

RTK

The ROME ToolKit (RTK) implements an abstraction for a set of screen elements such as buttons, text labels, sliders or windows.

RTK does not implement its own process for handling events. Instead it relies on the application to forward events to its event handler.

Requirements

RMP	RTK derives its widgets from RMP. It also uses some functionality provided by RMP.
GIA	RTK uses GIA for graphic operations. RTK does not talk directly with a graphics driver.

Module Options

RTK_DEBUG	Enables debug output for RTK.
-----------	-------------------------------

Data definitions

RTK contains a set of widgets, each of which defines its own structure. Some widgets are derived from others, extending their functionality and data structures. Widget data structures are defined in the header file of each widget. The following widget types are defined in RTK:

`rtk_button_t`, `rtk_canvas_t`, `rtk_drawarea_t`, `rtk_label_t`, `rtk_label_button_t`, `rtk_label_pixmap_button_t`, `rtk_pixbutton_t`, `rtk_pixmap_t`, `rtk_pixmap_button_t`, `rtk_pixtogglebutton_t`, `rtk_slider_t`, `rtk_text_input_t`, `rtk_togglebutton_t`, `rtk_widget_t`, `rtk_window_t`

RTK “Process” Operation

RTK implements a single event handler which has to be called by the application using RTK once an event is received. RTK basically handles GIA events such as mouse button clicks or keyboard events.

Shared Library Macros and Routines

`rtk_init`

```
int rtk_init(
```

void)

The *rtk_init* function should be called before any other RTK functions. This function initializes RTK internal structures and prepares RTK for operation. *rtk_init* returns 0 on success, -1 otherwise.

rtk_handler

int *rtk_handler*(
 ROME_MESSAGE *ev)

The *rtk_handler* routine should be called by the application using RTK. It handles events coming from GIA such as mouse clicks and key strokes. *rtk_handler* returns *RMP_HANDLED* if the event could be handled within RTK, *RMP_NOT_HANDLED* otherwise.