ESO - EUROPEAN SOUTHERN OBSERVATORY

E U R O P E A N S O U T H E R N O B S E R V A T O R Y
Organisation Européenne pour des Recherches Astronomiques dans l'Hémisphère Austral
Europäische Organisation für astronomische Forschung in der südlichen Hemisphäre

VERY LARGE TELESCOPE

Gasgano User's Manual

VLT-PRO-ESO-19000-1932

Issue 2
11 April 2002
56 pages

Prepared:	P. Amico N. Kornweibel F. Primas M. Zamparelli		
	Name	Date	Signature
Approved:	M. Peron		
	Name	Date	Signature
Released:	P. Quinn		
	Name	Date	Signature

CHANGE RECORD

Issue	Date	Affected Paragraphs(s)	Reason/Initiation/Remarks
1.0	27/9/99	All	Preliminary Draft
1.4	26/2/01	All	Draft PSO Commissioning MAR2001
1.4	26/2/01	All	Revised Version
2	11 April 2002	All	Public release

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. WELCOME TO *GASGANO*

Gasgano is a Data File Organiser developed and maintained by the European Southern Observatory (ESO) to help its user community to manage and organize in a systematic way the astronomical data observed and produced by all VLT compliant telescopes, i.e. by telescopes which are being operated through Observation Blocks. The tool also supports FITS files which are not generated by those telescopes but with a limited number of functionalities.

1.1 Purpose & Scope

This document is the user's manual for the ***Gasgano*** ESO Data Flow System File Organiser, in short ***Gasgano***. It is intended for guiding ESO-external users through its basic usage.

This manual describes ***Gasgano*** Version 1.5.1, the first public release of the tool. The tool was distributed to internal ESO users for the first time at the end of 1999 and since then it has successfully assisted Paranal Science Operations in their daily operations with the large amount of data generated by the VLT.

The manual is organized as follows: Chapter 2 gives an Overview of the tool, describing its main and most general features and functionalities; Chapter 3 leads the user through a simple session of ***Gasgano*** with the aim of showing the basic usage of the tool but in some detail; Chapter 4 offers the user to look deeper into ***Gasgano***, thus discovering how the tool can be customized depending on the user's needs; Chapter 5 is similar to Chapter 3, but is now devoted to explore more complex features of the tool (mostly intended for advanced users); Chapter 6 provides a quick reference guide to keywords, rules, keyboard-shortcuts, etc. At the end, three Appendixes complete the ***Gasgano*** documentation: System Requirements (App. A), a print out of the .gasganorc file (App. B), and some information on the Verbose Levels (App. C).

As you will find out in Appendix A, this manual does not contain an installation guide. Installation instructions are maintained online on the ***Gasgano*** web pages (http://www.eso.org/gasgano) The latest version of this manual will also be available from this web site.

1.2 Intended Audience

This manual is intended for all astronomers making use of ESO VLT compliant telescopes, who after obtaining/receiving their set of data need to process them in an efficient way. Familiarity with the high-level operational concepts used to operate VLT compliant telescopes (e.g. Observation Block) and with the ESO data format (in particular the hierarchical keywords in FITS files) is desirable.

1.3 Reference documents and web pages

[1] GEN-SPE-19400-794/1.1/25-Nov-97 - *Data Interface Control Document*

[2] VLT-SPE-ESO-19000-0842/0.92/14-Aug-97 - *VLT Data Flow System, Architectural Design Document*

[3] VLT-PRO-ESO-19000-1930/1.0/27-Sep-99 – *VLT Data Flow System, **Gasgano** DFS File Organiser Design Document*

[4] For more information on science operations and the VLT Data Flow, go to http://www.eso.org/outreach/info-events/ut1fl/whitebook/wb100.html

1.4 Glossary

Where appropriate, these definitions were adopted from the Data Interface Control Document ([1]). Minor wording changes have been made since ***Gasgano*** is used at several ESO facilities, not just the VLT.

Acquisition Template (AT): an Observation Block component. An Acquisition Template is used to specify how a target is acquired by the telescope. It may also specify any preliminary instrument configuration steps (e.g. set rotator to specific angle). It can contain parameters for interactive as well as automatic acquisitions. This template may define a different instrument/detector configuration from the templates within the Observation Description. Each science Observation Block contains at most one Acquisition Template.

Calibration Observation Blocks: Observation Blocks used to acquire calibration data.

Data Flow System (DFS): a system that can handle the flow of data and information within certain ESO facilities, in particular, the VLT. It includes subsystems for proposal handling, observation handling, science and engineering data archiving, science data pipeline processing and handling and data quality control [2]. The DFS is being developed by the *ESO Data Management and Operations Division* (DMD).

Data Organizer (DO): software, part of the DFS, used in the on-line environment to support the automatic pipeline data reduction of incoming data.

Data Products: data files delivered to PI/Co-Is as result of the execution of ESO observing programs or setup information submitted to ESO in relation to observing programs. Data products include observation frames, observatory calibrations, meteorological and seeing measurements relevant to primary observations, etc.

Exposure: a synonym for the acquisition of a single data frame, typically resulting in a single FITS file.

Header Keywords: header keywords can be standard FITS (8 character long, e.g. TELESCOP, MJD-OBS), ESO standard (e.g. HIERARCH ESO OBS PROG ID) or ***Gasgano*** Meta –Keywords (see sections and). Within ***Gasgano*** HIERARCH ESO keywords are specified in "dot" notation without the HIERACH ESO part. E.g.: OBS.PROG.ID.

Master calibration product: a reduced frame used as a master for calibration of science and calibration raw frames.

Observing Run: an approved ESO Program consists of one or more Observing Runs, each of which specify an independent combination of telescope, instrument, and observing operations mode (i.e. Service Mode or Visitor Mode).

Observing Program ID: see Run ID.

Observation: a coordinated sequence of telescope, instrument, and detector actions that results in a scientific or technical dataset.

Observation Blocks (OB): the smallest observational unit within the Data Flow System. An Observation Block contains a sequence of high level operations, called templates, that need to be performed sequentially and without interruption in order to ensure the scientific usefulness of an observation. Observation Blocks may contain only one target acquisition. Observation Blocks may contain scheduling and collect all status information of operations as they pass through the Data Flow System from definition, execution and reduction. Observation Blocks are used in both Visitor and Service Mode to acquire data.

Observation Block ID: the unique identifier of an Observation Block. It is stored in the keyword HIERARCH ESO OBS ID. Valid OB ID examples: 19368, 200083938

Raw Frame: Result of execution of an exposure by the VCS. The execution of an Observation Block may generate one or more Raw Frames. Raw frames are stored as FITS files. Their headers contain all the information relevant for pipeline data reduction QC and archiving.

Reduction Block (RB): A Reduction Block contains all the information needed to reduce one or more raw frames: it contains the name of the pipeline recipe to be applied, the list of input frames, the list of associated master calibration files and the name of the output pipeline products to be generated.

Reduction Block Scheduler (RBS): software, part of the DFS, which operates on Reduction Blocks, obtained from the Data Organizer: it analyzes the RB, checks the presence of input files and fires the corresponding data reduction recipe.

Reduction recipe: standard procedure for reducing observational data in a standard way. Recipes are implemented for each of the instrument standard templates. Those scripts take as input raw frames and execute them in a particular Data Reduction System (DRS).

Pipeline Product: Result of the execution of a Reduction Block. Pipeline products are stored on disk as FITS files (image or table) or ascii files (e.g. PAF file).,

Run ID: Same as Observing Program ID. the ID of an Observing Run. The format is kPP.X-0000(X) where kpp.c-nnnn where k is blank for normal, 1 for key programmes, pp is the period, c the category defined for proposals approved by the ESO OPC and nnnn a sequential number. Valid Run ID examples: 63.A-0352(A), 167.A-4563(B). The Run ID is stored in the HIERARCH ESO OBS PROG ID keyword.

Service Mode (SM): observing operations mode where the astronomer submits a detailed description of their observing program to ESO for later possible execution. Service Mode programs are executed primarily in order of their ESO Observing Programs Committee assigned priority but only when the astronomer specified observing conditions are achieved onsite.

Science Observation Blocks: Observation Blocks used to acquire scientific data.

Template: a high-level data acquisition operation. Templates provide the means to group commonly used procedures into well-defined and standardized units. They can be used to specify a combination of detector, instrument, and telescope configurations and actions. Templates have input parameters described by a template signature, and produce results that

can serve as input to other templates. As an example, an Acquisition Template takes target coordinates and produces through an interactive procedure the precise positions used later, e.g. to place an object on a slit.

Visitor Mode (VM): observing operations mode where the astronomer is present at the telescope while their observing program is being executed**.**

VLT Control System (VCS): the software and hardware tools that are used to control directly VLT instruments, telescopes, and related hardware. It enables and performs the acquisition of scientific and technical data. Originally designed for the Very Large Telescope, the VCS has also been installed at the La Silla New Technology Telescope (NTT) and 3.6m telescope.

1.5 Abbreviation and Acronyms

AT Acquisition Template

DFS Data Flow System

ESO European Southern Observatory

GUI Graphical User Interface

OB Observation Block

RB Reduction Block

SM Service Mode

UT (VLT) Unit Telescope

VM Visitor Mode

VLT Very Large Telescope

VRB Virtual Reduction Block

VCS VLT Control System

1.6 Conventions used in this document

- Tips are highlighted by a "candle bullet" (see for instance in Section 2.3)
- Examples and default values are typed using Courier fonts.
- ***Gasgano*** graphical user interface related items are typed in **bold**.
- Keystrokes combinations are typed in ***underlined italic***.

1.7 How to get help or report problems with *Gasgano* or this manual

In all of the following situations:

- ***Gasgano*** bug reports

- Suggestion for improvements to this manual
- Technical assistance

contact the **ESO User Support Group (**usg-help@eso.org**).**

2. OVERVIEW

Gasgano is a GUI software tool for managing and viewing data files produced by VLT Control System (VCS) and the Data Flow System (DFS). The main purpose for developing it was to provide a user-friendly tool capable to deal with the large amount of data generated by the VLT and the other ESO astronomical facilities.

As you will go through this guide, you will find out that the ***Gasgano*** distribution includes a series of satellite files that are instrument-dependent (e.g. classification rules). At the time of release, ***Gasgano*** includes instrument packages for the following (VLT) instruments: FORS1 and FORS2, ISAAC and UVES. As new instruments will be released to the community, these will be included in future versions of the ***Gasgano*** distribution kit.

Gasgano also provides functionality for accessing databases. This functionality is **not available in the current release of the tool distributed to the public**, even though some of the menus/ preferences allow to change options related to those databases.

2.1 Files supported by *Gasgano*

Gasgano understands and supports the specific types of files listed in the following sections, and ignores the others.

2.1.1 Raw Data

All VLT compliant instruments generate data which are stored as FITS files. In addition to the FITS standard keywords, ESO uses a set of primary keywords and a set of hierarchical keywords in its data products. The additional primary keywords, although not belonging to the official FITS standard are widely used by the community. One of them, MJD-OBS is essential for the DFS and ***Gasgano*** (e.g. to sort files). It represents the modified Julian Date at the start of the observation. Hierarchical keywords are used as means to group keywords that belong to the same logical entity, i.e. domain. A ESO hierarchical keyword starts by convention with "HIERARCH ESO" and is followed by words describing the domain. All ESO official dictionaries are available on-line through the following link:

http://archive.eso.org/Tools/DidRep/DidRepWebQuery

Important hierarchical keywords for ***Gasgano*** are:

HIERARCH ESO PROG ID = `60.A-9203(A)' : corresponds to the unique identifier of the parent Observing Run.

HIERARCH ESO OBS ID = 200103093 : corresponds to the unique identifier of the parent Observation Block

Note that ***Gasgano*** supports and uses shorter format for hierarchical keywords: The "HIERARCH ESO" words are not used and all spaces are replaced by dots. For example: HIERARCH ESO INS FILT1 ID is being shortened to INS.FILT1.ID.

Gasgano will be able to understand and load FITS files which do not contain those keywords.

However the functionality of the tool will in that case be restricted.

2.1.2 Pipeline Products

The VLT instrument pipelines generate for each raw frame and/or each set of raw frames one or more pipeline products of the types described below.

- **FITS images and tables**: These files contain a number of hierarchical keywords which are being used by ***Gasgano*** e.g.:

  HIERARCH ESO PRO CATG = 'MASTER_IMG_FLAT_BADPIX' : is used to identify a FITS file as being a pipeline product. The keyword value is also used to classify the data.

  The keywords RECi RAWj contain all the information describing the set of input frames (raw and calibration) used to generate the product. They provide means to asssociate the pipeline products with the corresponding raw frames and master calibration data. Note that pipeliene products created before October 2001 do not contain those keywords.

  HIERARCH ESO PRO REC1 RAW1 = 'ISAAC.2001-06-09T07:14:46.460.fits'

  HIERARCH ESO PRO REC1 RAW2 = 'ISAAC.2001-06-09T07:16:05.599.fits

  HIERARCH ESO PRO REC1 CAL1 NAME = 'IS_SSFM_010608A_SK_s03t_216.fits'

  HIERARCH ESO PRO REC1 CAL1 CATG = 'MASTER_SP_FLAT'

  HIERARCH ESO PRO REC1 CAL1 DATAMD5 = 'dda44920b1dda6fe222fc5a0a5092040'

- **PAF and ASCII files**

  Some of the instrument pipelines (e.g. ISAAC) create in addition to the primary pipeline products additional ones which contain for instance Quality Control information such as zero points. Those products are stored on disk either in a Parameter File (PAF) format or in an ascii format.

2.1.3 Reduction Blocks

Reduction Blocks are entities created and used in particular by the pipeline in its on-line environment. They contain all the information needed to reduce one or more raw frame, i.e. name of the recipe, list of appropriate calibration frames and name of the outputs to be generated. They provide means for associating pipeline products to raw and master calibration data.

2.2 What *Gasgano* does

The functionalities offered by ***Gasgano*** can be summarized as follows:

- **Data Grouping/Sorting:** ***Gasgano*** groups files located on a specified set of directories automatically into folders and subfolders. Files are put into the same folder/subfolder if they have some common features: same Observing Run ID/ same Observation Block ID within the same observing run. This holds for all types of files supported by ***Gasgano***. Users have the option to change the virtual view by first grouping their data around the

directory to which they belong or/and by defining an additional hierarchical level. The grouping is auto-refreshed every time a new file is loaded into the tool. We note that ***Gasgano*** is capable of recognizing raw files compared to pipeline products: the former are always displayed in blue, whereas the latter are displayed in red (reduced frames) or grey (e.g. reduction blocks). Furthermore per default , pipeline products are grouped together with the raw and master calibration frames that were used to generate them, making the raw-processed data association very intuitive. Within a folder files are ordered after their alphabetical names. The option exists to order them by the value of the MJD-OBS keyword.

- **Data Classification:** ***Gasgano*** overcomes the well-known difficulty of understanding what type a data file is (e.g. BIAS, DARK, SCIENCE, etc.) if one just looks at the filenames of his/her data. The archive filenames used for ESO data products are certainly unique, but user-unfriendly. ***Gasgano***, instead, automatically assigns a classification tag to all FITS files (tables and images) which have been loaded by the tool, by applying a set of keywords-based logical rules. These are instrument dependent and can be edited/modified by the user (cf Chapter 4). A set of classification rules for FORS1, FORS2, UVES and ISAAC is part of the ***Gasgano*** delivery package. Per default pipeline products are being classified using the hierarchical keyword PRO CATG.

- **Data Browsing/Filtering:** ***Gasgano*** allows the users to select a list of keywords to be displayed in the tree view for each FITS file which has been loaded by the tool. Users might browse these folders in a friendly and effective way, by filtering their contents based on logical keywords-based expressions that can be specified on the fly. Filtering Searching functionality allows searching folders for specific keyword values.

- **Data Viewing/Searching:** ***Gasgano*** provides means for viewing and searching all the relevant information of a data file. Viewing options include displaying FITS images and headers information (as a whole or based on a selected set of keywords only, e.g. by using the Filter option). The Search option allows searching keywords/keyword values within a displayed header.

- **Data Handling:** ***Gasgano*** allows the user to make mouse selections on the files displayed on the tree. The headers of selected files are visualized in detail (or just in part, depending on the filtering option used). Selected files may additionally be moved, copied, archived, piped into an external display tool. The Report option provides a useful tool to generate a report on the selected files. These options (Move, Copy, Archive, Report, ..) are ***Gasgano*** pre-defined commands. Furthermore, ***Gasgano*** may also be used as a front-end application to other UNIX tools, such as instrument pipelines. Once a set of files has been selected, users can send them as direct inputs to external executables and scripts.

2.3 *Gasgano* at a glance

The purpose of this section is to introduce the tool, so that interested users can familiarize

with its look and main functionalities. A more detailed description of a simple session will follow in the next chapter.

Figure 1 shows the main screen of the tool. It consists of three panels: the N**avigation Panel** at the top, the **File Association Panel** in the middle and the **File Detail Panel** at the bottom. Sliding horizontal bars separate the panels and allow panel re-sizing.

- On the left edge of each sliding bar there are two control buttons, of triangular shape, which allow expanding and collapsing of the panel, both upwards and downwards.

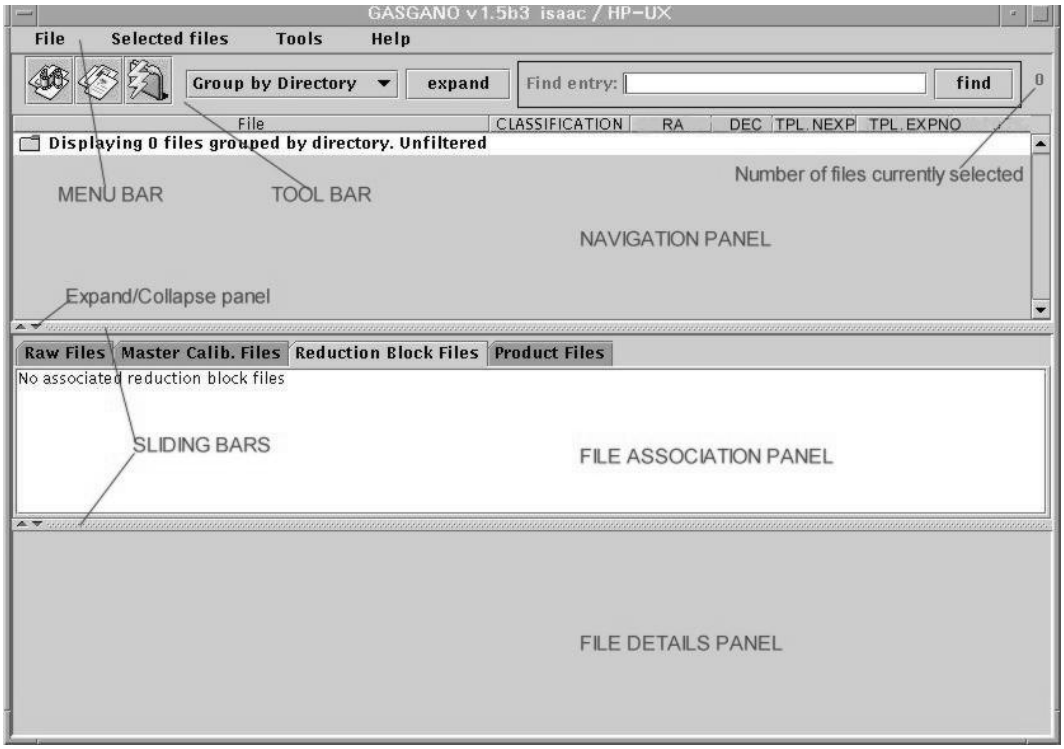

Figure 1: *Gasgano*'s main panel. The three sections are, from top to bottom, the Navigation Panel, the Association Panel, the File Detail panel. Sections are separated by horizontal sliding bars, which can be moved up or down with "click and drag" or can be collapsed using the little triangular icon on their left side. The Menu bar contains File, Selected Files, Tools and Help pull-down Menus.

I. The **Navigation Panel** can be identified by (from top to bottom) :

- the **Menu bar** (**File, Selected Files, Tools** and **Help**)
- the **Toolbar,** showing from left to right:
  - the **External Viewer button** which starts a viewing application (e.g. skycat) and displays the selected file,
  - the **Frame Report button**, which creates reports in a tabular format containing a user-defined list of keyword values of the selected file(s).
  - the S**cript Board button**, which launches the script board window, to run

scripts on the selected file(s).

- the **Grouping Menu button,** which changes the grouping of the files in the **Navigation Panel** (e.g. grouping around directory contents or telescope).
- the **Expand/Collapse button,** which unfolds/folds all the branches of the tree**.**
- the **Find Entry** field and f**ind** buttom, which allow the user to search the **Navigation Panel** for any entry (to be typed in).
- the **File counter:** at the right end of the **Toolbar** a number indicates the total number of files selected in any given moment.

Files loaded by the tool are shown in the **Filesystem** view of the **Navigation Panel**, This is what is called and we will refer to as a virtual view. For each FITS file in the view a set of (user-defined) list of keywords is displayed. The keyword names are displayed at the top of each column. In the example shown in Figure 1, these are respectively: CLASSIFICATION, RA, DEC, TPL.NEXP, TPL.NEXPNO.

II. The **File Association Panel** displays associations between set of raw frames, pipeline products and the master calibration frames used to process them. Its top part is organised into four different registers:

- **Raw Files**,
- **Master Calibration Files**,
- **Reduction Blocks,**
- **Product Files**.

Given a file selected in the filesystem view of the navigation panel, clicking on one of those registers will result in the corresponding and appropriate type of data being displayed.

III. The **File Detail Panel** shows the information contained in the header of a file, previously selected in the **Navigation Panel.** This panel can be used to filter the header using a user-defined list of keywords and to search for strings in the header itself. Depending on the type of file selected, the **File Details Panel** displays different details: the header of fits and tfits files, the content of PAF and ASCII files (including reduction blocks).

Once a set of data files has been loaded into the tool (more on this step in Chapter 3), the main ***Gasgano*** panel resembles Figure 2. Here, the **Navigation Panel** displays 98 files, as declared in the first row of the panel which indicates how many files are loaded, whether or not they are grouped or filtered. The default grouping mechanism is being used, that is the files are organised in folders, according to their Run ID (e.g. 67.D-0351(A)) and their OB ID (e.g. 105201). An additional hierarchical level has been added. Files belonging to a given Observation Block are grouped around the value of the TPL START keyword which indicates the time of the execution of the corresponding template. Each folder is identified and shown with a node icon, a folder icon and a line of text describing the folder (**folder line).** The latter may be, for instance, the combination of the OBS.PROG.ID + INSTRUME keywords, as seen in Figure 2 (e.g. 60.A-9021(A) ISAAC).

Clicking on any folder line or icon on the display views it, i.e. its content is displayed.

Raw files are displayed in blue, reduced in red, text files in grey. The icons for each file differ according to the file type. For each FITS file (image or table) the values of some selected and user definable keywords are displayed. In this example CLASSIFICATION (SPECTRA_COMBINED), DET.DIT (60.0000), DET.NDIT (1), INS.MODE (SWS1-…), INS.FILT1.ID (SK) and INS.GRAT.WLEN (2.1) are displayed.

The file IS_SSCM_010608A_SK_s03t_216_105201.fits is selected, as can be seen by the light blue highlight band. On the upper right corner of the window the number 1 gives the total number of selected files. The selected file is a pipeline product (an ISAAC short wavelength science combined spectrum) and the **Association Panel** shows its associated files. In the **Master Calib. Files** register two frames, with their corresponding path in the file system, are listed. These frames are the master calibration frames used during the pipeline processing of the corresponding raw science spectrum. The IS_SSAM* file is the arc frame and the IS_ISSFM* file the spectroscopic flat. Both files are loaded in the data set. Clicking on one of these files in the **Association Panel**, will move the selection in the **Navigation Panel** to highlight it.

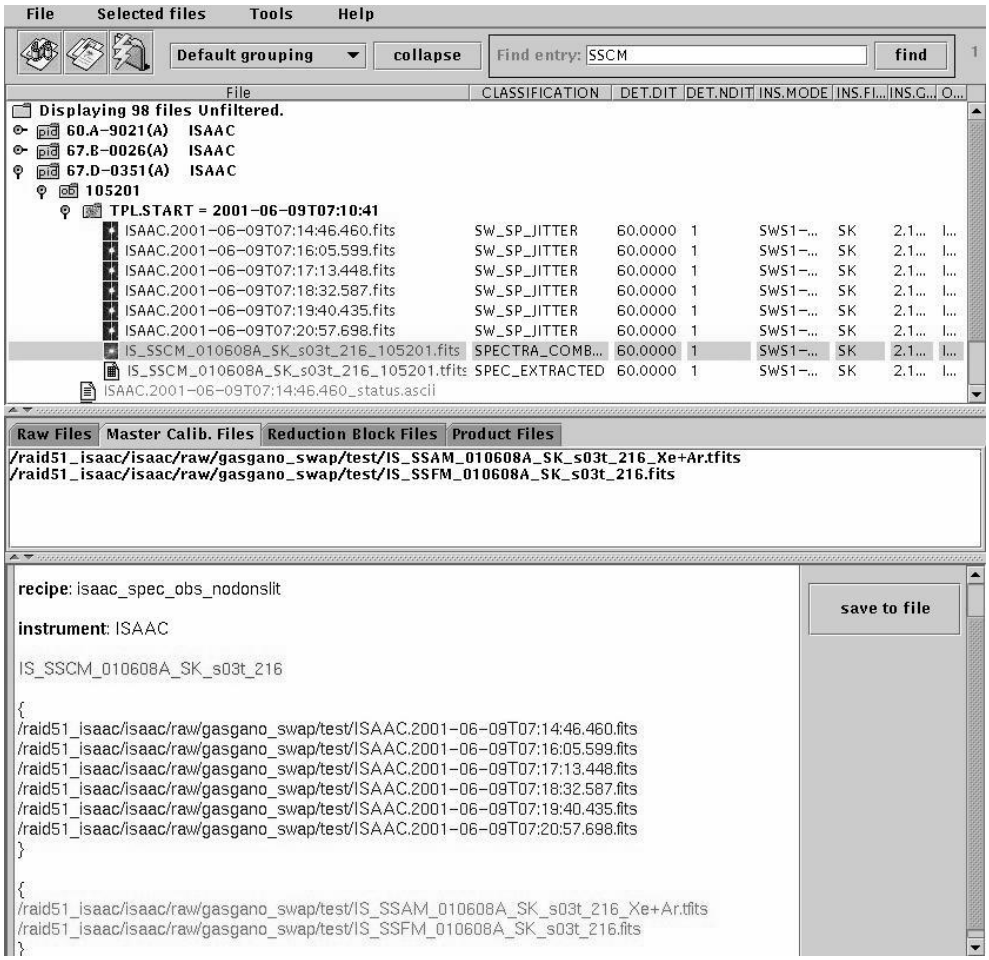

Figure 2: *Gasgano*'s main panel after loading a typical VLT data set. In this specific example files observed with the VLT Antu telescope and ISAAC are shown. Refer to the text for a description of this figure.

The lower **File Details** Panel displays the virtual reduction block associated with the science

spectrum. As explained in the glossary, reduction blocks contain all the "ingredients" used to generate a pipeline product . The reduction block can be an ASCII file created by the Data Organiser, or can be virtual, i.e. created on the fly by ***Gasgano*** using header keywords information (PRO keywords). Note that pipeline produced before October 2001 do not contain the required PRO keywords. In a reduction block for a given selected file these are the components displayed in the **File Details** Panel:

- the data reduction **recipe** used (isaac_spec_obs_nodonslit),
- the instrument (ISAAC), the basename of the data products: (IS_SSCM_010608A_SK_s03t_216),
- the set of raw files that were reduced (those included in the first set of curly braces),
- the optional set of master calibration frames (given in the second set of curly braces).

3. A SIMPLE SESSION

This chapter describes a simple session with ***Gasgano*** and its most basic functionalities. If you are a first time user please take some time to read this part.

3.1 Running *Gasgano*

To start the application, at the prompt, enter the following command :

- `gasgano &`

  If ***Gasgano*** starts correctly, the main screen will appear. If the DISPLAY setting is incorrect or .Xauthority is not correctly set, the application will not start, and messages similar to the following will be displayed on the screen:

  ```
  DISPLAY environment variable not set - exiting
  ```

  or

  ```
  Xlib: connection to "xt313:0.0" refused by server
  Xlib: Client is not authorized to connect to Server
  ```

- Beware that it may take some seconds before getting the main ***Gasgano*** panel.

3.2 Your local directory structure and configuration files

With the default installation, a directory ***Gasgano*** is created under ${HOME}. This directory contains all the executables and the accessory files. These files should never be modified directly by the user, except:

- the configuration file called .gasganorc created in ${HOME} (cf Appendix B for an example).
- the additional directories used to store:
  - keyword lists for reports and filtering of headers (defaulted to ${HOME}/***Gasgano***/keywords),
  - classification rules files,
  - rules for file filtering (both defaulted to ${HOME}/***Gasgano***/),
  - and user defined scripts (defaulted ${HOME}/***Gasgano***/scripts).

The user may change the default values by means of the **Preference editor** (see Chapter 4).

If ***Gasgano*** does not find the .gasganorc configuration file, or if this file is not correct, following information message will be displayed:

```
failed to load properties file: /diska/astro1/.gasganorc
Using default values
```

In such case, simply click on the OK button.

3.3 Exiting *Gasgano*

To exit ***Gasgano***, select **File>Quit** on the Menu bar or type ***CTRL+Q***.

3.4 Loading files into *Gasgano*

To load files into the application, click on **File>Add/Remove Files…**. A window appears (see below, Figure 3).

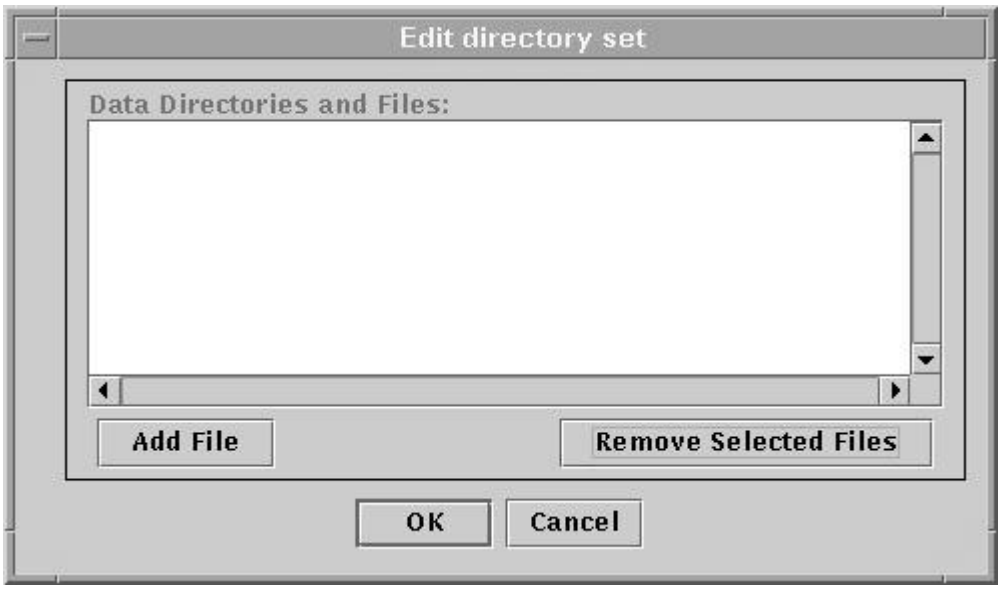

Figure 3: the Edit directory set **window allows adding or removing directories/files to/from the** Navigation Panel. **In this example no files are loaded. Clicking on** Add File **starts another panel that allows browsing in the file system.**

To add directories and files, click on the **Add File** button. A new window appears, as shown in Figure 4.

The first time the panel is opened, the content of the ${HOME} directory is shown.

Subsequently the application will "remember" the last selected directory during the same ***Gasgano*** session (it always display the $HOME directory when the tool is just started). It is possible to navigate through the file system either double clicking on a folder to change directory or using the first button after the combo box on the upper toolbar to move up one level.

One file can be loaded by a single click on its name and then pressing the **Add File** button. A set of files can be selected as well, using the Shift-click feature (for consecutive items), or the Ctrl-click (for non-consecutive feature).

A directory can also be loaded (its name should appear in the field **File Name**: at the bottom), but in that case, multiple selection is NOT possible (only the first selected directory is loaded).

Loading a directory will automatically load all its subdirectories. If some of them are not needed, they have to be selected in the **Edit directory set** panel (cf Figure 3) and removed by

clicking on the **Remove Selected Files** button**.** When removing directories or files from the list of loaded ones, multiple selection is possible. To learn how to do multiple selections, go to Section 3.6.

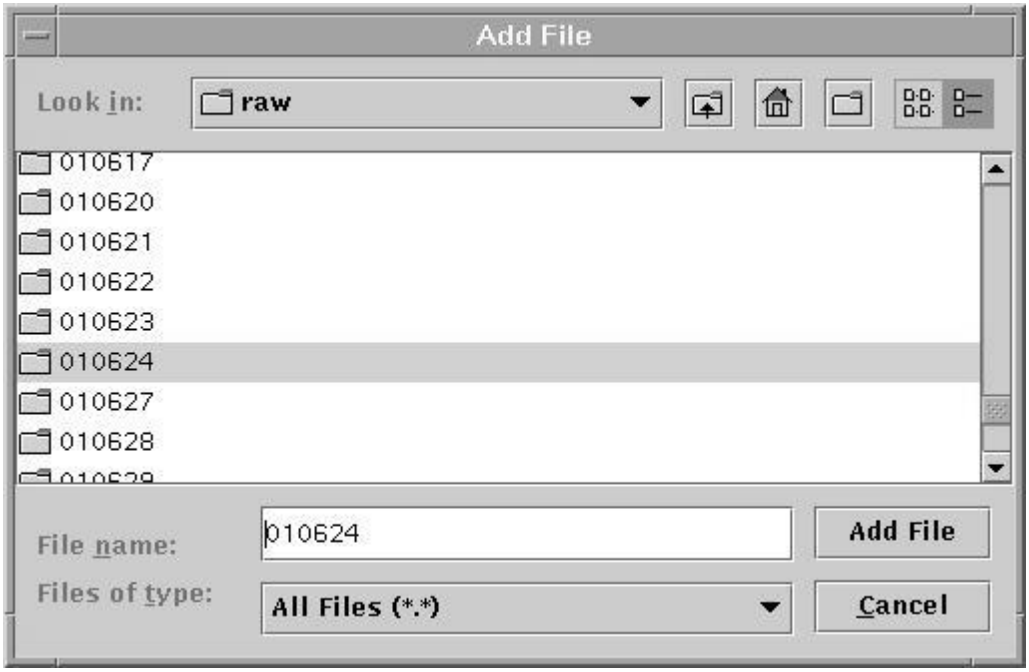

Figure 4: The Add File **Panel allows browsing the file system to select directories and files to be loaded into the application. To select a directory/file, single click on it and then click on the** Add File **button. Multiple selection when adding directories is** NOT **possible.**

Once files are loaded into the tool, the **Navigation Panel** will look like Figure 5 .

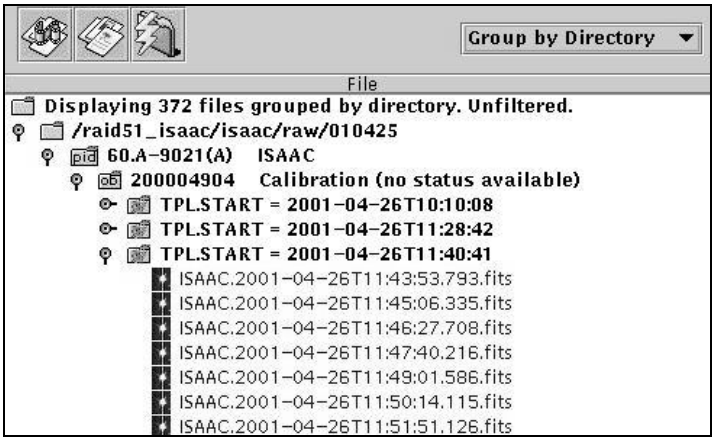

Figure 5: snapshot of the Navigation Panel **when the** Group by Directory **option is selected. In addition to the three levels OBS.PROG.ID (60.A-9021(A)), OBS.ID (200004904) and TPL.START, the path of the files is included at the top: these files are under the directory /**raid51_isaac/isaac/raw/010425**.**

3.5 Refreshing the contents of the folders

It is possible to request a rescanning of the directory contents if they have changed (for instance after moving/deleting files) by using the **Refresh option,** under **File>Refresh** in the

Menu bar. The automatic refresh option is useful when the content of the loaded directories changes frequently and can be activated by setting up the refresh time in the **File>Preferences… Data** register.

3.6 Changing the view

The view in the **Navigation Panel** can be changed in various ways, according to the user's preferences or needs.

3.6.1 Grouping files

On the **Toolbar** of the **Navigation Panel**, on the right side of the icon buttons, there is a selector menu (combo box), which allows to select the grouping mode. Its default value is set to **Default Grouping**. The other available options are **Group by Directory** and **Group by Telescope**. The default setting leaves the tree unchanged and it displays files grouped according to their OBS.PROG.ID, OBS.ID and any other optional keywords defined by the user. The **Group by Directory** option adds an extra layer on top of the default view and groups files according to their location in the file system. The complete absolute path of each group of files present in the same directory is displayed in the panel, as shown in Figure 5.

The **Group by Telescope** option works in the same way as the **Group by Directory** but adds a top layer which group files according to the keyword TELESCOP (e.g. ESO-VLT-U1).

3.6.2 Expanding and collapsing the view

The first time a data set is loaded into the application, the tree in the selected view (e.g. the **Default grouping**) appears collapsed, and only the list of the top folders (e.g. the list of OBS.PROG.ID) is displayed. In order to view the content of the folders below, single click on the folder line (at any position, i.e. on the node, on the folder icon or on the name). Be aware that a second click closes the folder (that is, the double click has no effect, and if you are fast enough no changes in the view are noticeable!!). Clicking through the various layers of the tree will expand/collapse it. It is possible to expand or collapse the entire tree at once, by means of the **Expand/Collapse** toggle button on the **Toolbar**.

3.6.3 Editing the list of keywords to be displayed in the file system view

The list of keywords to be displayed in the file system view can be edited/modified using **File>Preferences… Data** register. Short names for ESO hierarchical keywords (e.g. TPL.ID) might be used.

3.6.4 Resizing the fields

All fields in the **Navigation Panel** can be resized to accommodate the length of the keywords' values. The fields names are displayed in a bar just below the main **Toolbar,** in the order specified in the **Preferences Display and Printing register -> FileSystem – Displayed Keywords** item (accessible from the **Preferences** menu under **File**). When the mouse pointer is on the separation line between two fields, it changes its dimensions. Clicking the mouse left button allows grabbing the separation line and resizing the field to the preferred length.

3.6.5 Selecting Files

Any operation you may want to do on your file(s), such as viewing the header content,

displaying a frame with a viewer, copying/moving, creating reports, etc, requires a selection. Selected files appear highlighted in light blue. Single selection is done by single clicking on the object (a file, a directory, a folder). Multiple selection is possible in two ways:

After selecting the first file via a single click, the user can:

1) press the ***Shift*** key while mouse clicking on another file: this selects a block of contiguous files;

2) press the ***Ctrl*** key while mouse clicking on another file: this adds individual files to the selected list.

Other options for file selection, include:

1) Click on any node in the tree with the right mouse button: this expands the node and the sub-nodes below it and selects all the files within.

2) Click on any already expanded node with the right mouse button: this selects all files within that node.

Pressing the ***Ctrl*** key while clicking with the right mouse button on a node will allow multiple selections of nodes.

In all cases, if files are selected, the action of clicking will deselect all of them except the file on which the user clicks on.

In Chapter 6, Table 2 summarises all mouse click options.

- ***Right clicking*** on the uppermost folder icon in the **Navigation Panel** (e.g. in the one with the Displaying 98 files. Unfiltered **folder line**), selects all the files loaded by the application. To deselect them, either *left-click* on a single file or use the **Expand/Collapse** button on the **Toolbar.**

3.7 Finding tools

It is possible to find entries in the **Navigation Panel** and in the **File Details Panel** by means of the **Find tools**. To find an entry in the **Navigation Panel**, type the string to be looked for in the field **Find an entry** on the upper **Toolbar** and then click on **find**, to the right. Clicking twice or more on it will search for other entries of the same string in the panel. The string is searched "as it is", i.e. no wildcard characters must be entered. The search is case insensitive. Once the tool finds in the panel the line containing the string, this gets highlighted.

To find an entry in a fits header proceed in the same way but from the **File Details panel**, using the **Find in header** field on the panel's toolbar.

3.8 Displaying Files

Gasgano offers the possibility to define default file viewers, as explained in Section 4.1.2, even though NO viewers are distributed together with the application. The simplest way to display a file is to select it and to click on the leftmost button icon on the **Toolbar** (if you move your mouse pointer on this icon, you will see it reads External Viewer). The selected file is passed to the defined application/script, which displays the file. It is possible to define

three types of viewers, which will be automatically started by the tool, according to the type of selection in the **Navigation Panel**:

1) Single tfits table: the tool runs the tfits viewer

2) Single fits selection: the tool runs the fits file viewer

3) Multiple fits file selection: the tool runs a viewer that accepts data cubes in input

It is advisable to use a script which calls the viewer application and let it run in the background. This avoids ***Gasgano*** to freeze. Some viewers have remote interfaces which allow scripts to display frames in an existing session. ESO does not provide such interface yet but might do it in the future, for instance for skycat.

In addition to the **Toolbar** button, the viewer can be launched to display a selected file from the **Menu** bar**,** choosing **Files>Display ...** item**.**

3.9 Copying/Moving Files

Gasgano can move or copy files from one location to another one, as selected by the user. After selection of file(s), click on the **Menu Bar** on **Selected Files>Copy** or **Selected Files>Move.** A window similar to that shown in Figure 4 appears: it allows browsing into the file system in order to select the destination directory. When the location has been selected, click on the **Copy/Move** button. According to the operation here selected, ***Gasgano*** may behave in different ways:

1) If files are moved to a new directory (i.e. not loaded), the directory is automatically loaded by the tool

2) If files are copied to a new directory the directory is not automatically loaded

3) If files are copied or moved to a directory, which is already loaded, nothing happens. Manual refresh has to be forced (click on **File>Refresh** in the **Menu bar**.)

- It is not possible to delete a file from ***Gasgano***, but you can move files to a "wastebasket" directory in your file system and then remove them manually.

3.10 Creating a tar file

To create a tar file out of a set of selected files click on the **Menu bar**, **Selected Files>Tar…. .** A window similar to the one shown in Figure 4 appears. Select the destination directory and type the name of the tar file in the field **File name,** at the bottom. Click on the **tar** button. The tool will freeze" for the time needed to create the tar file (i.e., the time needed by the underlying UNIX workstation to perform the job). This can be quite long if the number of files is large. The syntax of the tar command used by ***Gasgano*** is:

tar –cbf 20 - <list of parameters>

where <list of parameters> is the list of files selected in ***Gasgano***.

3.11 Creating a keyword list file

Keyword lists are ASCII files used by ***Gasgano*** to filter file headers or to print reports, so that only the keywords present in the list are shown in the **File Details Panel** or printed.

To create a keyword list file from scratch, click from the **Menu Bar** on **Tools>Edit Keyword Lists…**. A new window, called **Keyword List Editor** appears (cf Figure 6). It contains two columns, **labelled Keyword Name** and **Report Label**. At the bottom there are the buttons **Insert Row** and **Delete Row**. On the upper **Menu Bar**, the **File Menu** contains the entries: **Open, New, Save As** and **Quit.**

Add as many rows as needed with the **Insert Row** button. In the first column enter the fits keyword as it appears in the file header, i.e. in uppercase and including the dot notation (but without the HIERARCH ESO part). The **Report Label** column allows the user to specify the label used by the application when creating a report. The syntax and the contents are free.

Once the list is complete, click on **File>Save As…** to save the file, but before remember to click somewhere else so that the last value you entered is stored !

The file should be (optionally) saved in the directory specified in the **Preferences Data** register for the keyword lists, since this is the directory where ***Gasgano*** looks when trying to load a keyword list file (GIVE NAME of DIR!).

To edit an existing file, click on **File>Open** in the **Menu bar** of the keyword list editor. The tool opens the usual file system browse window, in the default directory (as specified in the **Preferences**). Browse and/or select the file, then click **Load**. Edit the file as needed and then save. To create a new file, click on **File>New**.

File>Quit quits the editor without saving the file.

The keyword list files are standard ASCII files and can be edited with any editor, without having to use the built in editor.

Figure 6: The Keyword List Editor window. The list of keywords has been edited and the file saved to disk.

3.12 Filtering file headers

On the top part of the **File Detail Panel,** there are a **Load Filter** button and a **Filter** toggle on/off button. Click on **Load filter** to have the browsing window opening on the default directory for keyword lists. Select the file and click on **Load**. Click on the toggle button **Filter**, so that it is blackened. If a file in the **Navigation Panel** is selected, only the header keywords present in the keyword list file will then be displayed. To remove the filter, deselect the toggle button **Filter**. To load another keyword list file, repeat from the beginning.

When the fits header filtering is on, the **Find in header** works only on the currently displayed keywords. You may need here to enlarge the main ***Gasgano*** panel to see all keywords associated to the current selected file.

3.13 Filtering files

Gasgano offers the possibility to filter files loaded in the **Navigation Panel**, on the basis of keyword properties or values. The concept is similar to that of the classification rules. It is possible to set a constraint on a keyword, or a set of keywords by means of AND/OR logical expressions, and to have the tool displaying only those files that fulfill that condition.

To filter files select **File> Filter Files** from the **Menu** bar. A new panel, called **Filter Builder**, appears (cf Figure 7). It consists of an upper window, empty at first use. After each use of the filtering option, the filter rules will be memorized and automatically listed in this window for re-use. The lower window allows editing in the same way as the **Classification rules builder** window. Refer to Section 4.2, for a description of the syntax and how to edit these rules. The rule displayed in the lower window of the **Filter Builder** is the one that is

applied to the loaded data set, once the **OK** button is pressed. Rules are saved in the ${HOME}/***Gasgano***/filter.rul file, unless otherwise specified in the **Preferences**.

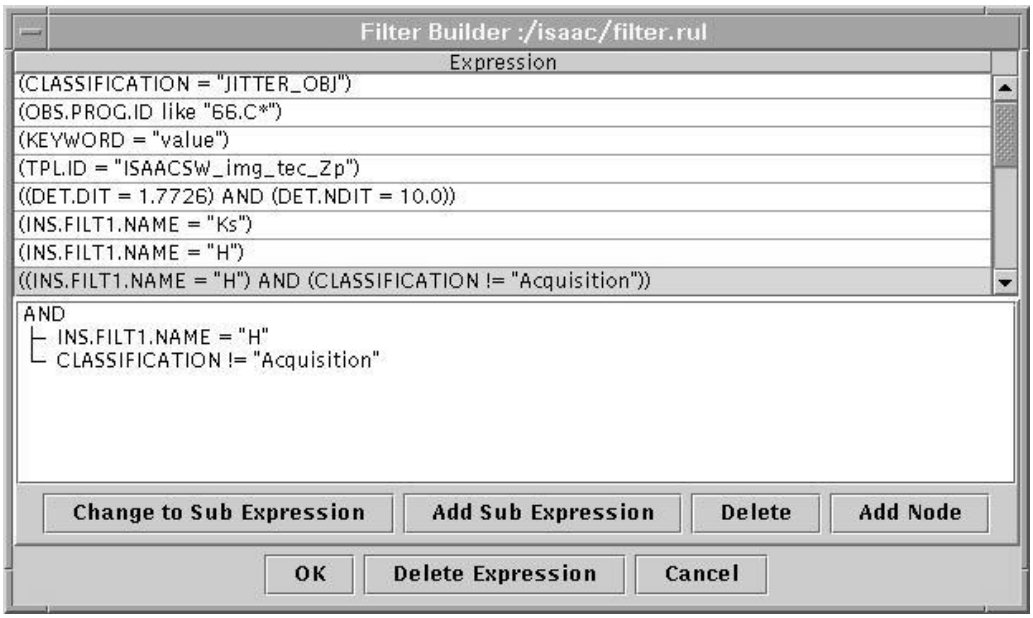

Figure 7: The Filter Builder **window. In this example, rules are saved in the file** filter.rul. **It is possible to configure the file and its location through the** File>Preferences **panel. The upper window shows a list of rules, with the selected rule (highlighted in blue) displayed in the lower window. This is the one which is applied to the data set loaded by the application: all files with filter set to "H" and with CLASSIFICATION other than "Acquisition" are displayed. The others are "filtered out" by the application.**

Once the filter has been applied, the first row of the **Navigation Panel** displays the following message (for this example) :

Displaying X files. Filter: ((INS.FILT1.NAME="H") AND (CLASSIFICATION!="Acquisition"))

Where "X" is equal to an integer number (including 0 if no files that fulfill the constraints are found). To remove the filter click on the **Menu bar** the **File>Unfilter files**.

- When a data set is filtered, the application keeps in memory the files that have been filtered out. Therefore, removing a filter does not require any re-loading of the files from disk.
- The filter files option is very useful when dealing with a large data set. It is usually recommended to load at first the entire data set and then to apply filters. One application is, for instance, to filter out all calibration files (DPR.CATG!="CALIB") when just a quick look at the scientific frames is needed.

3.14 Creating a report

A report is an ASCII file that displays in a tabular format file header keywords (or a sub-set of them, as specified) for a group of fits and/or tfits files. To create a report a keyword list file is needed. If none is available, create one as explained above. Then:

1) Select fits or tfits files in the **Navigation Panel** ;

2) Click on the **Report** button on the main **Toolbar** (the second icon button from left) or click on **Selected files>Report…;**

3) The **Load Keyword list** window appears. At first use it is empty, while for subsequent iterations the last loaded keyword list is kept in memory;

4) If it is empty, click on the **Load Keyword List** button at the bottom; the browse window opens in the default directories for keyword list files. Browse and select a file.

5) Modify the loaded file if needed, using the **Insert Row** and **Delete Row** buttons. To add a row above another one, select reference line and right click on the **Insert Row** button. Edit the fields as needed.

6) Click on **OK**.

7) The report is generated and the report window is opened (see Figure 8).

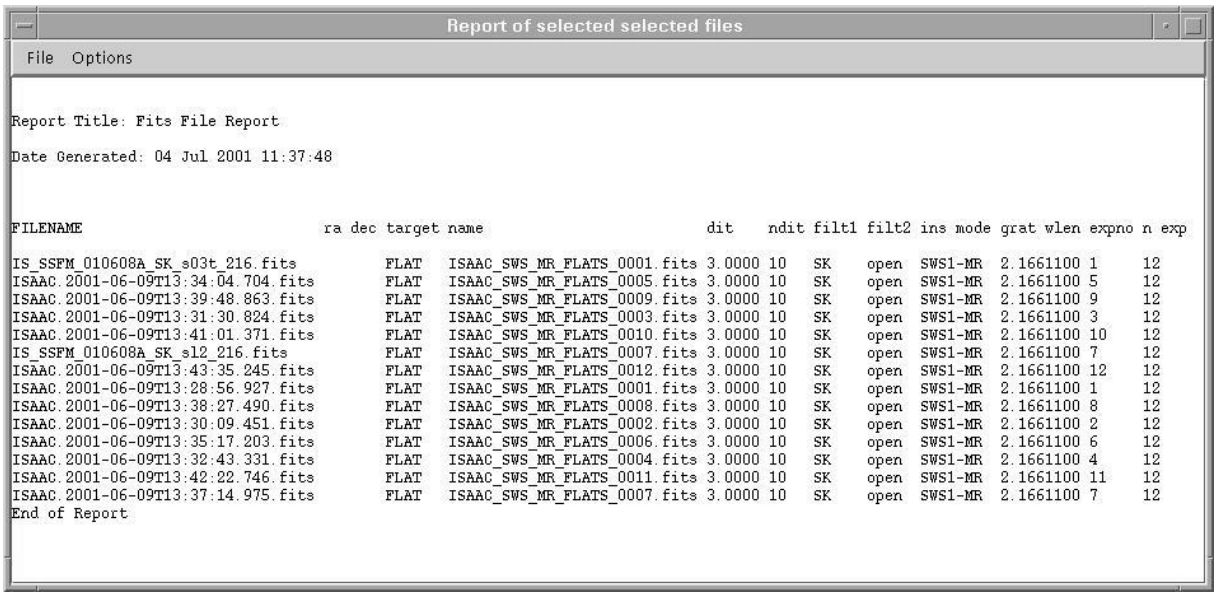

Figure 8: The Report **window. A list of files (previously selected in the** Navigation Panel**) is displayed with their keyword values. If a keyword is not present in the header, this is replaced by a blank space (like in the case of** ra **and** dec **for this example).**

On the **Menu bar** of the **Report** window the **File** and **Options Menus** are available:

The **File Menu** contains the **Print, Save As…** and **Quit** entries. **Print** sends the report file to the default printer (as specified in the **Preferences, Display and Printing** register**); Save As**… allows saving the file to disk (the usual browse window is open).

The **Options Menu** contains the **Select Fonts** entry. When selected it opens the **Edit fonts** window, which allows selecting a different font type and size from the one selected in the **Preferences, Display and Printing** register**.** The selected font is used only for the current report. To save it permanently edit the entry in the **Preferences.**

- Monospaced fonts are ideal for printing because they maintain the correct alignment of the columns. Be aware that other fonts may not do this.

4. CONFIGURING *GASGANO*

Gasgano was designed to be flexible and configurable to the greatest possible extent in order to accommodate a wide range of uses. The tool comes with a standard configuration set, which can be tailored to specific user requirements by means of the **Preference Editor**.

4.1 Running the Preference Editor

To access the **Preference Editor**, click on **File>Preferences…**. The **Preference Editor** window will appear. It is organised in registers: **Data, Display and Printing, Menus and Instruments.**

Clicking on any of these registers activates a configurable panel. All panels have an **OK** and **Cancel** buttons on the lower part. Accepting the preferences via the **OK** button will have the twofold effect of saving them into the .gasganorc configuration file and enforcing them in the current open session.

- Though not required, the preference file can be edited using any ASCII editor. Refer to Appendix B for the list of available entries.

4.1.1 Configuring the Data register

The Data register, shown in Figure 9, contains general settings of the file system: filenames of rules (for filter and classification), recognised text files extensions and the location of the keyword list directory (the directory where all user-defined keyword lists are stored). The **Data register** is divided into three main sections:

1) In the upper part, the **Data Directories and Files** field specifies which directories or files are automatically loaded at start-up. Adding entries to this panel and accepting the preferences with the **OK** button automatically loads the corresponding data sets into memory, if not yet loaded. Selecting a directory file selects all files in that directory. To load a directory:

   - click on the **Add File** button. A dialog box, which allows navigation through the file system, is opened.
   - Select the desired directory or file and accept the selection by clicking on the **Add File** button (a more detailed description of the **Add File** dialog box can be found in section 3.4).
   - To remove a directory from the loaded list, select it with a single mouse click and then click on the **Remove Selected Files** button. When removing items from the list, multiple selections (***CTRL+left click*****)** are possible**.**

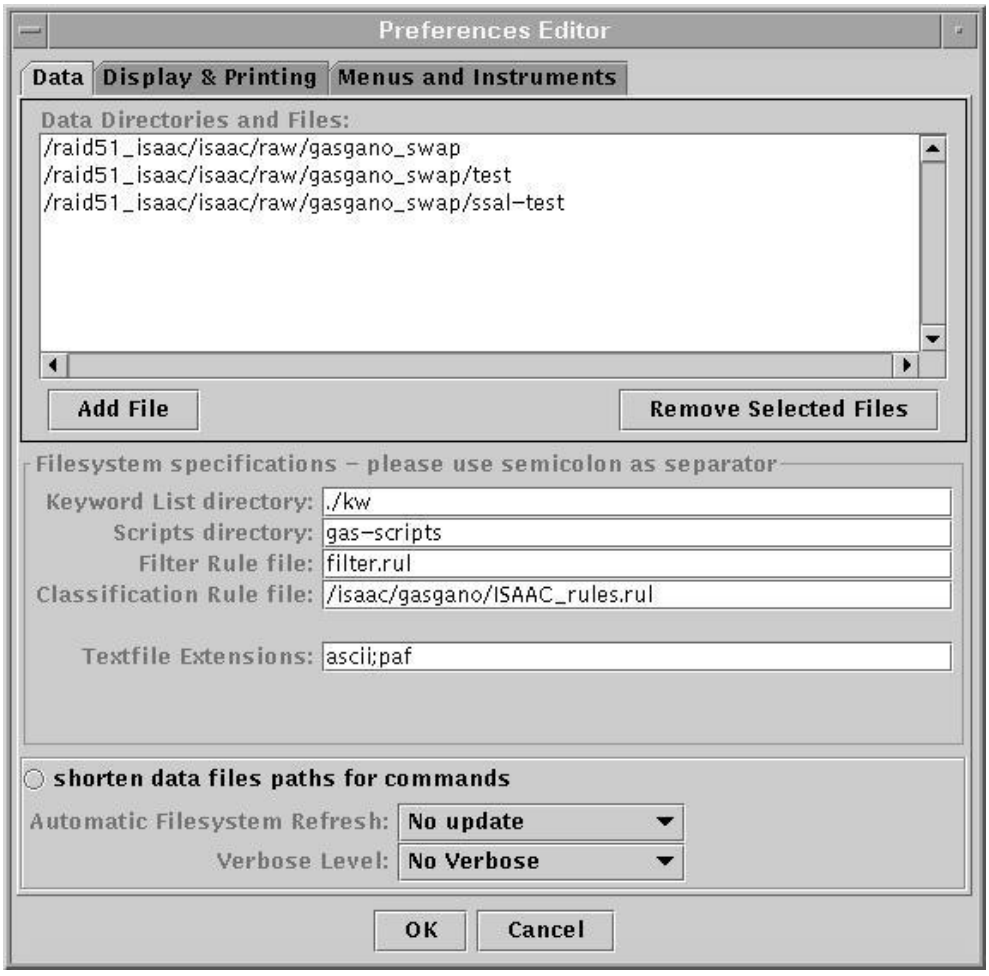

Figure 9: The Data register in the Preferences... Menu

- In the middle section of the register, the **Filesystem** specifications allow the user to define:

  a. the directory where to store user-defined keyword lists, to be used, for example, to create printable reports or to filter the fits header displayed in the **File Detail** panel in the main screen. The default value is `${HOME}/Gasgano/keywords`;

  b. the scripts directory, where to store user defined scripts or executables external command (see Chapter 5). The default value is set to ${HOME}/***Gasgano***/scripts, under the account's ${HOME} directory;

  c. the name of the file which contains the rules to filter the files in the **Navigation Panel;** the default file is filter.rul, located into the account's ${HOME}/***Gasgano*** directory;

  d. the name of the file that contains the classification rules for the raw files. It is defaulted to ${HOME}/***Gasgano***/VLT.rul.

  e. the recognised text files extensions; it is defaulted to ascii;paf, but as many extensions as desired can be added. Entries should be separated with a semicolon.

Example: in the default case, the tool recognises pippo.ascii as a simple text file. When the file is loaded and selected, its content is displayed in the **File Detail** panel. The file pippo.txt will not be recognised and therefore not loaded (ignored) by the tool.

- When this list is modified, a manual refresh of the data set must be forced (Click on **File>Refresh** in the Menu bar). Moreover, if an extension is deleted from the list, the change takes effect the next time the tool is started or, in other words, the files with this extension are not automatically unloaded by the tool.

- As already mentioned, ***Gasgano*** recognises file types based on their file extension. For fits files these extensions are .fits (2-D frame) and .tfits (table). It is not possible to configure the file extensions for fits or tfits files. Therefore, the fits file pippo.rtd will not be recognised by the tool. The user should take care of renaming these files to the default extensions.

❑ The bottom section deals with miscellaneous settings:

a. The **shorten data files paths for commands** button specifies whether the application should try to determine a common path for all data files supplied to an external command, and execute a 'cd' to that path prior to executing the external command itself. This can be useful to overcome shell limitations in the maximum amount of characters permitted on the command line when calling an external application or scripts. If the user selects a set of files with more than one path, the file names are not shortened.

b. The **Automatic Filesystem Refresh** sets the frequency for a refresh procedure: the tool scans the loaded directories and verifies whether a file has been modified (and therefore its header needs to be read in again) or whether files have been added/removed from a directory in the current data set. The possible settings are: **No update** (default), **1,2,5,30 minutes**.

- This option is useful when you are running data reduction procedures that create new files in the loaded directories or when you are moving files (from the command line) into/from loaded directories. This option should be turned off when you have loaded a large number of files, since the refresh action freezes the application for a certain amount of time, roughly proportional to the number of files loaded.

c. The **Verbose Level** option is used for debugging purposes only and specifies which debug level should be monitored while the application is running. This option should always be set to **No Verbose**, which is the default value. For more information about the meaning of the other verbose levels, refer to Appendix C of this manual.

- Whenever a **Verbose Level** is activated, diagnostic messages are written to stdout. When troubleshooting, your ESO technical contact may ask you to activate one of the verbose levels and report these messages.

- It is recommended to start ***Gasgano*** with "***Gasgano*** > my_log_file 2>&1 &" if you want to reproduce an error. This command saves all messages from stdout and stderr in the file my_log_file, which can be used for troubleshooting

4.1.2 Configuring the Display & Printing register

The **Display & Printing** register, shown in Figure 10, contains settings for the **Navigation Panel** in the **Main Screen,** settings for the printer options and for configuring the file viewers.

The configurable items in this register are:

1) **Filesystem – Display Keywords** editable list: keywords in this list are displayed by the tool in the **Navigation Panel.** The display follows the list ordering. The following keywords may be listed:

   a) All 8 character standard fits and ESO keywords: Example: RA, DEC, EXPTIME, ORIGFILE;

   b) HIERARCH ESO specific keywords, without the HIERARCH ESO prefix. Example: INS.FILT1.ID, TPL.START, INS.GRIS1.NAME;

   c) ***Gasgano*** meta-keywords, which are defined by the tool only but are available for display purposes: CLASSIFICATION, FILE_PATH and PIPE_PRODUCT. More on these keywords and their meaning in Section 6.4.

- CLASSIFICATION is, among the meta-keywords, the most useful to display: it represents the classification of a raw science/calibration file done by the tool according to the classification rules, which are fully configurable by the user. More on classification rules and how to configure them in Section 4.2.

To enter a keyword, type its name inside a field. The buttons **Insert Row** and **Delete Row**, allow the user to add more rows (a scroll bar appears on the right side, when there are more rows than those which fit in the field) or to delete selected row(s) (no multiple selection is available). The keyword names are case sensitive (must be uppercase) and the tool will mark the field as *INVALID*, whenever this rule is not respected. There are no syntax or validity checks, therefore if a wrong keyword name is typed in, no error message is displayed, but there will be an empty column in the **Navigation Panel**. The order of the keywords in the list is the order in which they will be displayed in the **Navigation Panel**.

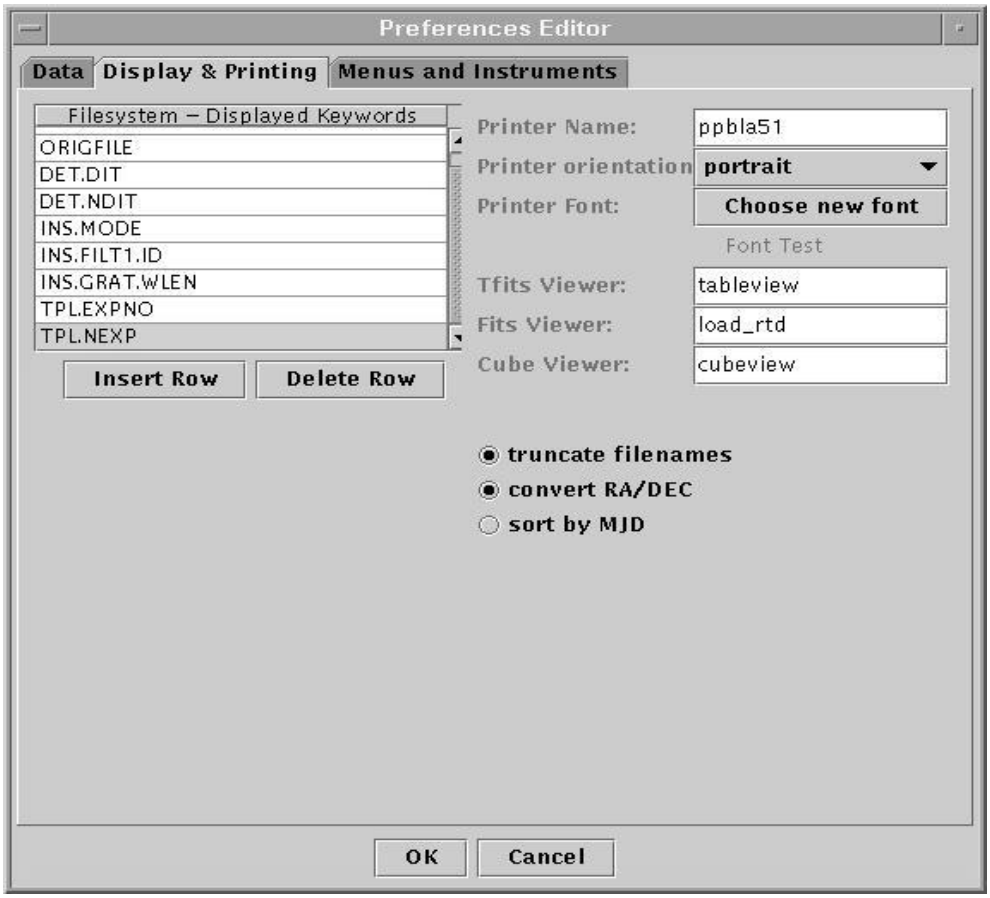

Figure 10: The Display & Printing **register in the** Preferences… **Menu. The fields are filled in with possible entries. These entries are dependent on the system settings and user preferences.**

- To add a keyword above an existing one in the list, right-click on the **Insert Row** button.

2) The printer set-up fields, situated on the right half side of the register, include:

   a) **Printer Name**, with no default value. Specify in this field the "Unix" name of your printer; e.g. ppbla51.

   b) **Printer orientation**, default value is portrait. It can be set to Landscape.

   c) **Printer Font,** default value is Monospaced, 8pts. The chosen font can be checked interactively as it is selected (Font Test). To change the font type, size and characteristics, click on the **Choose new font** button.

3) **The viewers setup** fields, located below the printer setup fields, allow specifying the UNIX command to launch file viewers. Three different viewers are possibly configurable:

   a) **Tfits Viewer**, launched by the application whenever a single tfits file is selected;

   b) **Fits Viewer,** launched by the application whenever a single fits file is selected;

   c) **Cube Viewer**, launched by the application whenever more than one fits file is selected;

It is responsibility of the user to define calls to external viewers. No viewers are delivered with ***Gasgano*** and the three fields are empty by default. It is possible to specify a viewer in two ways:

I. Type the name of the application as you would do from a terminal command line. The path to the application you plan to use should be in the ${PATH}. Example: skycat. The "&" is not allowed in the field and therefore ***Gasgano*** will freeze while the viewer is running. In order to be able to load one image after the other, by clicking on the filename in the application panel, it is recommended to use method II).

II. You call an external shell script, which in turns calls the correct application and let it run in the background.

4) The **truncate filenames** toggle on/off button allows the user to specify whether the filenames are displayed with their complete path or not in the Navigation Panel view. Default is **on**.

5) The **Convert RA/DEC** toggle button allows the user to specify whether the entries for right ascension (RA) and declination (DEC), if present in the **Displayed Keywords** list, should be displayed in sexagesimal or in decimal notation. Default value is **on,** i.e. sexagesimal.

6) The **Sort by MJD** toggle button, forces the application to display files sorted by the content of the keyword MJD-OBS. This option is useful when the files have been renamed following a user defined code, which would sort them in a different order than that of the UT time of observation. Default value is **off**.

4.1.3 Configuring the Menus and Instruments register

The **Menus and Instruments** register, shown in Figure 11**,** allows configuration of the virtual view in the **Navigation Panel** by adding additional instrument-specific hierarchical level to group files under common registers. The **Menus and Instruments** register contains an editable table, which contains two columns: the Instrument column lists all available instruments and the criteria column - both are editable. The **Instrument** column should contain the name of the instrument as specified by the INSTRUME keyword in the header. The **Criteria** column should contain a keyword which will be used to group together all the files having the same value for that specified keyword.

- For many instruments, and especially ISAAC, the use of TPL.START is particularly useful whenever an OB contains several templates. TPL.START is identical for all files belonging to the same template instance, and it is set to the UT starting time of its execution. For FORS1 and FORS2, in case of observation with many different masks, grouping by means of MASK.ID, might prove useful. If the OBS.ID is the smallest "meaningful" container for your data set, and you want to group your files according to another keyword but on a higher level of grouping, it is preferable to make use of the file filtering options, described in section 3.1.3.

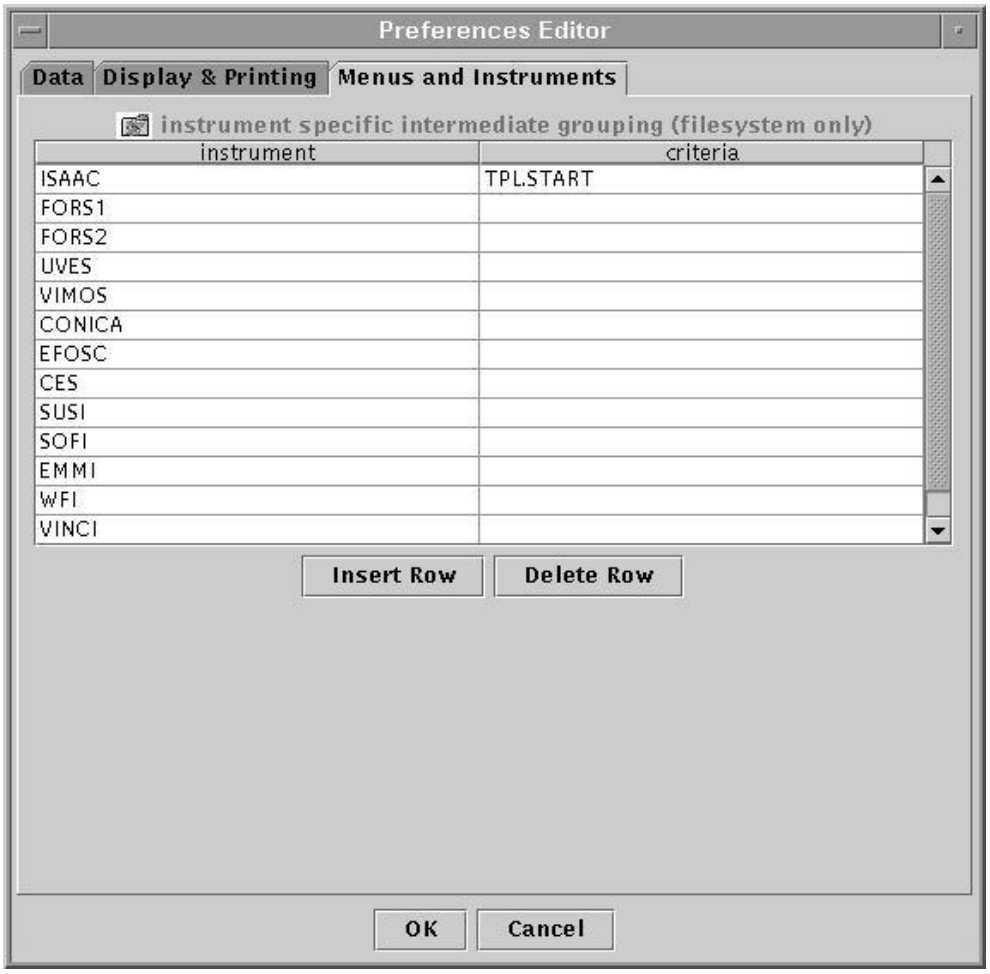

Figure 11: The Menus and Instruments **register in the** Preferences… **Menu**

4.2 Editing the classification rules

Gasgano is a powerful tool to classify files based on keyword expressions. The concept of classification of files is used within the VLT DFS by the Data Organiser, which classifies files according to rules related to fits keyword and then creates reduction blocks to be sent to the data reduction pipeline. Classification of files according to user specified rules might be useful for quick look assessment of a data package and for data reduction purposes.

Gasgano classifies files as they are loaded, comparing the value of specific keywords in their headers with the rules specified in the file ${HOME}/***Gasgano***/VLT.rul (by default). The location of this file can be changed via the **Data** register of the **Preference Editor** (see subsection 4.1.1). The rules are applied in the order they are written in the classification rule file. If there is any conflict between rules, identical or similar (intersecting) rules, the first in the file has the precedence. ***Gasgano*** is distributed to the users with a default classification rules file, which can be edited to meet the user's specific needs. This file contains basic classification rules for all the available VLT instruments and it will be updated every time a new instrument becomes operational (available in the distribution kit). In addition, instrument specific classification rules files are also distributed. Example: ISAAC.rul, UVES.rul, etc.

The use of the default files is encouraged for all first time users. More experienced users who

wish to modify the existing rules and to create their own sets of rules are encouraged to read sections 4.2.2 and 4.2.3.

If the user foresees working with files produced by only one single instrument, it is advisable to configure ***Gasgano*** to use the appropriate instrument-specific classification rules file, instead of the generic one. It is also recommended to edit the file and delete or update the classification rules according to user's specific data reduction needs. The classification rules supplied with ***Gasgano*** are similar to those used by the ESO data reduction pipeline, therefore tailored to an automatic data reduction strategy for large amounts of data.

- It is very important to spend some time configuring the classification rules. This has to do with the tool's performance. Assuming an average data package of more than 500 files, the time needed by the application to load them into memory and to classify them is proportional to the number of classification rules that have been specified. The same amount of time will be needed whenever you will need to reload the files or reclassify them (e.g., after having changed the lay-out of the columns in the **Navigation Panel** on the **Main Screen)**. Selecting just the rules you need, speeds up significantly this process.

The classification rules file is not in ASCII format and therefore can be edited only within the tool, by means of the classification rules editor. To access the file click on the **Menu Bar Tools>Classification rules…** . The **Classification Rule Builder** window appears, as shown in Figure 12.

The window is divided into two sections, the list of rules in the upper part and the editor panel in the lower part. At the bottom, a set of buttons allows maintenance of the list of rules. A rule selected in the upper panel is displayed in the lower panel, where it can be edited. For each rule, an Instrument and a Classification label must be specified. The Instrument label tells the tool to apply the rule to all the files that share the same keyword INSTRUME, whereas the Classification label appears in the **Navigation Panel** if the CLASSIFICATION keyword is on the list of keywords to be displayed **(Preferences Panel, Display** options register).

The upper panel consists of a table with three columns: The **Instrument** and **Classification** columns display the content of the above mentioned labels. The **Rule** column displays the classification criteria.

The lower panel allows display, creation and editing of classification rules. The rules are graphically rendered in a sort of **RPN (Reverse Polish Notation).** In short, RPN is a way of expressing arithmetic and/or logical expressions avoiding the use of brackets to define priorities for evaluation of operators. Here are some examples (not all of them are examples taken from the supplied classification rules files):

1) <u>The simplest rule</u>. The simplest rule classifies frames using a one keyword classification criteria:

   ISAAC TWFLAT TPL.ID="ISAACSW_IMG_CAL_TWFLATS"

   Where ISAAC is the Instrument, TWFLAT is the Classification label and the Rule states that if in the header of a fits file the keyword TPL.ID (observing template identifier) is equal to ISAACSW_IMG_CAL_TWFLATS, then this file must be classified as TWFLAT (twilight flat). In the lower panel the rule will be displayed simply as:

TPL.ID="ISAACSW_IMG_CAL_TWFLATS"

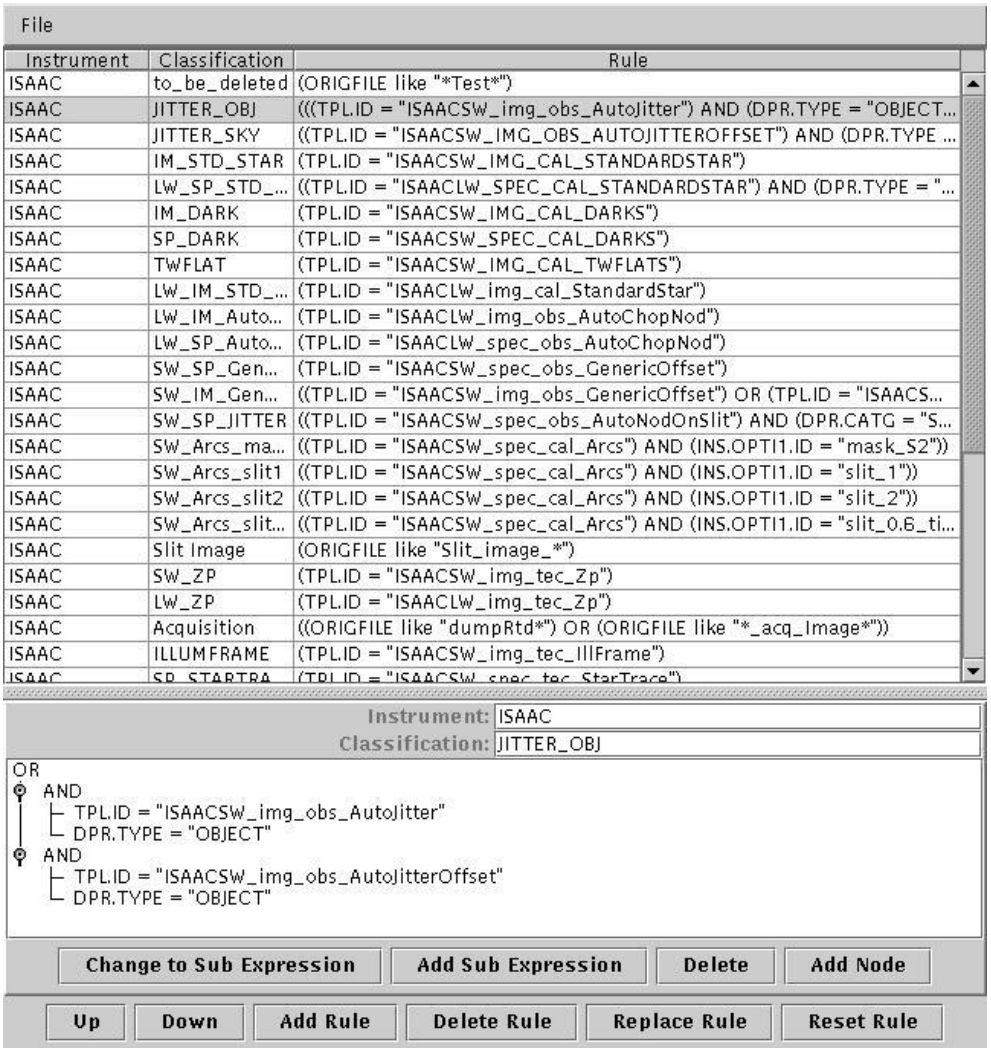

Figure 12: The Classification Rules Builder Panel. Some of the rules for the ISAAC instrument are displayed in the upper panel. The lower panel shows the classification rule for files of the type JITTER_OBJ. This rule is explained in more detail in the following text.

2) Double condition rule. In this case a double condition on one or two keywords is required to classify a file:

AND
TPL.ID="ISAACLW_IMG_CAL_DARKS"
DET.MODE.NAME="UncorrHighBias"

which classifies an ISAAC file as LW_Uncorrelated_Dark, whenever both conditions are satisfied. Note that AND is referred to as a **Node** and TPL.ID="ISAACLW_IMG_CAL_DARKS" is called a **Sub Expression.**

OR

ORIGFILE like ”dumpRtd*”
ORIGFILE like ”*_acq_Image*”

which classifies a file as Acquisition image whether its ORIGFILE (Original filename) keyword contains the string dumpRtd* or the string *_acq_Image*.

3) Triple (multiple) condition rule. The simplest example for this rule is to check if three (or more) conditions are all met or one of three (or more) condition is met:

AND
DPR.CATG=”CALIB”
DET.CHIPS=1.0
DPR.TYPE=”DARK”

which classifies a UVES file as a DARK_BLUE.

OR
OBS.PROG.ID like ”60.A-9023*”
EXPTIME < 60
INS.FILT1.ID != “V_BESS”

which classifies a file as FORS1 frame that was either observed under the engineering standard Run ID or its exposure time is less than 1 minute or its filter is not a V_BESS (this example is not in the distributed version of the classification rules).

4) Complex rules. These rules include combinations of nodes (AND and OR) and multiple sub expressions (conditions on keywords). The indentation gives the order of execution of each single block of constraints.

AND
  DPR.CATG=”CALIB”
  AND
    DPR.TYPE like “*WAVE*”
    DPR.TYPE like “*LAMP*”
  DPR.TECH=”SPECTRUM”

which classifies a frame as ARC_LSS for FORS2. The inline translation of this rule in “algebric” notation would be:

(DPR.CATG=”CALIB”) & ((DPR.TYPE like “*WAVE*”)&(DPR.TYPE like “*LAMP*”))&(DPR.TECH=”SPECTRUM”)

Instead, within ***Gasgano***, it is possible to simplify the rule by eliminating the innermost AND (&) clause and putting all 4 constraints under the same AND node.

AND
  DPR.CATG=”SCIENCE”
  DPR.TYPE=”SKY”

DPR.TECH="MOS"
OR
TPL.ID="VIMOS_mos_obs_Stare"
TPL.ID="VIMOS_mos_obs_jitter"

which classifies as MOS_SCIENCE frames taken with VIMOS. The corresponding expression in "algebric" notation would be:

```
(DPR.CATG="SCIENCE")&(DPR.TYPE="SKY")&(DPR.TECH="MOS")&((TPL.ID="V
IMOS_mos_obs_Stare")OR (TPL.ID="VIMOS_mos_obs_jitter")).
```

OR
AND
TPL.ID="ISAACSW_img_obs_AutoJitter
DPR.TYPE="OBJECT"
AND
TPL.ID="ISAACSW_img_obs_AutoJitterOffset
DPR.TYPE="OBJECT"

which classifies as JITTER_OBJ frames taken with ISAAC. Again, the corresponding expression in "algebraic" notation would be:

```
((TPL.ID="ISAACSW_img_obs_AutoJitter)&(DPR.TYPE="OBJECT")) OR
((TPL.ID="ISAACSW_img_obs_AutoJitterOffset)&(DPR.TYPE="OBJECT")
```

The same classification rule can be written in different ways. As a rule, it is recommended to privilege first human readability, like in the case above, and then to use the logical expression with the least possible number of constraints. The higher this number, the slower will be the tool in classifying files.

- It is the responsibility of the user to ensure that there are no contradicting rules or that no rule is a subset of another. The tool applies the rules for file classification in the order in which they have been specified in the classification rules file. Therefore, the first rule is always applied.

- You may need to use a rule, that is a subset of another one. For example:

AND
TPL.ID="ISAACLW_IMG_OBS_AUTOJITTER"
DPR.TYPE="OBJECT"

and

AND
TPL.ID="ISAACLW_IMG_OBS_AUTOJITTER"
DPR.TYPE="OBJECT"
SEQ.CUMOFFSETX = 0

If this order is maintained in the classification rule file, the second rule will never be applied.

Inverting the order of the rules ensures that both will be applied. As a rule, always put the most specific or restrictive rule on top.

4.2.1 Changing the order of the rules in the classification rules file

To change the order of the rules in the classification rules file, use the two **Up** and **Down** buttons at the bottom of the window: first select the rule that has to be moved and then click on the button so many times as many position changes are required.

4.2.2 Editing an existing rule – Very Advanced users

The easiest way to configure a personal set of classification rules is to modify the rules distributed with the tool: the file VLT.rul contain rules for all instruments of the VLT (commissioned and operative at the time of this release), while ISAAC.rul, FORS12.rul and UVES.rul contain the individual classification rules.

- It is possible to append a classification rule file to another one. Click the **Menu File>Append other rule file** in the **Classification Rule Builder.** The browse window opens. Find and select the file, click on **Append.**

Suppose you want to modify the following rule:

AND
TPL.ID=”ISAACLW_IMG_CAL_DARKS”
DET.MODE.NAME=”UncorrHighBias”

into

OR
TPL.ID like ”*Jitter*”
TPL.ID notlike ”*Generic*”

1) To edit the node AND, double click on it (alternatively single click and then click on the leftmost button EDIT). A window with a toggle button appears from which you can choose between the AND/OR option. Select OR. Confirm with OK.

2) Double click on the first term of the logical expression, and a window appears (see Figure 13). On the top part there are two fields to be edited with a toggle buttom in between. The first field contains the keyword, as seen in the header (without the HIERARCH.ESO part) of ESO standard fits files. Type in it TPL.ID. The syntax for keywords is case sensitive and requires dots in between strings.

3) Select with the toggle button the field “like”

4) Type in in the leftmost field: ”*Jitter*”

5) Confirm with OK.

6) Repeat steps 2) –5) to edit the second sub expression.

As a safety feature, whenever a wrong syntax value is inserted in the field and ***OK*** is pressed, the content of the field is reset to the previous value.

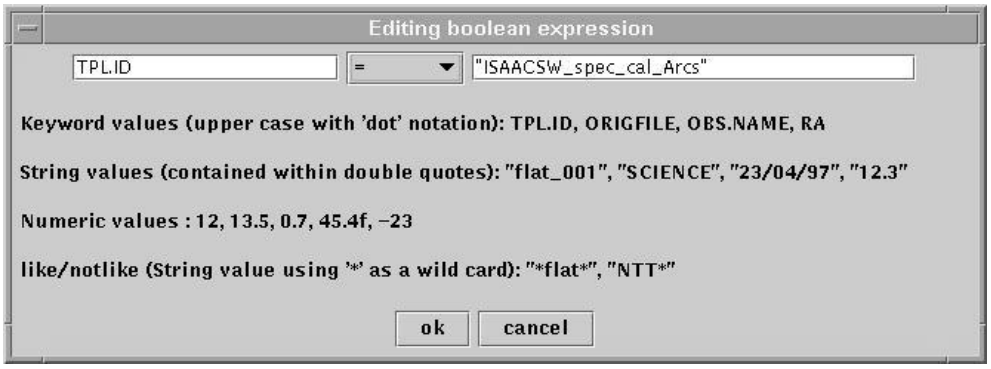

Figure 13: Editing Boolean expression window. Syntax and examples are explained in the text below the editable fields.

In the **Editing Boolean expression** window, and below the two fields that can be edited, there are some basic examples on how to type the rules. As a basic rule, beware of using always double quotes for strings, using * as wildcard character (this works only with the logical expression like and notlike), and do not use quotes for numeric values.

It is possible to add a node after a sub expression, simply selecting the sub expression and clicking on the **Add Node** button. Viceversa, it is possible to add a sub expression under a node, clicking on the node and then on the **Add Sub Expression** button. If an operation is not allowed (like adding a node under another node) the corresponding button is 'greyed out'. It is possible to transform a node into a sub expression, by selecting the node and then clicking on the **Change to Sub Expression** button. In this case, all sub expressions under that node, but the first one, are deleted.

Once the rule is modified, you submit it to the file, using one (or more) of the buttons present at the bottom of the window (**Up, Down, Add Rule, Replace Rule, Reset Rule, Dismiss,** respectively). There are various possibilities:

1) Click on the **Replace Rule** button, to replace the existing rule with the new one

2) Click on the **Add Rule** button to add this rule to the list of already existing ones (but remember to edit the **Classification field** to avoid having two rules with the same name**)**

3) Click on the **Reset Rule** button, to go back to the original rule (which is like UNDO)

Before adding or replacing the rule, be sure of checking that the **Instrument** field is correct.

- If you add a rule before changing its classification, or, in other words, you have two rules with the same name, you can select the wrong one and delete it via the **Delete Rule** button. Alternatively you can select it, modify the **Classification** field and then use the **Replace Rule** button

- What happens if there are two identical rules or two rules with the same name? Not much, really! The second rule is just redundant information, since the tool classifies the files using the rules in the order they are specified in the classification rule file. The second instance of the rule will therefore never be applied.

4.2.3 Creating a new rule – Very Advanced users

If the application is started with no classification rules file (you will notice that because a window with a warning will pop up at start), the **Classification Rule Builder** window will appear empty. To create a new rule, follow this step by step sequence:

1) Fill in the **Instrument** field. E.g.: ISAAC

2) Fill in the **Classification** field. E.g.: SCIENCE_imaging

3) Click on the rule editor panel, the KEYWORD="value" string appear. Either

   a) Click on the **Edit** button and edit the rule (***double click*** on the string is also possible).

   or

   b) Click on the **Add Node** button.

Case a. applies when one single constraint rule needs to be edited, whereas case b. allows adding a node with two sub expressions.

To edit and modify the nodes and/or the sub expressions proceed as described in the previous section.

4) Click on the **Add Rule** button.

Repeat steps 1)-4) until all the rules have been added.

4.2.4 Closing the Classification Rule Builder window

Once all the rules have been modified or created click on the **Dismiss** button, which will automatically save all changes in the rule file (which is the one specified in the **Preference, Data** register).

Once the window has been closed, the changes are implemented only if a "reclassify" is forced. To do so, click on **File>Reclassify** in the Menu bar**.**

4.3 The .gasganorc file

The .gasganorc file is created in the $HOME directory the first time the tool is launched. The first version of this file contains default settings, which the user can modify as explained in the previous sections.

Every time the set up of the tool is modified, by means of the **Preference** panel, the corresponding setting is saved into the .gasganorc file. This is a standard ASCII file, containing a list of entries, with the format:

`KEYWORD=value`. E.g. `CLASSRULE_FILE=/my_`***`Gasgano`***`_path/UVES.rul` specifies the name of the classification rule file.

A list of all the possible entries in the .gasganorc file and their meaning is given in Appendix B.

It is highly recommended to modify the preferences from within the tool, instead of editing manually the .gasganorc file.

5. AN ADVANCED SESSION

This chapter is intended for advanced users, who are already familiar with the basic functionalities of ***Gasgano*** but wish to exploit the tool further. This section is not complete yet and only describes how to call executables/scripts on selected files.

5.1 Running external executables

A set of selected files can be sent to a user selected executable. The files selected in the **Navigation Panel** are passed as command line parameters, one file per parameter, with the full path name. On demand, the full path may be dropped and the external command executed after a `cd' into the common directory of all selected files (This requires that all selected files are stored in the same directory.). In order to drop the path, activate "shorten data files paths for commands" button in the **File>Preferences Data** register.

There are two ways to send selected files to a customized program:

1. Select one data set and use the option **Run** from the **Selected Files** menu in the **Menu bar**. A browse window appears. Locate the script, select it and click on **Run** (or double click on the script name to execute). Click ***Ok*** in the confirmation window.

2. Use the S**cript Board**. This option is recommended when instrument dependent external commands have to be executed frequently.

5.2 The Script Board

The Script Board is an additional window, which shows a set executable files in a user-definable directory.. The window can be left open while using ***Gasgano***. It is activated by clicking on the **Script Board** button (the third from the left) on the **Toolbar** or selecting **Script Board** from the **Tools** menu or typing Ctrl-s when the mouse pointer is on the main ***Gasgano***'s window. To execute a command one has to double-click on it.

The **confirm** option will trigger a pop-up window in which one additional confirmation needs to be given before being able to run the specified executable.

On the Script Board the **Supply classification** option will pass the CLASSIFICATION Meta-keyword along with each file name. The sequence passed to the script is of the kind:

```
Filename1 CLASSIFICATION1 Filename2 CLASSIFICATION2 …
```

The **Run in background** option is disabled in the present version of ***Gasgano***.

The **Script Board** automatically looks for the executables in a directory (the **common script** directory. This directory cannot be a link to another directory) specified using the **Preferences Editor** in the Data register. When the **Script** Board is open and a data set (one or more files) is selected, ***Gasgano*** checks if all files share the same INSTRUME keyword. If this is the case, the **Script Board** looks into an instrument specific directory, specified as a subdirectory of the script directory named after the INSTRUME keyword itself. A typical example situation looks as follows:

```
<Gasgano-script-dir>/files common to all instruments (if any)

<Gasgano-script-dir>/ISAAC/files for isaac....
```

```
<Gasgano-script-dir>/FORS1/files for fors1....

<Gasgano-script-dir>/EMMI/1.57/files for EMMI release 1.57....,
```

where <Gasgano-script-dir> is the **common script** directory set in the **Preferences**.

If the selected files do not share a common `INSTRUME` keyword ***Gasgano*** shows the files in the `<Gasgano-script-dir>` directory. The **Override** button on the top left corner of the **Script Board**, overrides the instrument specific settings and repositions the view of the **Script Board** in the **common script** directory.

5.3 Passing a directory path to an executable

It is possible to pass the path of a directory (loaded into the application) on the file system to an executable. To do this:

1) Select the **Group by directory** view on the **Tool bar.** In the **Navigation** Panel a collapsed list of folders with the directories names (paths) should be visible;

2) Click on the desired directory with the **middle** button of the mouse. A beep sound should be heard. This way ***Gasgano*** has saved into an internal clipboard the directory name.

3) Double click on the desired script in the **Script Board.** If the **confirm** option on the board is active the confirm window should give the message:

```
/{gas-script-dir/../your_script /path_of_the _selected directory
```

- Clicking with the middle mouse button on the `OBS.PROG.ID` or `OBS.ID` folders in the **Navigation Panel** has the effect of saving the content of the folder line into the internal clipboard. Since the tool does not make a check on the meaning of the line but passes it "as is" to the executable, it is responsibility of the user to adapt the script according to the way ***Gasgano*** passes arguments. If for example, the user clicks with the middle button on the `OBS.PROG.ID` folder containing data for a run ID, the folder line passed to the executable is of the kind: `OBS.PROG.ID INSTRUME` (e.g. `60.A-9021(A) ISAAC`)

5.4 Example scripts

Gasgano can run executables, in the sense that it passes a list of parameters (the files currently selected in the **Navigation Panel**) to an external application. The executable may be anything (a shell script, a C program, etc) that can be run from the command line. It is responsibility of the user to create executables that handle the list of parameters (file names and/or classification string) in a correct way.

In all the following examples of scripts, substitute mypath with the appropriate path to the executable in your file system.

Examples:

1. Parameters passed to an executable: when files are selected and passed to an executable ***Gasgano*** may sends three different outputs. Consider the script that saves the list of parameters sent by ***Gasgano*** to the file /tmp/output.txt

```
##!/bin/sh
```

```
#
# Example shell script to be used with Gasgano:
# Print output – save to output.txt the list of
parameters sent #             to an executable.
#-----------------------------------------------------
------
echo $* > /tmp/output.txt
```

Assume that the following files are selected in the **Navigation Panel**:

```
ISAAC.2001-06-08T22:28:22.108.fits
ISAAC.2001-06-08T22:28:47.361.fits
ISAAC.2001-06-08T22:29:12.282.fits
ISAAC.2001-06-08T22:29:38.666.fits
ISAAC.2001-06-08T22:30:03.865.fits
ISAAC.2001-06-08T22:30:28.985.fits
ISAAC.2001-06-08T22:30:54.343.fits
```

1) Per default files are passed as a list of names with full path separated by a space:

```
/diska/test/ISAAC.2001-06-08T22:28:22.108.fits
/diska/test/ISAAC.2001-06-08T22:28:47.361.fits
/diska/test/ISAAC.2001-06-08T22:29:12.282.fits
/diska/test/ISAAC.2001-06-08T22:29:38.666.fits
/diska/test/ISAAC.2001-06-08T22:30:03.865.fits
/diska/test/ISAAC.2001-06-08T22:30:28.985.fits
/diska/test/ISAAC.2001-06-08T22:30:54.343.fits
```

   Even if the files are shown in the **Navigation Panel** without the path (the option **truncate filenames** in the **Preferences, Display** register is turned on), the path is passed to the executable.

2) In case the **Supply Classification** toggle button on the **Script Board** window is turned on, each file and its classification separated by a space are passed as a list:

```
/diska/test/ISAAC.2001-06-08T22:28:22.108.fits TWFLAT
/diska/test/ISAAC.2001-06-08T22:28:47.361.fits TWFLAT
/diska/test/ISAAC.2001-06-08T22:29:12.282.fits TWFLAT
/diska/test/ISAAC.2001-06-08T22:29:38.666.fits TWFLAT
/diska/test/ISAAC.2001-06-08T22:30:03.865.fits TWFLAT
/diska/test/ISAAC.2001-06-08T22:30:28.985.fits TWFLAT
/diska/test/ISAAC.2001-06-08T22:30:54.343.fits TWFLAT
```

3) In case the **Supply Classification** toggle button on the **Script Board** window is turned off and the **shorten data files paths for command** toggle button in **Preferences, Data** register is set to off, files are passed as a list separated by spaces:

```
ISAAC.2001-06-08T22:28:22.108.fits ISAAC.2001-06-
08T22:28:47.361.fits ISAAC.2001-06-08T22:29:12.282.fits
ISAAC.2001-06-08T22:29:38.666.fits ISAAC.2001-06-
08T22:30:03.865.fits ISAAC.2001-06-08T22:30:28.985.fits
```

```
ISAAC.2001-06-08T22:30:54.343.fits
```

2. A simple parser for the list of parameters: this shell script expects as input a list of file names from ***Gasgano*** (it assumes the option **Supply Classification in the** Script is turned off). The file should be saved in the **common script** directory (e.g. call it `crea_list_of_frames`). When executing, select the files in the **Navigation Panel** and double click on **crea_list_of_frames** in the Script Board**.** If the **Script Board** is positioned on an instrument specific script directory, use the **override** button.

```
#!/bin/sh
#
# Example shell script to be used with Gasgano:
# crea_list_of_frames.sh - creates a file framelist.ascii
to be used with eclipse recipes
#-------------------------------------------------------
------
#-------> move to the working directory
GAS_TMP_DIR=/mypath/Gasgano
cd $GAS_TMP_DIR

#-------> Clean old stuff

if [ -s ${GAS_TMP_DIR}/framelist.ascii ]
then
        rm ${GAS_TMP_DIR}/framelist.ascii
fi

#-------> Format input for eclipse routine

list=`echo $*`

for i in $list
do
        printf "%s\n" $i >> framelist.ascii
done
```

6. QUICK REFERENCE GUIDE

6.1 Summary of Inputs

Table 1: Keyboard shortcuts

SHORTCUT	ACTION
CTRL+q	Quit ***Gasgano***
CTRL+r	Open keyword list editor
CTRL+s	Toggle open/close the Script Board
CTRL+c (in the File Detail Panel)	Copy content of selection to Clipboard
CTRL+v (in a text file, such as Find)	Paste the content of the Clipboard

Table 2: Mouse actions

BUTTON	CONTEXT	ACTION
Left click	Navigation panel, on nodes, folder icons, folder lines.	Expand the corresponding item, showing its sub-components
Left click	Navigation panel, on files	Select the file
Right click	Navigation panel, on nodes, folder icons, folder lines.	Selection of all files included in the folder and its subfolders
Right click	Keyword lists editors (also in **Preferences** editor) onto **Add/Insert Row** button	Add/insert empty row above selected row
Ctrl-Left click	Navigation panel, on files	Multiple selection of non contiguous files
Shift-Left click	Navigation panel, on files	Multiple selection of contiguous files selection (first file must be already selected)
Ctrl-Right click	Navigation panel, on nodes, folder icons, folder lines.	Multiple selection of contents of non contiguous folders (nodes)
Shift-Right click	Navigation panel, on nodes, folder icons, folder lines.	Multiple selection of contents of contiguous folders (nodes)
Middle mouse button	Navigation panel, on nodes, folder icons, folder lines.	Save the content of the folder line into an internal clipboard. The content of the clipboard can be then used via the Script Board. It emits a beep.

6.2 Keywords

Gasgano uses keywords for expressions, classification, display and report generation. There are no keyword-dictionaries or defined keyword-sets in ***Gasgano****;* therefore entries of non-existent or bad keyword names will not be detected. For example a FITS file report requiring the PIPPO keyword to be printed will not produce an error. The column in the resulting report will be left blank.

ESO Hierarchical keywords can be expressed in both full (e.g.: HIERARCH ESO TEL AIRM START) and dot (e.g.: TEL.AIRM.START) notation. While ***Gasgano*** supports both forms of notation for data entry, typically the dot notation is used for display purposes.

6.3 Meta Keywords

Gasgano defines internally a number of keywords, which are used within the application but are not read from a FITS file header. These keywords are called Meta Keywords and have been designed to avoid any conflict with keywords read from a FITS file. This is achieved by giving all Meta Keywords "illegal" names.

The Meta Keywords currently in use in ***Gasgano*** are the following:

- CLASSIFICATION: a keyword representing the FITS file Classification, as defined by means of the classification rules. The keyword is "illegal" since it is not hierarchical and it is over 8 characters long.
- FILE_PATH: a keyword representing the full path of the data file on disk. This keyword is "illegal" as it contains an underscore ('_') character.
- PIPE_PRODUCT: a keyword indicating if the file is a raw frame or pipeline product. A raw frame will have a PIPE_PRODUCT value of "F" while a pipeline product file will have a PIPE_PRODUCT value of "T". This keyword is "illegal" as it contains an underscore ('_') character.

6.4 Keyword Expressions

This section discusses the use of keyword expressions in ***Gasgano***. Keyword expressions are used to classify FITS data files and filter the data file set.

Keyword expressions return a boolean value. They are a combination of Operands and Operators.

Operands have three different types: character strings, numbers and keywords.

- Keywords are represented as the names of the keyword and return either a Float or a String value.
- Strings are represented as characters enclosed in double quotes (e.g SKY). A String cannot return a Float value.
- Floats are numeric values, (12, 0.1, 45.67, etc.). Floats can not return a String value.

Supported Operators are listed in Table 3 (where LHS = Left Hand Side; RHS = Right Hand Side).

Table 3: Keyword expression operators

OPERATOR	DESCRIPTION
AND	Boolean operator which evaluates to **true** if and only if all of its two or more sub-expressions evaluate to true.
OR	Boolean operator that evaluates to **true** if one of its two or more sub-expressions evaluates to true.
=	Returns **true** if the LHS equals the RHS.
!=	Returns **false** if the LHS equals the RHS.
>	Returns true if the LHS is greater than the RHS.
>=	Returns true if the LHS is greater or equal to the RHS.
<	Returns true if the LHS is less than the RHS.
<=	Returns true if the LHS is less than or equal to the RHS.
Like	Performs a string comparison where the RHS must be a string operand. The RHS may contain '*' characters which are treated as wild cards when matching with the LHS. **Note: Only the "*" wild card is valid, there are no other wild card operators available with the Like operator.**
NotLike	As for Like but the opposite result is returned upon evaluation.

6.5 Keyword Expression Classes

A keyword expression is a tree structure with AND and OR operators as nodes and comparison operators (=, >, <, Like, etc) as leaves of the tree structure. Evaluation of the expression involves evaluation of each of its leaves and nodes in a depth first traversal until a result for the root node is determined.

The nodes of the tree are either AND or OR operators containing two or more branches. The leaves of the nodes must be comparison operators (=, !=, <, etc.). Each comparison operator contains a RHS (Right Hand Side value) and LHS (Left Hand Side value) and an operator.

Comparison operators may be evaluated as Float objects or String objects depending on the values of operands. A comparison will be evaluated using Float objects only if all operands return a valid Float value. Failing this, the comparison is tried using String objects. If comparison using Floats or Strings is not possible, the expression will fail to evaluate.

Figure 6 shows a couple of simple expressions and their string representations.

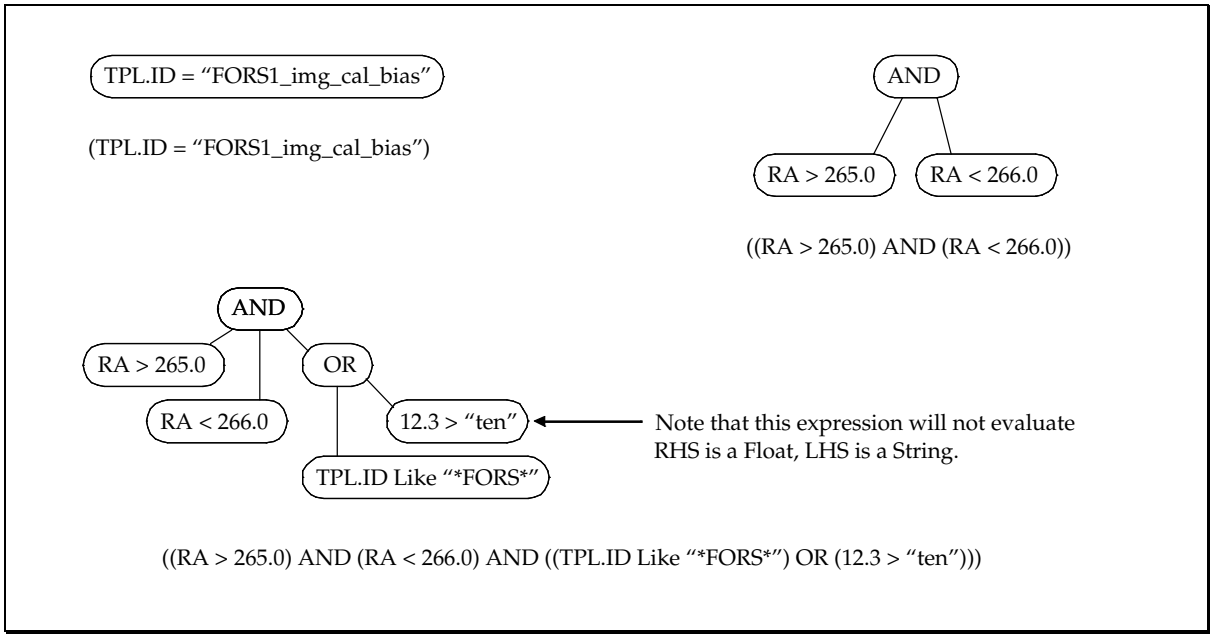

Figure 14: Expression Tree Examples

APPENDIX A – SYSTEM REQUIREMENTS

System definition and requirements

Software requirements: supported platforms are Linux RedHat 7.1, HP-UX 11.0, as well as Solaris 5.6. For viewing fits files, an image display software such as skycat or Saoimage is needed.

Hardware Requirements: ***Gasgano*** requires at least 100 Mb of RAM (depending upon data set size and usage) so it's advisable to run it on machines with not less than 512MB RAM. If visualised on a remote X11 terminal, make sure that at least 48 MBs of memory is available.

Processor speed becomes an issue only when accessing large numbers of compressed fits files.

Downloading *Gasgano*

Gasgano can be downloaded from the dedicated Web page, located at:

http://www.eso.org/gasgano.

Installation instructions are also posted on this web page.

Contact the User Support Group (usg-help@eso.org) if you need assistance.

APPENDIX B - .GASGANORC SETTINGS

Printout of .gasganorc (09.04.2002)

```
#Gasganó Properties Editor
CLASSRULE_FILE=gasgano/VLT.rul
COMPONENT_FONT=Dialog:14
CUBE_TO_VIEWER=
DATA_FILES=
DBOBS_DISPLAY_COLUMNS=
DBOBS_DISPLAY_COLUMNS_SIZE=
DBOBS_INSTRUMENT=
DBOBS_INSTRUMENTS=
DBOBS_PERIOD=
DBOBS_REQUIRED=
DBOBS_SERVER_ID=
DBOBS_SERVER_URL=
DBOBS_USERID=
DB_SERVER_URL=
DB_USER_NAME=
DISPLAY_COLUMNS=CLASSIFICATION:RA:DEC:ORIGFILE:TPL.ID
DISPLAY_COLUMNS_SIZE=131:131:130:131:130
FILTER_FILE=gasgano/filter.rul
FITS_TO_VIEWER=
FRAME_BOUNDS=111,158,920,650
INSTRUMENT_GROUPING=ISAAC=;FORS1=;FORS2=;UVES=;VIMOS=;CONICA=;EFOSC=;CES=;SUS
I=;SOFI=;EMMI=;WFI=;VINCI=;MIDI=;
KEYWORDLIST_DIR=gasgano/keywords
MENU_CUSTOM1=
MENU_CUSTOM1_LBL=Custom  Menu1
MENU_CUSTOM2=
MENU_CUSTOM2_LBL=Custom  Menu2
MENU_CUSTOM3=
MENU_CUSTOM3_LBL=Custom  Menu3
MENU_CUSTOM4=
MENU_CUSTOM4_LBL=Custom  Menu4
MENU_RETRIEVE=dpclient  -renameschema ARCFILE -outpath
MJD_SORTING=false
OBSTATUS_MAPPING=P:Partially defined D:Defined Q:queued for scheduling I:Initiated S:Started p:Paused
during execution A:Aborted X:Executed R:Reduced C:Completed L:released K:Cancelled
OB_STATUS_FROM_DB_REQUIRED=false
PRINTER_NAME=lp
PRINT_FONT=Monospaced:16
PRINT_ORIENTATION=P
RADEC_CONVERSION=true
SCRIPTS_DIR=gasgano/scripts
SHORTEN_FILES_PATH=true
SHORT_FILENAME=true
TEXTFILE_EXTENSIONS=ascii;paf
TFITS_TO_VIEWER=
UPDATE_TIMER=0
VERBOSE_LEVEL=0
```

The following table contains the correspondence between entries in the **Preference Editor Panel** and those in the .gasganorc file.

Field in preference editor	Entry in resource file .gasganorc	User's guide
Register Data		
Data_Directories_and_files with button [Add File] and button [Remove Selected Files]	DATA_FILES	The list of directories or files automatically loaded at application startup
"Keyword List directory:"	KEYWORDLIST_DIR	The default directory where the application looks for keyword list files
"Scripts directory:"	SCRIPTS_DIR	The directory where the application looks for external commands
"Filter_Rule file:"	FILTER_FILE	The file containing the rules for Navigation Panel filtering (Filesystem view)
"Classification Rule file:"	CLASSRULE_FILE	The file containing the classification rules
"Textfile_Extensions:"	TEXTFILE_EXTENSIONS	Recognized file extensions for ASCII files
"Shorten data files path for commands:"	SHORTEN_FILES_PATH	True/false: specifies whether command piped to the shell should be shortened
"Automatic Filessystem Refresh:"	UPDATE_TIMER	1,2,5,30 frequency for automatic filesystem scanning
"Verbose_Level:"	VERBOSE_LEVEL	Category of messages to be verbosed
Register Display&Printing		
Filesystem_Displayed_Keywords	DISPLAY_COLUMNS	List of semicolon-separated keywords displayed in the navigation panel
Printer_Name	PRINTER_NAME	Name of printer
Printer_orientation	PRINT_ORIENTATION	L/P: Landscape or Portrait
Printer_font	PRINT_FONT	The font used for printing
Font Test	----	
Tfits_Viewer	TFITS_TO_VIEWER	Application to be executed to visualize tfits files
Fits_Viewer	FITS_TO_VIEWER	Application to be executed to visualize fits files
Cube_viewer	CUBE_TO_VIEWER	Application to be executed to visualize a set if fits files
truncate_filenames	SHORT_FILENAME	True/false: specifies whether the filename in the navigation panel should be specified with its complete path.
convert_RA/REC	RADEC_CONVERSION	Specifies whether RA and DEC keywords are visualized in sexagesimal notation from the original decimal

Sort by MJD	MJD_SORTING	Specifies whether fits and tfits files in the navigation panel are sorted according to they MJD or according by name
Register Menus&Instruments		
"instrument specific intermediate grouping (filesystem only)"	-----	
IntermediateGroupTable	INSTRUMENT_GROUPING	List of semicolon separated pairs of INSTRUMENT=KEYWORD to be used for additional grouping

APPENDIX C – VERBOSE LEVELS

The verbose level can be set in the **Data** register of the **Preferences Menu**. The verbose facility is to be used for debugging purposes only, upon request from ESO Support personnel and is in no way meant to be complete, user-friendly or understandable for the users. The default value of this option should be set to **No Verbose** during normal use of the tool.

The possible values and their description are:

DataModelChange: changes like addition or removal of OBComponents andVRB to the application memory are displayed

DataModelEvent: notification to the Navigation Panel tree of addition/removal of components

ExternalCommands: not used

BaseName: verboses information on how the basename for an OBComponent isdetermined

Timing: activates collection of statistics for header read,classification and overall file read times.