R3 GUI Reactors

Author: Carl Sassenrath, RMA

Revision History
6-May-2011/15:48:36+2:00 A

Table of Contents

1. Concept

Events on GUI faces trigger `reactors`.

When a user pushes a button, something happens. The GUI reacts to user input. The result may be a common action such as opening or closing a window, or it may be a very specific action. A reactor is just a type of `shortcut` for a common action. They make the GUI easier to use by eliminating the need to write detailed code.

Reactors are attached to faces. They are usually invoked by the actors of the face style (but not exclusively, there are examples of face reactors being invoked by an actor of an attached face; reactors can be invoked by some user code as well). A reactor can include a single argument value. If more than one argument is needed, they must be enclosed in a block.

Multiple reactors can be attached to the same face. Each reactor is processed in the order that it appears, so the consequences of a reactor's actions must be considered; it can impact the reactors that follow.

2. Using Reactors

Reactors are specified in the GUI dialect that is used to create a layout.

For example, here's a layout that opens a window and displays three buttons, each of which that specifies a reactor:

```
view [
        button "REBOL" browse http://www.rebol.com
        button "Alert" alert "This is an alert."
        button "Close" close
]
```

The obvious actions occur when each button is pressed. The `browse` reactor opens a web browser, the `alert` opens a pop-up alert layout, and `close` will close the current window.

3. Standard Reactors

These reactor functions are provided as a standard part of the GUI:

Reactor	Description
alert	open an alert message box window
attach	set a target value to our value
back (N/A)	return to prior layout (on layout stack)
browse	open web browser to given URL
call (N/A)	run a shell script on local OS
clear	clear face value (empty it)
close	close the window
debug	trace face operation
do	evaluate a REBOL block, file, or URL
download (N/A)	opens download progress layout, starts download
dump	probe the contents of the related face object
edit (N/A)	opens editor for specified content
halt	halt the interpreter
link (N/A)	fetch reb content and evaluate it
launch	run file or URL in a new instance of REBOL
open (N/A)	open a file using local OS methods
quit	exit the application
play (N/A)	plays a sound
print	print output
reset	reset face to original value
scroll	scroll a target face
send (N/A)	sends a message to a target
set	set the state of another face
submit	submit face (or layout) to service or website
undo (N/A)	undo last change
upload (N/A)	opens upload progress layout, starts upload
view (N/A)	view a new "page"

Here are a few examples:

```
button "Alert" alert "This is an alert."
button "Browse" browse http://www.rebol.com
button "Close" close
button "Do" do [alert "It worked!"]
button "Dump" dump
```

```
button "Halt" halt
button "Launch" launch %test.r
button "Print" print "print to console"
button "Quit" quit
button "Run" run %explorer.exe
button "View" view [button "Close" close]

f1: field "Field 1"
button "Focus on 1" focus 'f1
```

4. Multiple Reactors

A face can have more than one reactor. Each reactor is processed in the order that it appears. The example below shows how two fields are reset using a single reset button:

```
user: field
pass: field
button "Reset" reset 'user reset 'pass
```

Reactors are processed from left to right, so be sure to consider that.

For example, this line works:

```
button "Submit" submit cgi-url clear 'user
```

But, in this line the field is cleared, so the submitted text is empty:

```
button "Submit" clear 'user submit cgi-url
```

So, that’s not useful.

5. Creating Custom Reactors

.

Note

Feature in this section is not yet implemented.

The `make-reactor` function can be called to create custom reactors for your GUI. One or more new reactors can be defined. The format of a reactor definition is similar to that of a function, except the `function` creator such as `func` is not used.

Here’s an example reactor definition:

```
make-reactor [
        my-browse: ["Open web browser." arg [url! file!]] [browse arg]
]
```

6. Calling Reactors

When implmenting a style actor, when a final result is known, it may be necessary to call one or more face reactors.

The `do-face` function calls reactors. This line calls all reactors:

```
do-face face
```

To call a specific reactor, you can write:

```
do-face/only face 'reactor
```

where `reactor` is its name.

7. Examples

Here’s an example layout that uses a few different reactors:

```
view [
        hpanel 2 [
                button "Do" do [request "Got it!" "It worked."]
                button "Browse" browse http://www.rebol.com
                button "Run" run %explorer
                button "Alert" alert "This is an alert."
        ]
        hpanel 2 [
                f1: field "Field 1"
                f2: field "Field 2"
                button "Focus on 1" focus 'f1
                button "Focus on 2" focus 'f2
        ]
        hpanel 2 [
                button "Close" close
                button "Halt" halt
                button "Quit" quit
                button "Print" print "print this message"
                button "Dump" dump
        ]
]
```