The Galileo Help System v1.0

Paolo Marcucci
Osservatorio Astronomico di Trieste
November 1992 - Pubblicazione Osservatorio Astronomico di Trieste n. 1456

Abstract

The Galileo Help System (TNG HELP) is a hypertext-based, context-sensitive help tool, designed to give the user easy and quick access to all the information required to operate the Galileo Telescope. The description of the functionalities and programming of the Galileo Help System is given in this document.

1.0 Introduction

In modern, state of art, software systems a powerful help system is no longer an add- on feature but a real must. These softwares are so complex that even skilled users tend to forget the syntax or use of functions and commands. Printed manuals are not always at hand, thus the only source of information lies in an on-line help system.

The Galileo Help System is basically a library that an application programmer can link to his/her main program in order to simplify the management of help data.

2.0 Help data files

Help data files are ASCII files that contain either help texts and links between help topics. An help topic is a self-consistent "chunk" of information, of usually no more than a few dozen lines, that may have a series of links to other topics. These links can be represented as branches of an information tree, which can be navigated in both directions. So if a link in a topic is selected, it must be possible to "backwalk" the link and return to the previous topic. It sounds obvious, but it is not.

In a help data file, consisting of pure ASCII characters, topics and links are enclosed in a pair of "\" characters. E.g. `\maintopic\This is the main topic\`. The first word is the topic internal name, while the ensuing sentence is a topic description. Such a line will not be displayed in the help text, but it will be parsed and stored in a so-called "SubTopic list" where the user can click over it and obtain the help related text.

The structure of a help data file is provided here:

```
..main
This is the main
help screen.
\first\First topic\
\second\Second topic\
..end(main)
..first
This is a description
of the first topic, with
a link to the second.
\second\A link to another topic\
..end(first)
..second
This is a description
of the second topic, with
no links.
..end(second)
```

Topics are defined by a starting and an ending line, both with a double dot at the beginning. The starting line contains, after the double dot, the topic internal name, either uppercase or lowercase. The ending line contains the word "end" and the internal name of the topic is enclosed in a parenthesis.

Links, as described beforehand, are enclosed in two pairs of "\" characters, with the internal name and a description. Note that, while the internal name always has to be the same in all the links, the description may vary depending on the context. Selecting either "Second topic" in the main screen or "A link to another topic" in the first screen takes the user to the second screen.

The text width must not exceed 75 characters, while its length is, in theory, unlim- ited. It would be advisable, however, to write short texts on many topics with several links, thus facilitating the user's life (whilst further complicating the programmer's...).

3.0 The Help Tree

As seen, help information can be represented as a number of branches in a "help tree". The root of this tree is the main topic, that is the only predefined name in the tree. The first links are listed in this special topic, and a brief description, or a welcome screen, may be displayed. Usually the first topics will be "How to get more help", "Topic index" and others of this level of description.

```
                       main
                         |
               +---------+--------+
               |                  |
             index              howto
               |
      +--------+---------+
      |        |         |
    first    second    third
      |                  |
  +---+--+               |
  |      |               |
fourth fifth           sixth
         |
      seventh
```

FIGURE l. Help tree with internal names listed

4.0 The Help window

On calling the library function Help, a window will pop up on the workstation screen. The position of this window is centered around the middle of the application main win- dow, but it may be moved to a new location. When the window is closed, the last position is stored in memory so that another call to the Help function will display the window reflecting the last used position.

The Help window comprises three areas: the menu bar, the help text and the sub-topic list. The menu bar contains two buttons needed to navigate backwards in the help tree: the Index button takes the user to the first screen of the help file (the main topic), the Back button takes the user one level back in the help tree. For example, if the user is consulting the seventh screen (see Figure 1), by pressing the Back button the user will pass to the fifth screen, by pressing it again he will move to the first screen, and so on. The current maximum number of backward levels available is 20.

In the help text area, the text of the topic selected is displayed. The user can scroll the text using the right scrollbar. In the subtopic list area, the descriptions of the related links are listed with an optional scrollbar. A single click over an item in the list takes the user to the related screen.

To close the help window, double-click over the Motif control box (in the upper left corner) or press Meta-F4

5.0 Using the Help System from a program

To use the Galileo Help System from a program, the programmer has only to link his/her main program to the Help. o library, or in the case of an application developed in the Galileo operating environment, to the tnglib. a library

The Help library provides the application programmer with three functions. They are as follows:

5.1 CreateHelp

The CreateHelp function builds the Motif set of widgets that defines the overall appearance and functionality of the Help window (menu bar, subtopic list, etc.) and defines the name of the help data file (see "Help data files"). There are two function parameters: the first, of the widget type, specifies the parent widget to which the help window belongs, the second, of char[] type, is the name of the help data file.

A typical call is:

```
    CreateHelp (mainwindow, "main. hlp");
```

where mainwindow is the main window widget identifier. Note that the function does not return any error status. If the help data file name is incorrect, it will be reported in the first call to the Help function

5.2 Help

The Help function actually displays a help screen. It has only one parameter, of char[] type, that defines the help topic to be displayed. The help topic passed as a parameter is the internal name of the topic and not its description. The call takes this form:

```
    Help ("main");
```

this call shows the main help screen, bearing in mind that this is the only predefined topic and is equivalent to pressing the Index button in the menu bar.

5.3 ChangeHelp

This function changes the name of the help data file without rebuilding the help window widget. The call is:

```
    ChangeHelp ("another.hlp");
```

This function may be called anywhere for an unlimited number of times in the application program after a call to CreateHelp is executed.

Appendix A - Programming interface definition

Here the complete definition of the Help library public function is described:

```
void CreateHelp (w, f)
  Widget w;
  char f[];

void Help (s)
  char s[];

void ChangeHelp (s)
  char s[];
```

It is important to note that the application program must be already based on Motif, to take full advantage of the Help library. Character oriented programs cannot benefit from the function described in this manual.

Appendix B - Example program

```
/***************************************************************
*                                                              *
* NAME: Get Help                                               *
*                                                              *
* AUTHOR: P. Marcucci - Osservatorio Astronomico di Trieste    *
* TYPE: program                                                *
* FILE: gethelp.c                                              *
* LANGUAGE: C                                                  *
*                                                              *
* FUNCTIONAL DESCRIPTION:                                      *
* Example program for the Help library.                        *
*                                                              *
* REQUIREMENTS:                                                *
*                                                              *
* HISTORY                                                      *
* 920910 PM-OAT Creation                                       *
*                                                              *
***************************************************************/

#include
#include
#include
#include

#include
#include
#include
#include
```

```
/*
** Help callback
*/
void HelpCB ( )
  {
  Help ("main");
  }

/*
** main - main logic for table program
*/

void main (argc,argv)
  unsigned int argc;
  char **argv;

  {
  int i ;
  Display *display;

  Widget app_shell;
  XtAppContext app_context;

  XtToolkitInitialize ();
  app_context = XtCreateApplicationContext();
  display = XtOpenDisplay (app_context,NULL,argv[0],
                           "",NULL,0,&argc,argv);

  i = 0;
  app_shell = XtAppCreateShell (argv[0],"",
                                applicationShellWidgetClass,
                                display,args,i);

  i = 0;
  mainw = XmCreatePushButton(app_shell,"pushbutton",args,i);
  XtManageChild (mainw);
  XtAddCallback(mainw,XmNactivateCallback,HelpCB,NULL);

  CreateHelp (mainw, "test.hlp");

  XtRealizeWidget (app_shell);
  XStoreName (display,Xt Window(app_shell),
              "Help system test");

  XtAppMainLoop(app_context);
  }
```

This example must be compiled with the usual X/Motif command line switches, and must be linked with Help.o (or tnglib.a in the case of the Galileo environment) and the Xwindows, XToolkit and Motif libraries.