CLASS
Continuum and Line Analysis Single-dish Software

A GILDAS working group software

Document probably older than you think

```
Continuum and Line Analysis Single-dish Software

                    or

                              Sensible
                              Simple
Continuum and Line Analysis   Sophisticated System (*)
                              Sympathetic
                              Super
```

Version 4.0

The GILDAS working group is a collaborative project of the Observatoire de Grenoble (1,3) and IRAM (2), and comprises: G. Buisson[1], L. Desbats[1], G. Duvert[1], T. Forveille[1], R. Gras[3], S. Guilloteau[1,2], R. Lucas[1,2], and P. Valiron[1].

(1) Laboratoire d'Astrophysique
Observatoire de Grenoble
BP 53 X
414 Rue de la Piscine
F-38402 Saint Martin d'Hères CEDEX

(2) Institut de Radio Astronomie Millimétrique
300 Rue de la Piscine
F-38406 Saint Martin d'Hères

(3) CEPHAG
Observatoire de Grenoble
F-38402 Saint Martin d'Hères CEDEX

Contributions from and invaluable discussions with J. Cernicharo, P. Begou[3], S. Delahaye[3], A. Dutrey[1,2], C. Kahane[1], P. Monger, J.L. Monin[1], R. Neri[2] and all GILDAS users are gratefully acknowledged.

Contents

1 Introduction

CLASS is a software package for reducing spectroscopic data, and also continuum drifts obtained on a single-dish telescope. CLASS supersedes the older **LAS** program which was restricted to spectroscopic data.

The originality of CLASS with respect to similar systems already in use is in the way an observation may be identified. In addition to the traditional scan number which can be used to uniquely refer to an observation, the system also enables one to use *Selection Criteria* as in a data base management system. This faculty, added to a powerful command monitor, SIC , allows easy manipulation of large volumes of data ; the list of observation numbers to be added to get the mean spectrum at one position need no longer be typed in, but may be found by CLASS itself.

CLASS is divided in different parts, called "Languages", which have somewhat independent functions (*). Language LAS contains all the general utility functions to handle the data structure, plot the spectra or drifts and calibrate them. Language ANALYSE contains functions to analyse calibrated spectra in more detail. Language GTVL includes commands for basic graphic actions performed by the graphic library used by CLASS; the same library is used by GREG .

On a standard installation, CLASS is entered by just typing `CLASS`. In addition to this manual, the reader should consult the SIC manual, and for further processing, the GILDAS and GREG manuals.

2 Cookbook

This part is a list of recipes enabling the beginner or the occasional user the get on the air very rapidly, without losing his time searching the system's on-line HELP facility. A CLASSic session is the following:

```
DEVICE XLAND WHITE                    ! 1
FILE IN BRUT.BUR                      ! 2
SET LINE 13CO(1-0)                    ! 3
SET SOURCE IC348                      ! 4
SET TELESCOPE IRAM-30M-B30            ! 5
SET OBSERVED 15-AUG-1984              ! 6
SET ANGLE SECONDS                     ! 7
SET COORDINATES EQUATORIAL            ! 8
FIND/OFFSET 0  25                     ! 9
SUM                                   ! 10
!
SET UNIT VELOCITY                     ! 11
SET MODE X -1 14                      ! 12
SET MODE Y -0.5 7.5                   ! 13
SET PLOT HISTOGRAM                    ! 14
PLOT                                  ! 15
HARDCOPY/PLOT                         ! 16
!
SET WINDOW 3 6 8 10                   ! 17
BASE 4/PLOT                           ! 18
PLOT                                  ! 19
!
LINES 0                               ! 20
GAUSS                                 ! 21
FIT                                   ! 22
RESIDUAL                              ! 23
PLOT                                  ! 24
!
FILE OUT REDUIT.30M NEW               ! 25
SWAP                                  ! 26
WRITE                                 ! 27
!
SAVE IC348                            ! 28
EXIT                                  ! 29
```

2.1 Getting on the air

CLASS needs a data file, as well as (usually) a graphic output device. The first thing is then to define the device to be used, (command 1), here an X-window terminal, on which an landscape style window with white background is created. The input file containing spectra is then opened (command 2).

2.2 Adding Spectra

Usually one begins by adding up the individual spectra obtained at a single position on the sky. To do this, define the Selection Criteria of the spectra, including the Line Name (3), the Source Name (4), the Telescope and backend configuration used (5), and the date of observations (6). In this example we consider all the observations of the 13CO(1-0) transition performed using the IRAM 30-m telescope in the direction of IC348 on 1984 August 15th. Commands (7) and (8) further define the angular coordinate system to be used, and the corresponding units. Executing the `FIND` command (9) builds up an *Index* of these observations, where the selection criteria defined by the previous `SET` commands are further extended by the `/OFFSET` option which limits the search to the position (0.,25.) (in equatorial coordinates and arc seconds, as defined by (7) and (8)). The `SUM` command then adds up all the spectra of the current index.

2.3 Visualization

To see the result, command (11) defines the X axis units (here Velocity, but channel number or Frequency may be used). Then commands (12,13) define the plotting limits (in Velocity and Antenna Temperature respectively). Command (14) specifies that the spectrum will be plotted as a histogram, instead of the simpler, but less realistic, segmented line. The `PLOT` command (15) draws the axes, labels, the spectrum and its title. Each of these operations may be performed separately if needed.

A hardcopy of this plot can be obtained as in command line (16). This creates a file which is automatically sent to a plot spooler. The plot appears on the local system plotter device.

2.4 Calibration

Generally the spectrum baseline is not perfect. One may thus want to subtract a baseline, calculated by a least-squares fit, excluding the intervals defined in command line (17) 3-6 and 8-11 (in velocity units according to (11)) in which the observed line is supposed to be present. This is done by command line (18), which computes a 4th degree polynomial (argument of the `BASE` command), plots the result superimposed on the preceding plot, then subtracts this polynomial from the spectrum. The original version of the spectrum is copied in the second "memory" of the system (T memory). The result of the baseline subtraction, in memory R, is displayed by the `PLOT` command (19). Note that the limits defined in (12,13) stay in use.

2.5 Analysis

If the result is satisfactory, the spectrum is further analyzed by more sophisticated means, such as model-fitting by Gaussian curves. One defines the number of curves to be used (20); 0 means the automatic minimization with a single Gaussian, without initial values for the parameters. The `GAUSS` command (21) activates the minimization algorithm, which finally converges (yes it does !). One plots the fitted Gaussian by the `FIT` command (22). The residuals may also be determined by the `RESIDUAL` command (23) which first copies the R "memory" into the T "memory", then subtracts the Gaussian. These residuals are then plotted (in the current limits) by the `PLOT` command (24).

2.6 Writing

If everything seems OK, one usually saves the results of the fit. One must first open an output file (25) ; in this example the file is created by means of the last argument "`NEW`" ; to open an existing

file, omit it. Then go back to the true spectrum, not the residuals, by exchanging "memories" R and T by the `SWAP` command (26), and copy the contents of the R "memory" onto the output file by the `WRITE` command (27). The limits of the plot, the baseline, the Gaussian parameters as well as the list of observations used to obtain this final result are automatically saved in the same operation.

2.7 End of session

You can simply get out by `EXIT` (29), but if you want to keep the current "environment" of CLASS , e.g. all the parameters defined by `SET` commands, use the `SAVE` command (28), which writes the corresponding `SET` commands on a command file (here `IC348.CLASS`). Later you will be able to re-execute all these commands and rebuild the previous environment, by entering : `@ IC348`, thus avoiding a lot of tedious typing.

That's it, you now know (nearly) everything. Good luck...

3 CLASS Manual

3.1 More about CLASS

3.1.1 Files

CLASS uses two files of data; one for input and one for output, which may be the same actual file. The input file is only used to read. An observation contains several independent sections. These file are defined by the command `FILE (IN,OUT,BOTH) Filename`, possibly followed by `NEW` if a new file is to be initialized. The default extension of files is site dependent : it is `.30M` at Pico Veleta, `.BUR` at the Groupe d'Astrophysique in Grenoble and on Plateau de Bure, and so on).

Commands `FIND` and `GET` operate only on the input file. Command `WRITE` operates only on the output file. When both are identical, `WRITE` creates physically a new version of the spectrum in the file, keeping the previous one. In some cases one may override the old version by the new one by the `UPDATE` command. To do this, the file must be used for both input and output, and each of the sections of the new version must fit in the space used by the old one. Other commands are subject to similar restrictions, in particular `KEEP` to save the gaussian fit results. If you want to play safe, always use `WRITE`.

CLASS also keeps a log file (named `SYS$LOGIN:CLASS.LOG`, created by the SIC monitor) and a message file `SYS$LOGIN:CLASS.MES`, which are systematically purged at the end of each session. These files may be used to keep track of a batch or interactive work. Finally the session environment may be saved as a command file containing a set of `SET` commands.

3.1.2 R and T Memories

CLASS keeps 2 observations in memory, one in an area called the R memory, the other in the T memory. The R memory is the only one that may be accessed directly ; the T memory is only used for operations on spectra (additions,...). This works like a HP calculator. The command `SWAP` exchanges both memories.

3.1.3 Observation and Version Numbers

Within CLASS , an observation should represent a single observing configuration, e.g. for spectra a single direction observed at a single central frequency with a single spectral resolution and in one polarization only. All versions of a given observation represent different stages of the data reduction. Each time the observation is modified (using `UPDATE` or `WRITE`) the version number increases. In principle, in CLASS , only the last version of a given observation is relevant. Previous versions are never deleted, with the exception of `UPDATE` command which overwrites the current version, so you can go back to previous stages of reduction in case of big mistakes.

Provided you respect this use of the version number, data reduction can be largely automated. Failing to do so, i.e. using the same observation number for very different things, implies that you have to remember yourself which version corresponds to which configuration. This was unfortunately the case for previous releases of the **OBS**erving program on the 30-m (See **CAL** manual), but should be solved in the new version for which Scan numbers and Observations numbers are different. Note that, for bookkeeping purpose, CLASS keeps also track of a Scan number, which can be used as a selection criterium.

3.1.4 Adding Spectra

Four parameters define the way spectra are added. These are the align mode, the combination mode, the integration weighting, and the behaviour with respect to bad channels.

Four alignment modes are available, by the means of the command
`SET ALIGN Mode :`

- `CHANNEL` in which spectra are added channel by channel. This is only useful when the spectra have been obtained in strictly identic conditions. Warning messages are given when this is not the case.
- `VELOCITY` in which the velocity scale is used to align the spectra. This enables you to add spectra of different origin. An interpolation is performed if needed. If individual spectra have differing spectral resolutions, the lowest spectral resolution is used for the result.
- `FREQUENCY` in which the rest frequency is used to align the spectra.
- `POSITION`, in which continuum drifts are aligned regarding to the position along the drift.

`CHANNEL` and `POSITION` only are relevant for Continuum observations, while `CHANNEL VELOCITY` and `FREQUENCY` are relevant for Line observations.

Two combination modes are possible with the command
`SET ALIGN MODE Combination :`

- `INTERSECT` where only the intersection of individual spectra is kept
- `COMPOSITE` where the reunion of the individual spectra is kept (as in a spectral scan for example)

Three weighting types may be used, with the command
`SET WEIGHT Type :`

- `TIME` for weights proportional to the observing time, divided by the square of the system noise ;
- `SIGMA` for weighting by the inverse square of the rms noise of each individual spectrum.
- `NONE or EQUAL` for equal weighting. Caution : equal weighting behaves differently in `SUM` and `ACCUMULATE` commands. `SUM` produces the average of spectra, while `ACCUMULATE` gives the sum of the two spectra. After division by the number of added spectra, `ACCUMULATE` will thus give the same result as `SUM`.

Bad channels are dealt with in two possible ways, defined by the command
`SET BAD Mode :`

- `OR` where resulting channels are declared bad if they were declared as such in at least one of the individual spectra ;
- `AND` where resulting channels are declared bad if they were bad in all individual spectra.

Default values are `ALIGN CHANNEL INTERSECT`, `WEIGHT TIME`, and `BAD OR`.

Two other parameters control whether summing spectra is allowed or not. Positions are checked according to `SET MATCH Tolerance` or `SET NOMATCH`. If (absolute) positions differ by

more than the tolerance parameter, an error message is generated. The tolerance is specified in current angle units. The homogeneity of the calibration is checked according to the `SET CALIBRATION Beam_Tolerance Gain_Tolerance` or `SET CALIBRATION OFF` commands. Beam_Tolerance is the maximum difference allowed in the beam efficiencies to add spectra (default 0.02) and Gain_Tolerance the maximum difference between the gains in the image band (default 0, which means not checked).

There are two ways of adding spectra : the commands `SUM` and `ACCUMULATE`. `SUM` operates globally on all the spectra in the index, while `ACCUMULATE` adds the R and T memories into R. `SUM` is generally better for systematic methods, `ACCUMULATE` for special cases. The drawback of `ACCUMULATE` is in the need for initialization ; one needs a spectrum in T and a spectrum in R to begin with ...

3.1.5 Selection Criteria

Default selection criteria are defined by the `SET` command. For most selection criteria, an option to the `FIND` command exists, with the same name, which may be used to impose temporary values to the `FIND` command ; the default values are unchanged by the `FIND` options.

- `SET TYPE Name` is more than a selection criterium, since it specifies on which type of observations the CLASS program works. Name can be "Continuum", "Line" or "Spectroscopy", or "Skydip".
- `SET LINE Name` for the line name to be used. A line name of the form `ABC*` indicates that all lines beginning by `ABC` are to be selected. The default is `*`, i.e. any line name.
- `SET NUMBER n1 n2` for the range of observation numbers. Default is `* *`, i.e. any observation number ; `* n2` specifies all observation numbers smaller than n2.
- `SET OBSERVED d1 d2` for the range of observing dates. A date is specified in the format `dd-mmm-yyyy`, e.g. 19-jan-1985. Default is `* *`, i.e. any date ; `19-JAN-1985 *` means any date later than January 19^th, 1985.
- `SET OFFSET o1 o2` for offsets of the position to be used (in the system and units specified by `SET COORDINATE` and `SET ANGLE`). Default is `* *`.
- `SET RANGE w e s n` is a less restrictive way to specify position offsets. A rectangular area of sky is defined by its west, east, south and north limits (in current angle units).
- `SET REDUCED d1 d2` for a range in reduction dates ; same specifications and defaults as for `SET OBSERVED`.
- `SET SOURCE Name` for the source name ; same specifications as `SET LINE`.
- `SET SCAN s1 s2` for a range of original scan numbers. Scan numbers should not be confused with Observation numbers (the numbers by which an observation is uniquely identified). They are essentially "historical" numbers defined by the acquisition system, but usually with different "observations" (in the CLASS meaning) for a single scan. The scan number is kept only for bookkeeping purpose.
- `SET TELESCOPE Name` for the Telescope name. For the IRAM 30-m telescope, the telescope name contains coded into the last 3 letters the backend used for the observations. Similar conventions are used for the POM-2 and IRAM 15-m telescopes.

The tolerance parameter defined by `SET MATCH` also influences on the position searches, since this parameter (in the current angle unit) is used to check agreement with the specified limits. Another option to the `FIND` command is `/ALL` which enables one to find all the versions of of observations satisfying the selection criteria (otherwise only the most recent version is selected). Note that the system is intended to work only with the last version of observations, so that the use of the `/ALL` option should remain exceptional.

3.1.6 The Plot

Plotting data is done by using GREG. All GREG commands are available in CLASS.

Plotting spectra is controlled by several parameters

- `SET UNIT Type` defines the unit of the X axis, which may be C (for Channel number), V (for Velocity), F (for Frequency) or I (for Image).
- `LAS\SET PLOT Type` defines the plotting type `PLOT` (Normal or Histogram) ; Normal gives straight lines connecting the data points (this is the default since it is faster). Histogram gives a more realistic representation of spectroscopic data.
- `SET MODE X (or Y) Type` defines the plotting limits in X or Y, where type stands for `TOTAL` (all channels plotted in X, complete scale in Y), `AUTO` (take the plotting limits in use when the spectrum was last written), or two numbers for fixed limits ; X or Y specify the axis on which the type is to apply. For X axis, the limits are in the current units (C, V or F). For F, specify the offset from the rest frequency in MHz (note: the caption and the numbers on the axis will refer to absolute rest frequencies).

Several commands result in plotting. These are :

- `BOX`, which plots the frame. The Y axes are labelled in temperature units ; the X axes may be in the following units : Velocity, Frequency, Image frequency, or Channel number. The upper X axis may be labelled in a different unit than that of the lower axis. Units for both axes are entered by the command `SET UNIT L U`, where L and U stand for the units of lower and upper axes and may be any of V, F, I, or C. The second parameter U is optional ; if not entered, it defaults to L.

  `BOX` accepts the option `/UNIT` which specifies a unit temporarily different from the current one (given by the `SET UNIT` command). The parameter `UPPER` will modify only the unit for the upper axis of the frame. For instance : `BOX /UNIT F UPPER` will give velocities on the lower axis (if this is the current unit specified by `SET UNIT V`) and rest frequencies on the upper axis.
- `SPECTRUM`, which plots the spectrum, in the current mode, clipped into the current box. An offset may be given as argument to plot two spectra above each other for comparison.
- `TITLE`, which writes a header above the frame. The title format is controlled by the `SET FORMAT` command.
- `PLOT`, which performs all of `CLEAR; BOX; SPECTRUM; TITLE` in a single operation.

3.1.7 Baselines

The BASE command subtracts polynomial baselines of degree < 30. The fitting algorithm uses Chebyshev polynomials, and does not allow any extrapolation outside the fitting range. It is thus important to fit the baseline out to the maximum extension of the wanted spectrum. If extrapolation is needed, a constant value will be used outside the fitting range, equal to the polynom value at the boundary. The algorithm warns if the polynomial degree is too high.

One first defines line windows by the command SET WINDOW. Values may be entered numerically as arguments:

```
SET WINDOW wl1 wu1 [wl2 wu2 [...]  ]
```

or graphically with the crosshair cursor if SET CURSOR ON has been selected. In this case enter the values in the same order as above by typing "N" or " " (space bar); "C" cancels the last value entered ; "H" types a help message and "E" terminates the operation.

Up to 20 windows may be defined. BASE then fits a polynomial to the rest of the spectrum. However, only the "visible" parts of the spectrum are used and bad channels are taken out. The degree of the polynomial is defined by SET BASE n, or temporarily by the BASE command itself with its argument. . The /PLOT option plots the fitted baseline in the current box. The area in the windows as well as the rms noise, are computed.

A baseline can be computed for one spectrum, and then subtracted from a different one using BASE LAST. This may be helpful for example at Pico-Veleta where you may remove from the 100 kHz backend the baseline determined from the 1 MHz one. Be sure that you do not change the X-unit between the time you computed the baseline and the time you remove it...

Sinusoidal baselines may also be subtracted, using the command

```
BASE SINUS Amplitude Period Phase
```

where Amplitude Period and Phase are initial guesses for a minimization routine. A linear baseline is added to the sinusoid in any case.

3.1.8 Frequency Switching

Spectra obtained by Frequency Switching need to be folded before further processing. It is usually a good idea to remove a baseline before the spectra are folded in order to use as much baseline as possible. The folding is done by command FOLD which reads from the corresponding section all the necessary parameters. FOLD only operates in the R memory. The number of channels is decreased to keep only the relevant part of the resulting spectrum.

3.1.9 Profile Analysis

The CLASS user may analyse spectra by fitting profiles. The minimization method is taken from the MINUIT system of CERN, modified and optimised for this purpose. Reliability proves to be good. Five types of profiles are presently available, and can be selected by the METHOD command :

- METHOD GAUSS
  This is the default type of profile. One may use up to five gaussians, which might depend on each other as specified by a system of control codes associated with each variable. For each of these gaussians, the primary parameters are 1) Area, 2) Position, and 3) Width (FWHM). The current X unit (for the lower axis) is used. Code 0 means that the parameter is adjustable ; 1 that it is fixed ; 2 that the parameter (head of group) is adjustable and that another parameter, coded 3, is fixed with respect to it ; 4 that the parameter is a fixed head of group.

- `METHOD SHELL`
  Profiles are like those encountered in envelopes of stars. The primary parameters are Area, Position, Width and Horn to Center ratio. The aspect of the profile varies from parabola (as obtain in optically thick lines) for Horn/Center = -1 to flat-topped lines (unresolved optically thin lines) for Horn/Center = 0 and double peaked profiles (resolved optically thin lines) for Horn/Center > 0. The profile is symmetric. Presently only code 0 and 1 can be used, and up to 5 independent lines can be fitted in a single spectrum. The X unit must be frequency.

- `METHOD NH3(1,1) or NH3(2,2) or NH3(3,3)`
  Profiles taking into account hyperfine structure of ammonia with a gaussian distribution of velocity are fitted. Primary variables are 1) The product (Main Group Opacity) times (Excitation Temperature minus Background Temperature) 2) Velocity 3) Line Width (FWHM) and 4) Main Group Opacity . Up to 3 independent lines can be fitted, and only codes 0 and 1 are allowed. The X unit must be Velocity.

- `METHOD HFS FileName`
  This method is similar to the previous one, but the HyperFine Structure parameters are read from a file instead of being known by CLASS . The first line of this file must contain the number of hyperfine components (< 40). The other lines must contain, for each component, the velocity offset and the relative intensity. The parameters are the same as for NH3 method.

- `METHOD CONTINUUM`
  This method is used for continuum drifts. It fits a gaussian and a linear baseline in the drift. If beam-switching was used and the reference beam is along the drift direction, two dependent gaussian are used to optimize signal to noise. The method does not require any user input.

The commands of this profile analysis system are :
`LINES, GAUSS, FIT, DISPLAY, ITERATE, RESIDUAL, KEEP`.

- `LINES N` defines the number of components and prompts for the initial values of the parameters for each component. This command has no effect for method `CONTINUUM`. Parameters are read in list directed format in the following order :

  ```
  Code, Intensity, Code, Position, Code, Width, [Code, Parameter 4]
  ```

  The code is an integer number between 0 and 4. Note that, though the program works on the area (or other quantities as for `NH3` methods), you have to give the intensity, since this quantity is more intuitive than area. The use of the list directed format makes things easier when only one parameter has to be modified (cf Fortran norms). The number of lines N may be zero ; in this case the program finds out reasonable starting values by itself.

  Values may be also entered graphically if `SET CURSOR ON` was selected. After entering `LINES N`, first point the cursor to one side of the line, strike one key, point the cursor the other side, strike another key. The program computes the moment of the spectrum between these boundaries and use it to set up starting values. Proceed like this for all components. One drawback of this way of entering values is that you cannot change the control codes. It should be used only for entirely independent and free lines.

- `GAUSS`
  activates minimization, then prints out the results after convergence. A Simplex method is first used to ensure convergence, then a Gradient method to refine the results, and compute the errors.

- `ITERATE`
  is similar to `GAUSS`, but starts from the previous minimization results. Only the Gradient method is used. Consequently, this command is only useful close to the minimum.

- `FIT N`
  plots the Nth component obtained by fitting ; if N is not given, the sum of all components is plotted.

- `RESIDUAL N`
  subtracts the Nth component from the current spectrum, or the sum of all components is N is not given). In this process, the R spectrum is first copied into T, then the difference is done in R.

- `DISPLAY`
  Prints the results of fitting from the current spectrum, without recomputing it ...

- `KEEP`
  Saves the fit results in the input file, which must be opened also for output. `KEEP` is in fact a reduced version of `UPDATE`, and to be used with the same care as `UPDATE`.

- `SET MASK ...`
  Defines masks in the spectrum for the fit. This commands has the same syntax and behaviour as `SET WINDOW`. Masked regions will not be used for the fit.

Fit results are always saved by a `WRITE` command.

3.1.10 Miscellaneous

- `DIVIDE` makes the ratio of the R and T spectra. The two spectra must have the same velocity scale.

- `FFT` plots the power spectrum of the current observation. It might help identify spurious ripples. Editing of the fourier transform is possible, so that these ripples may be suppressed.

- `NOISE` generates a gaussian noise as intense as in the current spectrum using the rms value determined by the `BASE` command, or using a rms value given as an argument. `NOISE Value NEW` will create a noisy spectrum of given noise level into R, after copying R in T.

- `RESAMPLE` resamples the R spectrum on the specified grid. If the final sampling is coarser than the original one, a smoothing occurs to the final sampling.

- `SMOOTH` operates a Hanning smoothing by default and divide the number of channels by two. Other arguments can be specified to use other methods. `SMOOTH AUTO` uses a sophisticated variable-resolution algorithm, but it requires the channels to be really independent and this is apparently seldom the case in radio astronomy. `SMOOTH GAUSS Width` convolves the spectrum by gaussian of given Width in current units ; it does not take care of bad channels. `SMOOTH BOX N` make the average of N adjacent channels and divides the number of channels by N.

3.2 Continuum Processing

3.2.1 Introduction

So far, we have handled only Spectroscopic data, but Continuum data can be processed by CLASS . Currently, only continuum drifts can be reduced. The basic idea is to treat continuum drifts as spectra would be. Accordingly, very few commands behave differently in Continuum and Spectroscopy modes.

Continuum mode is accessed by typing command
`SET TYPE CONTINUUM`
the prompt changes to `CAS>` (Continuum Analysis System). You can return to Spectroscopy mode later on typing command
`SET TYPE SPECTROSCOPY or LINE`
and the prompt changes to `LAS>` (Line Analysis System).

3.2.2 Differences between Continuum and Spectroscopy Mode

Some commands have slightly different behaviour in Spectroscopy and Continuum modes.

- `SET UNIT` has no effect in Continuum mode.
- `SET ANGLE` also controls the plotting units in Continuum mode.
- `METHOD` : only `GAUSS` and `CONTINUUM` methods are allowed in Continuum mode.
- `LINES` has no effect with `CONTINUUM` method.
- `HEADER` uses a different format for Continuum and Spectroscopy modes.
- `STRIP` produces a map from a set of parallel drifts. The index must define such an ensemble of drifts.
- FITS format support is experimental for continuum data.
- With `CONTINUUM` method, `PRINT FIT` command only outputs a single component, and the component number is not written.
- `SET ALIGN CHANNELS` and `SET ALIGN POSITION` are the only available alignment modes in Continuum mode.

Except for these restrictions, the behaviour of other commands is similar. Note that command `FIND` only selects data of the current type.

3.3 Skydip Processing

CLASS is able to reduce skydip data. Skydip mode must be selected using command
`SET MODE SKYDIP` which also changes the prompt to `SAS>` (Skydip Analysis System). Commands `FIND, HEADER, GET, PLOT` and `WRITE` may be used as for Continuum and Line modes, but the only other valid command is `REDUCE` which fits the sky emission using atmospheric information available in the data, and displays the results.

3.4 On-the-fly Data Processing

The format of CLASS data files has been recently extended to allow the reduction of data obtained in the On-the-fly (OTF) mode by the 30-m telescope. In that mode the data is taken while the telescope beam is moving across the source to be mapped. The raw data is pre-processed (amplitude calibration) by the CAL program available at the telescope, and written in a CLASS file. Data from a scan across the source is written as a single observation, thus containing many spectra at different positions. Each spectrum is included in the observation as a 'record' or 'dump', labeled by time, and telescope position. This new kind of data is handled by CLASS in a way similar to the normal data, but a few commands behave differently:

- `GET` has an option `/RECORD n` to load an individual record of an OTF scan in the R buffer. After that this data may be plotted in the usual way.
- `PLOT` has an option `/SCAN` which displays the entire scan as a 2-dimensional image, with velocity/frequency in the X axis, record number in the Y axis, intensity being rendered as grey/color scale.
- `BASE`, `FFT`, `RESAMPLE` work on all the records of an OTF scan at once.
- `GRID` may be used to tabulate, grid and plot the OTF data into GILDAS data cubes that may be further processed with GILDAS tasks.

3.5 Map Processing: GILDAS

3.5.1 Maps of Spectra

Using the `MAP` command, it is possible to produce a plot of spectra in the current index, arranged in a map. Use the option `/CELL Size_x Size_y` to specify the size of a spectrum, in current angle units. Without this option a default is taken (the actual separation of the spectra). Option `/GRID` will produce frames around the spectra. The argument `MATCH` can be given to fix the aspect ratio of the boxes to the cell sizes.

The map size can be controlled using commands `SET PAGE` and `SET BOX_LOCATION`. Labels can be suppressed by option `/NOLABEL` (and ticks will not be drawn if of size 0.0). Option `/NUMBER` will add the observation number with each spectrum.

On multi-window displays (e.g. X-Window terminals), after the `MAP` command has been used, the `POPUP` command may be used to display in another window a spectrum selected either from its observation number or from its offsets. `POPUP` can also be used after the `STAMP` command. The `STAMP` command allows to display many observations at once, without requesting the X and Y axis scales to be fixed.

3.5.2 GILDAS Interface

For mapping purpose, CLASS is able to produce images at the GILDAS format. GILDAS (the Grenoble Image and Line Data Analysis System) is an ensemble of routines which is able to process images of up to four dimensions. It contains a variety of routines to transpose, resample, reproject, smooth, add, merge, divide, etc... images, and a superset of GREG which allows to produce contour plots from these images. Other programs can also do false color displays of the images on image processors.

3.5.3 Contour maps: Building a data cube

The command `CUBE` builds a 3-D image from the current index. It is assumed that the current index defines such an image (i.e. corresponds to a single line, observed with a single resolution, and towards a single source). The first axis of the cube corresponds to the current X unit, the second to RA (or L) and the third to DEC (or B). Projection information is written so that the coordinate system is automatically recognized when processed by GILDAS .

Please refer to the GILDAS documentation for later processing. The recommended first steps are i) a transposition (task `TRANSPOSE` to put the axes in RA,DEC,Velocity ordering and ii) the tasks `FILL_CUBE` (for undersampled data) or `MAKE_CUBE` (for oversampled data) to resample the data cube on a finer grid for nice contouring.

An alternative processing technique is to use command `GRID`, which creates a table suited for map making using the task `GRID_CUBE`. This route is recommended for oversampled data, while the use of `MAP` and `FILL_CUBE` is probably better for undersampled data.

Finally `GRID /IMAGE` will directly produce data cubes from oversampled or undersampled data, using a Gaussian convolution kernel for interpolation, and reasonable defaults for the cube size and pixel size in both axes.

`CUBE` and `GRID` can also produce images or tables of several mathematical functions of the input spectra rather than simple channel maps. Please refer to the internal help for details.

3.5.4 Velocity-Position Plots

Although by adequate use of transpositions, you can produce velocity-position plots from data cubes, it is also possible to create a 2-D image for processing by GILDAS to produce Velocity-Position plots by using command `STRIP`. This command works on the current index, which must correspond to a real strip. Relevant information is written to a 2-D image which can later be processed by GILDAS . The first axis corresponds to the current X unit (Velocity or Frequency or Channels) and the second axis to the offset.

3.5.5 Continuum Maps

Command `STRIP` can also be used to produce a continuum map from an ensemble of parallel continuum drifts. The index must define such an ensemble.

3.6 The PRINT Command

In addition to the previous commands, the command `PRINT` offers a way to list a number of additional informations. It has several functions :

- `PRINT FIT`, which prints the results of profile fits. For each spectrum, N lines are written (N being the number of fitted components), and each line contains in the following order (1) the component number, (2) then observation number, (3,4) the two cartography offsets, (5,6) area of gaussian and corresponding error, (7,8) same for position, (9,10) same for width, (11) intensity, (12,13) rms on the baseline and on the line. Offset are in the current coordinate system and units. The current method is used.

  For Continuum method, only a single gaussian is written. The written information is oriented towards pointing measurements : (1) the observation number (2,3) Azimuth and elevation (4,5) area of gaussian and error, (6,7) position, (8,9) width, (10) intensity, (11,12) rms on baseline and signal, (13,14) collimations. All angular values are in the current angle unit. The values are followed by the source name.

- `PRINT AREA`, prints the area of the line computed by the `BASE` command. Each line contains (1,2) offsets, (3) area, (4) rms noise.
- `PRINT AREA V1 V2 V3 V4 ...`, prints areas within velocity slices (if such is the current X unit, but one could use channels or frequencies). Ranges are V1-V2, V2-V3, V3-V4, ... etc. Each line will contain (1,2) the offsets, followed by the areas in order.
- `PRINT CHANNEL List`, prints values of channels in the list. The list is specified in the FOR n1 TO n2 BY n3 format. Total number of channels is however limited to 15.
- `PRINT MOMENT V1 V2 V3 V4 ...`, prints moments (area, position, width), of the data within the velocity (or channels or frequencies, depending on the current units) V1-V2, V3-V4, etc... Each line contains (1,2) the offsets, (3,4,5) the moments for V1-V2, (6,7,8) for V3-V4 etc...
- `PRINT POINTING`, prints results of CONTINUUM method fits printed in a format adapted to pointing constants measurements. The output is suited for further processing and determination of pointing constants using the **POINT** program.
- `PRINT FLUX`, prints results of CONTINUUM method fits printed in a format adapted to flux determination. The output is suited for further processing using the `FLUX` language in the **ASTRO** program.

As all commands using an ensemble of spectra, `PRINT` works on the whole current index. Output is by default printed on the screen, but may be directed onto a file by the `/OUTPUT Filename` option.

Alternatively, the same information may be written to a "Table" (a special kind of GILDAS image). The Table format is much faster and suppresses some of the limitations of the formatted output on the number of columns written. Table format is obtained using option `/TABLE Tablename`.

3.6.1 GreG Tables

The command `GREG` is intended to produce a direct interface with GREG for plots of spectra. It creates a GILDAS Table which can be read by GREG . The table contains the following columns for Spectra :

```
1. Intensity
2. Channel number
3. Velocity
4. Offset frequency
5. Rest frequency
6. Image frequency
7. Fitted profiles if any - fit(i),i=0,nline - in column 7+i,
for the current method.
```

The output table can be put later in a formatted way using GILDAS task `LIST` if needed. For continuum data, the table contains

```
1. Intensity
2. Channel number
3. Angular offset (radian)
4. Fitted profile if any.
```

The table may be used as input to GREG to produce fancy plots, or by the GILDAS software for other applications. In particular, the SIC monitor (command `LET`) is able to subtract any of the fits from the spectrum to produce residuals if needed. It is possible to merge different tables, add columns to a table, etc... For example, from two spectra at the same velocity resolution it is possible to merge the two tables and compute the ratio of the spectra, as well as the errors on this ratio.

3.7 Publication Quality Plots

With the hardcopy facility provided by the graphic library used by CLASS , and the `DRAW` command which allows most of the wanted annotations, it is perfectly possible to produce quality publication plots of spectra directly. Note that all GREG commands are available in CLASS to fully annotate plots, superpose spectra with related data, stack various plots ...

3.7.1 Hardcopy

The command `HARDCOPY` is the same as the GREG `HARDCOPY` command. Please refer to GREG documentation or the internal help for a description.

Note that you do not need a graphic terminal to prepare a plot. It is only much more convenient, but the plot and the way it is displayed are two completely independent things.

3.7.2 Annotations

Command `DRAW` can be used to annotate a plot. The basic operations performed by command `DRAW` are :

- `DRAW TEXT Xpos Ypos "Text" Centering`
  to draw a text at position (Xpos,Ypos) (in current units) with the specified centering code. This command works more or less like the GREG command of same name. Please refer to the GREG manual for details. In particular, you can include Greek letters and Symbols in the text using the escape character \. A strange thing may appear on the screen, but it is O.K. on the plot. If you have doubts, use `ZOOM REFRESH` before command `HARDCOPY` to see exactly what your plot looks like.

- `DRAW UPPER Xpos "Text"`
  to draw a vertically oriented text at position Xpos, with a vertical line connecting the beginning of the text to the current spectrum. This text and line are written at position Xpos, in units of the upper axis. Typically, this command is used to mark spectral line identifications.

- `DRAW LOWER Xpos "Text"`
  same as above, but with Xpos in units of the lower axis.

- `DRAW WINDOW [Level]`
  shows the current windows by marks on the graphic plot. Level is an optional arguments indicating at what Y value the marker should be put (Default 0).

- `DRAW MASK [Level]`
  same as above but for the current masks.

- `DRAW KILL [Channel]`
  kills the specified channel (current one if using the cursor) by attributing it the "blanking" or "undefined" value.

- `DRAW FILL [Channel]`
  Fills the specified channel (current one if using the cursor) by interpolation between the nearest non-blanked channels. The channel must have been killed before.

Any other character will not draw anything, but simply returns the cursor position, with corresponding values of the velocity, frequency, image frequency, channel number.

3.7.3 Plot Quality

CLASS is usually used for interactive look at spectra, hence its default values are all oriented towards fast plotting on screen. These defaults can be changed by command `SET`. If the value of a parameter is not controlled by CLASS, the command will be passed on to GREG for processing.

The following GREG presentation parameters are useful :

- `SET BOX_LOCATION`
  It can be set `to LANDSCAPE, PORTRAIT, SQUARE` or 4 numbers indicating the position of the box in the plot page (in centimeters).

- `SET CHARACTER Size`
  Control the size of characters in centimeters.

- `SET FONT Quality`
  Select the character quality to be used, `SIMPLEX` or `DUPLEX`. The fonts are identical to the ones used by GREG , and the character handling is the same (in command `DRAW TEXT`).

- `GREG\SET PLOT ...`
  Define the plot page size. This command is identical to command `SET PLOT_PAGE` in GREG (but in CLASS , `SET PLOT` is already used to indicate whether spectra are broken lines of histograms). Refer to the GREG manual for details.

- `SET TICK Size`
  Define the tick size in centimeters.

3.8 Variables

3.8.1 The Rope to Hang Yourself

CLASS makes use of SIC variables to allow more flexibility in the processing, in particular in procedures. SIC variables are extremely powerful, with the side effect that if you want, you can corrupt your data by overwriting some information. CLASS attempts to prevent the most disastrous errors by defining some of the most critical variables as `READONLY`. They cannot be overwritten by the user, but their values can be used in expressions, either arithmetic or logical. However, an unprotected mode is available for specific processing using the command `SET`VARIABLE.

3.8.2 Index Variables

The variable FOUND refers to the number of observations in the index. It is declared Read-Only of course. Its main use is as a test variable for actions which should be performed only if something exists in the index. As an example, the following procedure reduces a mapping observing run by averaging all spectra at each position, and requires only two user input : the two steps in Lambda and Beta.

```
SAY "Procedure MAP.CLASS : reduce mapping observations by averaging all" -
"spectra for each point. Input required from user : XSTEP and YSTEP" -
"the map spacings in both directions"
!
LAS\FIND
SIC\IF (FOUND.EQ.0) THEN
    SIC\RETURN
SIC\ENDIF
SIC\DEFINE DOUBLE XMIN XMAX YMIN YMAX XSTEP YSTEP ! Local variables
!
! Determine offset extrema.
LAS\GET FIRST
SIC\LET XMIN = OFF_LAMBDA
SIC\LET XMAX = OFF_LAMBDA
SIC\LET YMIN = OFF_BETA
SIC\LET YMAX = OFF_BETA
SIC\FOR I 2 TO FOUND
    LAS\GET NEXT
    SIC\LET XMIN = MIN(XMIN,OFF_LAMBDA)
    SIC\LET XMAX = MAX(XMAX,OFF_LAMBDA)
    SIC\LET YMIN = MIN(YMIN,OFF_BETA)
    SIC\LET YMAX = MAX(YMAX,OFF_BETA)
SIC\NEXT
!
! Now make map. XSTEP and YSTEP must be specified by user
SAY "Enter step in Lambda"
LET XSTEP =
SAY "Enter step in Beta"
LET YSTEP =
SIC\FOR XOFF XMIN TO XMAX BY XSTEP
    SIC\FOR YOFF YMIN TO YMAX BY YSTEP
        LAS\FIND/OFFSET XOFF YOFF
        SIC\IF (FOUND.GT.0) THEN
            LAS\SUM
            LAS\WRITE
        SIC\ENDIF
    SIC\NEXT
SIC\NEXT
SIC\RETURN
```

The FIND command does not return an error, but set FOUND = 0, if it finds nothing. A second variable related to the index is the INDEX array, of dimension FOUND, which contains the observation numbers of all observations in the index.

3.8.3 Default Header Variables

The most important header parameters are defined by default as SIC variables in a protected mode; the others, of less frequent use, can be accessed if required by the user (see "Advanced Processing"). The default variables are (RW means Read-Write variable, RO, Read-Only).

```
TELESCOPE       Character*12, RW, Telescope name
NUMBER          Integer,      RW, Observation number
VERSION         Integer,      RO, Version number
DATATYPE        Integer,      RO, Type of observation
                                  0 Line, 1 Continuum, 2 Skydip
QUALITY         Integer,      RO, Quality of data
SCAN            Integer,      RO, Original scan number
UTOBS           Double,       RO, UT of observation (Radians)
LSTOBS          Double,       RO, LST of observation (Radians)
AZIMUTH         Real,         RW, Azimuth of observation (Radians)
ELEVATION       Real,         RW, Elevation of observation (Radians)
TSYS            Real,         RW, System temperature
TIME            Real,         RW, Integration time (Seconds)

SOURCE          Character*12, RW, Source name
LAMBDA          Double,       RW, Longitude of source (Radians)
BETA            Double,       RW, Latitude of source (Radians)
OFF_LAMBDA      Double,       RW, Offset in longitude (Radians)
OFF_BETA        Double,       RW, Offset in latitude (Radians)
EPOCH           Real,         RW, Epoch of coordinates (Years)

LINE            Character*12  RW, Line name
CHANNELS        Integer,      RO, Number of channels
REFERENCE       Real,         RW, Reference channel
FREQ_STEP       Real,         RW, Frequency step by channel (MHz)
VELO_STEP       Real,         RW, Velocity step by channel (km/s)
VELOCITY        Real,         RW, Velocity of reference channel
FREQUENCY       Double,       RW, Rest frequency at reference channel
IMAGE           Double,       RW, Image frequency "   "   "   "

BEAM_EFF        Real,         RW, Telescope beam efficiency
FORWARD_EFF     Real,         RW, Telescope forward efficiency
GAIN_IMAGE      Real,         RW, Image to signal band ratio
WATER           Real,         RO, Water vapor content (mm)
PRESSURE        Real,         RO, External pressure (hPa)
AMBIENT_T       Real,         RO, External temperature (K)
CHOPPER_T       Real,         RO, Chopper temperature (K)
COLD_T          Real,         RO, Cold load temperature (K)
```

```
TAU_SIGNAL      Real,          RO, Opacity in signal band
TAU_IMAGE       Real,          RO, Opacity in image band
ATM_SIGNAL      Real,          RO, Atmospheric temperature
                                   in signal band
ATM_IMAGE       Real,          RO, Atmospheric temperature
                                   in image band

RX              Real[8192]     RO, X values of data points
RY              Real[8192]     RW, Y values of data points
```

8192 is currently the maximum size of the spectra, but the variables RX and RY are redimensioned to the effective number of channels for each spectrum.

3.8.4 Advanced Processing

All header parameters can be defined as SIC variables for specific processing of the data, either as Read-Only or as Read-Write, using the command `SET VARIABLE`. Read-Write mode is to be used with caution, since even critical variables (e.g. the number of channels) can be modified. Refer to command `SET`VARIABLE for more details.

By using the appropriate variables and the SIC mathematical and logical facilities, customized data processing becomes possible, as well as complete data editing.

4 From CLASS to FITS

To bring back CLASS spectra to your home institution, where the CLASS format may not be supported, CLASS offers the possibility of writing standard FITS format files. CLASS to FITS conversion (and vice-versa) is done by command `FITS`. In addition, all functionalities provided by the SIC command `DEFINE FITS` are of course available. For a description of the FITS format see the original paper by Wells et al. (Astron. and Astrophys. Suppl.).

The CLASS `FITS` command has the following syntax:

```
FITS READ Filename[.fits]
```

to read a FITS file and create CLASS data from it, or

```
FITS WRITE Filename[.fits] [/BITS Nbits] [/MODE SPECTRUM|INDEX]
```

to write a FITS file from CLASS data.

In addition, default values can be supplied by the `SET FITS` command.

```
SET FITS BITS Nbits
SET FITS MODE Spectrum|Index|None
```

4.1 CLASS FITS format

4.1.1 Simple SPECTRUM mode

FITS headers written by CLASS depend on the informations present in the corresponding CLASS headers. Any missing information will also be omitted in FITS (and vice versa). A typical FITS header written by CLASS looks like this :

```
SIMPLE  =                    T         /                                    
BITPIX  =                   16         /                                    
NAXIS   =                    4         /                              (1)   
NAXIS1  =                  253         /                              (2)   
NAXIS2  =                    1         /                              (3)   
NAXIS3  =                    1         /                              (3)   
NAXIS4  =                    1         /                              (3)   
BSCALE  =  0.1038147092913E-03         /                                    
BZERO   = -0.2413805246353E+01         /                                    
DATAMIN = -0.5815605640411E+01         /                                    
DATAMAX =  0.9877878427505E+00         /                                    
BUNIT   = 'K       '                   /                              (4)   
CTYPE1  = 'FREQ    '                   /                              (5)   
CRVAL1  =  0.0000000000000E+00         / Offset frequency                   
CDELT1  =  0.1000000014901E+06         / Frequency resolution               
CRPIX1  =  0.1345000000000E+03         /                                    
CTYPE2  = 'RA      '                   /                              (6)   
CRVAL2  =  0.8388750229169E+02         /                                    
CDELT2  = -0.5555555975722E-02         /                                    
CRPIX2  =  0.0000000000000E+00         /                                    
CTYPE3  = 'DEC     '                   /                              (6)   
CRVAL3  = -0.1777777752148E+01         /                                    
```

```
CDELT3  =  0.0000000000000E+00         /
CRPIX3  =  0.0000000000000E+00         /
CTYPE4  = 'STOKES  '                   /                              (7)
CRVAL4  =      1.0000000000000         /
CDELT4  =      0.0000000000000         /
CRPIX4  =      0.0000000000000         /
TELESCOP= 'IRAM-30M-B20'               /
OBJECT  = 'ORI-I-2     '               /
GLAT    =  0.0000000000000E+00         / Galactic latitude          (8)
GLON    =  0.0000000000000E+00         / Galactic longitude         (8)
EPOCH   =  0.1950000000000E+04         /                            (9)
BLANK   =  0.9878914356232E+00         / Blanking value
LINE    = '*           '               / Line name                  (10)
RESTFREQ=  0.1152712040000E+12         / Rest frequency             (11)
VLSR    =  0.1300000000000E+05         / Velocity of ref. channel   (12)
DELTAV  = -0.2600757479668E+03         / Velocity resolution        (13)
IMAGFREQ=  0.1074062118530E+12         / Image frequency            (14)
TSYS    =  0.4787839660645E+03         / System temperature         (15)
OBSTIME =  0.7500000000000E+02         / Integration time           (16)
SCAN-NUM=  0.4386000000000E+04         / Scan number                (17)
TAU-ATM =  0.8740132451057E+00         / Atmospheric opacity        (18)
NPHASE  =                    2         / Number of frequency phases (19)
DELTAF1 = -0.5000000000000E+07         / Frequency offset Phase 1   (20)
PTIME1  =  0.3750000000000E+02         / Duration of Phase 1        (20)
WEIGHT1 =  0.1000000000000E+01         / Weight of Phase 1          (20)
DELTAF2 =  0.5000000000000E+07         / Frequency offset Phase 2   (20)
PTIME2  =  0.3750000000000E+02         / Duration of Phase 2        (20)
WEIGHT2 = -0.1000000000000E+01         / Weight of Phase 2          (20)
BEAMEFF =  0.56                        / Beam efficiency            (21)
FORWEFF =  0.88                        / Forward efficiency         (22)
GAINIMAG=  1.0000000000000E+00         / Image sideband gain ratio  (23)
ORIGIN  = 'LAS-Grenoble-VAX'           /
DATE    = ' 7/ 9/85'                   / Date written
DATE-OBS= '29/ 5/85'                   / Date observed
DATE-RED= ' 7/ 9/85'                   / Date reduced
ELEVATIO=  0.5064780612975E+02         /  Telescope elevation     (24)
AZIMUTH =  0.1919660046612E+03         /  Telescope azimuth
UT      = '12:50:47.384'               / Universal time at start
LST     = '06:09:00.479'               / Sidereal time at start of observation
HISTORY REL  0.5064780612975E+02       /  Telescope elevation     (24)
HISTORY RAZ  0.1919660046612E+03       /  Telescope azimuth
HISTORY RUT  12:50:47.384    Universal time at start of observation
HISTORY RST   6:09:00.479    Sidereal time at start of observation
HISTORY SCAN LIST 4383-4386                                        (25)
END
```

1. Although only one axis is really necessary, it is very convenient to define four, use the first one for the channels, and the three last ones to code the positions and stokes parameters.

2. The first axis is used to define effectively the spectrum. Thus `NAXIS1` is the number of channels.

3. `NAXIS2, NAXIS3,` and `NAXIS4` are all one for a single spectrum. Note however that it is possible to store a raster map with a similar header as this one.

4. Could be Janskys

5. First axis defined in terms of frequency (in the signal sideband in case of double sideband operations). The frequency of a specific channel is given by
   `F(i) = RESTFREQ + CRVAL1 + ( i - CRPIX1 ) * CDELT1`
   in which the Rest frequency `RESTFREQ` is defined later in the header.

6. Second axis, Right Ascension RA (as in this case) or Galactic Longitude GLON. The information as presented here is slightly incomplete, since it would be in general necessary to have an information about the kind of projection used. On most radio telescopes, it is simply assumed that the angular offset in RA is divided by the cosine of Declination to represent "true" angular offsets (valid only for a small field). Small telescopes may need more elaborate projection systems. In the current example, the position really observed is
   `Dec = CRVAL3 + ( 1 - CRPIX3 ) * CDELT3`
   `Ra = CRVAL2 + ( 1 - CRPIX2 ) * CDELT2 / COS(Dec)`
   That is, `CDELT2` and `CDELT3` represents angular offsets from the reference position `(CRVAL2,CRVAL3)` in a Global Sinusoidal projection (`RADIO` projection).

7. Stokes parameters as defined in the basic paper of Wells et al.

8. Galactic latitude and longitude of the *reference* position, *i.e.* of the position `(CRVAL2,CRVAL3)`. If one was using galactic coordinates instead of equatorial ones, the RA and DEC would appear here instead.

9. Epoch of these coordinates

10. Molecular line name, for bookkeeping

11. Rest frequency

12. LSR Velocity of the reference channel. Heliocentric velocities can be used also.

13. Velocity spacings of the channels. This information is duplicate with the rest frequency and frequency spacing of channels, but convenient. The velocity of a given channel is thus given by
    `V(i) = VLSR + ( i - CRPIX1 ) * DELTAV`

14. Image frequency, for double sideband operation.

15. System temperature, necessary for some weighting when adding a number of spectra.

16. Integration time, used for the same reason as above.

17. Scan number, for bookkeeping.

18. Atmospheric opacity in the signal sideband.

19. For multi-phased spectra (i.e. frequency switching) number of phases.

20. For each phase, the frequency offset, the phase length and weight.
21. The telescope beam efficiency
22. The telescope forward efficiency
23. The ratio of gains in the image and signal sidebands (in case of double sideband operation)
24. Some "History" comments. Whether this information should be given with specific keywords or in an History record is still an open question. This information is not really needed for further data reduction, but it helps bookkeeping.
25. The list of scan numbers of the spectra added to produce this one.

The FITS interface for Continuum data is still experimental. Try it, and send your comments...

4.1.2 BINTABLE version

In addition to the simple "one FITS file per Spectrum" mode, CLASS supports the Binary Table extension, in reading and writing. The FITS keywords for the Binary Table format are similar to those of the simple Spectrum mode, but a complete set of spectra can be handled in a single binary table. Keywords which would change from one spectrum to the next are handled as Binary columns of the table, with the `TTYPEi` FITS keyword holding the name of the corresponding simple FITS keyword. The spectra are handled in a `MATRIX` storage.

Reading such FITS files, CLASS will support any value for the `TTYPEi` keywords.

However, when creating such FITS files, CLASS (currently) makes two assumptions:

- The whole current index is to be written as a single BINTABLE
- This index represents a 2-D map of spectra

Accordingly, the index must be consistent: a single line, a single frequency setup... Two columns are written with the spectra: `CDELT2` and `CDELT3`, which hold the map offsets. Any other CLASS index can result in an inconsistent BINTABLE **sometimes with no warning at al!**.

4.1.3 Once FITS always FITS

This is the basic principle of FITS, but the IAU FITS committee keeps making revisions of the FITS format. So things are expected to change, with new versions of CLASS providing new FITS output, but still reading old FITS.

4.2 From FITS to CLASS

```
FITS READ Filename[.fits]
```

will read a FITS file and create CLASS data from it. It is expected to work under the following conditions:

1. The `Filename.fits` file contains one spectrum, with (a subset of) the FITS keywords which are described in the previous section. FITS Keyword redefinition is possible.
2. **OR** The `Filename.fits` contains a BINTABLE, also with recognized FITS keywords.

3. **No more, no less**

This may look akwardly restrictive, but is already powerful if you have thought about your data destination when creating the FITS file.

Moreover, CLASS offers a very general way to read **any** FITS file and create a CLASS data file from it. This can be done using the `DEFINE FITS` command of SIC, combined with the `MODEL` command of CLASS. An example of a simple CLASS procedure to do so is given below

```
! Usage: @ fits_to_class Fits_DataCube_file_name
define fits a &1
sic\let num 0 /new integer
set priv on
!
sic\for m 1 to a%dim[3]
  sic\for l 1 to a%dim[2]
    let num = num+1             ! Load the data array
    if (a%ndim.eq.3) then
      analyse\model a[l,m]
    else if (a%ndim.eq.4.and.a%dim[4].eq.1) then
      analyse\model a[l,m,1]
    else
      say "Cube is not a cube. Please check dimensions in header"
      set priv off
      pause
    endif
! Produce a meaningful header
    let lambda = a%convert[2,2]
    let off_lambda = (l-a%convert[1,2])*a%convert[3,2]
    let beta = a%convert[2,3]
    let off_beta = (m-a%convert[1,3])*a%convert[3,3]
    let source = a%source
    let line = a%line
! Assume a velocity axis, would have to be changed for a frequency axis
    let velocity = a%convert[2,1]
    let velo_step = a%convert[3,1]
    let reference = a%convert[1,1]    ! reference pixel of velocity axis
    let frequency = a%restfre   ! Presumably undefined for a foreign cube ...
    let freq_step = a%freqres   !
    las\modify teles "Edit_procedure"
    las\write num
  sic\next
sic\next
set priv off
```

4.2.1 FITS Keyword redefinition

A minimal number of keywords has been defined as part of the FITS standard, but additional ones can be (and have been) added by various groups to support their own needs. Thus, several "flavors" of FITS coexist. Unknown keywords are normally ignored, but CLASS supports

FITS keyword redefinition. If you receive a file with scan number coded as `NUMBER` (instead of `SCAN-NUM`), all you need to do is to define a SIC symbol named `NUMBER` with translation `SCAN-NUM`. This is done by typing `SIC\SYMBOL NUMBER SCAN-NUM`.

4.3 Writing FITS files

To write a FITS file from CLASS data, use the following command:

```
FITS WRITE Filename[.fits] [/BITS Nbits] [/MODE SPECTRUM|INDEX]
```

The command will create a simple FITS file from the current Spectrum (in `SPECTRUM` mode), or a FITS BINTABLE from the current Index (in `INDEX` mode). The number of bits can be controlled. Default values for the mode and the number of bits can be supplied by the `SET FITS` command.

```
SET FITS BITS Nbits
SET FITS MODE Spectrum|Index|None
```

In `INDEX` mode, it is up to the user to make sure that the index is consistent (same number of channels, etc..., for all spectra in index).

4.4 What happened to GFITS ?

(gee, you didn't look that old !)

Well, it disappeared. GFITS was mostly intended to work with tapes, which no longer exists nowadays. The new `FITS` command is simpler to use. The FITS format supported by CLASS has not changed, though.

5 Internal CLASS format

This section comes from a working document written in 1997. It will be updated at the same time as the whole CLASS documentation (Hopefully soon).

5.1 Contents of one observation

Data, i.e. observational parameters as well as spectra, is organized in the following way:

- one observation is self-contained. All the information needed to reduce it is recorded on the same few disk blocks. It may be one spectral scan, one continuum drift scan, a snapshot obtained with an array receiver, or one on-the-fly scan obtained with a single beam or an array receiver.
- each observation is divided in several sections, containing header parameters or data:
  - General information (date, times, local coordinates, sequence number, ...)
  - positional information (source, name, astronomical coordinates, epoch, offsets, ...)
  - spectral information (number of spectra, line names ans frequencies, resolutions, number of channels, ...)
  - description of the observing procedure
  - data
  - ...

  Each type of section is known by the system by means of an integer code. It is not required that all sectionsw should be present; some are written by the observing program at the time of the observations ; some may be dropped during the reduction, being replaced by new ones describing the results of the different staged of the reduction procedure.
- each observation begins with the following words:

length (bytes)	content
4	'2' (identification code)
4	number of logical blocks
4	number of byte describing the observation
4	address of the data section (bytes)
4	length of the data section (bytes)
4	the number of header sections
4	the index entry of the observation
4*nsec	the codes of the sections
4*nsec	the addresses of the sections
4*nsec	the lengths of the sections

  Thus the first section will e.g. be stored at the address (28+12*nsec), followed by the second section, and so on the the end of the observation. The length of one particular type of section is not fixed; adding extra parameters is then possible in each section.

5.2 File organization

Data is stored in a direct access file; the record length is 512 bytes. Blocks may contain either ovservationbs or indexing information. Each observation begins on a logical block limit. The first block of the file contains a file description in the following format:

length (bytes)	content
4	file identification code
4	next block available for writing on the file.
4	number of index entries per index extension.
4	nex: number of index extensions (max. 251)
4*nex	first block number of each index extension

The list of index extension block numbers extends onto the second block of the file.

The file identification code is a 4 character code used to indicate the internal storage format of the machine which has written the data ('1 '=Vax,;'1A '=IEEE, '1B '='EEEI'); this enable the reading machine to convert the data to its internal format.

Each file contains its own index. Each index entry has a length of 32 words; thus 4 index entries fill a 512-byte block. Several (by default 64) index entries are grouped in an index extension, of by default 16 consecutive blocks. The addresses of the first blocks of each extension are stored in the first block. During writing, observations are written at the end of the file, and indexed in the current index extension ; when the latter is full, a new index extension is created at the end of the file.

The contents of each index entry may depend on the application. for CLASS spectral line and continuum observations, it contains the following:

name	length (words)	content
XBLOC	1	observation block number
XNUM	1	observation sequence number
XVER	1	observation version number
XSOURC	3	source name (12 char)
XLINE	3	telescope name (12 char)
XDOBS	1	observation date
XDRED	1	reduction date
XOFF1	1	first offset (real, radians)
XOFF2	1	second offset (real, radians)
XTYPE	1	coordinate system ('EQ", 'GA', 'HO')
XKIND	1	Kind of observation (0: spectral, 1: continuum,)
XQUAL	1	Quality (0-9)
XSCAN	1	Scan number
XPOSA	1	Position Angle
XFRONT	2	(8 char) Front-end ID (PROPOSED)
XBACK	2	(8 char) Back-end ID (PROPOSED)
XPROC	2	(8 char) Procedure ID (PROPOSED)
XPROJ	2	(8 char) Project ID (PROPOSED)
(unused)	6	for future extensions

5.3 Examples of Header Sections

Here we describe the contents of the main header sections (excerpts from the storage declarations in the CLASS program itself). Note that, in the data file, the actual length of some of the sections is variable, e.g. the length of the switching information section depends on the number of different phases in the switching procedure.

5.3.1 General Parameters

```
INTEGER MGEN
INTEGER MGEN2
PARAMETER (MGEN=9,MGEN2=11)
INTEGER  RNUM                  ! Observation number
INTEGER  RVER                  ! Version number
INTEGER  RTELES(3)             ! Telescope name
INTEGER  RDOBS                 ! Date of observation
INTEGER  RDRED                 ! Date of reduction
INTEGER  RTYPEC                ! Type of coordinates
INTEGER  RKIND                 ! Type of data
INTEGER  RQUAL                 ! Quality of data
INTEGER  RSCAN                 ! Scan number
REAL*8   RUT                   ! UT of observation
REAL*8   RST                   ! LST of observation
REAL*4   RAZ                   ! Azimuth
REAL*4   REL                   ! Elevation
REAL*4   RTAU                  ! Opacity
REAL*4   RTSYS                 ! System temperature
REAL*4   RTIME                 ! Integration time
```

5.3.2 Position information

```
INTEGER MPOS
PARAMETER (MPOS=11)
INTEGER  RSOURC(3)             ! Source name
REAL*4   REPOCH                ! Epoch of coordinates
REAL*8   RLAM                  ! Lambda
REAL*8   RBET                  ! Beta
REAL*4   RLAMOF                ! Offset in Lambda
REAL*4   RBETOF                ! Offset in Beta
INTEGER  RPROJ                 ! Projection system
                               ! optional addition: *PROPOSED*
INTEGER RDESCR                 ! decript. system code (0=no,1='D',2='D *')
                               ! (D means def. by origin, D * by pole)
REAL*8   RSLOP                 ! descriptive system longitude
REAL*8   RSBOP                 ! descriptive system latitude
REAL*8   RSKOP                 ! descriptive system pos.angle
```

5.3.3 Spectroscopic information (length 18)

```
INTEGER MSPEC
PARAMETER (MSPEC=18)
INTEGER  RLINE(3)             ! Line name
REAL*8  RRESTF                ! Rest frequency
INTEGER  RNCHAN               ! Number of channels
REAL*4  RRCHAN                ! Reference channels
REAL*4  RFRES                 ! Frequency resolution
REAL*4  RFOFF                 ! Frequency offset
REAL*4  RVRES                 ! Velocity resolution
REAL*4  RVOFF                 ! Velocity at reference channel
REAL*4  RBAD                  ! Blanking value
REAL*8  RIMAGE                ! Image frequency
INTEGER  RVTYPE               ! Type of velocity
REAL*8  RSKYFR                ! Sky frequency
REAL*4  RVTELES               ! Vel. of telescope relative to ref.
```

5.3.4 Baseline information (for spectra of drifts)

```
INTEGER MWIND
INTEGER MBASE
PARAMETER (MWIND=20,MBASE=4+2*MWIND)
INTEGER  RDEG                 ! Degree of last baseline
REAL*4  RSIGFI                ! Sigma
REAL*4  RAIRE                 ! Area under windows
INTEGER  RNWIND               ! Number of line windows
REAL*4  RW1(MWIND)            ! Lower limits of windows
REAL*4  RW2(MWIND)            ! Upper limits of windows
```

5.3.5 Scan numbers of initial observations

```
INTEGER MSEQ
INTEGER MORIG
PARAMETER (MSEQ=100,MORIG=2*MSEQ+1)
INTEGER  RNSEQ                ! Number of sequences
INTEGER  RSTART(MSEQ)         ! Start can number of seq.
INTEGER  REND(MSEQ)           ! End scan number of seq.
```

5.3.6 Default plotting limits.

```
INTEGER MPLOT
PARAMETER (MPLOT=4)
REAL*4  RAMIN                 ! Min Y value plotted
REAL*4  RAMAX                 ! Max Y value plotted
REAL*4  RVMIN                 ! Min X value plotted
REAL*4  RVMAX                 ! Max X value plotted
```

5.3.7 Switching information (for spectra)

```
INTEGER MXPHAS
INTEGER MFSW
PARAMETER (MXPHAS=8,MFSW=2+6*MXPHAS)
INTEGER  RNPHAS               ! Number of phases
REAL*8  RDECAL(MXPHAS)        ! Frequency offsets
REAL*4  RDUREE(MXPHAS)        ! Time per phase
REAL*4  RPOIDS(MXPHAS)        ! Weight of each phase
INTEGER  RSWMOD         ! Switching mode (1:frequency, 2:position...)
REAL*4  RLDECAL(MXPHAS)       ! Lambda offsets
REAL*4  RBDECAL(MXPHAS)       ! Beta offsets of each phase
INTEGER RTYPED          ! type of offsets (EQ, BA) *PROPOSED*
```

5.3.8 Calibration parameters

```
INTEGER MCALIB
PARAMETER (MCALIB=21)
REAL*4  RBEEFF                ! Beam efficiency
REAL*4  RFOEFF                ! Forward efficiency
REAL*4  RGAINI                ! Image/Signal gain ratio
REAL*4  RH2OMM                ! MM of water vapor
REAL*4  RPAMB                 ! Ambient pressure (hPa)
REAL*4  RTAMB                 ! Ambient temperature (K)
REAL*4  RTATMS                ! Atmosphere temp. signal band
REAL*4  RTCHOP                ! Chopper temperature
REAL*4  RTCOLD                ! Cold load temperature
REAL*4  RTAUS                 ! Opacity signal band
REAL*4  RTAUI                 ! Opacity image band
REAL*4  RTATMI                ! Atmosphere temp. image band
REAL*4  RTREC                 ! Receiver temperature
INTEGER  RCMODE               ! Calibration mode
REAL*4  RATFAC                ! Applied calibration factor
REAL*4  RALTI                 ! Site elevation
REAL*4  RCOUNT(3)             ! Power of Atm., Chopp., Cold
REAL*4  RLCALOF               ! Longitude offset for sky measurement
REAL*4  RBCALOF               ! Latitude offset for sky measurement
REAL*4  RJYKEL                ! Jy/Kelvin Conversion factor. *PROPOSED*
```

5.3.9 For Skydips observations. No associated data.

```
INTEGER MSKY
INTEGER MSKYDIP
PARAMETER (MSKY=10,MSKYDIP=10+4*MSKY)
INTEGER  RSLINE(3)            ! Line name
REAL*8  RSREST                ! Rest frequency
REAL*8  RSIMAG                ! Image frequency
INTEGER  RNSKY                ! Number of points on sky
```

```
INTEGER  RNCHOP              !    -   -    -    - chopper
INTEGER  RNCOLD              !    -   -    -    - cold load
REAL*4   RELEV(MSKY)         ! Elevations
REAL*4   REMISS(MSKY)        ! Power on sky
REAL*4   RCHOPP(MSKY)        ! Power on chopper
REAL*4   RCOLD(MSKY)         ! Power on cold load
```

5.3.10 Gauss fit results (for spectra or drifts)

```
INTEGER MXGAUS
INTEGER MFIT
PARAMETER (MXGAUS=5,MFIT=3+6*MXGAUS)
INTEGER  RNLINE              ! Number of components
REAL*4   RSIGBA              ! Sigma on base
REAL*4   RSIGRA              ! Sigma on line
REAL*4   RNFIT(3*MXGAUS)     ! Fit results
REAL*4   RNERR(3*MXGAUS)     ! Errors
```

5.3.11 Continuum drift description (for drifts)

```
INTEGER MCONT
PARAMETER (MCONT=16)
REAL*8   RFREQ               ! Rest frequency
REAL*4   RWIDTH              ! Bandwidth
INTEGER  RNPOIN              ! Number of data points
REAL*4   RRPOIN              ! Reference point
REAL*4   RTREF               ! Time at reference
REAL*4   RAREF               ! Angular offset at ref.
REAL*4   RAPOS               ! Position angle of drift
REAL*4   RTRES               ! Time resolution
REAL*4   RARES               ! Angular resolution
REAL*4   RCBAD               ! Blanking value
INTEGER  RCTYPE              ! Type of offsets
REAL*8   RCIMAG              ! Image frequency
REAL*4   RCOLLA              ! Collimation error Az
REAL*4   RCOLLE              ! Collimation error El
```

5.3.12 Beam-switching parameters (for spectra or drifts)

```
INTEGER MBEAM
PARAMETER (MBEAM=5)
REAL*4   RCAZIM              ! Azimuth of observation
REAL*4   RCELEV              ! Elevation of observation
REAL*4   RSPACE              ! Beam spacing
REAL*4   RBPOS               ! Position angle of beams
INTEGER  RBTYPE              ! System for angle
```

5.3.13 Double gaussian and baseline fit results (for drifts)

```
      INTEGER MFCONT
      PARAMETER (MFCONT=19)
      INTEGER  RLCONT               ! Number of components
      REAL*4  RBCONT                ! Sigma on base
      REAL*4  RRCONT                ! Sigma on line
      REAL*4  RNCONT(8)             ! Results
      REAL*4  RECONT(8)             ! Errors
```

5.3.14 Multi-beam receiver

(Section contents to be finalized dec 1997 - jan 1998 *PROPOSED*).

```
      INTEGER MPIX
      PARAMETER (MPIX=64)
      INTEGER R_NPIX                ! number of pixels
      integer R_TYPEPIX             ! system code for pixel offsets
      REAL*4 R_TREC_PIX(MPIX)       ! Receiver temperature
      REAL*4 R_BEEFF_PIX(MPIX)      ! Beam efficiency
      REAL*4 R_FOEFF_PIX(MPIX)      ! Forward efficiency
      REAL*4 R_GAINI_PIX(MPIX)      ! Image/Signal gain ratio
      REAL*4 R_TSYS_PIX(MPIX)       ! System temperature
      REAL*4 R_COUNT_PIX(MPIX,3)    ! Power of Atm., Chopp., Cold
      REAL*4 R_TCHOP_PIX(MPIX)      ! Chopper temperature
      REAL*4 R_TCOLD_PIX(MPIX)      ! Cold load temperature
      REAL*4 R_LAM_PIX(MPIX)        ! Lambda offset
      REAL*4 R_BET_PIX(MPIX)        ! Beta offset
```

5.3.15 On-the-fly section receiver

(Section contents to be finalized jan 1998 *PROPOSED*).

```
      real*4 r_PA_OTF               ! position angle (E from N, rad., in basis)
      real*4 r_VEL_OTF              ! scanning speed (rad/sec)/
      real*4 r_integ_OTF            ! dump duration (aimed, sec.)
```

5.3.16 Pointing constants

(Section contents to be finalized jan 1998 *PROPOSED*).

```
      real*4 r_nule
      real*4 r_nula
      real*4 r_colstar
! more  needed ?
```

5.3.17 Data Section Descriptor

```
      INTEGER MDESCR
      PARAMETER (MDESCR=4)
      INTEGER R_NDUMP               ! Number of records
```

```
INTEGER R_LDPAR               ! Length of data header (longwords)
INTEGER R_LDATL               ! length of line data      (")
INTEGER R_LDUMP               ! length of record         (")
```

5.4 The Data Section

5.4.1 Single spectrum Data Section

A single spectrum data section contains the spectrum intensity of each spectral channel. It thus consists of NDATA real values.

5.4.2 Multiple spectra Data Section

A multiple spectra data section contains many spectra, obtained in a single observing scan: typically a drift scan across a source, a on-off scan with a multi-beam receiver, or a drift scan with a multi-beam receiver. The data section is considered as a multiple spectra data section is the special section 'Data Section Descriptor' is present. It consists of a `R_NDUMP` records (individual spectra); each spectrum has `R_LDATL` line channels, and is followed by a few data associated parameters (`R_LDPAR` words). The length of each record is thus `R_LDUMP=R_LDPAR+R_LDATL`; the total length of the data section should be: `NDATA=R_NDUMP*R_LDUMP`.

The data header presently contains:

offset	type	name	
0	Integer*4		Record number
1	Real*4	RAZ	Telescope Azimuth (radians)
2	Real*4	REL	Telescope Elevation (radians)
3	Real*8	RST	Sideral Time (radians)
5	Real*4	RLAMOF	Azimuth offset (radians)
6	Real*4	RBETOF	Elevation offset (radians)
7	Integer*4	R_IPIX	Pixel number (multibeam) *PROPOSED*
8	Real*4	R_INTEG	actual dump integration time (sec.) *PROPOSED*
9	Real*4	R_DOPEARTH	Vel Telesc/earth center *PROPOSED*
10	Real*4	R_DOPSUN	Vel Telesc/Sun *PROPOSED*
11	Real*4	R_DOPLSR	Vel Telesc/LSR *PROPOSED*
12	Real*4	RUT	Universal Time (seconds) in day RDOBS

The data header contents are in the process to be finalized (dec-1997).

6 LAS Language Internal Help

6.1 ACCUMULATE

```
LAS\ACCUMULATE
```

ACCUMULATE is used to integrate step by step an ensemble of spectra. The T spectrum is added to the spectrum in R with the current weights. R thus contains the current sum, and T the last sum. As command SUM, ACCUMULATE checks for the positional coincidence and calibration homogeneity according to SET MATCH and SET CALIBRATION respectively. Alignement of spectra is checked according to SET ALIGN command.

For EQUAL weights, ACCUMULATE computes the SUM of the two spectra, allowing addition of an ensemble of spectra after division by the total number of spectra.

ACCUMULATE also works on Continuum drift. The alignement may be Channel or Position in this case.

6.2 BASE

```
LAS\BASE [Deg] [/PLOT] [/POLYGON [filename]] [/CONTINUUM [flux]]
LAS\BASE SINUS Amplitude Period Phase [/PLOT]
LAS\BASE LAST [/PLOT]
```

BASE copies R into T, then subtracts from R a polynomial baseline of degree Deg, or a sinusoidal baseline if the first argument is SINUS. In the latter case, SINUS must be followed by the Amplitude, Period and Phase (in the same unit as the period, usually km/s or MHz) of a trial sinusoid. Minimization will be done including also a linear baseline in addition to the sinusoid. Zones defined by the SET WINDOW command are not used to fit the baseline.

BASE LAST will use the last determined baseline instead of computing a new one. It can be useful to find a baseline from one backend and apply it to another. This also applies for sinusoidal baselines.

/PLOT plots this baseline after the minimization.

BASE /CONTINUUM will divide the spectrum by the baseline rather than substracting it. This is useful to keep the continuum level, when continuum is present and variations of atmospheric emission are properly substracted out by fast enough switching. This is of course the case for correlation spectra (e.g. coming from an interferometer). The continuum level will be adjusted to the value of the argument 'flux', if present; e.g. /CONTINUUM 1.0 will produce a spectrum of line-to-continuum ratio. BASE /CONTINUUM does not work with BASE LAST or BASE SINUS.

Case of On-The-Fly scans:
BASE subtracts a baseline to each record of the OTF-scan
/POLYGON [filename]: Allows you to define the spectral window as a 2-D polygon. If a filename is given, the polygon is taken from it. Otherwise it is defined interactively. A polygon can be stored in a file with the GREG2\WRITE command.

6.3 BOX

LAS\BOX [Arg1] [Arg2] [Arg3] [Arg4] [/UNIT Type [UPPER]]

BOX draws a frame for the spectra; the units and limits are determined according to the options of the SET command (SET MODE ; SET UNIT), and the box size governed by SET BOX.

The first two arguments are used to modify the labelling of the lower and left axes respectively. They may take the values P for parallel labels (default for X axis), O for Orthogonal labels (default for Y axis) or N for No labels. A third argument is available to indicate that the ticks are to be In or Out of the box. The fourth argument, if present indicates for which axis labels should be plotted. Possible values are X, Y and None.

If the option /UNIT is present, it modifies (with no permanent action) the units of the X axis, or only of the upper X axis if UPPER is specified. Type may be also IMAGE to get image-side-band frequency units. For Continuum data, the /UNIT command has no effect, but the unit is controlled by the SET ANGLE command.

6.4 CALIBRATE

LAS\CALIBRATE [KEY val] [KEY val] ...

This command show the calibration parameters if given without arguments. Called with a list of arguments in the form "KEY value", where KEY stands for BEAM_EFF, FORWARD_EFF, GAIN_IMAGE, H2O, P_AMB, T_AMB, T_CHOPPER or T_COLD, and value is the corresponding value for, respectively, a (new) Beam efficiency, Forward efficiency, Image/Signal gain ratio, mm of water vapor, ambient temperature, hot load (chopper) temperature, cold load temperature, will recompute the calibration, update the header and rescale the spectrum accordingly.

6.5 DROP

LAS\DROP Obs [Ver]

Remove an Observation from the current index. The version number must be

specified if it is not the last in the input file, even if it is the last version in the current index.

6.6 DUMP

LAS\DUMP [Argument]

A debugging command no longer used very much. Without an argument, it lists a certain number of informations about the R Observation header (A CERTAIN number doesnot mean ALL, although it should...). With "D[ATA]" as argument, it lists in addition the data values. With "F[ILE]" as argument, it lists information about the input and output file, and finally with "I[NDEX]", it lists information about the current index.

6.7 FILE

LAS\FILE Type Name [NEW]

Selects the input and output files.

FILE IN name	defines the input file
FILE OUT name [NEW]	defines the output file ; initializes a file if NEW is precised.
FILE BOTH name	selects the same file for input and output.

The default extension can be specified using command SET EXTENSION (.BUR is recommended).

6.8 FIND

LAS\FIND [Append] [New_Data] [/ALL] [/LINE Lname] [/NUMBER n1 n2] [/OFFSET o1 o2] [/SCAN s1 s2] [QUALITY q] [/RANGE w e s n] [/SOURCE Sname] [/TELESCOPE Tname]

FIND performs a search in the input file to build a new index, according to selection criteria defined by the SET command. These criteria may be temporarily modified by the following options:

/ALL	all versions of each scan are searched for (if not present: only the last version)
/LINE Lname	search by line name
/NUMBER n1 n2	search for the specified range of observation numbers
/SCAN s1 s2	search for the specified range of scan numbers
/OFFSET a1 a2	search for these offsets
/QUALITY q	search for the data of quality better than Q
/RANGE w e s n	search for the specified range of offsets
/SOURCE Sname	search by source name
/TELESCOPE Tname	search for the telescope used

FIND by default overwrites the current index. Found observations may be

appended to the current index by specifying the argument APPEND ; an index compression occurs to avoid duplication of scans in the index.

FIND does not return an error if the index is empty, but the variable FOUND is set to 0. FOUND is always set to the number of observations in the index.

Argument NEW_DATA can be used to wait until new data is present in the input file. This possibility is intended for sites where data acquisition is done in CLASS format (Pico Veleta, Plateau de Bure) to use CLASS as an automated quick look facility. The behaviour is similar to that of command NEW_DATA, but all selection criteria are considered and no switching to a new observation type occurs.

6.9 FITS

```
LAS\FITS READ  file[.fits]
LAS\FITS WRITE file[.fits] /MODE SPECTRUM|INDEX [/BITS nbits]
```

Either READ the file.fits FITS file to the CLASS internal data format. Please note that not all FITS files may be read by CLASS, and some header information might be incomplete when doing so.

Or WRITE the file.fits FITS file from the CLASS internal data format. /MODE is a mandatory argument which can take the following values:

```
SPECTRUM : a simple FITS file is written from the current spectrum
           in memory is written.
INDEX    : a FITS BINTABLE is written from all the spectra in current
           index. The USER MUST make sure that all index spectra are
           consistent (same source, same line, same frequency setup,
           ...)
```

The number of bits per intensity value used when writing the FITS file may be controlled through the /NBITS optional argument. It can be: 16 or I*2(2 bits integer), 32 or I*4 (4 bits integer) or -32 or R*4 (4 bits real).

The default mode and number of bits can also be specified through the commands

```
SET FITS BITS nbits
SET FITS MODE SPECTRUM|INDEX
```

6.10 FOLD

```
LAS\FOLD
```

Folds frequency-switched spectra.

6.11 GET

```
LAS\GET [N]
```

GET copies R into T, and loads the Observation number N in R. The current index initialized by FIND is first explored, then, if needed, the whole input file. If N is absent, the previous (last read) Observation is recovered. If N is FIRST, the first Observation of the current index is loaded. If N is NEXT, the next Observation in the current index is loaded.

6.12 HEADER

```
LAS\HEADER
```

Displays some header information on the R Observation. The FULL format is used, with the informations selected in command SET FORMAT written. The lines written contain (in output order, but not all lines are necessarily present)

- General information line:
  Scan number and version
  Source name
  Line name
  Telescope name
  Date of observation
  Date of last reduction
- Position information line
  RA or l: right ascension or longitude (or azimuth)
  DEC or b: declination or latitude (or elevation)
  Epoch if equatorial coordinates are used
  Offsets in current units
  Type of coordinate (Eq, Ga, Az)
- Quality information line:
  Quality of calibration (as defined by a SET QUALITY)
  Tau: opacity at zenith
  Tsys: system temperature (outside atmosphere).
  Time: total integration time on source (minutes)
  El: elevation of source
- Spectral information lines (2):
  N: number of channel
  I0: reference channel (real)
  V0: velocity at reference channel
  Dv: velocity resolution (signed)
  F0: rest frequency at reference channel
  Df: frequency resolution (signed)

```
    Fi: image frequency at reference channel
  - Calibration information line:
    B_ef : Beam efficiency of telescope
    F_ef : Forward efficiency of telescope
    G_im : Gain in image band
  - Atmospheric information lines (2):
    H2O : millimeter of precipitable water vapor
    Pamb: ambient pressure
    Tamb: ambient (receiver cabin) temperature
    Tchop: Chopper temperature
    Tcold: Cold load temperature
    Tatm: Atmospheric temperature in the signal band
    Tau: Zenith opacity in the signal band
    Tatm_i: Atmospheric temperature in the image band
    Tau_i: Zenith opacity in the signal band
  - Continuum Drift information lines (2) :
    N: Number of points
    I0: Reference point
    A0: Angle offset at reference point
    Da: Angular spacing between points (signed).
    F0: Observing frequency
    Df: Band pass
    Pos. Ang. : Position angle of the drift.
followed by the list of Scan numbers added in the observation.
```

6.13 IGNORE

```
    LAS\IGNORE List_of_Observations
```

This command can be used to declared the specified list of Observations (from the INPUT file) to be ignored in all FIND operations. They effectively become invisible to CLASS (except in a LIST IN command), until a FILE IN command is typed again. The input file is not physically modified however. The list of observation may have the same format as the index list of a FOR command.

6.14 Language

```
    LAS\ Command Language Summary
ACCUMULATE      : Add R and T observation.
BASE [arg]      : Subtract a baseline.
BOX             : Draw a frame for data.
CALIBRATE       : [Re-]Calibrates a spectrum.
DROP num [ver]  : Take a scan out of the current index.
DUMP            : List some informations on the R spectrum.
FILE type name  : Define the input/output files.
```

```
FIND            : Search the input file for observations.
FOLD            : Folds a Frequency Switched spectrum.
GET [N]         : Read a scan in the input file.
HEADER          : Display some header information on the R spectrum.
IGNORE List     : Ignore scans from the Input file.
LIST [name]     : List header information about an ensemble of scans.
MODIFY          : Edit and change the scan header.
MULTIPLY fact   : Multiply the R spectrum by fact.
NEW_DATA        : Wait until new data present in input file.
PLOT            : Plot the observation in R, with box and title.
SAVE [name]     : Save the current parameters.
SET             : Enter a value for a parameter.
SHOW Arg        : Display some parameter.
SPECTRUM [Yoff] : Plot the R observation.
SUM             : Add up all the observations of the current index.
SWAP            : Exchange the contents of the R and T buffers.
TAG Qual List   : Change the quality of scans in the Output file.
TITLE           : Write a header above the plotted frame.
UPDATE          : Update R in the output file.
WRITE [Scan]    : Write R in the output file.
```

6.14.1 Language NEWS

- Exit LAS, welcome CLASS: support for continuum data.
- Quality flag for scans: new commands IGNORE and TAG, command DELETE has been deleted.

6.15 LIST

```
LAS\LIST [Name] [/BRIEF] [/LONG] [/OUTPUT File]
```

List header information about an ensemble of Observations. LIST is used for a quick look to Observation headers, in a more or less detailed format. Name specifies the file to be listed: IN or OUT ; if Name is not present, the current index is listed. A medium-sized format is used by default.

```
/BRIEF          : Brief format (Observation numbers and version numbers)
/LONG           : Long format
/OUTPUT File    : Send the list to a disk file instead of the terminal
```

6.16 MODIFY

```
LAS\MODIFY Item [Values...]
```

This is a general purpose command to edit and change the current observation header. Some actions are merely presentation "details", others do affect the information in the observation. Among these, RECENTER, WIDTH, IMAGE and SWITCH_MODE are typically used only when the software did not

agree with the hardware when the spectra was taken (i.e. when the information in the observation header is wrong). They should be used with caution.

MODIFY actions are limited. You should use the SIC\ command LET to modify header variables which are not available within the MODIFY command (Caution: this is the rope to hang you).

6.16.1 MODIFY FREQUENCY

LAS\MODIFY FREQUENCY Rfreq

Compute the velocity scale corresponding to a new rest frequency Rfreq. The sky frequency scale is not modified.

6.16.2 MODIFY OFFSETS

LAS\MODIFY OFFSETS O1 O2

Change the offsets. This may be required when you average spectra from different offsets, because the resulting "position" is meaningless (usually the position of the first observation added).

6.16.3 MODIFY BLANKING

LAS\MODIFY BLANKING New0 Change the blanking value in both header and data. This may be required before you can build a consistent strip or cube, specially if the spectra were initially loaded from a FITS tape.

6.16.4 MODIFY RECENTER

LAS\MODIFY RECENTER Rrchan

Change the reference channel in order to change the velocity, frequency and image frequency scales. The sky frequency scale is also altered in this command.

6.16.5 MODIFY BANDS

LAS\MODIFY BANDS

Exchange signal and image bands. This allows line fitting on image band for example. Currently, the spectrum is not rescaled even if the band

ratio is not 1.

6.16.6 MODIFY LINENAME

LAS\MODIFY LINENAME "New Name"

Modify the line name (for bookeeping).

6.16.7 MODIFY VELOCITY

LAS\MODIFY VELOCITY Rvoff

Compute the image and rest frequencies corresponding to a new velocity (in km/s). The sky frequency scale is unchanged by this command.

6.16.8 MODIFY WIDTH

LAS\MODIFY WIDTH Rfres

Change the frequency resolution (in MHz).

6.16.9 MODIFY IMAGE

LAS\MODIFY IMAGE Rimage

Change the image frequency (in MHz).

6.16.10 MODIFY BEAM_EFF

LAS\MODIFY BEAM_EFF Value

Change the beam efficiency. If the beam efficiency previously had a non zero value, the spectrum is rescaled according to the new value. Otherwise, only the beam efficiency is set to the new value.

6.16.11 MODIFY TELESCOPE

LAS\MODIFY TELESCOPE Name

Change the telescope name, for instance from IRAM-30M to 30M-MRT.

6.16.12 MODIFY POSITION

LAS\MODIFY POSITION Lambda Beta

Change the central position for the source. The offsets are also modified, as to be consistent with the new central position. This command is usefull when different parts of one source are mapped relative to several reference positions.

6.16.13 MODIFY SYSTEM

LAS\MODIFY SYSTEM System

Change the System in which the center of projection (central position of the source) has been defined. Valid systems are GALACTIC, EQUATORIAL, HORIZONTAL and UNKNOWN. This command is typically used for "foreign" data when the SYSTEM information has not been decoded correctly by the conversion package.

6.16.14 MODIFY SWITCH_MODE

LAS\MODIFY SWITCH_MODE Switching_mode Nphase Off_1 Time_1 Weight_1 Off_n Time_n Weight_n

Change the description of the switching mode. This command is typically used for "foreign" data when this information has not been decoded correctly by the conversion package.

For each phase, one should specify:
- the offset for the phase (in MHz for frequency switching, in current angular unit otherwise).
- the duration of the phase (in seconds)
- the weight given to phase (arbitrary units)

6.17 MULTIPLY

LAS\MULTIPLY Fact

Multiply the R Observation by Fact. MULTIPLY leaves the T Observation unaffected. Note that Tsys is also multiplied by the same (absolute) factor to keep proper weights. If only the data is to be multiplied, use the SIC\LET command on array RY:

6.18 NEW_DATA

[LAS\]NEW_DATA

Waits until new data has been written to the input file, then makes a new index from all new data. The type of observation can be changed by this command, but the index will contain only observations of one type. Selection criteria defined by command SET are ignored.

This command can be used to make a quick look or analysis of data produced by a real time acquisition system. Caution: only two programs should access the input file simultaneously, one for writing the other for reading.

6.19 PLOT

LAS\PLOT [/SCAN]

Display the Observation in R. PLOT is equivalent to the following sequence of commands:
CLEAR BOX SPECTRUM TITLE
PLOT /SCAN will plot an on-the fly scan as a 2-dimensional image with velocity/frequency in the X axis, record number in the Y axis, intensity being rendered as grey/color scale. Entering command POPUP after PLOT/SCAN enables displaying a single spectrum, selected by clicking with the middle button of the mouse.

6.20 SAVE

LAS\SAVE Name

SAVE creates a procedure file of name "Name.CLASS", containing all the current parameters of the program. This file may be executed at any time using the @ command: just type "@ name" after the LAS> prompt, or pass "@ name" as a parameter when invoking CLASS (by typing "CLASS @ name"). This file is composed of standard CLASS commands, and may be edited with any text editor.

6.21 SET

LAS\SET something [value1 [value2 [...]]] [/NOCURSOR]

This command is used to set a value for a CLASS parameter. If no argument is given, the default value for the parameter will be restored.

6.21.1 SET DEFAULT

LAS\SET DEFAULT

Restore most (hopefully all) parameters to their default value. The default values for individual parameters are documented under the corresponding subtopic.

6.21.2 SET PROCESSING

The following SET Parameter commands control data processing options:

```
LAS\SET ALIGN Type Range
LAS\SET BAD Check
LAS\SET BASE arg
LAS\SET CALIBRATION [Beam_Tol [Gain_Tol]] or OFF
LAS\SET CURSOR Arg
LAS\SET MASK [ml1 ml2 [...]]
LAS\SET MATCH Tol
LAS\SET NOMATCH
LAS\SET TYPE CONTINUUM or LINE or SKYDIP
LAS\SET VARIABLE Section_Name [Keyword]
LAS\SET VELOCITY L[sr]H[eliocentric]A[utomatic]
LAS\SET WEIGHT type
LAS\SET WINDOW [wl1 wu1 [wl2 wu2 [...]]]
```

6.21.3 SET DISPLAY

The following SET Parameter commands affect data display:

```
LAS\SET CURSOR Arg
LAS\SET FORMAT Type
LAS\SET LEVEL interactive_min_value logfile_min_value
LAS\SET MODE A [Type]
LAS\SET PLOT type
LAS\SET SYSTEM Type [Equinox]
LAS\SET UNIT Lower [Upper]
LAS\SET VARIABLE Section_Name [Keyword]
LAS\SET VELOCITY L[sr]H[eliocentric]A[utomatic]
```

6.21.4 SET SELECTION

The following SET Parameter commands affect data selection in FIND:

```
LAS\SET ANGLE Unit
LAS\SET LINE NAME
LAS\SET MATCH Tol
LAS\SET NOMATCH
LAS\SET NUMBER n1 n2
LAS\SET OBSERVED d1 d2
LAS\SET OFFSETS o1 o2
```

```
LAS\SET POSANGLE amin amax
LAS\SET QUALITY q
LAS\SET RANGE West East South North
LAS\SET REDUCED d1 d2
LAS\SET SCAN s1 s2
LAS\SET SOURCE name
LAS\SET SYSTEM Type [Equinox]
LAS\SET TELESCOPE Name
LAS\SET TYPE CONTINUUM or LINE or SKYDIP
```

6.21.5 SET ALIGN

```
LAS\SET ALIGN Type Range
```

Defines the way the spectra are aligned to be added together. Type may be C[hannel], V[elocity] or F[requency] for spectra, C[hannel] or P[osition] for drifts. The range may be I[ntersect] or C[omposite]. The default is CHANNEL INTERSECT. This means that continuum drifts in opposite directions are not properly added: ALIGN Position must be specified for that.

6.21.6 SET ANGLE

```
LAS\SET ANGLE Unit
```

Specify the angle unit for offsets. May be R[adian], D[egree], M[inute of arc], S[econd of arc]. Default is MINUTE. The ANGLE unit is also used to display Continuum drifts.

6.21.7 SET BAD

```
LAS\SET BAD Check
```

Specify the way ACCUMULATE and SUM behaves in the case of bad channels. Check may be OR or AND. Default is OR, i.e. the output channel is bad if any of the input channels is bad.

6.21.8 SET BASELINE

```
LAS\SET BASELINE arg
```

Specify the degree of polynomial baselines to be subtracted. Default is 1.

6.21.9 SET CALIBRATION

```
LAS\SET CALIBRATION [Beam_Tol [Gain_Tol]] or OFF
```

Specify the tolerance on the Beam efficiency (Beam_Tol) and gain image ratio (Gain_Tol), on turn or calibration checking. A value of 0 means no check. The default values are 0.02 and 0 respectively. These values are used by command SUM to verify that the calibration is coherent, and by command WRITE which writes the corresponding information in the output file only if Beam_Tol is non zero. Note that because the beam efficiency is less than 1, you could use Beam_Tol=1 to suppress the calibration checking but still write the information.

6.21.10 SET COORDINATES

```
LAS\SET COORDINATES [Automatic|Galactic|Equatorial]
```

Specify the system used when displaying the absolute coordinates of the source.

6.21.11 SET CURSOR

```
LAS\SET CURSOR Arg
```

Activates or deactivates the use of cursor for the commands SET WINDOW, SET MASK, SET MODE and LINES, according to the value of Arg: ON or OFF. Default is OFF.

6.21.12 SET EXTENSION

```
LAS\SET EXTENSION Ext
```

Defines the default extension for input and output files. Default is .BUR.

6.21.13 SET FIND

```
LAS\SET FIND UPDATE|NOUPDATE
```

FIND will check (or not) for new scans in the input file (see FIND New_Data).

6.21.14 SET FORMAT

```
LAS\SET FORMAT Type
```

Set the format of the title of observations plotted. The same format is used by the LIST command. The format may be BRIEF (Observation number and version, Source name, Line name, Telescope name and position offsets), LONG or FULL which is similar to the long format, but also displays the list of original Scans. The default is BRIEF. FULL format is always used for command HEADER.

Other keywords can be specified to indicate which type of information should be written for the LONG and FULL formats. This is done by command
LAS\SET FORMAT Keyword [ON] [OFF] (ON to write the information corresponding to the specified keyword, OFF to ignore it). The keywords are

- POSITION for position of the source, offsets, type of coordinates and Epoch
- QUALITY for the Opacity, System temperature, Elevation, Integration time.
- SPECTRAL for the number of channel, reference channel, velocity and resolution, signal and image frequency and resolution.
- CALIBRATION for the beam and forward efficiencies and the gain image ratio.
- ATMOSPHERE for the water vapor content, pressure, temperatures and opacities in signal and image bands.
- CONTINUUM for continuum drift information
- ORIGIN for the list of added scans.

The default is Position - Quality - Spectral for spectra, Position - Quality - Continuum for drifts. A line of general information is always written in any case.

6.21.15 SET LEVEL

LAS\SET LEVEL interactive_min_value logfile_min_value

Sets the level (0 to 10) of informational messages returned to the user in interactive mode and in the message file (CLASS.MES). The level attributed to a message (i.e., a warning message, an error message, a severe-error message) is somewhat dependent of the good humor of the programmer, but typically aknowledgement messages range 0-2, warnings 3-5, errors 6-8, severe-errors 9-10. To prevent printing of message milder than 7 at interactive level and 4 in CLASS.MES, type

LAS\SET LEVEL 7 4

6.21.16 SET LINE

LAS\SET LINE Name

FIND will select all observations according to the specified line name. Default is *. A syntax like NAM* can be used to find observations with line names beginning by NAM.

6.21.17 SET MASK

LAS\SET MASK [ml1 ml2 [...]]

Defines a mask for the Gaussian fits by its boundaries. The cursor will be used to specify missing arguments if SET CURSOR is ON, unless option /NOCURSOR is present. When using the cursor, enter: N to enter a value, C to cancel the last entry, H for help, E for exit. Default is no mask. The boundaries are in current unit.

6.21.18 SET MAP

LAS\SET MAP [Clear|Normal]

Will Clear (or Not) the screen before processing a MAP command. However, a "MAP Where" command does not clear the screen even in a "LAS\SET MAP Clear" mode.

6.21.19 SET MATCH

LAS\SET MATCH Tol

Turn on the checking of position-matching in ACCUMULATE. Tol is the position tolerance in current units. The default is SET MATCH 2 (arcsec). The tolerance is also used in FIND, MAP and a few other commands such as CUBE.

6.21.20 SET MODE

LAS\SET MODE A [Type]

Select the scale mode for axis A (X or Y). The mode can be TOTAL (all data plotted), AUTO (limits as found in the input file), or Amin, Amax for fixed limits. If no Type argument is given, the graphic cursor will be invoked to define the limits if SET CURSOR ON is selected. The default is AUTO.

6.21.21 SET NOMATCH

LAS\SET NOMATCH

Turn off checking of position-matching in ACCUMULATE and SUM. This does not change the tolerance for the other functions like FIND and MAP.

6.21.22 SET NUMBER

```
LAS\SET NUMBER n1 n2
```

FIND will select all Observations with numbers between n1 and n2. Default is * *.

6.21.23 SET OBSERVED

```
LAS\SET OBSERVED d1 d2
```

FIND will select all Observations with observing dates between d1 and d2. Dates are specified as DD-MMM-YYYY (i.e. 14-JUL-1789). Default is * *.

6.21.24 SET OFFSETS

```
LAS\SET OFFSETS o1 o2
```

FIND will select all observations with offsets within the specified tolerance (SET MATCH) of o1 and o2. See also SET RANGE. Default is * * (select all offsets).

6.21.25 SET PLOT

```
LAS\SET PLOT type
```

Set the plotting type for the spectra: N[ormal] for broken lines, or H[istogram]. Default is normal (50% faster plot).

6.21.26 SET POSANGLE

```
LAS\SET POSANGLE amin amax
```

Specify the position angle range (continnum drifts only) for subsequent searches.

6.21.27 SET QUALITY

```
LAS\SET QUALITY q
```

FIND will select only observations of quality better than Q (i.e. less than Q). When originally written, unless the real time acquisition system detected a severe problem, all observations have quality 0, a priori the best. The quality flag of an observation can be changed using the TAG command. See HELP TAG for the recommended quality scale.

6.21.28 SET RANGE

```
LAS\SET RANGE West East South North
```

FIND will select all observations with offsets in the specified range. Default is * * * *.

6.21.29 SET REDUCED

```
LAS\SET REDUCED d1 d2
```

FIND will select all observations with reduction dates between d1 and d2. Dates are specified as DD-MMM-YYYY (i.e. 14-JUL-1789). Default is * *.

6.21.30 SET SCAN

```
LAS\SET SCAN s1 s2
```

FIND will select all observations with Scan numbers between s1 and s2. The Scan number, attributed by the on-line acquisition system may be different from the observation number which is used by CLASS to refer to the observations. Default is * *.

6.21.31 SET SOURCE

```
LAS\SET SOURCE name
```

FIND will select all scans according to the specified source name. Default is *. Syntax like NAM* can be used to find all observations with source name beginning by NAM.

6.21.32 SET SYSTEM

```
LAS\SET SYSTEM Type [Equinox]
```

Specify the coordinate system to be used. Type may be E[quatorial] followed by an epoch, G[alactic], or A[utomatic]. Default is Auto, i.e. uses the type found in the data file. If the equinox of Equatorial coordinates is not specified, the previous value is used. The initial default is 1950.00. Coordinates and offsets are automatically converted to the specified coordinate system and precessed to the specified epoch if needed.

6.21.33 SET TELESCOPE

LAS\SET TELESCOPE Name

FIND will select all scans according to the specified telescope name. Default is *. Syntax like NAM* can be used to find all observations with telescope name beginning by NAM.

6.21.34 SET TYPE

LAS\SET TYPE CONTINUUM or LINE or SKYDIP

Defines on which type of data the program is working.

- LINE (or SPECTROSCOPY) (the default when entering LAS) selects Spectral Line data only, and the program prompt is LAS> (Line Analysis System).
- CONTINUUM selects continuum drifts results only, and the program prompt changes to CAS> (Continuum Analysis System).
- SKYDIP selects only skydips, and the prompt changes to SAS>

In CONTINUUM mode, a continuum drift is treated as a spectrum would be in LINE mode. You can add drifts, subtract baseline, make gauss fits, change plotting scales etc... The current angular units is used in plots. SKYDIP mode has very limited capabiblities: only command REDUCE may be used to analyse the SKYDIP results.

6.21.35 SET UNIT

LAS\SET UNIT Lower [Upper]

Specify the unit to be used for the lower and upper X axis. Allowed units are C[hannel], F[requency], [Image] and V[elocity]. Default is Velocity. For Continuum drifts, this unit is ignored, but the angular unit specified by SET ANGLE is used.

6.21.36 SET VARIABLE

LAS\SET VARIABLE Section_Name [Keyword]

Declare SIC variables corresponding to section of the header of the R observation. The header sections are POSITION, SPECTRO, BASE, HISTORY, PLOT, FSWITCH, CALIBRATION, SKYDIP, GAUSS, SHELL, NH3 (or HFS), DRIFT, BEAM, and CONTINUUM. Defining these SIC variables allows to access the header for mathematical computations using SIC facilities. All variables are declared as readonly, unless Keyword WRITE is specified as third argument. If Keyword OFF is specified, the corresponding SIC variables are deleted (to save space since the total number of SIC variables is limited).

The SIC variables have the name of the corresponding FORTRAN variables stripped from the first character (always R in the FORTRAN code). They are defined below

```
              GENERAL General parameters, always present:
INTEGER MGEN
INTEGER MGEN2
PARAMETER (MGEN=9,MGEN2=11)
INTEGER  RNUM           ! Observation number
INTEGER  RVER           ! Version number
INTEGER  RTELES(3)      ! Telescope name
INTEGER  RDOBS          ! Date of observation
INTEGER  RDRED          ! Date of reduction
INTEGER  RTYPEC         ! Type of coordinates
INTEGER  RKIND          ! Type of data
INTEGER  RQUAL          ! Quality of data
INTEGER  RSCAN          ! Scan number
REAL*8  RUT             ! UT of observation
REAL*8  RST             ! LST of observation
REAL*4  RAZ             ! Azimuth
REAL*4  REL             ! Elevation
REAL*4  RTAU            ! Opacity
REAL*4  RTSYS           ! System temperature
REAL*4  RTIME           ! Integration time

                    POSITION Position information:
INTEGER MPOS
PARAMETER (MPOS=11)
INTEGER  RSOURC(3)          ! Source name
REAL*4  REPOCH              ! Epoch of coordinates
REAL*8  RLAM                ! Lambda
REAL*8  RBET                ! Beta
REAL*4  RLAMOF              ! Offset in Lambda
REAL*4  RBETOF              ! Offset in Beta
```

```
INTEGER  RPROJ               ! Projection system

            SPECTRO Spectroscopic information (for spectra)
INTEGER MSPEC
PARAMETER (MSPEC=18)
INTEGER  RLINE(3)            ! Line name
REAL*8  RRESTF               ! Rest frequency
INTEGER  RNCHAN              ! Number of channels
REAL*4  RRCHAN               ! Reference channels
REAL*4  RFRES                ! Frequency resolution
REAL*4  RFOFF                ! Frequency offset
REAL*4  RVRES                ! Velocity resolution
REAL*4  RVOFF                ! Velocity at reference channel
REAL*4  RBAD                 ! Blanking value
REAL*8  RIMAGE               ! Image frequency
INTEGER  RVTYPE              ! Type of velocity
REAL*8  RSKYFR               ! Sky frequency
REAL*4  RVTELES              ! Velocity of telescope /reference frame

           BASE Baseline information (for spectra of drifts)
INTEGER MWIND
INTEGER MBASE
PARAMETER (MWIND=20,MBASE=4+2*MWIND)
INTEGER  RDEG                ! Degree of last baseline
REAL*4  RSIGFI               ! Sigma
REAL*4  RAIRE                ! Area under windows
INTEGER  RNWIND              ! Number of line windows
REAL*4  RW1(MWIND)           ! Lower limits of windows
REAL*4  RW2(MWIND)           ! Upper limits of windows

              HISTORY Scan numbers of initial observations
INTEGER MSEQ
INTEGER MORIG
PARAMETER (MSEQ=100,MORIG=2*MSEQ+1)
INTEGER  RNSEQ               ! Number of sequences
INTEGER  RSTART(MSEQ)        ! Start can number of seq.
INTEGER  REND(MSEQ)          ! End scan number of seq.

                     PLOT Default plotting limits.
INTEGER MPLOT
PARAMETER (MPLOT=4)
REAL*4  RAMIN                ! Min Y value plotted
REAL*4  RAMAX                ! Max Y value plotted
REAL*4  RVMIN                ! Min X value plotted
REAL*4  RVMAX                ! Max X value plotted

          FSWITCH Frequency switching information (for spectra)
```

```
INTEGER MXPHAS
INTEGER MFSW
PARAMETER (MXPHAS=8,MFSW=2+6*MXPHAS)
INTEGER  RNPHAS              ! Number of phases
REAL*8  RDECAL(MXPHAS)       ! Frequency offsets
REAL*4  RDUREE(MXPHAS)       ! Time per phase
REAL*4  RPOIDS(MXPHAS)       ! Weight of each phase
INTEGER  RSWMOD              ! Switching mode (frequency, position...)
REAL*4  RLDECAL(MXPHAS)      ! Lambda offsets
REAL*4  RBDECAL(MXPHAS)      ! Beta offsets of each phase

                  CALIBRATION Calibration parameters
INTEGER MCALIB
PARAMETER (MCALIB=21)
REAL*4  RBEEFF               ! Beam efficiency
REAL*4  RFOEFF               ! Forward efficiency
REAL*4  RGAINI               ! Image/Signal gain ratio
REAL*4  RH2OMM               ! MM of water vapor
REAL*4  RPAMB                ! Ambient pressure (hPa)
REAL*4  RTAMB                ! Ambient temperature (K)
REAL*4  RTATMS               ! Atmosphere temp. signal band
REAL*4  RTCHOP               ! Chopper temperature
REAL*4  RTCOLD               ! Cold load temperature
REAL*4  RTAUS                ! Opacity signal band
REAL*4  RTAUI                ! Opacity image band
REAL*4  RTATMI               ! Atmosphere temp. image band
REAL*4  RTREC                ! Receiver temperature
INTEGER  RCMODE              ! Calibration mode
REAL*4  RATFAC               ! Applied calibration factor
REAL*4  RALTI                ! Site elevation
REAL*4  RCOUNT(3)            ! Power of Atm., Chopp., Cold
REAL*4  RLCALOF              ! Longitude offset for sky measurement
REAL*4  RBCALOF              ! Latitude offset for sky measurement

          SKYDIP For Skydips observations. No associated data.
INTEGER MSKY
INTEGER MSKYDIP
PARAMETER (MSKY=10,MSKYDIP=10+4*MSKY)
INTEGER  RSLINE(3)           ! Line name
REAL*8  RSREST               ! Rest frequency
REAL*8  RSIMAG               ! Image frequency
INTEGER  RNSKY               ! Number of points on sky
INTEGER  RNCHOP              !    -   -    -    - chopper
INTEGER  RNCOLD              !    -   -    -    - cold load
REAL*4  RELEV(MSKY)          ! Elevations
REAL*4  REMISS(MSKY)         ! Power on sky
REAL*4  RCHOPP(MSKY)         ! Power on chopper
```

```
REAL*4  RCOLD(MSKY)          ! Power on cold load

            GAUSS Gauss fit results (for spectra or drifts)
INTEGER MXGAUS
INTEGER MFIT
PARAMETER (MXGAUS=5,MFIT=3+6*MXGAUS)
INTEGER  RNLINE              ! Number of components
REAL*4  RSIGBA               ! Sigma on base
REAL*4  RSIGRA               ! Sigma on line
REAL*4  RNFIT(3*MXGAUS)      ! Fit results
REAL*4  RNERR(3*MXGAUS)      ! Errors

        SHELL "Stellar shell" profile fit results (for spectra)
INTEGER MSHELL
PARAMETER (MSHELL=43)
INTEGER  RLSHEL              ! Number of components
REAL*4  RBSHEL               ! Sigma on base
REAL*4  RRSHEL               ! Sigma on line
REAL*4  RNSHEL(20)           ! Fit results
REAL*4  RESHEL(20)           ! Errors

             HFS "Hyperfine Structure" profile fit results
                         (for spectra, ex: NH3, HCN)
INTEGER MNH3
PARAMETER (MNH3=27)
INTEGER  RLNH3               ! Number of components
REAL*4  RBNH3                ! Sigma on base
REAL*4  RRNH3                ! Sigma on line
REAL*4  RNNH3(12)            ! Fit results
REAL*4  RENH3(12)            ! Errors

        ABS "Hyperfine Structure" ABSORPTION profile fit results
                         (for spectra, ex: NH3, HCN)
INTEGER MABS
PARAMETER (MABS=35)
INTEGER  RLABS               ! Number of components
REAL*4  RBABS                ! Sigma on base
REAL*4  RRABS                ! Sigma on line
REAL*4  RNABS(16)            ! Fit results
REAL*4  REABS(16)            ! Errors

             DRIFT Continuum drift description (for drifts)
INTEGER MCONT
PARAMETER (MCONT=16)
REAL*8  RFREQ                ! Rest frequency
REAL*4  RWIDTH               ! Bandwidth
INTEGER  RNPOIN              ! Number of data points
```

```
REAL*4  RRPOIN              ! Reference point
REAL*4  RTREF               ! Time at reference
REAL*4  RAREF               ! Angular offset at ref.
REAL*4  RAPOS               ! Position angle of drift
REAL*4  RTRES               ! Time resolution
REAL*4  RARES               ! Angular resolution
REAL*4  RCBAD               ! Blanking value
INTEGER  RCTYPE             ! Type of offsets
REAL*8  RCIMAG              ! Image frequency
REAL*4  RCOLLA              ! Collimation error Az
REAL*4  RCOLLE              ! Collimation error El

         BEAM Beam-switching parameters (for spectra or drifts)
INTEGER MBEAM
PARAMETER (MBEAM=5)
REAL*4  RCAZIM              ! Azimuth of observation
REAL*4  RCELEV              ! Elevation of observation
REAL*4  RSPACE              ! Beam spacing
REAL*4  RBPOS               ! Position angle of beams
INTEGER  RBTYPE             ! System for angle

    CONTINUUM Double gaussian and baseline fit results (for drifts)
INTEGER MFCONT
PARAMETER (MFCONT=19)
INTEGER  RLCONT             ! Number of components
REAL*4  RBCONT              ! Sigma on base
REAL*4  RRCONT              ! Sigma on line
REAL*4  RNCONT(8)           ! Results
REAL*4  RECONT(8)           ! Errors
```

6.21.37 SET VELOCITY

```
LAS\SET VELOCITY L[sr]H[eliocentric]A[utomatic]
```

Forces the velocity axis values to Lsr or Heliocentric values. Automatic reverts to the default mode, where the velocity type is taken from the header of the spectrum.

6.21.38 SET WEIGHT

```
LAS\SET WEIGHT type
```

Weighting to be used for summations: T[ime], E[qual] or S[igma] (for 1/sigma**2). The default is TIME. Sigma is not recommended unless you just made a baseline fit before. Equal behaves differently in SUM (produces the average) and ACCUMULATE (produces the SUM).

6.21.39 SET WINDOW

```
LAS\SET WINDOW [wl1 wu1 [wl2 wu2 [...]]]
```

Defines one or several line windows to be avoided by BASE when fitting a baseline. The cursor will be used to specify missing arguments if SET CURSOR is ON, unless option /NOCURSOR is present. When using the cursor, enter: N to enter a value, C to cancel the last entry, H for help, E for exit. The boundaries are in current unit. There is no default.

6.22 SHOW

```
LAS\SHOW Arg
```

Display some parameter defined by SET. Arg can be "ALL" to display all parameters specified by SET. See SET command.

6.23 SPECTRUM

```
LAS\SPECTRUM [Yoff]
```

Plots the spectrum in the R buffer in the units and scales defined by SET UNIT and SET MODE. Yoff (unit Kelvins, default 0) is a constant Y-offset added to the plotted values (useful to compare two spectra).

6.24 SUM

```
LAS\SUM
```

Adds all the spectra of the current index using the current weighting function (SET WEIGHT) and alignment mode (SET ALIGN). The position coincidence will be checked according to SET MATCH. The homogeneity of calibration is checked according to SET CALIBRATION. Bad channels are handled according to SET BAD. A sum may be interrupted by <^C>, but the results is then undefined.

If the data are On-The-Fly scans, all the individual positions are averaged.

6.25 SWAP

```
LAS\SWAP
```

Exchange the contents of the R and T buffers.

6.26 TAG

LAS\TAG Quality_Code List_of_Observations

Attributes a quality to a given list of Observations. Quality_Code is an integer in the range 0-9, and the recommended quality scale is

0 Unknown
1 Excellent
2 Good
3 Fair
4 Average
5 Poor
6 Bad
7 Awful
8 Should never exist
9 Reserved for deleted Observations

The operation is immediate and occurs in the OUTPUT file for all versions of all Observations specified in the list. If no list is given, the R memory is attributed the specified quality. A FIND operation will only select Observations of quality better than (i.e. less or equal to) the quality specified by the SET QUALITY command, or in the /QUALITY option.

6.27 TITLE

LAS\TITLE [/BRIEF] [/LONG]

Writes the header of the Observation currently in R above the plotted data. The format used is the one selected by SET FORMAT, which may be temporarily overridden by the options.

6.28 UPDATE

LAS\UPDATE

Update the LAST version of the observation in the R buffer in the output file ; the update OVERWRITES the observation. If more space is needed (e.g., the new version has more header information), an error is returned. You MUST then WRITE the observation to avoid loss of information. You can update only last versions. This command MUST BE USED WITH CAUTION ; in case of complex manipulations of observations, you may get confused and erase precious data.

The only typical case where it may be used is in the following sequence:
GET Number

GAUSS or MODIFY
UPDATE

6.29 WRITE

LAS\WRITE [observation]

Writes the observation in the R buffer onto the output file ; if a version of this observation is already in the file, a new version of the observation is created. A new observation number can be specified as argument. The information written depends on the operations done on the observation. Existing fit results are written. The calibration information is written only if the calibration is checked (see SET CALIBRATION command).

7 ANALYSE Language Internal Help

7.1 COMMENT

ANALYSE\COMMENT [TYPE|READ|WRITE|APPEND|EDIT|DELETE] ["comment"]

This command READs, TYPEs or DELETEs an existing comment section (max. length: 1020 characters). The options WRITE, APPEND and EDIT will create a comment section if none is present; the options WRITE or APPEND expect as argument the comment line (eventually inclosed in double quotes (") when the comment is more than one word long).

7.2 CUBE

ANALYSE\CUBE Filename [MATH_EXPRESSION] [/CHANNEL Cmin Cmax] [/STEP Stepx Stepy]

This command builds either a data-cube or a 2-dimensional map at the GILDAS format from the current index of scans. It is up to the user to ensure that the current index concerns a single source and a single line, observed with a unique spectral resolution, with all spectra centered at the same velocity.

If 'MATH_EXPRESSION' is present in the command line, ANALYSE\CUBE will make a 2-D map of (the result of Math_expression applied for each spectrum), for each spectra position. This is particulary handy when one use the SIC variables and functions (see HELP CUBE Functions, HELP LAS\SET VARIABLE and try the command SIC\EXAMINE /FUNCTION) in the Math_expression. In that case, The data cube axis types will be

RA or L
DEC or B, depending on the type of offsets

In the other case (i.e., a real data cube of channels values), The data cube axis types will be
Velocity or Channels or Rest or Image frequencies, depending on the current X unit.
RA or L
DEC or B, depending on the type of offsets

Option /CHANNEL can be used to limit the part of the spectra that will be included in the cube (no use if you plot a Math_expression).

Option /STEP can always be used to force the step size of the cube.

The resulting 2 or 3D data cube can then be processed as a whole using the GILDAS (Grenoble Image and Line Data Analysis System) software. GILDAS includes tasks to transpose, reinterpolate, change the projection systems, make overlays with the IRAS point sources, contours and so on from data cubes. Call GRAPHIC at system level and then type EXPLAIN for more information on GILDAS tasks.

7.2.1 CUBE FUNCTIONS

The Following Functions have been defined:
STDV(dummy) returns the integrated area (K.Km/s) of the current spectrum
TPEAK(dummy) returns the Peak temperature (K) of the current spectrum
VPEAK(dummy) returns the Velocity at Peak (Km/s) of the current spectrum
TDV(v1,v2) returns the integrated area (K.Km/s) of the current spectrum between velocities V1 and V2
TPEAKV(v1,v2) returns the Peak temperature (K) of the current spectrum between velocities V1 and V2
VPEAKV(v1,v2) returns the Velocity at Peak (Km/s) of the current spectrum between velocities V1 and V2
To create a map of The integrated area of your spectra, type:
CUBE filename STDV(0)

7.3 DISPLAY

ANALYSE\DISPLAY

Prints the results of the profile fit of R for the current method, if any result of course. For the Gauss (default) Method (see HELP METHOD), the command displays (1) the Line number, (2) the Area, (3) the Posi-

tion, (4) the Width and (5) the Intensity , each with their associated errors, and resulting RMS in the Base and Line windows.

7.4 DIVIDE

ANALYSE\DIVIDE Threshold

This command divides the spectrum in R by the spectrum in T, which is left unchanged. The two spectra must be on the same velocity scale (Use command RESAMPLE to do so). Threshold is a value to avoid zero divide. The output spectrum is blanked for channels where abs(T) < Threshold.

7.5 DRAW

ANALYSE\DRAW [Action [Parameters]]

This command executes the specified drawing action, or calls the graphic cursor and uses the returned character from the cursor to execute the corresponding action. The mnemonic for the cursor characters are : .nf
F for "FILL" To interpolate a bad channel
K for "KILL" To flag a bad channel
L for "LOWER" To flag line identifications, ...
M for "MASK" To show active masks
T for "TEXT" To annotate your spectrum
U for "UPPER" Similar to LOWER
W for "WINDOW" To show current line windows .fi any other to display the channel number, frequency, velocity and antenna temperature at the cursor position. The explicit form does not call the cursor and can be used in procedures.

7.5.1 DRAW FILL

ANALYSE\DRAW FILL Nchan

Interpolates the value of the specified bad channel using the nearest valid channels. Channel Nchan must be blanked before. In cursor mode, the channel number is retrieved from the cursor position.

7.5.2 DRAW KILL

ANALYSE\DRAW KILL Nchan

Deletes the specified channel, i.e. attributes the blanking value to it. In cursor mode, the channel number is retrieved from the cursor posi-

tion.

7.5.3 DRAW LOWER

```
ANALYSE\DRAW LOWER Value "text"
```

Draws a vertical line followed by the text "text" (written vertically also) at abscissa Value on lower axis. The line length is adjusted to approach closely the spectrum value at the corresponding abscissa.

7.5.4 DRAW MASK

```
ANALYSE\DRAW MASK [level]
```

Plots the active masks, with bars at the mask positions and Y=Level (Default 0).

7.5.5 DRAW UPPER

```
ANALYSE\DRAW UPPER Value "text"
```

Draws a vertical line followed by the text "text" (written vertically also) at abscissa Value on the upper axis. The line length is adjusted to approach closely the spectrum value at the corresponding abscissa.

7.5.6 DRAW TEXT

```
ANALYSE\DRAW TEXT X Y "text" I
```

Writes (horizontally) the string "text" at plot position (X,Y) in paper units, with the centering code I for the string.

7.5.7 DRAW WINDOW

```
ANALYSE\DRAW WINDOW [level]
```

Plots the current line windows, with bars at the mask positions and Y=Level (Default 0).

7.6 FFT

```
ANALYSE\FFT [/REMOVE]
```

FFT computes the Fourier Transform of R and plots it. R and T observa-

tions are unaffected. There is no control over the plot units, and no associated coordinate system (i.e. you cannot measure periods with the cursor). You can edit the Fourier Transform using the cursor. Strike key K to mark the beginning of an area to delete, move the cursor to the end of the area to delete and strike key K again. Then strike E to exit, and Fourier components are interpolated in module and phase from the boundaries of deleted areas.

Non interactive editing is also possible and can be useful for repetitive applications. The syntax is then FFT Start End [/REMOVE]

7.6.1 FFT /REMOVE

ANALYSE\FFT /REMOVE

Remove existing fits of the current method before computing the Fourier transform. If the Fourier Transform is edited, fits are added back upon exit.

7.7 FILL

LAS\FILL start end

Linearly interpolates the spectrum between start and end values (current units, not necessary channels). Can be used to remove mesospheric lines, several adjacent bad channels...

7.8 FIT

ANALYSE\FIT [N]

Plot the current fitted profile. The Nth line obtained the last time GAUSS was executed, or recovered by DISPLAY, is plotted on the screen. If N is 0, the sum of all lines is plotted, and if N is not given, the last value of N is used.

7.9 GAUSS

ANALYSE\GAUSS

GAUSS performs a fit of a theoretical profile in the R spectrum using initial values specified by the command LINES. The profile is determined by the METHOD command (GAUSS, SHELL or NH3 lines are currently implemented, and a special method, CONTINUUM, may be used for continuum da-

ta).

If LINES is 0, an attempt to guess initial values for a single line is done by computing the moments of the spectrum. A first minimisation is performed using the Simplex technique; the results are improved and the uncertainties computed using the Gradient technique. GAUSS uses the masks defined by SET MASK. If GAUSS did not converge, the errors quoted in the results are meaningless, and you should use command ITERATE to try to do better.

For CONTINUUM method, although a linear baseline is fitted with the gaussian, only the gaussian parameters are displayed. If the observation was made using beam-switching method and the reference beam appears in the drifts, a two-component gauss fit is done using separation specified in the data.

7.10 GREG

```
ANALYSE\GREG Name [/FORMATTED]  [/READ [X n] [Y m]
```

Writes in file Name the contents of current scan as a GILDAS table, or as a formatted file if the /FORMATTED option is present.

If the /READ option is present, performs the reverse operation: load the file content in the current scan. Only available for formatted files; the equivalent capability for GILDAS tables is available through a combination of DEFINE IMAGE and MODEL.

For the table, the information stored is :

1. Intensity
2. Channel number
3. Velocity
4. Offset frequency
5. Rest frequency
6. Image frequency
7. Fitted profiles if any - fit(i),i=0,nline - in column 7+i

The output table can be put later in a formatted way using GILDAS task LIST in you need.

The table may be used as input to GreG to produce fancy plots, or by the GILDAS software for other applications. In particular, SIC variables can be used to subtract any of the fits from the spectrum to produce residuals if needed. For plots, note that you can produce essentially the same within CLASS, except under very special circumstances, and moreover that only CLASS can handle the different X-unit simultaneously.

For the formatted list, less information is written (and it takes longer

```
to write it of course) :
    1. Current X unit in column 1
    2. Intensity in column 2
    3. Fitted profiles - (fit(i),i=0,nline) - in column 3+i

    For the /READ option, columns are X 1 Y 2 by default but can be
    changed. If X 0 is given as zero, then only Y is read and X is
    filled with channel numbers. Only very coarse header information
    is filled in. A lot of complementary header editing will often
    be needed.
```

7.11 GRID

```
    ANALYSE\GRID [Filename] [NEW|APPEND] [MATH_EXPRESSION]
    [/CHANNEL [C|V|F|I] min max]
    [/IMAGE [BEAM beam] [LAM nx rx vx ix] [BETA ny ry vy iy] [LIKE tem-
file]]
    [/PLOT [where]] [/SMOOTH fx fy]

This command builds a GILDAS table containing the following values
   Column 1   X offset (in radians)
   Column 2   Y Offset (in radians)
   Column 3   Weight of the data
   and
    Columns 4 to... : values of the specified Spectral channel(s)
   or
    Column 4: Result of the mathematical expression
from the current index of scans. IT IS UP TO THE USER to ENSURE that the
current index concerns a SINGLE SOURCE and a SINGLE LINE, observed  with
a  UNIQUE SPECTRAL RESOLUTION, with ALL spectra centered at the SAME VE-
LOCITY.  Option /IMAGE is used to grid the data  into  a  (2-d  or)  3-d
cube, option /PLOT to plot a map of channel-averaged data.

If  "Filename"  is not present the data is appended to the current table
(the one opened by the last GRID command). If "APPEND" is given  as  the
second argument the table Filename is opened and data is appended to it;
otherwise "NEW" may be specified to open a new table; this  is  the  de-
fault.

If  "MATH_EXPRESSION" is present in the command line, Column 4 will con-
tain the result of Math_expression applied for each spectrum.  This  is
particulary handy when one use the SIC variables and functions (see HELP
CUBE Functions, HELP LAS\SET VARIABLE and try  the  command  SIC\EXAMINE
/FUNCTION) in the Math_expression.
```

Option /CHANNEL can be used to limit the part of the spectra that will be included in the table (not used if you treat a Math_expression).

The resulting table can then be processed as a whole using the GILDAS (Grenoble Image and Line Data Analysis System) software. With /IMAGE a data cube is created from this table. GILDAS includes tasks to do the same job with more flexibility (GRID_SG and GRID_EXTEND), and transpose, reinterpolate, change the projection systems, make overlays with the IRAS point sources, contours and so on from data cubes. Call GRAPHIC at system level and then type EXPLAIN for more information on GILDAS tasks.

7.11.1 GRID /CHANNEL

GRID /CHANNEL [CHANNEL|VELO|FREQ|IMAGE] min max

Select the channels to be used for the tabulation and gridding process. This fixes the third dimension of the final data cube. Use CHANNEL (the default) to give min and max in channels numbers, or one of the other units: Velocity, Frequency, ir Image frequency.

7.11.2 GRID /IMAGE

GRID /IMAGE [BEAM beam] [LAM nx rx vx ix] [BETA ny ry vy iy] [LIKE tempfile]

Grid the tabulated data in a data cube (or a 2-d image if only one channel is processed, or a mathematical expression). The cube coordinate axes are normally defaulted from the position of the spectra, assuming that the angular resolution was 'beam' (in current angle units; a recommended default is taken for 30m telescope data); otherwise the LAMBDA and/or BETA axes may be forced by giving the number of pixels (nx,ny), the reference pixel (rx,ry), the coordinate value at the reference pixel (vx,vy) , and the pixel size (ix,iy, all in current angle units). Finally both coordinate axes may be taken from a template image (LIKE tempfile).

In the APPEND mode, none of the /IMAGE arguments should be given, the current image (from the previous GRID/IMAGE command) is incrementally recomputed, adding the new data. This is intended for a real time display while observing.

7.11.3 GRID /PLOT

GRID /PLOT [WHERE]

To be used only with /IMAGE. An image of the average of all the channels is plotted. WHERE plots GreG markers at the observed positions.

7.11.4 GRID /SMOOTH

```
GRID /SMOOTH fx [fy]
To be used only with /IMAGE. Smoothes the image by 'fx' and 'fy'.
```

7.12 ITERATE

```
ANALYSE\ITERATE
```

ITERATE tries the GRADIENT technique to improve the last results of the profile fit. Do not expect spectacular results if the absolute minimum is far away ! Typically, this command should be used if the previous fit did not converge, but seems close enough. If you are using the GAUSS method, be cautious not to change the X unit in the mean time.

7.13 KEEP

```
ANALYSE\KEEP
```

Save profile fit results in the output file. KEEP is equivalent to UPDATE, but only profile fit results are saved; KEEP is subject to the same restrictions as UPDATE.

7.14 Language

```
                    ANALYSE\ Language Summary
COMMENT          : Manipulates the COMMENT section of spectra
CUBE file [fnct]: Creates a 2 or 3-D data cube for processing by GILDAS.
DISPLAY          : Prints the results of the profile fit of R.
DIVIDE           : Divides R spectrum by T spectrum.
DRAW             : Calls the cursor or plot comments.
FILL start end   : Linearly interpolates the spectrum between start and end.
FFT              : Computes and edit Fourier Transform of R.
FIT [N]          : Plots the current fitted profile.
GAUSS            : Performs a profile fit.
GREG [name]      : Makes a TABLE or formatted file from current scan
GRID file [fnct]: Creates a data table or image for processing by GILDAS.
ITERATE          : Iterates the profile fit.
KEEP             : Saves Gaussian fit results in the output file.
LINES [N]        : Enters the initial values for the profile fit.
METHOD Arg       : Selects the line profile for fits.
MAP [M|W]        : Plots a map of spectra (or only their location).
MEMORIZE Arg     : Memorize the current observation.
MODEL Var        : Generates a CLASS spectrum from a 1D variable.
```

```
NOISE [S [NEW]] : Generates gaussian noise
PRINT [Arg]     : Prints values in a formatted file, or a table.
POPUP           : Zoom a spectrum from a STAMP or MAP plot.
RESIDUAL [n]    : Computes the residuals of the profile fit.
RESAMPLE Arg    : Resamples a spectrum on a specified grid
REDUCE          : Reduces a SKYDIP
RETRIEVE Arg    : Retrieve an observation from the memories.
SMOOTH [Arg...] : Smoothes the spectrum in R
STRIP File      : Creates an image for Position-Velocity plots.
STAMP           : Displays all spectra in index on one plot.
```

7.14.1 Language NEWS

```
   Version 4.1
 - Commands MODEL to generate spectra from SIC variables. 06-Nov-1989
 -  Commands MEMORIZE and RETRIEVE to temporarily save spectra.
   8-Apr-1988

Version 4.0
 - Support of method HFS in fits. 1-Feb-1988
 - Support of table format in command PRINT.  10-Jan-1988
 - Command REDUCE for skydips. 1-Dec-1987
 - Support for continuum in GAUSS, ITERATE, FIT, DISPLAY, PRINT and
   GREG. 1-Oct-1987
 - STRIP command now uses the  GILDAS format. 25-Aug-1986
 - Commands RESAMPLE and DIVIDE added.
 - TABLE format in command GREG. 1-Jul-1986
 - CUBE command for general processing (contour maps, velocity-position
   plots and so on) via the GILDAS software. 20-Jun-1986
```

7.15 LINES

```
ANALYSE\LINES [N] [/NOCURSOR] [/INPUT File_Name]
```

Enter the initial values for the profile fit. If argument "AUTO" is given instead of a number, then the number of lines found in the last read spectrum will be used for ITERATE command.

If N is not specified, then the last value for N is used. If N is 0, GAUSS will attempt to guess initial values for a single line by computing the moments of the spectrum. If N is not 0, LINES prompts for the initial values for line parameters. These values should be separated by spaces and can be entered as SIC expressions as follows:

```
     Code, Value,     Code, Value,     Code, Value,     Code, Value
  for GAUSS method
     -Temperature-     -Position-        -Width-
  for SHELL method
```

```
    -Area (K.MHz)-    -Position-      -Width (MHz)-    -Horn/Center-
  for NH3 method
    -Temperature-     -Position-      -Width (km/s)-     -Opacity-
```

The code is an integer interpreted as follows :

```
        0       adjustable parameter
        1       fixed parameter
        2       adjustable parameter (head of group)
        3       parameter fixed with respect to parameter coded 2 or 4
        4       fixed parameter (head of group)
```

Codes 2 3 and 4 are used to fit dependent lines (e.g. HCN, for which the displacements are 4.842 and -7.064 km/s, or -1.431 and 2.088 MHz, and line ratios 1:0.6:0.2). Codes 0 and 1 only are allowed for NH3 method. The value for a parameter with code 3 should be either the ratio to the head of group value (for both intensity and width), or the offset from head of group (for the position).

If the cursor is on, you will have to define the boundaries of the N lines. The program then computes the moments of the spectra between these boundaries as input values. You cannot change the codes in this way which is meaningfull ONLY FOR FREE and INDEPENDENT lines. This cursor selection is not used if you specify the /NOCURSOR option.

Guesses can be read from an input file using the /INPUT File_Name option. "Free" format is used to read in this input file. The first line must contain the number of lines, others are as the input at the keyboard.

Command LINES is not supported for method CONTINUUM.

7.16 MAP

```
    ANALYSE\MAP [Match|Where] [/CELL Size_X [Size_Y]] [/GRID] [/NOLABEL]
[/NUMBER]
```

Makes a map of spectra. Used in conjonction with the keyword W (Where), MAP will plot the location of spectra in the current index. Otherwise, MAP computes the offset limits for the spectra in the current index, automatically adjusts the size of a spectrum depending in the current box size, then draws the spectra at the position corresponding to the offsets, and finally draws the axes with labels. The only constraint of MAP in this mode is that the X and Y limits must be fixed so that the spectra share a common scale.

Argument MATCH can be given to force matching the X to Y ratio, which is otherwise adjusted so as to make the largest possible drawing. The options control presentation details.

The actual graphic output is delayed until the end of the index has been reached. This process may be slow on some inplementations and graphic terminals when the number of spectra is large. The MAP command can be interrupted by typing Control-C.

7.16.1 MAP /GRID

```
ANALYSE\MAP/GRID
```

A grid is plotted to separate the spectra.

7.16.2 MAP /CELL

```
ANALYSE\MAP/CELL Size_X [Size_Y]
```

If the option /CELL is given, the cell size of the grid is taken to be Size_X by Size_Y (default Size_X) current angular units. Spectra always fill the cells.

7.16.3 MAP /NOLABEL

```
ANALYSE\MAP/NOLABEL
```

If the option /NOLABEL is present, the coordinates are not written but ticks are still present. Tick can be suppressed by specifying a tick size of zero in command SET TICK.

7.16.4 MAP /NUMBER

```
ANALYSE\MAP/NUMBER
```

If the option /NUMBER is present, the observation number is written for each spectrum in the upper left corner.

7.17 MEMORIZE

```
ANALYSE\MEMORIZE [Name]
```

If a Name is specified, put a copy of the current observation in a "memory" called Name. The observation can later be retrieved using command RETRIEVE. If no Name is specified, list the currently defined memories. Up to ten memories can be defined at a time.

MEMORY Name/DELETE can be used to delete the "memory" area.

7.18 METHOD

```
ANALYSE\METHOD Arg [Parameters ...]
```

Select the line profile for fits. Five type of profiles are available.

- GAUSS Select gaussian profiles.
  Up to 5 Gaussian, dependent or independent can be fitted in a spectrum, according to command lines. Variables are Area Velocity and Width (FWHM).
- SHELL Horn-type profiles for Circumstellar Shells.
  Variables are 1) Area, 2) Position, 3) Full Width at zero level, and 4) Horn-to-center ratio (-1 for parabolic optically thick lines, 0 for flat-topped lines, Infinity for perfectly double peaked lines). Profiles are symmetric. Up to 5 dependent or independent lines may be fitted. Area units are K.MHz.
- NH3(1,1) or NH3(2,2) or NH3(3,3)
  Compute ammonia line profiles, with the assumptions of gaussian velocity distribution and equal excitation temperatures. Variables are 1) Main group opacity times (Excitation temperature minus Background temperature), 2) Velocity, 3) Width, and 4) Main group opacity. Up to 3 independent lines may be fitted, but it may be slow...
- HFS FileName for HyperFine Structure
  This is very similar to the NH3 method, but the coefficient to describe the hyperfine structure are read in the file FileName. The first line of this file must contain the number of hyperfine components (< 40). The other lines must contain, for each component, the velocity offset and the relative intensity. The parameters are the same as for NH3 method.
- CONTINUUM for Continuum drifts.
  Fit a single gaussian and a linear baseline at the same time, with automatic determination of the parameters. Uses beam-switched information for two-component fitting if required. The command METHOD CONTINUUM accept additional arguments, which can be used to customize the fit.

METHOD CONTINUUM [Width [Area_Ratio [Width_Ratio]]]
where

- Width is the width of the beam in arc seconds. By default, Width is a free parameter.
- Area_Ratio is the reference to main beam area ratio, used for beam-switch observing. By default, Area_Ratio is 1.
- Width_Ratio is the reference to main beam width ratio, used for beam-switch observing. By default, Width_Ratio is 1.

A * can be used instead of a numerical value for each parameter, and indicates that the parameter is free.

Note that in command LINES, one enters the peak line antenna temperature instead of the first variable in the guesses, except for SHELL method.

Command LINES has no effect for CONTINUUM.

CONTINUUM and GAUSS methods only are relevant for continuum data, while GAUSS, SHELL, NH3 and HFS methods can be used for spectroscopic data.

7.19 MODEL

```
ANALYSE\MODEL variable [X_variable]
```

Generates a CLASS spectrum from a 1D SIC variable. Note that the 1D variable can be (and frequently is) a subset of a higher dimensionality variable. This command is typically used either to generate test data, or more frequently to transfer data from a GILDAS image (or cube) to CLASS.

A second variable, if present, is interpreted as an associated axis scale for a presumably non linear frequency axis, as can occur for non resampled data from AOS backends (or optical spectra). Please note however that non linear axes are a largely untested feature in CLASS.

7.20 NOISE

```
ANALYSE\NOISE [S [NEW]]
```

Generates and draws a spectrum with only gaussian noise of rms S, to be compared with actual receiver noise. S defaults to the rms of the current spectrum. If argument NEW is present, the spectrum is not plotted but the R buffer is copied in T, and the generated noise put into the R buffer.

7.21 POPUP

```
  ANALYSE\POPUP Number
or
  ANALYSE\POPUP OffX OffY
```

POPUP Number will display a separate window containing a complete plot of an observation specified by number, if it has been previously displayed by a command STAMP or MAP.

POPUP OffX OffY will display a separate window containing a complete plot of an observation with offsets matching OffX and OffY, which has been previously displayed by a command MAP. If more than one observations have the same offsets, only the first one will be displayed.

In both cases, POPUP assumes that the index has not been modified since the last STAMP or MAP command.

POP with argument will active the cursor to select interactively th eobservation to be displayed. Point the cursor to its spectrum in the MAP or STAMP display and press P or click the mouse second button. Use the third button or E to exit the cursor mode. The first button or any key will display the offsets or observation number.

7.22 PRINT

```
ANALYSE\PRINT [Arg] [/OUTPUT File] [/TABLE File]
```

PRINT enables one to list some parameters in a GreG compatible format. Arg is an optional argument to precise what to print :

- FIT Results of profile fits in current method (default)
  The information written is (1) the observation number, (2,3) the offsets in current angle units, (4,5,...) fit results of the current method in the same order as in DISPLAY command. For the Gauss Method, this order is: (4) Line Area, (5) its incertainty (6) Position (7) incert. (8) Width (9) incert. (10) Intensity (11) Base RMS (12) Line RMS.

- AREA Integrated area computed by BASE
  The information written is (1) the observation number, (2,3) the offsets in current angle units, (4,5) Area and rms noise.
- AREA v1 v2 v3 ... Area in velocity ranges v1 to v2 and v2 to v3, ... etc.
  The information written is (1) the observation number, (2,3) the offsets in current angle units, (4,5,...) the areas.
- CHANNEL List list of channels (in the same format as the FOR-NEXT loop). Up to 15 channels at a time may be listed. The information written is (1) the observation number, (2,3) the offsets in current angle units, (4,5,...) the channels value.
- MOMENT m1 m2 ... Moments of the observations (area, mean velocity, width) in the intervals [m1,m2] [m3,m4] etc... Current units is used. The information written is (1) the observation number, (2,3) the offsets in current angle units, (4,5,6) area, position, width, (7,8,9) etc...

- POINTING Results of CONTINUUM method fits printed in a format adapted to pointing constants measurements. The information written is (1,2) The observation and scan number, (3) a code for scan direction, (4,5) Azimuth and Elevation in degrees, (6) Time in hours, (7,8) Collimation and error along drift direction, (9,10) the antenna number and station code, (11,12) Width and error, (13,14) Intensity and rms noise, (15) Source name.

- FLUX Results of CONTINUUM method fits printed in a format adapted to flux measurements. The information written is (1,2) The observation and scan number, (3) a code for scan direction, (4,5) Azimuth and Elevation in degrees, (6) Time in hours, (7,8) Position and error along drift direction, (9) the antenna number, (10,11) Width and error, (12,13) Intensity and rms noise, (14) Image gain ratio, (15,16) Signal and Image frequencies, (17) Source name, (18) Observing date.

/OUTPUT File Write the results in a formatted file named File instead of on the screen. The file is suited for later processing by GreG, in particular for contour maps.

/TABLE File Write the results in a table named File. This table can also be accessed by GreG, in a much faster (50 times) way than a formatted file. Tables can also handle many more columns than formatted files, and they are not limited in precision by the formatting. Mathematical operations can be done directly on the table columns. However the FIT results cannot be written to a table.

7.23 REDUCE

ANALYSE\REDUCE

This command reduces a SKYDIP observation, by fitting the sky emission using atmospheric parameters written in the observation, and displays the results : Water vapor content and Telescope Forward Efficiency

7.24 RESAMPLE

ANALYSE\RESAMPLE NX Xref Xval Xinc UNIT [shape] [width] [/NOFFT]

This command resamples a given spectrum on a specified grid. The arguments are:

NX the number of channels desired
Xref the new reference channel
Xval the value at the reference channel
Xinc the new channel separation
UNIT either VELOCITY or FREQUENCY to indicate the units used for Xval and Xinc.
SHAPE (optional) a keyword for the frequency response of the synthesized channels:
- TBOX means a rectangular function in the delay domain, as for unsmoothed correlator channels (sin(pi.x)/(pi.x) in the frequency domain)
- TPAR means a parabolic function in the delay domain, as for

smoothed correlator channels (smoothing function used at Plateau de Bure)
- FBOX means a rectangular function in the frequency domain (as for filterbank channels)
- FTRIANGLE means a triangular function in the frequency domain (as for filterbank channels, Hanning smoothed) By default the shape of the output channels is the same as the shape of the input channels.

WIDTH (optional) the width of the output channels in units of the channel separation. The default (and minimum value) is 1.

The resampling is done in the delay domain, and the data are Fourier transformed, unless the option /NOFFT is used. The Fourier transform is divided by the transform of the input channel shape, then extrapolated by zeroes (if interpolation in the frequency domain is required), multiplied by the transform of the desired channel shape, and finally transformed back to frequency domain.

The output channel separation usually needs to be slightly rounded (to enable use of the FFT, the input spectrum must contain a round number of output channels). To obtain an exact channel separation, the NOFFT option is necessary.

The WIDTH parameter is used to provide an oversampled output spectrum if needed.

The shape of the input channels is derived from the backend number read in the telescope name for IRAM-30m spectra: TBox for autocorrelator, FBOX for other (filter) backends. Otherwise, rectangular filter channels are assumed.

Resample may be used on on-the-fly scans too.

7.25 RESIDUAL

ANALYSE\RESIDUAL [n]

Compute and draw the residuals of the profile fit. Copies R into T, then subtract the line profile number N from the spectrum in R. If N is 0, the sum of all components is subtracted.

7.26 RETRIEVE

ANALYSE\RETRIEVE Name

Retrieve the observation in memory "Name" and copy it in the R observation.

7.27 SMOOTH

```
ANALYSE\SMOOTH [Arg...]
```

SMOOTH copies R into T, then degrades the frequency/velocity resolution of R. The arguments are used to specify the method :

- HANNING HANNING smoothing is used (default), in which the new spectrum has twice less channels and twice less resolution.
- AUTO uses a cross-validation smoothing with automatic choice of smoothing parameter. To be used with caution. Noise in adjacent channels must be independant, otherwise the algorithm flatly refuses to do any smoothing at all (but needs a long computing time to realise that...).
- GAUSS Width convolves (by multiplication in the Fourier plane) by a gaussian of specified width in current units.
- BOX Ncan averages Ncan adjacent channels to produce a spectrum with Ncan less resolution and channels.
- NOISE Flux Nc for each channel, sums at most Nc neighbouring channels until a total flux Flux is reached in the sum. Then attributes the average value (sum divided by number of channels added) to the channel. This smoothing is non-uniform, strictly positive, and has an obvious tendency to produce wings...

7.28 STAMP

```
ANALYSE\STAMP NX [NY] [/NUMBER]
```

This command is intended to have a quick look at all the observations in the index. It makes a plot of all the observations in the index, arranged in a "map" of NX by NY observations. (If the index contains many observations, they will be very small, hence the name of the command). The option /NUMBER writes the number of each observation in the upper left part of its box. The command does not require that X and Y scales be fixed.

Command POPUP allows a selective zooming of any observation displayed by command STAMP.

7.29 STRIP

```
ANALYSE\STRIP File
```

STRIP creates a GILDAS 2-D image which can be used later by GILDAS to produce Velocity-Position plots using command RGMAP. STRIP works on the current index, and checks that it defines a true strip along one of the main directions ; that means you should have build the current index with FIND/OFFSET Value * or FIND/OFFSET * Value. The current X unit is

used. The name of the output file must be given.

If you want to produce a strip not along one of the main directions, you will have to change the offsets of your spectra using command MODIFY OFFSETS, and writing the modified spectra in a new output file.

STRIP can also produce a GILDAS 2-D image from a set of parallel continuum drifts. The drifts must have the same steps, and be regularly spaced. The index must define a coherent map.

8 Error Messages and Recovery Procedures

Commands produce messages, either to signal their actions, or whenever they fail to execute properly to indicate the reason of the failure. The syntax of the message if the following `C-Facility, Error message text`
where C is a one letter severity code, Facility is the name of the command or of a subroutine called by the command, and "Error message text" is a short explanation.

Messages can be divided into four categories according to the severity code

- `S` (Success) or `I` (Information) messages indicate normal successful operations.
- `W` (Warning) messages indicate that the operation completed, but that the result may not be significant.
- `E` (Error) messages indicate a failure to complete the command. The result of the command is undefined, but these messages are usually signaled when checking the input parameters before executing the command. In particular, the Input and Output files are never affected by these errors.
- `F` (Fatal) messages indicate severe errors which happened while a command is executing. They may have left the input or output files in an incoherent state. These errors are typically hardware errors.

8.1 Notifying the Programmer or System Manager

Some recovery procedures ask you to submit an SPR (Software Performance Report). This means to signal the error to either the CLASS authors, the local contact person, or the system manager. Signalling the error does not mean simply just telling "Gee, I typed ABRACADABRA and the program crashed", it implies at least giving a copy of the `CLASS.LOG` and `CLASS.MES` files, and if requested, of the data files used when the error occured. The utilities are too complex to allow any action if these files are not given.

Please remember that if you do not signal errors, they will (at best) never be corrected. At worst, hardware errors which would have been recoverable when signalled in time may become completely unrecoverable.

8.2 Recovering Corrupted Data Files

First, let us express our most sincere hopes that you will never need to read this section... Due to hardware (or even software but this seldom happens) problems data files may become corrupted. Although this is exceptional, it can be a dramatic circumstance. The structure of the data files is fairly complex, and missing information in the index may prevent reading otherwise valid spectra in the data file. If you suspect a file may have been corrupted, first verify it by the following commands :

```
LAS> FILE IN Dubious_file_name
LAS> SET DEFAULT
LAS> FIND/ALL
LAS> FOR I 1 TO FOUND+1
LAS> GET NEXT
LAS> NEXT
```

If this series of command ends with the message
'E-GET, End of current index'
everything is correct.

If not the file is corrupted and you may try to recover it partly. If the FIND command executes correctly, drop (by command DROP) from the index all the scans which cannot be read correctly (these have been lost forever). Then executes the following commands

```
LAS> FILE OUT Recovered_File_Name NEW
LAS> GET FIRST
LAS> WRITE
LAS> FOR I 1 TO FOUND
LAS> GET NEXT
LAS> WRITE
LAS> NEXT
```

If the FIND command does not execute correctly, try a LIST IN command to get the scan and version numbers. If it works or if you already have such a list, you may try to get each scan individually by specifying the scan and version numbers, and write them to another output file. In this way you might be able to recover some spectra, but it is not guaranteed.

If nothing like this works and if your file contained *vital* information you cannot reconstruct in any way, you may consider sending it to the CLASS authors which may be able to do something more. To our knowledge, such a desperate case actually happened only once in the early times of CLASS (at that time named **LAS**), to the authors who spent about 3 hours to find a way to repair the file, but managed to do it. Do not forget to supply a complete log of the errors (CLASS.LOG and CLASS.MES files).

8.3 LAS and ANALYSE Messages and Recovery Procedures

The messages not present in this list should come from the SIC monitor itself. Please refer to the SIC documentation for these.

8.3.1 A and B

E-ACCUMULATE, Coordinate systems are not compatible
: LAS, SUM or ACCUMULATE commands. Two observations had different coordinate systems and could not be added.
User action : Convert them to a common coordinate system, or use the SET NOMATCH command to turn off position checking.

E-ACCUMULATE, Different calibrations
: LAS, SUM or ACCUMULATE commands. Two observations were done with different major calibration parameters (Beam Efficiency and possibly Forward Efficiency) and could not be added.
User action : May be they were not on the same temperature scale. The weighting (and values) is incorrect in such a case. Modify the corresponding calibration parameter and retry. You may bypass this calibration checking by using command SET CAL OFF.

W-ACCUMULATE, Different reference channels R: <Real> T: <Real>
: LAS, SUM or ACCUMULATE commands. Two spectra were added with the ALIGN mode CHANNELS, but had different reference channels, and hence possibly different velocity or

frequency scales.
User action : Check if these spectra were to be added. If you want to add them and preserve a common velocity (or frequency) scale, use the `ALIGN` mode `VELOCITY` (resp. `FREQUENCY`).

`W-ACCUMULATE, Different resolution R: <Real> T: <Real>`
`LAS, SUM or ACCUMULATE` commands. Two spectra were added with the `ALIGN` mode `CHANNELS`, but had different resolutions and hence possibly different velocity or frequency scales.
User action : Check if these spectra were to be added. If you want to add them and preserve a common velocity (or frequency) scale, use the `ALIGN` mode `VELOCITY` (resp. `FREQUENCY`).

`W-ACCUMULATE, Different rest frequencies R: <Real> T: <Real>`
`LAS, SUM or ACCUMULATE` commands. Two spectra were added with the `ALIGN` mode `CHANNELS` or `VELOCITY`, but had different rest frequencies and hence possibly different velocity or frequency scales.
User action : Check if these spectra were to be added. If these spectra correspond to the same molecular line, use the `MODIFY FREQUENCY` to reset a common velocity scale and the `ALIGN` mode `VELOCITY` before adding them.

`W-ACCUMULATE, Different velocities R: <Real> T: <Real>`
`LAS, SUM or ACCUMULATE` commands. Two spectra were added with the `ALIGN` mode `CHANNELS` or `FREQUENCY`, but had different velocities and hence possibly different velocity or frequency scales.
User action : Check if these spectra were to be added. If these spectra correspond to the same object, use the `MODIFY VELOCITY` to reset a common rest frequency scale and the `ALIGN` mode `FREQUENCY` before adding them.

`W-ACCUMULATE, Drift not aligned in position`
`LAS, SUM or ACCUMULATE` commands. Two continuum drifts were added but are not coincident in position.
User action : Select `POSITION` alignment using command `SET`, and retry.

`W-ACCUMULATE, Origin table overflow`
`LAS, SUM or ACCUMULATE` commands. Too many spectra were added and the list of original scan numbers is full. Scan numbers of any other added spectrum will be lost.
User action : none. The spectrum still contains the sum the all desired spectra, only the list is wrong.

`E-ACCUMULATE, Positions are not compatible`
`LAS, SUM or ACCUMULATE` commands. Two spectra have different positions and have not been added.
User action : If you want to add them, use the `SET NOMATCH` command before.

`E-ACCUMULATE, <String> ranges do not intersect`
`LAS, SUM and ACCUMULATE` commands. Two spectra were to be added with the `ALIGN` mode specified by `<String>` and `INTERSECT` mode, but do not overlap.
User action : what do you want to add ?...

`W-ACCUMULATE, Spectra not aligned in sky frequency R: <Real> T: <Real>`
`LAS, SUM and ACCUMULATE` commands. Two spectra were added with `ALIGN` mode `CHANNELS`, but have a different sky frequency scale.

User action : use the `SET ALIGN FREQUENCY` if you want to preserve a common frequency scale.

`E-ACCUMULATE, Too many channels requested`
`LAS, SUM and ACCUMULATE` commands. Two spectra were added with `ALIGN` mode `COMPOSITE`, but the final spectrum requires more channels than reserved in the program.
User action : most likely you are trying to make a composite spectrum with two unrelated frequency bands, such as CO and HCN... Note also that making composite spectra is not necessary to plot a few GHz from a spectral survey.

`E-ADD, Cannot add Continuum with Line data.`
`LAS, ACCUMULATE` command: The two observations are not of the same kind. `SUM` command: The index contains observations of different kinds.
User action : It is time to get some rest and a cup of coffee...

`W-ADD, Gives sum of spectra for EQUAL weight`
`LAS, ACCUMULATE` command: This command gives the sum of the spectra, not the average.
User action : This behaviour is consistent with that of `SUM` which gives the average over the index. Using several ACCUMULATE commands, and then dividing the final result by the number of added observations yields the same result as `SUM`.

`W-ADD, Still experimental for Continuum`
`LAS, ACCUMULATE and SUM` commands.
User action : Poor guinea pig... Do not worry too much, it usually works fine, but we would appreciate receiving detailed comments if you find something bizarre.

`E-BASE, Insufficient memory for work space`
`LAS, BASE` command. The work space to compute the best fit polynomial could not be dynamically allocated because the program exhausted its virtual memory quota.
User action : i) clear any plot and restart. ii) run CLASS in a single process (not from a sub-process). iii) Ask your system manager to extend the virtual memory limit.

`E-BASE, No line window`
`LAS, BASE` command. No line window has been defined.
User action : define one or more windows using the `SET WINDOW` command.

`E-BASE, No valid baseline`
`LAS, BASE` command : the windows covers the full spectrum and leave no channels to compute the baseline
User action : define windows which do not cover the spectrum, or (if possible) compute the baseline from another backend and use the command `BASE LAST` to apply it to this spectrum.

`E-BASE, Sinus not converged`
`LAS, BASE` command with `SINUS` method. The minimization routine failed to converge to a valid solution.
User action : The solution may still be sensible, look at it. If not, retry with different guesses.

8.3.2 C

`E-CBUF, Write error block <Integer>`
: `LAS`, Input/Output routines : an error occurred during a write operation to the output file. This may be due to a large number of reasons (for example the disk may be off line, write locked or damaged) but is more or less fatal.
User action : See "Recovering Corrupted Data Files"

`E-CHECK, Flag > 4`
: `ANALYSE, LINES` command : the input values contained an invalid flag.
User action : reenter valid values.

`W-CHECK, Line <Integer> alone in a <String> Group`
: `ANALYSE, LINES` command. Input values were entered for dependent lines, but no lines depend on the one specified by `<Integer>`.
User action : although this is just a warning, it is likely that you made a typing mistake. Verify your input values and codes.

`E-CHECK, No Independent <String>`
: `ANALYSE, LINES` command : input values were entered for dependent lines, but there is no independent line in `VELOCITY, AREA or WIDTH` (as specified by `<String>`.
User action : correct typing error.

`E-CHECK, Several groups in <String>`
: `ANALYSE, LINES` command : input values were entered for dependent lines, but all lines must be dependent of a single one in this case. This may be due to a typing error, but also to a problem which cannot be solved by the current minimization routine.
User action : Correct typing error if any.

`E-COBS, Observation not open for write nor modify`
: `LAS`, Input/Output routine : an observation is being closed after or write or modify operation, but has never been opened for such operation.
User action : Submit an SPR. The output file may have been corrupted, see "Recovering Corrupted Data Files"

`W-CONVERT, Earth velocity not converted`
: `LAS, GET` command or other observation reading routine. The velocity of the source was relative to Earth, and has not been converted to the current velocity system.
User action : None, but remember the velocity is wrong...

`W-CONVERT, Velocity type was unknown, Set to <String>`
: `LAS, GET` command or other observation reading routine. The velocity type was undefined, and has been (arbitrarily) declared as `<String>` (LSR, Helio, etc...), but no velocity conversion was done.

`W-CONVER, label too long, truncated`
: `LAS`, plotting routines. A label was too long to be drawn.
User action : None. Submit an SPR if you wish, with the label of course.

`F-COX, Output file not opened`
: `LAS`, Input/Output routine : an operation was requested on the output file, but no such file

has been opened.
User action : open the output file, and resubmit the command.

`F-COX, Write error block <Integer>`
`LAS`, Input/Output routines : an error occurred during a write operation to the output file. This may be due to a large number of reasons (for example the disk may be off line, write locked or damaged) but is more or less fatal.
User action : Notify your system manager. The output file may have been corrupted, see "Recovering Corrupted Data Files"

`E-CUBE, Cannot create output image`
`ANALYSE, CUBE` command. The output image could not be created. This message is followed by a message `'E-GDF_CRIMAG, ....'` which gives the reason of the failure. The most likely reason is an exceeded quota (either disk or virtual memory)
User action : if this is due to a disk quota, delete unwanted files and retry. If this is due to a virtual memory quota, clear any plot, run the program without parallel sub-processes, and if none of these solution works, ask your system manager.

`E-CUBE, Cannot write output image header`
`ANALYSE, CUBE` command. The output image header could not be written. This error is due to an hardware error on the disk.
User action : Notify the system manager, and retry again later.

`I-CUBE, Check tolerance (SET MATCH) please`
`ANALYSE, CUBE` command. The output cube is too large, possibly because of roundoff errors in the offsets.
User action : increase the `SET MATCH` to overcome roundoff errors.

`E-CUBE, Continuum data not supported`
`ANALYSE, CUBE` Command. Spectral line data is required to make a data cube. Use command `STRIP` to build a continuum map from a set of scans.

`E-CUBE, Cube too large, <Integer> by <Integer>`
`ANALYSE, CUBE` command. The output cube is too large, most likely because of roundoff errors in the offsets.
User action : see "`I-CUBE, Check tolerance (SET MATCH) please`"

`W-CUBE, Incoherent axis scale for scan <Integer>`
`ANALYSE, CUBE` command. One of the spectra in the index has an axis scale which does not match the cube definition. The spectrum has been written to the cube nevertheless, and the cube command proceeds.
User action : May be this spectrum should not be placed in the cube, so drop it from the current index and rebuild the cube. If spectra have been obtained under slightly different conditions (for example with different sidebands) you must bring them on a common axis scale (use RESAMPLE) before combining them in a cube.

`E-CUBE, Incoherent number of channels for scan <Integer>`
`ANALYSE, CUBE` command. One of the spectra has a number of channels different from the cube definition. The spectrum is not written to the cube, and the cube command aborts.
User action : drop this spectrum from the current index and restart command.

`E-CUBE, Incoherent offset type for scan <Integer>`
`ANALYSE, CUBE` command. The coordinate system of the scan differs from the one of the cube. The scan is not written, and command `CUBE` aborts.
User action : drop the scan from the index, and restart command.

`E-CUBE, Index is empty`
`ANALYSE, CUBE` command. A cube is being build from nothing...
User action : define your current index before by using command `FIND`.

`E-CUBE, Missing file name`
`ANALYSE, CUBE` command. The output cube name has not been specified
User action : specify file name

`E-CUBE, No input file connected`
`ANALYSE, CUBE` command. No input file is opened.
User action : open input file and define the current index before using command `CUBE`.

8.3.3 D and E

`W-DEVICE, No device active`
`LAS, DEVICE` command. No graphic device is currently connected.
User action : it is nevertheless possible to produce plots and hardcopies without a graphic device. A graphic device can be connected later at any time to visualize the plots.

`W-DISPLAY, No Fit for scan <Integer>`
`ANALYSE, DISPLAY` command. There is no fit (of the current method) for this scan.

`E-DIVIDE, Spectra do not have the same velocity scale`
`ANALYSE, DIVIDE` command. The R and T spectra must have the same number of channels and a common velocity scale to divide T by R.
User action : Use command `RESAMPLE` to rescale both spectra on the same grid.

`E-DIVIDE, Spectra have different number of channels`
`ANALYSE, DIVIDE` command. The R and T spectra must have the same number of channels and a common velocity scale to divide T by R.
User action : Use command `RESAMPLE` to rescale both spectra on the same grid.

`E-DRAW, Invalid Centering`
`ANALYSE, DRAW` command with argument `TEXT`. The centering code for the character string is invalid.
User action : valid code range is 1-9.

`E-DRAW, No cursor available`
`ANALYSE, DRAW` command. The `DRAW` command is used without a completely explicit form, but no cursor is available to supply the missing parameters, either because no graphic device is connected, or because the current graphic device has no cursor.
User action : supply all missing parameters to the command, or connect a graphic device with cursor.

`E-DRAW FILL, Channel is not blanked`
`ANALYSE, DRAW` command with argument `FILL`. The channel to be interpolated already

contains a valid value. This may be because you typed F by mistake in the cursor mode.
User action : If you really want to interpolate this channel, kill it before by using command `DRAW KILL`.

`E-DRAW FILL, Channel outside spectrum`
`ANALYSE, DRAW` command with argument `FILL`. The channel to be interpolated is outside the spectrum boundaries. This may be because you typed F by mistake in the cursor mode, or because the cursor lies outside the spectrum boundaries.
User action : Move the cursor to the channel to be interpolated before typing the code F.

`E-DRAW KILL, Channel outside spectrum`
`ANALYSE, DRAW` command with argument `KILL`. The channel does not exist in this spectrum, probably because the graphic cursor does not point where you would like.
User action : move graphic cursor towards the channel to kill.

`E-DROP, Current index is empty`
`LAS, DROP` command. The current index is empty, so that no scan could be dropped from it...

`I-DROP, <Integer>;<Integer> not in current index`
`LAS, DROP` command. The scan to be dropped is not in the current index. This is most likely due to a typing error.

`E-DROP, End of current index encountered`
`LAS, DROP` command with argument `NEXT`. The end of the index has been reached.

`Error encoding <Real>`
Sexagesimal label routines. The value to be encoded in sexagesimal form is out of valid range.
User action : Submit an SPR.

`I-EIX, New data present`
`LAS, NEW_DATA or FIND` command with argument `NEW`. New data has been found in the input file, execution resumes.

`W-EIX, No input file opened`
`LAS, NEW_DATA or FIND` command with argument `NEW`. No new data can be found in nothing...

`F-EIX, Read error file <String>`
`LAS, NEW_DATA or FIND` command with argument `NEW`. A read error occured while reading the header block of the input file when looking for new data.
User action : Retry, this is usually due to an interlock problem. If the error persists, check that no more than 2 processes are accessing the specified file.

`I-EIX, Switching to <String> Mode`
`LAS, NEW_DATA` command. The new data found is of a different type than the current one. Switching to the new type occured.
User action : Remember not all actions are possible on every type of observations.

`I-EIX, Waiting loop aborted by <^C>`
`LAS, NEW_DATA or FIND` command with argument `NEW`. The wait has been interrupted by user action. No new data was found.

8.3.4 F

`E-FILE, IN OUT or BOTH please`
`LAS, FILE` command. The first argument is incorrect (typing error).

`E-FILE, No default is provided for the file name`
`LAS, FILE` command. The second argument (file name) is missing.

`I-FIND, <Integer> observations found`
`LAS, FIND` command, information message.

`E-FIND, Nothing found`
`LAS, FIND` command, information message. Variable `FOUND` is reset to 0.
User action : if you believe something should have been found, check the various selection criteria. One is probably different from what you had in mind.

`I-FIND, <Integer> observations in index`
`LAS, FIND` command with argument `APPEND`, information message. The index has been appended and compressed to avoid duplicate entries to a given scan.

`W-FITCONT, Solution not converged`
`W-FITGAUSS, Solution not converged`
`W-FITNH3, Solution not converged`
`W-FITSHELL, Solution not converged`
`ANALYSE, GAUSS` command. The minimization failed to converge to a valid solution.
User action : Look at the results using command `FIT`. If they seem good enough, use command `ITERATE` until convergence is reached. If the warning persists, may be the problem is badly defined and the solution unreliable.

`E-FITNH3, Unit X must be Velocity`
`LAS, GAUSS` command for method `NH3`. The X unit must be velocity to fit NH_3 hyperfine structure.
User action : use command `SET UNIT X V`, reset new guesses using command `LINES` and retry.

`E-FITSHELL, No baseline removed`
`ANALYSE, GAUSS` command, `SHELL` method. A base line must have been removed before fitting.
User action : use the `BASE` command to remove a baseline, and retry.

`E-FITSHELL, Unit X must be Frequency`
`ANALYSE, GAUSS` command, `SHELL` method. "Circumstellar shells" like profile fitting is only possible in Frequency units.
User action : use command `SET UNIT X F`, check the initial guesses and retry.

`F-FIX, Current index is full`
`LAS, FIND` command, usually with `APPEND` argument. Too many spectra match the current selection criteria, and the index is too large. Some spectra have been omitted from the index.
User action : Restrict the search by adding more severe selection criteria. Note in any case that you will have troubles processing the connected input file because it contains too many spectra. It would be wise to split this file in smaller ones, according to selected criteria.

`E-FIX, More than <Integer> observations found`
: `LAS, GET` or other commands involving index finding operations, such as `DROP`. There are too many scans matching the requested criteria. Most likely, this error is due to an incoherent input file which contains twice the same scan (number AND version).
User action : It is possible to get rid of the duplicate scans by the following recipe. i) define a new output file to receive the valid spectra. ii) use a `FIND/ALL` without any selection criterium. iii) use the `GET NEXT` command to retrieve scans. iv) use the `WRITE` command for all scans but the duplicate scans that you should either skip or write under a new scan number (`WRITE Number` command). The new output file should be correct. If not see "Recovering Corrupted Data Files".

`E-FIX, Read error`
: `LAS, FIND` or other commands involving index finding operations. This may be due to a lack of input file (not all commands are explicitly protected against this specific mistake), but also to hardware problems.
User action : after checking for the input file, retry the operation. If the error persists, submit an SPR, and see "Recovering Corrupted Data Files"

`W-FOLD, Cannot fold a single phased spectrum`
: `LAS, FOLD` command. The current spectrum was not observed in frequency switching mode, so that it cannot be folded.

`E-FOLD, No channels left in spectrum`
: `LAS, FOLD` command. The align mode is `INTERSECT`, and the frequency switch is so large that no channel overlap from one phase to the others.
User action : Set the align mode to `COMPOSITE`, and redo the command.

`E-FSHELL, Wrong Arguments : <Real>`
: `ANALYSE, GAUSS` command with method `SHELL`. The minimization routines went into a forbidden region for the parameters. The minimization aborts.
User action : retry the fit with a different choice of input parameters.

8.3.5 G through I

`E-GAUSS, No baseline removed`
: `ANALYSE, GAUSS` command with method `GAUSS`. Minimization requires that a baseline be removed before.
User action : remove a baseline with command `BASE` and retry the fit.

`E-GET, End of Current Index Encountered`
: `LAS, GET` command with argument `NEXT`. There are no more scans in the index.

`I-GET, Entry <Integer> Found <Integer>;<Integer>`
: `LAS, GET` command, information message.

`E-GET, Index is empty`
: `LAS, GET` command with argument `FIRST`. There is no first observation.

`E-GET, No input file connected`
: `LAS, GET` command. No input file is opened, so that the requested scan cannot be read...

`E-GET, Observation <Integer>;<Integer> not Found`
: `LAS, GET` command. The specified scan was not found in the input file. The version number (second `<Integer>`) is output only if it was specified in the command.
User action : check for typing errors.

`I-GREG, Creating Formatted <String>`
: `LAS, GREG` command. This information message indicates that the output file `<String>` is formatted.

`I-GREG, Creating Table <String>`
: `LAS, GREG` command. This information message indicates that the output file `<String>` is a GILDAS Table. Note that this is the default.

`E-GREG, Cannot open <String>`
: `LAS, GREG` command. The output file (formatted or table) could not be created. This message is followed by more precise information about the failure. Most likely failures are: invalid name, exceeded quota, privilege violation.
User action : it depends on the precise reason. Purge and delete any unwanted file if a quota problem is the basic cause and retry.

`E-HARD, Conflicting options`
: `LAS, HARDCOPY` command. Two conflicting options (e.g. `/PORTRAIT` and `/LANDSCAPE`) were given
User action : remove the unwanted option

`E-HARD, Invalid plot mode <String>`
: `LAS, HARDCOPY` command. The scaling mode is invalid. Refer to the `HELP HARDCOPY` for valid scaling modes.

`E-HARD, Non standard plot page, Specify option please`
: `LAS, HARDCOPY` command. The plot page size does not match the default values, so that the orientation for hardcopy must be specified (`/LANDSCAPE` or `/PORTRAIT`).

`S-HARD, <String>/<String> created`
: `LAS, HARDCOPY` command with argument, information message. The first string indicate the vector file name, and the second the orientation.

`S-HARD, Plot request successfully enqueued on GAG_PLOTTER`
: `LAS, HARDCOPY` command with option `/PLOT`.

`I-INIT, <String> initialized`
: `LAS, FILE OUT` command with third argument `NEW` specified, information message. A new empty output file has been initialized.

`E-INIT, Open error file <String>`
: `LAS, FILE OUT` command with third argument `NEW` specified. The output file could not be created. This message is followed by more explanations about the precise reason (exceeded quota, privilege violation, invalid file name, etc...)
User action : respond according to the precise reason.

`E-INIT, Write error file <String>`
: `LAS, FILE OUT` command with third argument `NEW` specified, information message. The

output file was created, but could not be initialized properly. This is most likely due to a disk hardware error.
User action : Notify the system manager and retry.

`I-INPUT, <String> successfully opened`
`LAS, FILE IN` command. Information message : the specified input file has been opened.

`I-INPUT, <String> is reopened`
`LAS, FILE IN` command. This message appears whenever a new input file could not be opened, to indicate which file is still currently opened.

`I-INPUT, No input file opened`
`LAS, FILE IN` command. This message appears whenever a new input file could not be opened, and no input file was previously opened.

`F-INPUT, Non standard file <String>`
`LAS, FILE IN` command. The specified file could be opened, but is not a valid CLASS file.
User action : Check the file name.

`F-INPUT, Open error file <String>`
`LAS, FILE IN` command. The specified input file could not be opened. The message is followed by more details about the reason of the open failure.
User action : verify the file name you typed, and retry.

`F-INPUT, Read error file <String>`
`LAS, FILE IN` command. The specified file could be opened, but the first block could not be read. This is most likely because the file is not a standard CLASS file, but anything else. If you are sure that the file is a valid CLASS file, this indicates a hardware problem.
User action : Check the file name or notify the system manager accordingly.

`E-ITERATE, Bad input parameters`
`ANALYSE, ITERATE` command. The input parameters are non valid. This may happen because you have changed the X unit or the R spectrum till the last fit.
User action : verify X unit and change the input parameters, or use the `GAUSS` command.

8.3.6 K and L

`E-KEEP, Input file must equal output file`
`ANALYSE, KEEP` command. The `KEEP` command does an update of the spectrum, and the input and output files must be identical for this.
User action : open the input file as the output file and retry, or use the `WRITE` command.

`I-KEEP, Observation <Integer>;<Integer> successfully updated`
`ANALYSE, KEEP` command, information message.

`E-KSHELL, Wrong Arguments : <Real>`
`ANALYSE, GAUSS` or `FIT` commands with method `SHELL`. The minimization algorithm produced an invalid solution.
User action : This also means that the fit is not converged. Restart it with different input parameters.

`E-LAS, <String> Not yet implemented`
`LAS or ANALYSE` languages, You are trying to use an undocumented, unimplemented command.
User action : Don't

`W-LAS, Save file could not be opened`
`LAS, SAVE` command. The output file could not be opened, may be because of an exceeded disk quota, or invalid file name.
User action : check the disk quota (and space) and the file name and retry.

`E-LINES, Error opening guess file <String>`
`ANALYSE, LINES` command. The input file for guesses could not be opened. This message is followed by more details about the precise reason of the open failure. React accordingly.

`E-LINES, Error reading guesses from <String>`
`ANALYSE, LINES` command. The input file specified for guesses contains invalid data.
User action : check the input file format.

`E-LINES, Invalid number of lines`
`ANALYSE, LINES` command. The user specified either a negative number of lines or too many lines.

`E-LINES, Not implemented for Continuum method.`
`ANALYSE, LINES` command has no action for Continuum method.

`W-LINES, Null area found, use manual mode`
`ANALYSE, LINES` command. The area found in cursor mode is zero, may be because you forgot to move cursor between the two edges...
User action : Try to give slightly different boundaries with the cursor, or use the manual mode (`/NOCURSOR` or `SET CURSOR OFF`) as suggested.

`W-LINES, Use of cursor with dependant or fixed lines not supported`
`ANALYSE, LINES` command. The user is using the cursor to specify input values, but the flags indicate that some parameters are fixed or dependent.
User action : It is not forbidden to use the cursor in such a case, but the user should realized that the guesses will always be those found by the cursor routine, superseding any previous value (unless a slash `/` is typed in the cursor routine).

`I-LISTE, Input file is empty`
`I-LISTE, Input file contains`
`I-LISTE, Output file is empty`
`I-LISTE, Output file contains`
`I-LISTE, Current index is empty`
`I-LISTE, Current index :`
`LAS, LIST` command, information messages

`I-LISTE, List on file <String>`
`LAS, LIST` command, information message specifying to which file the listing has been sent.

8.3.7 M

`E-MAP, Continuum data not supported.`
`ANALYSE, MAP` command. The current index contains continuum drifts, instead of spectra. `MAP` only works on spectral line data.

`E-MAP, Incoherent offset type for scan <Integer>`
`ANALYSE, MAP` command. The current index contains offsets in different coordinate systems.
User action : drop the faulty spectrum from the index.

`E-MAP, Index is empty`
`ANALYSE, MAP` command. No map can be plotted.
User action : Define the current index using the `FIND` command and retry.

`E-MAP, Map too large, <Integer> by <Integer>`
`I-MAP, Check tolerance (SET MATCH) please`
`ANALYSE, MAP` command. The number of cells in the map defined by the index is too large, so that the spectra would be too small. This may be due to rounding errors in the offsets which lead to an anomalously small cell size.
User action : increase the tolerance in `SET MATCH` by a factor 2. If the error persists, give explicitly the cell size (`/CELL` option) If this does not help either, verify that the index does correspond to a single source.

`E-MAP, Mode X and Y must be fixed`
`ANALYSE, MAP` command. The X and/or Y axis has an automatic scaling enable, and must have a fixed scale.
User action : Specify a fixed scale for all spectra using the `SET MODE` command, and retry the `MAP` command.

`E-MAP, No input file connected`
`ANALYSE, MAP` command. No input file is opened.
User action : Open the input file, define the current index using the `FIND` command and retry the `MAP` command.

`E-METHOD, Cannot open HFS description file.`
`E-METHOD, Error reading HFS description file.`
`ANALYSE, METHOD` command with argument `HFS`. The Hyperfine structure of the line to be fitted could not be read from the specified file.
User action : Check the file name or content for proper format. See `HELP METHOD`.

`E-METHOD, Too many HFS components`
`ANALYSE, METHOD` command with argument `HFS`. The hyperfine structure specified has too many components.
User action : Relinking the program is required to increase this parameter. Contact the authors and supply the number of components required. You may solve adequately your problem by suppressing from the description the weakest components if they are negligible.

`E-MOBS, Observation to be modified was not found`
`LAS`, Input/Output routines, `UPDATE` or `KEEP` commands. The user attempted to modify an unexistant observation.

`E-MOBS, Non-standard obs. block <Integer>`
`LAS`, Input/Output routines, `UPDATE` or `KEEP` commands. The input/output file contains an invalid data block because it has been corrupted.
User action : see "Recovering Corrupted Data Files", and notify your system manager because this is usually due to hardware errors.

`I-MODIFY, Beam efficiency set to <Real>`
`ANALYSE, MODIFY` command. The beam efficiency value was previously undefined and has been set to the specified value. No scaling occured.

`E-MODIFY, Invalid beam efficiency <Real>`
`ANALYSE, MODIFY` command. The beam efficiency value must be between 0 and the forward efficiency (less than 1).

`E-MODIFY, MODIFY POSITION not supported for this type of projection ANALYSE, MODIFY` command. Offsets relative to the new central position cannot be computed in the present coordinate or projection system
User action : Submit an SPR, specifying the projection used. We will try to implement it properly in future releases.

`I-MODIFY, Offsets set to <String>`
`ANALYSE, MODIFY OFFSETS` command. Information message.

`I-MODIFY, Reference channel set to <Real>`
`ANALYSE, MODIFY RECENTER` command, information message.

`I-MODIFY, Image frequency set to <Real>`
`ANALYSE, MODIFY IMAGE` command, information message.

`W-MODIFY, Unknown projection type, Radio projection assumed`
`ANALYSE, MODIFY` command. Offsets relative to the new central position were in a unknown projection system.

`F-MOX, Output file not opened`
`LAS`, Input/Output routines. A `WRITE` operation was attempted before any output file was opened.
User action : open output file and retry the aborted command.

`F-MOX, Read error block <Integer>`
`LAS`, Input/Output routines. A `WRITE` operation failed because the output file index could not be read correctly.
User action : see "Recovering Corrupted Data Files" and notify your system manager.

`F-MOX, Write error block <Integer>`
`LAS`, Input/Output routines. A `WRITE` operation failed because the output file index could not be written correctly.
User action : see "Recovering Corrupted Data Files" and notify your system manager.

`E-MOX, Wrong index address <Integer>`
`LAS`, Input/Output routines. A `WRITE` operation failed because the index address is incorrect. This may be due to update a spectrum which does not exist in the output file.
User action : if the error was caused by an `UPDATE` or `KEEP` command, try a `WRITE` command instead ; else, or if the error persists, submit an SPR.

8.3.8 N-Q

`F-NOISE, No spectrum in memory`
`LAS, NOISE` command. The command was specified without arguments, but there is no spectrum in the R memory to estimate a noise level.
User action : Specify a noise level, or get a spectrum in R.

`I-OUTPUT, <String> is reopened`
`LAS, FILE OUT` command, information message.

`I-OUTPUT, <String> successfully opened`
`LAS, FILE OUT` command, information message.

`W-OUTPUT, No output file opened`
`LAS, FILE OUT` command. The specified file could not be opened, or is not a valid CLASS file, but no previous output file has been reopened.

`F-OUTPUT, Non standard file <String>`
`LAS, FILE OUT` command. The specified file could be opened, but is not a CLASS data file.
User action : probably the file name has been mistyped.

`E-OUTPUT, Old format not supported`
`LAS, FILE OUT` command. The specified file has an old data format, and cannot be used for output.
User action : this file can be used for input. You may create a copy of it in new format by reading and writing to a new file all observations contained in it.

`F-OUTPUT, Open error file <String>`
`LAS, FILE OUT` command. The specified output file could not be opened. The message if followed by more details about the specific reason for the open failure. The most likely reasons are i) a non existing file, ii) a privilege violation (the user has not write access to the output file), or iii) the file may also be locked by another user.
User action : If you want to create a new output file, specify the 3^{rd} argument `NEW` to the `FILE OUT` command.

`F-OUTPUT, Read error file <String>`
`LAS, FILE OUT` command. The specified file could be opened, but most likely is not a CLASS data file.
User action : verify the file name, which may have been mistyped. If it is correct, notify your system manager because the error is due to a hardware problem.

`E-PEN, Number outside boundaries 0 - 15`
`E-PEN, Default pen not changed`
`LAS, PENCIL` command. A bad pen number was specified, the pen number is left unchanged.

`W-PEN, Colour outside boundaries 0 - <Integer>`
`W-PEN, Default colour 0 used`
`LAS, PENCIL` command. A bad colour index has been specified, the colour index is reset to the default value 0.

`W-PEN, Dashed pattern outside boundaries 1 - 7`
`W-PEN, Default dashed pattern 1 used`

LAS, PENCIL command. A bad dashed pattern has been specified, the dash pattern is reset to the default value 1 (continuous line).

W-POLYNO, Baseline extrapolation is hazardous
LAS, BASE command. Some of the line windows touch the spectrum edges, so that the baseline must be extrapolated in this range. Extrapolation is made by assuming a constant value in the range, and not by using the polynom because the solution is unstable.
User action : If possible, try to avoid such situations by having a few channels free of lines at the spectrum edges. Note that the extrapolation is most likely incorrect.

W-POLYNO, Degree <Integer> would be even better
LAS, BASE command. The specified degree would have given as good a fit to the baseline. It is hence possible that the current degree is too high and creates spurious oscillations.

E-POLYNO, NAG Error in <String>, ifail = <Integer>
LAS, BASE command. The minimization failed or the minimum is ill-defined.
User action : try to change the degree of the baseline, or the definition of the line windows to obtain a better solution.

W-PRINT, At entry <Integer> Scan <Integer>;<Integer>
ANALYSE, PRINT command. The information to be listed is missing or incomplete for the specified scan. This message is usually preceded by "W-RSEC, Absent section <Integer>"

E-PRINT, Cannot print <String>
ANALYSE, PRINT command. The specified keyword is not (yet) recognized.

E-PRINT, No input file connected
ANALYSE, PRINT command. There is nothing to print...

E-PRINT, Option /TABLE invalid for FIT
ANALYSE, PRINT command. FIT results cannot be written to a GILDAS table, but only to a formatted file.

E-PRINT, No channel list
ANALYSE, PRINT CHANNEL command. The channel list is missing or incorrectly given.

E-PRINT, Syntax error in list : <String>
E-PRINT, Invalid list : <Integer> TO <Integer> BY <Integer>
E-PRINT, Incomplete list : <String>
E-PRINT, Empty list
ANALYSE, PRINT command (possibly with the CHANNEL argument). The list is incorrect in some way.
User action : Correct the faulty list and resubmit the command.

W-PRINT, Too many channels, list truncated
ANALYSE, PRINT CHANNEL command. The channel list is too long and has been truncated.
User action : check the output file to see which channels are missing and use another PRINT command to list the missing channels.

E-PRINT, Too many arguments in list
ANALYSE, PRINT command for velocity ranges. There are too many velocity ranges specified.

`E-PRINT, Unable to open file <String>`
`ANALYSE, PRINT` command with `/OUTPUT` option. The output file could not be created, possibly because of open file quota, disk quota, privilege violation to the directory, or simply invalid file name.

8.3.9 R

`E-RBUF, Read error block <Integer>`
`LAS`, Input/Output routines. The specified block could not be read. This is most likely due to a hardware error.
User action : see "Recovering Corrupted Data Files". Notify your system manager.

`W-RDATA, Too many channels, truncated`
`LAS`, Input routines. The observation contains too many data points (more than 2048). Only 2048 channels were read.

`W-REDUCE, Only valid for skydips`
`ANALYSE, REDUCE` command. This command is only effective on skydips. Nothing happened.

`E-RESAMPLE, Invalid unit type <String>`
`ANALYSE, RESAMPLE` command. The resampling unit is unknown, it should be Velocity or Frequency.

`E-RESAMPLE, New spectrum does not intersect the original one`
`ANALYSE, RESAMPLE` command. The resampling formula creates a spectrum which does not intersect with the current one.
User action : use the `HEADER` command to see the current velocity or frequency scale for the current spectrum, and provide `RESAMPLE` arguments which are consistent with this scale.

`E-RESAMPLE, Too many output channels, maximum is <Integer>`
`ANALYSE, RESAMPLE` command. The parameters imply too many channels.
User action : decrease the resolution.

`F-RIX, Input file not opened`
`LAS`, Index routine. There is no input file connected.

`F-RIX, Read error block <Integer>`
`LAS`, Index routine. Unless it is preceded by the "`E-RIX, Wrong index address`" message, this error indicates a hardware problem.
User action : notify your system manager.

`E-RIX, Wrong index address <Integer>`
`LAS`, Index routine. The specified address does not exist. This error is usually due to an attempt to use a command such as `GET` or `GET NEXT` after an input file change.
User action : For these commands, use a `FIND` command before, or specify an observation number.

`E-ROBS, Non-existing index entry <Integer>`
`LAS`, Input/Output routine. The specified entry does not exist in the input file. This error is usually due to an attempt to use a command such as `GET` or `GET NEXT` after an input file

change.
User action : For these commands, use a `FIND` command before, or specify an observation number.

`E-ROBS, Non-standard obs. block <Integer>`
`LAS`, Input/Output routine. The specified block does not have the valid CLASS data format.
User action : Submit an SPR, and see "Recovering Corrupted Data Files"

`E-ROBS, Read error in index block`
`LAS`, Input/Output routine. The main index block could not be read.
User action : Retry later. Submit an SPR and see "Recovering Corrupted Data Files" if the error persists.

`F-ROX, Output file not opened`
`LAS`, Index routine. There is no output file connected.

`E-ROX, Wrong index address <Integer>`
`LAS`, Index routine. The specified address does not exist in the output file.
User action : Submit an SPR.

`F-ROX, Read error block <Integer>`
`LAS`, Index routine. The specified block could not be read in the output file.
User action : notify the system manager. The file may have been corrupted, see "Recovering Corrupted Data Files".

`W-RSEC, Absent section <Integer>`
`LAS`, Input/Output routines. The specified section does not exist in the current scan.

`E-RUNLAS, No spectrum in memory`
`LAS`, main program. The current command requires a scan, but no one has been read yet.

8.3.10 S

`E-SET BOX_LOCATION, Four arguments required`
`LAS, SET BOX` command, information message, see help.

`E-SET MODE, Invalid axis type <String>`
`LAS, SET MODE` command. Only two axis are recognised by this command : X and Y.

`E-SET PLOT_PAGE, Two arguments required`
`LAS, SET PLOT` command, information message, see help.

`I-SET, Clearing the plot and resetting BOX_LOCATION`
`LAS, SET DEFAULT` command, information message.

`W-SET, Invalid <String> <String>`
`LAS, SET` command, information message, see help.

`I-SET, Default file extension set to <String>`
`LAS, SET` command, information message, see help.

`I-SET, Message display level set to <Integer>`
`LAS, SET MESSAGE`, information message. Note that the message level has effect upon the number of messages output by CLASS. The default level is 4. Higher values may suppress important messages. Lower values can be used.

`W-SKYDIP, Solution not converged`
`ANALYSE, REDUCE` command. The minimization failed to converge.
User action : None, but the results may be unreliable.

`E-SKYDIP, Stupid calibration No oxygen in atmosphere`
`E-SKYDIP, Stupid calibration No water in atmosphere`
`E-SKYDIP, Stupid calibration Zero atmospheric opacity`
`ANALYSE, REDUCE` command. The minimization converged to a stupid solution.
User action : The calibration parameters of the skydip are probably incorrect. Discard the result.

`E-SMOOTH, Box width out of range`
`ANALYSE, SMOOTH BOX` command. The Box width is less than 2 or too large.

`W-SMOOTH, Cross-validation not converged`
`ANALYSE, SMOOTH AUTO` command. The cross-validation did not converge, no smoothing has been done.
User action : None. There is no way to force the cross validation to converge.

`W-SMOOTH, Error in AUTO smoothing`
`ANALYSE, SMOOTH AUTO` command. An error in the cross-validation algorithm happened. No smoothing has been done.
User action : Submit an SPR, supplying the data on which `AUTO` smoothing was tried.

`E-SMOOTH, Number of points out of range`
`ANALYSE, SMOOTH` command with argument NOISE.
User action : Decrease the number of channels

`E-SMOOTH, Too few channels`
`ANALYSE, SMOOTH` command. There are too few channels to make the smoothing.

`E-SPECTRUM, No spectrum in memory.`
`LAS, SPECTRUM or PLOT` commands. Not spectrum has been read yet.

`E-STAMP, Mode X and Y must be fixed`
`ANALYSE, STAMP` command. All spectra must have the same scale. Use the SET MODE command to do so.

`W-STAMP, More spectra in index than plotted.`
`ANALYSE, STAMP` command. All spectra in index could not be plotted.
User action : Change the number of spectra along each direction, or drop observations from the index.

`E-STAMP, No input file connected`
`ANALYSE, STAMP` command. No input file is currently opened.
User action : Open the desired input file, define an index using command `FIND` and retry the command.

`E-STRIP, Cannot create output image`
`E-STRIP, Cannot write output image header`
`ANALYSE, STRIP` command. The output file could not be created.
User action : this is usually a problem of file name, privilege or disk quota. Check these parameters.

`E-STRIP, Continuum data not supported`
`ANALYSE, STRIP` command with `LINE` type. The index contains data of continuum type.
User action : drop the invalid observation from the index.

`E-STRIP, Index does not define a strip`
`ANALYSE, STRIP` command. The index contains spectra which cannot be aligned along one of the two main directions.
User action : Check the index by command `LIST`, and build one which only contains offsets changing in one direction. If you want to do a strip in a diagonal direction, the only possibility is to modify the offset so that they describe a linear strip. Do not forget the changes if you do so...

`E-STRIP, Index does not define a continuum map`
`ANALYSE, STRIP` command in `CONTINUUM` type. The index contains continuum drift which cannot be aligned.
User action : Check the index by command `LIST`, and build one which only contains offsets changing in one direction.

`E-STRIP, Too few points in position`
`ANALYSE, STRIP` command. The index contains less than two spectra.
User action : Most likely your index does not contain what you believe. Check the index by command `LIST`, and build the one you want.

`E-STRIP, Too few points continuum drifts`
`ANALYSE, STRIP` command in `CONTINUUM` type. The index contains less than two drifts.
User action : Most likely your index does not contain what you believe. Check the index by command `LIST`, and build the one you want.

`W-SUM, Added folded to unfolded spectra.`
`LAS, SUM` command. The index contains observations made in frequency switching mode which have not been folded, and some which have been folded (or were made in other switching mode).
User action : Check your index, drop the incoherent spectra.

`I-SUM, Entry <Integer> found <Integer> ; <Integer>`
`LAS, SUM` command, information message.

`E-SUM, Index is empty`
`LAS, SUM` command.
User action : Build the index using command `SUM`, and try again.

`E-SUM, No input file connected`
`LAS, SUM` command. There is no input file.
User action : define an input file using command `FILE`, build the index using command `FIND`, and try again.

`E-SUM, Sum interrupted by <^C>`
`LAS, SUM` command. The sum was interrupted by the user by pressing `<^C>`. In general, the R spectrum contains nothing sensible in this case.
User action : None. Do not write the spectrum.

8.3.11 T through Z

`W-TAG, Observation <Integer> not found`
`LAS, TAG` command. The observation to be tagged does not exist, or is not of the current type.

`E-TAG, Quality out of range`
`LAS, TAG` command. Quality must be in the range 0-9 (0 Unknown, 1 to 8 from Excellent to Very bad, 9 reserved for deleted observations).

`E-UPDATE, Can only update last versions`
`LAS, UPDATE` command. Only last versions can be updated, the command had no effect.
User action : Use command `WRITE` instead of command `UPDATE`, but note that the new spectrum will become the last version in this case.

`E-UPDATE, Input file must equal Output file`
`LAS, UPDATE` command. The command had no effect.
User action : open the input as the output file too, and retry the command.

`E-WBUF, Read error block <Integer>`
`LAS`, Input/Output routines. A `READ` operation failed. The output file may be corrupted.
User action : notify your system manager. If the output file is corrupted, try to recover as many spectra as possible (cf "Recovering Corrupted Data Files").

`E-WBUF, Write error block <Integer>`
`LAS`, Input/Output routines. A `WRITE` operation failed. This usually corresponds to an exceeded disk quota. This error may also be due to a write-lock error on the output device. In these cases, the `WRITE` operation can be done later. The problem can also be more serious and indicate a corrupted output file.
User action : check disk quota. Notify your system manager. If the output file is corrupted, try to recover as many spectra as possible (see "Recovering Corrupted Data Files").

`E-WDATA, Observation not open for write nor modify`
`LAS`, Input/Output routines. An `UPDATE` or `WRITE` operation failed because the observation was not opened for write.
User action : This indicates an internal logic error in CLASS and should never occur. Submit an SPR.

`E-WDATA, Insufficient space available for data section`
`LAS`, Input/Output routines. An `UPDATE` or `WRITE` operation failed because the observation length is declared too short.
User action : This indicates an internal logic error in CLASS and should never occur. Submit an SPR.

`F-WOBS, No output file opened`
`LAS`, Input/Output routines. A `WRITE` operation failed because there is no output file.
User action : Open an output file.

`F-WOX, Output file index is full`
: LAS, Index output routines. A WRITE operation failed because there are too many spectra in the output file. Nothing has been written.
User action : There are two possibilities i) open a new output file and write the spectrum. ii) you may consider compressing the output file by keeping only the last versions of the spectra, and then open the compressed output file and write the spectrum.

`F-WOX, Read error block <Integer>`
: LAS, Index routines. A WRITE operation failed because the index could not be read correctly. This probably indicates a hardware problem.
User action : notify your system manager.

`F-WOX, Write error block <Integer>`
: LAS, Index routines. A WRITE operation failed because the index block could not be written correctly. This probably indicates a hardware problem.
User action : notify your system manager.

`I-WRITE, Observation <Integer>;<Integer> successfully <String>`
: LAS, WRITE or UPDATE commands, information message. <String> indicates whether the observations was updated or written.

`E-WSEC, Observation not open for write or modify`
: LAS, Input/Output routines. A WRITE operation failed because the observation is defined as readonly.
User action : This indicates an internal logic error in CLASS. Submit an SPR.

`F-WSEC, Too many sections`
: LAS, Input/Output routines. A WRITE operation failed because there are too many sections in the current spectrum to be written in the current file. This should not happen in the basic CLASS , but may occur to people using extended CLASS capabilities with user-defined sections.
User action : Submit an SPR, specifying the number of user-defined sections.

`F-WSEC, Section <Integer> already written`
: LAS, Input/Output routines. A WRITE operation failed because the specified section already exists. This indicates an internal logic error.
User action : Submit an SPR.

`F-WSEC, Insufficient room available for section <String>`
`F-WSEC, Absent section <String>`
: LAS, Input/Output routines. An UPDATE operation failed because the specified section is longer in the R spectrum than in the output file.
User action : use the WRITE command instead (immediately to avoid leaving the observation half modified in the output file).

`E-ZOOM, No cursor available`
: LAS, ZOOM command without arguments. No cursor is available to define the zooming window.
User action : supply the zooming window explicitly.

Index