CLASS
Continuum and Line Analysis Single-dish Software

A GILDAS software

November 21st, 2006

Version 1.1

Questions? Comments? Bug reports? Mail to: `gildas@iram.fr`

The GILDAS team welcomes an acknowledgement in publications
using GILDAS software to reduce and/or analyze data.
Please use the following reference in your publications:
`http://www.iram.fr/IRAMFR/GILDAS`

Documentation

In charge: S.Bardeau[1], J. Pety[1,2].
Active developers: P. Hily-Blant[3], S. Guilloteau[4].
Main past contributors: B. Delforge, T. Forveille, R. Lucas.

Software

In charge: S.Bardeau[1], J. Pety[1,2].
Active developers: P. Hily-Blant[3], S. Guilloteau[4].
Main past contributors: B. Delforge, T. Forveille, R. Lucas.

1. IRAM
2. Observatoire de Paris
3. Observatoire de Grenoble
4. Observatoire de Bordeaux

Contents

Chapter 1

Introduction

CLASS is a software package for reducing spectroscopic data obtained on a single-dish telescope. It also has basic functionalities to reduce continuum drifts like pointing or focus.

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.

On a standard installation, CLASS is entered by just typing `class`. 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 FIT gathers the spectra fitting functionalities.

Those languages are described in this manual. In addition, CLASS imports many functionalities defined and documented in other GILDAS packages:

- The command line interpretor is imported through the SIC (basic), GUI (for widgets) and VECTOR (miscellaneous) languages.

- The plotting possibilities through the GTVL (basic graphic actions), GREG1 (curve plotting), GREG2 (image plotting) and GREG3 (data cube plotting) languages.

- And the ephemerids and atmospheric contributions through the ASTRO language.

In addition to this manual, the reader should thus consult the SIC manual, and for further processing, the GREG manual.

This version of the CLASS documentation reflects the full rewritting of CLASS in `FORTRAN90`. In this process, many things have been changed, hopefully improved. If you are an experienced CLASS user, you may first want to consult the IRAM memo which describes only the changes in this version of CLASS.

Chapter 2

Cookbook

This part is a list of recipes enabling the beginner or the occasional user the get on the air very quickly, without losing his time searching the system's on-line `HELP` facility.

2.1   A `CLASS`ic Session

```
 1    device image white
 2    set angle seconds
 3    set coordinates equatorial
 4    file in raw.30m
 5    set line 13co(1-0)
 6    set source ic348
 7    set telescope iram-30m-b30
 8    set observed 15-aug-1984
 9    find /offset 0  25
10    set weight time
11    average
12    set unit v f
13    set mode x -1 14
14    set mode y -0.5 7.5
15    set plot histogram
16    plot
17    hardcopy /print
18    hardcopy spectrum.ps  /dev ps fast
19    hardcopy spectrum.eps /dev eps color
20    set window 3 6 8 10
21    base 4 /plot
22    plot
23    lines 0
24    gauss
25    fit
26    residual
27    plot
28    sic delete reduced.30m
29    file out reduced.30m new
```

```
30    swap
31    write
32    save ic348
33    exit
```

    1 Define the output device.

2-3 Select the coordinate system and angle unit.

    4 Open the input file.

5-9 Build the index according to various criteria.

10-11 Averaged all spectra with weights $w_i = \Delta t \Delta \nu / T_{\mathrm{sys}}^2$.

12-16 Plot the averaged spectrum in a given velocity interval, with velocity for the lower axis and rest frequency for the upper axis.

17-20 Make hardcopies (directly to the printer, in a ps or eps file).

21-23 Subtract a polynomial baseline. Plot the fitted baseline. Plot the baseline subtracted spectrum. The baseline does not take into account channels corresponding to x in the two specified windows, in current units (here velocity).

24-26 Perform single Gaussian fitting and plot the result.

27-28 Compute the residuals and plot them.

29-30 Open a new output file.

   31 Recover the baseline subtracted spectrum.

   32 Write it.

   33 Write the input commands in file ic348.class.

2.2   Reading a spectrum from a formatted input file

```
dev xl w
greg1\column x 1 y 2 /file ''filename''
model y x
plot
```

2.3   Exporting a spectrum to a formatted file

```
file in toto
find
get f
sic output toto.dat
for i 1 to channels
  say 'rx[i]' 'ry[i]' /format g12.4 g12.4
next
sic output
```

2.4   Building a Datacube from a CLASS File

The building of a regular grid in CLASS is done in two steps, in a similar fashion as the production of a PdBI regular grid from UV tables. The first step involves the building of a table: the spectra are written as rows in the Gildas internal data format for efficiency (command `table`). When creating the TABLE, the spectra are *not* resampled on a grid. The second step is the resampling of the non-regularly spaced spectra on a regular grid (command `xy_map`).

A simple example is given below where the cube is produced from a single .30m file. Default parameters are used by the `xy_map` command to define both the convolution kernel (1/3 of the $HPBW$) and the output grid (1/2 of the $HPBW$).

```
file in map        ! Open the input file
find               ! Build the index
consistency        ! Check first that the index is consistent
let name thecube   ! Use global SIC variables to define output cube name
let type lmv        ! The output cube is named ''thecube.lmv''
table 'name' new   ! Build the non-gridded table
xy_map 'name'      ! Grid the data from the table
go view            ! Check the quality of the data reduction with the VIEW tool
```

The following example illustrates the way to build a cube from several .30m files. This is typically the case when the map has been repeated several times and the data have been calibrated and reduced in different files.

```
file in map1       ! Open the input file
find               ! Build the index
consistency        ! Check first that the index is consistent
let name thecube   ! Use global SIC variables to define output cube name
let type lmv        !
table 'name' new   ! Build the non-gridded table as a new table
file in map2       ! Open the input file
find               ! Build the index
table 'name'       ! Append the index to the current table
file in map3       ! Open the input file
find               ! Build the index
table 'name'       ! Append the index to the current table
...
xy_map 'name'      ! Grid the data from the global table
go view            ! Check the quality of the data reduction with the VIEW tool
```

2.5   Fitting an HyperFine Structure

2.5.1   Assumptions

A1: same excitation temperature for all the components of the multiplet

A2: Gaussian profiles for the opacity as a function of frequency

A3: the lines all have the same width

A4: the multiplet components do not overlap

A5: the main beam temperature is well suited for your source

2.5.2   Parameters of the multiplet

When selecting the HFS method, you must give the name of a file that contains the relative positions and intensities of the components of your multiplet:
```
LAS> method hfs hfs-n2hp.dat
```
Relative intensities may be normalized or not. Here follow two examples, for the N_2H^+ multiplet.

```
7                   ! 1st line: Number of components
 6.9360  1.         ! 1st column: Velocity shift, for each component,
 5.9841  5.         !                (in km/s) with respect to the reference
 5.5452  3.         !                component you choose (here the 5th one).
 0.9560  5.         ! 2nd column: The relative strength of each component.
 0.0000  7.         !                Here, they are not normalized.
-0.6109  3.         !
-8.0064  3.         !


7                    ! As before.
14.9424 1./27.       ! The reference velocity is now that of the last
13.9906 5./27.       ! component.
13.5516 3./27.       ! The intensities are normalized (you could also
8.9624  5./27.       ! put numerical values rather that fractions).
8.0064  7./27.       !
7.3955  3./27.       !
0.      3./27.       !
```

Let us call v_i and r_i the positions and relative intensities of the N components of the multiplet. We define $S = \sum r_i$. In the first case, we thus have $N = 7$ and $S = 27$.

2.5.3   The fitting procedure

According to assumptions A3 and A4, the opacity of the ith component is written:

$$\tau_i(v) = \tau_i \cdot \exp\left[-4\ln 2 \left(\frac{v - v_{0,i}}{p_3}\right)^2\right] \tag{2.1}$$

where p_3 is the common FWHM of all components. The central velocity of component i is $v_{0,i} = v_i + p_2$, where p_2 is the velocity of the reference component (*i.e.* the one with $v_i = 0$).
   The opacity of the multiplet is the sum of the N opacities:

$$\tau(v) = p_4 \sum_{i=1}^{N} r_i \cdot \exp\left[-4\ln 2 \left(\frac{v - v_i - p_2}{p_3}\right)^2\right] \tag{2.2}$$

Given the opacity $\tau(v)$, the antena temperature is given by

$$T_{\text{ant}}(v) = \frac{p_1}{p_4}\left(1 - e^{-\tau(v)}\right) \tag{2.3}$$

From these equations, we deduce:

$$
\begin{aligned}
\tau(v_i + p_2) &= p_4 \cdot r_i \qquad \text{(assumption A4)} \\
\sum_i \tau_i &= p_4 \cdot S \\
T_{\text{ant}}(v_i + p_2) &= \frac{p_1}{p_4}\left(1 - e^{-p_4 r_i}\right) \\
T_{\text{ant}}(v_i + p_2) &\approx p_1 \cdot r_i
\end{aligned}
$$

where the last equality holds in the optically thin regime.

This implies that the physical meaning of p_4 depends on the value of S: If the relative intensities are normalized to unity, then $S = 1$ and p_4 equals the sum of all centerline opacities.

The results of the HFS fitting procedure are:

```
Line      T ant * Tau           V lsr            Delta V           Tau main

 1     1.313 ( 0.018)     3.783 ( 0.001)     0.589 ( 0.003)     0.230 ( 0.006)
```

where `T ant * Tau`=p_1, `V lsr`=p_2, `Delta V`=p_3 and `Tau main`=p_4.

According to assumptions A1 and A5, the hyperfine structure fitting procedure allows you to deduce the excitation temperature, since (assuming the Rayleigh-Jeans regime is valid, which is not true at $\lambda < 3$mm...):

$$
T_{\text{ant}}(v) = T_A^*(v) = \frac{B_{\text{eff}}}{F_{\text{eff}}}[T_{\text{ex}} - T_{\text{bg}}](1 - e^{-\tau(v)}) \tag{2.4}
$$

Finally, the excitation temperature is given by:

$$
T_{\text{ex}} = T_{\text{bg}} + \frac{F_{\text{eff}}}{B_{\text{eff}}}\frac{p_1}{p_4} \tag{2.5}
$$

Note: the main group opacity is limited to the range $0.1 - 30$ since outside these limits, the problem becomes degenerate because the opacity no longer appears in the equations (in the optically thin limit, line ratio no longer depend on the opacity, and in the thick limit, $\exp(-\tau) \ll 1$).

2.5.4   Typical Analysis Sequence

This routine produces the figure 2.1.

```
 1  set plot histo
 2  set format brief
 3  clear
 4  clear alpha
 5  file in prov.30m
 6  find
 7  get f
 8  modify source TOTO
 9  set unit v c
10  method hfs hfs-n2hp.lin
11  minimize
```

19; 5 TOTO   N2H+(1−0)   IRAM−30M−B41   +0   +0 Eq 3069. 0

       1   0.570 (0.011)   14.692 (0.002)   0.513 (0.004)   0.274 (0.010)
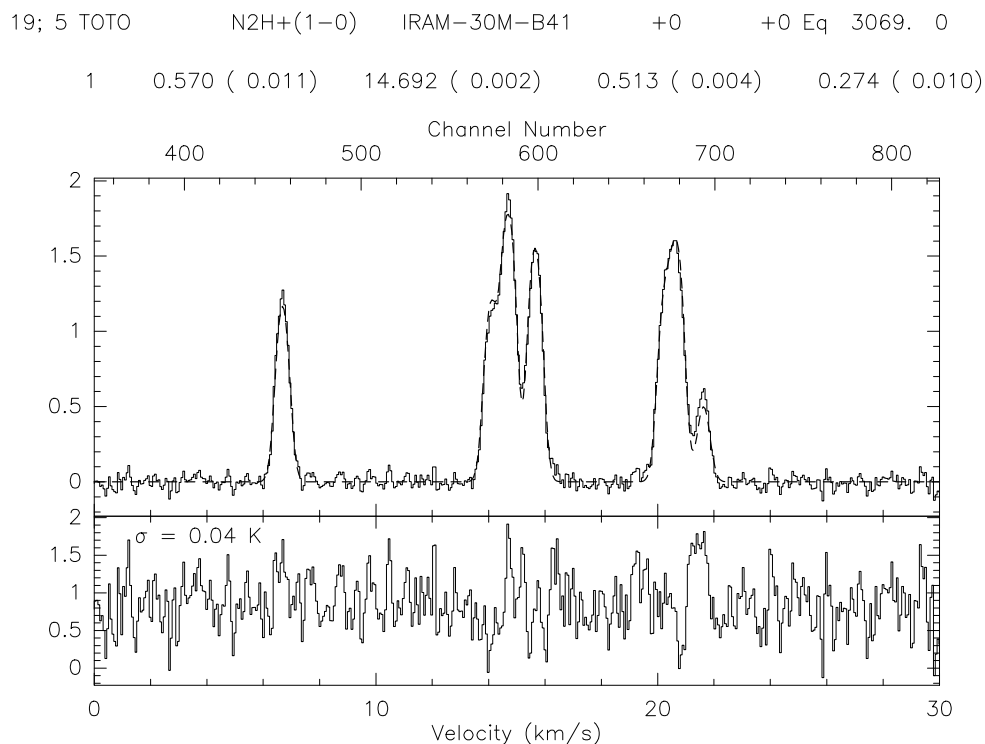
Figure 2.1: Result of the HFS fitting method (see 2.5.4).

```
12   set viewport 0.2 0.9 0.4 0.8
13   set mode x 0 30
14   box n o i p
15   spec
16   title
17   pen /c 0 /w 2 /das 2
18   visualize /pen
19   pen /def
20   display 1.5
21   set viewport 0.2 0.9 0.2 0.4
22   box p o i n /unit v upper
23   residual
24   spectr
25   define real rms
26   compute rms RMS ry
27   greg1\draw test 1 -0.5 "\gs = "'nint(rms*100)/100.'" K" 6 0 /box 7
```

2.6   Subtracting Baseline on Large Datasets

```
1   set verbose off
2   set level 10
```

```
 3  dev im w
 4  greg1\set plot portrait
 5  set format brief
 6  set angle sec
 7  file in co21.30m
 8  find /range -6 6 -6 6
 9  load
10  plot /index
11  set window /polygon 1
12  base 1 /index
13  plot /index
14  sic rename window-1.pol my-window-1.pol
15  sic delete co21-base.30m
16  file out co21-base.30m new
17  for i 1 to found
18     get n
19     base 1
20     write
21  next
22  file in co21.bas
23  find
24  load
25  plot /index
```

1. Makes CLASS quiet.

2. Turn off nearly all output messages.

3. Open an image window (necessary to plot 2D images).

4. Change the orientation to portrait.

5. Title in short format.

6. Set arcsecond units.

7. Open the file.

8. Build an index according to offsets.

9. Build a 2D array.

10. Display the whole index.

11. Define a polygon interactively to set spectral windows. By default the polygon is stored in a file window-1.pol.

12. Subtract a 1st order polynom to all records.

13. Display the result.

14. Change the polygon filename (to avoid overwritting it next time).

15. Delete output file if it exists.

16. Open the output file as new.

17-21 Make a loop over the index to subtract the baseline and write the result.

22-25 Display the result from the output file.

2.7   Reading/Writing FITS file

2.7.1   Exporting CLASS Spectra through FITS

```
1  set angle sec
2  file in co21.30m
3  find /range -2 2 -2 2
4  get f
5  set fits mode spectrum
6  fits write single.fits
7  find /range -2 2 -2 2
8  fits write index.fits /mode index
```

1. Angles are expressed in arcseconds.

2. Open the .30m file.

3. Build the index.

4. Store the first spectrum in `R` memory.

5. Default mode for writing `FITS` is set to single spectrum.

6. Write the `R` memory in `FITS` format file 'single.fits'.

7. Re-build the index.

8. Write the whole index in `FITS` format file 'index.fits'.

2.7.2   Importing FITS spectra into CLASS

```
1  set angle sec
2  fits read single
3  sic delete single.30m
4  file out single.30m new
5  write 1
6  file in single.30m
7  find
8  get f
9  plot
```

1. Angles are expressed in arcseconds.

2. Read a single `FITS` spectrum.

3-4 Open a new .30m file.

3. Write the FITS spectrum in the .30m file with index number 1.

6-9 Check the result by plotting the spectrum from the .30m file.

Chapter 3

User manuel

3.1 Generalities

3.1.1 Data and Log 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 subsections. 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 `.30m`. This can be changed with `SET EXTENSION .my_extension`.

CLASS also keeps two log files, named `class.log` and `class.mes`. They are created at CLASS start in the default directory `$HOME/.gag/logs`. This directory is defined in the SIC logical `gag_log:` which can be customized in the user customization file: `$HOME/.gag.dico`. The log files may be used to keep track of a batch or interactive work.

3.1.2 Observation and Version Number

Within CLASS, an *observation* should represent a single observing configuration, *e.g.* a single direction observed at a single central frequency with a single spectral resolution and in one polarization only (*i.e.* a single sky position, front-end and back-end combination). Each observation is given a number, named *observation number* at the time of creation of the CLASS data file. This number is then carried out in further manipulations.

Several version of a given observation may be stored in the same data file. Each version of a given observation thus represent different stages of the data reduction and all the versions of a given observation gives the history of the data processing. Each version of an observation is given a number (starting at 1), named *version number* at the time of creation of the CLASS data file. The version number increases automatically, each time the observation is modified (using `WRITE`). By default, only the last version of a given observation is relevant, *i.e.* `GET` reads the last version of an observation. It is possible (but not recommended) to store an observation without incrementing the version number with the `UPDATE` command so that 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 this, *i.e.* using the same observation number for very different things at time of creation of the CLASS data file, implies that you have to remember yourself which version corresponds to which configuration.

3.1.3   Scan and Subscan Number

For bookkeeping purpose, CLASS keeps track of a Scan and a Subscan number, which can be used as a selection criterium. The Scan number is attributed at observing time which is carried out in the CLASS data format. Moreover, two different observing mode are nowdays in common use in single-dish telescope:

- The pointed observing mode for which the telescope is pointed toward the source direction during all the integration to obtain only one spectrum. The scan is made of only one spectrum whose intensity is accumulated during the scan duration. Hence the Subscan number is always 1.

- The On–The–Fly (OTF) observing mode for which the telescope drift on source during the integration to make a small map. The scan is here composed of a collection of spectra dumped regularly (typically every 1 second) during a contiguous portion of time. Each dumped spectrum is also tagged by a subscan number whose value is incremented for each new OTF line (both to enable easy selection of a single line inside one OTF scan, and to ensure consistency with the 30m numbering). This subscan number is foreseen to always be greater than 1. There is one exception: when CLASS read data in old format, the subscan number is zero and a warning is issued.

3.1.4   Entry Number and Index

The `FIND` command offers the possibility to build an index of the observations matching several given selection criteria. The user can then easily process consistently only those observations. Each time a new index is formed using the `FIND` command, all the selected observations (which will belong to this index) are sequentially attributed a number, named *entry* number. This number goes from 1 to `found`, the number of observations in the current index. The entry number is never saved. It is just used as a number to process the current index in a loop.

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 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 19th, 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 "history" 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 spectra coming from Plateau de Bure Interferometer.

- **SET TYPE Name** specifies on which type of observations you deal with: "Continuum", "Line" or "Skydip".

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 to find all the versions of all 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.

Finally, the **SET SORT** force the **FIND** command to sort all the entries of an index in ascending order of a key parameter (*e.g.* lambda or beta offsets). **SET SORT number** implies the default order.

3.1.5   R and T Memories

CLASS keeps 2 observations in memory, in two different buffers, called **R** and **T**. The **R** memory is the only one that may be accessed directly; the **T** memory is only used for operations on spectra (additions,...). The **GET n** command places the spectrum corresponding to entry number **n** in the **R** buffer, while the previous **R** content is stored in the **T** buffer. The command **SWAP** exchanges both memories.

3.1.6   Large Set of Spectra

It is today possible with the IRAM–30 m to map a square degree field in CO (2-1). As an order of magnitude, this gives a final spectra cube of about 10^6 spectra with a slightly oversampling of $4''$. An observer who has just spent a few hours observing the same source in OTF mode may want to process all the dumped spectra at once even though they do not belong to the same scan.

The **LOAD** command gathers all the individual spectra currently in the index as a 2D array for future work, in particular visualization. This requires that all the spectra currently in the index are coherent, *i.e.* same source name, same line name and above all exactly the same frequency sampling. The latter can easily be achieved just by resampling. No checking is currently done about this, *i.e.* this is currently the responsibility of the observer to ensure a coherent frequency axis. The **/INDEX** option of commands like **PLOT** or **BASE** modifies there behavior to directly work on the 2D array defined with the **LOAD** command.

Nota Bene: The **STRIP** command which is producing a Velocity-Position plots, is *obsolescent*. Indeed the same functionality can be achived by the combination **LOAD; PLOT /INDEX** if the index is correctly choosen. And the **LOAD** command is much more powerful.

3.1.7   Variables

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`.

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 either to write a SIC loop that goes through the index or to test for actions which should be performed only if something exists in the index. 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.

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.

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.

3.2 Spectra Line Processing

3.2.1 Plotting Spectra

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).

- `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).

Single spectrum

Single spectrum are usually plotted using the following commands:

- `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.

Spectra map

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.

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.

Large set of spectra

An efficient way to look at a large set of coherent spectra (*e.g.* observed in OTF mode) is to plot them as a 2-D image where the intensity is colour coded. The image is formed by applying the

`LOAD` on the current index. Then the `/INDEX` option modify the single spectra plotting commands as follow:

- `BOX /INDEX`, which plots the frame, *i.e.* the entry number (Y axis) as a function of the velocity and/or frequency (X axis). The ranges of X and Y axes are controlled by the `SET MODE` command. `SET MODE Y` always control the intensity range.

- `SPECTRUM /INDEX`, which plots the image.

- `TITLE /INDEX`, which writes a header above the frame. Range of parameters (*e.g.* scan number, beta and lambda offsets, ...) are written.

- `PLOT /INDEX`, which performs all of `CLEAR; BOX /INDEX; SPECTRUM /INDEX; TITLE /INDEX` in a single operation.

3.2.2 Removing 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. See section 2.6 for a typical baseline fitting session.

The user first defines line windows by the command `SET WINDOW` with the following syntax:

```
SET WINDOW [wl1 wu1 [wl2 wu2 [...]]] [/VAR array]
           [/POLYGON [N] [filename1...filenameN]]
           [/NOCURSOR]
```

The `POLYGON` option enable the definition of 2D polygons on images obtain with `LOAD; PLOT /INDEX` when working on all the index. Line window values may be entered numerically as arguments, red from variables and line polygons may be red from input files. If available, the cursor may be used to define the windows or polygons. In the window 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. In the case of polygons, each left clic defines a gon and a right clic terminates the operations. The polygons may leak out of the image. Several polygons may be defined in case the line appear at very different velocities.

Up to 100 windows or 5 polygons may be defined. `BASE` then fits a polynomial to the part of the spectrum outside the line windows. 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.

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.

When working on an individual spectrum (not the index), 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...

When working on the whole index, the baseline are fitted spectrum per spectrum and the baseline–corrected spectra are stored in the 2D array ready for plotting with the next `PLOT /INDEX` command. However, baseline fitting results are lost and an explicit loop on the index entry must be use including a new baseline computation) to store the results with the `WRITE` command.

3.2.3   Folding Frequency Switched Spectra

Spectra obtained by Frequency Switching need to be folded to obtain the source spectra. 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 on the `R` buffer. The number of channels is decreased to keep only the relevant part of the resulting spectrum.

3.2.4   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, VELOCITY` and `FREQUENCY` are relevant for Line observations, while `POSITION` is relevant only for Continuum 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 `AVERAGE` and `ACCUMULATE` commands. `AVERAGE` 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 `AVERAGE`.

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 `AVERAGE` and `ACCUMULATE`. `AVERAGE` operates globally on all the spectra in the index, while `ACCUMULATE` adds the `R` and `T` buffers into `R`. `AVERAGE` 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...

3.2.5   Analyzing profiles

The CLASS user may analyse spectra by fitting profiles. The fitting commands are available from the `FIT` language. The minimization method is taken from the MINUIT system of CERN, modified and optimized for this purpose. Reliability proved 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` (see details below) 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.

`METHOD SHELL` in details. The fitted function is:

$$f(\nu) = \frac{\mathcal{A}}{\Delta\nu} \frac{1 + 4H\left[(\nu - \nu_0)/\Delta\nu\right]^2}{1 + H/3}$$

where the fitted parameters are:

1. $\mathcal{A}$: the area under the profile (in K MHz),

2. ν_0: the middle frequency (in MHz),

3. $\Delta\nu$: the full width at zero level (in MHz),

4. H: the Horn/Center parameter (dimensionless, see below)

The central value is $f(\nu_0) = \frac{\mathcal{A}}{\Delta\nu} \frac{1}{1+H/3}$ whilst the value at the edge is $f(\nu_0 + \Delta\nu/2) = \frac{\mathcal{A}}{\Delta\nu} \frac{1+H}{1+H/3}$. The edge-to-center intensity ratio value is thus dictated by the Horn/Center parameter H according to

$$\frac{f(\Delta\nu/2)}{f(0)} = 1 + H$$

The center-to-edge frequency shift corresponds to an expanding velocity

$$v_{\mathrm{exp}} = c\, \frac{\Delta\nu/2}{\nu_0}$$

Figure 3.1 shows synthetic shell-like profiles, for which the area takes values $\mathcal{A} = 5$ to 20 by 5 K MHz. The full width at zero level is $\Delta\nu = 3.83$ MHz in all cases, which corresponds to $v_{\mathrm{exp}} = 2.49$ km.s^{-1}at 230.537 GHz. The Horn/Center parameter is $H = 5$ (top) or $H = -1$ (bottom, for which the intensity at the edge is zero).

The **`FIT`** commands are:

- **`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

```
1 20.0   (1.26E-2)  230537.963(8.19E-6)  233527.141(8.19E-6)   2.49( 0.00)    4.98(0.02)    2.0
1 15.0   (1.77E-2)  230537.963(2.05E-6)  233527.141(2.05E-6)   2.49( 0.00)    4.95(0.03)    1.5
1 9.97   (2.03E-2)  230537.963(6.59E-6)  233527.141(6.59E-6)   2.49( 0.00)    5.01(0.05)    1.0
1 5.01   (1.79E-2)  230537.963(1.85E-4)  233527.141(1.85E-4)   2.49( 0.00)    4.91(0.08)    0.5
```



```
1 20.0   (2.11E-2)  230537.970(1.17E-3)  233527.134(1.17E-3)   2.45( 0.01)   -0.97(0.01)    7.8
1 15.1   (1.60E-2)  230537.967(1.16E-3)  233527.138(1.16E-3)   2.49( 0.00)   -1.00(0.00)    5.9
1 9.95   (2.06E-2)  230537.973(2.29E-3)  233527.132(2.29E-3)   2.43( 0.01)   -0.96(0.01)    3.9
1 5.02   (2.05E-2)  230537.968(1.41E-3)  233527.136(1.41E-3)   2.48( 0.00)   -1.00(0.02)    2.0
```
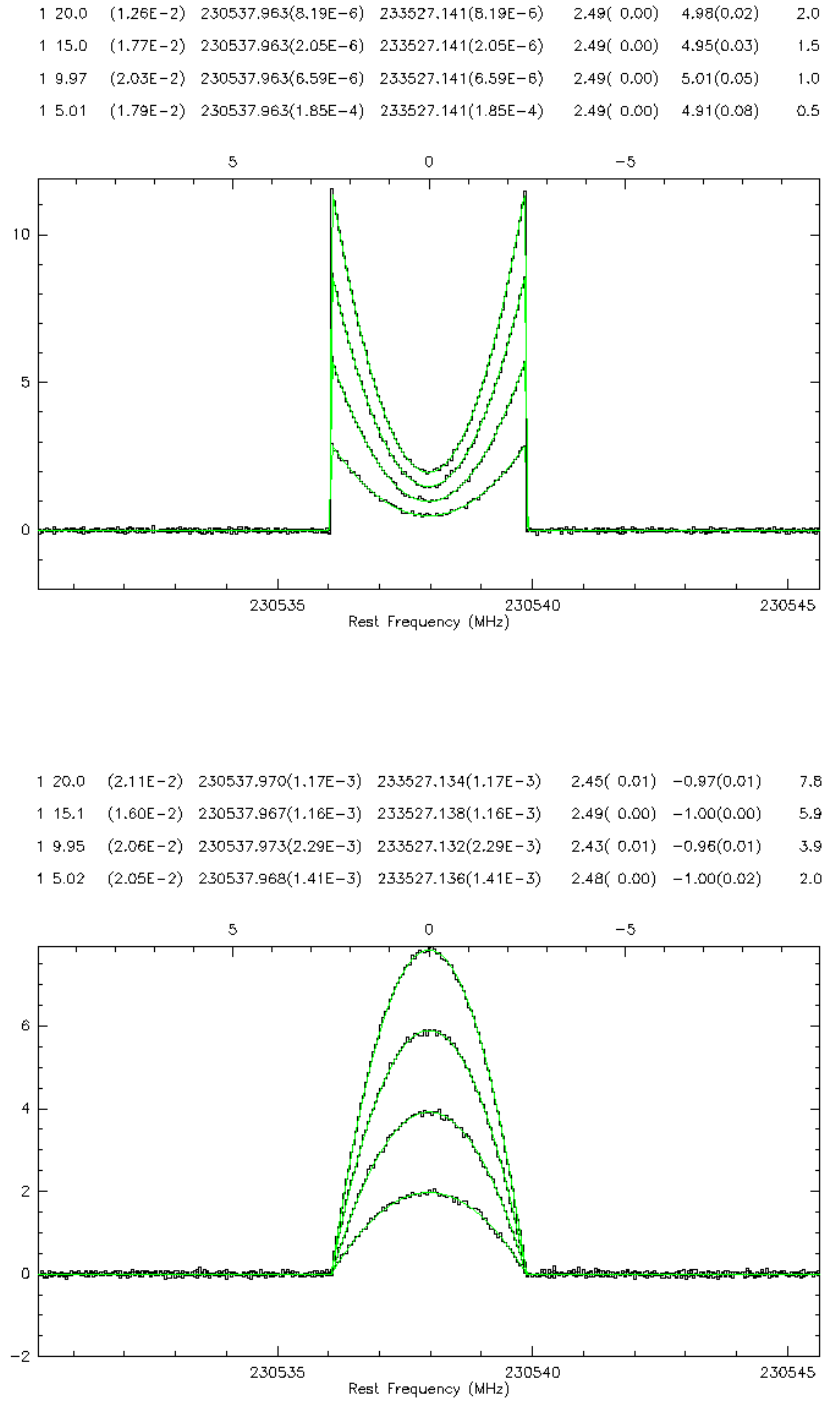

Figure 3.1: Four synthetic shell-like profiles, corresponding to increasing areas $\mathcal{A} = 5$ to 20 by 5 K MHz, a $\Delta\nu$ of 3.83 MHz or $v_{\mathrm{exp}} = 2.49$ km.s^{-1} at 230.537 GHz. The results of fitted shell profiles are indicated at the top. The upper scale is km.s^{-1}. *T*op: The Horn/Center parameter is $H = 5$. *B*ottom: Same as above for $H = -1$.

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 the cursor is available. 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.

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

- `VISUALIZE [N]  [/PEN]` 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 and made available through the corresponding variable section (see `SET VARIABLE` help).

3.2.6   Gridding Spectra on a 3-D 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.

For mapping purpose, CLASS is able to produce images at the GILDAS format. GILDAS (the Grenoble Image and Line Data Analysis System) is an set 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.

3.2.7 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.

3.3 Continuum and Skydip Processing

3.3.1 Continuum

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

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 a set 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.

3.3.2   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   Communication with the outer world

3.4.1   Listing Scientifically Valuable Results

Command PRINT offers a way to list a number of valuable information on screen or in an ASCII/binary file:

- 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 program.

As all commands using a set 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`.

3.4.2 Making Publishable Quality Figures

CLASS has many functionalities to directly produce publishable quality Figures. All the GREG commands are imported in CLASS to fully annotate plots, superpose spectra with related data, stack various plots and then make hardcopy (like Post-Scripts files). A few guidelines are given here on essential GREG commands. For more details, users are advised to read the GREG manual.

Moreover, CLASS commands like `DRAW` and `GREG` implements fancy functionalities of common use when producing figures around spectra.

GREG functionalities

CLASS is mainly 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`).

- `SET PLOT_PAGE ...` Define the page size. Warning: You will get into trouble if you want to abreviate this command to `SET PLOT` as `SET PLOT` is a valid CLASS command used to indicate whether spectra are broken lines of histograms. A way out of this is to abbreviate the GREG command as `G\SET PLOT`.

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

Note that you do not need to open a plotting window to produce a hardcopy through the **HARDCOPY** command. It is only much more convenient, but the plot (*e.g.* sequence of commands) and the way it is displayed (*e.g.* PS format, bitmap on a computer screen) are two completely independent things.

Bridge toward GREG functionalities

The spectra and the results of their analysis (like the fits) are in CLASS internal buffers not always easily accessible for plotting with the GREG commands. CLASS thus implements the **GREG** command which is intended to produce a direct interface with GREG for plots of spectra. It creates a GILDAS table which can be read using the standard GREG commands for further plotting. 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.

Annotations

CLASS has two DRAW commands to annotate plots: *i)* The standard GREG **DRAW** command which can be accessed by typing **G\DRAW** (please look at the GREG manual) and *ii* a special flavor of **DRAW** customized for annotating spectra. The CLASS flavor is the default one used when typing just **DRAW** The basic operations performed by this flavor of the **DRAW** command 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.

- `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 line 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.

3.4.3 Importing and Exporting Spectra From and To FITS

No data reduction package has all the functionalities any user dream about. But a user may know that the functionality he needs is available in a very specific package. Here comes the need to exchange data between packages. The current standard answer to this problem is FITS. 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
```

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.

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. A detailed description of the CLASS FITS flavors is given in chapter 4. 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`.

From CLASS to FITS

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).

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.

Chapter 4

Developer Manual (04-dec-2006)

Please check directly the CLASS sources to get the most up-to-date information.

4.1 Internal CLASS Format

4.1.1 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 spectrum or one continuum drift.

- 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, ...)

  - Data

  - ...

  Each type of section is known by the system by means of an integer code. It is not required that all sections 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 stages 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 up to 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.

4.1.2 File organization

Data is stored in a direct access file. The record length is 512 bytes. Blocks may contain either ovservations 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 file type (*single* versus *multiple*) and the internal storage format of the machine which has written the data. Table 4.1 summarizes the possible codes. This enables the reading machine to convert the data to its internal format.

	Vax	IEEE	EEEI
MULTIPLE	'1 '	'1A '	'1B '
SINGLE	'9 '	'9A '	'9B '

Table 4.1: CLASS file identification codes depending on the file type.

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 block 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

4.2 The Header and Data 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.

4.2.1 General Parameters

```
! GENERAL: General parameters, always present.
integer(kind=4), parameter :: class_sec_gen_id=-2
integer(kind=4), parameter :: class_sec_gen_len=9
type general
   sequence
   ! Not written in the entry (written in index instead)
   integer(kind=4) :: num     ! [           ] Observation number
   integer(kind=4) :: ver     ! [           ] Version number
   integer(kind=4) :: teles(3)! [           ] Telescope name
   integer(kind=4) :: dobs    ! [MJD-60549] Date of observation
   integer(kind=4) :: dred    ! [MJD-60549] Date of reduction
   integer(kind=4) :: typec   ! [     code] Type of coordinates
   integer(kind=4) :: kind    ! [     code] Type of data
   integer(kind=4) :: qual    ! [     code] Quality of data
   integer(kind=4) :: scan    ! [           ] Scan number
   integer(kind=4) :: subscan ! [           ] Subscan number
   ! Written in the entry
   real(kind=8)    :: ut      ! [  rad] UT of observation
   real(kind=8)    :: st      ! [  rad] LST of observation
   real(kind=4)    :: az      ! [  rad] Azimuth
   real(kind=4)    :: el      ! [  rad] Elevation
   real(kind=4)    :: tau     ! [neper] Opacity
   real(kind=4)    :: tsys    ! [    K] System temperature
```

```
   real(kind=4)     :: time    ! [    s] Integration time
   integer(kind=4) :: xunit    ! [ code] X unit (if X coordinates section is present)
end type general
```

4.2.2   Position information

```
! POSITION: Position information.
integer(kind=4), parameter :: class_sec_pos_id=-3
integer(kind=4), parameter :: class_sec_pos_len=17
type position
   sequence
   integer(kind=4) :: sourc(3) ! [   ] Source name
   real(kind=4)     :: epoch    ! [   ] Epoch of coordinates
   real(kind=8)     :: lam      ! [rad] Lambda
   real(kind=8)     :: bet      ! [rad] Beta
   real(kind=4)     :: lamof    ! [rad] Offset in Lambda
   real(kind=4)     :: betof    ! [rad] Offset in Beta
   integer(kind=4) :: proj     ! [rad] Projection system
   real(kind=8)     :: sl0p     ! [rad] lambda of descriptive system
   real(kind=8)     :: sb0p     ! [rad] beta   of descriptive system
   real(kind=8)     :: sk0p     ! [rad] angle  of descriptive system
end type position
```

4.2.3   Spectroscopic information (for spectra)

```
! SPECTRO: Spectroscopic information (for spectra).
integer(kind=4), parameter :: class_sec_spe_id=-4
integer(kind=4), parameter :: class_sec_spe_len=17
type spectro
   sequence
   integer(kind=4) :: line(3) ! [    ] Line name
   real(kind=8)     :: restf   ! [ MHz] Rest frequency
   integer(kind=4) :: nchan   ! [    ] Number of channels
   real(kind=4)     :: rchan   ! [    ] Reference channels
   real(kind=4)     :: fres    ! [ MHz] Frequency resolution
   real(kind=4)     :: foff    ! [ MHz] Frequency offset
   real(kind=4)     :: vres    ! [km/s] Velocity resolution
   real(kind=4)     :: voff    ! [km/s] Velocity at reference channel
   real(kind=4)     :: bad     ! [    ] Blanking value
   real(kind=8)     :: image   ! [ MHz] Image frequency
   integer(kind=4) :: vtype   ! [code] Type of velocity
   real(kind=8)     :: doppler ! [km/s] Doppler correction
```

4.2.4   Baseline information (for spectra or drifts)

```
! BASE: Baseline information (for spectra of drifts).
integer(kind=4), parameter :: class_sec_bas_id=-5
integer(kind=4), parameter :: mwind=100
integer(kind=4), parameter :: class_sec_bas_len=8+2*mwind
```

```
type base
   sequence
   integer(kind=4) :: deg        ! [        ] Degree of last baseline
   real(kind=4)    :: sigfi      ! [Int. unit] Sigma
   real(kind=4)    :: aire       ! [Int. unit] Area under windows
   integer(kind=4) :: nwind      ! [        ] Number of line windows
   real(kind=4)    :: w1(mwind)  ! [   km/s] Lower limits of windows
   real(kind=4)    :: w2(mwind)  ! [   km/s] Upper limits of windows
   real(kind=4)    :: sinus(3)   ! [        ] Sinus baseline results
end type base
```

4.2.5 Scan numbers of initial observations

```
! HISTORY: Scan numbers of initial observations.
integer(kind=4), parameter :: class_sec_his_id=-6
integer(kind=4), parameter :: mseq=100
integer(kind=4), parameter :: class_sec_his_len=2*mseq+1
type history
   sequence
   integer(kind=4) :: nseq        ! Number of sequences
   integer(kind=4) :: start(mseq) ! Start can number of seq.
   integer(kind=4) :: end(mseq)   ! End scan number of seq.
end type history
```

4.2.6 Default plotting limits

```
! PLOT: Default plotting limits.
integer(kind=4), parameter :: class_sec_plo_id=-7
integer(kind=4), parameter :: class_sec_plo_len=4
type plot
   sequence
   real(kind=4)    :: amin    ! [Int. unit] Min Y value plotted
   real(kind=4)    :: amax    ! [Int. unit] Max Y value plotted
   real(kind=4)    :: vmin    ! [   km/s] Min X value plotted
   real(kind=4)    :: vmax    ! [   km/s] Max X value plotted
end type plot
```

4.2.7 Switching information (for spectra)

```
! SWITCH: Switching information (for spectra).
integer(kind=4), parameter :: class_sec_fsw_id=-8
integer(kind=4), parameter :: mxphas=8
integer(kind=4), parameter :: class_sec_fsw_len=2+6*mxphas
type fswitch
   sequence
   integer(kind=4) :: nphas          ! [    ] Number of phases
   real(kind=8)    :: decal(mxphas)  ! [ MHz] Frequency offsets
   real(kind=4)    :: duree(mxphas)  ! [   s] Time per phase
   real(kind=4)    :: poids(mxphas)  ! [    ] Weight of each phase
```

```
      integer(kind=4) :: swmod              ! [code] Switching mode (frequency, position...)
      real(kind=4)    :: ldecal(mxphas)  ! [ rad] Lambda offsets
      real(kind=4)    :: bdecal(mxphas)  ! [ rad] Beta offsets of each phase
   end type fswitch
```

4.2.8   Calibration parameters

```
  ! CALIBRATION: Calibration parameters.
  integer(kind=4), parameter :: class_sec_cal_id=-14
  integer(kind=4), parameter :: class_sec_cal_len=25
  type calib
     sequence
     real(kind=4)    :: beeff    ! [     ] Beam efficiency
     real(kind=4)    :: foeff    ! [     ] Forward efficiency
     real(kind=4)    :: gaini    ! [     ] Image/Signal gain ratio
     real(kind=4)    :: h2omm    ! [   mm] Water vapor content
     real(kind=4)    :: pamb     ! [  hPa] Ambient pressure
     real(kind=4)    :: tamb     ! [    K] Ambient temperature
     real(kind=4)    :: tatms    ! [    K] Atmosphere temp. in signal band
     real(kind=4)    :: tchop    ! [    K] Chopper temperature
     real(kind=4)    :: tcold    ! [    K] Cold load temperature
     real(kind=4)    :: taus     ! [neper] Opacity in signal band
     real(kind=4)    :: taui     ! [neper] Opacity in image band
     real(kind=4)    :: tatmi    ! [    K] Atmosphere temp. in image band
     real(kind=4)    :: trec     ! [    K] Receiver temperature
     integer(kind=4) :: cmode    ! [ code] Calibration mode
     real(kind=4)    :: atfac    ! [     ] Applied calibration factor
     real(kind=4)    :: alti     ! [    m] Site elevation
     real(kind=4)    :: count(3) ! [count] Power of Atm., Chopp., Cold
     real(kind=4)    :: lcalof   ! [  rad] Longitude offset for sky measurement
     real(kind=4)    :: bcalof   ! [  rad] Latitude offset for sky measurement
     real(kind=8)    :: geolong  ! [  rad] Geographic longitude of observatory
     real(kind=8)    :: geolat   ! [  rad] Geographic latitude of observatory
  end type calib
```

4.2.9   For Skydips observations. No associated data.

```
  ! SKYDIP: For Skydips observations. No associated data.
  integer(kind=4), parameter :: class_sec_sky_id=-16
  integer(kind=4), parameter :: msky=10
  integer(kind=4), parameter :: class_sec_sky_len=10+4*msky
  type skydip
     sequence
     integer(kind=4) :: line(3)     ! [   ] Line name
     real(kind=8)    :: restf       ! [MHz] Rest frequency
     real(kind=8)    :: image       ! [MHz] Image frequency
     integer(kind=4) :: nsky        ! [   ] Number of points on sky
     integer(kind=4) :: nchop       ! [   ]   -   -   -    - chopper
```

```
      integer(kind=4) :: ncold      ! [   ]  -  -   -   - cold load
      real(kind=4)    :: elev(msky) ! [rad] Elevations
      real(kind=4)    :: emiss(msky) ! [  ?] Power on sky
      real(kind=4)    :: chopp(msky) ! [  ?] Power on chopper
      real(kind=4)    :: cold(msky)  ! [  ?] Power on cold load
   end type skydip
```

4.2.10 Gauss fit results (for spectra or drifts)

```
   ! GAUSS: Gauss fit results (for spectra or drifts).
   integer(kind=4), parameter :: class_sec_gau_id=-9
   integer(kind=4), parameter :: mxgaus=5
   integer(kind=4), parameter :: mgausfit=3*mxgaus
   integer(kind=4), parameter :: class_sec_gau_len=3+2*mgausfit
   type gauss
      sequence
      integer(kind=4) :: nline            ! [               ] Number of components
      real(kind=4)    :: sigba            ! [   Int. unit] Sigma on base
      real(kind=4)    :: sigra            ! [   Int. unit] Sigma on line
      real(kind=4)    :: nfit(mgausfit)   ! [area,v0,fwhm] Fit results
      real(kind=4)    :: nerr(mgausfit)   ! [area,v0,fwhm] Errors
   end type gauss
```

4.2.11 "Stellar shell" profile fit results (for spectra)

```
   ! SHELL: "Stellar shell" profile fit results (for spectra).
   integer(kind=4), parameter :: class_sec_she_id=-12
   integer(kind=4), parameter :: mshellfit=20
   integer(kind=4), parameter :: class_sec_she_len=3+2*mshellfit
   type shell
      sequence
      integer(kind=4) :: nline            ! [] Number of components
      real(kind=4)    :: sigba            ! [] Sigma on base
      real(kind=4)    :: sigra            ! [] Sigma on line
      real(kind=4)    :: nfit(mshellfit)  ! [] Fit results
      real(kind=4)    :: nerr(mshellfit)  ! [] Errors
   end type shell
```

4.2.12 Hyperfine structure profile fit results (for spectra)

```
   ! HFS: "Hyperfine Structure" profile fit results (e.g. NH3, HCN, for spectra).
   integer(kind=4), parameter :: class_sec_hfs_id=-13
   integer(kind=4), parameter :: mhfsfit=12
   integer(kind=4), parameter :: class_sec_hfs_len=3+2*mhfsfit
   type hfs
      sequence
      integer(kind=4) :: nline            ! Number of components
      real(kind=4)    :: sigba            ! Sigma on base
      real(kind=4)    :: sigra            ! Sigma on line
```

```
    real(kind=4)      :: nfit(mhfsfit)  ! Fit results
    real(kind=4)      :: nerr(mhfsfit)  ! Errors
end type hfs
```

4.2.13   Hyperfine structure absorption profile fit results (for spectra)

```
! ABS: "Hyperfine Structure" absorption profile fit results (e.g. NH3, HCN, for spectra).
integer(kind=4), parameter :: class_sec_abs_id=-18   ! Absorption fit
integer(kind=4), parameter :: mabsfit=16
integer(kind=4), parameter :: class_sec_abs_len=3+2*mabsfit
type absorption
    sequence
    integer(kind=4) :: nline          ! Number of components
    real(kind=4)    :: sigba          ! Sigma on base
    real(kind=4)    :: sigra          ! Sigma on line
    real(kind=4)    :: nfit(mabsfit)  ! Fit results
    real(kind=4)    :: nerr(mabsfit)  ! Errors
end type absorption
```

4.2.14   Continuum drift description (for drifts)

```
! DRIFT: Continuum drift description (for drifts).
integer(kind=4), parameter :: class_sec_dri_id=-10
integer(kind=4), parameter :: class_sec_dri_len=16
type drift
    sequence
    real(kind=8)    :: freq    ! [ MHz] Rest frequency
    real(kind=4)    :: width   ! [ MHz] Bandwidth
    integer(kind=4) :: npoin   ! [    ] Number of data points
    real(kind=4)    :: rpoin   ! [    ] Reference point
    real(kind=4)    :: tref    ! [   ?] Time at reference
    real(kind=4)    :: aref    ! [ rad] Angular offset at ref.
    real(kind=4)    :: apos    ! [ rad] Position angle of drift
    real(kind=4)    :: tres    ! [   ?] Time resolution
    real(kind=4)    :: ares    ! [ rad] Angular resolution
    real(kind=4)    :: bad     ! [    ] Blanking value
    integer(kind=4) :: ctype   ! [code] Type of offsets
    real(kind=8)    :: cimag   ! [ MHz] Image frequency
    real(kind=4)    :: colla   ! [   ?] Collimation error Az
    real(kind=4)    :: colle   ! [   ?] Collimation error El
end type drift
```

4.2.15   Beam-switching parameters (for spectra or drifts)

```
! BEAM: Beam-switching parameters (for spectra or drifts).
integer(kind=4), parameter :: class_sec_bea_id=-11
integer(kind=4), parameter :: class_sec_bea_len=5
type beam
    sequence
```

```
      real(kind=4)     :: cazim   ! [ rad] Azimuth of observation
      real(kind=4)     :: celev   ! [ rad] Elevation of observation
      real(kind=4)     :: space   ! [ rad] Beam spacing
      real(kind=4)     :: bpos    ! [ rad] Position angle of beams
      integer(kind=4) :: btype    ! [code] System for angle
   end type beam
```

4.2.16   Double gaussian and baseline fit results (for drifts)

```
   ! CONTINUUM: Double gaussian and baseline fit results (for drifts).
   integer(kind=4), parameter :: class_sec_poi_id=-15
   integer(kind=4), parameter :: mpoifit=8
   integer(kind=4), parameter :: class_sec_poi_len=3+2*mpoifit
   type pointing
      sequence
      integer(kind=4) :: nline          ! Number of components
      real(kind=4)    :: sigba          ! Sigma on base
      real(kind=4)    :: sigra          ! Sigma on line
      real(kind=4)    :: nfit(mpoifit)  ! Fit results
      real(kind=4)    :: nerr(mpoifit)  ! Errors
   end type pointing
```

4.2.17   Comment Section

```
   integer(kind=4), parameter :: class_sec_com_id=-1
   integer(kind=4), parameter :: class_sec_comm_len=256
   type comment
      sequence
      integer(kind=4) :: ltext   ! Length of comment
      character ctext*1024       ! Comment string
   end type comment
```

4.2.18   Data Section

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

4.3   Old OTF data format

This section is kept for backward compatibility. Indeed CLASS is still able to read old OTF data format. Once CLASS, always CLASS...

4.3.1   Data Section Descriptor

```
   ! Data Section Descriptor.
   ! (Old OTF data format, kept for backward compatibility).
   integer(kind=4), parameter :: class_sec_desc_id=-30
   integer(kind=4), parameter :: class_sec_desc_len=4
   type descriptor
```

```
    sequence
    integer(kind=4) :: ndump   ! Number of records
    integer(kind=4) :: ldpar   ! Length of data header (longwords)
    integer(kind=4) :: ldatl   ! length of line data      (")
    integer(kind=4) :: ldump   ! length of record         (")
    integer(kind=4) :: rec     ! Current Record Number
  end type descriptor
```

4.3.2   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	AZ	[rad] Telescope Azimuth
2	Real*4	EL	[rad] Telescope Elevation
3	Real*8	ST	[rad] Sideral Time
5	Real*4	LAMOF	[rad] Azimuth offset
6	Real*4	BETOF	[rad] Elevation offset
12	Real*4	UT	[sec] Universal Time in day DOBS

4.4   CLASS **FITS format**

Starting with dec06 release of GILDAS, the FITS support of CLASS is largely improved. In addition to the old possibility of writing one spectrum per file (see the "Simple SPECTRUM Mode" section), it is now possible to read from the same FITS file, one spectrum in the primary Header/Data Unit and collections of spectra in bintables stored as FITS extensions (see the "BINTABLE Mode"). Moreover, it is now possible to store in the same bintable, spectra of different number of channels. The distinction between the SPECTRUM and BINTABLE mode is mainly historical as the software automatically toggles from one mode to another.

4.4.1   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 kind of 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...

4.4.2 BINTABLE Mode

In addition to the simple "one FITS file per Spectrum" mode, CLASS supports the Binary Table extension, in reading and writing. CLASS reads and processes all the Binary Table extension of the FITS file. 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. Reading such FITS files, CLASS will support any value for the `TTYPEi` keywords. If a column and a keyword have the same name, the column prevails. If several columns have the same name (try to avoid this), the last one prevails.

The spectra are handled in a `SPECTRUM` column.

Here is an example with a tiny primary Header Data Unit and a Binary Table extension header.

```
SIMPLE  =                    T
BITPIX  =                    8
NAXIS   =                    0
EXTEND  =                    T
BLOCKED =                    T
ORIGIN  = 'My observatory'
CREATOR = 'Me        '
END

...

XTENSION= 'BINTABLE'
BITPIX  =                    8          / Always 8.
NAXIS   =                    2          / Always 2: tables have 2 dimensions.
NAXIS1  =                 7275          / Number of bytes per row.
NAXIS2  =                    4          / Number of rows.
PCOUNT  =                    0          / Usually 0.
GCOUNT  =                    1          / Always 1.
TFIELDS =                   18          / Number of columns.
EXTNAME = 'MATRIX  '                    / Just a name, not important.
EXTVER  =                    1          / Always 1.
MAXIS   =                    4          / Number of dimensions in the data.
MAXIS1  =                 1793          / Dummy number of channels (see TTYPE1).
MAXIS2  =                    1          /
MAXIS3  =                    1          /
MAXIS4  =                    1          /
CTYPE1  = 'FREQ    '                    / Dim1: freq => MAXIS1 = Nb channels.
CRVAL1  =  0.0000000000000E+00          / Frequency offset, always 0.
CDELT1  =  0.1000000000000E+07          / Frequency resolution [Hz].
CRPIX1  =  0.2565000000000E+03          / Dummy reference channel (see TTYPE18).
CTYPE2  = 'RA      '
CRVAL2  =  0.0000000000000E+00
CDELT2  =  0.0000000000000E+00
CRPIX2  =  0.0000000000000E+00
CTYPE3  = 'DEC     '
CRVAL3  =  0.0000000000000E+00
CDELT3  =  0.0000000000000E+00
CRPIX3  =  0.0000000000000E+00
CTYPE4  = 'STOKES  '
CRVAL4  =  0.0000000000000E+00
CDELT4  =  0.0000000000000E+00
CRPIX4  =  0.0000000000000E+00
SUBSCAN =                    1          / Subscan number.  Often 1.
LINE    = 'CS(5-4)     '                / Name of your line, up to 12 chars.
OBJECT  = 'JUPITER     '                / Name of your source, up to 12 chars.
RESTFREQ=  0.2449356430000E+12          / Rest (signal) frequency at ref chan.
VELO-HEL=  0.0000000000000E+00          / Velocity at ref.  chan [m.s-1].
```

```
VELDEF  = 'RADI-HEL'                    / Type of velocity.
GAINIMAG=  0.9999999776483E-02          / Ratio Image/Signal.
BEAMEFF =  0.8600000143051E+00          / Beam efficiency.
FORWEFF =  0.8600000143051E+00          / Forward efficiency.
EPOCH   =  0.1950000000000E+04          / Epoch of coordinates.
DATE-RED= '15/07/97'                    / Date of reduction.
TTYPE1  = 'MAXIS1  '                    / Number of channels for this spectrum.
TFORM1  = '1J      '                    / [integer]
TTYPE2  = 'SCAN    '                    / Scan number.
TFORM2  = '1J      '                    / [integer]
TTYPE3  = 'TELESCOP'                    / Name of your backend.
TFORM3  = '12A     '                    / [Up to 12 characters]
TTYPE4  = 'TSYS    '                    / System temperature.
TFORM4  = '1E      '                    / [K]
TTYPE5  = 'IMAGFREQ'                    / Image freq at ref channel.
TFORM5  = '1E      '                    / [Hz]
TTYPE5  = 'DELTAV  '                    / Velocity resolution.
TFORM6  = '1E      '                    / [m.s-1]
TTYPE7  = 'TAU-ATM '                    / Atmospheric opacity.
TFORM7  = '1E      '                    / [neper]
TTYPE8  = 'MH2O    '                    / Water vapor content.
TFORM8  = '1E      '                    / [mm]
TTYPE9  = 'TOUTSIDE'                    / Atm temperature near the antenna.
TFORM9  = '1E      '                    / [K]
TTYPE10 = 'PRESSURE'                    / Atm pressure on the antenna.
TFORM10 = '1E      '                    / [hPa]
TTYPE11 = 'ELEVATIO'                    / Elevation.
TFORM11 = '1E      '                    / [deg]
TTYPE12 = 'AZIMUTH '                    / Azimuth.
TFORM12 = '1E      '                    / [deg]
TTYPE13 = 'DATE-OBS'                    / Observing date.
TFORM13 = '23A     '                    / 'dd/mm/yy' for ex.  See below.
TTYPE14 = 'UT      '                    / Universal time at start.
TFORM14 = '1D      '                    / [s]
TTYPE15 = 'LST     '                    / Sideral time at start.
TFORM15 = '1D      '                    / [s]
TTYPE16 = 'OBSTIME '                    / Integration duration.
TFORM16 = '1E      '                    / [s]
TTYPE17 = 'SPECTRUM'                    / Your data.
TFORM17 = '1793E   '                    / [K]
TTYPE18 = 'CRPIX1  '                    / Reference channel for this spectrum.
TFORM18 = '1E      '                    / [real]
END
```

Known keywords

HISTORY 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.

MAXIS Number of axes.

MAXISi Number of elements for axis i. i goes from 1 to MAXIS. The first axis is usually used to define effectively the spectrum. Thus MAXIS1 corresponds to the number of channels. The second and third one can be used to define the coordinates of the spectra when storing a full spectra cube.

CTYPEi The type of physical coordinate on axis i: frequency, velocity, right ascension, time... The unit of the values of CRVALi and CRPIXi are conventionnally defined by the value of the CTYPEi keyword. The authorized values and associated units are

FREQ or FREQUENCY Frequency axis. Unit: [Hz]. Frequency for channel number n is defined by: F(n) = RESTFREQ + CRVALi + (n - CRPIXi) * CDELTi

LAMBDA or WAVELENG wavelength axis. Unit: $[\text{m}^{-1}]$.

VELO or FELO Velocity axis. Unit: $[\text{m.s}^{-1}]$. VELO corresponds to the radio convention while FELO implies channels regularly gridded in frequency but expressed as velocity in the optical convention. CLASS will process VELO and FELO in the same way *** JP. The referential is defined by the second part of the keyword:

- -LSR or CITY or empty second part: Local Standart Rest velocity.
- -HEL Heliocentric (barycentric) velocity.
- -OBS or -TOP The frame of rest of the observer/telescope (topocentric).
- -EAR or -GEO Geocentric velocity

The full keyword may thus looks like: FELO, VELOCITY, VELO-GEO...

RA- or RA Right ascension axis. Unit: [degrees]. Right Ascension RA 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)

DEC- or DEC Declination axis. Unit: [degrees]. Same caveats as for RA.

GLAT Galactic latitude axis. Unit: [degrees]. Same caveat as for RA.

GLON Galactic longitude axis. Unit: [degrees]. Same caveat as for RA.

TIME or UT or UTC Time axis. Unit: [s]. The time since DATE-OBS.

STOKES Stokes axis. The Stokes parameter of the data as defined in the basic paper of Wells et al.

CRPIXi The array location of the reference pixel along axis i. CRPIXi's value may be a fractional number of pixels and/or outside of the limits of the array. This descriptor is optional for degenerated axes (*i.e.* 0 or 1 element).

CRVALi The value of the physical coordinate on axis i at the reference pixel.

CDELTi The increment in physical coordinates along axis i. This descriptor is optional for degenerated axes. Warning: Although there is a difference between the frequency spacing between channels and the frequency resolution of each channel, CLASS currently handles only one value for both.

BSCALE A value of 1 is assumed when it is undefined. The following formula will be applied to the FITS data: `CLASS_spectrum = BZERO + BSCALE * FITS_spectrum`

BZERO A value of 0 is assumed when it is undefined. The following formula will be applied to the FITS data: `CLASS_spectrum = BZERO + BSCALE * FITS_spectrum`

DATAMIN Minimum value of your spectrum. Understood by CLASS but unused.

DATAMAX Maximum value of your spectrum. Understood by CLASS but unused.

BLANK Blanking value. If not specified, -1e38 assumed. `BSCALE` and `BZERO` are applied to the blanking value.

PCOUNT Usually 0. But if you provide your data as Variable Length arrays, then you will have to tweak this value which specifies the size of the variable length array heap, plus a gap (if any). See `THEAP`.

GCOUNT Always 1 in Binary tables.

THEAP Byte offset of heap area. The heap area is where the Variable Length Array data are stored. They are usually stored just after the binary table. The heap does not need to begin at the end of the main table data. The user may choose to provide a gap between the main table and the beginning of the heap. The size of this gap, in bytes, can be found using the value of the `THEAP` keyword. If there is no gap, the value of `THEAP` is `NAXIS1 * NAXIS2`. If there is, the size of the gap is the difference between the value of `THEAP` and `NAXIS1 * NAXIS2`. A value of `NAXIS1 * NAXIS2` is assumed when this keyword is not defined.

BEAMEFF The main-beam efficiency.

FORWEFF or ETAFSS The forward spillover and scattering efficiency.

GAINIMAG Ratio Image/Signal.

MH2O Water vapor content. Unit: [mm].

PRESSURE The ambiant atmospheric pressure. Unit: [hPa].

TOUTSIDE The ambient temperature. Unit [K].

NPHASE Value: from 1 to 8. For multi-phased spectra (i.e. frequency switching) number of phases.

DELTAFj Frequency offset for phase j. Unit: [Hz].

PTIMEj Duration of phase j. Unit: [s].

WEIGHTj Weight of phase j.

TELESCOP A string value giving the telescope name. CLASS truncates this string to the first 12 characters. There is a CLASS convention about TELESCOP, which is not mandatory: TELESCOP contains the name of the instrument, the name of the backend and the name of the subband (if any). For example: "HIFI-HRV-02" for subband 2 of the High Resolution Spectrometer (Vertical polarization) of HIFI (one of the Herschel's instrument).

AZIMUTH The azimuth at TIME. Unit: [degrees]. If the TIME axis is non-degenerated, then this is the azimuth at the TIME of the first pixel on the TIME axis.

ELEVATIO The elevation at TIME. Unit: [degrees]. Same caveat as for AZIMUTH.

DATE-OBS or DATE_OBS A string giving the observation date and optionally the time at the observation start using the new FITS y2k convention. Since CLASS does not handle the TIMESYS keyword yet to indicate the time system, UTC is assumed. Accepted formats are

- 'dd/mm/yy' (1900 is added to yy)
- 'yyyy/mm/dd'
- 'yyyy/mm/ddThh:mm:ss.s', where 'T' stands for the letter 'T' and has no meaning except as a separator between date and time. The trailing time is ignored, you have to provide it through UT, UTC or LST.

DATE-RED or DATE_RED or DATE Same as DATE-OBS, except it is the date at which the data have been reduced.

SCAN or SCAN-NUM or SCAN_NUM Scan number of the observation, according to the CLASS terminology.

SUBSCAN Subscan number of the observation, according to the CLASS terminology.

UT or UTC Universal time of the start of the observation provided as a string or a number. Then the format is

- 'hh:mm:ss.s', when this keyword contains a string.
- The number of seconds after midnight, when this keyword contains a number.

LST Sideral time of the start of the observation. See UT or UTC for information about the format.

OBSTIME or EXPOSURE The effective integration time. Unit: [s]. Necessary for some kind of weighting when adding several spectra.

TSYS The system temperature. Unit: [K]. Necessary for some kind of weighting when adding several spectra.

TAU-ATM or TAU_ATM The opacity at RESTFREQ. CLASS does not handle the opacity in the image band yet.

EPOCH or EQUINOX The epoch of coordinates. Possible values are 1950 or 2000.

OBJECT A string value giving an object name. CLASS truncates this string to the first 12 characters.

LINE Molecular line name, for bookkeeping. CLASS truncates this string to the first 12 characters.

RESTFREQ The observed frequency at the reference pixel of the frequency-like axis. Unit: [Hz].

IMAGFREQ The image sideband frequency corresponding to **RESTFREQ** for double sideband operation. Unit: [Hz].

VELOCITY The velocity at the reference channel. The referential frame must be defined by **VELDEF**.

VELO-LSR or VLSR The velocity at the reference channel, understood as a velocity in the Local Standard Rest frame. Thus, you don't need to use **VELDEF** to specify the frame. Unit: [m.s^{-1}].

VELO-OBS or VELO-TOP The velocity at the reference channel, understood as a velocity in the frame of rest of the observer/telescope. Thus, you don't need to use **VELDEF** to specify the frame. Unit: [m.s^{-1}].

VELO-HEL The velocity at the reference channel, understood as a velocity in the Heliocentric (barycentric) frame. Thus, you don't need to use **VELDEF** to specify the frame. Unit: [m.s^{-1}].

VELO-EAR or VELO-GEO The velocity at the reference channel, understood as a velocity in the Geocentric frame. Thus, you don't need to use **VELDEF** to specify the frame. Unit: [m.s^{-1}].

DELTAV or DELTAVEL 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 n is thus given by **V(n) = VLSR + (n - CRPIXi) * DELTAV**. If **DELTAV** is not provided, then the velocity resolution will be calculated by **DELTAV = -c / RESTFREQ * CDELT1**, where **CDELT1** is the frequency spacing and **c** the celerity of light.

VELDEF The velocity definition and frame (8 characters). The first 4 characters describe the velocity definition. Accepted definitions are

- **RADI** Radio convention.
- **OPTI** Optical convention.
- **RELA** Relativistic convention.

CLASS only handles **RADI** and does not handle **OPTI** or **RELA**. If nothing specified, **RADI** is assumed. This is the list of the values CLASS understands

- **LSR** or **RADI-LSR** Local Standart Rest frame.
- **HEL** or **RADI-HEL** Heliocentric (barycentric) frame.
- **OBS** or **RADI-OBS** Frame of rest of the observer/telescope. Note that **TOP** is not handled here.
- **EAR** or **RADI-EAR** Geocentric frame.

Known columns and data storage

Most of the keywords described above can be used as column names. The value contained in the column prevails on the value contained in the Keyword. This can be useful if you need to store spectra with different lengths in your table: You can use a column named **MAXIS1** containing the number of channels for each spectrum of the table (the trailing bytes requeried to fill the **DATA** column will be ignored then).

Here is the list of authorized column names which can also be keywords: `DATAMAX`, `DATAMIN`, `MAXISi`, `CRVALi`, `CRPIXi`, `CDELTi`, `BEAMEFF`, `FORWEFF` or `ETAFSS`, `GAINIMAG`, `MH2O`, `PRESSURE`, `TOUTSIDE`, `AZIMUTH`, `ELEVATIO`, `DATE-OBS` or `DATE_OBS`, `DATE-RED` or `DATE_RED` or `DATE`, `SCAN` or `SCAN-NUM` or `SCAN_NUM`, `SUBSCAN`, `LST`, `UT` or `UTC`, `TAU-ATM` or `TAU_ATM`, `TELESCOP`, `OBSTIME` or `EXPOSURE`, `TSYS`, `EPOCH` or `EQUINOX`, `OBJECT`, `IMAGFREQ`, `LINE`, `RESTFREQ`, `VELOCITY`, `VELDEF`, `VELO-LSR` or `VLSR`, `VELO-OBS` or `VELO-TOP`, `VELO-HEL`, `VELO-EAR` or `VELO-GEO`, `DELTAV` or `DELTAVEL`.

The following names are exclusively used for columns because their value (your data) can not be factorized

MATRIX or SPECTRUM or SERIES or DATA The spectra intensities. Unit: [K]. This is the only mandatory column. Indeed, why would we make a FITS file without any spectrum? This column must also be unique, *i.e.* only one data column per binary table. The number of elements of this column, specified in `TFORM`, depends on the way you want to store your data (Fixed or Variable Length Arrays). More explanations are available below

WAVE Frequency of each channel. Unit: [Hz]. This column is optional. If it is present, the spectra is considered as "irregularly sampled": each channel has its own frequency and the frequency axis description; In other words, the `RESTFREQ`, `CDELTi`, `CRVALi`) keyword are not used at all even when present. `MAXISi` is still used, since it gives the number of channels. If this column is absent, then the frequency axis will be described in the usual way.

Both the `SPECTRUM` and `WAVE` columns must have the same number of elements to obtain a meaningful result. Those columns may either be filled with

Fixed Length Arrays This may be the easiest way to store the data but it may waste a lot of space if you have spectra with different number of channels. The steps are:

1. Description of the data column.
   (a) First, chose a precision: E for single precision floating point number (4 bytes), D for double precision floating point number (8 bytes).
   (b) Second, browse your collection of spectra to find the one with the highest number of channels (for example, 1793).
   (c) Now you can give `TFORMi` the value 1793E:
   ```
   TTYPE17 = 'SPECTRUM'
   TFORM17 = '1793E    '
   ```

2. Description of the number-of-channel column. If your collection of spectra contains spectra with different sizes, then you must specify the number of channels of each spectrum. To do that, you have to use a column `MAXISi`. `i` corresponds to the number of your frequency/wavelength/velocity axis: usually 1.
   ```
   TTYPE1  = 'MAXIS1 '
   TFORM1  = '1J       '
   ```
   The values stored in this column will override the value given to the keyword `MAXIS1` at the beginning of the bintable header. If all your spectra have the same number of channels, then you don't need a `MAXISi` column. But make sure the `MAXISi` keyword at the beginning of the bintable header has the correct value, which is the number of channels of your spectra.

3. Description of the reference channel. You may like to chose the middle of your spectra to be the reference for frequencies. If you have spectra with different sizes, then the reference channels may be different for each spectrum. To override the value of the `CRPIX1` keyword, you must use a `CRPIX1` column.

   ```
   TTYPE18 = 'CRPIX1  '
   TFORM18 = '1E      '
   ```

   If your reference channel is always the first one, you can forget about this column and give the value 1 to the `CRPIX1` keyword.

4. Filling the olumns. When it comes to fill the column with the data, make sure you are always storing 1793 (for example) elements. If you are to store a spectrum with only 512 channels, then store those 512 elements and pad with 1281 zeroes (single or double precision zeroes, not bytes). You can see here the waste of space when you have a significant amount of spectra much smaller that the widest one.

Variable Length Arrays To save space and avoid filling half of your data columns with padding zeroes, you can use the Variable Length Array format described in the FITS standard (Definition of the Flexible Image Transport System, version 2.1b, section 8.3.5).

The idea is not to store the data directly in the bintable records, but in the heap area located after the bintable. In the `SPECTRUM` or `WAVE` column, you will find an Array Descriptor which basically consists in a pointer to a location in this heap area (an offset), and a number of elements to read. By doing this, no padding is required any longer. The FITS standart defines two kinds of Array Descriptors: P (two 32bits unsigned integers values) and Q (two 64 bits unsigned integers values). CLASS only handles P.

You will not need a `MAXIS1` column in this case, since the number of channels is written in the `SPECTUM` column. You may still need a `CRPIX1` column to define your reference channels.

4.4.3 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.

Chapter 5

Internal Helps

5.1 LAS Language Internal Help

5.1.1 Language

```
            LAS\ Command Language Summary
   (+ indicates a NEW command and - an obsolete command)


   ACCUMULATE          Add R and T observation.
   AVERAGE             Average all the observations of the current index.
   BASE [arg]          Subtract a baseline.
   BOX                 Draw a frame for data.
   CALIBRATE           [Re-]Calibrate a spectrum.
   CATALOG             Load a line catalog
  +CONSISTENCY         Check the consistency of the current index
   COPY                Write the current index into output file.
   DROP num [ver]      Take a scan out of the current index.
   DUMP                List some informations on the R spectrum.
   EXTRACT X1 X2 [U]   Extract a subsection of the R spectrum.
   FILE type name      Define the input/output files.
   FIND                Search the input file for observations.
   FITS                Write or read a fits file
   FOLD                Fold 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 [in|out]       List header information about an ensemble of scans.
  +LOAD                Build a 2D set of spectra from INDEX
   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.
  -RECORD              Obsolete
   SAVE [name]         Save the current parameters.
   SET                 Enter a value for a parameter.
   SHOW Arg            Display some parameter.
```

```
    SPECTRUM [Yoff]     Plot the R observation.
    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 [Obs]         Write R in the output file.
```

5.1.2   ACCUMULATE

```
    LAS\ACCUMULATE  [/RESAMPLE  [NX  Xref  Xval  Xinc  Unit]]  [/NOCHECK
[SOURCE|POSITION|LINE|SPECTROSCOPY|CALIBRATION]]
    LAS\ACCUMULATE /2010
```

ACCUMULATE /2010 uses the old (obsolescent) averaging engine (as of year 2010). No options are allowed in this mode.

ACCUMULATE is used to add a pair of spectra together. The T spectrum is added to the R spectrum using the current weighting function (see SET WEIGHT). On return R contains the sum spectrum, and T the observation previously in R. To add more than 2 spectra, you should build an index and AVERAGE it in a single step instead (this will prevent cumulative resamplings, respect the respective weights, and preserve at best the noise properties).

The spectra are added channel by channel by default, if possible. Else the option /RESAMPLE must be invoked (see subtopic for details).

ACCUMULATE checks the consistency between R and T, in particular the positional coincidence and calibration homogeneity according to SET [NO]MATCH and SET CALIBRATION respectively. The option /NOCHECK allows to disable some (using one or several names) or all (no name given) of these checks.

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. For other weightings, ACCUMULATE computes the average of the two spectra.

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

ACCUMULATE /RESAMPLE

```
    LAS\ACCUMULATE /RESAMPLE [NX Xref Xval Xinc Unit]
```

Without this option, R and T will be added channel by channel. A consistency test will check that they both have the same X axes. SET ALIGN values are ignored in this case. Prefer this solution when possible,

since it is faster and safer.

With the /RESAMPLE option, the X axis of the output sum will be automatically computed according to the alignment mode (SET ALIGN). In particular, the output spectrum has the coarsest resolution of the two input spectra. Those are then internally resampled and added on this new axis.

A custom output X axis can be given with the syntax /RESAMPLE NX Xref Xval Xinc Unit. The input spectra are internally resampled and added on this axis. See HELP RESAMPLE for possible uses and caveats. The output resolution can not be better than the coarsest resolution of the input spectra to avoid oversampling.

Remember that if it is enabled, the resampling is time consuming and has non trivial effects on the spectra noise.

5.1.3 AVERAGE

```
    LAS\AVERAGE  [/RESAMPLE [NX  Xref  Xval   Xinc   Unit]]   [/NOCHECK
[SOURCE|POSITION|LINE|SPECTROSCOPY|CALIBRATION]]
    LAS\AVERAGE /2010 [/NOMATCH]
```

AVERAGE /2010 uses the old (obsolescent) averaging engine (as of year 2010). The only option allowed in this mode is /NOMATCH.

Average all the spectra of the current index using the current weighting function (see SET WEIGHT). The spectra are added channel by channel by default, if possible. Else the option /RESAMPLE must be invoked (see subtopic for details).

Bad channels are handled according to SET BAD. A sum may be interrupted by <^C>, but the result is then undefined.

The command checks the consistency of the index, in particular the positional coincidence and calibration homogeneity according to SET [NO]MATCH and SET CALIBRATION respectively. The option /NOCHECK allows to disable some (using one or several names) or all (no name given) of these checks.

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

AVERAGE /RESAMPLE

```
     LAS\AVERAGE /RESAMPLE [NX Xref Xval Xinc Unit]
```

Without this option, all the spectra will be added channel by channel. A consistency test will check that they both have the same X axes. SET

ALIGN values are ignored in this case. Prefer this solution when possible, since it is faster and safer.

With the /RESAMPLE option, the X axis of the output sum will be automatically computed according to the alignment mode (SET ALIGN). In particular, the output spectrum has the coarsest resolution of the input spectra. Those are then internally resampled and added on this new axis.

A custom output X axis can be given with the syntax /RESAMPLE NX Xref Xval Xinc Unit. The input spectra are internally resampled and added on this axis. See HELP RESAMPLE for possible uses and caveats. The output resolution can not be better than the coarsest resolution of the input spectra to avoid oversampling.

Remember that if it is enabled, the resampling is time consuming and has non trivial effects on the spectra noise.

5.1.4 BASE

```
LAS\BASE [Deg] [/PLOT [Ipen]] [/INDEX] [/OBS] [/CONTINUUM [Flux]]
LAS\BASE SINUS Amplitude Period Phase [/PLOT [Ipen]]
LAS\BASE LAST [/PLOT [Ipen]]
```

When working on single spectra, 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. Zones defined by the SET WINDOW command are not used to fit the baseline. The T content may be recovered with SWAP.

BASE SINUS Amplitude Period Phase
(Phase is in the same unit as the period, usually km/s or MHz) Subtract a sinusoid. Minimization will be done including also a linear baseline in addition to the sinusoid.

BASE LAST
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. 0th and 1st order polynomials, and sinus fit are extrapolated if the R spectrum extends beyond the LAST spectrum limits. 2nd order (and higher) polynomials are not extrapolated and the edge values are used beyond the LAST spectrum limits.

BASE /PLOT
Plot this baseline after the minimization, using the Ipen-th pen properties. The pencil 2 (red at startup) is used by default.

BASE /INDEX
Subtracts a baseline to all records that have been loaded. This option is not needed if SET ACTION INDEX. /PLOT and /INDEX options are incom-

patibles.

BASE /OBS
indicate that the baseline is to be fitted to the current R buffer ob-
servations. This option is not needed if the action level is OBS (de-
fault). /OBS and /INDEX are incompatible.

BASE /CONTINUUM [Flux]

1) Divide the spectrum by the baseline (rather than subtracting it), and
then 2) multiply by the average flux. This is useful to keep the contin-
uum level, when continuum is present and variations of atmospheric emis-
sion are properly subtracted out by fast enough switching. This is of
course the case for correlation spectra (e.g. coming from an interferom-
eter). 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.

5.1.5   BOX

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

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.

Arg1(2) modify the labelling of the lower (left) axis
         = P for parallel labels (default for X axis)
         = O for orthogonal labels (default for Y axis)
         = N for no labels
Arg3   indicates that the ticks are In or Out ofthe box
Arg4   controls the labelling of the upper X axis and
         may have the same values as Arg1 or 2 (P, O, N).

The default is BOX P O I

/UNIT
    If 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. /INDEX produces a box suited for 2D color
    plot with a wedge (the Y-left axis gives the number of spectra).

/OBS
    Plot a box for a single spectrum.

```
/INDEX
    Plot a box suited for 2D display: Y axis is the  temperature  scale,
    and  Z  axis  is the observation number. All three axis (X, Y and Z)
    may be defined by using the SET MODE command.
```

5.1.6   CALIBRATE

```
LAS\CALIBRATE [KEY val] [KEY val] ...
    No arguments: Display the calibration parameters.
    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_CHOP-
PER 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.
```

5.1.7   CATALOG

```
LAS\CATALOG [Filename]
```

```
Read an ASCII file of lines (frequency in GHz and  name)  into  the  SIC
structure  named  CLASS%LINE. When the filename is missing, CATALOG uses
by default the ASTRO line catalog. Each row of the  line  file  contains
the description of a line as follows:
```

```
    115.271204        12CO-1-0
```

```
Comment lines start with the exclamation mark (!).
```

```
The structure has 3 members:
    CLASS%LINE%N          [      integer] number of lines read
    CLASS%LINE%FRES[N]  [       double] line frequency in MHz
    CLASS%LINE%NAME[N]  [character*12] line name
```

5.1.8   CONSISTENCY

```
LAS\CONSISTENCY [/NOCHECK [keyword1] [keyword2] ... ]
```

```
Check the consistency of the current index according to particular head-
er parameters. By default, check for SOURCE, POSITION, OFFSET, LINE  and
SPECTROSCOPY  consistency.  CALIBRATION is checked if SET CALIBRATION is
active (see HELP CALIBRATION for  details).  The  position  OFFSETs  are
checked if SET MATCH is ON (use SET NOMATCH to deactivate it).
```

```
The  consistency  is  automatically  tested  with some commands (AVERAGE
/NEW, ACCUMULATE /NEW, STITCH, LOAD, TABLE).
```

```
/NOCHECK
     Without keyword, this option disables all six checks.  With  a  key-
     word,  disable the relevant checks. Current keywords are SOURCE, PO-
     SITION, OFFSET, LINE, SPECTROSCOPY and CALIBRATION.
```

5.1.9 COPY

```
LAS\COPY [/SORTED]
```

Write all the observations in current index into output file. For effi-
ciency reasons, COPY refuses to copy a sorted index (SET SORT is other
than NONE), but user can enforce the copy with the /SORTED option.

5.1.10 DROP

```
LAS\DROP Obs [Ver]
```

Removes 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. This commands has an effect only on
the current index, i.e. until a new index is built with a FIND.

5.1.11 DUMP

```
LAS\DUMP Argument [Buffer]
LAS\DUMP /SECTION [Name] [R|T|P]
```

With no argument, lists the content of all present header sections of
the current R buffer. Arguments are:
```
  ADDRESSES     Lists addresses
  DATA [R|T|P] Lists in addition the data values of buffer R, T or P
  FILE          Lists information about the input and output files
  INDEX         Lists information about the current index
  MEMORY        Lists the memory usage of the Class buffers
  OTF           Lists OTF informations
  PLOT          Lists current plot definition
```

```
/SECTION [Name] [R|T|P]
     Lists parameters of a given section. Default is all for R buffer.
```

5.1.12 EXTRACT

```
LAS\EXTRACT [X1 X2 [Unit]] [/INDEX]
```

In its default mode, EXTRACT copies R into T, extracts a subsection of
R, and save the result in R. If the option /INDEX is present, or if SET
ACTION is INDEX, a subsection of each observation in the index is ex-
tracted and saved in the output file.

Without arguments, the cursor is invoked (if available) to define the
subsection range. If the arguments X1 and X2 are present, the subsection
is defined by this range, in the given Unit (if any) or by default in
the current X axis unit (SET UNIT). X2 can be indifferently greater or
lower than X1, and Unit is a single character in the four C)hannel,
V)elocity, F)requency or I)mage. Note that if Unit is F or I, the number
of extracted channels can vary for two similar observations (same setup)
because of different Doppler factors.

If the range overlaps the input spectrum limits, a warning is raised and
the missing data is filled with the blanking value. Requesting a range
off the input spectrum is an error. The command FIND /FREQUENCY can help
you to build an index with spectra overlapping one frequency or one
range of frequencies.

On return, the observation header is consistently updated: the reference
channel is shifted and number of channels updated. The PLOT section
(which remembers the previous plotting limits) is disabled because it
probably does not makes sense for the extracted data array.

Only regularly sampled X axes are supported.

5.1.13   FILE

```
      LAS\FILE Mode Name [SINGLE|MULTIPLE] [/CONVERT] [/OVERWRITE]
```

Selects the input and output files.
FILE IN name   open an existing file as input
FILE OUT name   open an existing file as output
FILE BOTH name   select the same file for input and output.
FILE UPDATE name   open a file for Updates only.
FILE OUT name Type   creates a new output file, of the specified type
                      Option /OVERWRITE will force overwriting an existing fil
FILE OUT name Type /CONVERT   converts an existing output file to
                              the specified type

Files have two different Types: SINGLE or MULTIPLE.

File of type MULTIPLE allows to have several version of the same spec-
trum (same ObservationNumber). This type is the "historical" one for
CLASS. It can be opened in all modes.
For file of type SINGLE, the ObservationNumber is a unique identifier of
the spectrum: only one version exists. As a consequence a file of this
type cannot in general be opened both IN and OUT. Such files can be
opened in mode UPDATE, which allows spectra to be modified in place,
typically via a TAG or UPDATE command after a MODIFY. Command WRITE is
not allowed in this mode.

This new type has been introduced to simplify and speed up On-The-Fly processing.

The default extension is .30m but can can be specified using command SET EXTENSION.

5.1.14 FIND

```
    LAS\FIND [APPEND|NEW_DATA|UPDATE] [/ALL] [/LINE name] [/FREQUENCY f1
f2] [/NUMBER n1 n2] [/SCAN s1 s2] [/OFFSET o1 o2] [/SOURCE name] [/RANGE
w e s n] [QUALITY q] [/TELESCOPE name] [/SUBSCAN s1 s2] [/ENTRY]  [/SEC-
TION name]
```

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 name      search by line name
/FREQUENCY f1 f2 search by signal or image frequency value or range
/NUMBER n1 n2   search for the specified range of observation numbers
/ENTRY  n1 n2   search for the specified range of entry numbers
/SCAN s1 s2     search for the specified range of scan numbers
/SUBSCAN s1 s2  search for the specified range of subscan numbers
/OBSERVED d1 d2 search for these dates (JJ-MMM-YYYY)
/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 name    search by source name
/TELESCOPE name search for the telescope used
/SECTION name   search for observations which have this section defined
```

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.

FIND APPEND

```
    LAS\FIND APPEND [/OPT1 ...]
```

FIND by default resets the current index. Found observations may be appended to the current index by specifying the argument APPEND. A compression occurs to avoid duplication of observations in the index.

FIND NEW_DATA

```
    LAS\FIND NEW_DATA [/OPT1 ...]
```

Argument NEW_DATA can be used to find the new (and only new) observa-
tions (if any) present in the input file. FIND options or equivalent SET
tunings apply. If no new observations are present, the current index
will be empty.

FIND NEW_DATA does not wait for new data, but you can use the command
NEW_DATA to do so.

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.

FIND UPDATE

    LAS\FIND UPDATE [/OPT1 ...]

The input index will first be appended with the new observations (if
any) present in the input file. This saves time because a (much) smaller
part of the input file is actually read. The current index is then reset
and recomputed from the input index (i.e. with old and new observa-
tions), as usual.

FIND UPDATE does not wait for new data, but you can use the command
NEW_DATA to do so.

This keyword can be omitted if the SET FIND mode is UPDATE.

5.1.15   FITS

    LAS\FITS READ File[.fits]
    LAS\FITS WRITE File[.fits] /MODE   SPECTRUM|INDEX   [/BITS   Nbits]
[/CHECK]

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 head-
er information might be incomplete when doing so. Alternatively, FITS
cubes can be converted to the Gildas Data Format by the command VEC-
TOR\FITS, and then imported as a Class index by the command ANALYSE\LMV.

Or WRITE the File.fits FITS file from the CLASS internal data format.
/MODE is a mandatory option which can take the following values as argu-
ment:
  SPECTRUM a simple FITS file is written from the current spectrum in
           memory is written. The number of bits per intensity value
           used when writing the FITS file may be controlled through the
           /BITS optional argument. It can be: 16 or I*2 (2 bytes inte-
           ger), 32 or I*4 (4 bytes integer) or -32 or R*4 (4 bytes re-
           al). Default is 32.

> 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 intensity values (and irregularly sampled X axis
> values, if any) are always written as 4 bytes real.

The default mode and number of bits can also be specified through the
commands:
```
SET FITS MODE SPECTRUM|INDEX
SET FITS BITS Nbits
```
The option /CHECK will print the FITS HEADER on the terminal.

5.1.16 FOLD

    LAS\FOLD

Folds a frequency-switched spectrum.

5.1.17 GET

    LAS\GET [N [version]|FIRST|LAST|NEXT|PREVIOUS|ZERO]

GET copies R into T, and loads the Observation number N in R. The cur-
rent 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.

Instead of an observation number, several keywords can also be used:
  - FIRST: the first Observation of the current index is loaded,
  - LAST: the last Observation of the current index is loaded,
  - NEXT: the next Observation in the current index is loaded,
  - PREVIOUS: the previous Observation in the current index is loaded,
  - ZERO: the internal Observation counter is reset to 0, so that
              next GET NEXT will get the first observation. In this
              unique case, R and T are not modified.

5.1.18 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 neces-
sarily 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)
    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.
```

5.1.19 IGNORE

```
      LAS\IGNORE List_of_Observations [/SCAN iscan]
```

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

```
/SCAN iscan
    Ignore all observations with scan number 'iscan'.
```

5.1.20   LIST

LAS\LIST [IN|OUT] [/BRIEF] [/LONG] [/SCAN] [/OUTPUT File] [/TOC]
[/VARIABLE VarName]

List header information about an ensemble of Observations. LIST is used
for a quick look to Observation headers, in a more or less detailed for-
mat. IN or OUT keyword specifies the file to be listed. If absent, the
current index is listed. A medium-sized format is used by default.
```
LIST IN        List the whole input file (all versions, see FIND /ALL)
LIST OUT       List the whole output file (all versions)
/BRIEF         Brief format
/LONG          Long format
/SCAN          Scan based compact format
/SCAN /BRIEF   Scan numbers followed by the number of spectra in each scan
/TOC           Display the table of contents. See subtopic for details.
/OUTPUT File   Send the list to a file
```

Some options are particularly well suited for large ensemble of spectra:
/SCAN displays the index on a scan number basis (defined by a unique
SOURCE, LINE, TELESCOPE and SCAN number), /SCAN /BRIEF gives the list of
scan numbers with their total number of spectra. /SCAN and /LONG options
are incompatible.

LIST /TOC

LAS\LIST [IN|OUT] /TOC [Attr1 Attr2 ... AttrN] [/VARIABLE VarName]

LIST /TOC displays the Table Of Contents of the current index, or of the
input (LIST IN) or output file (LIST OUT). A summary of the different
values of each attribute is made, plus a final summary of all the combi-
nations (setups). The attributes can be:
```
    - SOURCE
    - LINE
    - TELESCOPE
    - OFF1
    - OFF2
```

```
        - ENTRY
        - NUMBER
        - BLOCK
        - VERSION
        - KIND
        - QUALITY
        - SCAN
        - SUBSCAN
```
By default, SOURCE, LINE and TELESCOPE are used.

The table of contents is also saved in a SIC, user-defined structure
thanks to the option /VARIABLE (default is TOC%). The content of the
structure is updated after each call to the command LIST /TOC. One must
take care that subsequent calls to FIND can lead to values in TOC% in-
consistant with the current index.

5.1.21 LOAD

```
        LAS\LOAD [/NOCHECK keyword1 [keyword2 [keyword3] ...]]
```

Pack the whole current index in a 2D array. By default, check the con-
sistency of the index (see CONSISTENCY). The option /NOCHECK allows to
ignore some inconsistency, according to the optional keywords.   Current
keywords are SOURCE, POSITION, LINE and SPECTROSCOPY.

5.1.22 MODIFY

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

This is a general purpose command to edit and change the current obser-
vation 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 infor-
mation in the observation header is wrong). They should be used with
caution.

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

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.

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.

MODIFY BEAM_EFF

```
LAS\MODIFY BEAM_EFF Value
LAS\MODIFY BEAM_EFF /RUZE B0 Sigma
```

Change the beam efficiency.

The first syntax allows to define the new beam efficiency from a single
value for the whole spectrum. If the beam efficiency previously had a
non zero value, the spectrum intensities and affected header parameters
are rescaled according to the old and new values. Otherwise, only the
beam efficiency is set to the new value.

The second syntax allows to use the Ruze's equation, which is more suit-
able for large band spectra:
```
Beeff(lambda) = B0*exp(-(4*pi*Sigma/lambda)**2)
```
B0 is the scaling factor with no units (between 0 and 1), and Sigma the
width factor in microns.
If the beam efficiency previously had a non zero value, a rescale factor
is evaluated for each channel and is applied to the intensities, the new
beam efficiency is set to the one at the center of the spectrum, and the
affected header parameters are rescaled according to this new value.
Otherwise, only the beam efficiency is set to this new value.

MODIFY DOPPLER

```
LAS\MODIFY DOPPLER [SIGN|*|Value]
```

Change the Doppler factor according to the CLASS convention (-V/c). With
no argument, checks whether the header value is consistent with the
CLASS value computed according to the date and observatory. The checking
precision corresponds to 0.6 m/s in V. A logical SIC variable,
DOPPLER_PB, is set to true if the values disagree. If an argument is
present:

  SIGN   change the sign of the Doppler factor but do not recompute it,
  *   compute and update the Doppler factor according to the observ-
         ing date-time and the observatory position (this position can
         be customized with SET OBSERVATORY),
  Value set the Doppler factor to <Value>

MODIFY FREQUENCY

      LAS\MODIFY FREQUENCY Rfreq

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

MODIFY IMAGE

      LAS\MODIFY IMAGE Rimage

Change the image frequency (in MHz).

MODIFY LINENAME

      LAS\MODIFY LINENAME "New Name"

Modify the line name (for bookeeping).

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 (usu-
ally the position of the first observation added).

MODIFY POSITION

      LAS\MODIFY POSITION Lambda Beta

Change the central position for the source to (lambda_0,beta_0). The
offsets are also modified, as to be consistent with the new central po-
sition. This command is usefull when different parts of one source are
mapped relative to several reference positions. Only RADIO projection is
handled.

If the coordinate system is EQuatorial:
     lambda = lambda_0 x 12 /pi
     beta   = beta_0   x 180/pi
If the coordinate system is GALactic:
     lambda = lambda_0 x 180/pi
     beta   = beta_0   x 180/pi

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.

MODIFY SOURCE

        LAS\MODIFY SOURCE source_name

Modify the source name.

MODIFY SYSTEM

        LAS\MODIFY SYSTEM System

Force the System in which the center of projection (central position of
the source) is defined. Valid systems are GALACTIC, EQUATORIAL, HORIZON-
TAL and UNKNOWN. This command is typically used for "foreign" data when
the SYSTEM information has not been decoded correctly by the conversion
package. No correction is applied on spectrum position.

MODIFY TELESCOPE

        LAS\MODIFY TELESCOPE Name

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

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.

MODIFY WIDTH

        LAS\MODIFY WIDTH Rfres

Change the frequency resolution (in MHz).

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 cor-
rectly 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)

5.1.23 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:
   SIC\LET RY Fact*RY

5.1.24 NEW_DATA

   [LAS\]NEW_DATA [Interval]

Waits until new observations have been written to the input file, and
then return. No modification of the input or current index is done. It
is then expected to use FIND NEW_DATA|UPDATE to rebuild the indexes with
these new observations (see corresponding HELP for details).

The waiting time between 2 check can be customized as first argument (in
seconds, default 5).

5.1.25 PLOT

   LAS\PLOT [/INDEX] [/OBS]

PLOT /OBS
  Display the Observation in R. It is equivalent to the following se-
  quence of commands:
    CLEAR BOX SPECTRUM TITLE
  The type of PLOT can be specified by SET PLOT. The limits for the X
  and Y axes are set by SET MODE X and SET MODE Y.

PLOT /INDEX
  Equivalent to the following sequence of commands:
    CLEAR BOX /INDEX SPECTRUM /INDEX TITLE /INDEX
  Plot a 2-dimensional image (built with the LOAD command) with veloc-
  ity/frequency for the X axis, index number for the Y-axis, and in-
  tensity being rendered as grey/color scale. Use SET MODE Y to choose
  the color lower and upper limits. Entering command POPUP after PLOT
  /INDEX enables displaying a single spectrum, selected by clicking
  with the middle button of the mouse. Use SET MODE Z to control the
  second dimension of the plot (e.g. number of entries).

Default is /OBS. The default can be modified with SET ACTION.

5.1.26 RECORD

```
LAS\RECORD k
```

Obsolete.

5.1.27 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.

5.1.28 SET

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

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

SET ACTION

```
LAS\SET ACTION I[ndex] O[bservation]
```

Defines the default action level: current index or current observation
(R buffer). Default is OBServation.

SET ALIGN

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

Defines the way the spectra to be added are aligned (Type) and combined
(Range). Possible combinations are:
Type: C[hannel]   for spectra and drift
      V[elocity]   for spectra only
      F[requency]   for spectra only
      P[osition]   for drift only
Range: I[ntersect] or C[omposite]
Default is SET TYPE C I. Note: This means that continuum drifts in oppo-
site directions are not properly added: ALIGN Position must be specified
for that.

SET ANGLE

```
LAS\SET ANGLE Unit
```

Specify the angle unit for offsets. Unit is one of: RAD[ian], DEG[ree],
MIN[ute of arc], SEC[ond of arc], MAS[milli second of arc]. Default is
SECond. The ANGLE unit is also used to display Continuum drifts.

SET BAD

        LAS\SET BAD Check

Specify the way the commands which have to mix channels behave in the
case of bad channels. There are two families of such commands:
  - resampling (part of) commands: RESAMPLE, and optionally ACCUMULATE,
     AVERAGE, and STITCH if resampling is needed,
  - addition (part of) commands: ACCUMULATE, AVERAGE, and STITCH.

Check may be OR or AND. Default is OR, i.e. the output channel is bad if
any of the input channels is bad. AND indicates to use all the non-bad
channels (if any) to compute the output channel.

SET BASELINE

        LAS\SET BASELINE arg

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

SET BOX

        LAS\SET BOX L[andscape] P[ortrait] D[efault]

Calls the GREG1\SET PLOT and GREG1\SET VIEWPORT commands with default
values suited for CLASS plots. If no keyword, restore the default box:
SET VIEW 0.1 0.95 0.15 0.70. The box size can be modified using the
GREG1\ command directly as for instance
        LAS> g\set view 0.2 0.3 0.2 0.3

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 the command CONSISTENCY (and the other commands
using it implicitly) to verify that the calibration is consistant. They
are also used by command WRITE which writes the corresponding informa-
tion 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.

SET CURSOR

```
LAS\SET CURSOR Arg
```

SET CURSOR IS OBSOLETE. Cursor availability is detected automatically, and can be deactived if needed for some commands by a /NOCURSOR option.

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.

SET DEFAULT

```
LAS\SET DEFAULT
```

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

SET DISPLAY

```
     The following SET Parameter commands affect data display:
LAS\SET ACTION action_level
LAS\SET FORMAT Type
LAS\SET LEVEL interactive_min_value logfile_min_value (obsolete)
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]
```

SET EXTENSION

```
LAS\SET EXTENSION Ext
```

Defines the default extension for input and output files. Default is '.30m'.

SET FIND

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

FIND will check or not for new entries in the input file (see FIND UPDATE). Default is no implicit update (NOUPDATE).

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 off-
sets), LONG or FULL which is similar to the long format, but also dis-
plays the list of original Scans. The default is LONG. 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 cor-
responding 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.

SET FREQUENCY

        LAS\SET FREQUENCY Freq1|* [Freq2|*] [SIGNAL|IMAGE]

Add a selection by frequency for the command FIND. This criterion is
suited only for spectroscopic data.

  - SET FREQUENCY Freq1
      select all observations which frequency range intersects the given
      frequency,
  - SET FREQUENCY Freq1 Freq2
      select all observations which frequency range intersects at least
      partially the given range,
  - SET FREQUENCY * Freq2
      select all observations which frequency range is at least partial-
      ly under Freq2,
  - SET FREQUENCY Freq1 *
      select all observations which frequency range is at least partial-
      ly above Freq1,
  - SET FREQUENCY *
      select all observations i.e. disable the selection by frequency.
      This is the default.

An optional keyword can be added after the values to indicate whether the selection should be performed by SIGNAL or IMAGE frequencies. Default is SIGNAL.

Users should be aware that this selection criterion requires to read more informations in the observation header, and thus has a computation cost which can become important for large datasets.

SET LEVEL

```
LAS\SET LEVEL interactive_min_value logfile_min_value
```

SET LEVEL IS OBSOLETE AND HAS NO EFFECT ANYMORE.
It can be mimicked with the SIC MESSAGE command (see HELP for details).
To alter the verbosity of CLASS commands to terminal and to message file, one should modify the message filters of the CLASS package.

For example:
```
   LAS90> SIC MESSAGE CLASS S=FEW ! Change the on-screen CLASS filter
   LAS90> SIC MESSAGE CLASS      ! Print the two CLASS filters
   class   on-screen active   filter: FEW------
   class   to-mesfile active   filter: FEWRI--CU
```
This keeps only the most important messages (Fatal, Errors, and Warnings) to the terminal, whereas Results, Informations and Commands are also printed in the message file by default.

SET LINE

```
LAS\SET LINE Name
```

FIND will select all observations according to the specified line name. Default is *. Wildcards are accepted at any position in the line name (e.g. NAM*, N*ME or N*M*): they match 0 or more characters.

SET MASK

```
LAS\SET MASK [ml1 ml2 [...]] [/VAR array]
```

Defines a mask for the Gaussian fits by its boundaries. If available, the cursor will be used to specify missing arguments, unless option /NOCURSOR is present. When using the cursor, enter: N (or Left mouse button) to enter a value, C (or Middle button) to cancel the last entry, H for help, E (or Right button) for exit. The boundaries are in current unit. Default is no mask. The boundaries are in current unit. See DRAW MASK to display the mask(s).

```
SET MASK /VAR mask_array
   Define the masks from rank 1 array mask_array[1+2*NMASK]. The array
```

```
    must be of the following form:
      mask_array[1]=NMASK       number of masks
      mask_array[2*i:2*i+1]    boundaries of mask num. 'i'
```

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.

SET MATCH

```
      LAS\SET MATCH Tol
```

Turn on the checking of position-matching in CONSISTENCY (and the other
commands using it). 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.

SET MODE

```
      LAS\SET MODE  X | Y | Z | ALL  [mode]
```

Select the scale mode for axis X, Y or Z, or all. The mode can be:
```
  AUTO    : limits as found in the observation header (DEFAULT),
  TOTAL   : all data plotted,
  CURRENT : fix the current values in use (whatever if they were
            Automatic, Total, or already Fixed), so that the next
            spectra plotted will reuse the same
  [void]: i)  limits are entered explicitly by the user
                  LAS\set mode x x1 x2
          ii) limits are entered interactively when no mode is
              given (if the cursor is available)
```

For 2D data (built with the LOAD command), SET MODE Y defines the color
range. The second dimension range (e.g. observation number) is defined
with SET MODE Z.

SET NOMATCH

```
      LAS\SET NOMATCH
```

Turn off checking of position-matching in CONSISTENCY (and the other
commands using it. This does not change the tolerance for the other
functions like FIND and MAP.

SET NUMBER

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

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

SET OBSERVATORY

```
LAS\SET OBSERVATORY *
LAS\SET OBSERVATORY Name Longitude Latitude Altitude
```

If OBSERVATORY is *, Class will guess the observatory from the TELESCOPE name. This is the default.

If an observatory name is provided with its longitude, latitude (both sexagesimal strings) and altitude (in kilometers), Class will use those values independently of the TELESCOPE name. This is useful in particular for observatories unknown by Class.

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 * *.

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).

SET PLOT

```
LAS\SET PLOT type level
```

Set the plotting type and level. Level is I[ndex] or S[can]. If level is S, then type is N[ormal] or H[istogram]. Normal is considered to be 50% faster plot, but Histogram gives a better reading of the channel positions and values. Default is Histogram.

SET POSANGLE

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

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

searches.

SET PROCESSING

The following SET Parameter commands control data processing op-
tions:
```
LAS\SET ALIGN Type Range
LAS\SET BAD Check
LAS\SET BASE arg
LAS\SET CALIBRATION [Beam_Tol [Gain_Tol]] or OFF
LAS\SET MASK [ml1 ml2 [...]] [/VAR array]
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 [...]]] -
                [/POLYGON [N] [filename1...filenameN] [/VAR array]]
```

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 sys-
tem 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.

SET RANGE

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

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

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
* *.

SET SELECTION

The following SET Parameter commands affect data selection in FIND:
```
LAS\SET ANGLE Unit
```

```
LAS\SET LINE Name
LAS\SET FREQUENCY Freq
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
```

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 * *.

SET SUBSCAN

```
LAS\SET SUBSCAN s1 s2
```

FIND will select all observations with subscan numbers between s1 and s2. The subscan number reflects the record number during an On-The-Fly acquisition. Default is * *.

SET SORT

```
LAS\SET SORT keyword
```

Defines the sorting when building the INDEX with FIND. The sorting is in ascending order. Default sorting is recovered with SET SORT NONE.

Keyword is one of:
```
    BETA  KIND  LAMBDA  LINE NUMBER NONE OBSERVED QUALITY SCAN SUBSCAN
    SOURCE TELESCOPE VERSION
```

SET SOURCE

```
LAS\SET SOURCE Name
```

FIND will select all scans according to the specified source name.   De-
fault is *. Wildcards are accepted at any position in the source name
(e.g. NAM*, N*ME or N*M*): they match 0 or more characters.

SET SYSTEM

        LAS\SET SYSTEM Type [Equinox]

Specify the coordinate system to be used by the current R observation
and next ones loaded in memory. 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 2000.00.
Coordinates and offsets are automatically converted to the specified co-
ordinate system and precessed to the specified epoch if needed.

SET TELESCOPE

        LAS\SET TELESCOPE Name

FIND will select all scans according to the specified telescope name.
Default is *. Wildcards are accepted at any position in the telescope
name (e.g. NAM*, N*ME or N*M*): they match 0 or more characters.

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 Spec-
     tral 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.

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 Ve-
locity. For Continuum drifts, this unit is ignored, but the angular unit
specified by SET ANGLE is used.

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 ar-
gument. 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.
They are defined below:

              SECTIONS Information about present sections
LOGICAL   R%HEAD%PRESEC(ISEC) [   ] List of present sections
Codes for ISEC can be found in CLASS%CODE%SEC%.

              GENERAL General parameters, always present:
INTEGER*8   R%HEAD%GEN%NUM   [   ] Observation number
INTEGER*4   R%HEAD%GEN%VER   [   ] Version number
CHARACTER*12 R%HEAD%GEN%TELES   [   ] Telescope name
CHARACTER*12 R%HEAD%GEN%CDOBS   [day] Date of observation
CHARACTER*12 R%HEAD%GEN%CDRED   [day] Date of reduction
INTEGER*4   R%HEAD%GEN%TYPEC   [code] Type of coordinates
INTEGER*4   R%HEAD%GEN%KIND   [code] Type of data
INTEGER*4   R%HEAD%GEN%QUAL   [code] Quality of data
INTEGER*8   R%HEAD%GEN%SCAN   [   ] Scan number
REAL*8   R%HEAD%GEN%UT   [rad] UT of observation
REAL*8   R%HEAD%GEN%ST   [rad] LST of observation
REAL*4   R%HEAD%GEN%AZ   [rad] Azimuth
REAL*4   R%HEAD%GEN%EL   [rad] Elevation
REAL*4   R%HEAD%GEN%TAU   [neper] Opacity
REAL*4   R%HEAD%GEN%TSYS   [K] System temperature
REAL*4   R%HEAD%GEN%TIME   [s] Integration time
INTEGER*4   R%HEAD%GEN%XUNIT   [code] X unit, if X coordinate section is p
Codes for TYPEC, KIND, QUAL and XUNIT can be found in SIC%CODE%COORD%,
SIC%CODE%KIND%, CLASS%CODE%QUAL% and SIC%CODE%XUNIT% respectively.

                 POSITION Position information:
CHARACTER*12 R%HEAD%POS%SOURC   [   ] Source name
REAL*4   R%HEAD%POS%EPOCH   [code] Epoch of coordinates
REAL*8   R%HEAD%POS%LAM   [rad] Lambda
REAL*8   R%HEAD%POS%BET   [rad] Beta
REAL*4   R%HEAD%POS%LAMOF   [rad] Offset in Lambda
```

```
REAL*4         R%HEAD%POS%BETOF   [ rad] Offset in Beta
INTEGER*4      R%HEAD%POS%PROJ    [code] Projection system
REAL*8         R%HEAD%POS%SLOP    [ rad] Lambda of descriptive system
REAL*8         R%HEAD%POS%SBOP    [ rad] Beta of descriptive system
REAL*8         R%HEAD%POS%SKOP    [ rad] Angle of descriptive system
Codes for PROJ can be found in SIC%CODE%PROJ%.


               SPECTRO Spectroscopic information (for spectra)
CHARACTER*12   R%HEAD%SPE%LINE       [    ] Line name
REAL*8         R%HEAD%SPE%RESTF      [ MHz] Rest frequency
REAL*8         R%HEAD%SPE%IMAGE      [ MHz] Image frequency
INTEGER*4      R%HEAD%SPE%NCHAN      [    ] Number of channels
REAL*8         R%HEAD%SPE%RCHAN      [    ] Reference channels
REAL*8         R%HEAD%SPE%FRES       [ MHz] Frequency resolution
REAL*8         R%HEAD%SPE%FOFF       [km/s] Frequency offset
REAL*8         R%HEAD%SPE%VRES       [km/s] Velocity resolution
REAL*8         R%HEAD%SPE%VOFF       [km/s] Velocity at reference channel
REAL*4         R%HEAD%SPE%BAD        [    ] Blanking value
INTEGER*4      R%HEAD%SPE%VTYPE      [code] Type of velocity
REAL*8         R%HEAD%SPE%DOPPLER    [    ] Doppler factor = -V/c (CLASS conven
Codes for VTYPE can be found in SIC%CODE%VELO%.


               BASE Baseline information (for spectra of drifts)
PARAMETER      MWIND=100
INTEGER*4      R%HEAD%BAS%DEG             [       ] Degree of last baseline
REAL*4         R%HEAD%BAS%SIGFI          [      K] Sigma
REAL*4         R%HEAD%BAS%AIRE          [K.km/s] Area under windows
INTEGER*4      R%HEAD%BAS%NWIND          [       ] Number of line windows
REAL*4         R%HEAD%BAS%W1(MWIND)      [  km/s] Lower limits of windows
REAL*4         R%HEAD%BAS%W2(MWIND)      [  km/s] Upper limits of windows
REAL*4         R%HEAD%BAS%SINUS(3)       [       ] Sinus baseline result
REAL*4         WINDOW(FOUND,1+2*RNWIND)  [  km/s] 2D plots: window for each r


               HISTORY Scan numbers of initial observations
PARAMETER      MSEQ=100
INTEGER*4      R%HEAD%HIS%NSEQ          [  ] Number of sequences
INTEGER*4      R%HEAD%HIS%START(MSEQ)   [  ] Start scan number of seq.
INTEGER*4      R%HEAD%HIS%END(MSEQ)     [  ] End   scan number of seq.


                 PLOT Default plotting limits.
REAL*4         R%HEAD%PLO%AMIN   [  ] Min Y value plotted
REAL*4         R%HEAD%PLO%AMAX   [  ] Max Y value plotted
REAL*4         R%HEAD%PLO%VMIN   [  ] Min X value plotted
REAL*4         R%HEAD%PLO%VMAX   [  ] Max X value plotted


          FSWITCH Frequency switching information (for spectra)
PARAMETER      MXPHAS=8
```

```
INTEGER*4     R%HEAD%FSW%NPHAS            [    ] Number of phases
INTEGER*4     R%HEAD%FSW%SWMOD            [code] Switching mode (frequency, p
REAL*8        R%HEAD%FSW%DECAL(MXPHAS)    [ MHz] Frequency offsets
REAL*4        R%HEAD%FSW%DUREE(MXPHAS)    [   s] Time per phase
REAL*4        R%HEAD%FSW%POIDS(MXPHAS)    [    ] Weight of each phase
REAL*4        R%HEAD%FSW%LDECAL(MXPHAS)   [ rad] Lambda offsets
REAL*4        R%HEAD%FSW%BDECAL(MXPHAS)   [ rad] Beta offsets of each phase
Codes for SWMOD can be found in SIC%CODE%SWITCH%.


                 CALIBRATION Calibration parameters
REAL*4        R%HEAD%CAL%BEEFF      [     ] Beam efficiency
REAL*4        R%HEAD%CAL%FOEFF      [     ] Forward efficiency
REAL*4        R%HEAD%CAL%GAINI      [     ] Image/Signal gain ratio
REAL*4        R%HEAD%CAL%H2OMM      [   mm] water vapor
REAL*4        R%HEAD%CAL%PAMB       [  hPa] Ambient pressure
REAL*4        R%HEAD%CAL%TAMB       [    K] Ambient temperature
REAL*4        R%HEAD%CAL%TATMS      [    K] Atmosphere temp. signal band
REAL*4        R%HEAD%CAL%TCHOP      [    K] Chopper temperature
REAL*4        R%HEAD%CAL%TCOLD      [    K] Cold load temperature
REAL*4        R%HEAD%CAL%TAUS       [neper] Opacity signal band
REAL*4        R%HEAD%CAL%TAUI       [neper] Opacity image band
REAL*4        R%HEAD%CAL%TATMI      [    K] Atmosphere temp. image band
REAL*4        R%HEAD%CAL%TREC       [    K] Receiver temperature
INTEGER*4     R%HEAD%CAL%CMODE      [ code] Calibration mode
REAL*4        R%HEAD%CAL%ATFAC      [    K] Applied calibration factor (Tcal)
REAL*4        R%HEAD%CAL%ALTI       [    m] Site elevation
REAL*4        R%HEAD%CAL%COUNT(3)   [count] Power of Atm., Chopp., Cold
REAL*4        R%HEAD%CAL%LCALOF     [  rad] Longitude offset for sky measurem
REAL*4        R%HEAD%CAL%BCALOF     [  rad] Latitude offset for sky measureme
REAL*8        R%HEAD%CAL%GEOLONG    [  rad] Geographic longitude of observato
REAL*8        R%HEAD%CAL%GEOLAT     [  rad] Geographic latitude of observator
Codes for CMODE can be found in CLASS%CODE%CALIB%.


          SKYDIP For Skydips observations. No associated data.
PARAMETER     MSKY=10
REAL*8        R%HEAD%SKY%SREST          [MHz] Rest frequency
REAL*8        R%HEAD%SKY%SIMAG          [MHz] Image frequency
INTEGER*4     R%HEAD%SKY%NSKY           [   ] Number of points on sky
INTEGER*4     R%HEAD%SKY%NCHOP          [   ]   -   -   -   -  chopper
INTEGER*4     R%HEAD%SKY%NCOLD          [   ]   -   -   -   -  cold load
REAL*4        R%HEAD%SKY%ELEV(MSKY)     [rad] Elevations
REAL*4        R%HEAD%SKY%EMISS(MSKY)    [  ?] Power on sky
REAL*4        R%HEAD%SKY%CHOPP(MSKY)    [  ?] Power on chopper
REAL*4        R%HEAD%SKY%COLD(MSKY)     [  ?] Power on cold load


            GAUSS Gauss fit results (for spectra or drifts)
PARAMETER     MXGAUS=5
```

```
INTEGER*4      R%HEAD%GAU%NLINE              [             ] Number of components
REAL*4         R%HEAD%GAU%SIGBA              [   Int. unit] Sigma on base
REAL*4         R%HEAD%GAU%SIGRA              [   Int. unit] Sigma on line
REAL*4         R%HEAD%GAU%NFIT(3*MXGAUS) [area,v0,fwhm] Fit results
REAL*4         R%HEAD%GAU%NERR(3*MXGAUS) [area,v0,fwhm] Errors


          SHELL "Stellar shell" profile fit results (for spectra)
PARAMETER      MSHELLFIT=20
INTEGER*4      R%HEAD%SHE%LSHEL              [   ] Number of components
REAL*4         R%HEAD%SHE%BSHEL              [   ] Sigma on base
REAL*4         R%HEAD%SHE%RSHEL              [   ] Sigma on line
REAL*4         R%HEAD%SHE%NSHEL(MSHELLFIT) [   ] Fit results
REAL*4         R%HEAD%SHE%ESHEL(MSHELLFIT) [   ] Errors


      NH3 "Ammoniac and Hyperfine Structure" profile fit results
                       (for spectra, ex: NH3, HCN)
PARAMETER      MHFSFIT=12
INTEGER*4      R%HEAD%NH3%LNH3              [   ] Number of components
REAL*4         R%HEAD%NH3%BNH3              [   ] Sigma on base
REAL*4         R%HEAD%NH3%RNH3              [   ] Sigma on line
REAL*4         R%HEAD%NH3%NNH3(MHFSFIT)    [   ] Fit results
REAL*4         R%HEAD%NH3%ENH3(MHFSFIT)    [   ] Errors


      ABS "Hyperfine Structure" ABSORPTION profile fit results
                       (for spectra, ex: NH3, HCN)
PARAMETER      MABSFIT=16
INTEGER*4      R%HEAD%ABS%LABS              [   ] Number of components
REAL*4         R%HEAD%ABS%BABS              [   ] Sigma on base
REAL*4         R%HEAD%ABS%RABS              [   ] Sigma on line
REAL*4         R%HEAD%ABS%NABS(MABSFIT)    [   ] Fit results
REAL*4         R%HEAD%ABS%EABS(MABSFIT)    [   ] Errors


          DRIFT Continuum drift description (for drifts)
REAL*8         R%HEAD%DRI%FREQ   [ MHz] Rest frequency
REAL*4         R%HEAD%DRI%WIDTH  [ MHz] Bandwidth
INTEGER*4      R%HEAD%DRI%NPOIN  [    ] Number of data points
REAL*4         R%HEAD%DRI%RPOIN  [    ] Reference point
REAL*4         R%HEAD%DRI%TREF   [   ?] Time at reference
REAL*4         R%HEAD%DRI%AREF   [rad?] Angular offset at ref.
REAL*4         R%HEAD%DRI%APOS   [rad?] Position angle of drift
REAL*4         R%HEAD%DRI%TRES   [   ?] Time resolution
REAL*4         R%HEAD%DRI%ARES   [rad?] Angular resolution
REAL*4         R%HEAD%DRI%CBAD   [    ] Blanking value
INTEGER*4      R%HEAD%DRI%CTYPE  [code] Type of offsets
REAL*8         R%HEAD%DRI%CIMAG  [ MHz] Image frequency
REAL*4         R%HEAD%DRI%COLLA  [    ] Collimation error Az
REAL*4         R%HEAD%DRI%COLLE  [    ] Collimation error El
```

Codes for CTYPE can be found in SIC%CODE%COORD%.

```
        BEAM Beam-switching parameters (for spectra or drifts)
REAL*4        R%HEAD%BEA%CAZIM  [ rad] Azimuth of observation
REAL*4        R%HEAD%BEA%CELEV  [ rad] Elevation of observation
REAL*4        R%HEAD%BEA%SPACE  [ rad] Beam spacing
REAL*4        R%HEAD%BEA%BPOS   [ rad] Position angle of beams
INTEGER*4     R%HEAD%BEA%BTYPE  [code] System for angle
Codes for BTYPE can be found in SIC%CODE%COORD%.

    CONTINUUM Double gaussian and baseline fit results (for drifts)
PARAMETER     MPOIFIT=8
INTEGER*4     R%HEAD%POI%LCONT           [   ] Number of components
REAL*4        R%HEAD%POI%BCONT           [   ] Sigma on base
REAL*4        R%HEAD%POI%RCONT           [   ] Sigma on line
REAL*4        R%HEAD%POI%NCONT(MPOIFIT)  [   ] Results
REAL*4        R%HEAD%POI%ECONT(MPOIFIT)  [   ] Errors
```

SET VELOCITY

```
    LAS\SET VELOCITY Mode
```

Specify the velocity kind to be used by the current R observation and
next ones loaded in memory. Mode can be
  - L[sr]   Forces the velocity axis values to Lsr values
  - H[elio] Forces the velocity axis values to Heliocentric values
  - A[uto] The velocity type is taken from the header of the spectrum
Default mode is Auto.

SET VIRTUAL

```
    LAS\SET VIRTUAL ON|OFF
    LAS\SET VIRTUAL CHECK|NOCHECK
```

First command enables or disables the buffering of spectra read in
CLASS. Default is ON.

A bug in this buffer handling was present from May2006 to Nov2008. See
the news for Dec08 release at http://www.iram.fr/IRAMFR/GILDAS/ for more
details. Since occurence was small and is difficult to predict, the sec-
ond command enables or disables an error which is raised when the bug
would have been encoutered. This is intended to help users to decide if
they should rerun or not the processing of their data with the latest
CLASS releases where the bug is fixed. Default is NOCHECK.

SET WEIGHT

```
    LAS\SET WEIGHT Type
```

Define the weighting to be used for summations. Choices are:
  T[ime], for Time*abs(Fres)/(Tsys**2) (spectrum) or
              Time*abs(Width)/(Tsys**2) (continuum),
  S[igma], for 1/Sigma**2,
  E[qual], for equal weights.

The default is TIME. Sigma is not recommended unless you just made a
baseline fit before. Equal weight behaves differently in AVERAGE/STITCH
and ACCUMULATE (which produces the sum).

SET WINDOW

```
      LAS\SET WINDOW [wl1 wu1 [wl2 wu2 [...]]] [/VAR array]
                [/POLYGON [N] [filename1...filenameN]]
                [/NOCURSOR]
```

Defines one or several line windows to be avoided by BASE when fitting a
baseline. If available, the cursor will be used to specify missing argu-
ments, unless option /NOCURSOR is present. When using the cursor, enter:
N (or Left mouse button) to enter a value, C (or Middle button) to can-
cel the last entry, H for help, E (or Right button) for exit.   The
boundaries are in current unit. There is no default.

SET WINDOW /POLYGON [N] [filename1...filenameN]
  Define the spectral window(s) through polygon(s). N is the number of
  polygons. If filenames are present, polygons are red from these files.
  If not, they are defined interactively with the cursor, and saves in
  files 'i'.pol, where 'i' is the number of the polygon.

SET WINDOW /VAR win_array
  Define the window from rank 1 array win_array[1+2*NWIND]. The array
  must be of the following form:
    win_array[1]=NWIND   number of windows
    win_array[2*i:2*i+1]   boundaries of window num. 'i'

5.1.29   SHOW

```
      LAS\SHOW Arg
```

Display some parameter defined by SET. Arg can be "ALL" to display all
parameters specified by SET. See SET command. SHOW command is not stored
in the stack of commands.

5.1.30   SPECTRUM

```
      LAS\SPECTRUM [Yoff] [/INDEX] [/OBS] [/PEN Ipen]
```

/OBS
    Plots the spectrum in the R buffer in the units and scales defined

by SET UNIT and SET MODE. Yoff (default 0) is a constant Y-offset added to the plotted values (useful to compare two spectra). The option /PEN allows to use a custom pen.

/INDEX
    Plots the 2D array from the last LOAD command.

Without option, the behavior depends on SET ACTION (see PLOT). Default is OBS.

5.1.31 STITCH

LAS\STITCH [/RESAMPLE NX Xref Xval Xinc Unit] [/NOCHECK [SOURCE|PO-SITION|LINE|SPECTROSCOPY|CALIBRATION]] [/IMAGE]

Stitch (i.e. make the union of) all the spectra of the current index using the current weighting function (see SET WEIGHT). The command behaves like AVERAGE /RESAMPLE except it ignores the SET ALIGN status and composes the spectra on the signal frequency axis. Use the option /IMAGE to compose them on the image frequency axis.

Bad channels are handled according to SET BAD. A sum may be interrupted by <^C>, but the result is then undefined.

The command checks the consistency of the index, in particular the positional coincidence and calibration homogeneity according to SET [NO]MATCH and SET CALIBRATION respectively. The option /NOCHECK allows to disable some (using one or several names) or all (no name given) of these checks.

STITCH /RESAMPLE

LAS\STITCH [/RESAMPLE NX Xref Xval Xinc Unit]

Without this option, the X axis of the output sum will be automatically computed as the composition of all the input frequency axes. The output spectrum has the coarsest resolution of the input spectra. Those are then internally resampled and added on this new axis.

A custom output X axis can be given with the option /RESAMPLE NX Xref Xval Xinc Unit. The input spectra are internally resampled and added on this axis. See HELP RESAMPLE for possible uses and caveats. The output resolution can not be better than the coarsest resolution of the input spectra to avoid oversampling.

5.1.32 SWAP

LAS\SWAP

Exchange the contents of the R and T buffers.

5.1.33   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 ver-
sions 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 op-
tion.

5.1.34   TITLE

```
LAS\TITLE [/BRIEF] [/LONG] [/FULL] [/INDEX] [/OBS]
```

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

/INDEX
    Writes the content of the loaded index in a compact form: Range of
    scan numbers, of spectrum, of observing dates and of offsets. It may
    be combined the other options (/LONG and /FULL give the same re-
    sult).

Without option, the behaviour is fixed by SET ACTION level. Default is
OBS.

5.1.35   UPDATE

```
LAS\UPDATE
```

Update the LAST version of the observation in the R buffer in the output
file. You can update only the last version. The update OVERWRITES the

observation. This command MUST BE USED WITH CAUTION; in case of complex
manipulations of observations, you may get confused and erase precious
data.

If more space is needed (e.g., the new version has more header informa-
tion), an error is returned. You MUST then WRITE the observation to
avoid loss of information.

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

5.1.36   WRITE

    LAS\WRITE [ObsNum]

Writes the observation in the R buffer onto the output file. If a ver-
sion of this observation is already in the file, a new version of the
observation is created. A new observation number can be specified as ar-
gument. If the implicit (R%HEAD%GEN%NUM) or explicit (argument ObsNum)
value is 0, automatic numbering is used i.e. a new (unused yet) number
is chosen (in pratice highest number in use plus one).

The behaviour depends on the "File Type" (see FILE command):
- for files of type MULTIPLE, a new version of the spectrum is written
  if the ObsNum already exists in the file.
- for files of type SINGLE, the ObservationNumber is automatically set
  to the next one available in the file. If an ObsNum is specified in
  the command, WRITE checks if it already exists. If so, it returns an
  error; if not, it sets the specified ObsNum to the written spectrum.
- for both, the version value (R%HEAD%GEN%VER) is always incremented,
  assuming something has changed in R between the time it has been load-
  ed in memory (or last written) and the current write.
On return of the command, R%HEAD%GEN%NUM and R%HEAD%GEN%VER are set to
the values which have been used.

The information written depends on the operations done on the observa-
tion. Existing fit results are written. The calibration information is
written only if the calibration is checked (see SET CALIBRATION com-
mand).

In the case of (old) on-the-fly data, WRITE automatically splits the da-
ta into records with entry numbers starting at 1 unless the observation
number of the first record is given explicitly.

5.2   ANALYSE Language Internal Help

5.2.1   Language

```
          ANALYSE\ Command Language Summary
  (+ indicates a NEW command and - an obsolete command)

  COMMENT            Manipulate the COMMENT section of spectra
 -CUBE file [fnct]   Obsolete. Replaced by TABLE and MAP\XY_MAP
  DIVIDE             Divide R spectrum by T spectrum
  DRAW               Call the cursor or plot comments
  FILL begin end     Fill from begin to end with noise, blank, etc.
  FFT                Compute and edit Fourier Transform of R or P
  GREG [name]        Make a TABLE or formatted file from current scan
 -GRID               Obsolete. Replaced by TABLE and MAP\XY_MAP
 +LMV                Convert a data cube into a set of spectra
  MAP [M|W]          Plot a map of spectra (or only their location)
  MEMORIZE Arg       Memorize the current observation
  MODEL Var          Generate a CLASS spectrum from a 1D variable
  NOISE [S [NEW]]    Generate gaussian noise
  PRINT [Arg]        Print values in a formatted file, or a table
  POPUP              Zoom a spectrum from a STAMP or MAP plot
  REDUCE             Reduce a SKYDIP
  RESAMPLE Arg       Resample a spectrum on a specified grid
  RETRIEVE Arg       Retrieve an observation from the memories
  SMOOTH [Arg...]    Smooth the spectrum in R
  STAMP              Display all spectra in index on one plot
  STRIP File         Create an image for Position-Velocity plots
 +TABLE              Build a GILDAS table from the current index
```

5.2.2   COMMENT

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

This command READs, WRITEs or DELETEs an existing comment section (max. length: 1020 characters) of the current R buffer.
```
  READ      display the comment section
  WRITE     write (or append) the input comment given between
            double quotes.
  EDIT      write a comment given interactively
  DELETE    delete and reset the comment section
  APPEND    Obsolescent
```

5.2.3   CUBE

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

Obsolete. Replaced by ANALYSE\TABLE and MAP\XY_MAP

5.2.4 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.

5.2.5 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 :
```
     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
```
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.

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.

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 position.

DRAW LOWER

```
ANALYSE\DRAW LOWER value "text" [angle]
```

Draws a vertical line with label "text" (written vertically by default, or with orientation 'angle' in degree) at abscissa 'value' in lower axis

unit. The line length is adjusted to approach closely the spectrum value
at the corresponding abscissa.

DRAW MASK

        ANALYSE\DRAW MASK [level]

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

DRAW MOLECULE

        ANALYSE\DRAW MOLECULE frequency name [angle]

Display a molecule name at the given frequency (irrespective of the
units from the SET UNIT command), with a given orientation (angle, de-
fault 90). The name lies on the lower X-axis.

        frequency   Frequency in MHz of the line
        name   String giving the name to be written
        angle   Orientation, in degree, with respect to X-axis

DRAW UPPER

        ANALYSE\DRAW UPPER Value "text" [angle]

Draws a vertical line with label "text" (written vertically by default,
or with orientation 'angle' in degree) at abscissa Value in upper axis
unit. The line length is adjusted to approach closely the spectrum value
at the corresponding abscissa.

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.

DRAW WINDOW

        ANALYSE\DRAW WINDOW [Yup] [Ylow]]

Plots the current line windows, with bars. Ylow and Yup refer to the
bottom and upper levels of the bars.

5.2.6   FFT

        ANALYSE\FFT [/REMOVE [wmin wmax]]
                    [[s1 e1] [s2 e2] ... [s10 e10] /KILL]
                    [/INDEX] [/OBS] [/NOCURS]

FFT computes a Fourier Transform and plots it. It acts either on a sin-
gle spectrum (from the R buffer) or on the INDEX. The default behavior
is defined with SET ACTION, but can always be ignored with /OBS or /IN-
DEX. Identically, the option /NOCURS overrides the default cursor. /KILL
allows to linearly interpolate the fourier transform, either interac-
tively or by supplying values. The /REMOVE option indicates that the
fourier transform is to be computed from a spectrum linearly interpolat-
ed in some intervals (specified or taken from the last fitted profiles).
R and T observations are unaffected.

FFT /KILL

        FFT [[s1 e1] [s2 e2] ... [s10 e10] /KILL]

Perform a linear interpolation (on real and imaginary parts) on the
fourier transform. The intervals are defined interactively (if no value
supplied and the cursor is available) or not (if values in MHz-1 are
supplied).

FFT /REMOVE

        ANALYSE\FFT /REMOVE [wmin wmax]
If no value supplied, the sum of existing fits of the current method is
subtracted before computing the Fourier transform. If an interval is
supplied, linearly interpolate between wmin and wmax.

5.2.7   FILL

        ANALYSE\FILL s1 e1 [[s2 e2] ...] [/INTER] [/NOISE [rms]] [/BLANK
[bval]]

Performs the specified action in the interval(s) specified with starting
and ending values (in current units).

/INTER
    Makes a linear interpolation. When combined with /NOISE [rms], a
    Gaussian noise is added to the linear interpoltion.
/NOISE [rms]
    Fill in the interval(s) with a Gaussian noise of a given rms. The
    rms defaults to the one computed in the BASE section. If this sec-
    tion is absent, it is computed from Tsys, resolution and integration
    time in the GENERAL section. If one of these values is null (e.g.
    spectrum was generated by the command MODEL), an error is raised.
    This option can be combined with /INTER.
/BLANK [bval]
    Fill in with blank value unless bval is specified.

5.2.8 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 com-
bination 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 residu-
als 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.

5.2.9 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 temp-
file]]
```

```
[/PLOT [where]]
```

Obsolete. Replaced by ANALYSE\TABLE and MAP\XY_MAP

5.2.10   LMV

```
ANALYSE\LMV Filename [/SCAN LAMDBA|BETA] [/STEP DLAMBDA [DBETA]]
```

Read the spectra cube stored in input Filename and convert it in a set
of spectra automatically written in the currently opened output file.
The spectra cube can be ordered in LMV, VLM, LVM, etc... FITS files are
automatically converted.

By default, the command sorts the spectra to mimic the scanning path of
a telescope along the lambda direction, shifted on beta for each scan.
Each constant beta row has a unique SCAN number. The sorting can be ro-
tated by 90 degree by using the /SCAN BETA option.

Fully blanked spectra in the input cube are ignored. If the cube spatial
axes are largely oversampled, the /STEP option allows the user to select
1 spectrum every DLAMBDA (resp. DBETA) in the LAMBDA direction (resp.
BETA). If DBETA is omitted, it defaults to DLAMBDA. The default value of
DLAMBDA and DBETA (if the /STEP option is not invoked) is 1, implying a
conversion of the whole cube.

User should take care that many parameters of the CLASS spectrum headers
have no meaning here. They are left to their default values.

5.2.11   MAP

```
ANALYSE\MAP [Match|Where|Keep] [/CELL Size_X [Size_Y]] [/GRID] [/NO-
LABEL] [/NUMBER]
```

Makes a map of spectra or display the true locations of the spectra.
Offset limits are automatically computed, and it adjusts the size of a
spectrum depending in the current box size. If not in WHERE-mode, 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.

MATCH
    To force matching the X to Y ratio, which is otherwise adjusted so
    as to make the largest possible drawing. The options control pre-
    sentation details.

WHERE
    Only displays the spectra locations with crosses (the marker can not
    be modified). Can be combined with MATCH. Global variables idx%loff
    and idx%boff contains offsets positions of all entries of the cur-
```

rent index, and may also be used, with GREG1\ commands (g\limit,
g\box, g\point), to display the spectra locations. In this case, the
marker can be modified (g\set marker).

KEEP
    Will not recompute the grid. Uses the last grid. Can be useful for
    online display.

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.

MAP /GRID

        ANALYSE\MAP/GRID

A grid is plotted to separate the spectra.

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.

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.

MAP /NUMBER

        ANALYSE\MAP/NUMBER

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

5.2.12   MEMORIZE

        ANALYSE\MEMORIZE [Name]
        ANALYSE\MEMORIZE Name|* /DELETE

If a Name is specified, put a copy of the current observation in a "mem-
ory" called Name (limited to 12 characters). The observation can later

be retrieved using command RETRIEVE. If no Name is specified, list the currently defined memories. Up to 100 memories can be defined at a time.

MEMORIZE Name /DELETE can be used to delete one "memory" area. Use a wildcard (*) to delete all memories.

5.2.13 MODEL

ANALYSE\MODEL Y_Variable [X_Variable] [/BLANK Bval] [/REGULAR [Xref Xval RestFreq]] [/FREQUENCY LineName RestFreq] [/XAXIS Xref Xval Xinc Unit]

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 X axis scale for a presumably non linear or irregularly sampled 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, you should avoid this solution when possible.

There are three (exclusive) ways to define the X axis: provide a X array, /XAXIS option, /REGULAR option (see below). The model header will have a minimal values set. On the other hand, if none of these three is used, the command will copy RX and header from the previous R buffer in memory.

/BLANK Bval
    Allow the user to provide the blanking value. Default is -1000.

/REGULAR [Xref Xval RestFreq]
    Indicate that data are regularly spaced along the X axis. If no reference pixel and value at reference are supplied, default values are set to respectively half the channel number and to the velocity in the nearest channel.

/FREQUENCY LineName RestFreq
    Indicate line name and the rest frequency. Default rest frequency is 3.10^5 MHz.

/XAXIS Xref Xval Xinc Unit
    Define the X axis with its reference channel (Xref), its value at reference channel (Xval), and the increment per channel (Xinc). Units are km/s or MHz if Unit is VELOCITY or FREQUENCY respectively. If Unit is FREQUENCY, Xval is expected to be the rest frequency.

5.2.14   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 computed in the BASE section of the current spec-
trum. If this section is absent, it is computed from Tsys, resolution
and integration time in the GENERAL section. If one of these values is
null (e.g. spectrum was generated by the command MODEL), an error is
raised.

If the 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.

5.2.15   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 dis-
played 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 observa-
tions 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 eob-
servation 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.

5.2.16   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
```

The information written is (1) the number of lines fitted, (2) the
observation number, (3,4) the offsets in current angle units,
(5,6,...) fit results of the current method in the same order as in
DISPLAY command. For the Gauss Method, this order is: (5) Line Area,
(6) its incertainty (7) Position (8) incert. (9) Width (10) incert.
(11) Intensity (12) Base RMS (13) 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
```

Area in velocity ranges v1 to v2 and v2 to v3, ... etc. The infor-
mation written is (1) the observation number, (2,3) the offsets in
current angle units, (4,5,...) the areas.

```
CHANNEL List            List of channels
```

The list 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 observa-
tion number, (2,3) the offsets in current angle units, (4,5,...) the
channels value.

```
MOMENT m1 m2 ...        Moments of the observations
```

The moments (area, mean velocity, width) are computed in the inter-
vals [m1,m2] [m3,m4] etc... Current unit 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                Pointing fit results
```

Results of CONTINUUM method fit are 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 num-
ber and station code, (11,12) Width and error, (13,14) Intensity and
rms noise, (15) Source name.

```
FLUX                    Flux measurements results
```

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 Eleva-

tion 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) Sig-
nal 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 ac-
  cessed 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 op-
  erations can be done directly on the table columns. However the FIT
  results cannot be written to a table.

5.2.17 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

5.2.18 RESAMPLE

        ANALYSE\RESAMPLE Nx Xref Xval Xinc Unit [Shape] [Width] [/[NO]FFT]

This command resamples a given spectrum on a specified X axis in given
unit. The arguments are:
  Nx   the number of channels desired,
  Xref the new reference channel,
  Xval the value at the reference channel. Changing the reference fre-
        quency or velocity has non trivial effects on the axis. Use ’=’
        when possible (see below), else change the value with caution,
  Xinc the new channel separation. In case of FREQUENCY resampling, the
        resolution is expected to be given in the restf frame. However,
        RESAMPLE does not modify the current frame of the spectrum (i.e.
        the Doppler factor is left unchanged), so the input frequency
        resolution will be corrected to the current frame.
  Unit either VELOCITY or FREQUENCY to indicate the units used for Xval
        and Xinc.
  Shape (optional) a keyword for the frequency response of the synthe-
        sized channels (see subtopic for details). 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.
```

```
Using  an equal sign '=' for any of these arguments indicates to use the
same value as before. Using a wildcard '*' will choose an automatic  de-
fault:
  Nx    is set such as the last input and output channels are aligned,
  Xref  is set such as the first input and output channels are aligned,
  Xval  the reference frequency and velocity are left unchanged (same as
        =)
  Xinc  same as =
  Shape same as =
  Width same as =
```

```
/NOFFT is the default behavior. In this case, the output spectrum  is  a
subset  of  the original one if their axes are aligned to a given preci-
sion. If not, the spectrum is resampled by direct interpolation. In this
mode  the  exact channel separation requested will be used. The counter-
part to this technique is that correlation is introduced  between  adja-
cent  channels  and thus affects the noise intensity (e.g. it is divided
by sqrt(2) if the axis is shifted by half a channel).
```

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

RESAMPLE /FFT

```
      RESAMPLE /FFT
```

```
The  resampling  is  done  in  the delay domain and the data are Fourier
transformed.  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).
```

RESAMPLE SHAPE

```
Four keywords are used to describe the frequency response of the synthe-
sized channels:
- TBOX  means  a  rectangular  function  in the delay domain, as for un-
        smoothed 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 fil-
        terbank channels)
  - FTRIANGLE means a triangular function in the frequency domain (as for
        filterbank channels, Hanning smoothed)

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.

5.2.19   RETRIEVE

        ANALYSE\RETRIEVE Name

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

5.2.20   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 (default)
    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 to-
    tal 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 ten-
    dency to produce wings...

5.2.21 STAMP

```
ANALYSE\STAMP NX [NY] [/LABEL SCAN|NUMBER|SOURCE|LINE [...] ]
```

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 command does not require that X and Y scales be fixed.

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

STAMP /LABEL

```
ANALYSE\STAMP NX [NY] /LABEL SCAN|NUMBER|SOURCE|LINE [...]
```

The option /LABEL writes the required additional information (scan number, observation number, source or line name) in some part of the box to help identification. Several additional informations can be displayed. number of each observation in the upper left part of its box.

5.2.22 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.

To produce a strip along another direction than X or Y, change the offsets of your spectra using command MODIFY OFFSETS, and write 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.

PV cuts might also be done easily in GREG from a GILDAS table or cube with more versatility.

5.2.23 TABLE

```
     ANALYSE\TABLE Filename [OLD|NEW] [/MATH Expression1 ... ExpressionN]
[/RANGE rmin rmax unit] [/RESAMPLE Nc Ref Val Inc Unit [Shape]  [Width]]
[/FFT] [/FREQUENCY name rest-freq] [/NOCHECK [SOURCE|POSITION|LINE|SPEC-
TROSCOPY]]
```

Build a GILDAS table from current index entries. No gridding is done.
This table may be then regridded on a Regular Grid to build a datacube
for further use in GREG.

```
  OLD     Append to table 'filename'
  NEW     Create a new table

  /MATH
    Compute one or more mathematical expressions instead  of  storing  a
    spectrum.  See subtopic FUNCTIONS for a list of available functions.

  /RANGE
    By default, the whole X axis is stored in the GILDAS table. A  range
    can be imposed, in which case the unit must be specified.

  /RESAMPLE
    By default, the spectra are resampled, on-the-fly while building the
    TABLE, on the first spectrum X axis. This option allows  to  enforce
    that  resampling  is done on the given X axis. See HELP RESAMPLE for
    possible uses and caveats.

  /FFT
    To specify that resampling is done in time domain.

  /FREQUENCY
    To change the line name and/or the rest frequency.

  /NOCHECK
    By default, the consistency of the index is checked against  SOURCE,
    POSITION, LINE and SPECTROSCOPY parameters (see LAS\CONSISTENCY) be-
    fore building the table. If the SPECTROSCOPY section appears  incon-
    sistent,  the spectra are resampled on-the-fly on the first spectrum
    X axis. Other sources of inconsistency are not recoverable  and  the
    table  is  not  built. This option without keyword allows to disable
    all consistency checks. With keyword(s), disable  the  corresponding
    checks.

Warnings:
    1.  /MATH, /RANGE and /RESAMPLE are exclusive
    2.  Continuum data not supported
```

TABLE FUNCTIONS

You can use any Sic mathematical formula (e.g. involving some R buffer variables). The following functions are also predefined:

```
STDV(dummy)    returns the integrated area (K.Km/s) of the current  spec-
               trum
TPEAK(dummy)   returns the Peak temperature (K) of the current spectrum
VPEAK(dummy)   returns  the  Velocity at Peak (Km/s) of the current spec-
               trum
TDV(v1,v2)     returns the integrated area (K.Km/s) of the current  spec-
               trum 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  spec-
               trum between velocities V1 and V2
```

To create a table containing the integrated area of your spectra, type:
```
    TABLE Filename /MATH STDV(0)
```

5.3   FIT Language Internal Help

5.3.1   Language

```
           FIT\ Language Summary
```

```
DISPLAY               Prints the results of the profile fit of R.
VISUALIZE [N]         Plots the current fitted profile.
MINIMIZE              Performs a profile fit.
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.
RESIDUAL [n]          Computes the residuals of the profile fit.
```

5.3.2   DISPLAY

```
    FIT\DISPLAY [Yoff] [/NOPLOT] [/METHOD]
```

Displays the results of the profile fit of R for the current method, in the terminal and in the plot header. Yoff indicates the Y offset for the results to be printed on the plot; default value is 0.

For the Gauss (default) method (see HELP METHOD), the command displays (1) the Line number, (2) the Area, (3) the Position, (4) the Width and (5) the Intensity , each with their associated errors, and resulting RMS in the Base and Line windows.

Note that the values are also available in the R% variable, for use in
procedures. See HELP MINIMIZE for details.

    /NPLOT: if present, print the fit result but do not display them on
the plot.
    /METHOD: if present, print the fit result for the corresponding
method.

5.3.3   ITERATE

    FIT\ITERATE [N]

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

5.3.4   KEEP

    FIT\KEEP

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

5.3.5   LINES

    FIT\LINES [N [Guesses]] [/NOCURSOR] [/INPUT File_Name] [/SHOW]

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,
MINIMIZE will attempt to guess initial values for a single line by com-
puting the moments of the spectrum. If N is not 0, the user has to pro-
vide the initial values for line parameters. N is limited to 5. These
values should be separated by spaces and can be entered as SIC expres-
sions as follows:

Method	Code,Value	Code,Value	Code,Value	Code,Value
GAUSS	Temperature	Position	Width	
SHELL	Area (K.MHz)	Offset (MHz)	Width (MHz)	Horn/Center

```
NH3      Temperature    Position     Width (MHz)    Opacity


HFS      Temperature    Position     Width (MHz)    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).

There are 4 possibilities for the user to provide the line parameters:
  - If /INPUT File_Name is used, guesses are read from the input file.
    "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.
  - The guesses can also be entered from the command line, as double-
    quoted strings, one per line, e.g.
        FIT\LINES 2 "0 0.20 0 0.00 0 25.00" "0 0.10 0 0.00 0 50.00"
  - If the cursor is available, you will have to set interactively on
    the plot 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.
  - If none of the 3 methods above is used, then the user is prompted in
    the terminal for the guesses. The parameters must be entered togeth-
    er, separated by blanks, one set of guesses per line.
The option /SHOW can be used for a feedback of what initial guesses have
just been defined.

Command LINES is not supported for method CONTINUUM.

5.3.6 METHOD

        FIT\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 spec-
    trum, according to command lines. Variables are Area Velocity and
    Width (FWHM).

- SHELL   Horn-type profiles for Circumstellar Shells.
  Variables are 1) Area [K.MHz] 2) Center offset [in MHz] 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 per-
  fectly double peaked lines). Profiles are symmetric. Up to 5 depen-
  dent or independent lines may be fitted. X axis must be frequency.
- NH3(1,1) or NH3(2,2) or NH3(3,3)
  Compute ammonia line profiles, with the assumptions of gaussian ve-
  locity distribution and equal excitation temperatures. Variables are
  1) Main group opacity times (Excitation temperature minus Back-
  ground   temperature),   2)   Velocity,   3)   Width   (FWHM   =
  sqrt(8.0*alog(2.0)) * line 2nd order moment), 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 de-
  scribe the hyperfine structure are read in the file FileName. The
  first line of this file must contain the number of hyperfine compo-
  nents (< 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 infor-
  mation for two-component fitting if required. The command METHOD
  CONTINUUM accept additional arguments, which can be used to cus-
  tomize 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 in-
dicates 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.

5.3.7   MINIMIZE

```
      FIT\MINIMIZE
```

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 data).

MINIMIZE performs the fit only on the part of the spectrum selected by
SET MODE X. It also ignores the regions under the masks defined by SET
MASK.

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. If MINIMIZE 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 observa-
tion was made using beam-switching method and the reference beam appears
in the drifts, a two-component gauss fit is done using separation speci-
fied in the data.

The fitting results are saved in the R% variable, for e.g. reuse in pro-
cedures:
```
  Method       Structure
  GAUSS        R%HEAD%GAU%
  SHELL        R%HEAD%SHE%
  NH3/HFS      R%HEAD%HFS%
  ABSORPTION   R%HEAD%ABS%
  CONTINUUM    R%HEAD%POI%
```
See HELP SET VARIABLE for the detailed meaning of each component. For
example, the gaussian fit parameters can be found in:
```
  Parameter       Variable
  Area            R%HEAD%GAU%RESULT['(ILINE-1)*3+1']
  Error(Area)     R%HEAD%GAU%ERROR['(ILINE-1)*3+1']
  Position        R%HEAD%GAU%RESULT['(ILINE-1)*3+2']
  Error(Position) R%HEAD%GAU%ERROR['(ILINE-1)*3+2']
  Width           R%HEAD%GAU%RESULT['(ILINE-1)*3+3']
  Error(Width)    R%HEAD%GAU%ERROR['(ILINE-1)*3+3']
```
where ILINE is the number of one of the fitted lines, between 1 and
R%HEAD%GAU%NLINE.

5.3.8 RESIDUAL

```
FIT\RESIDUAL [n]
```

Compute the residuals of the last MINIMIZE 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.

5.3.9 VISUALIZE

```
FIT\VISUALIZE [N] [/PEN [i]]
```

Plot the current fitted profile. The Nth line obtained the last time
MINIMIZE 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 giv-
en, the last value of N is used.

   /PEN [ipen]: if ipen is present, use the given pencil 'ipen' prop-
erties (defined with "pen ipen"), otherwise use the last defined pencil.
Without option, use pencil 2 properties.

5.4   MAP Language Internal Help

5.4.1 Language

```
APODISE             Apodization
DECONVOLVE          Deconvolve
XY_MAP              Build an LMV spectra cube from an XY table
```

5.4.2 APODISE

```
MAP\APODISE
```

Not yet available. Work in progress.

5.4.3 DECONVOLVE

```
MAP\DECONVOLVE
```

Not yet available. Work in progress.

5.4.4 XY_MAP

```
MAP\XY_MAP [FilenamePrefix] [/NOGRID] [/TYPE LMV|VLM|...]
```

Build an LMV spectra cube and an associated weight image from an XY ta-
ble. The LMV cube (.lmv), WEI image (.wei) and XY table (.tab) share the
same file name, given without extension. By default, data is gridded with

a convolution (general case), unless /NOGRID option is invoked (typical-
ly when the data is already regularly placed).

By default, the spatial grid of the LMV cube will be large enough to use
all the spectra of the XY table. The pixel size will ensure Nyquist sam-
pling (i.e. at least 2 pixels per beam). The grid position reference and
position angle will be taken from the XY table header. The convolution
kernel is a Gaussian of size one-third the telescope beamwidth, the con-
volution being computed on a limited support (large enough to ensure ex-
cellent accuracy).

However, the griding process may be finely tuned through user defined
values of the MAP structure. The header of an existing LMV cube may be
used as a template for the grid definition. To do this, the user must
set the MAP%LIKE variable to the template filename. When MAP%LIKE is set
to an empty string, the user may define the grid by hand through the
following parameters

```
MAP%CELL      [arcsec   ]   X and Y pixel size
MAP%SIZE      [pixel    ]   X and Y map   size
MAP%ANGLE     [degree   ]   Grid position angle
MAP%RA        [  HH:MM:SS]   Grid reference position
MAP%DEC       [+DDD:MM:SS]   Grid reference position
MAP%SHIFT     [logical  ]   Shift and/or Rotate grid?
```

The natural beamwidth of the telescope is set in order of precedence by

```
TAB%MAJOR     [radian   ]   The resolution in the table header, if defined
MAP%TELE      [string   ]   The telescope name
MAP%DIAM      [m        ]   The telescope diameter (if MAP%DIAM = 0)
MAP%BEAM      [arcsec   ]   The beam size (if MAP%TELE = "" and MAP%DIAM = 0
```

Because of the convolution, the spatial resolution of the grided data
will by default be 5.4% wider than the telescope beamwidth at the obser-
vation frequency. It is however possible to grid the data with a wider
Gaussian in order to smooth the data (e.g. to increase the signal-to-
noise ratio of extended emission). The user have to define the required
resolution of the grided data through

```
MAP%RESO     [arcsec   ]   Final resolution of the grided data (default is
```

the command will deduce the size of the griding kernel from the tele-
scope beamwidth and the required resolution.

It is advised to avoid changing the following more technical parameters.
However those parameters are tunable through

```
MAP%COL       [integer  ]   Range of columns to grid
MAP%WCOL      [integer  ]   Column of weights
MAP%CTYPE     [integer  ]   Convolution kernel type
MAP%SUPPORT   [arcsec   ]   Convolution kernel support size
```

For ease of use, a sensible value of each griding parameter will auto-
matically be computed from the context if the value of the corresponding
MAP structure component is set to its default value, i.e. zero value and
empty string. The only exception is the position reference and/or posi-
tion angle of the grid: Changing them requires in addition to set
MAP%SHIFT to YES.

The output LMV cube can be then analysed with the standard GreG tools
like GO VIEW. However, it is also possible to reimport the cube in Class
as a set of spectra thanks to the command ANALYSE\LMV (see HELP for de-
tails).

XY_MAP /NOGRID

```
    MAP\XY_MAP [Filename] /NOGRID
```

XY_MAP /NOGRID builds an LMV spectra cube and an associated weight image
from an XY table, but does not regrid the data with a convolution. The
command assumes that it will encounter points which are regularly
spaced, and that it can deduce from these the typical parameters of the
spatial grid (pixel size, map size, ...).

The parameters of the output cube can still be overridden by user's
MAP%LIKE and MAP% objects.

XY_MAP /TYPE

```
    MAP\XY_MAP [Filename] /TYPE LMV|VLM|...
```

By default, XY_MAP produces an LMV cube. This can be overridden by the
option /TYPE to reorder the dimensions as desired. All permutations are
supported. In particular, VLM is the natural ordering of the internal
buffers (i.e. it should be faster to create the cube in this order), and
it should also speed up cube reading when it is accessed by frequency
slices.

5.5 DSB2SSB Language Internal Help

5.5.1 Language

```
        DSB2SSB\ Language Summary
```

```
INITIALIZE
DECONVOLVE
```

5.5.2 INITIALIZE

```
DSB2SSB\INITIALIZE [Continuum_Offset]
```

Builds the data array for deconvolution from all spectra in the index. A continuum offset can be provided.

This instruction MUST be called before your first call to DECONVOLVE.

5.5.3 DECONVOLVE

```
DSB2SSB\DECONVOLVE Tolerance  Itmax  [Lambda1  [Lambda2  [Lambda3]]]
[/GAIN [First_Guess]] [/KEEP]
```

Deconvolves DSB dataset loaded with INITIALIZE into SSB spectrum. Minimizes the function:

        Chi^2 - lambda1 F1 - lambda2 F2

until either the difference in Chi^2 between the iterations is less than the Tolerance, or the maximum number of iterations Itmax has been reached.

```
Here:
  Chi^2 = Sum (dsb - ssb_lsb*gain_lsb - ssb_usb*gain_usb)^2
          /(weight * nu)

  F1 : entropy of the channels.
  F1 = - Sum ssb/SSB * log ssb/SSB

  F2 : entropy of the gains.
  F2 = - Sum gain_ssb/e * log gain_ssb/e
and
  dsb           - double sideband data
  ssb           - single sideband data
  gain          - gain of single sideband data
  weight        - T_sys^2/(integrationime * bandwidth)
  nu            - number of free parameters:
  dsb_channels - ssb_channels - sideband_gains
  SSB           - Sum ssb
```

If Lambda1 or Lambda2 or Lambda3 are not specified, they are set to 0 and the algorithm does NOT try to maximize the entropy term.

By default the deconvolution makes no attempt at fitting the sideband gains, unless the option /GAIN is specified.

DECONVOLVE /GAIN

```
DSB2SSB\DECONVOLVE /GAIN [First_Guess]
```

When this option is present, the deconvolution will try to adjust the
sideband gains, fitting one constant value per sideband for every LO
setting.

--> make sure you have run the deconvolution without gain fitting first!
Here's a typical sequence of commands to turn on the gain fitting during
the deconvolution:

```
> initialize
> deconvolve 1e-5 100 /keep
> deconvolve 1e-5 100 /gain
```

First_Guess can be set to 1 or 0.

By default, First_Guess = 1 (the initial guess for signal and image
sideband gains is that they are equal to 1 for every LO setting).

Should any information on the sideband ratio be available in the header
variable GAINI, the user can enforce this as initial guess for the gain
fitting by setting First_Guess = 0.

DECONVOLVE /KEEP

    DSB2SSB\DECONVOLVE /KEEP

Keeps the outcome of the deconvolution in memory and uses it as initial
guess for the next (if any) run. The initial guess is reset to 0 whenev-
er the command INITIALIZE is used.

5.6   EXPERIMENTAL Language Internal Help

5.6.1   Language

         EXPERIMENTAL\ Language Summary

These commands are experimental and their name, calling sequence, and
behavior are subject to changes without advise.

```
FILTER          Apply a filter on the R spectrum
RMS             Compute the RMS of the index
SUBTRACT        Compute T-R
UNBLANK         Remove blanks in a Class TABLE
VARIABLE        Create sic variables mapping the observation(s)
WAVELET         Decompose the R spectrum into wavelets
```

5.6.2   FILTER

    EXPERIMENTAL\FILTER [/STORE]

Compute the Mean Absolute Deviation of the R spectrum, and store the result in R%HEAD%BAS%SIGFI if the option /STORE is present.

5.6.3 RMS

```
    EXPERIMENTAL\RMS  [/RESAMPLE  [NX  Xref  Xval  Xinc Unit]] [/NOCHECK
[SOURCE|POSITION|LINE|SPECTROSCOPY|CALIBRATION]]
```

Produce the RMS spectrum of the index.

5.6.4 SUBTRACT

```
    EXPERIMENTAL\SUBTRACT
```

Compute the difference T-R. Consistency of the input spectroscopic axes
is checked. Output header is NOT IMPLEMENTED: T header is reused.

5.6.5 UNBLANK

```
    EXPERIMENTAL\UNBLANK [/MODE REJECT|NOISE|INTERPOLATE]
```

5.6.6 VARIABLE

```
    EXPERIMENTAL\VARIABLE Section1 [Section2 ...] [/MODE READ|WRITE|OFF]
[/INDEX]
```

VARIABLE /INDEX will read (from file) and save (in memory) in Sic arrays
the named sections of all entries in the current index. The Sic arrays
will be available in the structure IDX%HEAD%, under each named section.
Any further call to FIND and DROP (i.e. commands which modify the cur-
rent index) will destroy the Sic arrays. For efficiency purpose, they
are not recreated implicitly, the command VARIABLE /INDEX must be called
again instead.

VARIABLE USER /INDEX will load the user section for all entries, if the
associated hooks to the command are declared in Class. The Sic arrays
will be available in the structure IDX%USER%<owner>%<title>%

The option /MODE indicates to create read-only (READ) or read-write
(WRITE) variables. The keyword OFF will delete the variables of the
named sections and free their associated memory. WRITE is not allowed on
/INDEX arrays.

5.6.7 WAVELET

```
    EXPERIMENTAL\WAVELET [/BASE [N]]
```

Compute the wavelets of the R spectrum, and store all the orders in the
Sic array named WAVELET. If the option /BASE N is present, the N-th or-
der is subtracted from the R spectrum (default N is 5).

Chapter 6

Error Messages and Recovery Procedures (Out-of-Date)

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.

6.1   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.

6.2   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).

6.3   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.

6.3.1 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.

6.3.2 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`.

6.3.3 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.

6.3.4   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.

6.3.5   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.

6.3.6 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`) 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.

6.3.7   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.

6.3.8   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 3rd 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.

6.3.9   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.

6.3.10   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.

6.3.11   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