number of options are shown. The box has a whitish background and a vertical scrollbar to the right. Selection is done by left-clicking. A typical use of selection boxes is when browsing for files.

Radio buttons are also used sparsely. They consist of two or more normal *grey* buttons with the one button indicating the selected option being pressed down (which is indicated by a lighter tone of grey and the text being shown in *blue*).

A push-button is also a *grey* button which is used to select or deselect something by clicking the button. When clicked, the button has a lighter tone of grey and the text is being shown in *blue*.

There are also action buttons, which have the same appearance as radio buttons and push buttons. Some action is taken when you click on such a button — you can always consult the button's help text if you want to know more about what the button does.

Reported values

In monitors and some other windows values are shown colour-coded. Values are shown with *blue* colour when they are normal, while *orange* colour is used to indicate values that you should pay attention to and *red* colour is used to indicate bad values.

In the monitors, you may also see *yellow* values on *pink* background. Those values are the last receiver values, but they are old values, since there are no new values available.

The buttons at the bottom

All BIFROST windows have a set of buttons at the lower left and some windows also have another set of buttons at the lower right. There are also some windows (like the observation log display) that have a coloured status box in the space between the two sets of buttons.

The buttons on the lower left always include the **Help** button that brings up a one-page help window and button(s) to close the window. The close button is most often called **Close** but on the main menu it is called **Exit** to indicate that it is not just closing the window but actually closing down the entire BIFROST system (with a control question to avoid mistakes). In some windows you will instead have two buttons **Accept** and **Dismiss**. These are BIFROST windows where you set up things but no action is taken until you explicitly ask for it. In this case the **Accept** button will apply those settings and then close the window, while the **Dismiss** button will just close the window discarding any changes you made in the window. The **Accept** button is disabled if BIFROST is busy or if you have selected some bad option in the window. The text on the **Dismiss** button turns red if you have changed something in the window — this is to warn you that these changes will be lost if you dismiss the window.

The buttons on the lower right are used for a wider range of tasks. They are typically used to perform certain actions or open other windows. In some windows that have **Accept** and **Dismiss** buttons, you will have two related buttons in the lower right: **Current** and **Apply**. The **Apply** button does the same thing as the **Accept** button but leaves the window open. It will be disabled for the same reasons as the **Accept** button. The **Current** is used to restore the settings to their current values. It will thus only be enabled if you have modified something in the window without **Apply**ing the modification. The

Current button does the same thing as if you would **Dismiss** the window and then reopen it.

If the window has a coloured status box, then the colouring is changed as follows:

- **Green:** The system is busy observing/working.
- **Yellow:** The system is idle and waiting for new commands.
- **Red:** The last action failed.

Using the mouse buttons (How to click)

Being a graphic program, BIFROST relies heavily on point-and-click operations. BIFROST assumes that you have a mouse with three buttons, but if you only have two buttons (or a touchpad with only two buttons), you can emulate the middle-mouse button by holding down the `<Shift>`-key and then use the left button. The middle-mouse button is the least used button and is only used in some graphic windows and in some text boxes.

- **Left-mouse button:** This is the button used for all actions, clicking on buttons, selecting things in menus and list boxes as well as marking things in graphic windows.
- **Middle-mouse button:** This button is only used in some special circumstances which are listed below and can always be emulated by holding down the `<Shift>`-key together with the left-mouse button.
- **Right-mouse button:** This button is used exclusively for the pop-up help, so wherever the cursor is, you can always right-click and get the proper help text in a pop-up window.

Note that there is an important preference affecting the right-mouse button that you can set in the BIFROST. The default is that the help window popping up comes with a **Continue** button which you have to click when you want to close the window, but you can change this so that the pop-up window only is shown as long as the right-mouse button is pressed down.

The following list contains the usage of the middle-mouse button that a normal observer may encounter:

- In several zoomable windows (the quick-look display, the Tsys display and the map editor in zoom mode) the left-mouse button zooms in and the middle-mouse button zooms out.
- When the map editor is used in edit mode, the left button adds new map positions while the middle mouse removes existing map positions.
- In the quick-look FITS-set selection window, middle-clicking on a FITS-file selects that file for display in the quick-look FITS-header viewer but without displaying the spectrum itself.
- In the observation log display window, the middle-mouse can be used to clear away all earlier lines in the log.
- In the command file observing window, middle-mouse clicks can be used to temporarily comment out lines in a command file.
- In the graphic catalog display window, holding down the middle-mouse button creates a marker which can be used to check positions and rise and set times.
- Middle-clicking on a session with an existing diary in the diary status window will open that diary in a PDF-viewer.

Running BIFROST

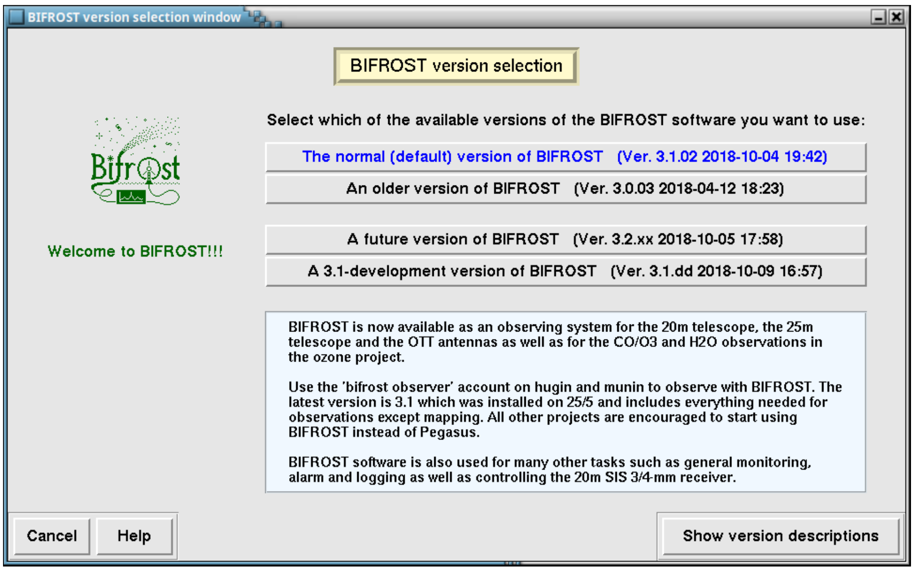
BIFROST is run from the bifrost account (the oper account for the OTT). Observing computers are usually set up to log you in to that account automatically. They may also start up BIFROST automatically for you. There can only be one instance of BIFROST running at any time, except for the ozone project which can have two BIFROST systems running in parallel: one for CO/O_3 and one for H_2O.

You can also run BIFROST off-line on any Linux computer which has the BIFROST package installed. In that case you can run it from any account and you should start it by typing the command "`bifrost --X`" provided that you have the directory */bifrost/bin* in your "*PATH*". On off-line computers, each user can only run one instance of BIFROST, but up to five different users may each run a BIFROST session simultaneously.

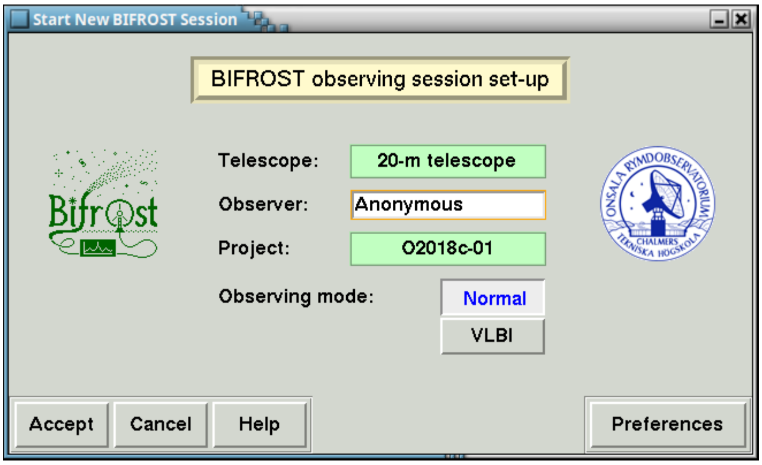
In off-line mode, you can play around and familiarize yourself with BIFROST without wasting telescope time or risking damaging any equipment. BIFROST simulates all action including producing fake spectra to make the experience as similar as possible to real observations. You can also prepare source catalogs, develop command files, set up map files and save configuration files which you can then move to the observing computer and use for your real observations.

Starting a BIFROST session

On an observing computer you can start BIFROST by clicking on the -button in the lower panel. Another possibility is to click on the -button and to get the BIFROST tools menu and there click on the **Start observing session**. If you instead prefer to start BIFROST from a terminal window, you just type "`bifrost`".



The version selection window where you select which version of BIFROST to use. (Version selection works with any BIFROST command if you specify the "`--X`" option.)



The observing session start-up window, where you have to provide your name, specify your project code and select observing mode. The telescope choice is only available when you run off-line sessions.

On an off-line computer you can open the BIFROST tools menu with command "`bifrostwin`" or start BIFROST directly with the command "`bifrost --X`". (The "`--X`" argument tells the system to launch the graphic version selector — this is the default behaviour on-line but not off-line.)

The first window you will see is the BIFROST version selection window where you have to select which version you will be using. Unless told otherwise, you should click on **The normal (default) version of BIFROST**.

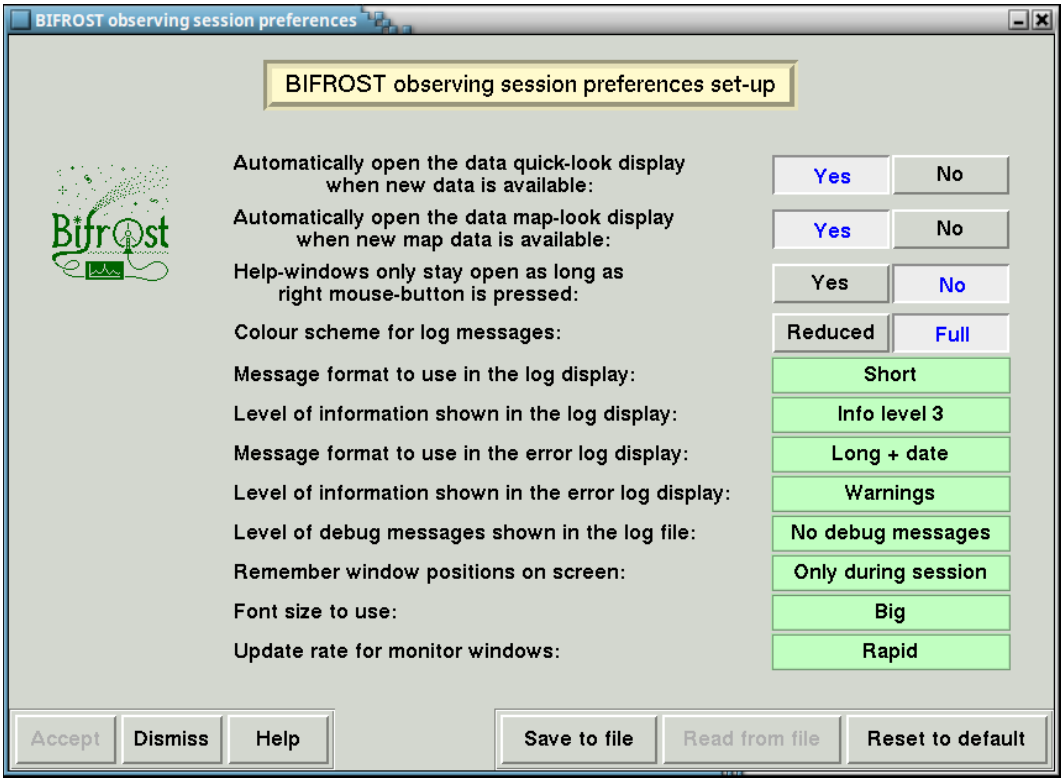
The next window before BIFROST is started is the BIFROST observing session set-up window. Here you need to enter your name and select which project to use with the **Project** menu. You also need to select the observing mode: normal (spectral line) observations or VLBI-observations. When finished, click **Accept** to start the BIFROST observing session.

If you have selected a new project and this is the first time you start up BIFROST for this project, the BIFROST observing session preferences set-up window will pop up, since BIFROST wants you to select your preferences before starting up. If you have time, you can check the different options, but you can also just leave the default selection — you can always change the preferences whenever you want during your session. Click on the **Save to file**-button when you are finished.

The start up process takes a few seconds as the five different programs are started up and set up communication channels between each other. Several windows will open with the last one being the main menu which won't open until BIFROST has verified that all parts are running and talking to each other.

BIFROST observing modes

All BIFROST systems have two radio buttons in the session start-up window which are used to select the observing mode. For the ozone project, the choice is between the CO/O_3 or the H_2O systems, while for all the telescope the choice is between **Normal** and **VLBI**.



This is the default selection of preferences you will see when starting up the first observing session for a new project.

The main difference between **Normal** and **VLBI** mode is that in VLBI-mode, BIFROST is listening to commands from a field system computer running a VLBI schedule. On the 20m and 25m telescopes, the IF/LO set-up looks a bit different between the two modes, since you specify a rest frequency in normal (spectral line) mode and a net-LO in VLBI-mode. This affects how you do pointing tests in VLBI-mode, since you will have to adjust the velocity scale for spectral line sources to compensate for the different frequencies. Also in VLBI-mode you have access to the phase cal and the possibility to run VLBI schedules (SNAP files) directly in BIFROST. Another difference is that BIFROST doesn't apply any Doppler corrections in VLBI-mode.

Exiting a BIFROST session

You click on the **Exit** button in the main menu window. There will always be a control question popping up and asking you if you really want to close down the observing session. There may also be further questions if you want to switch off certain equipment before exiting.

Changing project in BIFROST

You can not change project while you are running BIFROST, since the project environment is set up by the start program when you start up BIFROST. Instead, you just exit BIFROST and start it up again.

Handling a crashed BIFROST session

BIFROST rarely crashes and when it does, the problem is usually that some component of the observing system has been killed or crashed. If your problem is that BIFROST is hanging waiting for the executive to finish some command you have sent or due to some error which occurred, you should read the following chapter about "*Aborting and stopping BIFROST*", which describes how to handle such problems.

In case of a real crash, you will need to run the **clear_bifrost** script to kill the remaining parts of BIFROST and remove all temporary files that BIFROST may be using. You can either do that by opening the BIFROST tools menu with the -button in the bottom panel on the screen and then click on **Kill observing session**, or you can click on the -button to get a terminal window and then type the command "`clear_bifrost`".

Aborting and stopping BIFROST

There are buttons to interrupt something the BIFROST executive is doing located in several windows: in the observation log display, in the observing menu and in the command file observing window. It doesn't matter which window you use, since the buttons do the same thing. The location in different windows is just for your convenience.

The **Abort** button will tell the executive to stop what it is doing immediately. BIFROST may not return the control to you immediately, since there may be clean-up tasks it wants to perform (for example removing an inserted calibration load if asked to abort in the middle of a calibration).

The **Stop** button will tell the executive to stop gracefully. The typical use is when you want to terminate an observation but want BIFROST to finish the current observing loop before stopping.

The **Skip** button is only available when you are running a command file. It tells BIFROST to **Abort** the action of the current line in the command file, but keep the command file running and immediately continue with the next line instead.

If an unexpected error occurs while BIFROST is doing something, the executive may be hanging in a state where it still flags that it is busy, thus preventing you from sending new commands to it. There are two more buttons that can be used to deal with such a situation:

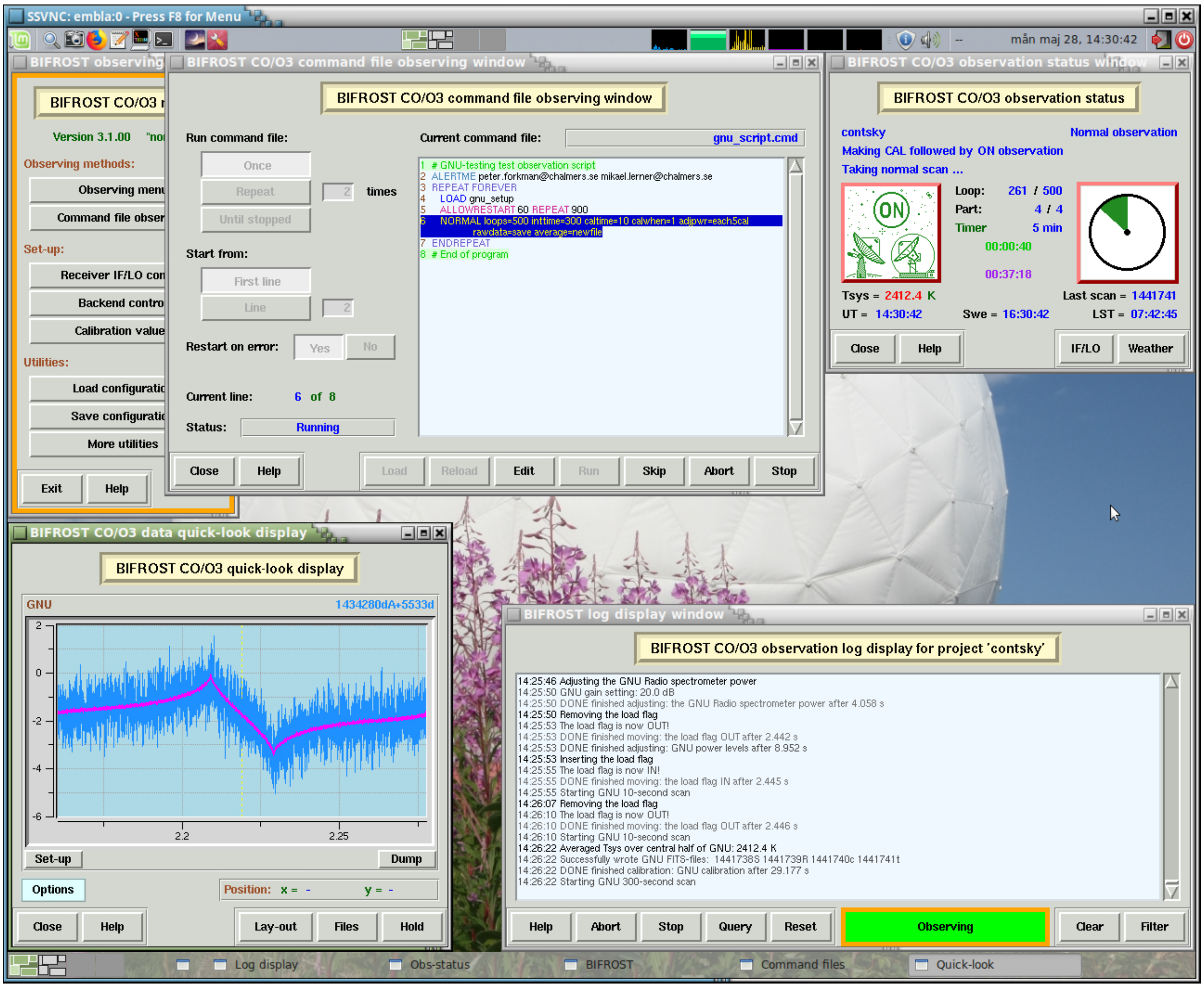
The **Query** button tells the executive to immediately report back what it is doing, or in the case of a crash, what it thinks it is doing. This button can be used at any time to query the executive and it usually produces a list of tasks and subtasks being executed.

The **Reset** button can be used to clear the busy flag if the BIFROST is hanging. You should have tried to use the **Abort** button before resorting to the **Reset** button.

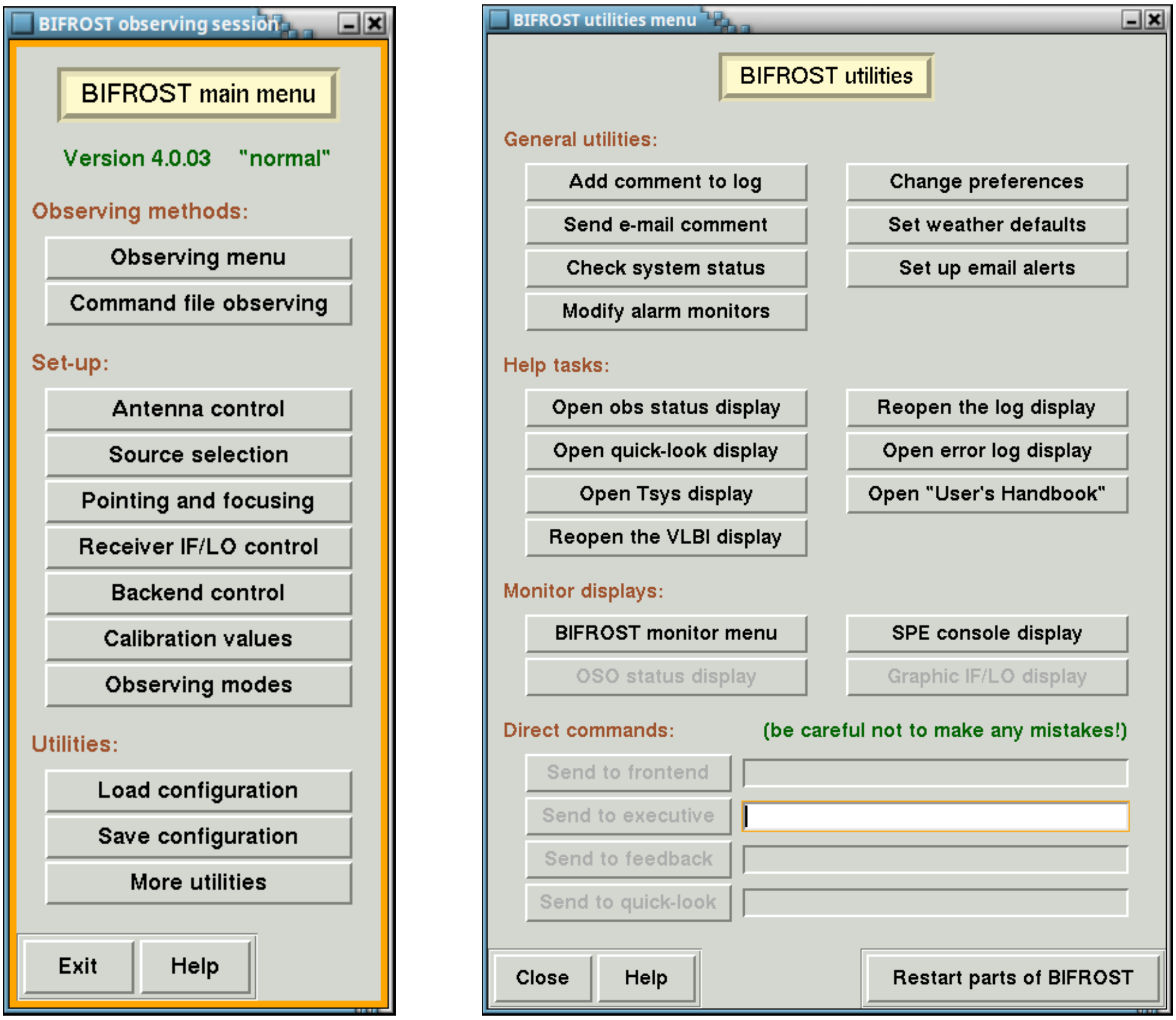
In the unlikely case that some part of the BIFROST system really is hanging or have been killed, you will have to **Exit** BIFROST and restart it. If the **Exit** button doesn't work, you will have to resort to the heavy artillery: either open the BIFROST tools menu and click on **Kill observing session** or open a terminal window and type the command "`clear_bifrost`". This script will remove any trace of a crashed BIFROST session.

The main BIFROST windows

The BIFROST system consists of quite a number of different windows for configuring, running and monitoring an observation. Three windows (four in VLBI-mode) are opened automatically when you start BIFROST, but for a minimalistic approach you only need to have two of them open; all others can be closed. The two mandatory windows are the main menu and the observation log display.



An example view of a BIFROST system in action. This is the system used to control the CO/O_3 radiometer at Onsala Space Observatory. The screen dump was taken during standard observations controlled by a simple command file which can be seen in the big window in the top. The main menu is shown (partly hidden) to the upper left while the observation status window is shown to the upper right and the observation log to the bottom right. The data is shown in the quick-look window at the bottom left.



The BIFROST main menu *and the* BIFROST utilities *menu (which you get by clicking on the* **More utilities** *button) for the 20m telescope.*

The main menu

The main menu pops up at the top left of the screen after a few seconds and it is from here you control your observing session (it may pop up at some other place if you have set the preference telling BIFROST to remember window positions between sessions). It is one of the two mandatory windows which you can't close during an observing session.

At the top of the main menu, you will see the name and code of the BIFROST version you are using. Then there are three groups of buttons which all open different kinds of windows.

The topmost group contains buttons to the windows you will use to perform the observations. When BIFROST is started the **Observing menu** button will be disabled since you haven't configured the system yet. The **Command file observing** button will, however, be enabled since you may want to run a command file that will do all the set-up for you.

The middle group contains buttons to all the windows you need to use if you are configuring the system manually. You can do things in different order, but the recommendation is to go through them one by one starting with the topmost button. Some button may be disabled if it is dependent on another configuration being made first. Also, the text on a number of buttons will be red to warn you that these configurations have not been set up yet. The text on a button may also go red, if a problem develops with that system; for example, if a backend produces an error during operation, BIFROST will abort the observation and flag the backend as *unconfigured*.

The third group contains three buttons: two dealing with configuration files and one, **More utilities**, which opens a utility window with a lot more buttons (that didn't fit into the main menu). The configuration file buttons are intended to save you a lot of the trouble of going through all the buttons in the middle section. The idea is that once you have configured the system, you use the **Save configuration** to save the configuration to a file, and the next time you want to use the very same configuration, you use the **Load configuration** button to restore it.

The **Exit** button in the main menu is the only way to terminate the observing session in a nice way. There will always be a control question if you really want to quit, in case you clicked the button by mistake. There may also be some question whether you want to switch off certain equipment (in case you forgot) or leave it on.

The utilities menu

To keep the main menu simple, a lot of less essential functions have been put into a second menu called the BIFROST utilities. You can open it with the **More utilities** button at the bottom of the main menu and it contains a lot of buttons to open other windows. The window is divided in four sections, but the exact lay-out will vary between the different observing systems.

The first part contains a set of useful utilities. **Add comment to log** can be used to add comments to the observing log as well as to the observing diaries. **Change preferences** is used to modify the preferences for the BIFROST observing system. **Send email comment** allows you to email comments to the BIFROST administrator, typically reporting some bug or feature you have discovered. **Set weather defaults** is used to set weather defaults to be used in the emergency situation that no real weather data is available. **Check system status** can be used to poll the configuration and status of the hardware. **Set up email alerts** is used to specify one or more email-addresses to which BIFROST will forward any error or warning message that appear in the log. The **Modify alarm monitors** are only available on the 20m and 25m telescopes and is used to specify how BIFROST should react if certain hardware problems occur (e.g. problems with the antenna, cryo system or receiver phase-lock). In VLBI-mode, you can use the **Reopen the VLBI display** button to reopen the VLBI status display, if you have closed it. On the CO/O_3 and H_2O systems, there are also buttons here to control the elevation mirror and the load flag.

The second part contains buttons that open (or reopen) various of the BIFROST windows. The **Open obs status display** reopens the observation status window in case you have closed it. The **Reopen the log display** can be used if the observation log display for some reason get killed. The **Open quick-look display** can be used to open the quick-look facility, although this should rarely be needed since there is a preference which normally forces BIFROST to automatically open it if it is closed and a new spectrum is available. The **Open error log display** can be used to open the error log display which is similar to the observation log display except that it only shows error and warning messages. The **Open Tsys display** (not available for the OTTs) opens the Tsys display which is graphic window showing the measured Tsys-values in a chart recorder-like diagram. The button **Open "User's Handbook"** opens this manual in a PDF-viewer, while the **Reopen the VLBI display** (only available in VLBI-mode) reopens the VLBI status window.

The third part contains buttons to open some monitor display. Other buttons in this section may open non-BIFROST windows such as console displays for backends. In some systems there are also currently some non-functioning buttons as place-holders for yet-to-be-implemented monitor windows.

The fourth part contains boxes where you can type in Tcl/Tk commands and send them to different parts of the observing system. This is dangerous stuff unless you know what you are doing and make sure that you don't mistype any command, so it is rarely needed for normal observers. If it would be needed you will be told exactly what to do.

The **Restart parts of BIFROST** button will take you to the BIFROST reload and restart menu where you can reload and restart parts of the BIFROST observing system. This is not recommended for normal observers, since it is safer to just exit and restart the BIFROST observing system. The window is, however, very useful to staff when debugging and modifying the BIFROST software.

The observation log display

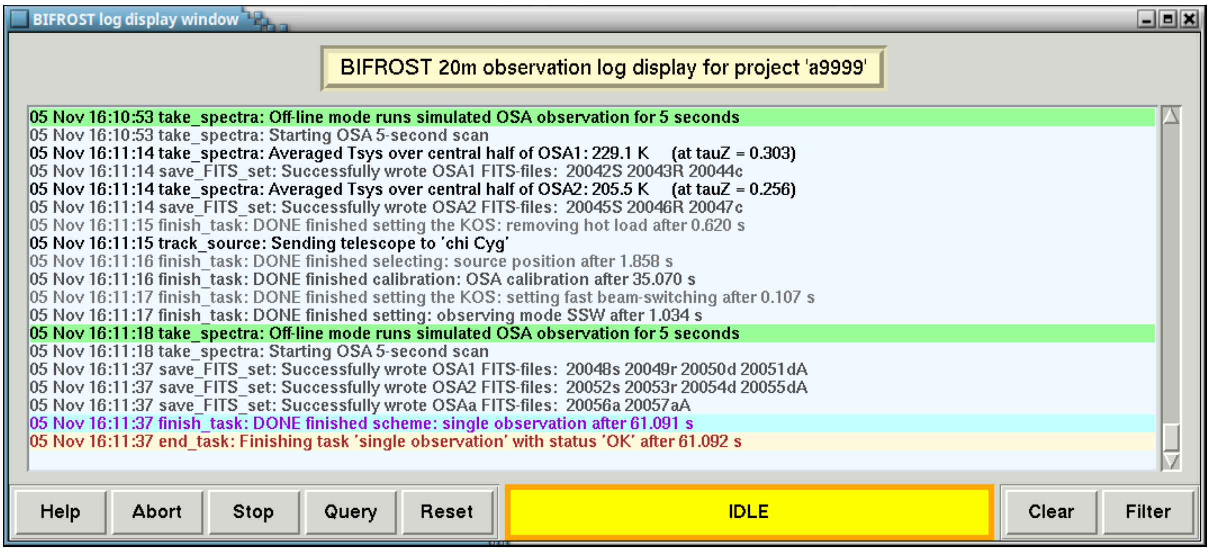
The observation log display pops up at the bottom right corner of the screen (if you haven't moved it and set the preference telling BIFROST to remember window positions between sessions). It is one of the two mandatory windows which you can't close during an observing session.

The window contains a big box where all the log messages from the observing session appear and a bottom line with buttons and a status field.

All log messages that have a higher importance than a selected importance level will be shown in the big log box. Both the importance level threshold and the format used to display the messages are selected with the preferences. They can also be modified from the window that you open with the **Filter** button. If you scroll upwards in the log and new messages are added, then the window jumps to the bottom where the new messages have been added, so that you don't miss them. This behaviour can be blocked by holding or just leaving the mouse cursor over the scrollbar.

The status field between the two sets of buttons is either yellow showing the word "`IDLE`" or green showing a verb indicating what BIFROST is doing (e.g., "`Configuring`", "`Calibrating`" or "`Observing`"). The green colour indicates that BIFROST is busy and won't accept any new commands, so you can only send new commands when the status colour is yellow. (The colour scheme is inherited from Arecibo where green means that you are using the telescope, which is good, and yellow means that you are sitting idle, which means that you are wasting the taxpayers' money.)

The left suite of buttons contains the mandatory help page as well as buttons for interrupting och checking what the BIFROST executive is doing. The **Abort** button is



The observation log display is the window where the log is displayed.

used to tell the executive to stop what it is doing immediately, while the **Stop** button tells BIFROST what it currently is doing at a convenient time. The **Stop** button is typically only used during observations and BIFROST will then finish the ongoing observation loop before terminating the observation gracefully.

The **Abort** button is the one to use, if you think that BIFROST for some reason is hanging. In such a situation, you should, however, first use the **Query** button to ask the executive what it thinks it is doing. If the **Abort** button doesn't help and the executive stays in a busy state, then you have to resort to the **Reset** button which tells the executive to abandon whatever it thinks it is doing and reset itself to an idle state.

If the executive already is idle, then the **Abort**, **Stop** and **Reset** buttons won't do anything. The **Query** button you can use whenever you want — it will report a list of the task and subtasks the executive currently is performing.

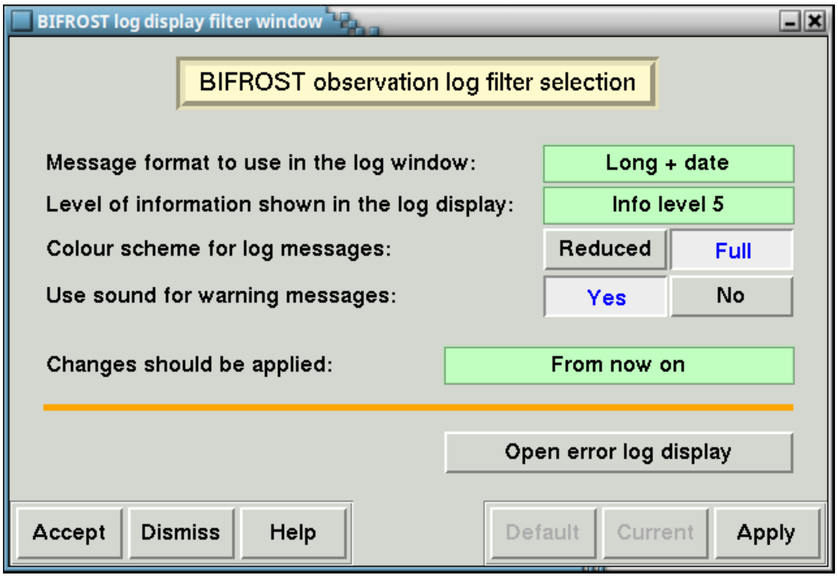
The two buttons to the right deals with the log display itself. The **Clear** button removes all messages from the log box. You can also remove log messages directly in the log box by middle-mouse clicking on a line. All lines above the selected line will then be removed from the display log.

The last button is the **Filter** button which opens the observation log filter selection window. From that window you can change the format as well as the amount of log messages shown in the log display. You can also apply this change retroactively: you can thus restore log messages that you cleared away with the **Clear** button or add more detailed messages if you ran into some kind of trouble. Further details are given in the subchapter below.

You can reopen the observation log display with the **Reopen the log display** in the **More utilities** menu, if it happens to disappear.

The observation log filter selection window

The observation log filter selection window is opened with the **Filter** button in the observation log display. With this window you can modify which message types ("*levels*")



The observation log filter selection window is used to change the format or the amount of messages shown in the observation log filter selection *window.*

you want to see in the observation log display as well as the format used to display them. You also have to select from when the change should be applied. If you select **From now on** BIFROST will keep the log as it is but start using the new format or level from now on (once you have clicked **Accept** or **Apply** to apply the change). However, this menu also gives you the option of applying the modification retroactively. In that case, BIFROST will clear the log being displayed and then rescan the observing log file and fill in the log in the new format or with the new level.

This feature can thus be used if you want to go back and see more detailed messages, maybe to understand a problem you have encountered. You could then change the message level and set the **Changes should be applied** a few minutes back in time.

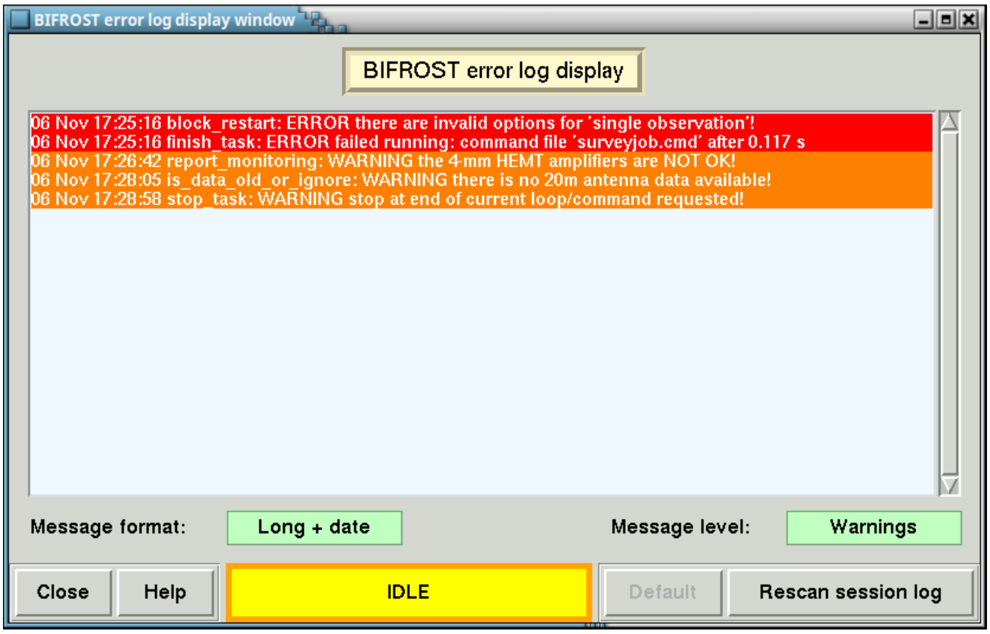
There is also an option for sounds for warning messages which is a relic from Arecibo and which currently doesn't do anything since none of the BIFROST systems use sounds.

The default values for the message format, the message level and the colour scheme are taken from the preferences you have selected. If you do modify them, the **Default** button will be enabled and clicking on it will restore the selection to the selection in the preferences.

Below the orange line there is a direct-action button which is used to open the BIFROST error log display.

The error log display

The BIFROST error log display is an alternative log display which only shows you errors, warning or other important messages without the need to scroll through a large log. You can open it either from the observation log filter selection window (via the **Filter** button in the observation log display) or from the BIFROST utilities window (via the **More utilities** in the main menu).



The error log works like the observation log display *except that it only shows error, warnings and important messages.*

The window contains a big box with the messages, and below the big box there are two pull-down menus to select what message format to use and what types of messages ("*message level*") to show. The choice is between **Errors**, **Warnings**, **Problems**, **Alerts**, **Notes** and **Commands**, which is the order of decreasing importance. All messages more important than the level you choose will also be displayed; thus if you select **Errors**, only errors will be shown, while all six message types will be shown if you instead select **Commands**.

Note that you will have to use the **Rescan session log**, to update the window if you change either the format or what message types to include!

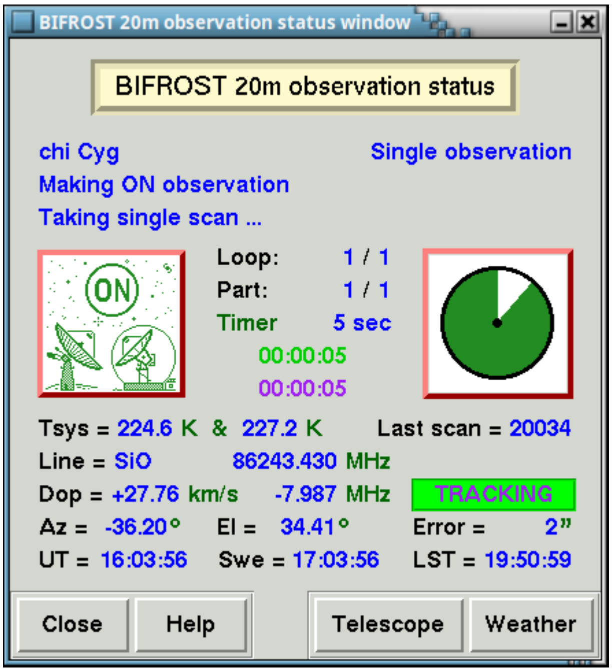
There is a status box in the bottom row just like in the observation log display. However, this status box is just telling you whether BIFROST is busy rescanning the session log and updating the error log display.

The default values for the message format and the message level are taken from the preferences you have selected. If you do modify them, the **Default** button will be enabled and clicking on it will restore the selection to the selection in the preferences.

You can leave the BIFROST error log display open permanently or just open it when needed, as you wish.

The observation status window

The observation status window pops up at the top right corner of the screen (if you haven't moved it and set the preference telling BIFROST to remember window positions between sessions). This window provides you with a feedback of what is going on both



The observation status window for the 20m telescope during a single observation. The window provides a lot of information about what is going on.

with text messages and using graphical icons. It also displays some important information such as time, the last measured Tsys-values and the last used FITS-number. The information varies depending on which BIFROST system is being used.

The central part of this window consists of two square boxes and five short lines in between. The left box is an icon box that displays an icon illustrating what is going on. The right box is a graphic clock, that either is a timer counting down the seconds using a pie chart until some action is finished. If the expected time is not known, then the graphical clock instead flashes to indicate that it is active.

The icons are shown in different colours depending on what kind of activity they represent (configuring, slewing, observing etc.). Yellow icons indicate that the executive is idle and ready to receive new commands. The standard idle-icon is a solid yellow box with the text "IDLE". When run in VLBI-mode, there are two alternative idle-icons: a telescope icon with the text "VLBI" in the sky, which indicates that the executive is idle but it is connected to an active FS (the activity is determined by the periodic tracking queries that a FS sends during a VLBI observation). There is a similar idle-icon with the "SNAP" instead, which indicates that there is an active SNAP file being executed locally by BIFROST (this is used for example for satellite tracking).

The five lines between the two boxes are only used when needed. The first line shows the total number of loops to perform and which loop that is currently being performed. The second line shows the total number of parts in the current loop and which of those parts that is currently being performed. As an example, an observation loop may consist of five parts: a power adjustment, a calibration-ON and a calibration-OFF spectrum followed by a reference data measurement and a signal data measurements.

The third line tells you how long time the current action is supposed to take, provided that that is known (for example “`Timer 30 sec`”); otherwise the word “`Waiting ...`” is shown. This is the information that controls the graphic clock in the box to the right. It also controls the digital clock on the fourth line, which either counts down how many seconds remain (green digits) or count up how many seconds have passed (orange digits).

The fifth line is also a digital clock. It counts the total time elapsed since the current task was started and it is shown with purple digits.

Above the boxes there are three lines. The first line tells you the name of the current source (for telescopes) or the name of the project (for the radiometer systems) as well as the name of the current task. The following two lines gives you more detailed information about what currently is going on.

Below the boxes there are five lines for the 20m and 25m telescopes. The first of these lines shows the Tsys-value(s) from the last calibration and the last FITS-number used so far. The left part of the second line gives you the name of the selected line as well as the rest frequency (or the net-LO in VLBI-mode) while the right part of the line normally is blank. When the telescope is slewing, it will tell you how long time it will take to reach the commanded position.

The third line tells you the currently applied Doppler correction (both in velocity and in frequency) as well as a status flag for the telescope. The fourth line tells you the current position of the telescope and the error between commanded and actual position (note that the unit used for the error changes with the size of the error). The final line shows you the current time and LST.

For the OTTs, there are only three lines below the boxes, since the OTTs don’t produce any spectra: the telescope position line, the clock line and one line with two telescope status flags and an indication of from where the telescope is being controlled.

For the ozone project radiometers, there are two lines below the boxes: the Tsys/FITS-scan line and the clock line.

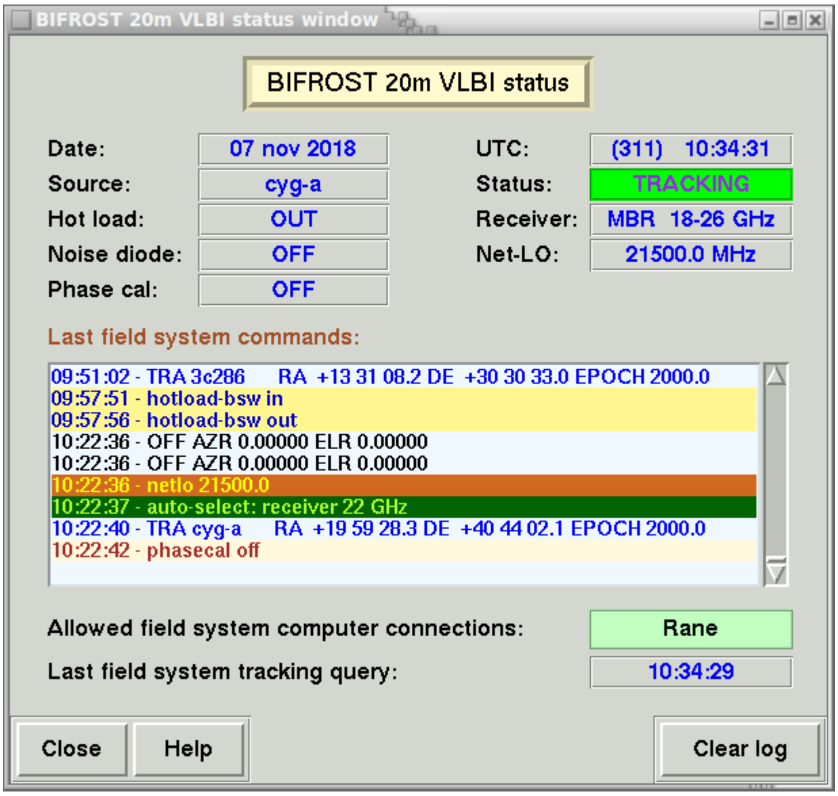
Apart from the **Close** and **Help** buttons, there are two more buttons which will open two useful monitor windows. One is the **Weather** button which opens the weather display. The other button is the **Telescope** button if you are using a telescope or the **IF/LO** button if you are using an ozone project radiometer. The **Telescope** button opens the telescope position display window, while the **IF/LO** button opens the IF/LO display window.

You can reopen the observation status window with the **Open obs status display** in the **More utilities** menu, if you have closed it.

The VLBI status window (VLBI-mode only)

The VLBI status window only appears if BIFROST is run with a telescope in VLBI-mode. It pops up next to the main menu (it may pop up at some other place if you have set the preference telling BIFROST to remember window positions between sessions).

The top part of the window contains a number of status boxes showing the important stuff you need to keep an eye on when running VLBI observations like current source, telescope status, net-LO frequency and status of calibrators. If any of the boxes turn orange, then that is a sign that it has been set manually from BIFROST and that the value differs from the last received command from the VLBI field system.



The VLBI status window on the 20m telescope during a 22 GHz VLBI session.

The big box is a log of all incoming commands from the field system with appropriate colour-coding. All the commands are, of course, also shown in the observation log display but there they will be mixed with information of what BIFROST did in response to those commands. Here just the commands are shown as they are received.

Below the box with the command log, there are two lines. On the first one is a pull-down menu used to select which field system BIFROST should be listening to. If set to **Any**, BIFROST will accept commands sent from any field system, and set to **None** BIFROST will reject any incoming command.

The second line is the time stamp for the last tracking query from the field system. When a VLBI schedule is running, the field system computer sends a query to BIFROST every few seconds to ask whether the telescope is tracking or not. That this time stamp is updated is thus a sign that the FS is talking to BIFROST. The arrival of these queries are also used by BIFROST to decide whether to show the **VLBI** icon (indicating an ongoing experiment) or not in the observation status window.

The **Clear log** button clears away all messages from the **Last field system commands** box, and can be used whenever you want.

You can reopen the VLBI status window from the BIFROST utilities window, if you have closed it.

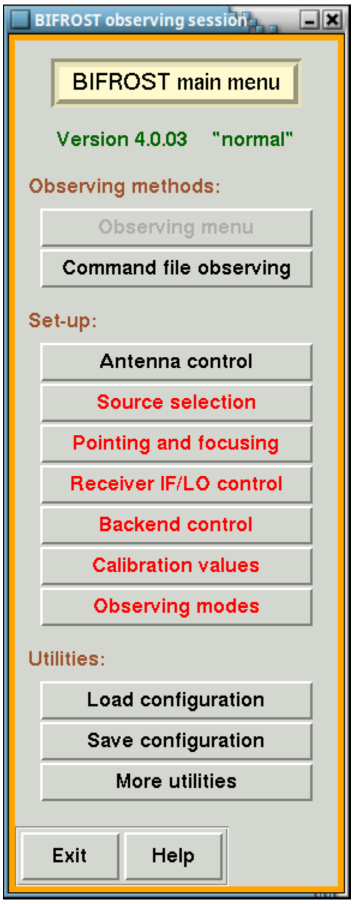
Configuring the system

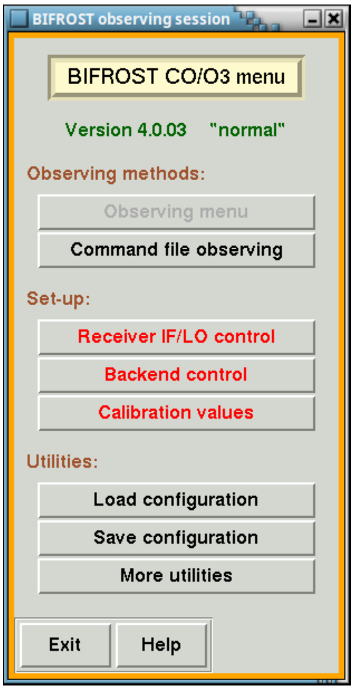
Before you can start observing you need to configure the system. BIFROST does not rely on any information about the current configuration, and when an observing session is started up all devices are flagged as unconfigured. You will thus have to make a full configuration either manually or by loading a configuration file.

When starting up BIFROST, you will see that a number of buttons have red text — this is to indicate that these systems need to be configured. The **Observing menu** button will also be disabled until all systems have been properly configured.

The typical way to handle configurations is to first go through the different configurations, set them up manually and then save them to configuration files. After that you just load the proper configuration file when you want to change configuration.

For VLBI observations, it is not necessary to go through all the steps, unless you plan to do some single dish observations to check the set-up or verify the pointing.





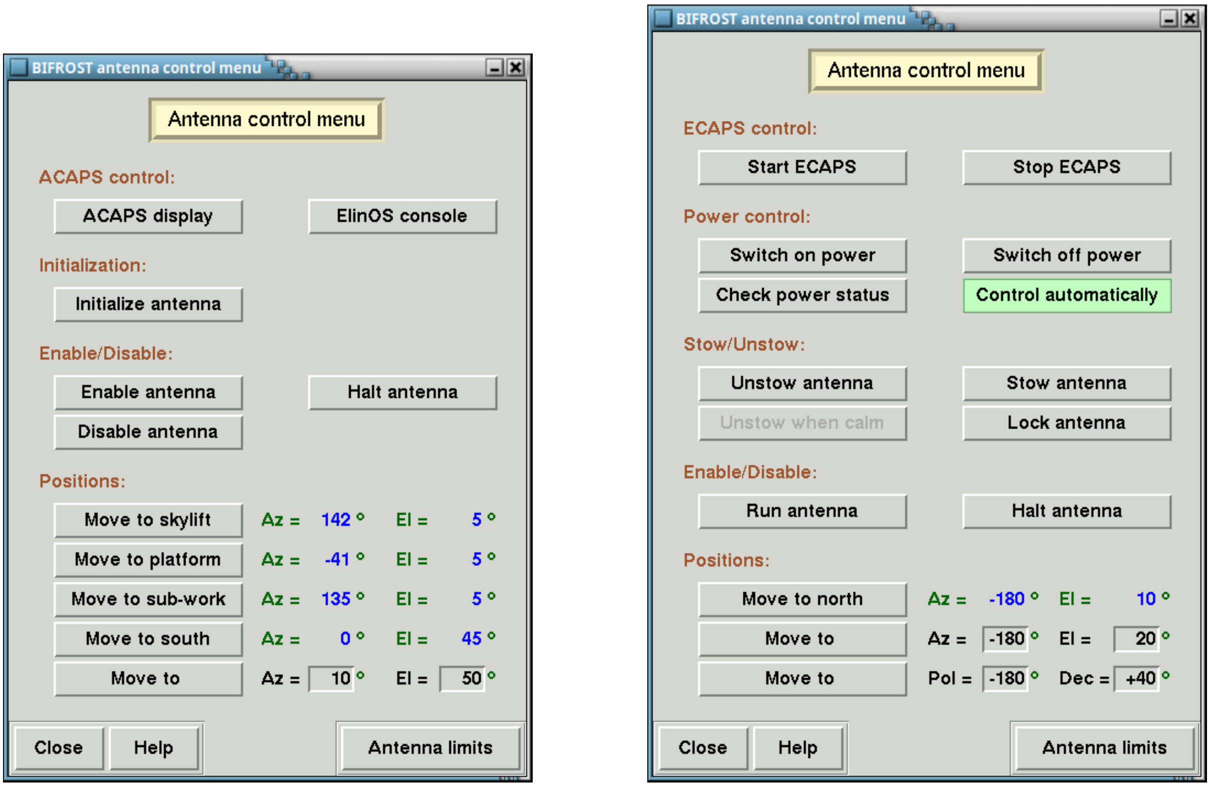
The 20m telescope and the CO/O_3 main menus as they appear when an observing session just has been started. Almost all buttons in the **Set-up** *section have red text to indicate that they need to be configured and the* **Observing menu** *is disabled until all parts have been properly configured.*

Manual configuration

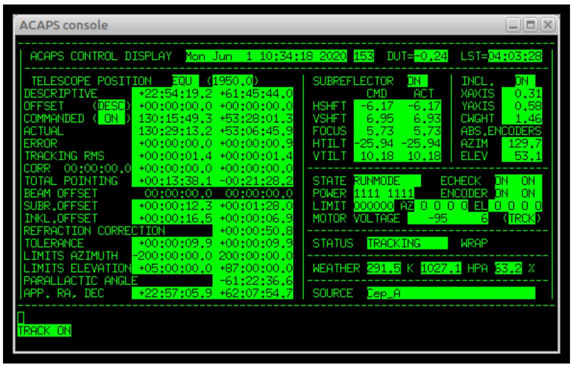
To configure the system manually, you use the buttons in the middle section of the main menu. You should typically go through the buttons in order starting with the topmost one, but it is not strictly necessary. However, there are certain things that need to be done in the right order: you need to enable the telescope before you can send it to a source, and you need to indicate which backend(s) you are going to use in the receiver IF/LO selection menu before you can configure them. Also, for the 20m telescope, if you are using a receiver with needs tuning (the SIS 3/4-mm receiver), it is useful to select the source before you set up the frequency so that you get the Doppler correction taken into account when tuning. For the 20m telescope, you should at least enable the antenna so that the correct pointing and focusing models can be sent to the telescope automatically when you select receiver.

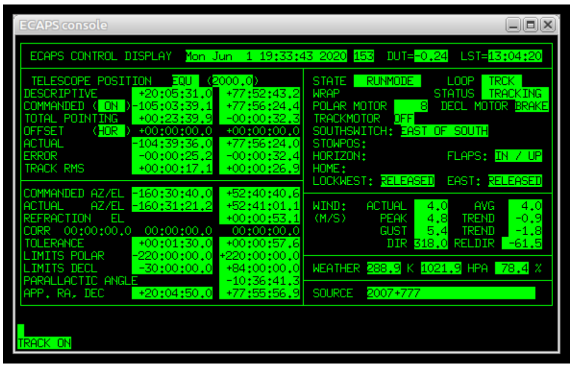
Antenna control

The Antenna control menu window is available on all telescopes, and you may or may not need to use it depending on whether the telescope already is up and running or not. The window is a bit different from most other BIFROST windows in the sense that it just consists of action buttons that will perform their action immediately when you click on the button.



The Antenna control menu *window as it appears on the 20m telescope (left) and on the 25m telescope when it is in VLBI-mode (right). All buttons in these windows are action buttons that do something immediately when clicked.*





The ACAPS *control display for the 20m telescope (left) and the* ECAPS *control display for the 25m telescope (right). It is not strictly necessary to have them open, and there are monitor windows within BIFROST that provides the same information, but they are usually left open, and the* ECAPS *is always opened automatically when ECAPS is started.*

ACAPS and ECAPS are the telescope control software packages controlling the 20m and 25m telescopes, respectively.

The ACAPS display is an old-style ASCII terminal display running on JUPITER which is brought up by opening the **ACAPS display** and then type the command "`display`" in this window. If the power has been cut to the 20m telescope, the command "`cycle`" has to be given in ACAPS window to reactivate the motor controller. If the ACAPS software has been switched off, it is necessary to first start it up with the "`startup`" command.

ECAPS has a similar display although it is not run on a separate computer, and the **Start ECAPS** will start the entire ECAPS control system including the ECAPS display window.

The **Power control** section includes buttons to switch on or off the power to the 25m telescope via the APC. There is also a button to check the current status of the APC power switch and a menu that tells BIFROST whether it is allowed to control the APC switch itself or if the switch is only controlled manually. When set to automatic, BIFROST will automatically switch on the power before unstowing the telescope and switch off the power after it has been stowed.

The 25m telescope needs to be stowed when not in use and there are thus buttons to run the stow and unstow sequence. Both these operations take a number of minutes and a pop-up window will appear which shows the progress through the various steps of these procedures. The **Lock antenna** button can be tried if the telescope has reached the stow position but failed to engage the locks that will hold it in position. In VLBI-mode there is an extra **Unstow when calm** button present. It can only be used if it is very windy and the **Auto-unstow if wind** option in the Antenna limits menu has been set to allow automatic unstowing. Normally, the automatic unstowing would only take place provided that the telescope has been automatically stowed due to the wind, but if it is too windy to unstow the telescope when a VLBI observation begins, you can use this button to tell BIFROST that you want bypass this condition and have the telescope unstowed automatically when the wind situation permits.

The **Initialize antenna** will reinitialize the incremental encoders on the 20m telescope. This can be done at any time, but it is only really needed after a restart of the ACAPS software. The ACAPS display turns red when it is mandatory to initialize the encoders.

The buttons in the **Enable/Disable** section can be used to enable and disable the telescope. The **Enable antenna** button will release the brakes and activate the motors on the 20m telescope, while the **Disable antenna** button will apply the brakes and switch off the motors. The **Halt antenna** button will stop any telescope motion but leave the telescope with the motors on and the brakes off.

For the 25m telescope, **Run antenna** enables the telescope tracking while **Halt antenna** disables the tracking.

The bottom part of the window is a set of action buttons that will send the telescope to a specific position which either is predefined or one of the observer's choice. These buttons only work provided that the telescope is up and running.

The proper way to park the telescopes is to use the **Stow antenna** for the 25m telescope. For the 20m telescope, it is to send the telescope to one of the fixed positions and once it has reached that position click **Disable antenna**.

Antenna limits

The Antenna limits menu window is available on all telescopes. There is no button for it in the main menu — instead you open it from the Antenna control menu. You can typically ignore this window, since the default values are good for most situations and these values will be applied automatically by BIFROST.

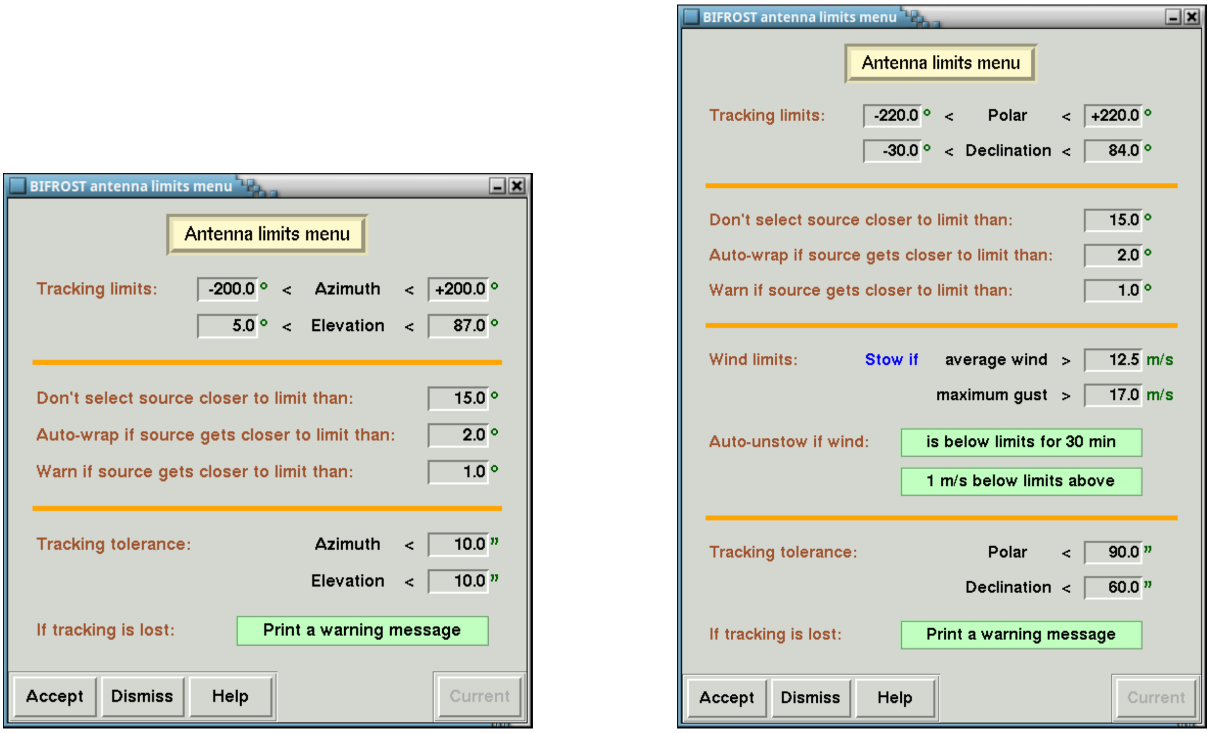
The **Tracking limits** set up the software limits used by the telescope control system (ACAPS or ECAPS). The telescope will never go outside these limits.

The **Tracking tolerance** sets the window which the telescope control system use to determine whether the telescope is tracking or not. There is also a menu where you can decide whether you want to get explicit warning messages when the tracking is lost or not. On the 25m telescope, you typically set a smaller tracking tolerance on the declination axis since the tracking is done in polar. Depending on which frequency band you are using on the 25m telescope and the wind situation you may want to increase the tolerance in the polar coordinate to reduce the number of warnings.

The **Wind limits** on the 25m telescope are used to set the wind limits when BIFROST should stow the telescope. Apart from this, ECAPS also has a built-in limit that will auto-stow the telescope if the wind hits 13.5 m/s.

The **Auto-unstow if wind** only appears in VLBI-mode on the 25m telescope. It allows BIFROST to automatically unstow the telescope once the wind has improved. It is only activated provided that this menu is set to allow it (the default is to *not* allow it) *and* there is a FS connected to BIFROST *and* the telescope has been stowed automatically due to high wind. The last condition can be bypassed by using the **Unstow when calm** button in the Antenna control menu if it is too windy when you want to unstow the telescope.

The three options with long descriptions are there to add a bit better handling of the telescope when observing sources in the north where the telescope can reach the source either from the east or from the west. The default behaviour of ACAPS and ECAPS is to always minimize the slew time. Thus if you ask the telescope to go to a source that



The Antenna limits menu *window as it appears on the 20m telescope (left) and on the 25m telescope (right). To get to this window you first have to open the* Antenna control menu, *since you typically don't need to change the default values.*

actually is just about to hit the limit, the telescope will go there anyway. The **Don't select source closer to limit than** option tells BIFROST to check if this is the case and it will force the telescope to go around to the other side if the source is closer to the limit than the specified angle.

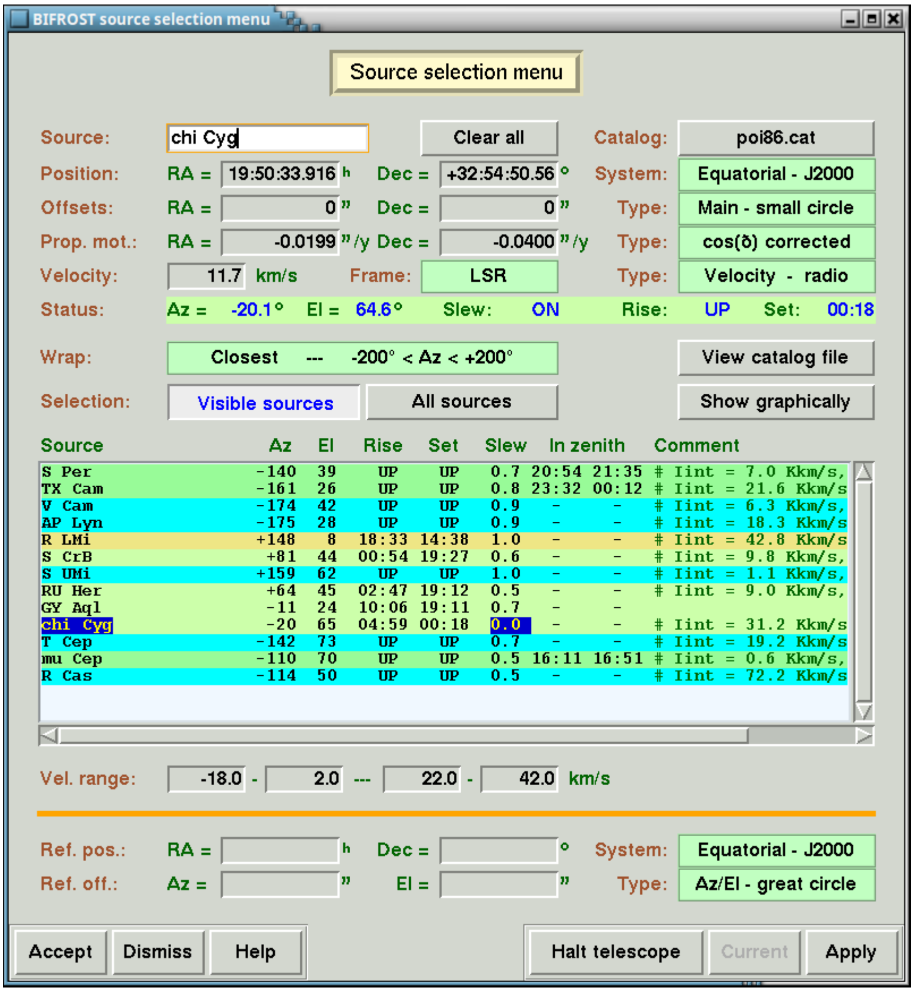
The **Auto-wrap if source gets closer to limit than** is used by BIFROST during spectral line observations. To avoid a situation where the source hits the limit in the middle of an integration, which will trigger an automatic unwrapping of the telescope, BIFROST checks the position before the integration is started and forces the telescope to unwrap if this limit has been passed.

The **Warn if source gets closer to limit than** will start issue warning messages when the source is getting close to hit the limit.

Note that the last three limit options only apply if the source is going to hit the limit — it doesn't apply to a source that is moving away from the limit.

Source selection

The Source selection menu is used to select where the telescope should be pointing. You could enter coordinates by hand, but the best way to avoid any mistakes is to put your sources in a source catalog and then select a source from the catalog. There are a number of general catalogs available with pointing sources, planets and other strong sources. You select catalog with the **Catalog** button in the top right corner.



The Source selection menu *where the source* chi Cyg *has been selected from the 86 GHz pointing catalog on the 20m telescope. Apart from very accurate positions, this pointing catalog also includes proper motions and velocity ranges for the spectral line.*

The available sources are shown in the source list together with their current horizontal coordinates, the rise and set time, the slew time to reach the source and the zenith zone entrance and exit time, if the source will pass through the zenith zone. You can switch between listing all sources in the catalog or only those that are currently visible with the **Selection** buttons.

The sources are colour-coded depending on their visibility:

- **Blue:** the source is circumpolar
- **Light green:** the source is up and will be up for a while
- **Dark green:** the source is up but will later go into the zenith zone
- **Light yellow:** the source is up but will set within 1 hour
- **Orange:** the source will set before the telescope can reach the source
- **Red:** the source is down
- **Pink:** the source is down but will rise within 1 hour
- **Strong yellow:** the source will enter zenith zone within 1 hour

- **Violet:** the source is currently in the zenith zone
- **Brown:** the source is never visible from Onsala
- **Light grey:** the source is a fixed position
- **Dark grey:** the source is a fixed position not visible from Onsala

The minimum information needed to specify a source is its name (which can contain spaces and certain special characters), its position and its radial velocity (including what type of velocity it is).

The position of the source can be given in several different coordinate systems: horizontal, galactic and ecliptic (B1950, J2000 or today's epoch). For the 25m telescope, you can also use polar coordinates. You can have an offset added to this position with the offset given either as horizontal coordinates or in the same coordinate system as the position. Proper motions can also be added, if desired.

When adding offsets, you have to specify whether they are *great circle offsets* or *small circle offsets*. The difference is how the longitudinal coordinate is treated: for *small circle offsets* the longitudinal offset is applied with the value provided, while for *great circle offsets* the value is multiplied by $1/\cos(latitude)$ to maintain an equal angular distance.

Ephemeris-based sources are treated differently. They should still be included in a catalog file, but the actual position and velocity will be taken from an ephemeris file that should be downloaded from JPL Horizons. BIFROST will ask you if you want to download new ephemeris data, when you are using ephemeris-based sources. You can have several different ephemeris files for the same object — BIFROST will automatically identify the one with the most recent ephemeris. A number of the entry fields and menus will be locked and shown in blue, when you have selected an ephemeris-based object.

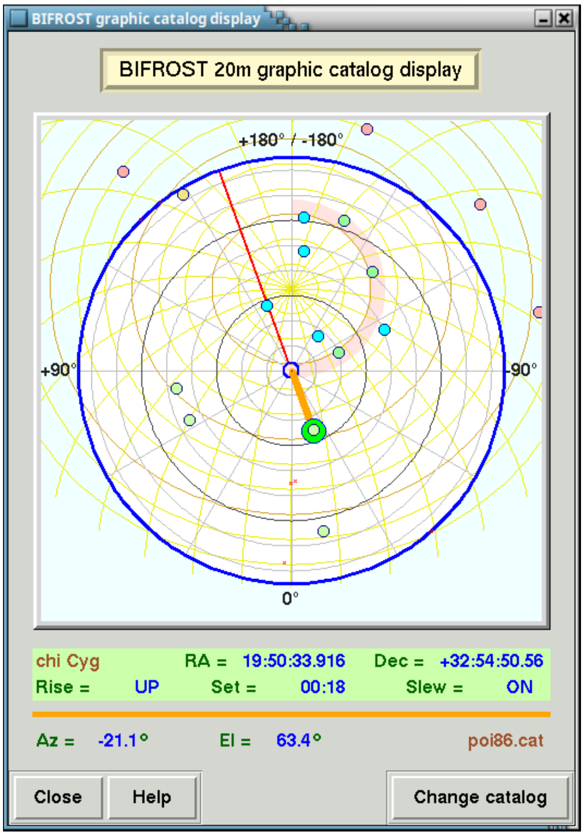
You can use the **Show graphically** button to open the graphic catalog display which will show you the current location of the sources on the sky.

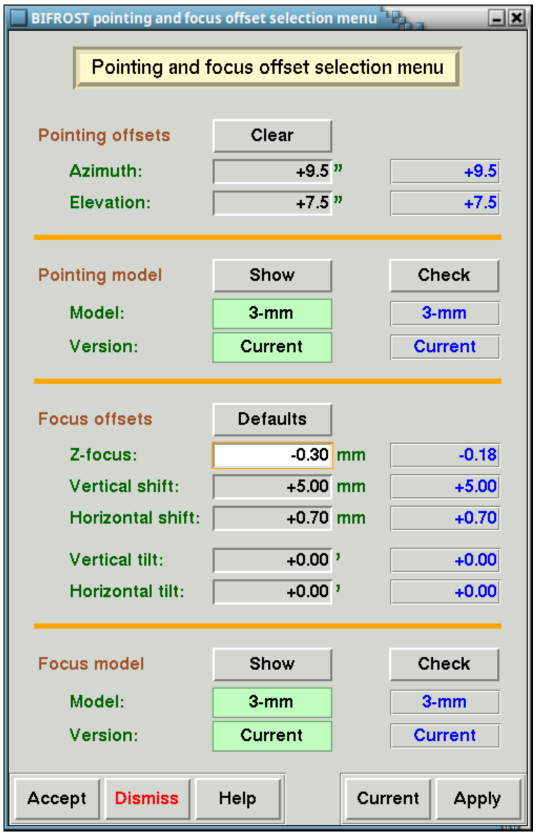
Once you select a source either by clicking in the source list or by clicking on a source in the graphic catalog display, the uppermost six lines are updated with data about the selected source.

The **Vel. range** values for the baseline boxes may also be filled in, if the catalog contains information about where they are supposed to be located. Those values are needed for spectral line pointings (but should be left blank for continuum pointings). This is a problem in VLBI-mode since the net-LO setting won't produce a proper velocity scale. To solve this, there is an extra button in VLBI-mode, **Fix VLBI vel. range** which will recalculate the velocities for the baseline boxes to match the frequency given by the net-LO so that the line do end up between the boxes.

The **Wrap** menu specifies how the telescope should handle sources located in the overlap region in the north where they can be reached either from east or from west. The default action is to minimize the slew time, but with this menu you can force the telescope to a specific side.

The two lines below the orange separator can be used to specify a reference position for the selected source to be used as the OFF-position when doing *position switching*. The reference position can either be given as an absolute position or as offsets relative the source position. If no reference position is specified, then a default offset specified in the Observing mode parameters menu will be used.





The graphic catalog display *(to the left) displays where the sources in a source catalog appear in the sky as well as where the telescope is pointing. In this case the target is* chi Cyg *and information about that object is shown below the graphics. The* Pointing and focus offset selection menu *is shown to the right and we can see that the observer is using the SIS 3-mm receiver and just is about to manually set the* **Z-focus** *to –0.30 mm.*

The graphic catalog display

The graphic catalog display is a graphic tool that can be launched from the Source selection menu or run as a stand-alone program. It plots the current position of the sources in the selected catalog on the sky together with the position of the telescope. The sources are colour-coded in the same way as in the Source selection menu.

Zenith is at the center of the plot and north is upwards. The blue lines indicate the part of the sky accessible to the telescope and are adjusted depending on the limits set in the Antenna limits menu. There are two coordinate systems indicated: horizontal coordinates are shown with grey circles and lines, while hour angle and declination are shown with yellow lines.

The telescope is indicated by a big green circle with an orange line. There is a red line indicating how far you can go before the telescope needs to wrap around. On the 20m telescope version, there is also a pinkish half-circle leading up to the zenith zone, which is there to warn you that sources in that area will hit the zenith zone. There are also some small red crosses indicating some special telescope positions, for example, park and stow positions.

When the mouse is moved over a source in the display, some information about that source is displayed on the two lines below the graphics: the coordinates, the rise and set time as well as the slew time needed for the telescope to reach the source. You can select a source by left-clicking on it, if the display has been opened from the Source selection menu.

If you hold down the middle mouse button in the graphic box, you will create an imaginary source which you can move around. Data about this imaginary source will be displayed below the graphics just as for a real source.

The graphic catalog display can actually be used for more than just showing the currently selected catalog graphically. It can be used to view the contents of a catalog and it can also be used to edit a catalog file. When loading a catalog into the graphic catalog display, its syntax is checked and if your catalog contains some ephemeris-based sources, you will be asked if you want to download new ephemeris for those sources.

Pointing and focusing

The Pointing and focus offset selection menu only exists for the 20m telescope but the other telescopes have a similar window that just contains the pointing part. The window is used both to select the model as well as to set offsets to be added to the model.

The pointing offsets are, not surprisingly, entered in the **Pointing offsets** part of the window. There is a **Clear** button which can be used to reset the offsets to zero, but you will still have to click on **Accept** or **Apply** for any change to take effect.

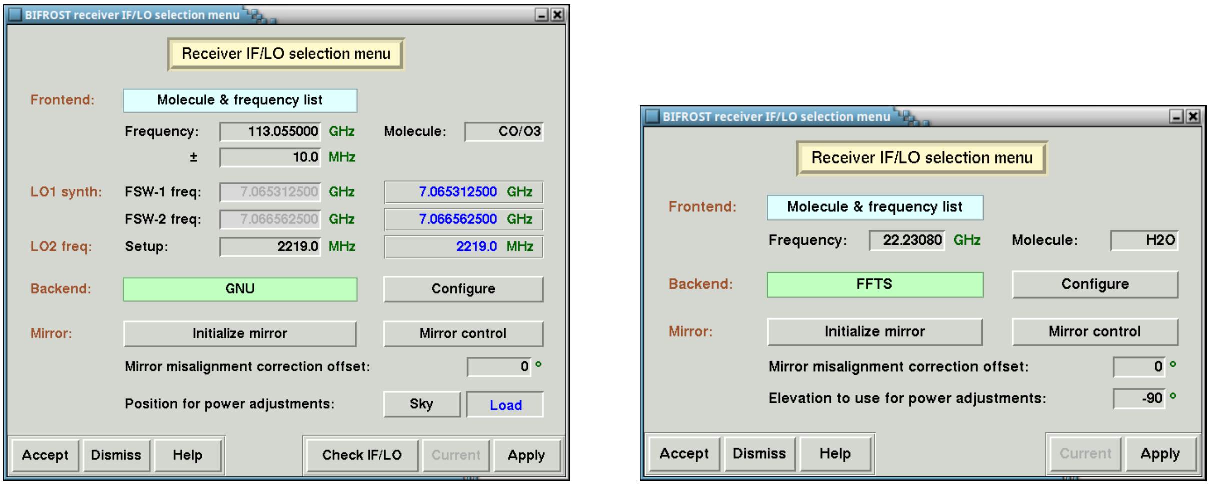
The column of boxes to the right shows you the values the telescope currently is using.

In the **Focus offsets** part of the window, the **Z-focus** is the only parameter which the observer should modify. Some receivers need non-zero offsets for the **Vertical shift** and the **Horizontal shift**. You can get these default values by clicking on the **Defaults** button, and you should not modify those values.

The **Show** buttons will print out information about the selected model to the observation log display including the numerical values of the model parameters. The **Check** buttons also print out the model parameters but they get the values from the telescope control system, so that is a possibility to check that the right parameters have been loaded.

Receiver IF/LO control

The Receiver IF/LO selection menu window is the window where you select receiver to use, observing frequency, IF/LO set-up and backend(s) to use (the actual configuration of the backend(s) is, however, done from the backend control window(s) which you can reach either via the **Configure** button or via the **Backend control** button in the main menu). On the CO/O_3 and H_2O systems, the Receiver IF/LO selection menu is also used to set up mirror parameters and select the position used for power adjustments. A mirror initialization will be done automatically as part of the set-up, but you can also do one manually with the **Initialize mirror** button or send the mirror to a certain position with the **Mirror control** button.



The Receiver IF/LO selection menu *window for the CO/O_3 and H_2O systems with the default configurations for each system.*

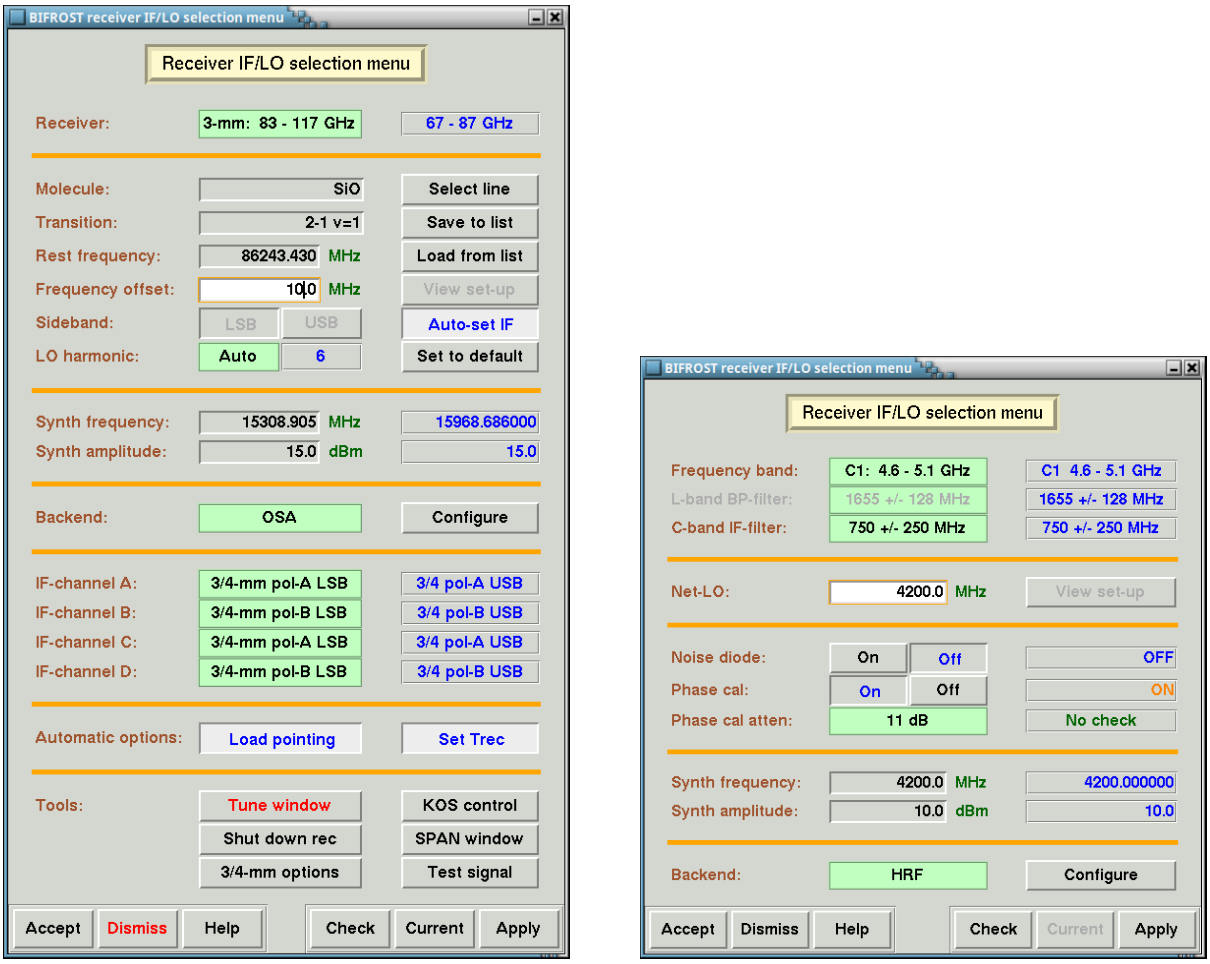
The **Check** or **Check IF/LO** buttons can be used to check the current set-up of the system on the 20m, 25m and the CO/O_3 system. The read-out values are shown in the boxes in the right column.

For the 20m and 25m telescopes, the **Select line** button can be used to open the Spectral line selection menu to browse the 2009 version of the Lovas catalog for spectral lines, or you can type in whatever you like in the **Molecule**, **Transition** and **Rest frequency** fields. The **Frequency offset** can be used to offset a spectral line a bit which is useful when observing with certain backends that have a false spike showing up in the middle of the spectra. You can use the **Save to list** and **Load from list** to build up your own list of frequencies (including offsets). The **Set to default** button resets the settings to the default frequency (typically the pointing frequency) for each receiver.

The **Sideband** and **LO harmonic** is only used for the SIS 3/4-mm receiver on the 20m. The **Sideband** selection is only available for the parts of the frequency bands where it is possible to choose sideband — at the edges of the band where there is no choice, it will be set automatically. The **IF-channel** menus further down will be set up to the correct sideband automatically, if the **Auto-set IF** push button is pressed. **Note** that you don't want the **Auto-set IF** button selected, if you are using a different set-up, for example, if you plan to observe both sidebands simultaneously.

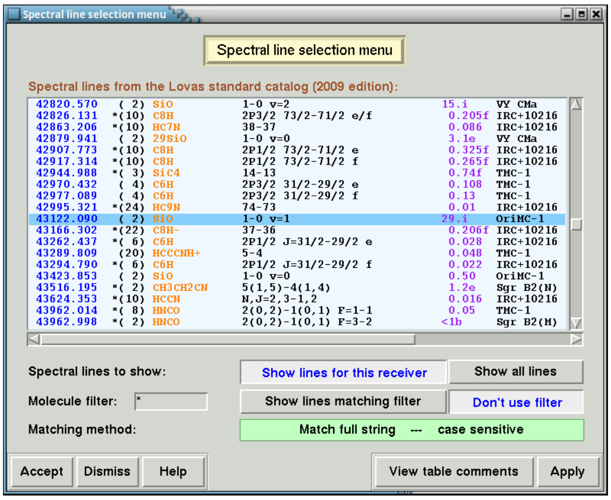
On the 20m telescope, the **IF-channel** menus can be used to connect the desired sideband and polarization as you wish, for example if you plan to observe both sidebands with the SIS 3/4-mm receiver. **Pay attention** to the setting of the IF-channels, if you switch between different frequencies using different sideband settings.

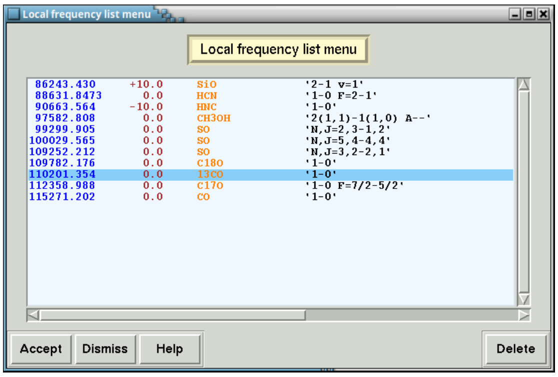
Also on the 20m telescope, there are two push-buttons in **Automatic options**. When these two buttons are selected, BIFROST will automatically load the proper pointing model and the proper Trec values for the selected receiver. Otherwise you will have to do it manually with the **Pointing and focusing** and **Calibration values** buttons, respectively, in the main menu.



The Receiver IF/LO selection menu *window for the 20m telescope and the 25m telescope. In the left window the observer is about to switch from 4-mm to 3-mm observations running BIFROST in normal (spectral line) mode. In the right window we see a C-band set-up in VLBI-mode with the phase cal switched on. Following VLBI-standard, the frequency is given as a net-LO instead of a rest frequency.*

Finally, there is also a set of **Tools** in the 20m Receiver IF/LO selection menu window. The buttons in the left column are only used with the SIS 3/4-mm receiver and will be explained in the next chapter. The **KOS control** and **SPAN window** buttons are also mainly useful for that receiver. The **SPAN window** is a remote display for a spectrum analyzer in the control room, which is mostly used to show the Gunn diode phase-lock when the SIS 3/4-mm receiver is used. The **KOS control** opens the SIS 3/4-mm KOS status window to check the quasi-optics when the SIS 3/4-mm receiver is used, and the Multiband receiver KOS control menu when the multiband receiver is used. While the former window is a graphic window showing the positions of the mirrors, the latter just has a pair of radio buttons to switch between the sky and the load. The **Test signal** button is mostly for staff use and opens a small window where you can switch on a test signal to test the system.





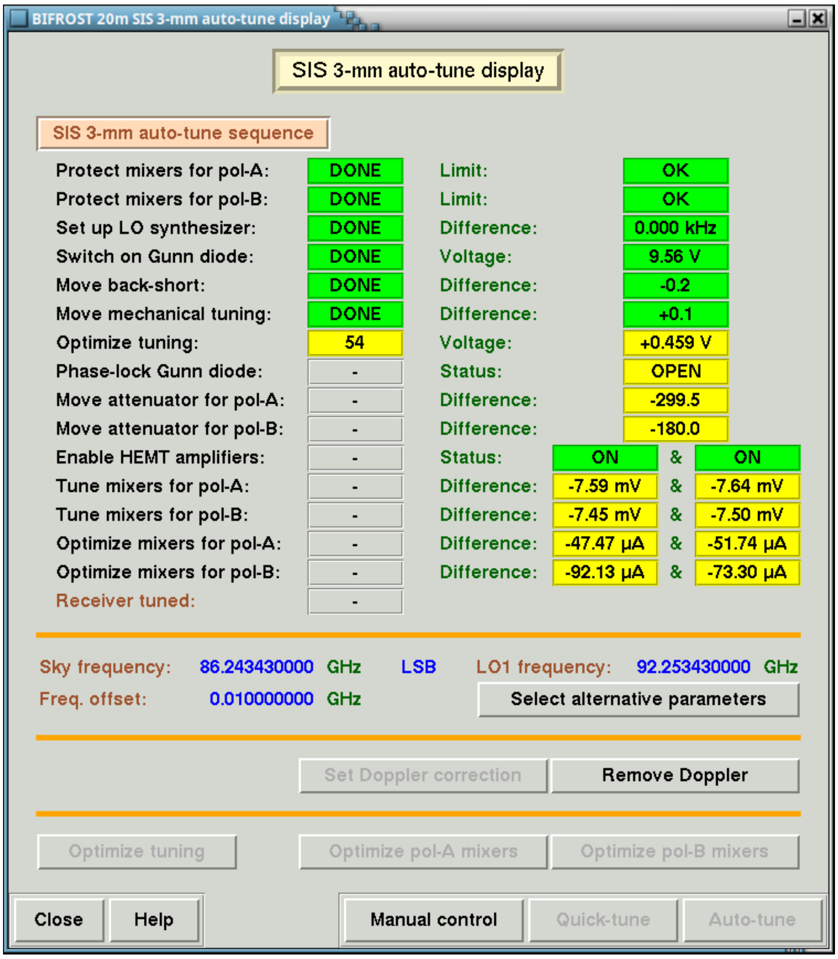
The Spectral line selection menu *is used to browse the Lovas catalog for spectral lines. You can scroll right (or enlarge the window) to also see what telescope made the first detection and get the references. Clicking on telescopes or references gives you a pop-up with more information. You can build up your own short-list of frequencies by* **Apply***ing them in the* Spectral line selection menu *and then click the* **Save to list** *button in the* Receiver IF/LO selection menu. *You can select lines from your short-list by opening the* Local frequency list menu *with the* **Load from list** *button.*

On the 20m and 25m telescopes, the Receiver IF/LO selection menu is one of the few windows which look differently depending on whether your are observing in normal or in VLBI-mode. The main difference is that you specify a **Net-LO** instead of a **Rest frequency** in VLBI-mode, so there are no fields for **Molecule**, **Transition** or **Frequency offset**, nor any buttons for line selections. Also, on the 20m telescope, the buttons for **Noise diode** and **Phase cal** are only shown in VLBI-mode (to save space, since the window is big), but there is an extra **Tools** button: **Power control** which can be used for power adjustments. When observing in normal mode, this button is instead located in the **Observing menu**.

On the 20m and 25m telescopes, the **Synth frequency** and **Synth amplitude** are set automatically depending on the receiver and frequency selected. You can modify them if you want, but in the case of the **Synth frequency**, this will only be useful for VLBI-observations or for some testing, since the frequency will be recalculated automatically as soon as BIFROST tries to update the Doppler setting or receives a new **Net-LO** from the FS.

The SIS 3/4-mm receiver

The SIS 3/4-mm receiver on the 20m telescope is much more complicated than all other receivers, since it uses Gunn diodes with mechanical tuning for the LO generation as well as superconducting SIS-mixers for the 3-mm part, which also need optimization

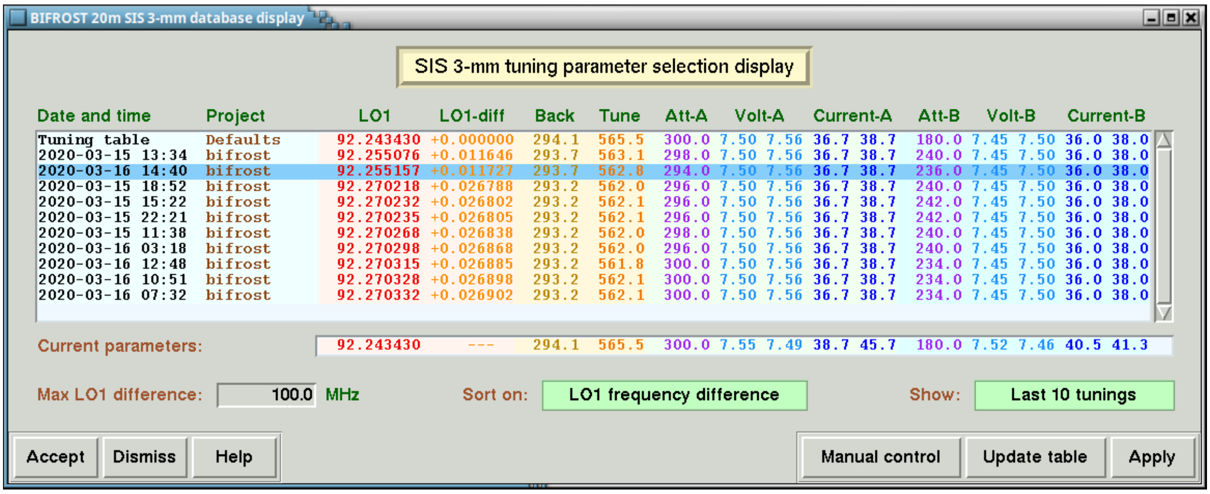


The SIS 3/4-mm auto-tune display *is a status window which shows the progress of the various steps taken when tuning the SIS 3/4-mm receiver. Here, the tuning process has reached the step to optimize the mechanical tuning of the Gunn diode.*

each time the receiver is tuned. So while setting a frequency on all other receivers is accomplished in a second or so, the SIS 3/4-mm receiver needs to go through a complicated procedure which takes several minutes to complete.

The tuning is done automatically as a part of the configuration when you press the **Accept** or **Apply** in the Receiver IF/LO selection menu, provided that it is needed. If the receiver is already properly tuned, then you will be asked if you want to retune or not (for command file observations, BIFROST will assume that you don't want to retune, and you have to provide an extra flag if you want to force a retuning). You can also start the tuning manually by clicking on the **Tune window** button which will take you to the SIS 3/4-mm auto-tune display.

The SIS 3/4-mm auto-tune display is a window showing you the status of the tuning process. The window looks differently depending on whether you tune the 3-mm or the 4-mm receiver, with the 4-mm receiver having fewer steps and being faster since it does not have the SIS mixers which take time to optimize. The larger part of the window is a sequence panel showing you the progress of the tuning process. There are two columns



The SIS 3/4-mm tuning parameter selection display *can be used to select tuning parameters from a recent tuning instead of taking them from the tuning tables. This can speed up the tuning process if the values in the tuning table are far off from the optimal values. In this example, the value for* **Att-B** *given by the tuning table (180.0) is way too low, so starting from the values obtained in a recent tuning will save at least a minute.*

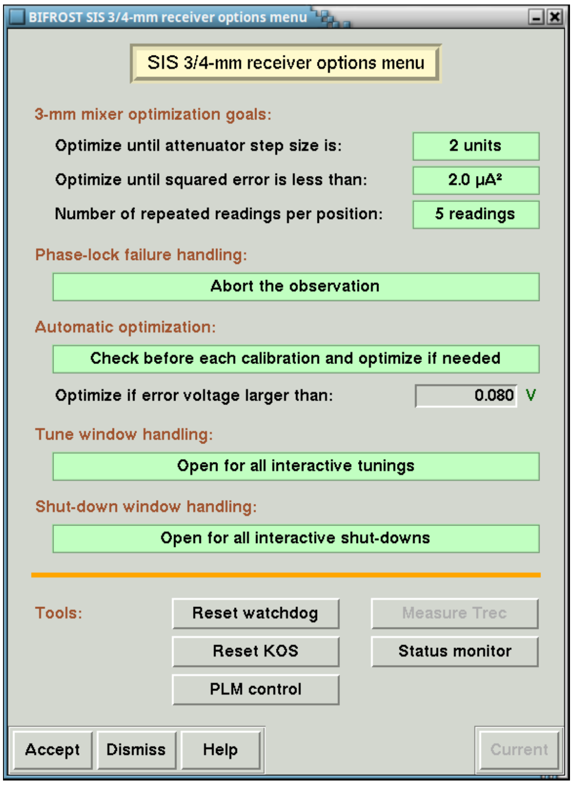
with boxes: the left one shows you which steps have been completed and the right one shows the outcome of the steps. The current step is indicated with a yellow box with a timer counting down the expected time the step will take. For the optimization steps, this number is just a guess, but for the other steps, it is usually fairly accurate.

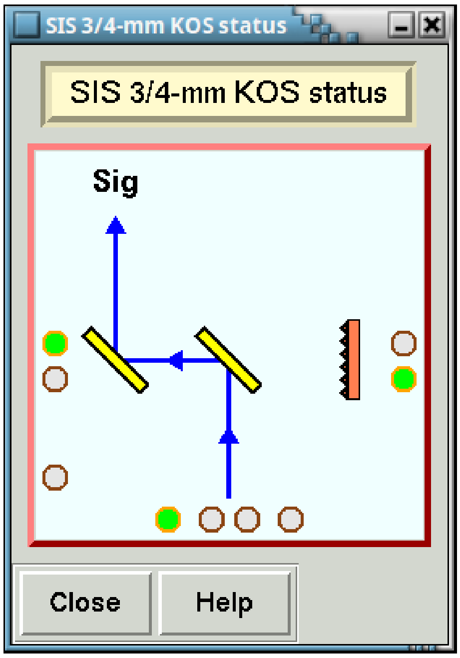
The frequency you are tuning to is shown below the first orange line, and you can add or remove the Doppler correction for the currently selected source with the **Set Doppler correction** and **Remove Doppler** buttons.

The tuning is started either with the **Auto-tune** or the **Quick-tune** buttons. The difference is that while **Auto-tune** will go through all steps, **Quick-tune** will skip the optimization steps, which are the most time-consuming ones. There are three optimization steps for the 3-mm receiver but only one for the 4-mm, so **Quick-tune** is not that useful for the 4-mm receiver. If you discover that you need one or more of those steps, you can run them manually with the **Optimize tuning**, **Optimize pol-A mixers** and **Optimize pol-B mixers** buttons.

The success of the **Quick-tune** and the speed of the **Auto-tune** depend on the tuning parameters being close to the optimal values. In interactive mode, the tuning parameters are taken from the tuning table, and those values are not always the most optimal values, so you may save time by instead using the parameters from a recent tuning. You can select alternative parameters by clicking the **Select alternative parameters** button which will open the SIS 3/4-mm tuning parameter selection display where you can browse through the recent tunings.

For command file observations, BIFROST will automatically browse the tuning history for a good set of tuning parameters, unless you specifically instructs BIFROST to use the tuning tables.





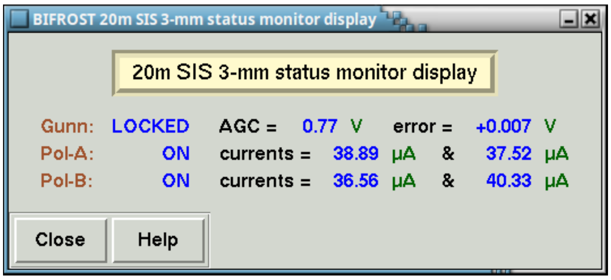
The SIS 3/4-mm receiver options menu *(left) contains a number of extra parameters for handling the 20m SIS 3/4-mm receiver as well as some special tools for this receiver. The* SIS 3/4-mm KOS status *is a small graphic status window which shows you the status of the quasi-optical system (called* KOS*) for the SIS 3/4-mm receiver. The KOS can actually be controlled manually from this window by clicking on the small circles.*

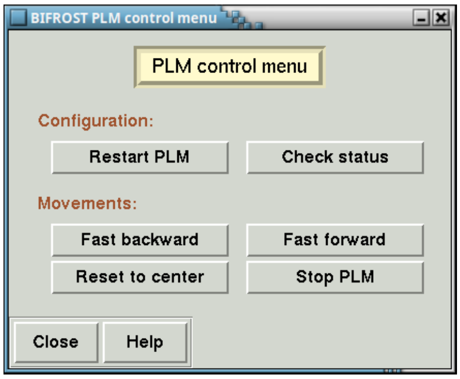
You can shut down the SIS 3/4-mm receiver with the **Shut down rec** button in the Receiver IF/LO selection menu. This button opens a window similar to the SIS 3/4-mm auto-tune display, but designed for the shut-down process and with much fewer steps. The shut-down process will be run automatically, if you switch from one receiver to another. You will also be asked if you want to shut down the SIS 3/4-mm receiver when you exit your observing session, if you have been using that receiver.

There are several other windows used only with the SIS 3/4-mm receiver which you access through the **Tools** part of the Receiver IF/LO selection menu.

The SIS 3/4-mm KOS status window is a small graphic window showing the status of the SIS 3/4-mm receiver quasi-optics. There are two mirrors, one on each beam, switching between the sky and loads as well as a central mirror sitting on a sledge which control which beam the receiver will see. This sledge can be moved at two different speeds (0.5 Hz and 1 Hz) to provide beam-switching, with only the higher speed used for normal observing. The small buttons indicate the selected position of the mirrors and can also be used to command the mirrors.

The SIS 3/4-mm receiver options menu contains a number of extra options for this particular receiver, where you can specify goals for the mixer optimization (relaxed goals can save you minutes when tuning the 3-mm receiver), behaviour if the phase-lock is





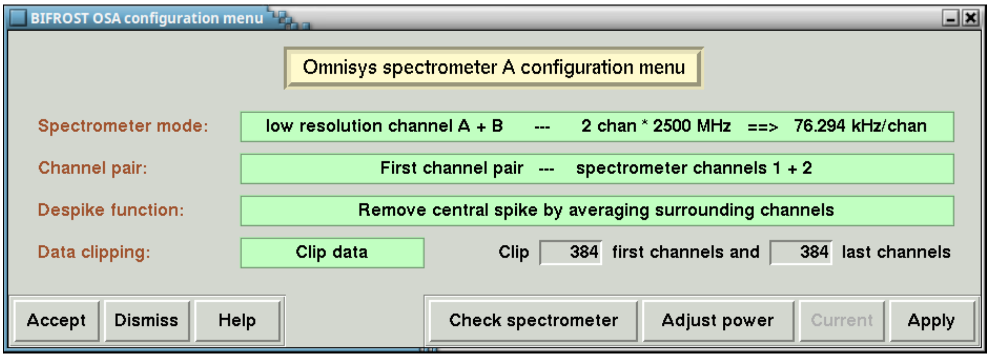
The SIS status monitor display *(left) is a small status window which can be used to monitor the status of the tuning of the SIS 3/4-mm receiver. The four mixer currents are only shown for the 3-mm receiver. The* PLM control menu *(right) can be used to manually control the* path-length modulator *(PLM) for the SIS 3/4-mm receiver, but it is only really needed if there is a problem with the PLM. Both windows are opened from the* SIS 3/4-mm receiver options menu.

lost, auto-optimization of the Gunn if the Gunn tuning becomes non-optimal (highly recommended, especially if you are observing sources with different Doppler shifts), as well as options for when the SIS 3/4-mm auto-tune display and SIS 3/4-mm shut-down display windows should pop up.

There is also a set of tools for the receiver in the lower part of this window. There are two reset buttons, two buttons to open other windows and a button intended for receiver temperature measurements which have not yet been integrated into the system though. The **Reset watchdog** is only needed in the rare occasion that there has been a temperature spike in the cryostat, since the receiver watchdog then will set a flag blocking the use of the SIS-mixers for the 3-mm receiver. The **Reset KOS** will reset and reinitialize the quasi-optic system for the receiver by cycling the power to it. The windows you get when clicking on **Status monitor** and **PLM control** are explained below.

The 20m SIS 3/4-mm status monitor display is a small monitor window that can be left open when the SIS 3/4-mm receiver is in use, since it can give an indication if the receiver starts to drift away from the optimal tuning. The topmost line shows the status of the Gunn diode and the two other lines show the status of the amplifiers, and in the case of the 3-mm receiver also the SIS mixer currents. The AGC voltage should be above 0.5 V and the error voltage less than 0.100 V for a good tuning. The mixer currents should stay blue — however, for some frequencies, the mixer currents may become orange or red already during the tuning process and then there is nothing to do about that, since that will be due to differences between the receiver and the interpolation of the expected values from the tuning tables.

The PLM control menu can be used to manually control the *path-length modulator* (PLM) for the SIS 3/4-mm receiver, which can be used to slowly move the receiver one wavelength during an observation which may help to cancel out standing waves. This window is rarely needed during observations since you select whether to use the PLM or not in the observing scheme parameters when you are observing. The only reason for



The Omnisys spectrometer A configuration menu *window for the 20m telescope is used as an example of what the typical backend control windows look like.*

a normal observer to open this window would be if the PLM has become disabled and needs to be restarted.

Backend control

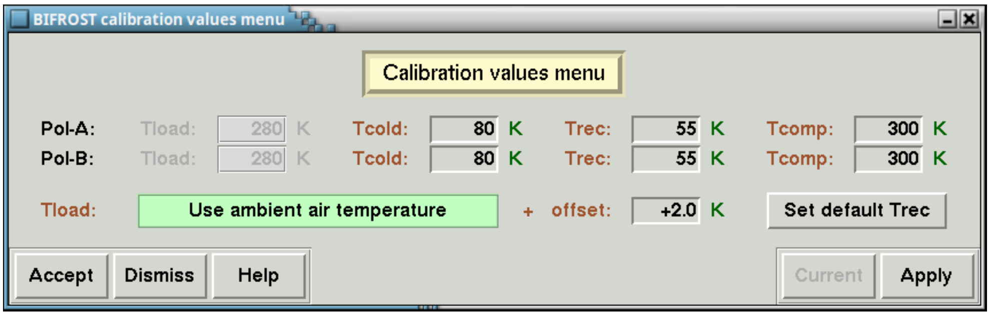
You select which backend(s) to use in the Receiver IF/LO selection menu but you configure the backend(s) with the **Backend control** button. Each backend has its own unique configuration window, but most of them are fairly similar. They all have at least two set-ups: the **Spectrometer mode** menu where you select the configuration which typically is the resolution, and the **Data clipping** which can be used to cut away some channels at the edges of the spectra. You often get some channels with unphysical high values at the edges and if you clip them away you won't have to bother about them, and the auto-scaling will work fine when you are looking at the spectra in the quick-look facility or other data reduction programs.

Some spectrometers have a spike in the center channel, and you can then use the **Despike function** to remove this spike. This spike can be a problem if you have a narrow spectral line in the center of the spectra, and for this reason, there is a possibility in the Receiver IF/LO selection menu to add a small frequency offset to move the spectral line away from this problematic channel.

The *Omnisys spectrometer A* on the 20m telescope has four channels grouped into two pairs. All four are needed when you observe in 4 GHz mode, since the individual channels only can handle 2.5 GHz and you thus need two channels to cover the 4 GHz bandwidth. You also need the four channels if you observe with the SIS 3/4-mm receiver and want to observe both sidebands simultaneously. However, if you just observe a single sideband at a bandwidth of at most 2.5 GHz, the two pairs will be copies of each other, and you should just select one of them.

Calibration values

The Calibration values menu is used to set up the values used when calibrating the data. For systems with two polarizations, you can set the parameters individually for each polarization. You are supposed to enter temperatures for the hot load and the cold load (although it is currently not used for the 20m and 25m telescopes) as well as the



The Calibration values menu *window for the 20m telescope.*

receiver temperature. On the 20m telescope, you have the option to enter fixed values for the hot load temperature or use the ambient temperature with an offset.

Also, on the 20m and 25m telescopes you should provide Tcomp, a *comparative system temperature*. This value is used to calculate a *comparative integration time* of your observation, which is the amount of time it would have taken to reach the same result if the Tsys would have been Tcomp. The actual formula is:

$$t_{comp} = (t_{ontime} - t_{blanking}) \cdot \left(\frac{T_{comp}}{T_{sys}}\right)^2$$

The *comparative system temperature* can be used when doing map observations to adjust the amount of data taken in each map position during changing weather conditions so that the different map positions reach a more equal theoretical noise.

Observing modes

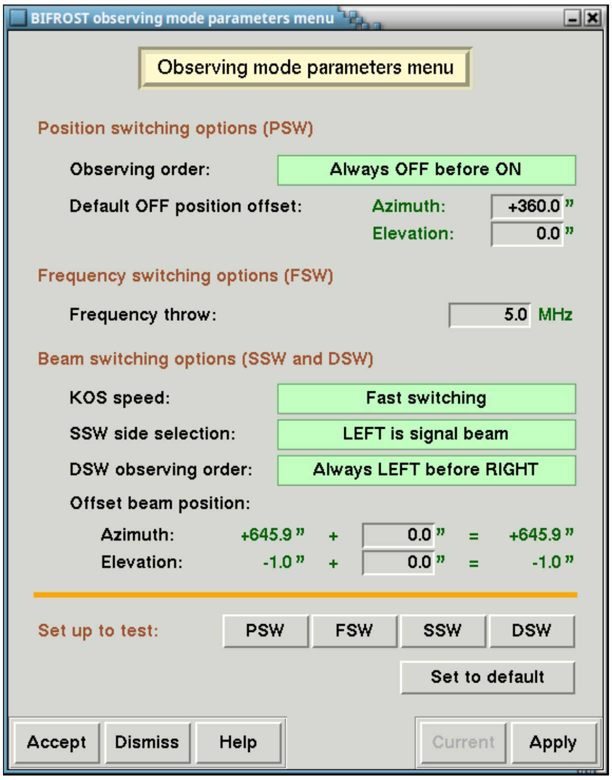
You will only find the Observing mode parameters menu on the 20m telescope. It is a window where you set up some basic parameters for the different observing modes, and for many types of observations you can just click **Accept** without making any changes. The most common parameter to change in this window is the *frequency throw* to be used for *frequency switching* observations.

For *position switching* observations, you can select in which order the ON and OFF spectra should be taken as well as the offsets to use for the OFF-position, if the source to be observed does not have an explicit OFF-position defined.

For *beam switching* observations, you can select the speed of the KOS beam switch. For *single beam switching*, you can choose if you want the left or the right beam to be the signal beam pointing towards the source. For *dual beam switching* you can select in which order the left and right beam should be used as signal beam. The **Offset beam position** is the offset that is used to make the right beam the signal beam. This offset could be adjusted in the rare case that some mirror has been knocked out of position.

The **Set up to test** buttons can be used to force BIFROST to set up the system in a specific observing mode. They are only there for testing, since BIFROST always will set the observing mode automatically during the observations.

This is one of the few windows that does have a **Set to default** button that will restore all the parameters to the recommended default values.



The Observing mode parameters menu *allows you to set some fundamental parameters for the different observing modes used on the 20m telescope. The window is shown here with all the default settings.*

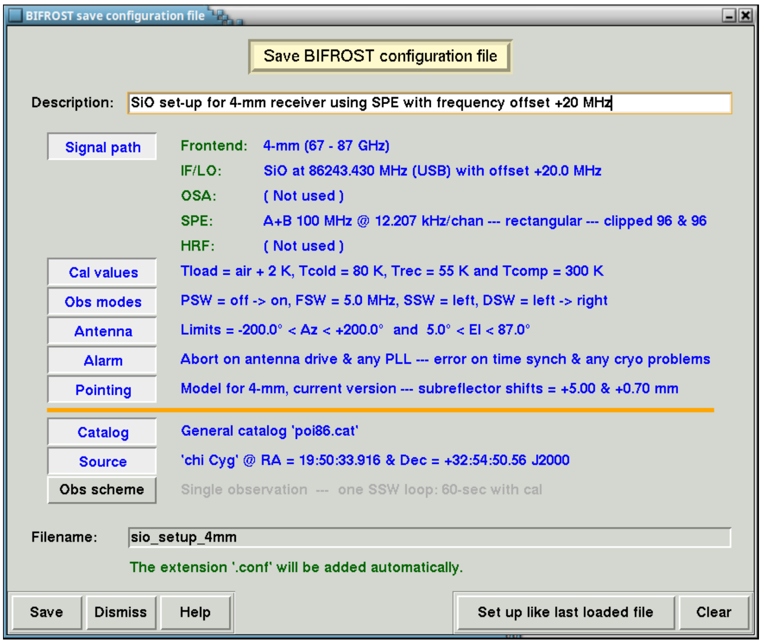
Using configuration files

Although you can do all your configurations interactively with the windows you get in the "*Set-up*" part of the main menu, it is usually much more efficient to save your configurations to configuration files. You can then switch between your configurations by simply loading the desired configuration file. You can create and apply configuration files with the **Save configuration** and **Load configuration** buttons in the main menu.

Configuration files themselves are configurable, since you decide what configuration parts you want to save in the files. In the Save BIFROST configuration file window you will see a list of configurations and buttons to select which configurations to include in the configuration file, for example **Signal path**, **Cal values** and **Obs scheme**. The **Signal path** combines receiver, frequency set-up and backend configurations into one block.

Configuration files are often used to change frequency and backend set-up and it then makes sense to include most of the available configurations to get everything set up at once. You are, however, free to use configuration files as you want. You could very well use small configuration files that containing nothing more than a source selection or another set of observing parameters.

When loading a configuration from a configuration file, you can choose between **Load only** and **Load and apply**. The **Load only** option will open all affected set-up windows and fill in all the values from the configuration file but will leave it to you to actually



A configuration file is about to be created. In this example, all set-ups except the set-up for the observing scheme have been selected for saving. The summary of the set-ups is generated automatically, but the description at the top should be added by the observer. The orange line divides set-ups that can be used in command files from the ones that can't.

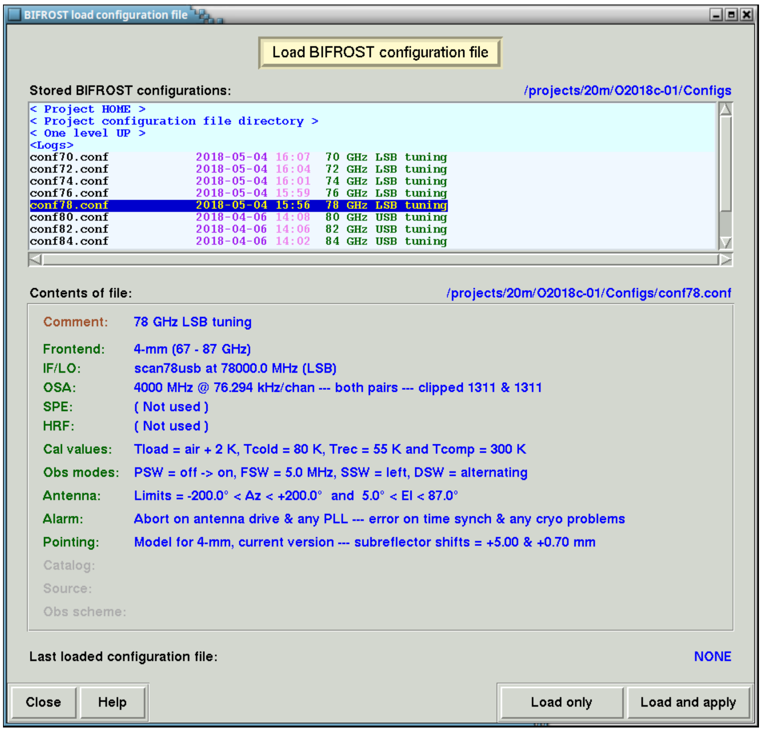
apply the new configuration. This is the alternative to choose if you want to check or make some modification to the configuration before applying it. If you know that the configuration file has exactly the configuration you want, you can use the **Load and apply** button to have the file loaded and applied automatically. If you have included the observing scheme set-up in the configuration file, you will have to start the observation manually even if you load the configuration file with **Load and apply**.

Note that when you use **Load and apply**, a small wait window will pop up. This window will grab the focus, so you don't interfere with the loading process. The window will disappear automatically once everything from the configuration file has been loaded and applied.

Modifying a configuration file

Although configuration files are ASCII-files, you are not allowed to modify them yourself. If you do, the correct behaviour of BIFROST can not be guaranteed. The proper way to modify a configuration file is to load it into BIFROST, make the modifications you want and **apply them**, and then save the modified configuration. In the Save BIFROST configuration file window, there is a button **Set up like last loaded file** which you should use before saving the modified file to make sure the new file has the same contents as the old file (apart from the modifications).

You should add a description of the configuration, since it will be shown in the list when you are selecting a configuration file to load. Apart from that, BIFROST will



The window for loading a configuration file is shown here. The description you entered when you saved the configuration file is shown in the list of files, and if you click on one of the files, you will also see the automatically generated summary for that configuration file.

automatically generate a summary of the configuration file contents, and you will also be shown this summary when you click on a configuration file in the Load BIFROST configuration file window.

The need for configuration files

Configuration files are a necessity if you are going to perform advanced command file observations, since the only way to change a configuration in a command file is to load a configuration file. Note though that some parts of a configuration (for example an observation scheme configuration) are ignored when used in a command file, since you are supposed to provide these parameters in another way in a command file.

Not all set-ups are suitable for command file usage though. The set-up of the observing scheme parameters, for example, should be done directly in the command file. The Save BIFROST configuration file window contains an orange line. The set-ups above the line will be used if the configuration file is used in a command file, while the set-ups below the line will be ignored.

In general it is a good idea to save all your different configurations to configuration files. It is then easy to switch between the different configurations without the risk of forgetting something or doing something wrong.

Interactive observing

Interactive observing is done from the BIFROST observing menu which you open with the **Observing menu** button in the main menu. This button is disabled until you have performed a complete configuration of the system either by going through all the individual windows in the "*Set-up*" part of the main menu, or by loading a configuration file with **Load configuration**.

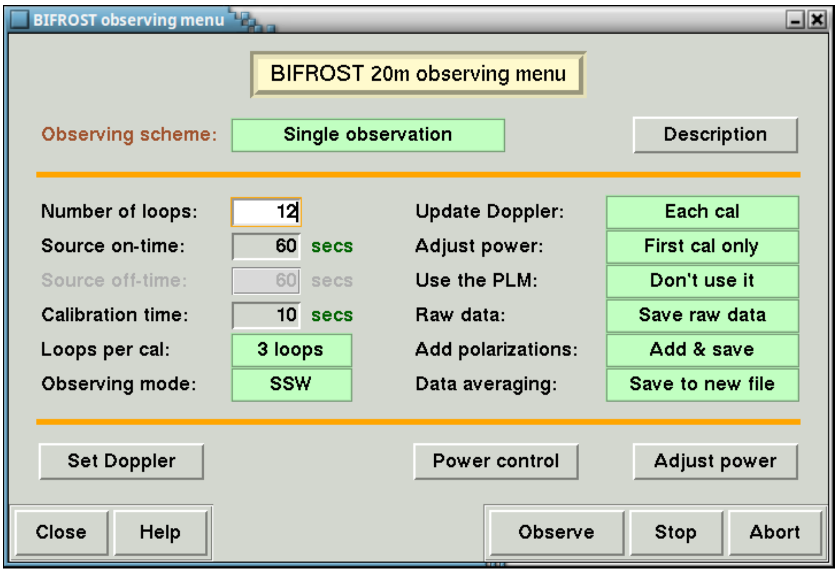
Once the system is configured, the observing menu may disappear if something happens that may call into question the validity of the configuration. For example, if BIFROST receives error messages when talking to a backend, BIFROST will play it safe and flag the backend as unconfigured.

The observing menu

The observing menu is used for all interactive observations. It offers you the choice between several different "*observing schemes*", which produce different kinds of observations such as simple calibrations, pointings, sky dips or single position integrations.

Each of the observing schemes has a one-page description which describes what the observing scheme does and gives some explanation about its parameters. However, for more detailed information on any parameter, you should right-click on it to open the specific help text for that parameter. The description page (which technically is just another help page) is opened with the **Description** button.

Each scheme has its own (independent) set of observing parameters which you have to set up before you start observing. The parameters are shown in two columns between the two orange lines.



The observation menu with the **single observation** *scheme selected on the 20m telescope.*

Most of the parameters are selected with menus, but there are also some parameters with numeric values (for example the integration time) where you will have to type in the value you want. If a parameter has an illegal value, the background will turn red and the **Observe** button will be disabled.

Some parameters may be disabled depending on the situation, for example, the parameter for calibration time will be disabled if you have selected not to perform any calibrations.

The observing menu window works a bit differently from most other windows in BIFROST, where you can play around with parameters and then click on a **Dismiss** or a **Current** button to "*undo*" all the changes you made. The observing menu window does not have this feature, so when you change a parameter, there is no "*undo*" function.

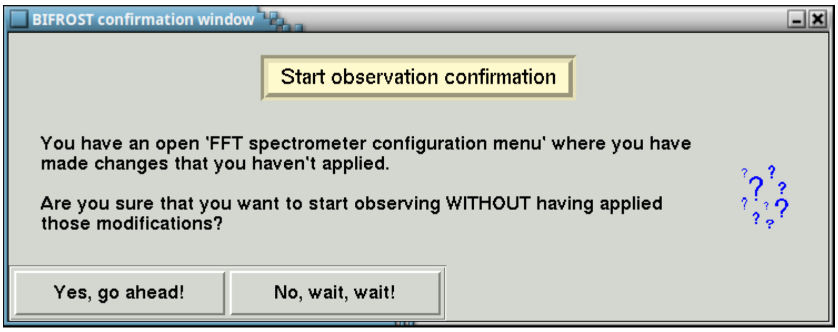
You can modify the parameters for the next observation or even change observing scheme while an observation is going on, since the parameters are sent only once to the BIFROST **executive** when you click on the **Observe** button.

Observing menu buttons

Below the second orange line, there is one or more action buttons depending on which observing system you are using. All systems have an **Adjust power** button which can be used to adjust the power levels immediately — there is also an **Adjust power** parameter that can be used to set up power adjustments during the observation.

For the 20m telescope, there is also a **Power control** button which will open the power control menu where you can see the power measured power levels and adjust power individually on the optic fibers in the cabin or in a backend — the normal power adjustment adjusts the power at all places every time it is called.

The 20m and 25m telescopes both have a **Set Doppler** button which can be used to set the Doppler correction immediately. Again, there is a parameter (**Update Doppler**) which controls Doppler settings during an observation.



Here the observer has opened the spectrometer window, modified the resolution and left that window open without applying the new configuration. When the observer then hits the **Observe** *button, BIFROST wants to know if this was done on purpose or if this was a mistake.*

The text on the **Observe** button may turn red under certain circumstances. This happens if you have a configuration window open and make a change in that window and leave the window open without applying the change. If you click on the **Observe** button when the text is red, you will get a pop-up window telling you what BIFROST is worried about and you will have to confirm that you really want to start observing anyway.

The observing menu window has two buttons for interrupting an ongoing observation: the **Abort** button which will tell BIFROST to abort the observation immediately, and the **Stop** button which tells BIFROST to stop the observation at a convenient time, which typically is after the current loop has finished. The **Abort** button may not give control back to you instantly, since BIFROST may perform some clean-up activities first like telling the backend(s) to stop taking data, removing a calibration load or moving the telescope back to the source position, if the telescope has been in an off-position.

The **Abort** and **Stop** buttons have the same function as the buttons with the same names in the observation log display window. Thus you don't need to have the observing menu window open to interrupt an observation.

Command file observing

Command files are a way to automatize observations by writing a list of commands to be executed. A command file could consist of anything from just a single line of code to perform a specific observation to a complex system of files that call each other with the `INCLUDE` command and can run observations completely automatically for an entire weekend. It is entirely up to the observer to decide how and to what extent (s)he wants to use command files.

Everything you can do interactively, you can also do with a command file — and actually a bit more. For example, command files offer the possibility to automatically pause the observations if the weather is bad, and even simple things like just keeping on integrating on the same source could thus benefit from being run via a command file. In interactive mode, this responsibility is left to the observer. Another example is if there is a temporary technical glitch, such as a network problem, which triggers an error. In interactive mode, it is up to the observer to check what the problem is and restart the observation, while a command file can use the automatic restart functions to recover and get going again.

Command files are simple ASCII-files and they are written in a simple language developed for BIFROST. You can prepare them with any editor, but it is highly recommended to use the command file editor provided with BIFROST, since it will provide you with instant help on command syntax, explanations of each command as well as showing examples of usage. Another important feature is the on-the-fly syntax check, which immediately will tell you if any command or argument is not syntactically correct.

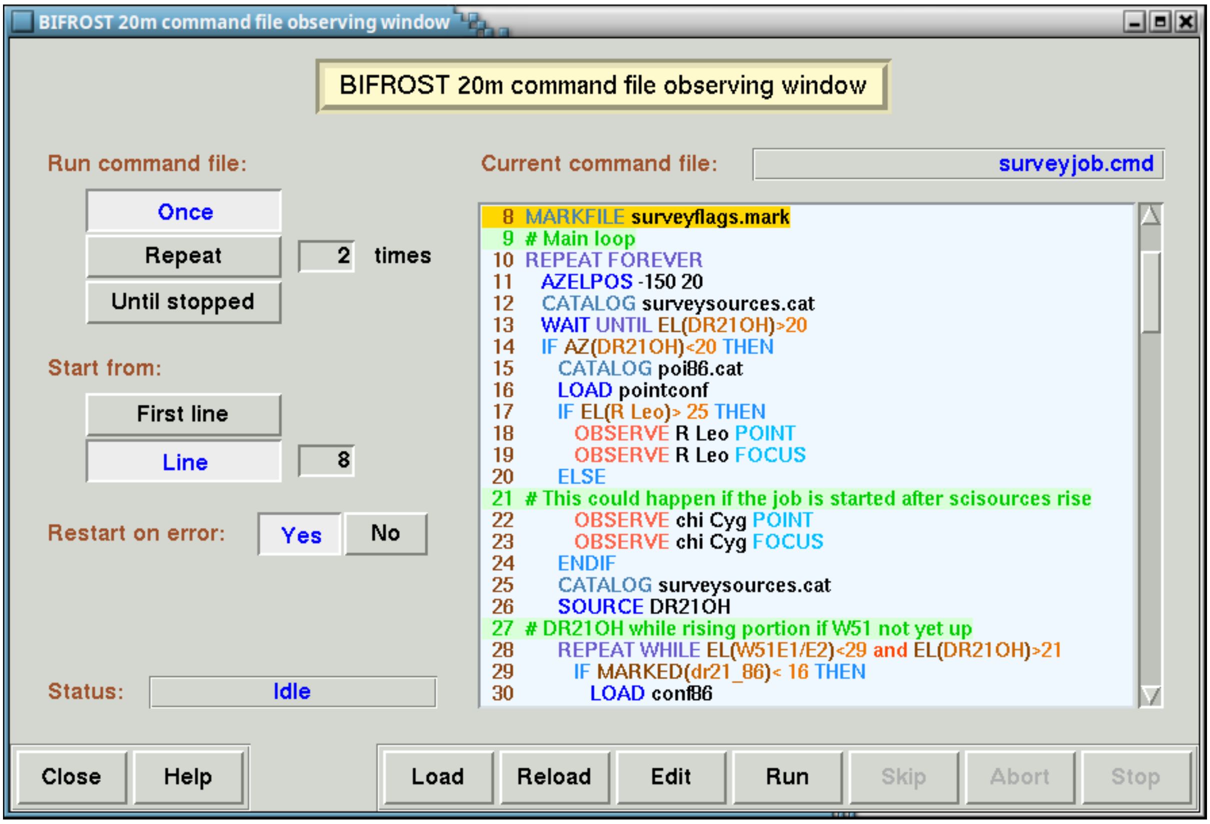
Running command files

Command files are run from the command file observing window which you can open with the **Command file observing** button in the main menu. Contrary to the **Observing menu** button which only is enabled once you have configured your system properly, the **Command file observing** button is always enabled, since you may run a command file that does all the configurations by itself.

It is not allowed to close the command file observing window while a command file is running. You can click on the **Close** button, but the window will be iconified instead of being closed. Clicking the **Command file observing** button in the main menu will deiconify the window.

Features in the command file observing window

The command file observing window is dominated by the big box where the currently loaded command file is shown. The command file is shown with colour-coding: valid commands are shown using various kinds of blue, purple, black and orange colours, while comments are shown in green and invalid lines in olive. Line numbers are added automatically and are shown in brown for valid lines and in yellow for continuation lines. Invalid lines have red line numbers and will be ignored.



A command file is ready to be used on the 20m telescope. Line 8 has been selected as the start line and the automatic restart on error has been enabled.

You can "*comment out*" lines in the command files directly in the command file observing window without the need to do it physically in the command file itself. Clicking on a line with the middle-mouse (or `<Shift>`-left) will toggle it between being active or deactivated. Only active lines will be executed when you then run the command file. You can only use this feature when the command file is not being executed.

The **Run command file** part is a set of buttons to select whether the command file should be run once or several times. Repetitions can easily be handled in the command file with more flexibility using the `REPEAT`-command, but this is a legacy from the CIMA-system which doesn't have the `REPEAT`-command, and some observers may still find it useful.

Selecting a start line can be done with some buttons but you can also just left-click on the line you want to start from. Note that you should avoid starting a command file from inside a structure such as an `IF`-statement or a `REPEAT`-loop, since BIFROST considers it an error to encounter an `ENDIF`- or `ENDREPEAT`-statement without first having seen a corresponding `IF`- or `REPEAT`-statement.

The **Restart on error** buttons decide whether BIFROST should try to restart a command file in case of a problem. If the **Yes** button is selected, BIFROST will restart the command file **once** from the start line in case of a problem. This behaviour can be modified in the command file with the `ALLOWRESTART` command. If the **No** button is selected, no restart will be attempted and all `ALLOWRESTART` commands will be ignored.

Note that even if a restart has been requested, there are situations where BIFROST won't attempt a restart, since the problem seems to be unsolvable — an example of such an unsolvable problem is if you have asked BIFROST to load a configuration file that doesn't exist.

There is a small **Status** box which shows you whether the command file is running or not. It will also show you if you have requested an abort or a stop.

Between the **Restart on error** and the **Status** lines there is some space that only is used when a command file is running. Then a line is showing up telling you which command file line currently is being executed. If you use the **Run command file** buttons to run the command file several times, a second line will appear telling you which loop BIFROST is currently doing.

Action buttons in the command file observing window

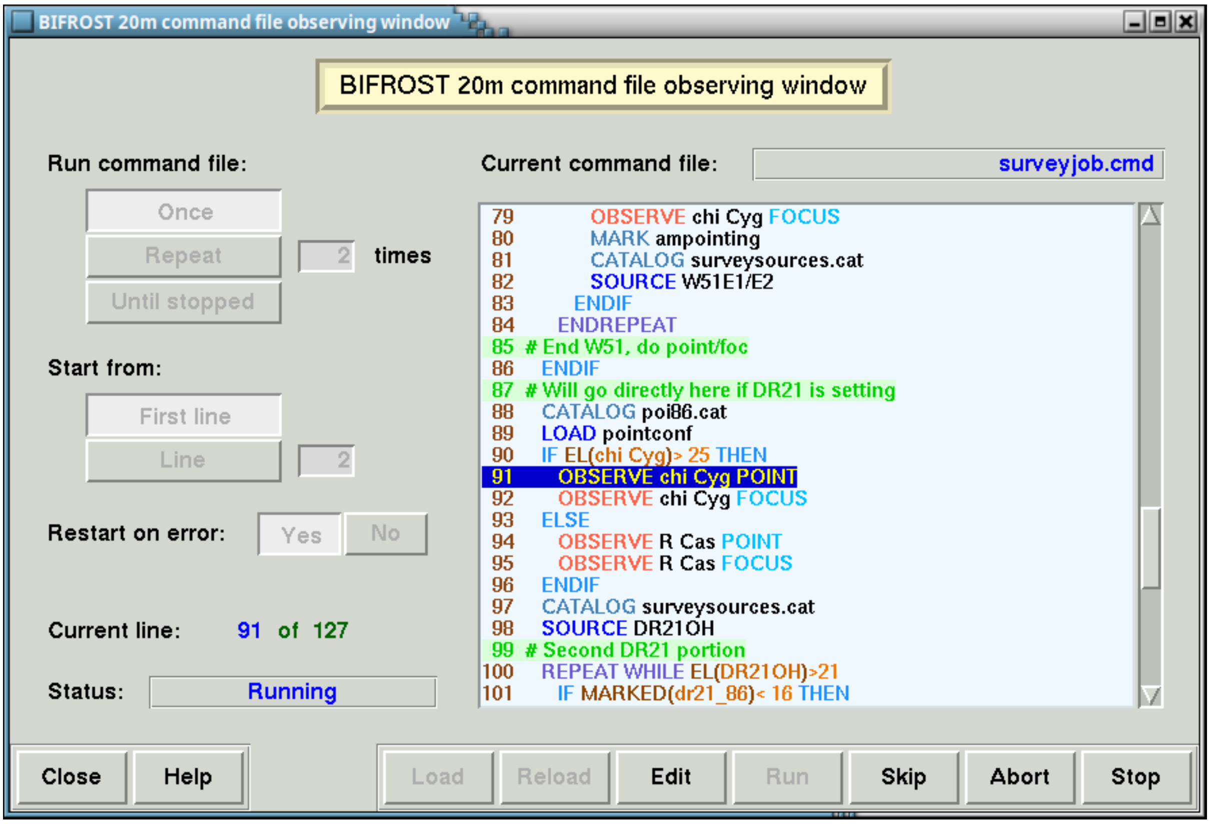
There are seven action buttons in the lower right corner:

- The **Load** button opens a window where you can select which command file to load.
- The **Reload** button reloads the last loaded file without asking you for its name. This is useful to reload the same file after you have modified it.
- The **Edit** button will launch the BIFROST command file editor where you can modify your command file. Note that the editor is running as a stand-alone program! This means that BIFROST does not know about any changes you may make to the command file. You thus have to save the changes in the command file editor and then use the **Reload** button to import the changes into BIFROST. It also means that clicking on the **Edit** button several times will start several copies of the editor.
- The **Run** button will start execution of the command file from the start line you have selected.
- The **Skip** button is only available when running command files. It will abort the current line being executed but it won't stop the execution of the command file, which will continue on the next available line.
- The **Abort** and **Stop** buttons have the same function as during interactive observations. The **Abort** will terminate the command file execution immediately while the **Stop** button will terminate the execution at a convenient time, typically when the current command or observing loop has finished. The function is the same as for the **Abort** and **Stop** buttons in the observation log display window.

During command file execution

There are a number of changes in the command file observing window once you hit the **Run** button to start running a command file.

The buttons for setting up the command file execution are all disabled and the three action buttons for stopping the execution or skipping the current command are enabled. The yellow start line marker in the command file box is replaced with a blue marker indicating the line currently being executed. If the command file is too long to fit into the command file box, the box will scroll automatically to keep the blue line with the current line visible.



A command file is running on the 20m telescope. The line currently being executed is indicated with a blue background.

A line will appear above the **Status** box telling you which line currently is being executed. If you have set **Run command file** to something else than **Once**, then a second line will also appear telling you the number of the current loop.

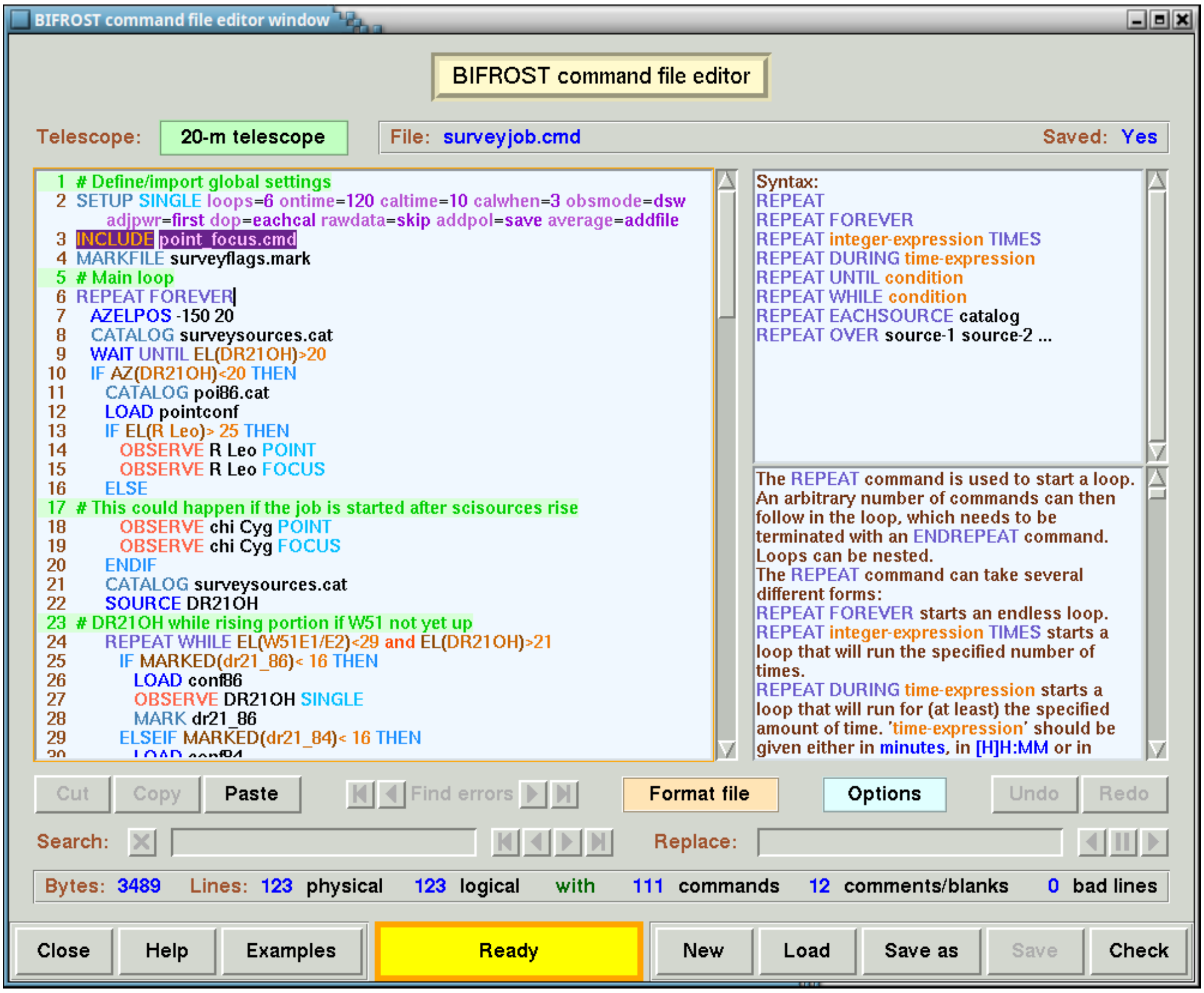
Other modifications are that the **Close** button iconifies the window instead of closing it, and the possibility to comment out lines using the middle-mouse button is disabled.

The command file editor

BIFROST command files are ordinary ASCII-files, and you can thus create and edit them with any editor. However, it is strongly recommended that you do use the command file editor that comes with BIFROST. It won't have all the fancy features of your favourite editor, but it is specially designed to help you writing command files.

Command file names are supposed to end with the suffix "*.cmd*", but it is not a strict rule. If they do, BIFROST will immediately recognize them as command files when you try to load them — if they don't you have to tell BIFROST to show you all available files, otherwise they won't show up in the list of files.

The command editor is always run in stand-alone mode and can be launched from a terminal with the command "`cmdwin`" or if you already have a command file you want to modify: "`cmdwin filename`". The command file editor can also be opened with the



A command file is being edited in the BIFROST command file editor*. The cursor is on line 6 which contains a "*`REPEAT`*"-statement, hence, the two boxes to the right provide help on that particular command.*

Edit button in the command file observing window. Note though that when other stand-alone tools like the quick-look facility or the monitor system are used within a BIFROST observing session, those services are integrated into the system — this is not true for the command file editor which is still run in stand-alone mode. That means that if you click on the **Edit** button several times, you will launch multiple copies of the command file editor. Also, when you save a modification in the editor, it is **not** exported back automatically to the BIFROST observing session. You will have to reload the modified file with the **Reload** button in the command file observing window.

Command file editor features

The command file editor is dominated by three big boxes. The biggest box to the left is the editor box where your command file is shown and where you can edit it. The command file will be shown as a colour-coded listing just as when you observe, and the syntax of the command file is being checked while you type. Line numbers are added automatically and must not be present in the command file itself.

When you type something, the editor will wait a second or so and then it will check the syntax of what you wrote. If there is some problem the line will be flagged as bad and will be shown in olive-coloured text with the (first) offending word shown in red. Sometimes a ?-sign appears. This sign tells you that there is something missing in that place. Type in what is missing and the ?-sign will disappear. If the automatic syntax checking for some reason doesn't work, you can use the **Check** button at the bottom right corner to force the editor to check the syntax again.

The upper right box is the syntax box. Its contents will be modified dynamically depending on where the insertion cursor is located in the command file and what the line it points to contains. If the line has a valid command, this box will show you the syntax of that command and its arguments. If you are editing an incomplete command, you will instead see a list of possible command completions here. If you are at an empty line, a list of all available commands will be shown.

The lower right box is the help box. Its contents will also be modified dynamically depending on where the insertion cursor is located in the command file and what the line it points to contains. This box explains commands and their arguments. It also gives examples of how they can be used.

Below the boxes there are two lines with buttons for copy and paste, search and replace, quick jump to lines with errors, undo and redo buttons as well as two menus. The **Format file** action menu contains a number of formatting commands that can be used to switch between uppercase and lowercase, wrap or unwrap long lines and indent the command file. The **Format using 'Option' menu** action will apply all formatting command in one go, using the options that have been set up in the **Options** menu. Don't be afraid of trying the formatting options — you can always use the **Undo** button, if you don't like the result.

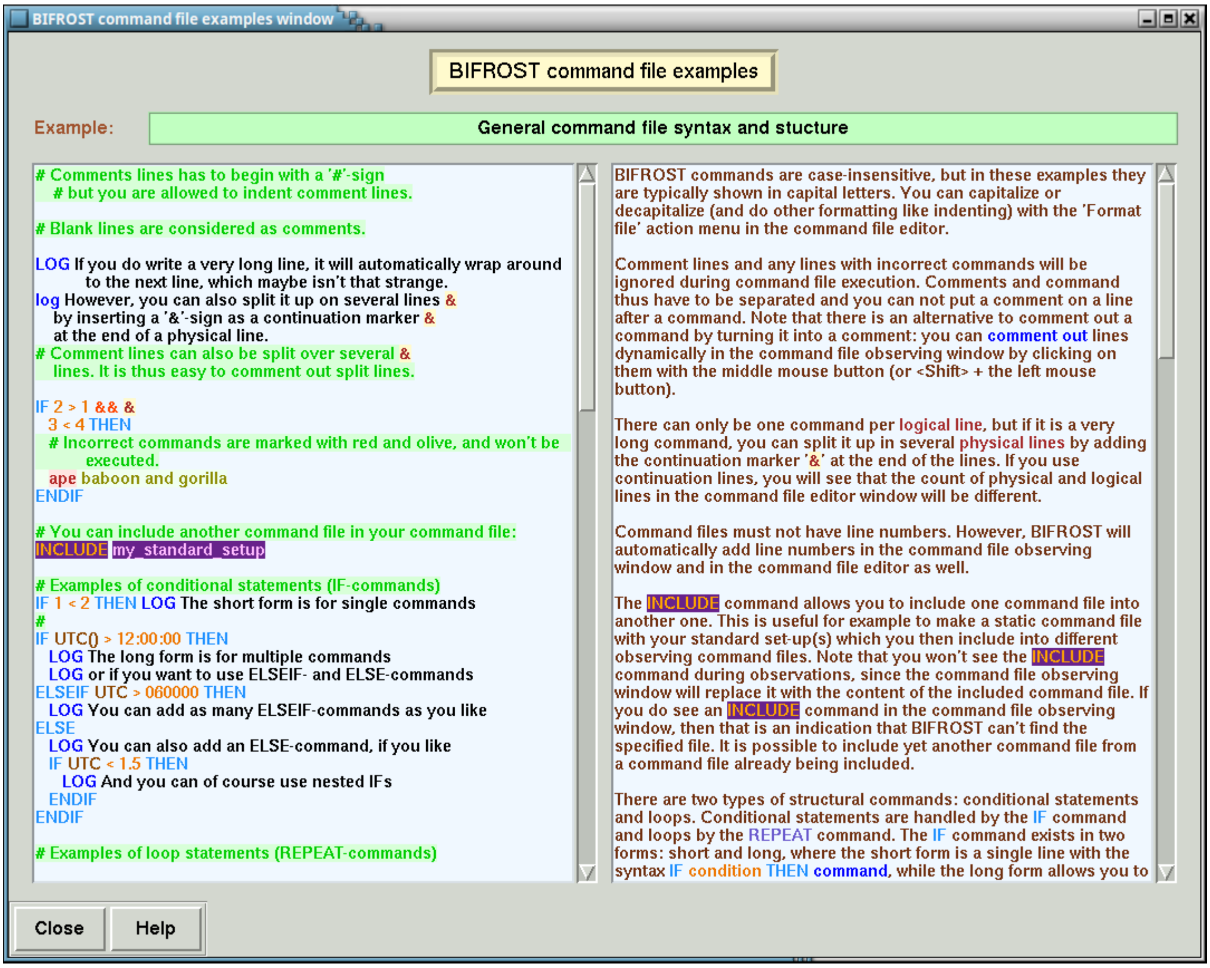
The **Options** menu also contains some options for how you want the search function to work and an option to save a back-up of the original file.

The command file editor keeps track of all modifications you make to a file, so you can use the **Undo** button repeatedly to get all the way back to the version of the file you started with (and back again with the **Redo** button if you want).

There is a third line at the bottom of the window providing some statistics on the command file.

Command file tutorials

Although the command file editor provides extensive help on each command, an observer may find it useful to read some tutorials explaining different concepts and showing larger pieces of example code. A set of tutorials is available in the BIFROST command file examples window which is opened by the **Examples** button in the BIFROST command file editor.



The BIFROST command file examples *window contains a set of tutorials with example code and explanations.*

Command file syntax

The command files use a simple syntax which has been developed specifically for BIFROST. The idea is that it should be easy to use, yet flexible enough to allow for the construction of command files that can run for days.

General syntax

Each line in the command file should contain either a command or a comment. A comment line starts with the "#"-sign. Blank lines are counted as comments. You are free to indent the lines as you wish.

If you have long commands and don't like long lines, you can split up the command on several lines by adding a "&"-sign as a continuation marker at the end of each broken line. The "&"-sign works also for comments, so if you put a "#"-sign in front of a multi-line command, the entire command is turned into a multi-line comment instead.

Even if a command is broken up over several *physical* lines, BIFROST will still count them as one *logical* line. Each physical line gets its own line number, but the line numbers of the continuation lines will have a different colour (yellow).

Commands are not case-sensitive so you can use uppercase or lowercase as you wish with a few exceptions. Names of files (for example configuration files) have to be written exactly as the files are called. The same is true for source names that need to be written exactly as they are written in the source catalogs. Also, values for observing scheme parameters should be written exactly as they appear in the documentation.

Line numbers must not be present in a command file. However, both the command file editor and the command file observing window dynamically add line numbers when displaying the command files. The main purpose of the line numbers is to keep track of the lines when the command file is executed.

Splitting up a command file

You don't need to have everything in one file. You can split it up in different files which you then include in your main command file with the `INCLUDE`-command. You can thus have one file with your set-up, another one for pointing and yet another for your observations.

Nested use of the `INCLUDE`-command is allowed as long as it doesn't form an endless loop. Note that you won't see any `INCLUDE`-commands when you load a split-up command file into the command file observing window, since BIFROST then will merge all parts into a single script for execution.

Control logic

There are two important commands available to control the flow of a command file. The `IF-ELSEIF-ELSE-ENDIF`-statements are used to execute different parts of the code conditionally. The `ELSEIF`- and `ELSE`-statements are optional, and you can use as many `ELSEIF`-statements as you want. You can also nest `IF`-statements inside each other as you wish.

If you just want to run a single command conditionally, you can use the short form: "`IF` *`condition`* `THEN` *`command`*" which doesn't need any `ENDIF`.

You can loop over a piece of code with the `REPEAT-ENDREPEAT`-statements. Control of the looping can be done in different ways and you specify how by adding some further argument to the `REPEAT`-statement. The simplest loop type is "`REPEAT FOREVER`" which starts an infinite loop (the argument `FOREVER` can be omitted). If you want to loop a certain number of times, you can use "`REPEAT` *`number`* `TIMES`" where *`number`* has to be an integer or an integer expression. If you want to loop for a certain amount of time, you can use "`REPEAT DURING` *`hh:mm`*". There is also the possibility to run the loop while/ until a certain condition is met using "`REPEAT WHILE` *`condition`*" or "`REPEAT UNTIL` *`condition`*". For a conditional loop you actually have the freedom of putting the `WHILE` or `UNTIL` condition after the `ENDREPEAT`-statement instead — doing that will guarantee that the loop is executed once before the first evaluation of the condition is done.

For the 20m and 25m telescopes as well as the OTTs, there are also two more special types of loops, where you loop over a set of sources. The sources can either be given explicitly in a list ("`REPEAT OVER` *`source-1 source-2 ...`*") or by giving the name of a source catalog ("`REPEAT EACHSOURCE` *catalog*"). Both these forms accept further arguments to modify the order of the sources in the catalog or in the list. The argument "`RANDOM`" can be used to randomize the order and "`SORT`", with further arguments, can be used to sort the sources. The "`SORT`" argument needs a sorting instruction which can be "`DECLINATION`", "`MINELEVATION`", "`MAXELEVATION`", "`ELEVATIONDIFF`" and "`MARKS`", which sorts the sources in order of lowest declination, lowest elevation, highest elevation, smallest difference in elevation with respect to a specified reference source, and lowest mark (see the chapter "*Keeping track of things with marks*"), respectively.

Loops can be nested in each other as you wish with one exception (for the 20m and 25m telescopes as well as the OTTs): you are not allowed to put two `REPEAT` commands which loop over a list of sources (either `EACHSOURCE` or `OVER`) inside each other.

There are two control commands that only can be used inside loops: `CONTINUE` which jumps back to the beginning of the loop and starts the next iteration (if there will be another iteration), and the `BREAK` command that will break out of the loop. These two commands only affect the innermost loop, if you have several loops nested inside each other.

If you need to wait, there is a set of commands for that: `UTCWAIT` and `LSTWAIT` will both wait until the specified time has arrived, while the `WAIT` command can be used to wait for a specified amount of time. You can also use the `WAIT` command for a conditional wait by using one of the two forms: "`WAIT WHILE` *condition*" or "`WAIT UNTIL` *condition*".

If you need to keep track of time, you can use the command "`TIMESTAMP` *stamp*" to set a time stamp and you can then check how much time has passed since the time stamp was set with the "`SINCE(`*stamp*`)`" function.

The `QUIT` command can be used to terminate the command file execution at that point.

Conditional expressions

There are several commands in BIFROST that use conditional expressions; `IF`-statements being the obvious example, but there also commands like "`REPEAT WHILE` *condition*" or "`WAIT UNTIL` *condition*". Conditional expressions typically consists of checks of built-in functions or logical expressions using built-in functions. You can combine expressions arbitrarily using `AND`, `OR`, `NOT` and parentheses (to change evaluation order).

The built-in functions available differ depending on which BIFROST system is being used, and they will be listed further on in this section. Typical functions include functions that return the current time (UTC or LST), the time that has passed since a timestamp was set and functions that handles variables, marks and block flags. For the telescope systems, there are a number of functions that can be used to check whether a source is observable or not and to find out where it is, as well as functions to check results of pointings and focusings and get data from the FITS-headers. There are also functions to check the weather or the last Tsys values, and functions that return internal information such as which loop is being executed, what the name of the current source and catalog are, and status of maps being observed. There is also a general function that can be used to access any parameter monitored by the BIFROST system.

In the following example, the IF-statement will be true if the UTC is in the range 22:15 and 03:30 and the last Tsys-measurement gave a value below 350 K and either it has passed more than 90 minutes since the timestamp "P3" was set or the elevation of the source "R Cas" is higher than 30°:

```
IF UTCRANGE(22:15:00 - 03:30:00) AND TSYS() < 350 AND
   ( SINCE(P3) > 90 OR EL(R Cas) > 30 ) THEN
```

Crash handling

There are two commands to deal with crashes and possible automatic restarts of the command file in case of a crash. The AFTERCRASH command can be used to register some clean-up actions to take in case the command file execution terminates with a crash.

The ALLOWRESTART command can be used to indicate when, how and where a restart of the command file should take place in case of a recoverable crash. Note that all ALLOWRESTART commands will be ignored unless the **Restart on error** option in the command file observing window is set to **Yes**.

Feedback to the observer

The commands LOG and NOTE can be used to add a message to the log, with the difference that NOTE also adds the message to the diary (for the 20m and 25m telescopes). The command ALERT also writes a message to the log, but it will also send the message via email to one or more recipients which you need to register with the ALERTME command.

Once you have registered email addresses with the ALERTME command, BIFROST will also send emails automatically when the command file terminates or if it crashes (and possibly restarts). There are also several configuration commands setting limits (for example high Tsys-values or large pointing offsets) that can take an extra ALERT argument which instructs them to send emails if the limits are violated.

It is possible to use all built-in functions as part of a message. The following message reports the last measured Tsys-value for the current source and also tells at which elevation it was observed:

```
LOG Tsys on SOURCE() at elevation EL(*) is TSYS().
```

Configuration commands

The most important configuration command is LOAD which is used to load a configuration file. There are no commands allowing you to set a frequency or change a backend resolution directly — all configurations have to be done with configuration files.

Configuration files used in command files have some limitations on their possible content. When saving a configuration file in the save configuration file window, the content is divided in two parts by an orange line. Content above that line can be included in a configuration file used in a command file, while content below the line will be ignored. Observing scheme parameters is one example of content that is not included, since the command files have other ways for setting up those parameters.

There are a number of other configuration commands as well, especially for the telescopes: ADJUSTPOWER can be used to adjust the power levels, SETPOINTING to set pointing offsets and, for the 20m telescope, SETSUB or SETFOCUS to control the subreflector. For pointing and focusing measurements, you can put limits on the results with MAXPOINTING and MAXFOCUS.

There are several commands to start up or shut down the telescopes: UNSTOW, STOW, INITIALIZE, ANTENNA and PARK. The slew speed on the OTTs can be controlled with the MAXSPEED command.

Selecting a source

When observing interactively, you can type in the coordinates of a source manually, but when running a command file there is no such option and you have to have all your sources in catalog file(s). You select a catalog with the CATALOG command, and you can then select and send the telescope to a source from that catalog with the "SOURCE" command. You can override the catalog selection by giving the catalog name together with the source name:

SOURCE *source* IN *catalog*

The same syntax can also be used in functions that expect a source name as an argument. There are also two wildcards that can be used instead of a source name: "*" which means the currently selected source and "@" which means the loop source inside a "REPEAT OVER" or a "REPEAT EACHSOURCE" loop.

The SOURCE command sends the telescope to the source regardless of whether it is up or not. To avoid this, it is usually better to use the OBSERVE command (described in the "*Observing commands*" chapter below), which only selects a source, if it is actually possible to observe it.

If you want to send a telescope to a fixed position, you can use the AZELPOS command (and on the 25m telescope also the POLDECPOS command). There are also a few preset positions which you can go to with the PARK command.

Observing commands

Each observing scheme has its own observing command. The different observing parameters are given in the format "parameter = value". The value can be enclosed with "" or {}. **Note** that while parameter names are case insensitive, values are *not*, and any string value has to written exactly as BIFROST expects it.

Default values will be used for any parameter not specified. You can change the default values with the SETUP command described in the next chapter.

A command to perform a simple calibration may look like:

```
CAL loops=1 caltime=10 adjpwr="first"
```

Lists of all observing schemes and their associated parameters are given at the end of this section.

For the 20m and 25m telescopes, you can call an observing command via the OBSERVE command, which is a wrapper that adds some intelligence to the observing. The basic syntax is:

OBSERVE *source observing-scheme*

but you can add more conditions. Here is an example where we want to observe the source R Cas for 1.5 hours:

```
OBSERVE R Cas IN my_catalog DURING 01:30 SINGLE loops=5 calwhen=5
```

which already hints at some advantages by using `OBSERVE`, but there are more built into the command. Consider the following code:

```
SOURCE R Cas
SINGLE loops=15 calwhen=5
```

Here, the telescope will be sent to R Cas regardless of whether the source is up or not. In case it is not, the command file will hang on the `SINGLE` command until the source is above the horizon and it will then perform the observation regardless of the weather situation. If we instead use `OBSERVE` and its support commands (`WINDOW`, `MAXTSYS` and for the 25m telescope also `MAXWIND`), we could write:

```
WINDOW 25 75 20
MAXTSYS 500
OBSERVE R Cas SINGLE loops=15 calwhen=5
```

In this case, the telescope will only be sent to R Cas provided that it is inside the observing window defined by the `WINDOW` command, which in this example is set to be within $25^\circ \leq El \leq 75^\circ$ and R Cas has to stay inside this range for at least 20 minutes (excluding the slew time to get there). Then, if the observation is started, the Tsys will be checked and if it is found to be above the 500 K limit set with the `MAXTSYS` command, then the observation will be paused and it won't resume until the Tsys is back below the limit (provided that R Cas still is within the window).

The `OBSERVE` command exists in forms for a single execution of the specified observing scheme or for repeated observations depending on the condition given. There are three single-execution forms:

```
OBSERVE source observing-scheme
OBSERVE source IF condition THEN observing-scheme
OBSERVE source INWINDOW min-elevation max-elevation minutes-observable
        observing-scheme
```

where the last form can be used instead of the `WINDOW` command. The following repeated-execution forms are available:

```
OBSERVE source WHILEUP observing-scheme
OBSERVE source UNTIL condition THEN observing-scheme
OBSERVE source WHILE condition THEN observing-scheme
OBSERVE source REPEAT number TIMES observing-scheme
OBSERVE source DURING time-period observing-scheme
```

The SETUP command

The `SETUP` command can be used to set up new default values for the observing commands. It uses the same syntax as the observing commands but the word `SETUP` should precede the name of the observing scheme. Thus instead of making a calibration with the command:

```
CAL loops=1 caltime=10 adjpwr=first
```

you could set up the parameters first with a `SETUP` command:

```
SETUP CAL loops=1 caltime=10 adjpwr=first
CAL
```

You can always overrule the defaults set by a `SETUP` command by adding another parameter value directly to the observing command. Here, the defaults are used except for the integration time which is increased to 15 seconds:

```
CAL caltime=15
```

It is recommended that you use the `SETUP` command to set up default values for all parameters even if you are happy with the preset default values. If you do that, the behaviour of the command file will not change if the default values in BIFROST are changed for some reason.

Keeping track of things with marks

Marks are simple integer flags that can be set and incremented to indicate if some action has been taken. A typical usage on the telescopes is to flag if a source has been observed or keep track of how many times it has been observed. A mark is an arbitrary string that is case-sensitive and may include spaces. You can use ""-signs to enclose the mark name but they are not necessary. A mark is set with "`MARK mark`". If the mark already exists, then its value is incremented by one. A mark can be unset with the command "`UNMARK mark`". If you want to clear all marks at once, you can use the `CLEARMARKS` command.

The value of a mark is checked with the `MARKED(`*`mark`*`)` function. The function returns "`0`" if the specified mark is not set.

As marks typically are used for book-keeping, you most likely want to save them in a mark-file. You open a mark-file with "`MARKFILE` *`mark-file`*" and close it with `CLOSEMARKFILE`.

Since a main purpose of marks is to keep track of which sources have been observed, there is a way to have all `OBSERVE` commands automatically set/increment marks corresponding to the name of the source being observed. This function is switched on with the command "`AUTOMARK ON`" and can be switched off with "`AUTOMARK OFF`". The name of the mark set or incremented is the name of the source. The mark will be set/incremented each time an OBSERVE command starts a new observation.

The name of the mark can be modified with the "`AUTOMARK PREFIX` *`prefix-string`*" command. The name of the marks will then be the specified string with the source name appended. The use of prefix can be useful if you want to observe the same source with different set-ups. The use of a prefix can be cancelled with the command "`AUTOMARK PREFIX NONE`"

Using variables

Variables can be used in may places. Variables can either be *local* in which case they are reset each time a command file is loaded or *stored* variables which are remembered between different command files and between observing sessions. The command `LET` is used to assign a local variable, while the command `SET` is used for stored ones.

The syntax is "`LET variable = expression`". **Note** that you need to put space around the "`=`"-sign. Variable names are not case-sensitive. A variable name can consist of letters and digits as well as the special characters "_", "`-`" and "`+`".

The value assigned could either be a constant or an expression which will be evaluated before the assignment. The type of the variable (integer, floating point or string) does not need to be declared as the type is set automatically depending on the value being assigned. The assignments can be any valid Tcl/Tk expression, and they can include the use of BIFROST functions or calls to internal BIFROST functions.

The `VAR(variable)` function is used to retrieve the value of a variable. Variables can be used in all expressions and in most places where BIFROST expects a string or a numerical value. However it can not be used for arguments of flag-type; for example, in the command "`LOAD configuration-file ADJUST`", you could use a variable to hold the name of the configuration file to load, but the "`ADJUST`" flag can not be replaced by a variable.

Numerical variables can be incremented with the "`INCR variable`" command. The command will increment the specified variable with "`+1`" unless you specify another value with "`INCR variable value`". This command is basically a shorter way of writing "`LET variable = VAR(variable) + value`". The value can be an arbitrary integer or decimal constant, either positive or negative.

You can erase all variables with the `CLEARVARS` command. **Note** though that this command removes all local variables as well as all stored variables used in *ANY* command file.

The use of variables can be very convenient but it also adds a bit more responsibility to make sure that a command file is correctly written. When not using variables, BIFROST can perform sanity checks on arguments already in the command file editor or when loading a command file into BIFROST, but when variables are used this check has to be deferred until the variable is actually used during command file execution.

Switching between different projects

It is possible to switch automatically from one project to another when running command file observations, which is very useful if there are several projects running at different LST intervals and they are using command file observing. The command is "`PROJECT` *`project-ID command-file-name observer name(s)`*", where *`command-file-name`* has to be an existing command file in the main project directory for project *`project-ID`*.

When a "`PROJECT`" command is encountered, the execution of the current command file is terminated. The system is left running as it is with one exception: if the SIS 3/4-mm receiver has been used on the 20m telescope, then it will be shut down. A restart script is launched and BIFROST is then exiting in the normal way. Once the restart script sees that BIFROST has closed down, it will start a new observing session with the project-ID and observer name provided, and tell that observing session to automatically start up command file observing using the provided command file.

Note that if something would go wrong with the start-up of the new observing session, there won't be any email messages sent out, since BIFROST won't know where to send them. It is thus a good idea to verify that the hand-over has been successful. One way of doing that is to include an "`ALERT`" command in the beginning of the command file that sends a message confirming that the new project indeed has started up.

There is no way of changing project within a BIFROST observing session, hence the method of using a restart script. This is a result of the way BIFROST has been implemented and it is a legacy design carried over from Arecibo, where some staff members strongly opposed the idea of allowing project changes within an observing session, fearing that unwanted settings from the previous project could affect the next one, if that would have been allowed.

Strategies for writing command files

It is a good strategy to divide up your command file in several smaller units which you call using the `INCLUDE` command. You can put set-up commands in one file, pointing and focusing procedures in one file and the actual observing procedure(s) in another file(s). By doing that you get smaller pieces of code which will be easier to work with, and it is also easier to reuse files with, for example, set-ups and pointing procedures for other observations.

One important aspect, if you want to write command files that can run automatically for an extended period, is to make it easy to restart. This is useful if you need to interrupt the execution of the command file for some reason, but it is even more important if you want BIFROST to recover from temporary glitches (typically network or communication problems), which do happen from time to time. When allowing **Restart on error**, BIFROST will pause for 1 minute before restarting the command file. It is thus useful to have a `LOAD` command in the beginning of a command file, so that BIFROST loads a well-known configuration. This will also check that all hardware is working, since the configuration will fail if that is not the case.

The **Restart on error** option only allows **one** attempt to restart a command file, so if you want your command file to handle multiple crashes, you will have to use the `ALLOWRESTART` command to clear the flag telling BIFROST that it already has made one restart attempt. A good strategy is to put an `ALLOWRESTART` command **after** a `LOAD` command. The idea is that if the crash was due to a temporary glitch, then the `LOAD` command will succeed and it is OK to continue observing and be prepared for another glitch. However, if the crash was due to a real failure, for example a backend that has crashed and needs staff intervention, then the `LOAD` command will fail and the command file execution will not go into an endless loop, where it in vain tries to configure the crashed backend once per minute.

A command file example

In this chapter you will be shown a complete example of a command file that will illustrate a number of different commands and how they can be used to create a command file that can run a fully automatic observing session.

```
# Example command file to observe three sources
ALERTME baboon@monkeys.se
WINDOW 20 75 15
MARKFILE monkey_marks
AUTOMARK ON
# Set up the observations
INCLUDE pointing_setup
SETUP SINGLE loops=15 ontime=60 offtime=60 caltime=10 calwhen="5"
        obsmode="fsw" dop="eachcal" adjpwr="first" useplm="dontuse"
        rawdata="skip" addpol="save" average="addfile"
# Start the observations when the first source is above 20 degrees
CATALOG monkey_catalog
WAIT UNTIL EL(baboon) > 20
ALERT Starting observations since source 'baboon' is up!
ANTENNA ON
ALLOWRESTART
AFTERCRASH PARK SHUTDOWNSIS
# Start the observing loop with regular pointings every three hours
REPEAT WHILE EL(baboon) > 20
  # Do a pointing and focusing, if it is time to do that
  IF SINCE(P1) > 03:00 THEN
    INCLUDE pointing_procedure
  ENDIF
  # Select the source to observe
  IF ISUP(chimpanzee) AND MARKED(chimpanzee) < 12 THEN
    SOURCE chimpanzee
  ELSEIF ISUP(gorilla) THEN
    SOURCE gorilla
  ELSE
    SOURCE baboon
  ENDIF
  LOAD monkey_config IFNOTLOADED
  MAXTSYS 800
  OBSERVE * SINGLE
ENDREPEAT
# All the sources are down so shut down the receiver, park the telescope and
        finish up
SHUTDOWNSIS
PARK SKYLIFT
ALERT The source 'baboon' is below 20 degrees and the telescope is now
        parked!
# End of command file
```

```
# Pointing set-up --- file "pointing_setup.cmd"
MAXPOINTING 10 IGNORE ALERT
MAXFOCUS -1.0 +1.5 IGNORE ALERT
SETUP POINT loops=3 azdist=21 eldist=21 ontime=15 offtime=15 caltime=10
        calwhen="3" obsmode="ssw" dop="eachcal" adjpwr="first"
        rawdata="skip" addpol="save" backend="SPEa"
        baseline="nobaselines" startpos="current" apply="auto"
        limit="relative"
SETUP FOCUS loops=2 dist=3.5 ontime=15 offtime=15 caltime=10
        calwhen="3" obsmode="ssw" axis="focus" dop="eachcal"
        adjpwr="never" rawdata="skip" addpol="save" backend="SPEa"
        baseline="nobaselines" startpos="current" apply="auto"
# End of pointing set-up
```

```
# Pointing and focusing procedure --- file "pointing_procedure.cmd"
MAXTSYS 600
IF INWINDOW(GX Mon IN poi86, 25, 70, 15) THEN
  LET source = GX Mon
  LET ontime = 25
ELSEIF INWINDOW(R Leo IN poi86, 20, 70, 15) THEN
  LET source = R Leo
  LET ontime = 15
ELSE
  LET source = R LMi
  LET ontime = 20
ENDIF
SOURCE VAR(source) IN poi86
LOAD pointing_setup
OBSERVE * POINT ontime=VAR(ontime)
IF TSYS() < 600 && POINTOK THEN
  FOCUS ontime=VAR(ontime)
  IF UTCRANGE(07:00:00 - 23:00:00) THEN ALERT Tsys = TSYS() at EL(*) for
      SOURCE() with result AZPOINT(), ELPOINT() and ZFOCUS() mm.
  TIMESTAMP P1
ENDIF
# End of pointing and focusing procedure
```

This command file example with the main program to the left and the two command files being `INCLUDE`*d shown to the right is described in the text.*

Goals of the command file example

The goal of this command file example is to use the 20m telescope to observe three sources called baboon, gorilla and chimpanzee with the SIS 3/4-mm receiver. The three sources reside in a local source catalog called monkeys_catalog.cat and the science configuration (choice of receiver, frequency, backend and so on) has been saved in a configuration file called monkey_config.conf. Of the three sources, baboon has the highest declination and is thus the one that is up for most hours. The other two sources are more important to observe, but they are up for fewer hours and, unfortunately, they are up at the same time as baboon, so the strategy will be to observe them as much as possible and observe baboon when the others are not accessible. We are going to use frequency switching and each observing block will consist of 15 one-minute integrations. The source chimpanzee is the highest priority, but we expect to reach the science goal for this source with twelve observing blocks so we are going to use markers to keep track of the number of blocks we have observed of this source.

The observing session will start before any of the sources is up so the telescope will be parked and the command file will have to wait until a proper moment and then start up and configure the system for observations. Likewise, we want the command file to park the telescope and shut down the system after the observations are completed. During the observations, we want to point and focus every three hours, and we are going to use a timestamp to keep track of the time.

Description of the main command file in the example

The command file starts by setting up an email address for messages and the observing window used by the OBSERVE command. Observations will only be performed if a source is located between 20° and 75° elevation and will stay in that range for at least 15 minutes (excluding slew time). We then open a mark-file and tell BIFROST that we want the OBSERVE to set marks automatically each time it completes an observing block.

Next we set up the defaults for the observations. The defaults for pointing and focusings are defined in a separate command file which we just INCLUDE here, while the observations of the scientific sources are specified: each observation block consists of 15 one-minute integrations in frequency-switching mode with calibrations between every five integrations.

We then select the catalog with the sources and tell BIFROST to wait until the first source is above 20° elevation. After that we send a message that the observations are about to start, and power up the telescope, in case it was powered down.

We then tell BIFROST that we want the command file to be restarted automatically in case of a recoverable crash. We also tell BIFROST that in the case of an unrecoverable command file crash, we want the telescope to be parked and the SIS 3/4-mm receiver to be shut down.

The real observing loop starts then and we will stay in this loop as long as the source baboon is observable. The first thing we do in the loop is to check the "P1" timestamp to see if we need to do pointing and focusing. If that is the case, we INCLUDE another command file to do that for us. We then check which source we are going to observe and send the telescope there: the first priority is chimpanzee provided that is up (as defined by the WINDOW command) and we still haven't performed the 12 observing blocks as we planned. Our second priority is gorilla provided that it is up — otherwise we will observe baboon.

We now load the science configuration, but with the conditional flag IFNOTLOADED, so that we do skip this step if the science configuration is already loaded. The LOAD command will thus only be executed when we start the command file and after a pointing and focusing.

The next thing is to set a limit for the OBSERVE command to only observe when the Tsys is less than 800 K. We do have the MAXTSYS just before the observation, since this limit is reset to another value when we are doing pointing and focusing.

The final command in the loop is then the actual observation command using the OBSERVE wrapper to pause the observations if the Tsys goes above the 800 K limit and also make sure that we only observe if the source really is inside the window defined by WINDOW. The "*" wildcard is used to tell OBSERVE to use the currently selected source. The OBSERVE command will automatically increment a marker with the name of the source when the observing block has been successfully completed.

As we leave the observing loop, there are a few clean up commands: we shut down the receiver and tell BIFROST to park the antenna before sending a final email to the observer.

This command file could easily be adapted to keep running the observations for several days at once: it is just to add a REPEAT FOREVER command in the beginning of the file and another ENDREPEAT at the end.

Description of the pointing set-up in the example

The pointing set-up command file listed on the top right consists of just a few lines. There are two SETUP commands to set up default values for the pointing and focusing measurements. This is a standard set-up with three loops for the pointing and two loops for the focusing. There are also two commands MAXPOINTING and MAXFOCUS used to set limits for the results obtained. Pointing offsets larger than 10" and focusing results outside the range $-1.0 \leq focus \leq +1.5$ mm will be flagged as bad. The IGNORE and ALERT options tell BIFROST that any result outside the allowed range should be ignored and an email should be sent to the observer reporting that the pointing/focusing produced a strange result.

Description of the pointing procedure in the example

The command file performing the pointing and focusing shown at the bottom right starts by resetting the allowed max Tsys to 600 K. We then have to select which source to use, and here we will use variables to store the selected source and the integration time needed. We use the general catalog poi86.cat, but we don't select it (to not disturb the catalog selection in the main command file) so we will have to give the catalog name explicitly each time we specify a pointing source. The preferred source is GX Mon, but it is a bit weak so we will not use it if it is below 25° and we will need a longer integration time of 25 seconds. Our second choice is R Leo which is a strong source only needing 15 seconds of observing time, but it is not up all the time either, so we select R LMi as a fall-back option.

After selecting a source and sending the telescope to it, we load the pointing configuration and tune the receiver with the `LOAD` command. When using the SIS 3/4-mm receiver, it is useful to select the source before tuning, since the tuning then will be done for the proper Doppler correction from the beginning minimizing the need to reoptimize the tuning, if the actual Doppler correction differs a lot from the value used during the tuning.

We then try to run the pointing measurement on the currently selected source with "`OBSERVE * POINT ontime=VAR(ontime)`". We use all the default parameters set up in the file "*pointing_setup.cmd*" except for the `ontime` parameter where we instead use the value we saved in a variable. If the weather is too bad, this command will hold off the pointing measurement until the Tsys is back below the 600 K limit we declared in the beginning. There is a possibility that the pointing measurement never is performed and that would happen if the weather stays bad and the source moves outside the allowed observing window as defined by the "`WINDOW 20 75 15`" command defined in the main command file.

After the pointing, we check that the Tsys is indeed less than 600 K and that the pointing measurement was successful, and if so we run the focusing measurement with the `FOCUS` command. Here we don't use the `OBSERVE` wrapper because we have done the pointing so we want to do the focusing immediately regardless of what the Tsys may turn out to be or if the source is just about to leave the observing window. After the focusing, we do send a report to the observer with the results of the pointing and focusing, provided that it is not in the middle of the night.

The final thing to do is to set the permanent timestamp "`P1`" to indicate the time when we performed the pointing and focusing. This timestamp is checked with the `SINCE(P1)` function in the main command to determine when it is time to perform the next pointing and focusing.

All BIFROST commands

The following are four lists of all available BIFROST commands. They are divided into four lists depending on the type of commands. The first list are *structural commands* which are used to control the flow of a command file. The second list is all general commands available in all BIFROST systems. Then follows a list of all commands which only are available in some BIFROST systems, and finally is the list of all the observing commands.

Note that even if a specific command is available in several BIFROST systems, the arguments it can take may depend on the specific system. See the on-line documentation in the command file editor for details.

All structural BIFROST commands

INCLUDE	Include another command file	*all*
IF	Start a conditional branch	*all*
ELSEIF	Start another conditional branch	*all*
ELSE	Start a branch for remaining conditional cases	*all*
ENDIF	End a conditional branch	*all*
REPEAT	Start a loop	*all*
ENDREPEAT	End a loop	*all*

All general BIFROST commands

CONTINUE	Start next iteration of a loop	*all*
BREAK	Break out of a loop	*all*
QUIT	Quit running the command file	*all*
PROJECT	Switch to run a command file for another project	*all*
WAIT	Wait for a certain time or for some condition	*all*
LSTWAIT	Wait for a certain LST-time	*all*
UTCWAIT	Wait for a certain UTC-time	*all*
LET	Assign a local variable	*all*
SET	Assign a stored variable	*all*
INCR	Increment a numerical variable	*all*
CLEARVARS	Clear all variables	*all*
TIMESTAMP	Set a timestamp	*all*
CLEARSTAMP	Clear a timestamp	*all*
MARK	Set or increment a mark	*all*
UNMARK	Clear a mark	*all*
CLEARMARKS	Clear all marks	*all*
MARKFILE	Open a mark-file	*all*
CLOSEMARKFILE	Close a mark-file	*all*

BLOCK	Set an external block flag	*all*
UNBLOCK	Clear an external block flag	*all*
ALLOWRESTART	Allow the command file to be restarted more times	*all*
AFTERCRASH	Register some actions to be performed after a crash	*all*
ALERTME	Register email address(es) for messages	*all*
ALERT	Send an email message	*all*
LOG	Write a message to the log	*all*
NOTE	Write a message to the log and the diary (20m, 25m)	*all*
EXEC	Execute an arbitrary command	*all*

All system-specific BIFROST commands

LOAD	Load a configuration file	*20m, 25m,* CO/O_3, H_2O
ADJUSTPOWER	Adjust the power levels	*20m, 25m,* CO/O_3, H_2O
WINDOW	Set an observing window	*20m, 25m*
CATALOG	Select a source catalog	*20m, 25m, OTTs*
SOURCE	Select and send the telescope to a source	*20m, 25m, OTTs*
AZELPOS	Send the telescope to a fixed position	*20m, 25m, OTTs*
POLDECPOS	Send the telescope to a fixed position	*25m*
WAITTRACK	Wait until the telescope is tracking	*20m, 25m, OTTs*
SETPOINTING	Set pointing offsets	*20m, 25m, OTTs*
CLEARPOINTING	Clear pointing offsets	*20m, 25m, OTTs*
SETSUB	Set subreflector parameters	*20m*
SETFOCUS	Set subreflector Z-focus	*20m*
CLEARFOCUS	Clear subreflector Z-focus	*20m*
MAXPOINTING	Set max pointing offsets for POINT	*20m, 25m*
MAXFOCUS	Set max focus range for FOCUS	*20m*
MAXTSYS	Set Tsys limit for OBSERVE	*20m, 25m*
MAXWIND	Set wind limits for OBSERVE	*25m*
AUTOMARK	Control marks set by OBSERVE	*20m, 25m*
OBSERVE	Smart wrapper for observations	*20m, 25m*
UNSTOW	Unstow the telescope	*25m, OTTs*
STOW	Stow the telescope	*25m, OTTs*
INITIALIZE	Initialize the telescope encoders	*20m*
PARK	Park the telescope	*20m, OTTs*
ANTENNA	Enable or disable the telescope	*20m, 25m, OTTs*
MAXSPEED	Set the max speed for the telescope	*OTTs*
RETUNE	Force the SIS 3/4-mm to retune	*20m*
SHUTDOWNSIS	Shut down the SIS 3/4-mm receiver	*20m*
STOPKOS	Stop the SIS 3/4-mm KOS	*20m*
SETUP	Set default values for an observing scheme	*20m, 25m,* CO/O_3, H_2O
CLEARAVERAGES	Clear average data files	*20m, 25m,* CO/O_3, H_2O
MIRROR	Initialize or set the mirror to a position	CO/O_3, H_2O
CHECKWEATHER	Pause if the weather is not OK	CO/O_3, H_2O

All BIFROST observing commands

RUN	Run backend observing scheme	*20m, 25m, CO/O_3, H_2O*
CAL	Simple calibration observing scheme	*20m, 25m, CO/O_3, H_2O*
SKYDIP	Sky dip observing scheme	*CO/O_3, H_2O*
NORMAL	Normal observation observing scheme	*CO/O_3, H_2O*
TRIPLE	Triple observation observing scheme	*CO/O_3*
BALANCED	Balanced power observing scheme	*H_2O*
POINT	Pointing measurement observing scheme	*20m, 25m*
FOCUS	Focusing measurement observing scheme	*20m*
SINGLE	Single observation observing scheme	*20m, 25m*
MAP	Map observation observing scheme	*20m, 25m*

All BIFROST functions

The following are two lists of all built-in BIFROST functions. The first list contains all general functions available in all BIFROST systems, which is followed by a list of functions which only are available in some BIFROST systems.

Functions marked with "`()`" needs one or more arguments, while functions marked with "`[()]`" can be used with and without an argument, and functions without parentheses don't use any arguments.

See the on-line documentation in the command file editor for details on how to use the functions.

All general BIFROST variables

UTC	Current UTC time in decimal hours	*all*
LST	Current LST time in decimal hours	*all*
UTCRANGE()	"**1**" if current time is within UTC range	*all*
LSTRANGE()	"**1**" if current time is within LST range	*all*
VAR()	Content of specified variable	*all*
SINCE()	Minutes since specified timestamp was set	*all*
LOOP[()]	Current loop iteration number	*all*
LOOPSINCE[()]	Minutes spent in current loop iteration	*all*
SHAME()	Content of specified SHAME-variable	*all*
MARKED()	Numeric value of specified mark	*all*
ISBLOCKED()	"**1**" if external block is set	*all*
BLOCKEDSINCE()	Minutes since external block was set	*all*
RAIN	Rain during last hour in mm	*all*
WIND	Average wind speed in m/s	*all*
GUST	Maximum wind gusts in m/s	*all*
DATE	Current date in the format YYYYMMDD	*all*

All system-specific BIFROST variables

AZ()	Azimuth of specified source	*20m, 25m, OTTs*
EL()	Elevation of specified source	*20m, 25m, OTTs*
SLEW()	Slew time in minutes to specified source	*20m, 25m, OTTs*

ISUP()	"**1**" if source is inside WINDOW window	*20m, 25m, OTTs*
ISOK()	"**1**" if source is inside WINDOW elevations	*20m, 25m, OTTs*
INWINDOW()	"**1**" if source is inside specified window	*20m, 25m, OTTs*
OBSERVABLE()	Minutes source stays inside window	*20m, 25m, OTTs*
RISING()	Minutes until source enters window	*20m, 25m, OTTs*
CATALOG	Name of current source catalog	*20m, 25m, OTTs*
CATALOGPATH	Name of current catalog with full address	*20m, 25m, OTTs*
SOURCE	Name of current source	*20m, 25m, OTTs*
LOOPCATALOG	Name of loop-catalog in REPEAT	*20m, 25m, OTTs*
LOOPSOURCE	Name of loop-source in REPEAT	*20m, 25m, OTTs*
TSYS[()]	Latest Tsys-value	*20m, 25m,* CO/O_3, H_2O
FITS()	Value of specified FITS-keyword	*20m, 25m*
OBSTIME()	Value of FITS-keyword OBSTIME	*20m, 25m*
INTTIME()	Value of FITS-keyword INTTIME	*20m, 25m*
COMPTIME()	Value of FITS-keyword COMPTIME	*20m, 25m*
DATAPEAK()	Value of FITS-keyword DATAPEAK	*20m, 25m*
DATAMEAN()	Value of FITS-keyword DATAMEAN	*20m, 25m*
DATAAREA()	Value of FITS-keyword DATAAREA	*20m, 25m*
DATARMS()	Value of FITS-keyword DATARMS	*20m, 25m*
POINTOK	"**1**" if last pointing was successful	*20m, 25m*
POLPOINT	Pointing offset obtained in polar	*25m*
DECPOINT	Pointing offset obtained in declination	*25m*
AZPOINT	Pointing offset obtained in azimuth	*20m*
ELPOINT	Pointing offset obtained in elevation	*20m*
FOCUSOK	"**1**" if last focusing was successful	*20m*
ZFOCUS	Z-focus value obtained	*20m*
MAPDONE()	"**1**" if the specified map is completed	*20m, 25m*
MAPSTATUS()	map status: new, started or completed	*20m, 25m*
MAPPROGRESS()	string with map progress	*20m, 25m*
OBSERVED	Number of times OBSERVE did observe	*20m, 25m*

Parameters for all different observing schemes

The following pages contain lists of all parameters for all the different observing schemes. Each line contains six columns: the name of the parameter as it appears in the **Observing menu**, the name of the parameter you should use in a command file, the type of the parameter, its range or allowed values, the default value and the systems for which this parameter is used (the word *all* refers to all systems except the OTTs since they don't have any observing schemes). The default values are changed with the `SETUP` command.

Note that parameter name is case-insensitive, but any value that consists of a word has to be spelt exactly as it appears in the list!

Run backend (RUN) observing scheme parameters

Number of loops	**loops**	int	*1 → 1000*	1	*all*
Source on-time	**ontime**	int	*1 → 3600*	10	*20m, 25m*
Source off-time	**offtime**	int	*1 → 3600*	10	*20m, 25m*
Integration time	**inttime**	int	*1 → 3600*	10/30[1]	CO/O_3, H_2O
Calibration time	**caltime**	int	*1 → 3600*	10	*all*
Cal load flag	**usecalload**	word	*yes no*	yes	CO/O_3
Signal elevation	**sigelev**	real	*-90 → 185*	20	H_2O
Ref. elevation	**refelev**	real	*-90 → 185*	90	H_2O
Observing mode	**obsmode**	word	*psw fsw*[2] *ssw*[2]	psw	*20m, 25m*
Observing mode	**obsmode**	word	*tpo frq*	tpo	CO/O_3
Observation type	**obstype**	word	*sig ref csig cref*	sig	*all*
Move telescope	**movetel**	word	*move dontmove*	move	*20m, 25m*
Use calibrator	**usecal**	word	*use dontuse*	use	*20m, 25m*
Adjust power	**adjpwr**	word	*never first eachloop*	first	CO/O_3, H_2O
Fetch system data	**fetchdata**	word	*never first each*	each	CO/O_3, H_2O
Data averaging	**average**	word	*noadd show save*	noadd	*all*

[1] the default is 10 for CO/O_3 and 30 for H_2O
[2] available depending on receiver choice on 20m

Simple calibration (CAL) observing scheme parameters

Number of loops	**loops**	int	*1 → 100*	1	*all*
Calibration time	**caltime**	int	*1 → 180*	10	*all*
Observing mode	**obsmode**	word	*psw fsw*[1] *ssw*[1] *dsw*[1]	psw	*20m, 25m*
Update Doppler	**dop**	word	*never first eachloop*	eachloop[2]	*20m, 25m*
Cal. elevation	**calelev**	real	*-90 → 185*	90	H_2O
Extra cal. spectrum	**calextra**	word	*extra skip*	skip	H_2O
Extra cal. elevation	**extraelev**	real	*-90 → 185*	5	H_2O
Adjust power	**adjpwr**	word	*never first eachcal each2cal each3cal each4cal each5cal*	first	*20m, 25m*
Adjust power	**adjpwr**	word	*never first eachloop*	first	CO/O_3, H_2O
Raw data	**rawdata**	word	*skip save*	skip[3]	*all*

[1] available depending on receiver choice on 20m
[2] the default is "`eachloop`" in normal mode and "`never`" in VLBI-mode
[3] the default is "`skip`" for 20m and 25m, and "`save`" for CO/O_3 and H_2O

Sky dip (SKYDIP) observing scheme parameters

Number of loops	**loops**	int	*1 → 1000*	1	CO/O_3, H_2O
Integration time	**inttime**	int	*1 → 300*	30	CO/O_3, H_2O
Calibration time	**caltime**	int	*1 → 120*	10	CO/O_3, H_2O
Cal. elevation	**calelev**	real	*-90 → 185*	90	H_2O
Loops per cal	**calwhen**	word	*-1 1 2 3 4 5 6 7 8 9 10*	1	CO/O_3, H_2O

Extra cal. spectrum	**calextra**	word	*extra skip*	skip	H_2O
Extra cal. elevation	**extraelev**	real	*-90 → 185*	5	H_2O
Ref. data taken	**refwhen**	word	*first each*	each	CO/O_3, H_2O
Ref. elevation	**refelev**	real	*-90 → 185*	90	H_2O
Elevation list	**elevlist**	reals	*list of 1→16 values*[1]	[2]	CO/O_3, H_2O
Adjust power	**adjpwr**	word	*never first eachcal each2cal each3cal each4cal each5cal eachloop*	first	CO/O_3, H_2O
Raw data	**rawdata**	word	*skip save*	save	CO/O_3, H_2O

[1] a list of up to 16 values in the range 0 → 180 for the CO/O_3 system and in the range -90 → 185 for the H_2O system

[2] the default values are "`19.5 23.5 30 42 90 138 150 156.5 160.5`" for CO/O_3 and "`20 40 60 80`" for H_2O

Normal observation (NORMAL) observing scheme parameters

Number of loops	**loops**	int	*1 → 1000*	1	CO/O_3, H_2O
Pairs per loop	**pairs**	int	*1 → 100*	1	H_2O
Integration time	**inttime**	int	*1 → 1200/300*[1]	60/30[2]	CO/O_3, H_2O
Calibration time	**caltime**	int	*1 → 180/120*[3]	10	CO/O_3, H_2O
Cal. elevation	**calelev**	real	*-90 → 185*	90	H_2O
Loops per cal	**calwhen**	word	*-1 1 2 3 4 5 6 7 8 9 10*	1	CO/O_3, H_2O
Extra cal. spectrum	**calextra**	word	*extra skip*	skip	H_2O
Extra cal. elevation	**extraelev**	real	*-90 → 185*	5	H_2O
Signal elevation	**sigelev**	real	*-90 → 185*	20	H_2O
Ref. elevation	**refelev**	real	*-90 → 185*	90	H_2O
Adjust power	**adjpwr**	word	*never first eachcal each2cal each3cal each4cal each5cal eachloop*	first	CO/O_3, H_2O
Raw data	**rawdata**	word	*skip save*	save	CO/O_3, H_2O
Data averaging	**average**	word	*noadd show newfile addfile*	noadd	CO/O_3, H_2O

[1] the range is 1 → 1200 for the CO/O_3 system and 1 → 300 for the H_2O system

[2] the default is 60 for CO/O_3 and 30 for H_2O

[3] the range is 1 → 180 for CO/O_3 and 1 → 120 for H_2O

Triple observation (TRIPLE) observing scheme parameters

Number of loops	**loops**	int	*1 → 1000*	1	CO/O_3
Zenith int. time	**zenithtime**	int	*1 → 1200*	60	CO/O_3
Side int. time	**sidetime**	int	*1 → 1200*	60	CO/O_3
Calibration time	**caltime**	int	*1 → 180*	10	CO/O_3
Loops per cal	**calwhen**	word	*-1 -4 1 2 3 4 5 6 7 8 9 10*	1	CO/O_3

Side elevation	**sideelev**	real	*0 → 180*	25	CO/O_3
Tsys mode limit	**tsyslimit**	real	*100 → 10000*	800	CO/O_3
Adjust power	**adjpwr**	word	*never first eachcal each2cal each3cal each4cal each5cal eachloop each2loop each3loop each4loop each5loop*	first	CO/O_3
Raw data	**rawdata**	word	*skip save*	save	CO/O_3
Data averaging	**average**	word	*noadd show newfile addfile*	noadd	CO/O_3

Balanced power (BALANCED) observing scheme parameters

Number of loops	**loops**	int	*1 → 1000*	1	H_2O
Pairs per loop	**pairs**	int	*1 → 100*	1	H_2O
Integration time	**inttime**	int	*1 → 300*	30	H_2O
Calibration time	**caltime**	int	*1 → 120*	10	H_2O
Cal. elevation	**calelev**	real	*-90 → 185*	90	H_2O
Loops per cal	**calwhen**	word	*-1 1 2 3 4 5 6 7 8 9 10*	1	H_2O
Extra cal. spectrum	**calextra**	word	*extra skip*	skip	H_2O
Extra cal. elevation	**extraelev**	real	*-90 → 185*	5	H_2O
Signal elevation	**sigelev**	real	*-90 → 185*	20	H_2O
Ref. elevation	**refelev**	real	*60*[1] *→ 120*[1]	105	H_2O
Ref. adjustment	**refcorr**	real	*-2 → 2*	0.19	H_2O
Min ref. elevation	**refmin**	real	*60 → 100*	95[1]	H_2O
Max ref. elevation	**refmax**	real	*80 → 120*	115[1]	H_2O
Balance ref. elev.	**balref**	word	*0 1 2 3*	2	H_2O
Adjust power	**adjpwr**	word	*never first eachcal each2cal each3cal each4cal each5cal eachloop*	first	H_2O
Raw data	**rawdata**	word	*skip save*	save	H_2O
Data averaging	**average**	word	*noadd show newfile addfile*	noadd	H_2O

[1] the reference elevation limits have to fit with the values selected for Min ref. elevation and Max ref. elevation

Pointing measurement (POINT) observing scheme parameters

Number of loops	**loops**	int	*1 → 200*	1	*20m, 25m*
Spacing in azim	**azdist**	int	*10 → 400*	21	*20m*
Spacing in elev	**eldist**	int	*10 → 400*	21	*20m*
Spacing in polar	**poldist**	int	*10 → 1000*	220	*25m*
Spacing in dec	**decdist**	int	*10 → 1000*	220	*25m*
Source on-time	**ontime**	int	*1 → 120/300*[1]	20	*20m, 25m*

Source off-time	**offtime**	int	*1 → 120/300*[1]	20	*20m, 25m*
Calibration time	**caltime**	int	*1 → 60/120*[2]	10	*20m, 25m*
Loops per cal	**calwhen**	word	*-1 1 2 3 4 5 6 7 8 9 10*	3	*20m, 25m*
Observing mode	**obsmode**	word	*psw fsw*[3] *ssw*[3] *dsw*[3]	psw	*20m, 25m*
Update Doppler	**dop**	word	*never first eachcal eachloop*	eachcal[4]	*20m, 25m*
Adjust power	**adjpwr**	word	*never first eachcal each2cal each3cal each4cal each5cal*	first	*20m, 25m*
Raw data	**rawdata**	word	*skip save*	skip	*20m, 25m*
Add polarizations	**addpol**	word	*noadd show save*	save	*20m, 25m*
Backend to use	**backend**	word	[5] *OSA1 OSA2 OSAa OSA3 OSA4 OSAb SPE1 SPE2 SPEa HRF1 HRF2 HRFa*	[6]	*20m, 25m*
Make baselines	**baseline**	word	*makebaselines nobaselines*	nobaselines	*20m, 25m*
Start position	**startpos**	word	*current clearfirst cleareach last*	current	*20m, 25m*
Apply results	**apply**	word	*auto internally noapply*	auto	*20m, 25m*
Limit results using	**limit**	word	*absolute relative*	relative	*20m, 25m*

[1] the range is 1 → 120 for 20m and 1 → 300 for 25m
[2] the range is 1 → 60 for 20m and 1 → 120 for 25m
[3] available depending on receiver choice on 20m telescope
[4] the default is "`eachcal`" in normal mode and "`never`" in VLBI-mode
[5] available depending on backend choice
[6] the default is "`SPEa`" for 20m and "`HRFa`" for 25m

Focusing measurement (FOCUS) observing scheme parameters

Number of loops	**loops**	int	*1 → 200*	1	*20m*
Focus spacing	**dist**	real	*1 → 20*	3.5	*20m*
Source on-time	**ontime**	int	*1 → 120*	20	*20m*
Source off-time	**offtime**	int	*1 → 120*	20	*20m*
Calibration time	**caltime**	int	*1 → 60*	10	*20m*
Loops per cal	**calwhen**	word	*-1 1 2 3 4 5 6 7 8 9 10*	3	*20m*
Observing mode	**obsmode**	word	*psw fsw*[1] *ssw*[1] *dsw*[1]	psw	*20m*
Focusing axis	**axis**	word	*focus vshift hshift vtilt htilt*	focus	*20m*
Update Doppler	**dop**	word	*never first eachcal eachloop*	eachcal[2]	*20m*
Adjust power	**adjpwr**	word	*never first eachcal each2cal each3cal each4cal each5cal*	first	*20m*

Raw data	**rawdata**	word	*skip save*	skip	*20m*
Add polarizations	**addpol**	word	*noadd show save*	save	*20m*
Backend to use	**backend**	word	[3] *OSA1 OSA2 OSAa OSA3 OSA4 OSAb SPE1 SPE2 SPEa*	SPEa	*20m*
Make baselines	**baseline**	word	*makebaselines nobaselines*	nobaselines	*20m*
Start position	**startpos**	word	*current clearfirst cleareach*	current	*20m*
Apply results	**apply**	word	*auto internally noapply*	auto	*20m*

[1] available depending on receiver choice
[2] the default is "`eachcal`" in normal mode and "`never`" in VLBI-mode
[3] available depending on backend choice

Single observation (SINGLE) observing scheme parameters

Number of loops	**loops**	int	*1 → 1000*	1	*20m, 25m*
Source on-time	**ontime**	int	*1 → 1200*	60	*20m, 25m*
Source off-time	**offtime**	int	*1 → 1200*	60	*20m, 25m*
Calibration time	**caltime**	int	*1 → 180*	10	*20m, 25m*
Loops per cal	**calwhen**	word	*-1 1 2 3 4 5 6 7 8 9 10*	3	*20m, 25m*
Observing mode	**obsmode**	word	*psw fsw*[1] *ssw*[1] *dsw*[1]	psw	*20m, 25m*
Update Doppler	**dop**	word	*never first eachcal eachloop*	eachcal[2]	*20m, 25m*
Adjust power	**adjpwr**	word	*never first eachcal each2cal each3cal each4cal each5cal*	first	*20m, 25m*
Use the PLM	**useplm**	word	*use dontuse*	dontuse	*20m*
Raw data	**rawdata**	word	*skip save*	skip	*20m, 25m*
Add polarizations	**addpol**	word	*noadd show save*	save	*20m, 25m*
Data averaging	**average**	word	*noadd show newfile addfile*	noadd	*20m, 25m*

[1] available depending on receiver choice on 20m telescope
[2] the default is "`eachcal`" in normal mode and "`never`" in VLBI-mode

Map observation (MAP) observing scheme parameters

Map file name	**mapfile**	word	*filename*	-	*20m, 25m*
Loops per position	**loops**	int	*1 → 100*	1	*20m, 25m*
Pos per PSW-off	**offwhen**	int	*1 → 100*	1	*20m, 25m*
Break condition	**breaktype**	word	*time rounds pos loops never*	never	*20m, 25m*
Break after	**breakwhen**	int	*1 → 1000*	180	*20m, 25m*
Source on-time	**ontime**	int	*1 → 1200*	60	*20m, 25m*
Source off-time	**offtime**	int	*1 → 1200*	60	*20m, 25m*

Calibration time	**caltime**	int	*1 → 180*	10	*20m, 25m*
Loops per cal	**calwhen**	word	*-1 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20*	3	*20m, 25m*
Observing mode	**obsmode**	word	*psw fsw*[1] *ssw*[1] *dsw*[1]	psw	*20m, 25m*
Update Doppler	**dop**	word	*never first eachcal eachloop*	eachcal[2]	*20m, 25m*
Adjust power	**adjpwr**	word	*never first eachcal each2cal each3cal each4cal each5cal*	first	*20m, 25m*
Use the PLM	**useplm**	word	*use dontuse*	dontuse	*20m*
Raw data	**rawdata**	word	*skip save*	skip	*20m, 25m*
Add polarizations	**addpol**	word	*noadd save*	save	*20m, 25m*
Data averaging	**average**	word	*noadd save*	noadd	*20m, 25m*

[1] available depending on receiver choice on 20m telescope
[2] the default is "`eachcal`" in normal mode and "`never`" in VLBI-mode

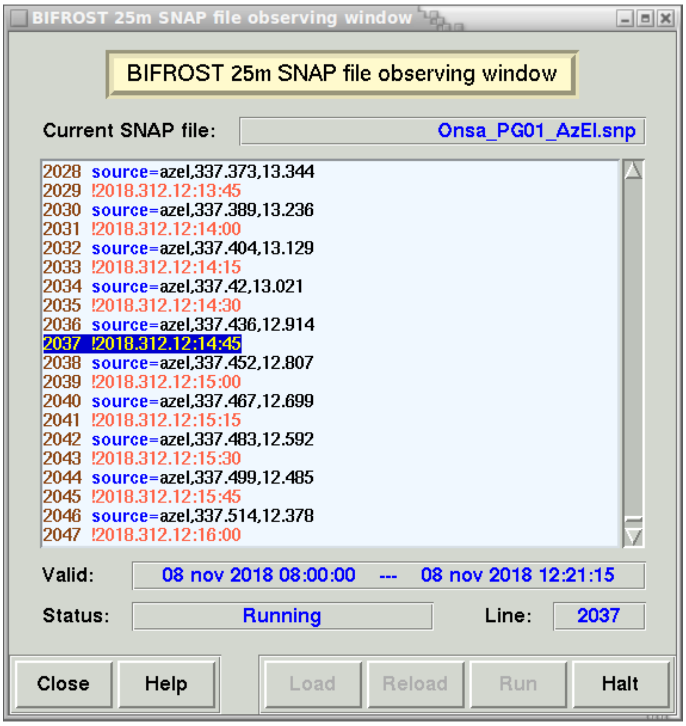
SNAP file observing

For the 20m and the 25m telescopes, there is a third observing method available provided that you start BIFROST in VLBI-mode. That is SNAP file observing and you get to the SNAP file observing window by clicking on the **SNAP file observing** button in the main menu. The SNAP file observing window allows you to run a pre-generated SNAP file.

SNAP files are the schedule files used by the VLBI Field Systems (FS) to control a VLBI observation. A SNAP file can include any valid command allowed by the FS, but only time tags and direct commands to control the telescope will be recognized by BIFROST.

Both absolute time tags and relative ones are recognized by BIFROST and you can mix them as you wish. However, the first time tag in a SNAP file has to be an absolute time tag, so that BIFROST knows when the SNAP file is supposed to be started.

As SNAP files are controlled by absolute time tags, they are only valid for a certain period of time. You can only run a SNAP file while it is valid or starting it a while before it becomes valid. The possibility to run a SNAP file disappears once it has expired.



A SNAP file with coordinates to track a satellite is running on the 25m telescope. The satellite position is updated with new horizontal coordinates every 15 seconds.

Running a SNAP file

Use the **Load** button to load a SNAP file into BIFROST. The SNAP file will be shown as a colour-coded listing in the big box. Line numbers are added automatically and must not be present in the SNAP file itself. Time tags are shown in orange and telescope commands are shown in blue and black, while comments are shown in green and all other commands in olive.

Once loaded you can use the left mouse button to select a start line (marked in yellow) provided that the SNAP file hasn't already expired or would run too far into the future. This is indicated by the Status box.

The **Run** button is used to start executing a SNAP file. All options to select start line, load new SNAP files and close the SNAP file observing window are disabled while the SNAP file is running. The current line being executed is shown in blue, when the SNAP file is running.

The **Halt** button halts the execution of a SNAP file. While halted, the starting line will be updated automatically as time progresses, so that you always start at the proper point if you want to restart the execution with the **Run** button.

Observing schemes

Observations are done using different observing schemes, which each perform a certain type of observation. The different BIFROST systems offer different observing schemes depending on the capabilities of the corresponding system (the OTTs don't offer any observing schemes, since they are only intended for VLBI). You have a handful of schemes to choose from starting with the simplest that just takes a spectra with the backend to the most complicated that can map a large number of positions on the sky. Each have their own set of parameters to control how the observation should be performed.

This chapter gives a brief overview of the available schemes and what they are used for. It is recommended that you use the on-line help to get more specific help on any parameter you wonder about.

Run backend

The run backend scheme is not intended for real observations, but is included for testing purposes. With this scheme you can take raw spectra with the backend(s). You can select whether you want to perform actions like moving the telescope or inserting calibration loads, or just fake them. There is no calibration or other post-processing of the data, so the only data product is the raw spectra.

Simple calibration

Calibrations can be made with the simple calibration scheme. It is a simple scheme that performs one or more calibrations and produces calibration spectra and Tsys-values.

You can choose whether you want to adjust the power levels before the calibration and if the raw data should be saved or not. For the 20m and 25m telescopes, you can also choose if you want to update the Doppler correction.

The calibration method used in simple calibration scheme is included in all more advanced observing schemes, so this scheme is not needed that often. One situation when it is useful is when you just want to make a calibration and then decide on what to do depending on the measured Tsys.

Sky dip

The sky dip scheme is only available for the CO/O_3 and H_2O radiometers. It produces a set of sky-switched spectra with the signal spectra taken at different elevations and the reference spectrum taken at a reference elevation. The elevations are specified in a list and you can give up to 16 values which will be observed in the order you have specified.

You can select whether a new reference spectrum should be taken for each signal elevation or if you want to use the same reference spectrum for all elevations. You can also choose whether you want to perform any calibration or adjust the power before the sky dip as well as whether raw data should be saved or not.

Normal observation

The normal observation scheme is only available for the CO/O_3 and H_2O radiometers. It is the basic scheme for obtaining scientific data and apart from having the same name, it works fairly differently on the two systems.

For the CO/O_3 system, a frequency-switched measurement is taken towards zenith on each loop. However, since the H_2O system is using sky-switching, which needs relatively short integration times, each loop consists of a number of sky-switched pairs taken at two different elevations. The data from the all the pairs within a loop is added together to produce one signal and one reference spectra per loop.

For both systems, you can choose when you want to perform calibrations or adjust the power as well as whether raw data should be saved or not, and whether you want to add data together from different loops to an average spectrum. For the H_2O system, you also have the possibility to take an extra calibration spectra at a different elevation as part of the calibration routine.

Triple observation

The triple observation scheme is only available for the CO/O_3 radiometer. It is a more advanced observing scheme which takes data with the CO/O_3 receiver in three different directions using frequency switching. The scheme makes four measurements per loop: one each towards the sides and two towards zenith. You can set different integration times to use at zenith and the sides. If the weather is too bad (as determined by the Tsys from the calibration), there is no point in doing measurements at the side elevations, and if that is the case then the loop is reduced to just a single zenith measurement. You can control when this is going to happen by setting a Tsys limit.

You can also choose when you want to perform calibrations (including before each integration in a loop) or adjust the power as well as whether raw data should be saved or not, and whether you want to add data together to three average spectra (one for each position).

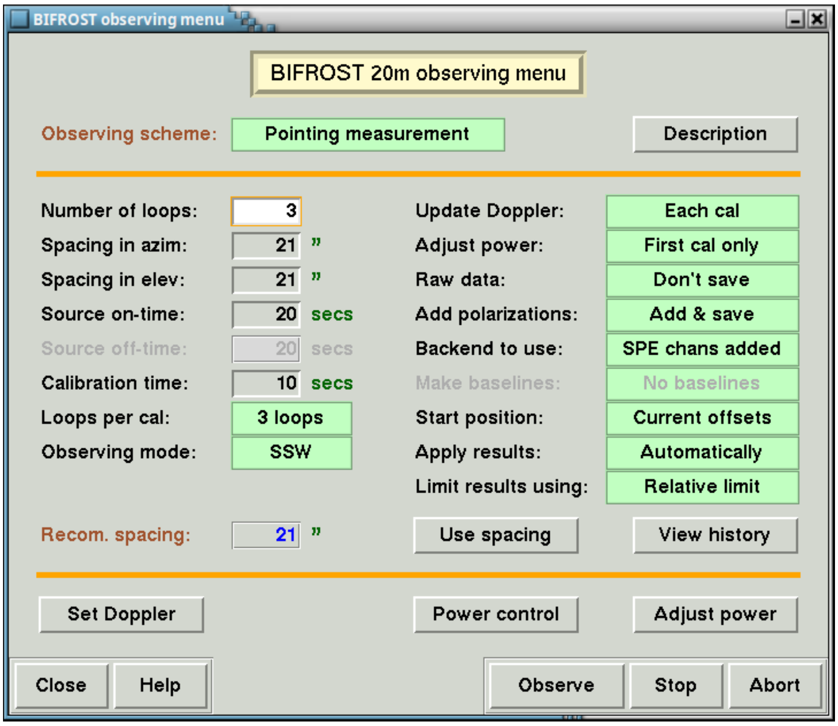
Balanced power

The balanced power scheme is only available for the H_2O radiometer. It is similar to the normal observation scheme with one important difference: instead of using a fixed reference elevation, it automatically adjusts the reference elevation (within a specified range) so that more or less of a small load is seen by the reference beam in order to balance the average power in the signal and reference beams. This is done by calculating the difference in total power between the two beams and then correct the reference elevation using a user-supplied adjustment factor. You can also select to make adjustments before starting an observation either automatically or manually by using the **Balance reference elevation** button. Quick measurements will then be performed and the reference elevation will be adjusted according to the result.

The remaining parameters are the same as for the normal observation scheme.

Pointing measurement

The pointing measurement scheme is only available for the 20m and 25m telescopes. It performs one or more five-point maps to determine the current offsets from the pointing model used. You have to specify the observing mode and the spacing to use in the two axes which depends on the frequency used, and the easiest thing to do is to use the recommended spacing by clicking on the **Use spacing** button.



The pointing measurement *scheme on the 20m telescope is shown here as an illustration of how to set up the parameters for an observing scheme in the* observing menu.

You can select when and if you want to apply Doppler corrections, perform calibrations or adjust the power levels. You also have to select if you want to save the raw data, if you want to add polarizations together and which backend channel you want to use for the data reduction.

There are some special parameters dealing with the pointing results and how it should be applied. There are several options for where to start the pointing from with the most advanced option being to search the pointing database and use the last pointing offsets found for the current source as the start point. Another parameter deals with how and if the results should be automatically applied or not, and finally there is a parameter controlling how to limit results to avoid the risk that too large offsets are applied.

Focusing measurement

The focusing measurement scheme is only available for the 20m telescope. It performs one or more three-point measurements on the specified focusing axis to determine the current offset from the focusing model used. The parameters are similar as those used for pointing measurement with a few exceptions.

You have to specify which of the five focusing axes you want to measure — in practice, it is only the **Z-focus** that can be easily measured and which should be checked periodically. As for the pointing measurement, you can use the **Use spacing** button to use the recommended spacing.

The other differences are that there is no option to get the start position from the focusing database, and there is no parameter to limit the results.

Single observation

The single observation scheme is only available for the 20m and 25m telescopes. It performs one or several measurements on the position of a single position on the sky, and is one of two available science schemes for the telescopes. You have to specify the observing mode to use as well as the integration time.

You can select when and if you want to apply Doppler corrections, perform calibrations or adjust the power levels. You also have to select if you want to save the raw data and if you want to add polarizations together.

There is also a parameter controlling whether you want to average the data, i.e. add together the data from individual measurements into *averaged spectra*. This can be done in several ways: you can have BIFROST create averaged spectra internally, which will be shown in the quick-look facility, but won't be saved to disk, or you can save them as separate FITS-files. In the latter case you have the choice if you want to create new average files or if you want to add the data to existing average files (provided that there are old average files for the current source and set-up).

If you are observing with the SIS 3/4-mm receiver on the 20m telescope, you will also have the choice if you want to use the *path-length modulator* (PLM) to try to cancel out the effects of standing waves.

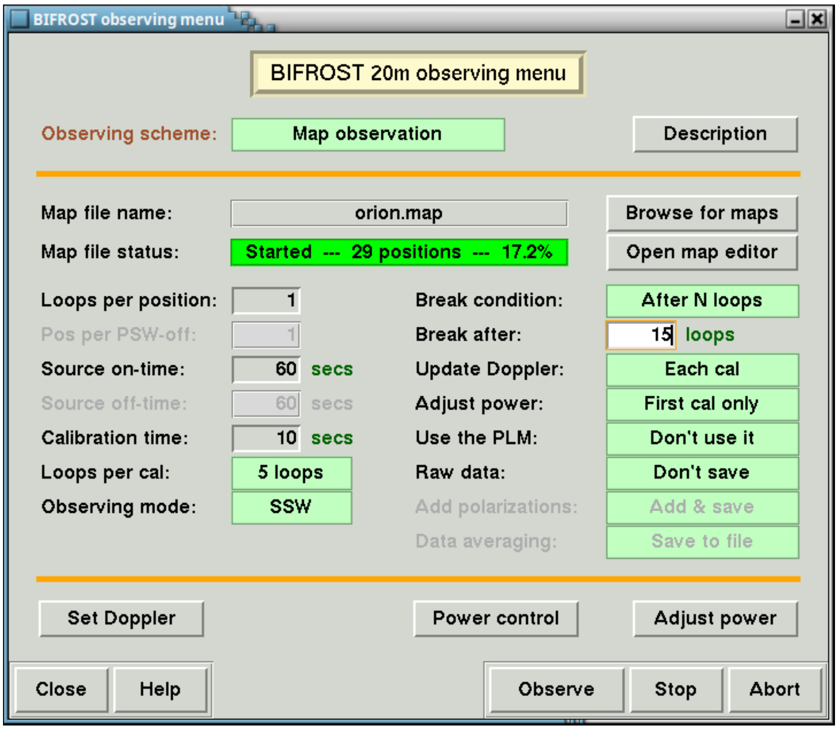
Map observation

The map observation scheme is a scheme where you observe a set of positions in an arbitrary-shaped raster map. It is only available for the 20m and 25m telescopes.

The map observation differs from all other observing schemes in that, apart from a set of parameters, it also needs a special "*map file*" which has to be prepared in the BIFROST map editor.

You have to specify the observing mode to use as well as the integration time and number of loops to perform in each map positions per round. If the observing mode is *position switching*, you can select to use the same off-spectra for several positions. For the 20m telescope, once you have started observing a map you can no longer switch between *frequency switching* and other observing modes.

You can select when and if you want to apply Doppler corrections, perform calibrations or adjust the power levels. You also have to select if you want to save the raw data, if you want to add polarizations together, and if you want to average the data. The last two parameters can not be modified once a map has been started.



The observation menu with the map observation *scheme selected on the 20m telescope. The lay-out of the map and further instructions on how it is to be observed is given in the specified map file. A couple of the parameters can no longer be selected since the map has already been started (5 of the 29 positions have already been observed).*

If you are observing with the SIS 3/4-mm receiver on the 20m telescope, you will also have the choice if you want to use the *path-length modulator* (PLM) to try to cancel out the effects of standing waves.

There are also two options specifying if and how to break the map observations. It is perfectly safe to interrupt a map observation interactively with the **Stop** button at any time — once you restart the observation, BIFROST will resume observing from the point where it was interrupted. However, you may want to use the break parameters to put in a planned break point, for example to allow for pointing and focusing.

A planned break can be set after a certain time interval or after a certain number of loops, positions or rounds. Loops and positions will be the same if you observe one loop per position. Positions and rounds will also be the same if you use another *mapping method* than **Loop over all**.

BIFROST may not obey the requested break parameters strictly: if you ask for 2 loops per position and to break after 9 loops, BIFROST will in fact make 10 loops before breaking to avoid leaving one position half-done.

The map file

Each map is associated with a "*map file*" which contains data about the geometry of the map and the map positions as well as instructions about how the map should be observed and which source, receiver, IF/LO and backend set-up that is used. It also contains internal bookkeeping information as well as status data and the names of all the FITS-files belonging to the map.

Map files exist in one of three states: *new*, *started* and *completed*. A *new* map has the lay-out of the map and the instructions on how it should be observed, but does not have any information about the source and system configuration. It can thus be used for whatever source.

Once you start observing a map, its status is changed to *started* and the information about the source and system configuration is added to the map file. From this moment you can only observe the map provided you have selected the same source and set up the system in the same configuration.

If the map isn't open-ended, i.e. if there is a mapping goal, then the map will be flagged as *completed* once the mapping goal has been reached for all map positions. Such a map will no longer be accepted for more observations. However, if you use the map editor to change the goal to allow more observing time in each position, the map status will be changed back to *started*, and you can continue observing the map.

As long as you haven't started to observe a map, you can make copies of the map file as you wish. However, once you have started, the map file becomes locked to a certain object and a certain set-up. If you want to make a copy of such a map file, you will have to use the **Duplicate** function in the map editor which creates a copy of a map, which is restored to the *new* state and thus can be used with any source or configuration.

Map files are ASCII-files, but you are **strongly advised** not to try to modify them by hand, since the results may be unpredictable.

Maps may contain anything from one up to 10 000 map positions. You will notice that manipulating a large map will be slower. One of the slowest operations is reading a large map file, since the desire is not just to read but perform a sanity check on all values. This takes too long for larger maps, and BIFROST will thus skip parts of this sanity check for larger maps. The **executive** is the part of BIFROST that tries the hardest and it may spend up to 15 seconds performing sanity checks when a map observation is started.

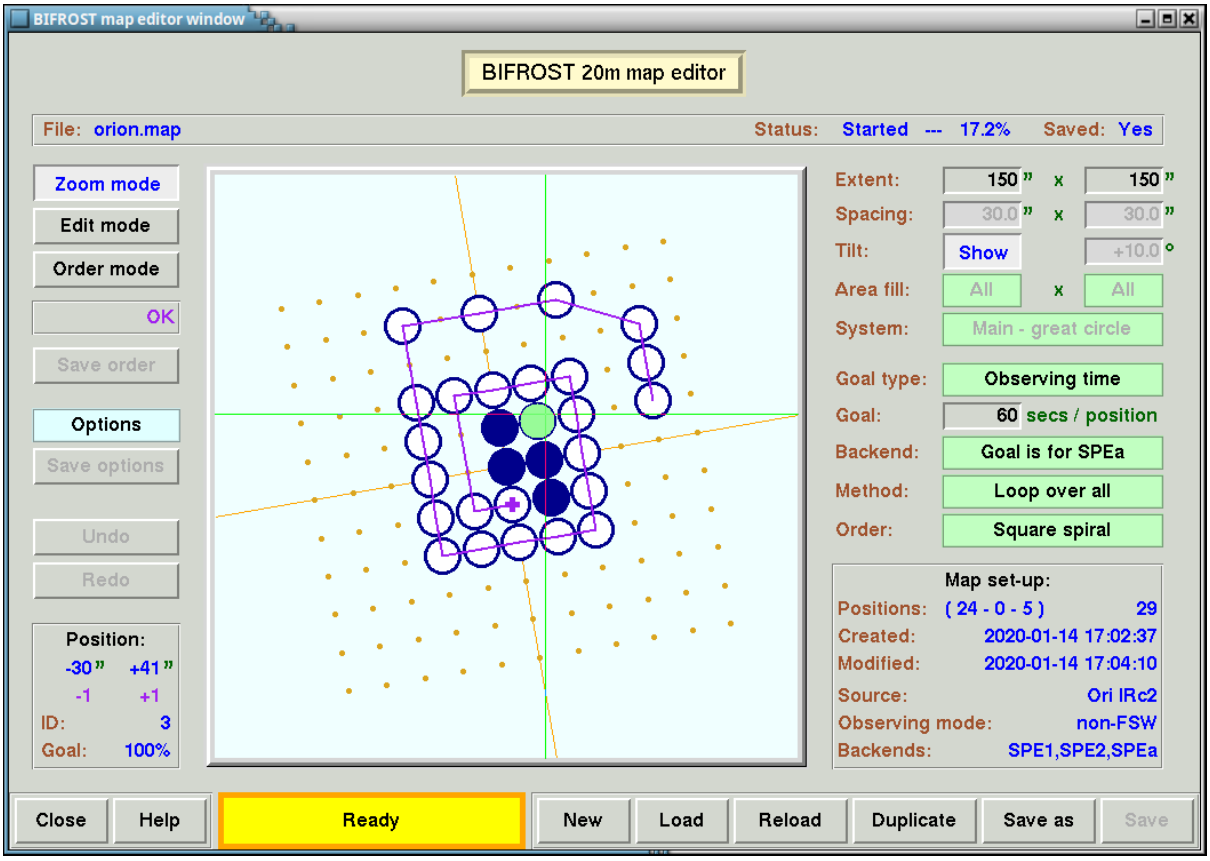
Maps with just one map position may sound silly, but they could actually be useful as an alternative to the single observation scheme, if you want to take advantage of the goal handling that comes with the map observation scheme.

The map editor

The BIFROST map editor is the tool used to create and edit "*map files*". It can be run both as part of a BIFROST observing session or in stand-alone mode (either from the BIFROST tools menu or via the command "`mapwin`" in a terminal). The status of the map will be updated automatically during a map observation, if the map editor has been opened from an observing session and is left open with the map being observed.

On the top of the window is a one-line box with the name of the map file, its status and a flag indicating whether the last changes have been saved or not.

At the bottom right, there are six buttons intended for loading and saving map files. Their usage should be fairly obvious, but two of the buttons are worth a couple of comments: the **Duplicate** button can be used to create a copy of a map file with all observations erased, so that it can be used for another source.



The map editor with a tilted map in which a few positions have already been observed (filled dark blue circles). The crosshair cursor hovers over map position 3 (highlighted with a light green disc) which has been completed, which we learn from the small box in the bottom left corner. The violet line indicates in which order the rest of the map will be observed.

The **Reload** button can be used to reload the current map file. This could be useful if you have made modifications that you want to discard or if you are modifying the map file as it is being observed. The map editor will normally be updated automatically during an observation, but if you make some modification without saving it, then BIFROST will be barred from making these updates.

The map editor left column

The central part of the map editor is the big graphic box where the map positions are shown. You modify the map by using the left or middle (or `<Shift>`-left) mouse buttons. What kind of actions the mouse buttons produce is decided by the selection of one of the three mode buttons in the top of the left column. In **Zoom mode**, you zoom in and out of the map, while in **Edit mode** you add and remove map positions. Note that you can add or remove an entire region by drawing a rectangle.

The **Order mode** is a special mode if you want to control in which order the map positions are written to the map file. In this mode you have to left-click once on all existing map positions in the order you want. When you are in **Order mode**, the small box below the **Order mode** which normally says `OK` will show you how many map positions you still have to mark. The **Save order** button can be used to save the currently selected order. **Note** that this refers to the order of the map positions in the map file — it does not control in which order the map positions will be observed *unless* you select As added or saved as the mapping order.

You don't need to be afraid of experimenting with the map — you can always go back with the **Undo** and **Redo** buttons.

The appearance of the big map box is controlled by a number of options that you can control via the **Options** menu. If you do change the options, the **Save options** button will be activated in case you want to save your new selection as a default setting.

The small status box at the bottom of the left column provides you with the current coordinates when you move the mouse inside the map box. If you move the mouse over a map position, you will also see its internal ID and its observing status.

The map editor right column

The upper part of the right column is used to set up the geometry of the map. The **Size** sets the extent from the origin of the map along the two axes. This doesn't matter for the map, but is for your convenience in setting up the proper scale of the graphic map box. The **Spacing**, however, is important, since this defines the grid of your map, and you are only allowed to put map positions on grid points. Thus, if you plan to observe an irregularly spaced map, you have to find the smallest common denominator and use that as the spacing.

The **Area fill** menus can be used if you plan to set up a sparse map. If you, for example, select 1 in 3 and then add (or remove) a rectangular block of map positions, then only every third position will be added (or removed). There are two menus so that you can set up the sparse sampling differently in the two axes.

You can tilt the map if you want by specifying a tilt angle in the **Tilt** entry field. The tilt is always defined as a counterclockwise rotation on the sky. **Note** that this also applies for mapping in equatorial coordinates or galactic coordinates (which both have the x-axis pointing towards left), and thus the tilt corresponds to the standard definition of the position angle. The **Show** button is a push-button you can use to select whether a tilted map should be shown tilted or not when you work with it in the map editor. The setting of this button does not affect your map file in any way.

The **System** menu is used to specify the coordinate system used for the map: the options are Az/El or Main which means the same coordinate system used for the source position. In this menu you also have to specify whether the longitudinal coordinate offsets should be small circle or great circle values — small circle values are used as they are and the distances are thus compressed closer to the pole, while great circle values are multiplied by $1/\mathit{cos}(\mathit{latitude})$ to maintain an equal angular distance.

Note that the x-axis goes in the opposite direction (from right to left) if the selected system is Main and the source is given either in equatorial coordinates or in galactic coordinates!

The **Spacing**, **Tilt** and **System** can only be set in a new map. Once you start observing the map, those entities will remain fixed.

The middle part of the right column deals with how the map should be observed. These options can be changed during an observation, but not all combinations make sense, so pay attention to what you are selecting.

The first three lines deal with the *mapping goal*. This is what determines when the observation of a map position has been finished. There are three goal types to choose from: Observing time, Comparative time and Keep going forever. The last one is for an open-ended map that will never be completed. For the other two, you need to specify a **Goal** and a **Backend** channel for which the goal should be applied. In fact, you need to specify the **Backend** channel also for open-ended maps, since this channel is the one used to report on the progress of the mapping.

The Observing time is the total physical time the backend has been taking data towards the source (including blanking time). This is the number of loops times the time you specified in **Source on-time** with two exceptions: if the observing mode is single beam switching, then the time is divided by two since only half the time has been spent on source, and if the backend channel selected is a channel with two polarizations coadded, then the time is multiplied by two.

The Comparative time is the amount of time it would have taken to reach the same result if the observed Tsys would have been the same as the *comparative system temperature* you have selected in the Calibration values menu (which defaults to 300 K, if you haven't changed it). By using the Comparative time you could thus ensure that all map positions reach a certain theoretical RMS, even if observing conditions vary a lot, and the different map positions thus may need different number of loops to reach this goal.

The **Method** menu is used to select which observing method to use of the three available methods. Finish each position will select the next map position according to the order chosen with **Order** menu, and will then keep observing this position until it reaches the mapping goal before moving on to the next one. This is the option to choose if you, in case the map can't be completed, prefer to get a part of your map observed to completion. Loop over all will also follow the order chosen with **Order** menu, but will loop over all positions. Such a loop is known as an observing round. If the mapping goals have not been fulfilled, more rounds will be observed. This is the option to choose if you, in case the map can't be completed or in the case of an open-ended map without a goal, want a map with as equal observing time in each position as possible. With Always improve worst, the mapping routine will compare all map positions with the mapping goal and select the worst position as the next position. Each observing round will only consist of one position, since all positions are reevaluated after each observation (and the setting of the **Order** menu will be more or less ignored). This is the option to choose if you, in case the map can't be completed or in the case of an open-ended map without a goal, want a map with as equal comparative observing time (or RMS) in each position as possible.

The **Order** menu is used to select in which order the map positions should be observed. You can select between random order, the order in which the map positions were added to the map, or a number of different geometric patterns. There are three situations when the order is not obeyed. The first is when there are map positions that have already reached the mapping goal. These map positions will not be observed again and will thus be ignored. The second situation is when the **Method** Always improve worst is used, since the mapping routine then always will look for the mapping position that has been observed the least. The third situation arise if you add map positions to a map that has already been fairly well observed. The mapping routine will then prioritise the new map positions to bring them up to par with the other positions.

The bottom part of the right column is a status box with information about the source, backend and observing mode selected as well as statistics on the map positions in the map. This box is mostly empty for a *new* map, since most of this information is added when the map has been *started*.

Some useful utilities

This chapter explains some of the useful utilities which you can find in the **General utilities** section of the BIFROST utilities window (which you open with the **More utilities** button in the BIFROST main menu).

The window used to add comments to the log file and the window used to set up BIFROST to forward all important messages to one or more email addresses. Comments added by the observer on the 20m and 25m telescopes will be included in the diary.

Adding a comment to the log (and the diary)

The Add comment to the BIFROST log is a simple facility to add comments to the BIFROST log. The **Message level to use** menu can be used to select what message level the comment should get in the log, which may be of interest if you are planning to search for a certain message level in the log files.

The message is added with the **Add to log** button, while the **Clear comment** button can be used to erase the comment box when you want to write a new message.

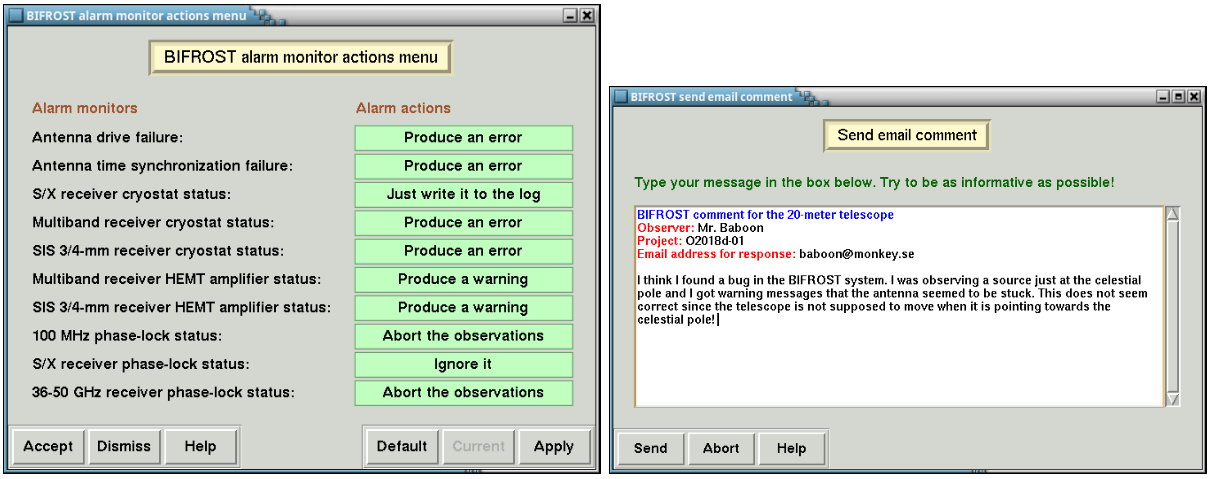
For the 20m and 25m telescopes, any comment added to the log will also be added to the *observing diaries*, if you choose to generate diaries.

Setting up email alerts

The BIFROST email alert set-up window can be used to set up email forwarding of all important messages from BIFROST to one or more email addresses. You select what type of messages you want to have forwarded with the **Send messages of type** menu and you can choose between "**Errors**", "**Errors and warnings**", "**Errors, warnings and problems**" and "**Errors, warnings, problems and alerts**".

The **Send test email to check recipient(s)** does just that: it sends a test email so that you can verify that you typed in the email addresses correctly and that you do get the emails sent from BIFROST. Note that this actually works also for off-line sessions provided that the off-line session is run on a computer connected to general network at Onsala Space Observatory.

You can cancel the email forwarding by setting **Send messages of type** to **None**.



The window used to configure the alarm monitoring actions (for the 20m and 25m telescopes) and the window to send a comment to the observatory staff.

Modify the alarm monitors

The BIFROST alarm monitor actions menu is only available on the 20m and 25m telescopes. It is used to specify what action BIFROST should take in case a hardware alarm condition occurs. Possible actions range from just ignoring the alarm, writing a message in the log, writing a warning or an error message, to aborting any ongoing observations. The possibility to choose message level (log, warning or error) can be useful, for example, if you decide to forward important messages via email (see previous page).

The fact that this set-up is "hidden" in the BIFROST utilities window and does not appear together with the other set-ups in the main menu, is because most observers don't need to change anything from the default set-up. The typical reason for changing some action is to disable some alarm when the hardware is acting up and falsely reports nonexistent errors.

You can save the selected actions in a configuration file (as an **Alarm** configuration), which is recommended if you have made any changes to the default actions.

This is one of the few BIFROST windows which has a **Defaults** button which you can click to reset the alarm monitor action to the recommended default actions.

Sending a comment about BIFROST via email

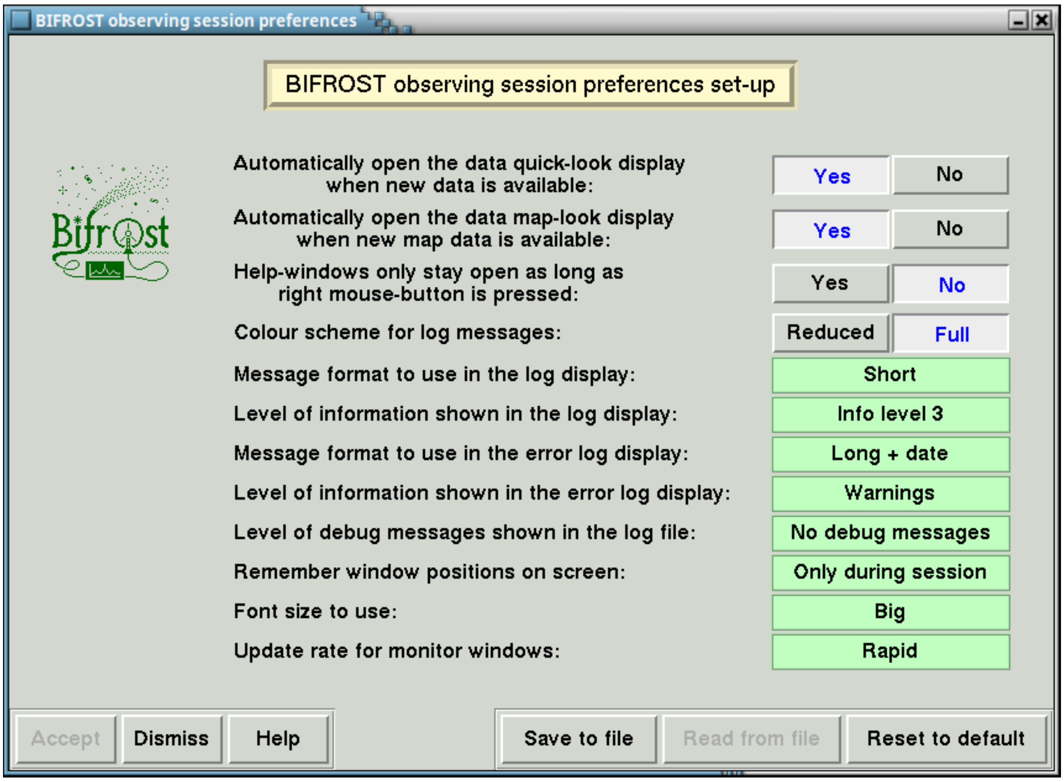
The Send email comment window is a remnant from Arecibo where it was used by observers to file bug reports and other comments about the observations when it was not possible to talk directly to the staff.

The "*Observer*" and "*Project*" fields are filled in automatically, but you can change them if you want. You are then supposed to fill in your email address for the response as well as your message before hitting the **Send** button.

Setting up preferences

There are a number of settings called preferences that you can set to customize BIFROST to behave in the way you like. They are set with the BIFROST observing session preferences set-up window which can be opened with the **Change preferences** in the BIFROST utilities window.

The preferences can be saved in a preference file which will be read each time BIFROST is started up. In fact, you are required to have a preference file, and BIFROST will actually force you to create one the first time you start up BIFROST for a new project. If you don't want to bother about them at that time, you can just save the suggested default values and then change the ones you don't like later on.



This is the default selection of preferences you will see when starting up the first observing session for a new project.

The available preferences

Here is a brief explanation of the currently available preferences:

- **Automatically open the data quick-look display when new data is available:** The default is that the quick-look display window will open automatically as soon as you get a new spectrum. With this preference you can turn off that behaviour.
- **Automatically open the data map-look display when new data is available:** The default is that the map-look display window will open automatically as soon as

you start observing a map (20m and 25m telescopes only). With this preference you can turn off that behaviour.

- **Help-windows only stay open as long as right mouse-button is pressed:** This preference select whether pop-up help messages (invoked by right-clicking on any feature in a BIFROST window) will stay open only as long as the mouse button is pressed or if they should stay open until you click a **Continue** button in the help window.
- **Colour scheme for log messages:** BIFROST uses different colours for all 27 different messages levels in the observation log display window. This preference can be used to sober it up, if you think there is too much colour used.
- **Message format to use in the log display:** This preference is used to select the format used for the messages shown in the observation log display window.
- **Level of information shown in the log display:** This preference is used to select which messages should be shown in the observation log display window. All messages with a level that is higher than the selected level will also be shown.
- **Message format to use in the error log display:** This preference is used to select the format used for the messages shown in the error log display window.
- **Level of information shown in the error log display:** This preference is used to select which messages should be shown in the error log display window. All messages with a level that is higher than the selected level will also be shown.
- **Level of debug messages shown in the log file:** This option should typically be left with its default "*No debug messages*" since the log files otherwise will become huge and the retroactive addition of debug messages in case of an error will provide you with the necessary debugging information.
- **Remember window positions on screen:** A few windows have default positions, but most windows don't so it is then up to the window manager to decide where they are going to appear on the screen. With this preference, you can tell BIFROST to remember where a particular window was last located on the screen and open it in the same position. This can be done just as long as the current observing session lasts or you can select to remember the positions permanently for future observing sessions as well.
- **Font size to use:** BIFROST windows can support four different font sizes which affect the size of the windows. With this preference you can adapt the font size depending on the size of your screen and how sharp your eyesight is.
- **Update rate for monitor windows:** This controls how often monitor windows like the observation status window should update. The default is "*Rapid*" and the idea is that if you are observing remotely and have a slow connection, you can slow things down.

BIFROST tools

There are several parts of the BIFROST observing system, e.g. the quick-look facility, that can be run in stand-alone mode outside of an BIFROST observing session. There are also some tools that only are available as stand-alone programs. The contents of the BIFROST tools menu depends on which BIFROST observing system is being used.

The BIFROST tools menu is started either by clicking on the -button or by typing "`bifrostwin`" in a terminal.

The BIFROST tools menu *as it appears for three different systems.*

All programs that can be started from the BIFROST tools menu can also be started from a terminal and if you right-click on each button, you will (among other help information) also get the command to use.

"Observing sessions" buttons in the tools menu

The **Observing sessions** part of the BIFROST tools menu contains two buttons: one that can be used to start a BIFROST observing session and one that can be used to kill a hanging or crashed observing session. The commands to use from a terminal are "`bifrost --X`" and "`clear_bifrost`".

"Observing tools" buttons in the tools menu

The **Command file editor** is used to create and edit command files. It is always run in stand-alone mode, even when it is launched from an observing session. It is described in the chapter "*The command file editor*" in the section "*Command file observing*". The command to use is "`cmdwin`" or "`cmdwin` ***`filename`***".

The **Map editor** (only 20m and 25m telescopes) is used to create or modify map files used when performing *map observations*. The map editor can be launched as part of an observing session but can also be run in stand-alone mode. It is described in the chapter "*The map editor*" in the "*Observing schemes*" section. In a terminal, it can be launched with the command "`mapwin`".

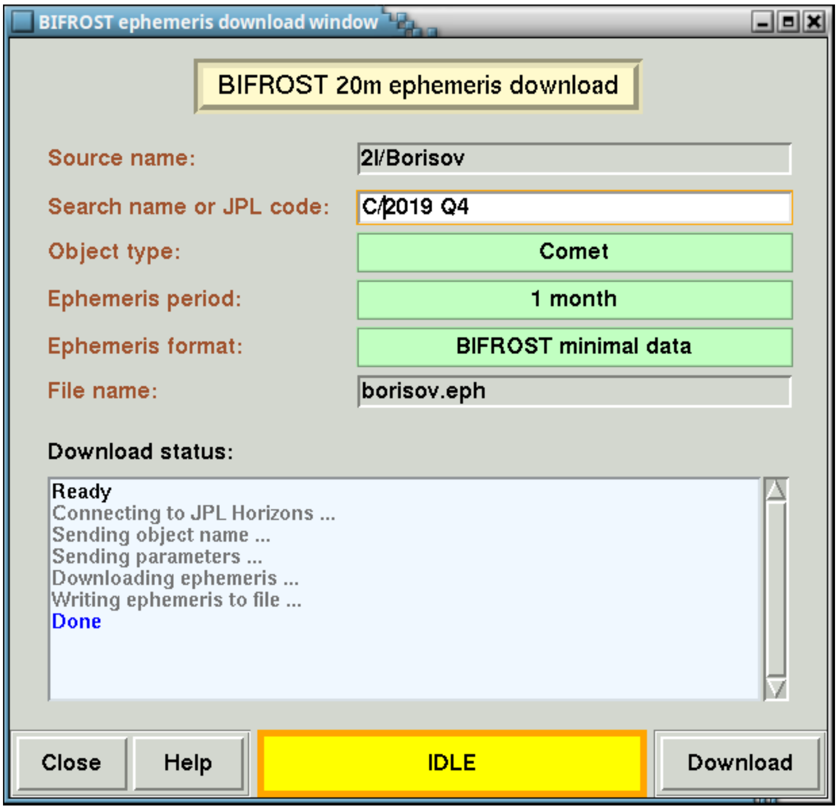
The **Graphic catalog display** (only telescopes) can be used to view where the sources in a catalog are located on the sky at present. If it is opened from the Source selection menu, you can even select a source from the graphic display by clicking on it. The tool is also useful for editing or checking the syntax of catalog files. It can be opened with the command "`catalogwin`".

The **Ephemeris download** window (only 20m and 25m telescopes) can be used to download an ephemeris for an object in the solar system from JPL Horizons. BIFROST will typically ask you if you want to download new ephemeris data whenever you are selecting a source catalog that contains ephemeris-based objects, but you can, of course, use this tool to download ephemeris whenever you want. The command to use in a terminal is "`ephemwin`".

The **Pointing history** (only 20m and 25m telescopes) is a window which gives you access to the database with data on all pointings. You can open this window from the observing menu when you have selected the pointing measurement scheme, or you can run it in stand-alone mode. It is started with "`pointwin`" from the command line.

The **Focusing history** (only 20m telescope) is a window which gives you access to the database with data on all focusings. You can open this window from the observing menu when you have selected the focusing measurement scheme, or you can run it in stand-alone mode. It is started with "`focuswin`" from the command line.

The **Diary writer** (only 20m and 25m telescopes) is a useful tool that can be used to create *observing diaries* which summarize what was done during an observing session. Diaries are described in the chapter "*Observation diaries*" in the section "*Documenting an observing session*". The diary writer is invoked with the command "`diarywin`".

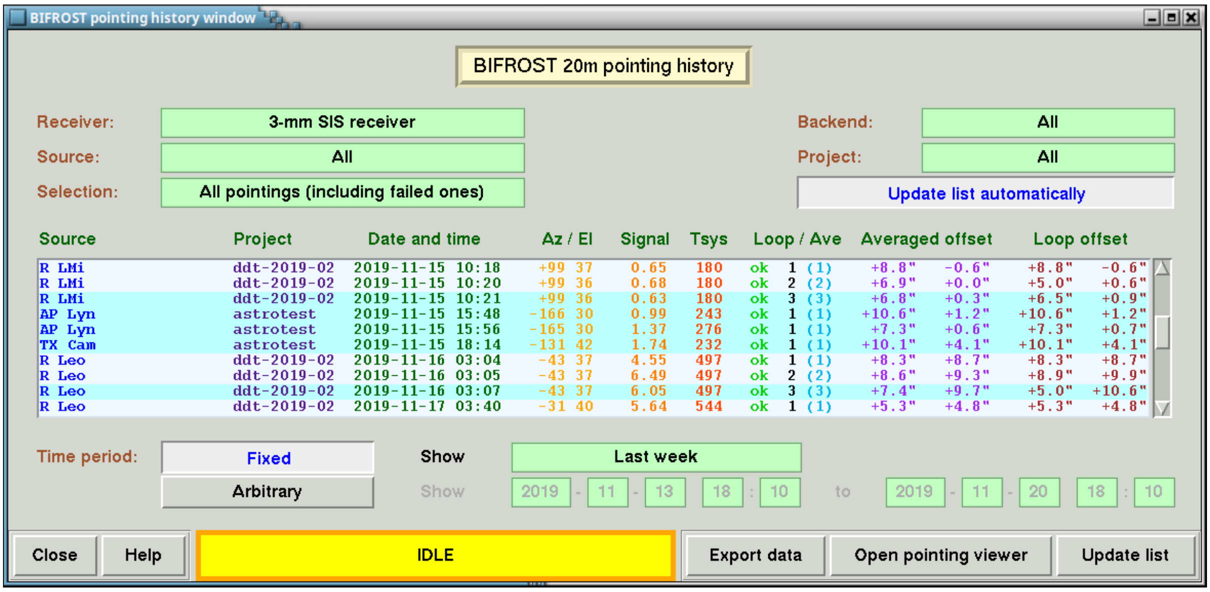


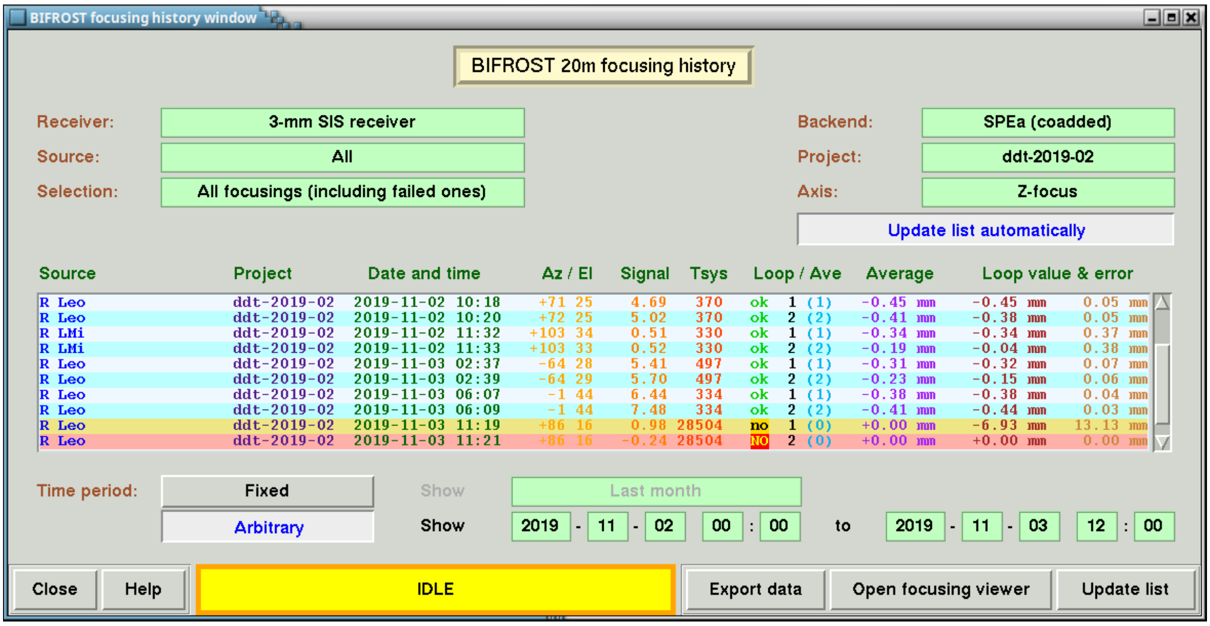
The BIFROST 20m ephemeris download *window has been used to download ephemeris data from JPL Horizons on the first interstellar comet. The* **Source name** *is the name to use in your catalog file. For this object, the JPL Horizons system does not recognize the name* "2I/Borisov", *so the alternative name* "C/2019 Q4" *has to be given.*

The **Spectrum analyzer** (only 20m telescope) is a tool that only is useful when the SIS 3/4-mm receiver is used and it is used to check the phase-lock of that receiver. It displays the output from a spectrum analyzer that sits in the control room, so for observers that are sitting at the console, they could just turn around and watch the instrument directly. However, for remote observers, this tool provides a way to access this instrument. There is usually no need to check it though. The command to use in a terminal is "`spanwin`".

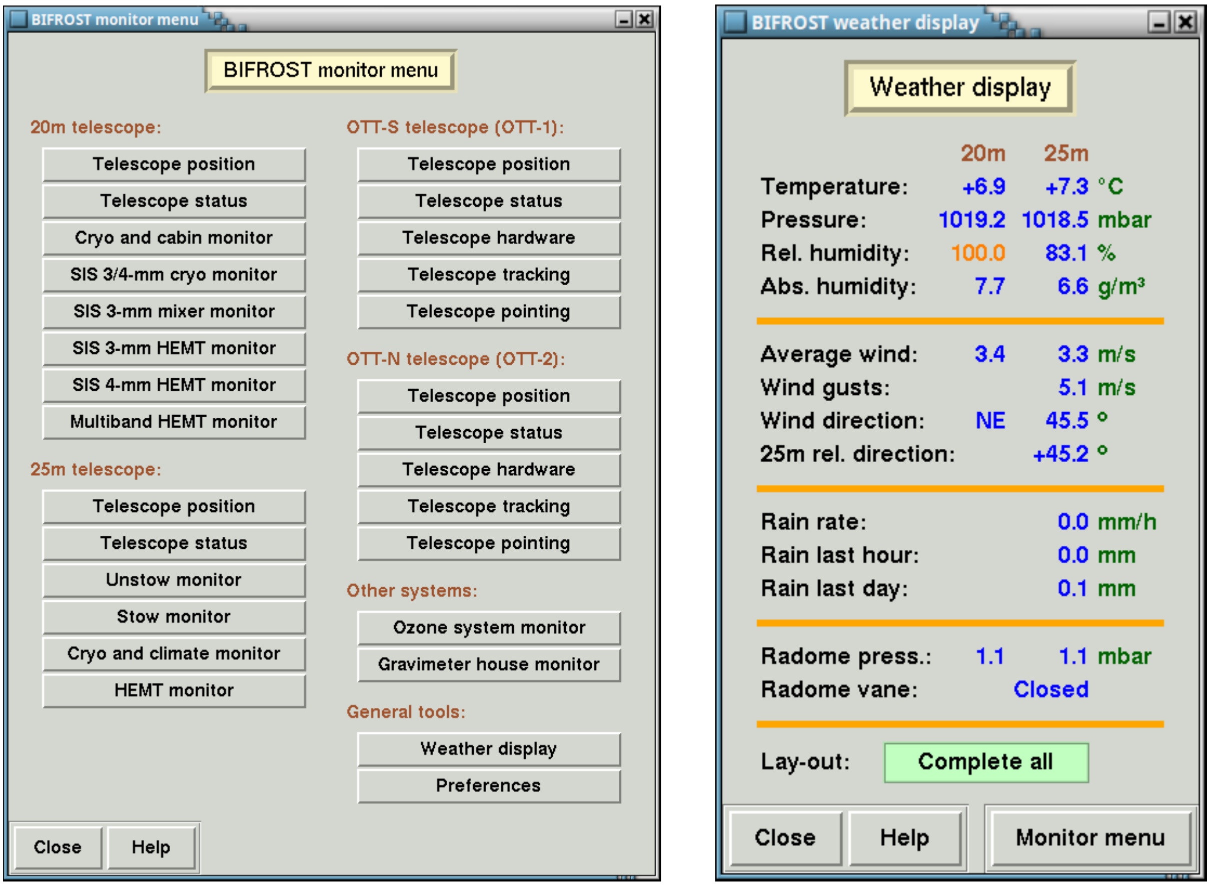
The **Quick-look data viewer** is usually run as part of the observing session, but you can also launch it in stand-alone mode to look at old spectra. It is described in the chapter named "*The quick-look facility*" in the section "*BIFROST FITS-files*" and can be launched with the "`quicklook`" command.

The **Map-look display** is also usually run as part of the observing session, but you can also launch it in stand-alone mode to look at the status of a map file or to follow the progress of an ongoing map observation. It is described in the chapter "*The map-look facility*" in the "*BIFROST FITS-files*" section. The command to use in a terminal is "`maplook`".





The BIFROST pointing history *and* BIFROST focusing history *windows show you data on all pointings/focusings that have been performed. You have a number of options you can use when selecting which pointings/focusings from the database you want to see. Clicking on any of the pointings/focusings in the list will bring up the* BIFROST pointing viewer *or* BIFROST focusing viewer *where you get all the details about that specific pointing/ focusing. The pointings/focusings entries are colour-coded depending on the result with white indicating a successful result as part of a loop and blue indicating the final result in the loop, while yellow indicates a result that is unrealistic and red indicates a result that didn't converge. The pointing/focusing list will be updated automatically whenever a new pointing/focusing has been performed, if any of these windows have been opened from the observing menu and left open.*



The BIFROST monitor menu *allows you to select among all the available BIFROST monitors, while the* Weather display *keeps you up-to-date about the weather situation.*

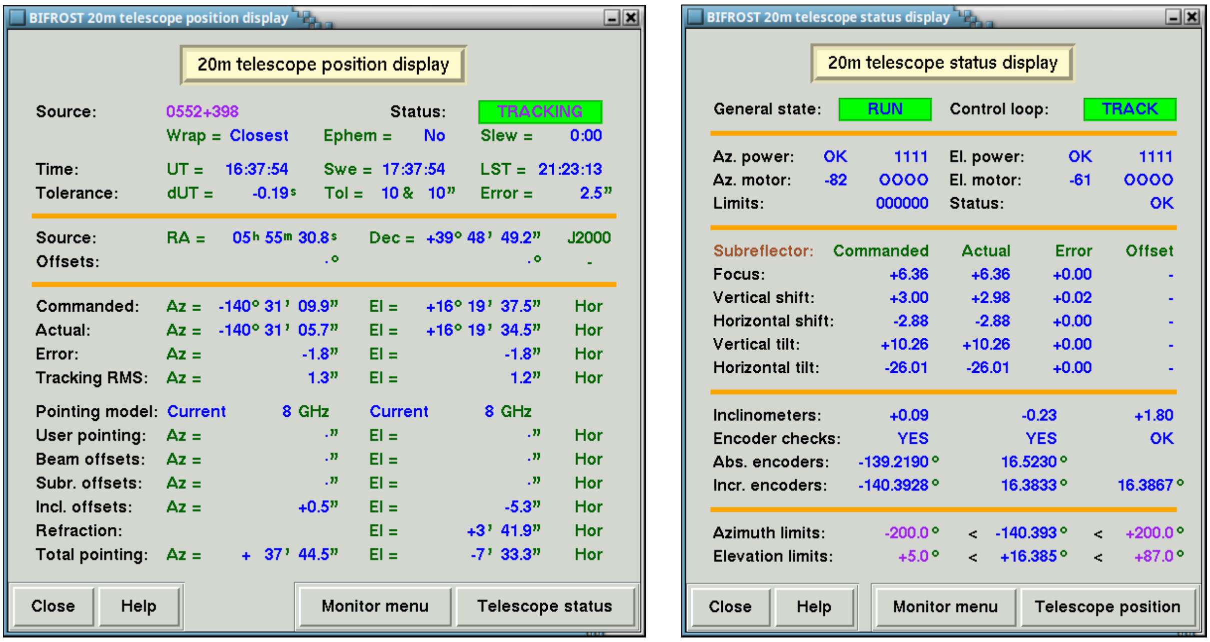
"Monitors" buttons in the tools menu

The **Monitor menu** is the general menu to select between all monitor displays available (for all systems). From here you can launch any of the other monitors. If you already have one monitor window open, you can bring up this general menu by clicking on the **Monitor menu** button located in the lower right corner of the monitor window. Type "`monwin`" to launch the monitor menu from a terminal.

The **Telescope position** display (only telescopes) can be used to monitor where the telescope is pointing and where it has been told to point. During an observing session, it can be opened with the **Telescope** button in the observation status window. It can also be started from a terminal with "`telposwin`" or "`telposwin` *`telescope`*".

The **Telescope status** display (only telescopes) can be used to monitor the status of the telescope. You can open this window by clicking the **Telescope status** button in either the BIFROST monitor menu or in the telescope position display. To launch it from a terminal, use "`telstatwin`" or "`telstatwin` *`telescope`*".

The **Weather monitor** can be used to bring up a weather monitor window. The size of this window can be modified since there is a menu button allowing the user to select what information should be displayed. It can be opened with the **Weather** button in



The telescope position display *and the* telescope status display *windows provide important information about what the telescope is doing. The appearance of these windows depend on the telescope.*

the observation status window during an observing session. The weather monitor can be launched with the command "`weatherwin`".

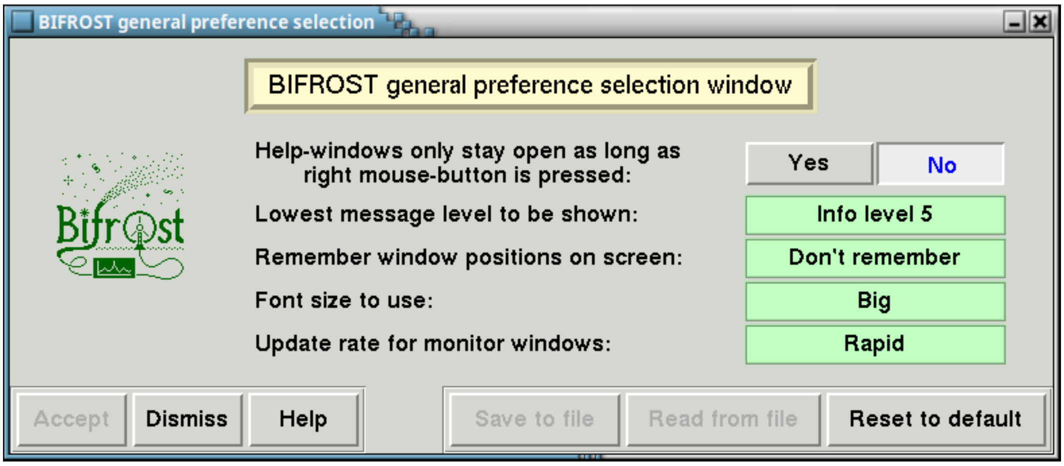
When opened from an observing session, any monitor window is actually run as part of the **feedback** program. In stand-alone mode, they are run by one single program. Thus, even if you launch several monitors independently from a terminal, you are in fact not running multiple monitor programs in parallel, because the different monitor windows will still be served by one single instance of the monitoring program. This program is launched when you open the first monitor window and does not quit until you close the last monitor window.

"Other tools" buttons in the tools menu

Of the buttons in the **Other tools** part of the BIFROST tools menu, it is only the **Preferences** button that is of interest for a normal observer. The other buttons are for staff usage. The **APC control** button can be used to power cycle certain equipment, and there may be some rare situation when this may be needed to reset some misbehaving piece of equipment. This will typically be done by staff, or otherwise the staff will give you instructions on how to use the APC control menu (the window is password-protected since we don't want anyone to power cycle something by mistake).

Preferences for stand-alone programs

The preferences for the BIFROST observing session has been described in the chapter "*Setting up preferences*". However, there is a separate set of preferences which is used by



The BIFROST general preference selection window *which contains the preferences used by all BIFROST programs run outside of an observing session.*

stand-alone programs. These preferences are basically a subset of the observing session preferences, but they are set up by a different program. You can launch it with the **Preferences** button in the BIFROST tools menu or by typing the command "`prefwin`" in a terminal.

It is important to note that you have to restart any stand-alone program after you have made a change to the preferences for the new preferences to take effect, since the stand-alone programs only read the preferences when they are launched. In contrast, the BIFROST observing session windows responds immediately to any change of their preferences.

Watching out for errors

The BIFROST system is picky about error conditions. It will abort whatever it is doing if a device does not respond in the expected way or if a command takes too long to finish. It may also flag the device as unconfigured, thus forcing you to reconfigure the device, just to ensure that the system really is in the desired state for your observations.

Catching the errors

You don't need to scroll through the observation log display if you want to quickly see just error or warning messages. There is a separate log window that just shows these most important message types called the error log display. It is opened with the **Open error log display** button either from the BIFROST utilities window (opened from the main menu with the **More utilities** button), or from the observation log filter selection window (opened by the **Filter** button in the observation log display window). You can select what message types it should be showing (among the most important ones) as well as the format used. The error log display can be left open permanently if you like.

You can also instruct BIFROST to send any error or warning messages to you via email. You get to the set-up window via the **Set up email alerts** button in the BIFROST utilities window (which is opened from the main menu with the **More utilities** button).

If you run command file observations, it is very useful to ask BIFROST to email you important information like when the command file has finished or if it crashes. You do this by including the "`ALERTME email-address(es)`" command in the command file to register one or more email addresses. You can even add your own messages in a command file with the command "`ALERT message`", e.g. "`ALERT completed observations with first frequency set-up`".

Restart of command files after errors

It is useful to write the command file in such a way that you can restart it in case it crashes due to an error (or just because you want to interrupt the command file for some reason). You can tell BIFROST that you want the command file to be automatically restarted by setting **Restart on error** to **Yes** in the command file observing window. You can get further control of how the restarts should be done by using the `ALLOWRESTART` command in your command files.

Note that BIFROST only tries to restart a command file if it is caused by an error that it seems possible to recover from. Thus if the command file tries to load a configuration file that doesn't exist, BIFROST will not bother to try a restart since chances that the missing file will appear magically are rather slim. However, if the crash was caused by a communication time-out with a piece of hardware or an error message from a backend, it is possible that the problem goes away if you wait a while or reconfigure the misbehaving device. In these cases it is worth trying to restart. It is useful to load a configuration file in the beginning of a command file, in case the problem is that some device needs a reconfiguration.

Monitoring hardware problems

For the 20m and 25m telescopes, BIFROST keeps monitoring some hardware that it doesn't really control such as the temperature, phase lock and status of amplifiers in the receivers. It also checks that the telescope is alive and doesn't get stuck. This monitoring is for the benefit of the observer, since you probably want to stop taking data if the receiver stops working properly. You can control whether and how you want to be informed about any such problem, and whether you want BIFROST to abort the observations or not. You do this from the BIFROST alarm monitor actions menu which is launched with the **Modify alarm monitors** button in the BIFROST utilities window.

The real alarm system for the hardware is running on another computer (the big screen on the right wall in the control room) and it will email responsible staff members. However, you should contact the AoD if there is a hardware problem, especially if it is related to the cryogenics of the receiver since it can take days to cool down a receiver again if it has warmed up.

Message logs in BIFROST

BIFROST offers the observer freedom to choose how much messages (s)he wants to see and in what format they should be displayed. The messages that are written to the log file and the messages that are shown in the observation log display are handled separately so you can modify your preferences for one of the streams without affecting the other.

All log messages are sent to the **logger** program which decides how to format them and which of them should be written to the log file and/or being passed on to the **feedback** program for display in the observation log display.

Message formats

The format in the log files is fixed and can not be modified. Each line contains a UTC date and time stamp, followed by the task code which consists of two 5-digit numbers within “<>”-brackets and separated by a colon. The task code is followed by the message level, the name of the procedure generating the message, a colon and finally the message itself. All fields are separated by a single space except the message level which is padded with extra spaces so that it always has the length of seven characters.

The task code contains the IDs of the current task and the current sub-task, which may be the same if the current task hasn't launched any sub-tasks. The ID is incremented each time a task or sub-task is started and it can be used to extract a certain task from the log file. Say that you want to check what happened during a certain calibration and you identify that this calibration had been given the task-ID 23056, you could extract only the lines in the log which contains the string “`<23056:`” and you would then get only the lines that were produced by this calibration. The task counter starts at 10001 and wraps back to this value when it hits 99999.

A typical line looks like this:

```
2018-Oct-04 09:03:42.47 <10142:10142> INFO1   config_spe: Configuring the FFT spectrometer
```

The log files are written both to the project area and to a system-wide archive for the 20m and 25m telescope. However, for the ozone systems and for the OTTs, the writing of log files to the project area have been disabled, and the only log written is the log to the archive area (which is located at */archive/Obslog/*`telescope`). There is a difference in format between the log written to the two areas, since the archive logs do have the project code inserted between the time stamp and the task code.

The format of the messages shown in the observation log display and in the error log display can be selected and changed whenever you want. There are eight different formats to choose, from the simplest one which just gives you the time and the message itself to the most complete which also adds the date, the message level, the name of the procedure generating the message and the task-IDs described above.

Message levels

Each message is assigned a *message level* which determines its importance. There are no less than 27 different levels with "ERROR" being the most important one and "DEBUG5" being the least important one.

When you select a message level, you tell BIFROST that you want to see messages of this level and all messages of higher importance, and want to ignore all messages at lower levels of importance.

These are the 27 message levels arranged in falling order of importance:

ERROR — WARNING — PROBLEM
ALERT — NOTE
COMMAND — BEGIN — END — DONE
TEST
INFO1 — INFO2 — INFO3 — INFO4 — INFO5
START — STOP
LOG1 — LOG2 — LOG3 — LOG4 — LOG5
DEBUG1 — DEBUG2 — DEBUG3 — DEBUG4 — DEBUG5

Note especially the way the numbered levels work: "INFO1" is more important than "INFO2" which is much more important than "INFO5".

For the observation log display you can select between the INFO-, the LOG- and the DEBUG-levels. The recommendation is to use one of the INFO-levels, unless you encounter some problem and want to see much more information.

For the log-files, everything down to LOG5 is always saved. Your only option is whether you also want to add some DEBUG-levels, which you are strongly advised against unless you are a staff member trouble-shooting something. Observations for a day typically generate 5–20 MB of log messages at the LOG5-level. This will quickly explode if you start adding DEBUG-levels. At DEBUG3, BIFROST is forced to report each and every byte it is sending back and forth to the hardware and at DEBUG5, each and every internal procedure has to report each time it is called. That means that at DEBUG5 you will get a constant flow of messages even if you are not doing anything, since you will start to see messages from all internal alarm and monitoring procedures.

Retroactive debug messages

The need to save DEBUG-messages is limited — especially since BIFROST will automatically add the missing DEBUG-messages to the log file in case it encounters an error while performing a task. This is done retroactively for the current task so that you will get the necessary details about what happened before the error occurred.

This is achieved by the BIFROST logger which receives all messages including all debug messages. The logger normally filters out the messages that should go to the log file and the ones that should go to the log display window. However, all messages from the current task are retained in memory and if the logger receives a signal that the task terminated with an error, then it cuts away all messages from the current task that have already been written to the log file and rewrites the log file from the messages stored in memory including the appropriate debug messages.

Not all DEBUG-messages will be written to the log file, since the signal that indicates that a task has crashed also contains an instruction telling which DEBUG-levels should be included in the log. A crash due to a simple user-error (like providing an incorrect name for a configuration file) would only add DEBUG1 messages, while a communication problem with a spectrometer would add everything down to level DEBUG3 or DEBUG4, so that the attempted communication can be analyzed in detail.

Changing message format and level retroactively

Via the **Filter**-button in the observation log display, you get to the observation log filter selection window where you can change the message format as well as the message level. You can then decide from when the changes should be applied. If you saw something strange in the log a couple of minutes ago, you could change the message level and select to apply the change for the last three minutes. The **logger** program will read the log file and replace the log shown with the messages that should be shown according to the new message level.

This will only work down to message level LOG5, so adding DEBUG-messages retroactively won't work, unless you selected to write DEBUG-messages as well to the log file, which you are advised not to do. However, in case of an error, BIFROST will add appropriate DEBUG-messages to the log file, so after a problem, you will be able to see DEBUG-messages related to the problem, if you choose to show DEBUG-levels.

BIFROST FITS-files

The data produced by BIFROST is written to FITS-files, which comply with the FITS-standard. A simple format is used where each spectrum is written to a separate file. Each file consists of two parts: an ASCII-header with keywords containing metadata about the observation and a binary block with the data itself. Both blocks are padded so that their size corresponds to an integer number of 2880-byte FITS-blocks. The header is padded with space characters while the data is padded with binary zeros.

FITS-file categories

There are basically three categories of FITS-files: raw spectra, data products and averaged data.

The raw spectra are exactly that: the raw data from the backend which maybe have been rescaled and had some channels at the edges clipped away. This data can be used if you want to redo the data reduction from scratch including calculating calibrations. In many cases you are given the choice if you want to save raw data or if you are just happy with the standard data products. Not saving raw data will save you disk space.

The data products are processed spectra where signal and reference spectra have been processed to produce a data product: a calibration or a data spectrum with a calibration applied. All data products are calibrated with the most recent calibration. Typically, each observing loop produces one set of data products.

Averaged data is data products that have been added together (using a weighting based on the system temperature). Although they should not be used for publication, they will show you approximately what the total spectra will look like and are very useful for checking weak sources and measuring the RMS achieved. BIFROST keeps track of sources and set-ups, which means that you can switch back and forth between them and still get the data added to the proper average file. For observing schemes that allow data averaging, there is typically an observing scheme parameter where you can specify whether average data should go to a new file or continue being added to an existing file, provided that there is an existing file for the given object and set-up.

FITS-file names

Each FITS-file gets a unique number which forms the name of the FITS-file together with a code and the "`.fits`" suffix. The first FITS-file generated by BIFROST on a system gets the number 10001 and this number is then incremented for each file created.

The code is a combination of letters and digits which identifies what type of data the FITS-file contains. It typically consists of one letter, two letters or a letter and a digit, although there also exists combinations of two letters and a digit as well as one letter and two digits. Uppercase and lowercase letters represent different types of files, for example the code "`S`" is given to the signal spectrum from a calibration while a signal spectrum from a non-calibration is given the code "`s`".

The most typical codes are "c" for a calibration spectrum with the raw signal and reference spectra are given the codes "S" and "R", respectively. A simple data spectrum is given the code "d" with the raw signal and reference spectra given the codes "s" and "r", respectively. If there are two polarizations that are coadded together, then the FITS-file is given the code "a". The letter "A" is added to the code, if the FITS-file contains several individual spectra that have been averaged together, e.g. "dA" and "aA".

There are quite a number of different codes used for more complicated observing schemes. For example, "135042D4.fits" is the calibrated data spectrum for the fourth (4) elevation in a sky dip (D), while "2421356rd2.fits" is the raw reference (r) spectrum for the second (2) dual-beam switch pair in the "*down*" (d) position of a pointing observation. A complete list of all codes are given below in the chapter "*FITS-files generated by different observing schemes*".

Even if you don't want to save raw spectra, you may still choose to see them in the quick-look facility. Such spectra are written temporarily to the */tmp*-disk and they receive four-digit numbers starting from 1001. The counter for temporary files is reset after each observation.

FITS-file administration

The FITS-files are put in the *Data* directory in your project area. There will be a directory created for each UTC-day and in that directory there will be a set of different subdirectories (e.g. *Calibrations*, *Data*, *Coadded* and *Raw*) where the FITS-files are placed depending on what they contain. The set of subdirectories vary depending on which BIFROST system is used.

There is one special category of FITS-files that end up in a special directory: if you average together data from several observations, those data are written to FITS-files in the subdirectory: *Data/Averages*. There is no UTC-date in the directory path since data from different days may be averaged together.

Data about each FITS-file is also written to two log files in the *Logs* directory in the project area. Meta-data from each FITS-file is written in a file called "*FITS_scan.log*", while data about which FITS-files belongs together in a *set* is written to "*FITS_set.log*". A FITS-set can typically be the two raw data spectra (signal and reference) as well as the calibrated data spectra for one backend channel (i.e. one polarization).

The "*FITS_scan.log*" is currently not used by BIFROST, but could be of interest to an observer who wants to make a database of the FITS-files. The "*FITS_set.log*" is used by the quick-look facility to quickly determine which FITS-files belong to each other.

FITS-files generated by different observing schemes

This chapter will list all different FITS-file types that are generated by the different observing schemes. For each file type, there will be four columns: the first one contains the FITS-file code used, the second one tells in which *Data* subdirectory this file type will be written, the third one describes what type of spectrum and the fourth column lists any conditions.

Run backend FITS-files

The following file types may be generated by the single observation scheme for all systems:

"**s**"	Raw	raw signal	*depending on choice*
"**r**"	Raw	raw reference	*depending on choice*
"**S**"	Raw	raw calibration signal	*depending on choice*
"**R**"	Raw	raw calibration reference	*depending on choice*
"**sA**"	Raw	averaged raw signal	*depending on choice*
"**rA**"	Raw	averaged raw reference	*depending on choice*
"**SA**"	Raw	averaged raw calibration signal	*depending on choice*
"**RA**"	Raw	averaged raw calibration reference	*depending on choice*

with the following file types only appearing in the CO/O_3 system, which uses frequency switching:

"**s**x"	Raw	raw signal FSW-x	*depending on choice*
"**r**x"	Raw	raw reference FSW-x	*depending on choice*
"**S**x"	Raw	raw calibration signal FSW-x	*depending on choice*
"**R**x"	Raw	raw calibration reference FSW-x	*depending on choice*

where "x" is "**1**" for frequency-1 and "**2**" for frequency-2.

Simple calibration FITS-files

The following file types are generated by the simple calibration scheme:

"**S**"	Raw	raw calibration signal	*optional*
"**R**"	Raw	raw calibration reference	*optional*
"**c**"	Calibrations	calibration	
"**t**"	Calibrations	Tsys	*only CO/O_3 and H_2O*

with the following file types only appearing in the CO/O_3 system, which uses frequency switching:

"**S1**"	Raw	raw calibration signal FSW-1	*optional*
"**S2**"	Raw	raw calibration signal FSW-2	*optional*
"**R1**"	Raw	raw calibration reference FSW-1	*optional*
"**R2**"	Raw	raw calibration reference FSW-2	*optional*

and the following file types only appearing in the H_2O system which uses both different elevations as well as a noise diode for the calibrations:

"**SN**"	Raw	raw signal with noise diode	
"**RN**"	Raw	raw reference with noise diode	
"**X**"	Raw	raw extra at other elevation	*optional*

Sky dip FITS-files

The following file types are generated by the sky dip scheme:

"**s**x"	Raw	raw signal at elevation x	*optional*

"**r1**"	Raw	common raw reference	*optional (if selected)*
"**r***x*"	Raw	raw reference at elevation *x*	*optional (if selected)*
"**d***x*"	Skydips	calibrated data at elevation *x*	

where "*x*" is a hexadecimal letter from **0** – **f**. The "**r1**" is used if all elevations use the same reference. The following additional file types appear only in the CO/O_3 system, which uses frequency switching:

"**s***x***1**"	Raw	raw signal at elevation *x* FSW-1	*optional*
"**s***x***2**"	Raw	raw signal at elevation *x* FSW-2	*optional*
"**r11**"	Raw	common raw reference FSW-1	*optional (if selected)*
"**r12**"	Raw	common raw reference FSW-2	*optional (if selected)*
"**r***x***1**"	Raw	raw reference at elevation *x* FSW-1	*optional (if selected)*
"**r***x***2**"	Raw	raw reference at elevation *x* FSW-2	*optional (if selected)*

Normal observation FITS-files

The following file types are generated by the normal observation scheme:

"**s**"	Raw	raw signal	*optional*
"**r**"	Raw	raw reference	*optional*
"**d**"	Data	calibrated data	
"**dA**"	Averages	averaged calibrated data	*optional*

Triple observation FITS-files

The following file types are generated by the triple observation scheme:

"**s**"	Raw	raw signal	*optional*
"**r**"	Raw	raw reference	*optional*
"**z**"	Data	calibrated data in zenith	
"**e**"	Data	calibrated data in east	*if weather OK*
"**w**"	Data	calibrated data in west	*if weather OK*
"**zA**"	Averages	averaged calibrated data in zenith	*optional*
"**eA**"	Averages	averaged calibrated data in east	*optional (if weather OK)*
"**wA**"	Averages	averaged calibrated data in west	*optional (if weather OK)*

Balanced power FITS-files

The following file types are generated by the balanced power scheme:

"**s**"	Raw	raw signal	*optional*
"**r**"	Raw	raw reference	*optional*
"**d**"	Data	calibrated data	
"**dA**"	Averages	averaged calibrated data	*optional*

Pointing measurement FITS-files

The following file types are generated by the pointing measurement scheme:

"**s***x*"	Raw	raw signal position *x*	*optional (non-DSW)*
"**r**"	Raw	common raw reference	*optional (PSW only)*
"**r***x*"	Raw	raw reference position *x*	*optional (FSW and SSW)*
"**s***x***1**"	Raw	raw signal 1 position *x*	*optional (DSW only)*
"**s***x***2**"	Raw	raw signal 2 position *x*	*optional (DSW only)*
"**r***x***1**"	Raw	raw reference 1 position *x*	*optional (DSW only)*
"**r***x***2**"	Raw	raw reference 2 position *x*	*optional (DSW only)*
"**p***x*"	Pointings	data position *x*	
"**a***x*"	Pointings	coadded data position *x*	*if dual polarizations*

where "*x*" is one of the five positions: "**l**" for **left**, "**r**" for **right**, "**c**" for **center**, "**d**" for **down** and "**u**" for **up**.

Focusing measurement FITS-files

The following file types are generated by the focusing measurement scheme:

"**s***x*"	Raw	raw signal position *x*	*optional (non-DSW)*
"**r**"	Raw	common raw reference	*optional (PSW only)*
"**r***x*"	Raw	raw reference position *x*	*optional (FSW and SSW)*
"**s***x***1**"	Raw	raw signal 1 position *x*	*optional (DSW only)*
"**s***x***2**"	Raw	raw signal 2 position *x*	*optional (DSW only)*
"**r***x***1**"	Raw	raw reference 1 position *x*	*optional (DSW only)*
"**r***x***2**"	Raw	raw reference 2 position *x*	*optional (DSW only)*
"**f***x*"	Focusings	data position *x*	
"**a***x*"	Focusings	coadded data position *x*	*if dual polarizations*

where "*x*" is one of the three positions: "**i**" for **inner**, "**m**" for **middle** and "**o**" for **outer**.

Single observation FITS-files

The following file types are generated by the single observation scheme:

"**s**"	Raw	raw signal	*optional (non-DSW)*
"**r**"	Raw	raw reference	*optional (non-DSW)*
"**s1**"	Raw	raw signal 1	*optional (DSW only)*
"**s2**"	Raw	raw signal 2	*optional (DSW only)*
"**r1**"	Raw	raw reference 1	*optional (DSW only)*
"**r2**"	Raw	raw reference 2	*optional (DSW only)*
"**d**"	Data	calibrated data	
"**a**"	Coadded	codded calibrated data	*if dual polarizations*
"**dA**"	Averages	averaged calibrated data	*optional*
"**aA**"	Averages	averaged coadded calibrated data	*optional (if dual pol)*

Map observation FITS-files

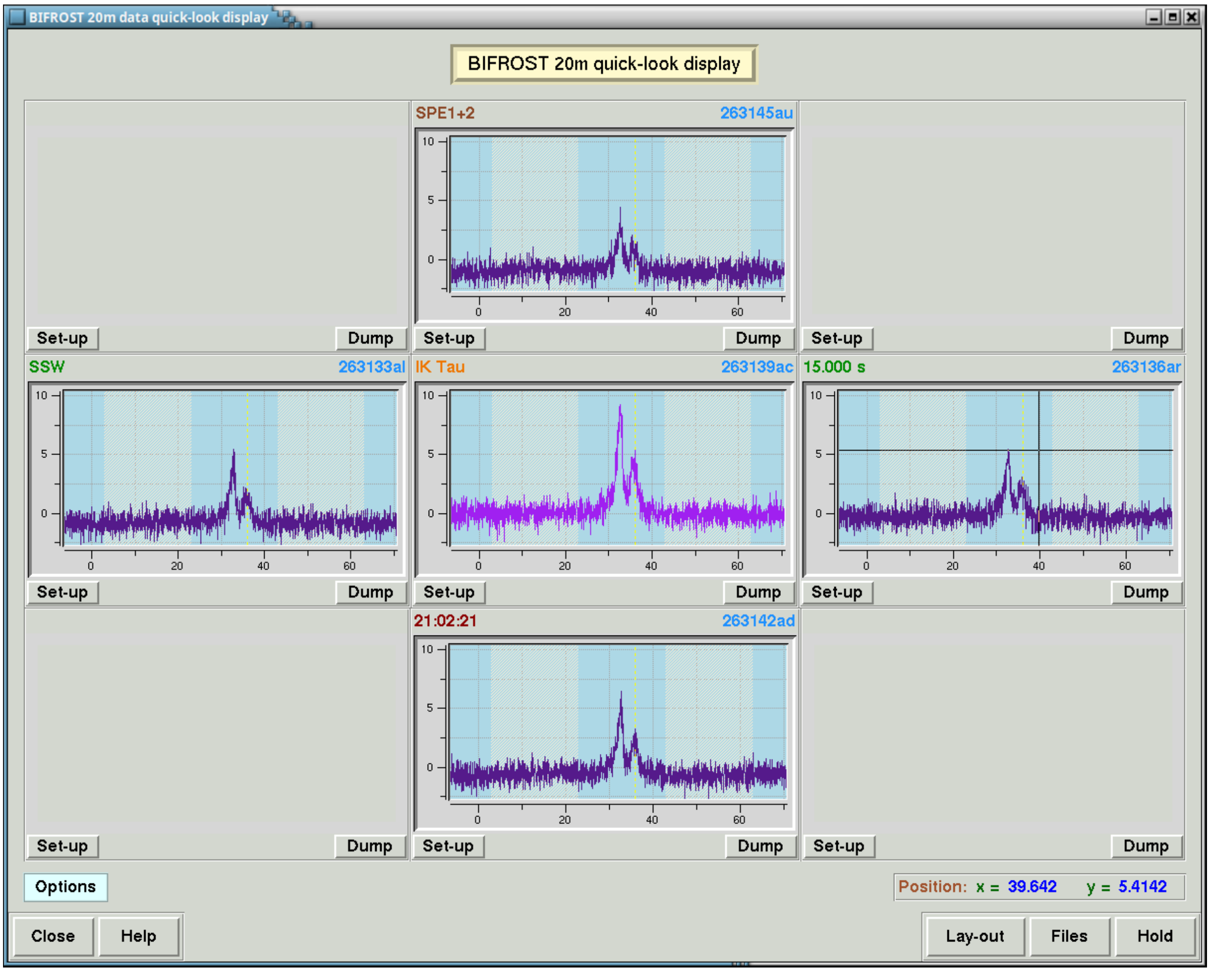
The following file types are generated by the map observation scheme:

"**s**"	Raw	raw signal	*optional (non-DSW)*

"**r**"	Raw	raw reference	*optional (non-DSW)*
"**s1**"	Raw	raw signal 1	*optional (DSW only)*
"**s2**"	Raw	raw signal 2	*optional (DSW only)*
"**r1**"	Raw	raw reference 1	*optional (DSW only)*
"**r2**"	Raw	raw reference 2	*optional (DSW only)*
"**D**"	Data	calibrated data	
"**A**"	Coadded	codded calibrated data	*if dual polarizations*
"**DA**"	Averages	averaged calibrated data	*optional*
"**AA**"	Averages	averaged coadded calibrated data	*optional (if dual pol)*

Looking at data

Since data is written as FITS-files there are many different programs available to view or reduce data. BIFROST comes with its own tools to view spectra, to check FITS-file headers or to check the status of a map observation. Those tools are the quick-look facility and the map-look facility. However, for any kind of data reduction you will need to use some other software. Programs often used at Onsala are **xs** and **class**.



The quick-look display *window can be configured in many different ways including how many plot-boxes to display. Here we see a configuration for the 20m telescope showing the results of a five-point pointing measurement. The dashed yellow line indicates the source velocity and the lighter-coloured regions show which parts of the spectra will be used for baseline subtraction. The position of the crossline cursor is shown in the small box to the lower right.*

The quick-look facility

The quick-look facility is used to show spectra during a BIFROST observing session, but can also be run in stand-alone mode to look at old spectra. The window is highly reconfigurable and can show several different plots at the same time. Different configurations are called *lay-outs* and each data type has its own lay-out. The quick-look facility will keep track of what kind of data it is supposed to display and will automatically switch between different lay-outs.

For each data type, the quick-look facility comes with a set of predefined lay-outs that you can choose from, but since the program offers lots of possibilities for customization, you can easily design your own lay-outs and use them instead.

The quick-look facility can also display FITS-files taken with the old PEGASUS system and may also be able to show FITS-files generated by other systems provided that they use some critical FITS-keywords in the same way as they are used at OSO. When displaying non-BIFROST-generated FITS-files, the mechanism for identifying different types of spectra won't work and all spectra will be treated in the same way.

Features of the quick-look display window

The quick-look display is dominated by the big plotting area which can hold up to nine different plot-boxes. Each plot-box contains a graph where the spectra are shown. Above the graph there is a title line where some information can be displayed and below the graph are two buttons: the **Dump** button can be used to generate a PostScript dump of the graph in the plot-box and the **Set-up** button takes you to a window where you can configure the contents of the plot-box.

Below the plotting area, there is a small box where you can read out the coordinates of the cursor, if you move it over a graph in a plot-box. There is a also a multiple-option menu called **Options** where you can select or deselect a number of general options, which are described further down in the "*General options*" chapter.

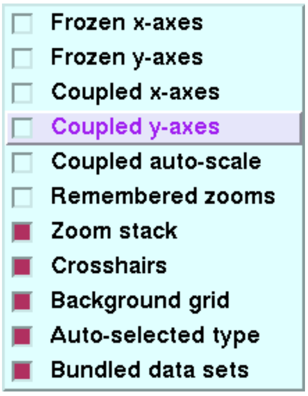
The **Lay-out** button will take you to the quick-look lay-out selection window where you can select which lay-out you want to use for each different type of data. The **Files** button will open the quick-look FITS-set selection window which lists all available sets of FITS-files. In that window, you can select different FITS-sets for viewing.

The **Hold** button is only present when the quick-look display is run as part of an observing session. It is a push-button that you can click to block the automatic updating of the window whenever a new spectrum arrives. This could be useful if you want to examine a particular spectrum while observations are going on. Once finished with that, you release the hold by clicking on the **Hold** button a second time.

General options

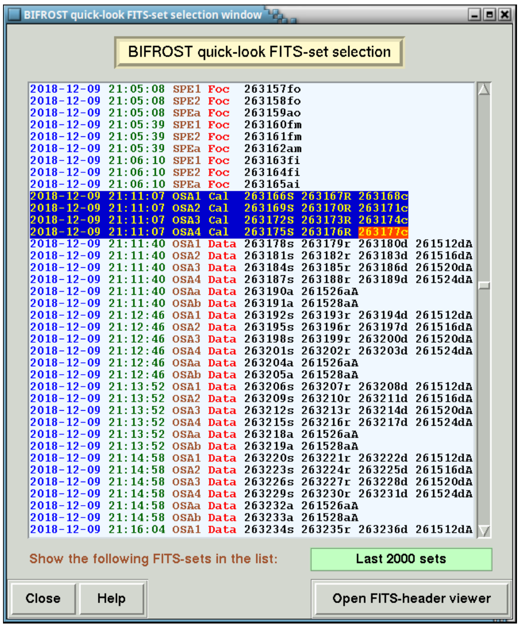
There are a number of general options controlling the behaviour of the graphs shown in the quick-look display:

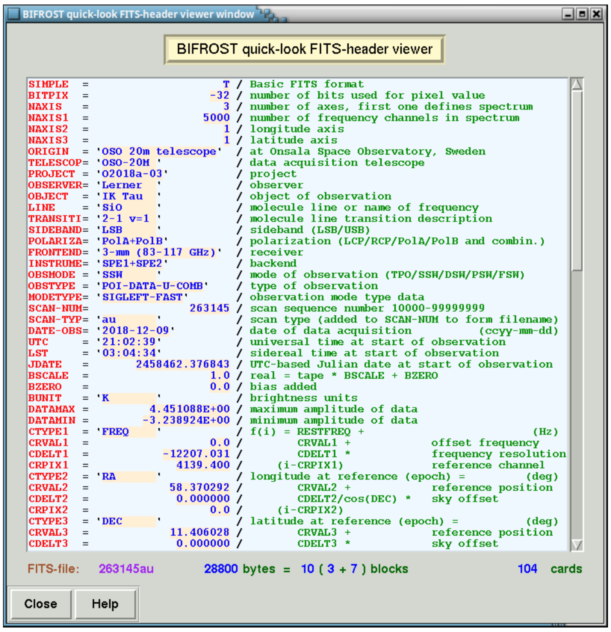
- **Frozen x-axes** and **Frozen y-axes:** control whether the scales on the axes should be reset automatically when new data is shown. Note that the axes always are reset when switching data type, unless the **Remembered zooms** option is used together with at least one of the **Frozen N-axes** options.



The **Options** *menu in the* quick-look display *window contains a set of options that control how the spectra are shown and how the plot window(s) behave.*

- **Coupled x-axes** and **Coupled x-axes:** control whether the same zoom should be applied to all plot-boxes. These options are only useful if you use a lay-out with more than one plot-box. Note that all plot-boxes need to have the same unit on the x-axis.
- **Coupled auto-scale:** controls whether the same auto-scaling should be applied to all plot-boxes so that all y-axes are scaled equally.
- **Remembered zooms:** controls whether a zoom setting should be remembered when switching data types. If enabled, the individual zoom settings (if any) for each data type will be remembered and reapplied when switching back to the same data type; otherwise any zoom settings will be cleared each time the data type is changed. You need to use at least one of the **Frozen N-axes** options for this option to be useful. A typical situation to use this option is when you are observing a narrow line but still want to see the full bandpass for the calibrations: you should then select **Frozen x-axes** and **Remembered zooms**, and zoom in properly on the line. When a calibration comes, the full bandwidth will be shown, but then when the next data spectrum comes, the zoom in on the x-axes is reapplied.
- **Zoom stack:** controls whether the quick-look display should keep a zoom stack so that if you zoom in with several steps, then you will zoom out through the same steps. If you don't use it, then any zoom out will immediately restore the full size of the plot.
- **Crosshairs:** controls whether you want to see crosshairs or not in a plot-box when you move the mouse over it.
- **Background grid:** controls whether there is a background grid or not.
- **Auto-select type:** controls whether the quick-look display automatically should switch to show the data type of the last received observation. This option is only used for on-line observing sessions. If enabled, the quick-look display will automatically switch the lay-out to match the data type of the last observation, for example switching from showing data to showing a calibration. If not selected, then only observations matching the currently selected data type will be shown; for example, if the type **Data** is shown, then calibrations will not be shown automatically.
- **Bundled data sets:** controls whether, for example, the different positions in a pointing should be considered as one data set or five individual ones.





The quick-look FITS-set selection *window (to the left) where your FITS-sets are listed. The FITS-sets selected to be shown in the* quick-look display *window are marked with a blue background. The* quick-look FITS-header viewer *window (to the right) allows you to inspect the contents of a FITS-header. You select the FITS-file to view from the* quick-look FITS-set selection *window.*

Selecting FITS-files to view

During an observing session, the latest obtained spectra will always be displayed automatically once they have been obtained. It is also possible to look at older spectra by using the quick-look FITS-set selection window, which is opened via the **Files** button in the quick-look display.

The data is shown as FITS-sets consisting of data that belong together for a certain backend channel (like the signal and reference spectra as well as the finished data product). Each line contains a FITS-set and shows the date and time when the data set was acquired, the backend used, the type of FITS-set and the FITS-files associated with the set.

You select a FITS-set to be shown in the quick-look display window by left-clicking on a line. The marked set is shown with blue background. There will also be up to four earlier sets of the same type marked with a yellowish background. These are used for graphs that are including older data. They will be used if you, for example, have a lay-out where you plot both current and previous data. A set can consist of several lines, if there is data taken at the same time with different backends or if the **Bundled data sets** option is selected.

The **Show the following FITS-sets in the list** menu can be used to select how many FITS-sets you want to see. There are options both for selecting a certain number of the most recent FITS-set or for selecting all FITS-sets taken during a recent time period. The default is **Today & yesterday**.

The quick-look FITS-set selection window is updated automatically with new data when it becomes available, when you run the quick-look viewer from a BIFROST observing session.

Viewing FITS-file headers

The quick-look FITS-header viewer can be used to look at the FITS-header of selected FITS-files. The window is opened by clicking the **Open FITS-header viewer** in the quick-look FITS-set selection window.

The FITS-header is showed with some colour-coding in the big box. Below the box is a single line with some meta-data about the FITS-file: its name, its size and the number of *cards* (i.e. lines with keywords) the FITS-header consists of. The size is given both in bytes and in number of 2880-byte blocks. You will also see how many of those blocks were used for the header and how many were used for the data.

You select the FITS-file in the quick-look FITS-set selection window. If you left-click on a FITS-file, you select that file and the spectra from that FITS-set will be also be displayed in the quick-look display window. If you don't want to show the spectra, you can middle-click on the FITS-file instead. The FITS-file selected for display in the quick-look FITS-header viewer is marked with an orange background.

Using different lay-outs

A lay-out is a configuration of the quick-look display window. It specifies how many plot-boxes should be shown, what data should be shown in each plot-box as well as how the data should be shown. BIFROST keeps track of what type of data it is showing (for example calibration, pointing or science data) and uses different lay-outs for each data type and it switches between them automatically depending on the data type.

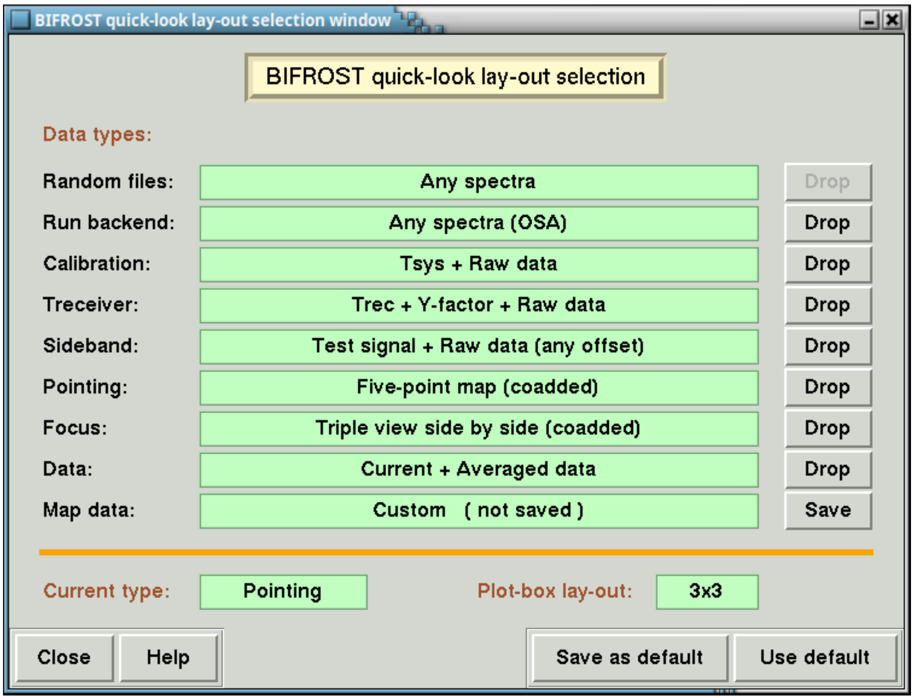
For each data type, there are a number of pre-defined lay-outs and you can use the BIFROST quick-look lay-out selection window to select which lay-outs you want to use. You can also create your own lay-outs and use them instead, if you prefer that.

The larger part of the window is the list of all different data types and the menus with the corresponding lay-outs. There is also a small button that typically says **Drop**. That button will permanently delete the selected lay-out. Since you must have at least one lay-out for each data type, the **Drop** button will be disabled if there is only one lay-out.

If you have modified a lay-out, the text on the menu will change to **Custom (not saved)** and the small button will from **Drop** to **Save**. You can then use the **Save** button to save the new lay-out. You will be asked for a name for the new lay-out before it is saved.

There are two menus below the orange line: the **Current type** tells you which data type is being shown in the quick-look display window, while the **Plot-box lay-out** menu tells you the current lay-out of plot-boxes. You only need to change this menu, if you are going to design your own lay-out with a different set of plot-boxes.

There are two buttons in the lower right corner: **Save as default** will save all the currently selected lay-outs as defaults for the future, while the other button, **Use default**, can be used to restore the defaults if you have been playing around selecting other lay-outs.



The quick-look lay-out selection *window is used to select which lay-out to use for each type of data. BIFROST comes with a number of predefined lay-outs, but you can easily create your own lay-outs and save them. In this example, the observer has created a new lay-out, which is used for map data, but the new lay-out has not yet been saved for future use.*

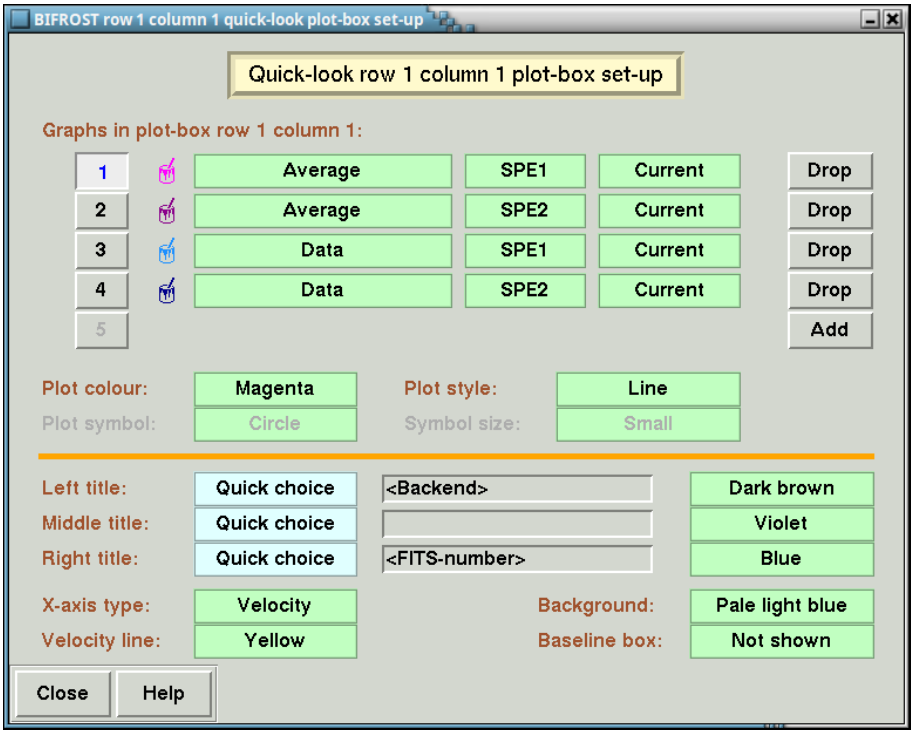
Customizing a lay-out

It is easy to customize a lay-out. You just click the **Set-up** button in the plot box you want to modify, and make the changes in the plot-box set-up window.

If you want to create a new lay-out, the best thing to do is to go through the available lay-outs and select the one that is most similar to what you want. If you want to change the number or arrangement of plot-boxes, you should do that with the **Plot-box lay-out** menu in the BIFROST quick-look lay-out selection window before you start modifying the individual plot-boxes.

A modified lay-out is used as long as the quick-look facility is running, so if you want to save it for use in future observing sessions, you should go into the BIFROST quick-look lay-out selection window and click on the **Save** button shown next to the **Custom (not saved)** option.

The upper part of the plot-box set-up window contains a list of the graphs to be shown. Each line has three menus that is used to determine what spectra to show: the first menu is used to select what type of data to show while the second menu is used to select backend. The third menu is adding a time aspect and can be used to select the current spectra or a previous spectra, if you want to compare them.



Each plot-box has a set-up window where you can set up what should be shown and how it should be shown. In this example the last data spectra as well as averaged spectra (if available) will be shown for the two channels of the SPE-backend. The part above the orange line is a list of the spectra to be shown and attributes that can be set individually for them, while the part below the line is used for setting titles and other general attributes.

You can remove graphs by clicking on the **Drop** button and add more graphs by clicking the **Add** button. You can have up to 18 graphs in a plot-box. Only the graphs for which there is data will be shown, so if you have a lay-out with graphs for different backends, and then you only observe with one of them, only those spectra will appear.

When the graphs are plotted in the plot-box, the list is processed in reverse order, so the graph with the highest number is added first and the graph with the lowest number is added last. Thus, if you show individual spectra together with averages in the same plot-box, you want to have the averages in the beginning of the list so they are plotted on top of the noisier individual spectra.

The colour of each graph is indicated with the small paint can symbol. To change the colour or the plot style, click on the button with the number of the graph to select it and then use the **Plot colour**, **Plot style**, **Plot symbol** and **Symbol size** menus as desired.

Below the orange line are some general options. You can have up to three titles shown in three different colours on the title line above the graph in the plot-box. You fill in the title text in the entry fields. You can use the **Quick choice** menus to add tags that tell BIFROST to dynamically insert information from the FITS-headers into the titles. The tag "`<FITS-number>`", for example, will be replaced by the actual FITS-number(s) of the spectra being shown. You can mix text and tags as you wish.

There are four more options at the bottom of the plot-box set-up window: the **X-axis type** which selects the unit to use for the x-axis, **Background** which sets the background colour, **Velocity line** which selects the colour of a vertical line indicating the desired frequency, and **Baseline box** which sets the background colour used to mark which part of a spectrum that is used for calculating baselines. The last two options are only used provided that they are not set to **Not shown** and that the proper data is found in the FITS-header.

You can only have one plot-box set-up window open, so if you click on another **Set-up** button, the currently open plot-box set-up window will be closed and replaced by a new plot-box set-up window.

The map-look facility

The map-look facility is used to check the status of a map observation and to follow its progress during a BIFROST observing session. It can also be run in stand-alone mode, and you can actually even follow the progress of a map observation in stand-alone mode, although the updates typically show up with a few seconds delay and not instantly as when run as part of an observing session.

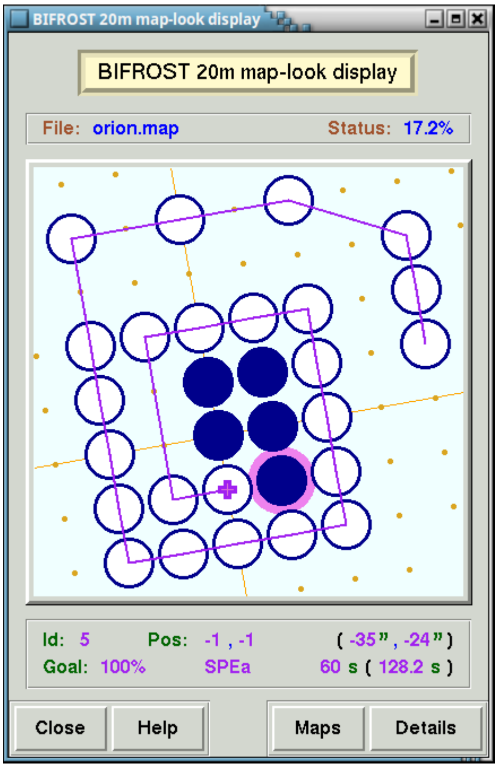
Technically, the map-look facility is part of the quick-look facility and it gets its data from the map files used for map observations. It is only available for telescopes supporting the map observation observing scheme, i.e. the 20m and 25m telescopes.

Features of the map-look display window

Unless you have changed the preference setting, the map-look display window will pop up automatically once you start observing a map. The window is dominated by the big map box, where the map positions in your map are shown. The map positions are shown as circles which are colour-coded to show the progress of the map: a white circle is a position that has not yet been observed and a dark blue circle is a map position that has been finished. Map positions that have been observed, but have not yet reached the mapping goal will be shown in various shades of blue depending on how far they are from reaching the goal.

There is a small box above the map which tells you the name of the map file and its overall status which either is given as a percentage of how far it is from being completed or the total observing time so far, in case it is en open-ended map.

Below the map there is another somewhat bigger box which tell you information about an individual map position. It has two lines, where the first line tells you the position of the map position both in units of the map spacing and in arcseconds as well as the internal ID of the particular map position. The second line tells you the status of the map position, either as a percentage of completeness or as the total integration time (in case of open-ended maps). It also gives you the name of the backend used when checking the map goal. The integration time is given to the right together with the *comparative integration time* given within parenthesis. The comparative integration time is the integration time you would have needed to get the same data if the observed Tsys would have been equal to the *comparative system temperature* you have selected in the Calibration values menu (which defaults to 300 K, if you haven't changed it).

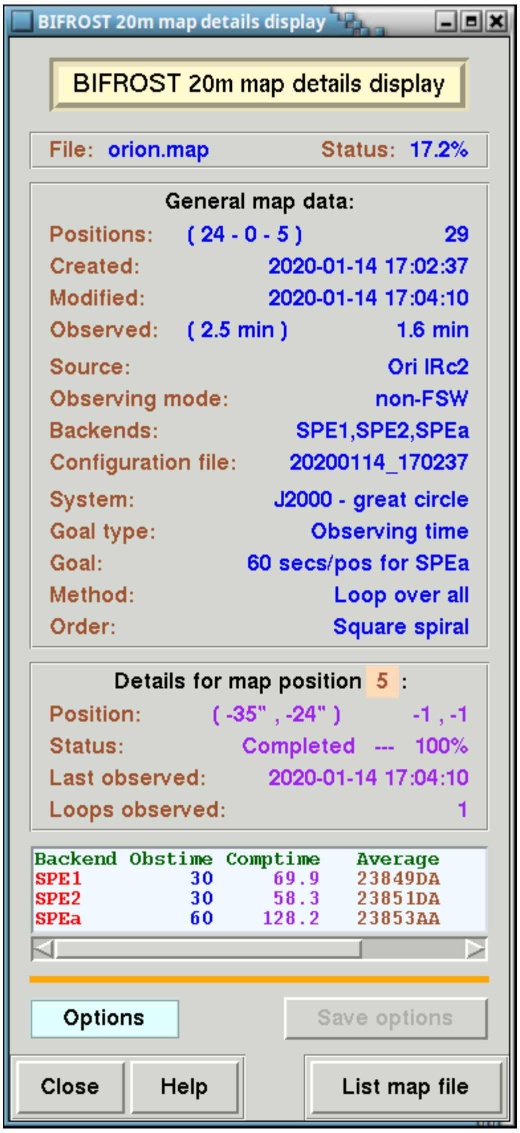


The map look display *showing the progress of an ongoing map observation. Five positions have already been observed with the last observed position marked with a purple ring. Information about this position is shown in the box (with purple text). The next position to be observed is indicated with the purple plus sign and the purple line indicates in which order the remaining map positions will be observed. Moving the mouse over another position will display data on that particular position (with blue text) and clicking on an observed position will send a signal to the* quick-look facility *to display the spectra from that map position.*

When you move the mouse in the map box, there will be a green disk indicating the closest position on the map grid. If there is a map position on that grid point, the data on that particular map position will be shown in the box below (with **blue** letters). If you click on a map position that has been observed, the spectra from that position will be shown in the quick-look facility.

If you leave the mouse outside the map box during an observation, then the data on the last observed map position will be shown automatically in the lower box (with **purple** letters). That specific map position will also be marked by a purple ring in the map box.

The map box do not support zooming — instead it will auto-adjust the size of the map so that the complete map always is shown. You can do some customization of the map box, similar to what you can do in the map editor. However, to save space, those options have been put in the **Options** menu which you will find in the window you get when clicking the **Details** button.



Clicking on the **Details** *button in the* map look display *window brings you the* map details display *window where you can see more details about the current map as well as data on the current map position (with purple text if it is the last observed position and blue text if it is a position below the cursor). Clicking the* **List map file** *button opens the* map file list display *window, where you can see a colour-coded version of what the map file actually looks like.*

You can use the **Maps** button if you want to load another map file into the map-look facility, while the **Details** button gives you more details about the map and the selected map position.

Getting the details of a map

The map details display window gives you much more details about the map and a selected map position. It contains four boxes, with the two top ones giving you general data on the map, while the two bottom ones gives you information about a specific map position. This position is selected in the map-look display window. It refers to the map position below the cursor when you move the mouse in the map box of that window, or to the last observed position if the mouse is outside the map box. In the first case, the text in the third box is shown with blue letters, while the letters are purple in the second case).

The fourth box provides data on each individual backend and backend channel used, including observing time and the name of the corresponding FITS-files. This box adapts its size depending on how many backend channels are used.

There is an option in the **Options** menu, if you want to see the data on the specific map position but want to skip the general map data to save space on your screen.

If you are really interested in looking at the map file itself, you can use the **List map file** button to bring up the map file list display window which will show you the map file as it is. It will add some colour-coding to help you read it, but that is also the only help you will get. You will recognize a number of the parameters shown as the parameters you selected when you created the map, but they are mixed with a set of internal parameters that BIFROST normally won't show you since they are part of the internal book-keeping.

Documenting an observing session

It is, of course, important to know what has been done during an observing session and BIFROST does provide a set of files and documents to help you with that.

Observing logs

The observing logs are written in the *Obslog* directory in the project area for the 20m and 25m telescopes — for the other systems, logs are only written to the */archive/Obslog* area. There is one log file per UTC-day and they are called "*bifrostlog_YYYYMMDD*" (or "*Obslog.YYYYMMDD*" on */archive/Obslog*), so an observing session running over UTC midnight will generate several separate log files.

The format is described in the **Message formats** chapter and the files will always contain all messages generated by the BIFROST system down to message level LOG5. You will also get DEBUG-messages for tasks which experience a crash during execution. Apart from that it is not recommended to add DEBUG-messages to the log files since they will increase dramatically in size (you can add DEBUG-messages by changing one of the preferences). A full day of observations will typically generate a log-file of 5–20 MB without any DEBUG-messages.

FITS-file databases

BIFROST automatically generates two files with data about the FITS-files it has generated. Both files are located in the *Logs* directory and they are called "*FITS_scan.log*" and "*FITS_set.log*".

The "*FITS_scan.log*" is a database containing all FITS-files that have been generated. The format is ASCII and meta-data about each FITS-file is written to this file with one line of comma-separated values per FITS-file. BIFROST is not using this database for anything and there are still no programs to use the information in it, but the information is there in case someone wants to use it.

The "*FITS_set.log*" is another database where FITS-files that belong together are listed as *sets*. A *set* is typically the two raw spectra (signal and reference) together with the calibrated data product and a possible average of the data. Each backend channel forms its own set, and so does coadded polarizations. Typical lines for an observation with the "*single observation*" scheme on the 20m telescope with the first OSA-pair with coadded polarizations and preceded by a calibration may look like:

```
2018-11-07 17:22:44 O2018c-01 OSA1 Cal 20096S 20097R 20098c
2018-11-07 17:22:44 O2018c-01 OSA2 Cal 20099S 20100R 20101c
2018-11-07 17:23:37 O2018c-01 OSA1 Data 20102s 20103r 20104d 20075dA
2018-11-07 17:23:37 O2018c-01 OSA2 Data 20105s 20106r 20107d 20079dA
2018-11-07 17:23:37 O2018c-01 OSAa Data 20108a 20081aA
```

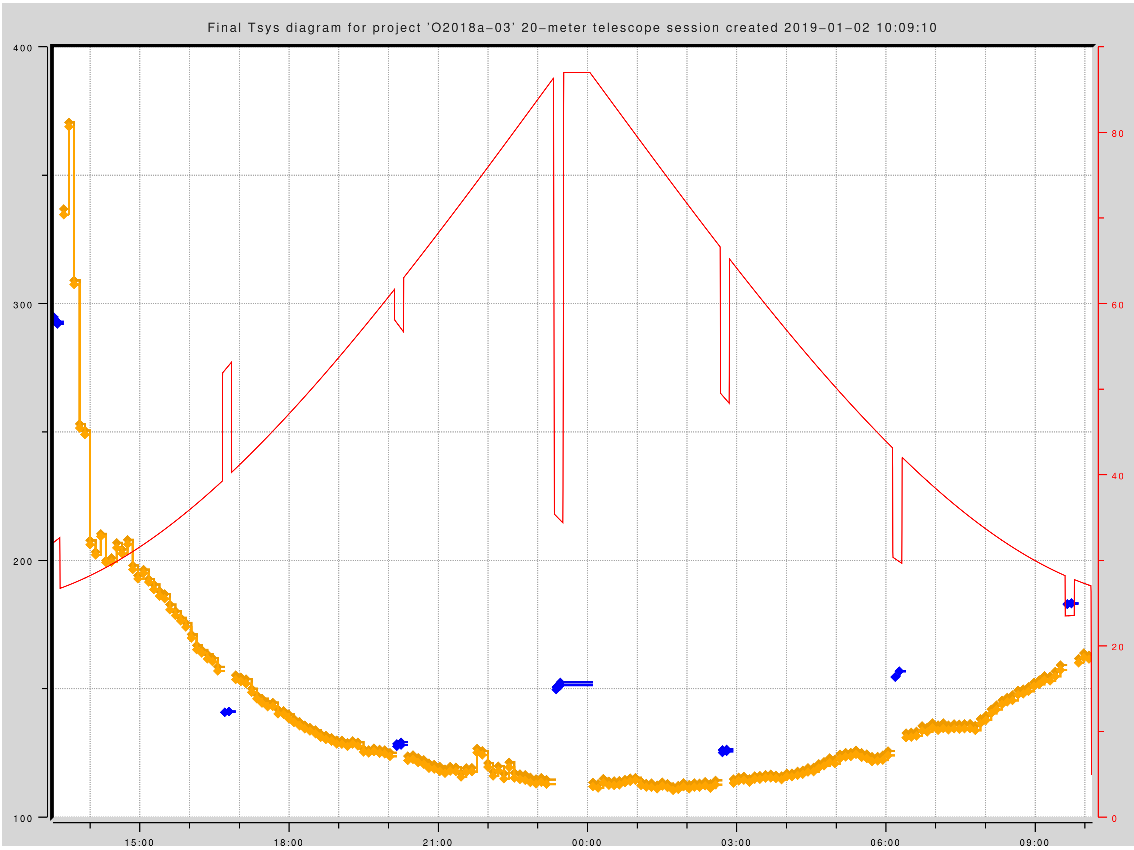
The fourth column contains a code identifying the backend channel while the fifth column contains a code specifying the observing scheme used. Both codes are three-letter or four-letter words.

The "*FITS_set.log*" is used by the quick-look facility to keep track of which FITS-files are related to each other.

Tsys-diagrams

With the Tsys display you can see how the *Tsys* has been varying during the observing session. Different frequency settings get different colours in the Tsys-diagram. You can open it with the **Open Tsys display** button in the BIFROST utilities menu. You can save the diagram shown as a PostScript-file if you like.

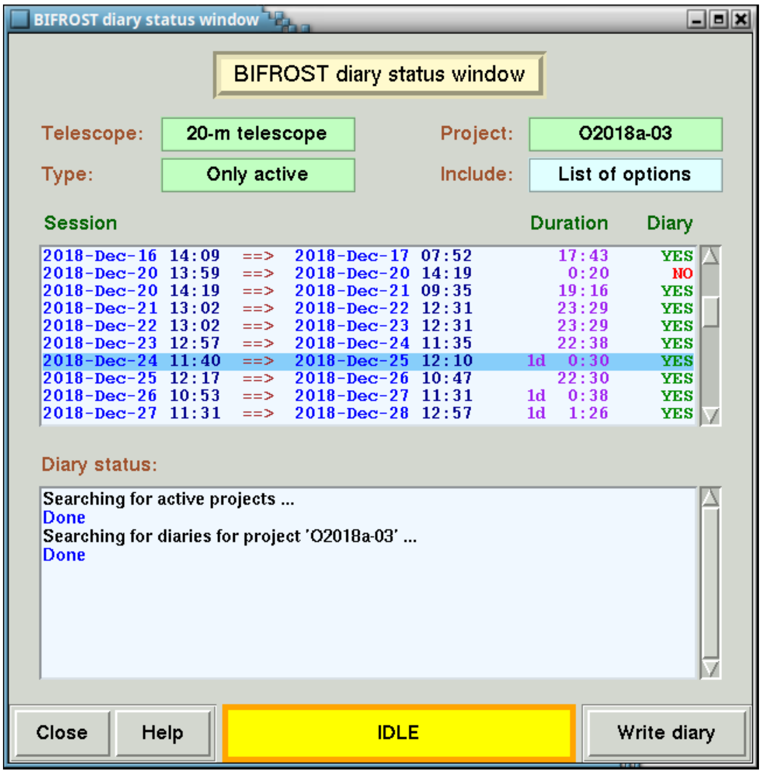
When you exit an observing session for the 20m or the 25m telescopes, BIFROST will automatically create a Tsys-diagram for you. It will be located in the *Diaries* directory in your project area, and you can have it included in your observation diary, if you want to.



An example of a Tsys-diagram that has been generated automatically by BIFROST. The change of colours represents a change in frequency. The red line and the y-scale to the right is the telescope elevation.

Observation diaries

For the 20m and the 25m telescopes, there is a possibility to create *observing diaries* which are PDF-files looking a bit like the classical observing sheets that observers for decades have been using to manually write down important information like time, scan numbers, Tsys, telescope position and sources observed. BIFROST provides a diary writer that can extract this kind of information from the observing logs and compile it into a human-readable overview. The format is landscape A4 and you have several options for what kind of information you want to include in the diary. The diary writer can also include the automatic Tsys-diagram in the report as well as a table with a break-down of how the observing time has been used (i.e. the observing efficiency).



The BIFROST diary status window *shows you the observing sessions you have and allows you to create diaries for them.*

Diaries are very useful for observations that have been performed with command files, since they will provide a comprehensive summary of what was done during the observations. This is especially true if the command file allows BIFROST to choose between different objects, or if the observations were paused because of bad weather.

The diary writer is called the BIFROST diary status window and can be launched by clicking on the **Diary writer**-button in the BIFROST tools menu which you can open with the -button, or by typing "`diarywin`" in a terminal window in your project area.

BIFROST diary for project **O2018c-01** Page 1 Period: *2018-05-25 16:37* ⟹ *2018-05-31 12:43*

UTC-time	*Duration*	*Az*	*El*	T_{sys}	*FITS-scans*	*Action*
25/5 16^{37}						Start of session
16^{38}						Using **75 GHz** model with $Az = 0$" $El = 0$" and Z-focus = 0.0 *mm*
25/5 16^{38}						Loading 'pointconf.conf' for **SiO** at **86.24343 GHz** with OSA `Frequency offset = -20.0 MHz --- OSA = 2 * 625 MHz, channels 1+2, clipped 12384+12384 channels`
25/5 16^{40}-	26^{32}					**Starting command file 'surveyjoblsb.cmd' from line 1**
16^{40}- -19^{20}	2^{40}					WAITING FOR 2 HOURS 40.0 MINUTES
19^{22}- -19^{28}	0^{06}	49°	35°	*244-245*	208826– 208857	**R Leo** — point (-1.8/-4.4, -1.5/-3.3) $Az = -1.6$" $El = -3.8$" `loops=2 azdist=21 eldist=21 ontime=20 offtime=20 caltime=10 calwhen=3 obsmode=ssw dop=eachcal adjpwr=first rawdata=skip addpol=save backend=OSAa baseline=nobaselines startpos=last apply=auto limit=relative`
19^{28}- -19^{31}	0^{04}	51°	34°	*242-243*	208858– 208877	**R Leo** — focus (-0.09, -0.13) Z-focus = −0.11 *mm* `loops=2 dist=3.5 ontime=20 offtime=20 caltime=10 calwhen=3 obsmode=ssw axis=focus dop=eachcal adjpwr=never rawdata=skip addpol=save backend=OSAa baseline=nobaselines startpos=current apply=auto`
19^{32}- -19^{37}	0^{05}	52°	34°	*244-244*	208878– 208909	**R Leo** — point (+4.0/-2.8, -1.5/-4.8) $Az = +1.3$" $El = -3.8$" `loops=2 azdist=21 eldist=21 ontime=20 offtime=20 caltime=10 calwhen=3 obsmode=ssw dop=eachcal adjpwr=first rawdata=skip addpol=save backend=OSAa baseline=nobaselines startpos=last apply=auto limit=relative`
25/5 19^{37}						Loading 'conf72.conf' for **scan72usb** at **72.000 GHz** with OSA `OSA = 2 * 4000 MHz, channels 1+2+3+4, clipped 1311+1311 channels`
19^{39}- -21^{30}	1^{51}	−136° −118°	22° 33°	*512-656* *366-449*	208910– 209085	**DR21OH** — single (366 - 656 K) 24 loops `loops=6 ontime=120 offtime=60 caltime=10 calwhen=3 obsmode=dsw dop=eachcal adjpwr=first useplm=use rawdata=skip addpol=save average=addfile`
25/5 21^{30}						Loading 'conf70.conf' for **scan70usb** at **70.000 GHz** with OSA `OSA = 2 * 4000 MHz, channels 1+2+3+4, clipped 1311+1311 channels`
21^{33}- -22^{56}	1^{24}	−117° −103°	34° 44°	*451-665* *369-526*	209086– 209217	**DR21OH** — single (369 - 665 K) 18 loops `loops=6 ontime=120 offtime=60 caltime=10 calwhen=3 obsmode=dsw dop=eachcal adjpwr=first useplm=use rawdata=skip addpol=save average=addfile`
25/5 22^{56}						Loading 'pointconf.conf' for **SiO** at **86.24343 GHz** with OSA `Frequency offset = -20.0 MHz --- OSA = 2 * 625 MHz, channels 1+2, clipped 12384+12384 channels`

An example of the first page of a diary generated automatically from an observing log.

The diary writer will check the observing logs for your project and identify the different observing sessions and check whether there already exist diaries for those sessions or not. This will be presented in a list in which you can select any observing session and generate a new diary with the **Write diary** button.

You can use the **Add comment to log** button in the BIFROST utilities window, if you want to include comments in the diary (as well as the log). If you want to include a comment from a command file, then you could use the command `NOTE` instead of the command `LOG`.

It is possible to edit the diary manually, since it is first created as a TEX-document before being transformed into a PDF with "`pdftex`". Note though that it is plain TEX and not LATEX. The PDF is written in the *Diaries* directory in your project area. The TEX-file is written to *Diaries/TeX* which will also include the format-file "*diary_format.tex*", which defines the format used in the diary and some logo-files.

Configuration log

Each time you start an observation, BIFROST will check whether the configuration of the system is the same as during the last observation, and if it is not, then a configuration file will be saved automatically. The name of this configuration file is also written to the

`CONFIGID` keyword in the FITS-headers — it is thus always possible to go back and see exactly what the configuration was. This can be used if you want to check what the configuration was or you could even use that configuration file, if you want to observe again with the same configuration.

The automatic configurations are written to the project area in the subdirectory *Configs/Logs/YYYY-MM-DD*, where *YYYY-MM-DD* is the current UTC date. The files are named "*auto_YYYYMMDD_HHMMSS.conf*", and it is the "*YYYYMMDD_HHMMSS*" part that is stored in the `CONFIGID` keyword.

The set-ups stored in the automatic configuration files are: the signal path (receiver, frequency & backends) and the calibration values. For the 20m and 25m telescopes, the antenna parameters (limits & wrap behaviour) and pointing model are included as well. For the 20m telescope, the focusing model and the obsmode set-up are also included.

Index