

# The Galileo Table Editor

*P. Marcucci, M. Pucillo*

*Pubblicazione Osservatorio Astronomico di Trieste n. 1450*

## *Abstract*

*This document contains a description of the functionalities of the Galileo Table Editor program which, together with the Workstation Control Software and the Panel Editor, is part of the Workstation Software System of the Galileo Project (TNG).*

## **1.0 Introduction**

The Galileo Table Editor (TE) is a software tool used to define all the data tables used by the Galileo Telescope and Instrument Control Software. These tables can be edited in an interactive fashion, using all the capabilities provided by the OSF/Motif-XToolkit-X11R4 combination, thus including device independent graphics and network support.

This program is part of the Galileo Software Development Kit, distributed to telescope and instrument developers along with the *Galileo Interactive Panel Editor* [1] and the *Workstation Control Software* [2]. In the latter an in-depth definition of the structure of the whole software system is given. In the following pages, only the capabilities of the Table Editor as seen by a developer of TNG applications will be described.

## **2.0 The program**

The TE program resides in the “~/exec” directory and can be executed either from any location in the directory structure or from the root window menu (selecting the *Table Editor* item). The name of the TE program is “table”. After the initialization, the TE main window appears on the workstation screen (see Figure 1).

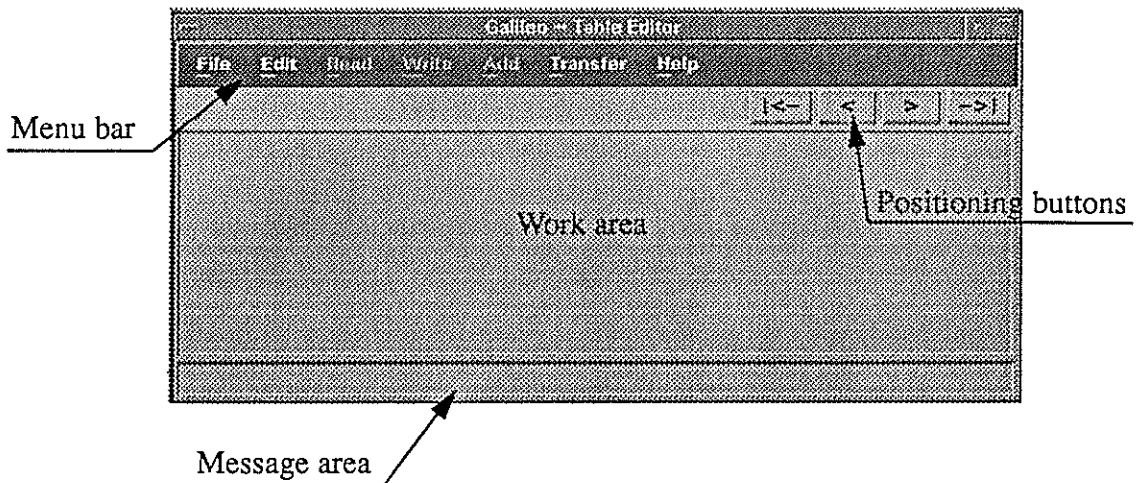


FIGURE 1. The main window

There are four major components in this window:

- the menu bar;
- the positioning buttons;
- the work area;
- the message area.

In the following section, a description of the convention on data file names is given.

## 2.1 Data file names

The Telescope Control System Software is based on a series of data tables that completely define the behavior of the overall system. These tables are defined in hierarchical order, reflected by the table file names themselves (see [2]).

The tree structure of the tables is the following:

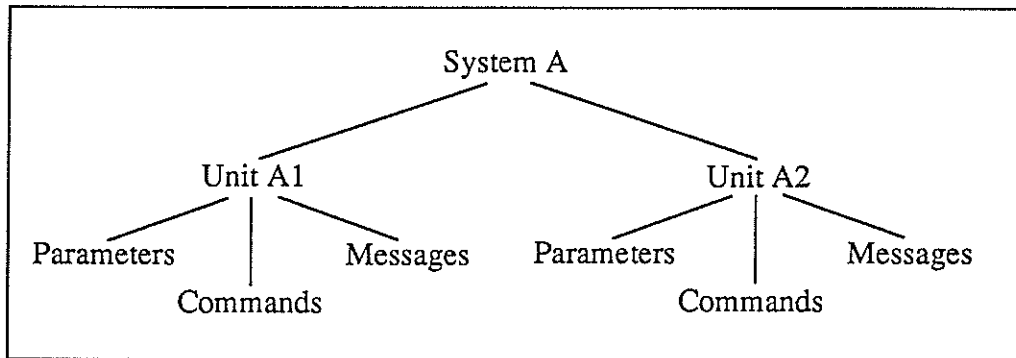


FIGURE 2. Tables tree structure

The file name is composed of two parts, separated by an underscore character, reflecting the tree structure of the tables: the first defines the system the table is related to, while the second defines the unit. The type of table (parameter, microcommand, etc.) is reflected by the file name suffix. It is reflected and not defined, because the structure of the table is internally described and self-consistent. The system part is 4 characters long whilst the unit part consists of 3 characters (see Table 1). A complete description may be found in [2].

TABLE 1. Example file names

Table file name	Table description
systems.scf	the main system definition table
vmaz.ucf	the description of the units of the VME system vmaz
vmaz_com.pcf	the list of telemetry parameters for the unit com of the system vmaz
vmaz_com.mccf	the list of the microcommands for the unit com of the system vmaz

## 2.2 The data directory

Table files are typically stored in a single directory defined by the environment variable TNGDATA. Every operation such as File/New or File/Open add this variable to the beginning of the table file name to ensure that all data files reside in this directory. In the current version of the system this variable is defined as “~/data”.

## 2.3 The menu bar

Seven selectable items can be found here which allow the user to fully manage the tables content: File, Edit, Read, Write, Add, Transfer and Help.

### 2.3.1 File

Pressing the File button will cause a pulldown menu to appear, presenting four choices: New, Open, List and Exit.

#### 2.3.1.1 *File/New*

Selecting this item will cause another pulldown menu to appear, presenting six choices which correspond to the six data types used (at present) in the Galileo Workstation Control Software; they are Systems, Units, Parameters, Microcommands, Panels and Messages. Note that Panel data files can be edited in a more interactive way via the *Interactive Panel Editor*. After selecting one of these file types, a dialog box will appear whereupon the user will be requested to enter the data file name. The user does not have to specify the table suffix as that will be added automatically by the program. At this point, a file of the selected type with the defined name will be created. The current version of TE does not open the newly created file, this operation is left to the user via the File/Open command.

#### 2.3.1.2 *File/Open*

Selecting this item will cause another pulldown menu which is similar to the one in the "File/New" operation. After selecting a file type, a dialog box will appear and the user will be requested to choose from a list of previously created files. At this point the file is opened and the work area is filled with field definitions and values (see Figure 3, "Filled work area," on page 4).

#### 2.3.1.3 *File/List*

Once a table is opened, a list of the existing records can be obtained by selecting this item. A little dialog box is displayed and may be moved to a suitable position on the screen (see

Figure 4, "List dialog box," on page 4). By double clicking on an item in the list box, the relative record will be read and displayed in the work area.

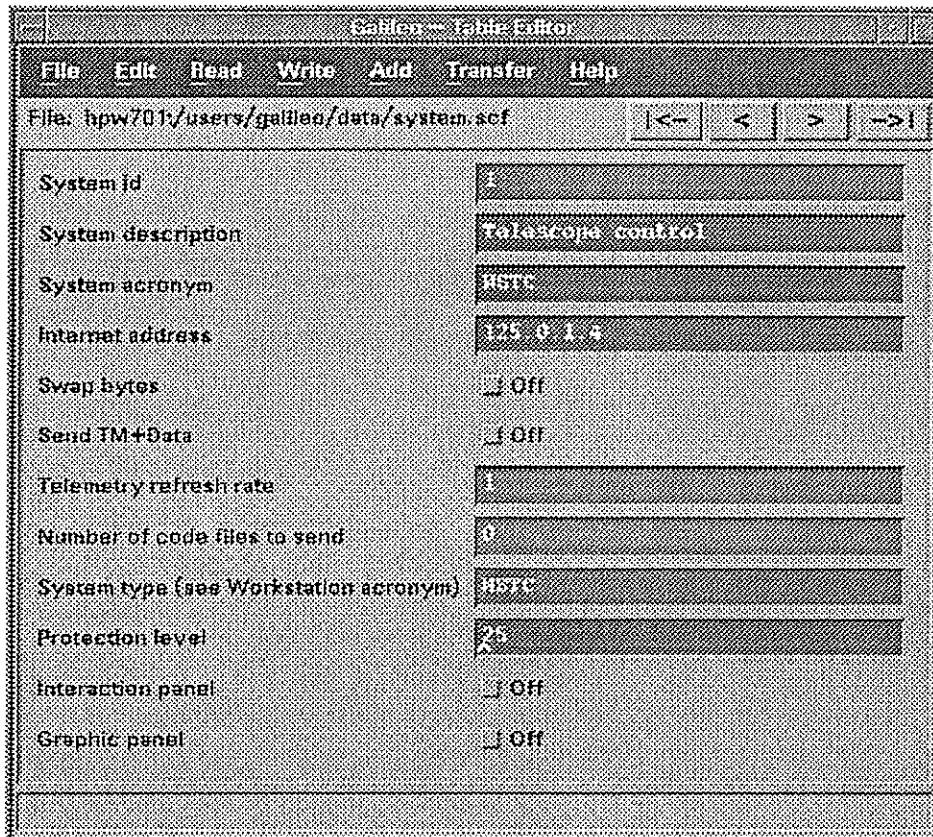


FIGURE 3. Filled work area

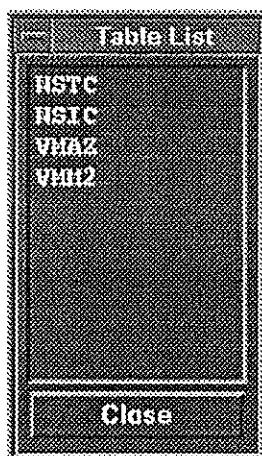


FIGURE 4. List dialog box

#### *2.3.1.4 File/Exit*

By selecting this item the TE will close the currently opened file thereby returning to the operating system.

### **2.3.2 Edit**

The Edit menu entry allows the user to perform edit operations on the records in the file.

#### *2.3.2.1 Edit/Undelete*

If a record were deleted using the Edit/Delete command (see “Edit/Delete”), it could be undeleted and added to the end of file.

#### *2.3.2.2 Edit/Delete*

This command deletes a record from the file and performs a “shuffle” operation, renumbering all the records in the file. The deleted record is stored in a temporary buffer, ready to be undeleted (see “Edit/Undelete”).

#### *2.3.2.3 Edit/Revert*

If the user has made modifications on a record and needs to cancel them, he has only to select this command and the record will be re-read from the disk.

### **2.3.3 Read**

Reads a record from the disk using the number in the first editable field to index the record in the file.

### **2.3.4 Write**

Writes a record on the disk using the number in the first editable field as the record index. **ATTENTION:** there is no such thing as a modification check, so care must be taken not to overwrite other records with the same record index.

### 2.3.5 Add

Writes a record on the disk using the next available record number as the record index. This command results the best choice to write new records.

### 2.3.6 Transfer

The Transfer menu bar item allows the user to import or export tables in ASCII format, compared to the proprietary operating system dependent format used by TE. This format has proved a convenient mode to transfer tables using e-mail programs like the DEC MAIL utility.

#### 2.3.6.1 *Transfer/Import*

The Transfer/Import command offers the user a choice among the table types and then subsequently a list of ASCII files stored in the data directory (see “The data directory” on page 3). Selecting one of them will cause TE to convert it into the proprietary format and to open the resulting table. Note that ASCII file names are composed using the original table file name with a capital “X” added at the end.

#### 2.3.6.2 *Transfer/Export*

This command converts the currently open table in ASCII format, adding a capital “X” to the end of the table file name and storing the resulting file in the data directory (see “The data directory” on page 3). For example, exporting the file `system.scf` will cause an ASCII file called `system.scfX` to be created. The format on an ASCII file is reported in Figure 5.



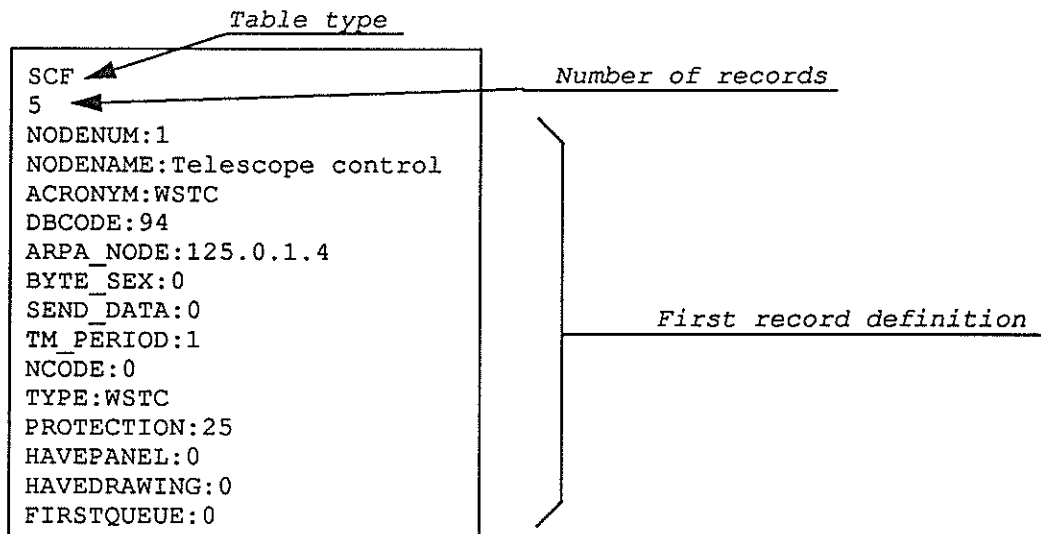


FIGURE 5. Import/Export file format definition

### 2.3.7 Help

Activating this command will transfer the user to the integrated Galileo help system.

## 3.0 Positioning buttons

The positioning buttons act as a VCR-style interface, with the possibility of moving the current record pointer one record forward, backward, to the beginning and to the end of the file (see Figure 6, "Positioning buttons," on page 8).



FIGURE 6. Positioning buttons

## 4.0 Using the Table Editor

The Table Editor uses the interaction model provided by OSF/Motif 1.1 with full use of the mouse and keyboard shortcuts. For example the Read command can be executed either

by pressing the Read item in the menu bar with the mouse or by typing Meta-R on the keyboard. Navigation between editable fields in the work area is performed using the up and down arrow keys.

There are two typical modes of use of the TE program: adding a new table or editing an existing table. In either modes, the user needs to add, modify or delete records in the table.

In order to read a record from the disk file, the user has to specify the record index in the first editable field of the work area and then issue a Read command (either by pressing Read in the menu bar or by typing Meta-R on the keyboard); the record will be read and displayed in the work area. If the record index is incorrect (if it is zero or greater than the number of records actually in the table) an error window will be displayed and the user has to acknowledge it before moving further ahead (see Figure 7).

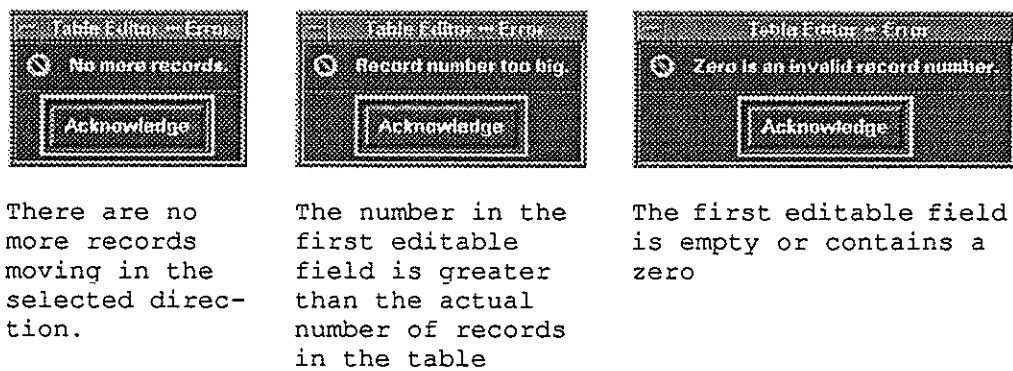


FIGURE 7. Error dialog boxes

When modifications on a record are made, they can be written onto the disk using the Write command. If the work area contains the definition for a new record, the user can choose the Add command to add the record to the end of the file. Note that in this case the contents of the first editable field are not considered, and the record index is defined by adding one to the number of records already present in the file.

## 4.1 Defining a new system

The definition of a new system occurs when a new VME or a new workstation is added to the telescope network. The user has only to add the system specification to the end of the `system.scf` file. This is the only file with a predefined name as all the others have a name composed by acronyms defined in `system.scf`. But defining a system is not sufficient, the user has also to describe the units contained in the system.

## 4.2 Defining a new unit

The name of the unit file of a system is composed of the acronym of the system itself plus the `ucf` suffix (e.g. `vmaz.ucf`). All the units contained in the system are defined herein. If the system is a workstation (its acronym starts with the "WS" letters) the units are the processes started at the activation of the main control program for that workstation. If the system is a VME (its acronym starts with the "VM" letters) the units are logical or physical subdivisions inside the VME. The unit file for a VME has to contain at least one record that describes the Communication Unit. Its acronym is `COM`.

## 4.3 Defining parameters, commands and messages files for a unit

After a unit is defined, telemetry parameters, microcommands and messages have to be added to it. The name for these files may be obtained adding the unit acronym and the system acronym. The `TE` program will automatically add the right suffix for every table type (`pcf` for telemetry parameters, `mccf` for microcommands, `msg` for messages).



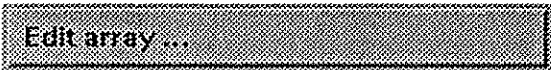
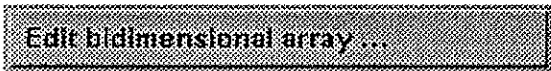
A special case is represented by the microcommands file of the unit `com` for VME systems. Here a special microcommand has to be inserted: the `EXIT` command that will cause the termination of all the processes on the VME. This command has the `vmecode` field set to **-10**.

## 5.0 The design of the work area

Different table types sport different work area layouts. How is this achieved?

The work area is divided into two major parts: item descriptions and item fields. In the one on the farther left the descriptions of items, taken by the `dbms.cfg` file in the `TNGCONFIG` directory [3], are displayed. On the right side, various types of OSF/Motif widgets are used to enter and edit item values. Item of type string, integer, long or float (as described in `dbms.cfg`) have a text edit field, with type and length checks. Items of type boolean have a toggle switch to set or unset their value. Composite items, mono or bidimensional arrays, have a button that controls a dialog box with input widgets (see Table 2, "Widget types," on page 10).

TABLE 2. Widget types

Item types	Edit widget
String, integer, long, float	
Boolean	
Intarr, longarr, floatarr, chararr	
Intdarr, longdarr, chardarr	

When editing a monodimensional array, a dialog box is displayed with a list of the items and the relative editing fields (either a Text or a ToggleButton widget) and a Close button. The user can enter values on the right side, and then a click on the Close button will close the dialog box.

When editing a bidimensional array, a dialog box is displayed with a list of the second level items and the relative activation buttons. Selecting one of these buttons will cause another dialog box to be displayed with a list of description and editing fields (either a Text or a ToggleButton widget) and a Close button. The user may enter values on the right side, and then a click on the Close button will close the dialog box.

## 6.0 Acknowledgements

We would like to acknowledge the contribution in the form of several valuable discussions of the TNG group at the Padova Observatory during both the design and the development of the Table Editor. Special thanks also go to the colleagues of the Astrophysical Technologies Group of the Trieste Observatory for the support given to this work.

## 7.0 References

- [1] P. Marcucci, M. Pucillo  
*The Galileo Interactive Panel Editor*  
Publ. Osservatorio Astronomico di Trieste n. 1451
- [2] A. Balestra, P. Marcucci, F. Pasian, M. Pucillo, R. Smareglia, C. Vuerli  
*Galileo Project. Workstation Software System*  
TNG Technical Report n. 9
- [3] P. Marcucci  
*DBlib - Disk i/o handling library*  
Publ. Osservatorio Astronomico di Trieste n. 1440