Astronomical Data Analysis Software and Systems IX
ASP Conference Series, Vol. 216, 2000
N. Manset, C. Veillet, and D. Crabtree, eds.

A Python-based IRAF Task Parameter Editor

M. D. De La Peña

Space Telescope Science Institute, Baltimore, MD 21218

Abstract. As part of the development of a new Python-based CL for IRAF tasks by the Science Software Group at STScI, we have developed a GUI-based parameter editor for IRAF tasks using Tkinter. This new parameter editor is intended to provide the equivalent functionality of the IRAF EPAR task, but to make parameter editing easier by using appropriate user interface elements, such as menu choice lists, action buttons, and file browsers. This paper describes the design and functionality of the parameter editor as well as planned enhancements.

1. Why a GUI Parameter Editor?

A well-designed GUI uses readily recognizable visual cues to help the user navigate efficiently through information. Additional functionality can be provided in hidden menus or separate views which can be invoked as needed so as to keep the main view streamlined and uncluttered. A GUI is able to make effective use of interfaces as appropriate to the type of values being displayed (e.g., drop-down menu for a choice list, or radio buttons for boolean options) while exploiting widely-used, modern-day technology (e.g., X11 Window System). Fundamentally, a GUI presents a view which is both aesthetic and functional to the user.

2. The Challenge

The challenge in creating a GUI parameter editor for IRAF is that by its very function, the editor becomes a GUI for all tasks in the system. Ideally, a GUI is designed as an integral part of an application, and therefore, it is able to present tailored views to the user with respect to combinations of input options. Through these customized views, the user can access the task very efficiently. Unfortunately, the only information available to the GUI-based parameter editor in the IRAF system are basic characteristics describing the parameters (e.g., data types, default values); there is no information available with respect to the relationship between parameters.

3. The Solution

The Python/Tkinter GUI-based IRAF task parameter editor is being designed to provide not only equivalent functionality to that available in the IRAF EPAR, but also to extend the capabilities of the task. Figure 1 presents an illustration of the prototype Python parameter editor interface. The top of the view lists the package and task names and contains the fundamental action buttons. Information regarding the meaning of an action button is obtained by placing the cursor over the button; the explanatory text appears in the narrow status bar located at the bottom of the application window. A scrolling canvas of parameter attributes (name, value input widget, and prompt string) occupies the majority of the view and is the main interaction region. This GUI-based IRAF task parameter editor is an integrated part of the new Python-based CL for IRAF tasks which is being developed by the Science Software Group at STScI (Greenfield & White 2000).

While the majority of the action buttons are self-explanatory, it should be noted that the WINDOW and HELP buttons work in conjunction with one another. The user has the choice of displaying the IRAF help pages in either a browser or in another window. The browser option invokes the HTML help system developed at STScI. The WINDOW option simply displays the IRAF help pages in another window accompanying the parameter editor. The WINDOW menu button at the top right allows the user to choose the preferred display with the button label reflecting the user's current choice.

4. The Interactive Canvas in Detail

The parameter canvas contains the widgets which allow the user to modify parameter values. Each parameter employs a widget which is most appropriate to the type of data required from the user. Table 1 is a summary of the parameters and corresponding widgets.

Table 1. Correspondence of parameter types and widget representations.

Parameter Type	Widget
Boolean	Radiobutton
Choice list (of any data type)	Menubutton/Menu
String	Entry box
Numeric value	Entry box
PSET	Button/New Edit Window

In order to keep the main GUI simple, additional functionality is invoked through the use of the action buttons or use of the right-most button on a mouse in string entry fields. Pressing the right-most button on a mouse while within a string entry box generates a pop-up menu with the options to: invoke a file browser, clear the entry box, or to "unlearn" the specific parameter value. Clearing the entry box removes the current value in the entry. "Unlearning" the entry restores the initial default value for this specific entry only. Invoking the file browser starts up an independent window which allows the user to examine the directory structure.

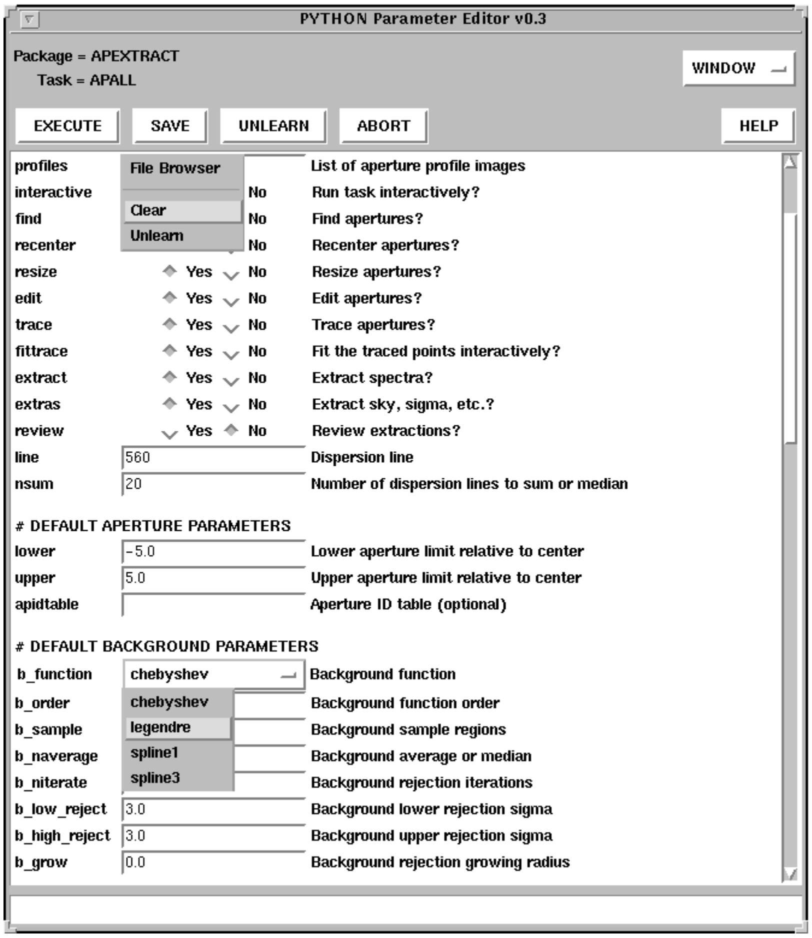

Figure 1. The GUI-based Python/Tkinter IRAF task parameter editor. In this figure the pop-up menu for the "profiles" parameter is deployed. The pop-up menu was invoked by pressing the right-most button on a mouse while in the entry field of the "profiles" parameter. This figure also shows a choice list implemented as a drop-down menu which is deployed for the "b_function" parameter. Once the function is chosen, the corresponding button label will be updated with the function name. The remaining parameters on this canvas require numeric input values in the entry box or are represented as radio buttons for the boolean option.

After numeric data has been entered, contemporaneous entry checking is done by pressing the Return key. Regardless, entry checking is done for numeric data when explicitly saving the parameter values (via the SAVE button) or by executing the task (via the EXECUTE button). Any associated error messages are displayed in the status bar at the bottom of the application window. The PSET is an action button that invokes a new edit window which is displayed in conjunction with the main view. While supplementary visual cues must be implemented to minimize confusion, it is intended for the task to have multiple windows visible and active concurrently.

5. Future Plans

The immediate future plans for the GUI parameter editor are to implement equivalent functionality to that which is currently available in IRAF EPAR. At the least this requires implementation of: the ability to tab from entry to entry via the RETURN key, supplement the menu options by allowing string entry with minimum match completion, and acquiring parameter values from alternate PSETS. However, the GUI medium inherently has more capabilities which will be exploited for the convenience of the users. One of the most useful of these planned capabilities is a customized directory browser which will incorporate an internal file browser. For FITS files, it is particularly useful to be able to examine the FITS headers of particular extensions in order to choose the proper data.

References

Greenfield, P. & White, R. 2000, this volume, 59