

NAME

pipeglade -- pipe-driven GTK+ interface

SYNOPSIS

pipeglade [[**-i** *in-fifo*] [**-o** *out-fifo*] [**-b**] [**-u** *builder-file*] [**-e** *parent-xid*] [**-l** *log-file*] [**-O** *err-file*]
 [--display *X-server*]] | [**-h** | **-G** | **-V**]

DESCRIPTION

pipeglade is a helper program that displays graphical user interfaces for other programs. It renders the GUI definition found in a GtkBuilder XML file (typically created using the glade(1) interface designer), and communicates with the main program solely through plain text messages via pipes or fifos. It provides access to a subset of features of a subset of widgets of GTK+ 3. Simple one-shot dialogs as well as more complex, long-running programs can be built using **pipeglade**; see *EXAMPLES*.

OPTIONS**-i** *in-fifo*

Creates a named pipe *in-fifo* if necessary, sets its permissions to 0600, and uses it for command input.

A command is a line of text. Its format is "*id:action [data]*", *id* being the name of the receiving widget; *data* is separated from the rest of the command by a single whitespace character. Commands whose first non-whitespace character is **#** are considered comments and ignored silently, as are empty input lines. Any occurrences of the two-character sequences **\n** and **\r** will be converted into newline or carriage return, respectively. Every other character following a backslash will be left unchanged, but the backslash will be removed. Invalid commands are reported on *stderr* and are otherwise ignored. See *WIDGETS* for applicable commands.

It is an error if *in-fifo* exists but is not a named pipe. The named pipe *in-fifo* is deleted upon successful program termination.

Default command input is *stdin*.

-o *out-fifo*

Creates a named pipe *out-fifo* if necessary, sets its permissions to 0600, and uses it for output of feedback messages from the graphical user interface.

A feedback message is a line of text; its format is "*id:info [data]*", *id* being the name of the sending widget. See *WIDGETS* for possible feedback messages.

It is an error if *out-fifo* exists but is not a named pipe. The named pipe *out-fifo* is deleted upon successful program termination.

Default feedback-message output is *stdout*.

-b Runs **pipeglade** in the background after printing its process ID to *stdout*.

It is an error if option **-b** is used and not both **-i** *in-fifo* and **-o** *out-fifo* are specified.

-u *builder-file*

Displays the graphical user interface *builder-file* which should be created using the glade(1) user interface designer and saved in GtkBuilder (.ui) format. Widget ids should consist of alphanumeric ASCII characters including hyphens and underscores. The main window must be given the id **main**.

Default is *./pipeglade.ui*.

-e *parent-xid*

Embeds the main window into the XEmbed socket *parent-xid* of another process.

-l *log-file*

Appends for each command a line of "*time activity*" to *log-file*, *time* being the number of microseconds elapsed during *activity*. *time* and *activity* are separated by a tab character. Permissions of *log-file* are set to 0600. A *log-file* name of '-' means *stderr*.

-O *err-file*

Redirects *stderr*, appending its output to *err-file*. Permissions of *err-file* are set to 0600.

--display *X-server*

Uses the display on *X-server* for user interaction.

-h Prints a help message and exits.

-G Prints the versions of the underlying GTK+ and cairo libraries and exits.

-V Prints the **pipeglade** version and exits.

WIDGETS

The widget classes able to communicate through **pipeglade** are *GtkBox*, *GtkButton*, *GtkButtonBox*, *GtkCalendar*, *GtkCheckButton*, *GtkColorButton*, *GtkComboBoxText*, *GtkDialog*, *GtkDrawingArea*, *GtkEntry*, *GtkEventBox*, *GtkExpander*, *GtkFileChooserButton*, *GtkFileChooserDialog*, *GtkFontButton*, *GtkFrame*, *GtkGrid*, *GtkImage*, *GtkImageMenuItem*, *GtkInfoBar*, *GtkLabel*, *GtkMenuItem*, *GtkNotebook*, *GtkOverlay*, *GtkPlug*, *GtkPrintUnixDialog*, *GtkProgressBar*, *GtkRadioButton*, *GtkScale*, *GtkScrolledWindow*, *GtkSeparator*, *GtkSeparatorMenuItem*, *GtkSocket*, *GtkSpinButton*, *GtkSpinner*, *GtkStatusbar*, *GtkSwitch*, *GtkTextView*, *GtkToggleButton*, *GtkTreeView*, *GtkTreeViewColumn*, *GtkViewport*, and *GtkWindow*.

pipeglade references a widget by the *id* attribute it is given in Glade. In the *.ui* file, a **GtkWindow** named *foo* would appear as

```
<object class="GtkWindow" id="foo">.
```

Unspecific Widget

Commands

"*id:load file*" reads arbitrary **pipeglade** commands from *file*. A non-empty *id* is required but ignored. **:load** commands may be nested but on attempts to read from the same file, the inner **:load** is ignored. There is also a **:save** command; see *GtkTextView* and *GtkTreeView*.

"*id:main_quit*" kills the user interface. A non-empty *id* is required but ignored.

GtkBox

Commands "*id:set_sensitive {0|1}*" makes the box grayed out (**0**) or responsive (**1**).

"*id:set_visible {0|1}*" hides (**0**) the box, or makes it visible (**1**). "*id:set_tooltip_text [text]*" sets the box's tooltip. Default is disabling the tooltip.

"*id:snapshot file { .ps|.eps|.epsf|.pdf|.svg }*" writes a snapshot of the box to an image file of the specified type.

"*id:set_size_request [width height]*" sets the box's minimum size to *width* x *height*. Default is the box's natural size.

"*id:style [prop:val[; prop:val ...]]*" sets the style. See *Style*.

Feedback none

GtkButton

Commands "*id:set_label string*" replaces the button text with *string*.

"*id:grab_focus*" puts the keyboard focus onto the button.

"*id:set_sensitive {0|1}*" makes the button grayed out (**0**) or responsive (**1**).

"*id:set_visible {0|1}*" hides (**0**) the widget, or makes it visible (**1**).

"*id:block {0|1}*" blocks (**1**) or unblocks (**0**) feedback messages. Initially, the button is unblocked.

"*id:set_tooltip_text [text]*" sets the button's tooltip. Default is disabling the tooltip.

"*id:force*" simulates a click on the button.

"*id:snapshot file*{*.ps|.eps|.epsf|.pdf|.svg*}" writes a snapshot of the button to an image file of the specified type.

"*id:set_size_request* [*width height*]" sets the button's minimum size to *width* x *height*. Default is the button's natural size.

"*id:style* [*prop:val*; *prop:val ...*]" sets the style. See *Style*.

Feedback "*id:clicked*"

GtkButtons with ids ending in **_ok**, **_apply**, **_cancel**, **_send_text**, and **_send_selection** may work differently; see *GtkDialog*, *GtkFileChooserDialog*, and *GtkTextView* for details.

GtkButtonBox

Commands "*id:set_sensitive* {**0**|**1**}" makes the box grayed out (**0**) or responsive (**1**).

"*id:set_visible* {**0**|**1**}" hides (**0**) the box, or makes it visible (**1**).

"*id:set_tooltip_text* [*text*]" sets the box's tooltip. Default is disabling the tooltip.

"*id:snapshot file*{*.ps|.eps|.epsf|.pdf|.svg*}" writes a snapshot of the box to an image file of the specified type.

"*id:set_size_request* [*width height*]" sets the box's minimum size to *width* x *height*. Default is the box's natural size.

"*id:style* [*prop:val*; *prop:val ...*]" sets the style. See *Style*.

Feedback none

GtkCalendar

Commands "*id:select_date* *yyyy-mm-dd*" selects the date on the calendar.

"*id:mark_day* *day*" marks *day* (1-31) on the calendar.

"*id:clear_marks*" unmarks all days on the calendar.

"*id:set_sensitive* {**0**|**1**}" makes the calendar grayed out (**0**) or responsive (**1**).

"*id:set_visible* {**0**|**1**}" hides (**0**) the calendar, or makes it visible (**1**).

"**id:block** {0|1}" blocks (1) or unblocks (0) feedback messages. Initially, the calendar is unblocked.

"**id:set_tooltip_text** [*text*]" sets the calendar's tooltip. Default is disabling the tooltip.

"**id:force**" simulates a click on the currently selected date.

"**id:snapshot file**{.ps|.eps|.epsf|.pdf|.svg}" writes a snapshot of the calendar to an image file of the specified type.

"**id:set_size_request** [*width height*]" sets the calendar's minimum size to *width* x *height*. Default is the widget's natural size.

"**id:style** [*prop:val*; *prop:val ...*]" sets the calendar's style. See *Style*.

Feedback "**id:clicked** yyyy-mm-dd"

"**id:doubleclicked** yyyy-mm-dd"

GtkCheckButton

Commands "**id:set_active** {0|1}" switches the check mark off (0) or on (1).

"**id:set_label** *string*" replaces the button text with *string*.

"**id:grab_focus**" puts the keyboard focus onto the button.

"**id:set_sensitive** {0|1}" makes the button grayed out (0) or responsive (1).

"**id:set_visible** {0|1}" hides (0) the button, or makes it visible (1).

"**id:block** {0|1}" blocks (1) or unblocks (0) feedback messages. Initially, the button is unblocked.

"**id:set_tooltip_text** [*text*]" sets the button's tooltip. Default is disabling the tooltip.

"**id:force**" simulates a click on the button.

"**id:snapshot file**{.ps|.eps|.epsf|.pdf|.svg}" writes a snapshot of the button to an image file of the specified type.

"**id:set_size_request** [*width height*]" sets the button's minimum size to *width* x *height*. Default is the widget's natural size.

"*id:style* [*prop:val*[: *prop:val* ...]]" sets the button's style. See *Style*.

Feedback "*id:1*" if switched on, or "*id:0*" otherwise.

GtkColorButton

Commands "*id:set_color color*" preselects the color; see *Color*.

"*id:grab_focus*" puts the keyboard focus onto the button.

"*id:set_sensitive* {*0*|*1*}" makes the button grayed out (**0**) or responsive (**1**).

"*id:set_visible* {*0*|*1*}" hides (**0**) the button, or makes it visible (**1**).

"*id:block* {*0*|*1*}" blocks (**1**) or unblocks (**0**) feedback messages. Initially, the button is unblocked.

"*id:set_tooltip_text* [*text*]" sets the button's tooltip. Default is disabling the tooltip.

"*id:force*" opens the dialog window.

"*id:snapshot file*{*.ps|.eps|.epsf|.pdf|.svg*}" writes a snapshot of the button to an image file of the specified type.

"*id:set_size_request* [*width height*]" sets the button's minimum size to *width* x *height*. Default is the widget's natural size.

"*id:style* [*prop:val*[: *prop:val* ...]]" sets the button's style. See *Style*.

Feedback "*id:color rgb(red,green,blue)*" or "*id:color rgba(red,green,blue,alpha)*". *red*, *green*, and *blue* lie between 0 and 255, and *alpha* between 0 and 1.

GtkComboBoxText

The **GtkComboBoxText** should contain a **GtkEntry**.

Commands "*id:prepend_text string*" and "*id:append_text string*" prepend/append a new selectable item marked *string*.

"*id:insert_text position string*" inserts item *string* at *position*.

"*id:remove position*" removes the item at *position*.

"*id:grab_focus*" puts the keyboard focus onto the combo box.

"*id:set_sensitive* {0|1}" makes the combo box grayed out (**0**) or responsive (**1**).

"*id:set_visible* {0|1}" hides (**0**) the combo box, or makes it visible (**1**).

"*id:block* {0|1}" blocks (**1**) or unblocks (**0**) feedback messages. Initially, the combo box is unblocked.

"*id:set_tooltip_text* [*text*]" sets the tooltip. Default is disabling the tooltip.

"*id:snapshot file* {*.ps|.eps|.epsf|.pdf|.svg*}" writes a snapshot of the combo box to an image file of the specified type.

"*id:set_size_request* [*width height*]" sets the combo box's minimum size to *width* x *height*. Default is the natural size.

"*id:style* [*prop:val* [*;* *prop:val* ...]]" sets the style. See *Style*.

Feedback "*entry_id:text text*", *entry_id* being the id of the child **GtkEntry**.

GtkDialog

A **GtkDialog** with id *foo* will be invoked by a *GtkMenuItem* or a *GtkImageMenuItem* with id *foo_invoke*.

The **GtkDialog** should have a 'Cancel' *GtkButton* with id *foo_cancel* (**main_cancel** if the dialog is the sole window of the GUI and therefore named **main**).

If the **GtkDialog** has an 'Ok' *GtkButton* with id *foo_ok*, it will take care of hiding the dialog window.

Commands "*id:set_title string*" replaces the text in the title bar with *string*.

"*id:resize* [*width height*]" changes the size of the dialog window to *width* x *height* pixels if specified, or to the default size.

"*id:move x y*" moves the dialog window to position (*x*, *y*).

"*id:fullscreen*" and "*id:unfullscreen*" switch fullscreen mode on and off.

"*id:set_sensitive* {0|1}" makes the dialog grayed out (**0**) or responsive (**1**).

"*id:set_visible* {0|1}" hides (**0**) the dialog, or makes it visible (**1**).

"*id:block* {0|1}" blocks (**1**) or unblocks (**0**) feedback messages from the dialog window. Initially, the window is unblocked.

"**id:set_tooltip_text** [*text*]" sets the dialog window's tooltip. Default is disabling the tooltip.

"**id:snapshot file** {*.ps|.eps|.epsf|.pdf|.svg*}" writes a snapshot of the dialog window to an image file of the specified type.

"**id:set_size_request** [*width height*]" sets the dialog window's minimum size to *width* x *height*. Default is its natural size.

"**id:style** [*prop:val*; *prop:val ...*]" sets the dialog window's style. See *Style*.

Feedback "**id:closed**" if the **GtkDialog** window was closed by the window manager.

GtkDrawingArea

Drawing commands expect a context parameter of the form $\{n|=n|n<m\}$ where *n* is an arbitrary non-negative integer that can be used later to reference the command. The first form, *n*, appends its command to the current list of drawing operations. The second form, *=n*, replaces the first drawing operation with context parameter *n* by the new command. The third form, *n<m*, inserts its command before the first drawing operation with context parameter *m*. Both second and third form fall back to appending.

The default coordinate system is a left-handed system with its origin in the upper-left corner. Angles increase from the positive X axis towards the positive Y axis (clockwise in the default system).

Commands "**id:arc** $\{n|=n|n<m\}$ *x y radius angle1 angle2*" adds a circular arc to the current path. The arc is centered at (*x*, *y*) and proceeds from *angle1* to *angle2* (in degrees).

"**id:arc_negative** $\{n|=n|n<m\}$ *x y radius angle1 angle2*" adds a circular arc to the current path. The arc is centered at (*x*, *y*) and proceeds in the direction of decreasing angles from *angle1* to *angle2* (in degrees).

"**id:close_path** $\{n|=n|n<m\}$ " adds a line segment from the current point to the point most recently passed to **id:move_to** or **id:rel_move_to**.

"**id:curve_to** $\{n|=n|n<m\}$ *x1 y1 x2 y2 x3 y3*" adds a cubic Bezier spline from the current point to (*x3*, *y3*), using (*x1*, *y1*) and (*x2*, *y2*) as control points. If there is no current point, the curve will start at (*x1*, *y1*).

"**id:fill** $\{n|=n|n<m\}$ " fills the current path and clears it.

"**id:fill_preserve** $\{n|=n|n<m\}$ " fills the current path without clearing it.

"**id:line_to** $\{n|=n|n<m\}$ *x y*" adds a line from the current point to (*x*, *y*), or creates a new

current point at (x, y) .

"**id:move_to** { $n|=n|n<m$ } x y " sets the current point to (x, y) .

"**id:rectangle** { $n|=n|n<m$ } x y $width$ $height$ " adds a rectangle to the current path. The top left corner is at (x, y) .

"**id:rel_curve_to** { $n|=n|n<m$ } $dx1$ $dy1$ $dx2$ $dy2$ $dx3$ $dy3$ " adds a cubic Bezier spline from the current point to $(dx3, dy3)$, using $(dx1, dy1)$ and $(dx2, dy2)$ as control points. All coordinates are offsets relative to the current point.

"**id:rel_line_to** { $n|=n|n<m$ } dx dy " adds a line from the current point to a point offset from there by (dx, dy) .

"**id:rel_move_for** { $n|=n|n<m$ } {**c|e|n|ne|nw|s|se|sw|w**} $text$ " moves the current point such that "**id:show_text** n $text$ " will place the specified reference point of $text$ on the original current point.

"**id:rel_move_to** { $n|=n|n<m$ } dx dy " moves the current point by (dx, dy) .

"**id:remove** n " removes the elements with context parameter n from the **GtkDrawingArea** id .

"**id:rotate** { $n|=n|n<m$ } $angle$ " rotates the user space axes by $angle$ (in degrees).

"**id:scale** { $n|=n|n<m$ } sx sy " scales the user space axes by (sx, sy) .

"**id:set_dash** { $n|=n|n<m$ } l " sets the dash pattern to l on, l off.

"**id:set_dash** { $n|=n|n<m$ } $l1on$ $l1off$ $l2on$ $l2off$..." resets the dash pattern to a line with arbitrary on/off portions.

"**id:set_dash** { $n|=n|n<m$ }" resets the dash pattern to a solid line.

"**id:set_font_face** { $n|=n|n<m$ } {**normal** | **italic** | **oblique**} {**normal** | **bold**} [$family$]" sets the font face for subsequent calls of **id:show_text**.

"**id:set_font_size** { $n|=n|n<m$ } $size$ " sets the font size for subsequent calls of **id:show_text**.

"**id:set_line_cap** { $n|=n|n<m$ } {**butt** | **round** | **square**}" sets the line cap style. Default is **butt**.

"**id:set_line_join** {*n*|=*n*|*n*<*m*} {**miter** | **round** | **bevel**}" sets the line junction style. Default is **miter**.

"**id:set_line_width** {*n*|=*n*|*n*<*m*} *width*" sets the line width. Default *width* is 2.

"**id:set_source_rgba** {*n*|=*n*|*n*<*m*} *color*" sets the color; see *Color*.

"**id:show_text** {*n*|=*n*|*n*<*m*} *text*" writes *text*, beginning at the current point.

"**id:stroke** {*n*|=*n*|*n*<*m*}" strokes the current path and clears it.

"**id:stroke_preserve** {*n*|=*n*|*n*<*m*}" strokes the current path without clearing it.

"**id:transform** {*n*|=*n*|*n*<*m*} [*xx* *yx* *xy* *yy* *x0* *y0*]" modifies the current transformation matrix such that

$$\begin{aligned}x' &= xx\ x + xy\ y + x0, \\y' &= yx\ y + yy\ y + y0.\end{aligned}$$

Default is resetting the current transformation matrix.

"**id:translate** {*n*|=*n*|*n*<*m*} *tx* *ty*" translates the user space origin by (*tx*, *ty*).

"**id:set_visible** {**0**|**1**}" hides (**0**) the drawing area, or makes it visible (**1**).

"**id:set_tooltip_text** [*text*]" sets the drawing area's tooltip. Default is disabling the tooltip.

"**id:snapshot** *file*{**.ps**|.eps|.epsf|.pdf|.svg}" writes a snapshot of the drawing area to an image file of the specified type.

"**id:set_size_request** [*width* *height*]" sets the drawing areas's minimum size to *width* x *height*. Default is the natural size.

Feedback none

GtkEntry

Commands "**id:set_text** *string*" replaces the user-editable text with *string*.

"**id:set_placeholder_text** *string*" sets the *string* that is displayed when the entry is empty and unfocused.

"**id:grab_focus**" puts the keyboard focus into the entry.

"*id:set_sensitive* {0|1}" makes the entry grayed out (**0**) or responsive (**1**).

"*id:set_visible* {0|1}" hides (**0**) the entry, or makes it visible (**1**).

"*id:block* {0|1}" blocks (**1**) or unblocks (**0**) feedback messages. Initially, the entry is unblocked.

"*id:set_tooltip_text* [*text*]" sets the entry's tooltip. Default is disabling the tooltip.

"*id:force*" lets the entry report its current content.

"*id:snapshot file*{.ps|.eps|.epsf|.pdf|.svg}" writes a snapshot of the entry to an image file of the specified type.

"*id:set_size_request* [*width height*]" sets the entry's minimum size to *width* x *height*. Default is the entry's natural size.

"*id:style* [*prop:val*[: *prop:val* ...]]" sets the style. See *Style*.

Feedback "*id:text text*", once for each change of *text*.

GtkEventBox

x, *y* are mouse pointer coordinates relative to the **GtkEventBox**.

Commands "*id:grab_focus*" puts the keyboard focus onto the event box.

"*id:set_sensitive* {0|1}" makes the event box insensitive (**0**) or responsive (**1**).

"*id:block* {0|1}" blocks (**1**) or unblocks (**0**) feedback messages. Initially, the event box is unblocked.

"*id:set_tooltip_text* [*text*]" sets the tooltip of the event box. Default is disabling the tooltip.

"*id:snapshot file*{.ps|.eps|.epsf|.pdf|.svg}" writes a snapshot of the event box to an image file of the specified type.

"*id:set_size_request* [*width height*]" sets the minimum size to *width* x *height*. Default is the natural size of the event box.

"*id:style* [*prop:val*[: *prop:val* ...]]" sets the widget style. See *Style*.

Feedback "*id:button_press b x y*", "*id:button_release b x y*" where *b* is the mouse button (normally

1, 2, or 3 for the left, middle, and right button; others may exist).

"*id:motion* *x y*" is reported repeatedly while the mouse is being moved with a button pressed.

"*id:key_press* *key*", *key* being the key's name (e.g. 'Control_L', 'Tab', 'a').

GtkExpander

Commands "*id:set_label* *string*" replaces the expander label text with *string*.

"*id:set_expanded* {*0|1*}" hides (**0**) the child widget, or makes it visible (**1**).

"*id:grab_focus*" puts the keyboard focus onto the expander.

"*id:set_sensitive* {*0|1*}" makes the expander grayed out (**0**) or responsive (**1**).

"*id:set_visible* {*0|1*}" hides (**0**) the expander, or makes it visible (**1**).

"*id:set_tooltip_text* [*text*]" sets the expander's tooltip. Default is disabling the tooltip.

"*id:force*" toggles the expansion state.

"*id:snapshot* *file*{*.ps|.eps|.epsf|.pdf|.svg*}" writes a snapshot of the expander to an image file of the specified type.

"*id:set_size_request* [*width height*]" sets the expander's minimum size to *width* x *height*. Default is the natural size.

"*id:style* [*prop:val*; *prop:val* ...]" sets the style. See *Style*.

Feedback none

GtkGrid

Commands "*id:set_visible* {*0|1*}" hides (**0**) the grid, or makes it visible (**1**). "*id:set_tooltip_text* [*text*]" sets the grid's tooltip. Default is disabling the tooltip.

"*id:snapshot* *file*{*.ps|.eps|.epsf|.pdf|.svg*}" writes a snapshot of the grid to an image file of the specified type.

"*id:set_size_request* [*width height*]" sets the grid's minimum size to *width* x *height*. Default is the grid's natural size.

"*id:style* [*prop:val*; *prop:val* ...]" sets the style. See *Style*.

Feedback none

GtkFileChooserButton

Commands "*id:set_filename path*" preselects *path* to the extent it exists.

"*id:grab_focus*" puts the keyboard focus onto the button.

"*id:set_sensitive {0|1}*" makes the button grayed out (**0**) or responsive (**1**).

"*id:set_visible {0|1}*" hides (**0**) the button, or makes it visible (**1**).

"*id:block {0|1}*" blocks (**1**) or unblocks (**0**) feedback messages. Initially, the button is unblocked.

"*id:set_tooltip_text [text]*" sets the button's tooltip. Default is disabling the tooltip.

"*id:force*" simulates fresh selection of the currently selected pathname.

"*id:snapshot file { .ps|.eps|.epsf|.pdf|.svg }*" writes a snapshot of the button to an image file of the specified type.

"*id:set_size_request [width height]*" sets the button's minimum size to *width* x *height*. Default is the widget's natural size.

"*id:style [prop:val[; prop:val ...]]*" sets the style. See *Style*.

Feedback "*id:file pathname*" if the selection has changed.

GtkFileChooserDialog

A **GtkFileChooserDialog** with id *foo* will be invoked by a *GtkMenuItem* or a *GtkImageMenuItem* with id *foo_invoke*.

The **GtkFileChooserDialog** should have an 'OK' *GtkButton* with id *foo_ok* (**main_ok** if the dialog is the sole window of the GUI and therefore named **main**).

The **GtkFileChooserDialog** may have a 'Cancel' *GtkButton* with id *foo_cancel* (**main_cancel** if the dialog is the sole window of the GUI and therefore named **main**).

The **GtkFileChooserDialog** may have an 'Apply' *GtkButton* with id *foo_apply* (**main_apply** if the dialog is the sole window of the GUI and therefore named **main**).

Commands "*id:set_filename path*" preselects *path* to the extent it exists.

"**id:set_current_name** *string*" makes *string* the suggested filename, which may not yet exist. *string* should either resemble an absolute path, or the *directory* must be set separately by **id:set_filename** *directory*.

"**id:set_title** *string*" replaces the text in the title bar with *string*.

"**id:resize** [*width height*]" changes the size of the dialog window to *width* x *height* pixels if specified, or to the default size.

"**id:move** *x y*" moves the dialog window to position (*x*, *y*).

"**id:fullscreen**" and "**id:unfullscreen**" switch fullscreen mode on and off.

"**id:set_sensitive** {**0**|**1**}" makes the dialog grayed out (**0**) or responsive (**1**).

"**id:set_visible** {**0**|**1**}" hides (**0**) the dialog, or makes it visible (**1**).

"**id:block** {**0**|**1**}" blocks (**1**) or unblocks (**0**) feedback messages from the dialog window. Initially, the widget is unblocked.

"**id:set_tooltip_text** [*text*]" sets the dialog window's tooltip. Default is disabling the tooltip.

"**id:snapshot** *file*{**.ps**|**.eps**|**.epsf**|**.pdf**|**.svg**}" writes a snapshot of the widget to an image file of the specified type.

"**id:set_size_request** [*width height*]" sets the widget's minimum size to *width* x *height*. Default is the widget's natural size.

"**id:style** [*prop:val*; *prop:val ...*]" sets the widget style. See *Style*.

Feedback "**id:file** *pathname*" and/or "**id:folder** *pathname*"

"**id:closed**" if the **GtkFileChooserDialog** window was closed by the window manager.

GtkFontButton

Commands "**id:set_font_name** *fontname*" preselects the font.

"**id:grab_focus**" puts the keyboard focus onto the button.

"**id:set_sensitive** {**0**|**1**}" makes the button grayed out (**0**) or responsive (**1**).

"**id:set_visible** {**0**|**1**}" hides (**0**) the button, or makes it visible (**1**).

"**id:block** {0|1}" blocks (1) or unblocks (0) feedback messages. Initially, the button is unblocked.

"**id:set_tooltip_text** [*text*]" sets the button's tooltip. Default is disabling the tooltip.

"**id:force**" opens the dialog window.

"**id:snapshot file**{.ps|.eps|.epsf|.pdf|.svg}" writes a snapshot of the button to an image file of the specified type.

"**id:set_size_request** [*width height*]" sets the button's minimum size to *width* x *height*. Default is the button's natural size.

"**id:style** [*prop:val*; *prop:val ...*]" sets the style. See *Style*.

Feedback "id:font *fontname*"

GtkFrame

Commands "**id:set_label** *text*" replaces the frame label text with *string*.

"**id:set_sensitive** {0|1}" makes the frame and its content grayed out (0) or responsive (1).

"**id:set_visible** {0|1}" hides (0) the frame, or makes it visible (1).

"**id:set_tooltip_text** [*text*]" sets the frame's tooltip. Default is disabling the tooltip.

"**id:snapshot file**{.ps|.eps|.epsf|.pdf|.svg}" writes a snapshot of the frame to an image file of the specified type.

"**id:set_size_request** [*width height*]" sets the widget's minimum size to *width* x *height*. Default is the frame's natural size.

"**id:style** [*prop:val*; *prop:val ...*]" sets the style. See *Style*.

Feedback none

GtkImage

Commands "**id:set_from_icon_name** *icon-name*" replaces the image with one of the standard icons.

"**id:set_from_file** *path*" replaces the image by the one found at *path*.

"**id:set_visible** {0|1}" hides (0) the image, or makes it visible (1).

"*id:set_tooltip_text* [*text*]" sets the image's tooltip. Default is disabling the tooltip.

"*id:snapshot_file* {*.ps|.eps|.epsf|.pdf|.svg*}" writes a snapshot of the image to an image file of the specified type.

"*id:set_size_request* [*width height*]" sets the images's minimum size to *width* x *height*. Default is the image's natural size.

"*id:style* [*prop:val* [*; prop:val ...*]]" sets the style. See *Style*.

Feedback none

GtkImageMenuItem

GtkImageMenuItem is deprecated.

Commands "*id:set_sensitive* {**0**|**1**}" makes the item grayed out (**0**) or responsive (**1**).

"*id:set_visible* {**0**|**1**}" hides (**0**) the item, or makes it visible (**1**).

"*id:block* {**0**|**1**}" blocks (**1**) or unblocks (**0**) feedback messages. Initially, the item is unblocked.

"*id:set_tooltip_text* [*text*]" sets the item's tooltip. Default is disabling the tooltip.

"*id:force*" simulates a click on the menu item.

"*id:set_size_request* [*width height*]" sets the item's minimum size to *width* x *height*. Default is the item's natural size.

"*id:style* [*prop:val* [*; prop:val ...*]]" sets the item's style. See *Style*.

Feedback A **GtkImageMenuItem** with id *foo_invoke* will invoke the *GtkDialog* or *GtkFileChooserDialog* with id *foo* if it exists. If there isn't any dialog attached to the **GtkImageMenuItem**, it reports "*id:active label*".

GtkInfoBar

Commands "*id:set_sensitive* {**0**|**1**}" makes the info bar grayed out (**0**) or responsive (**1**).

"*id:set_visible* {**0**|**1**}" hides (**0**) the info bar, or makes it visible (**1**).

"*id:set_tooltip_text* [*text*]" sets the info bar's tooltip. Default is disabling the tooltip.

"*id:set_size_request* [*width height*]" sets the info bar's minimum size to *width* x *height*.

Default is the info bar's natural size.

"*id:style* [*prop:val*; *prop:val ...*]" sets the style. See *Style*.

Feedback none

GtkLabel

Commands "*id:set_text string*" replaces the label text with *string*.

"*id:set_sensitive* {**0**|**1**}" makes the label grayed out (**0**) or clearly visible (**1**).

"*id:set_visible* {**0**|**1**}" hides (**0**) the label, or makes it visible (**1**).

"*id:set_tooltip_text* [*text*]" sets the label's tooltip. Default is disabling the tooltip.

"*id:snapshot file*{*.ps|.eps|.epsf|.pdf|.svg*}" writes a snapshot of the label to an image file of the specified type.

"*id:set_size_request* [*width height*]" sets the label's minimum size to *width* x *height*. Default is the natural size.

"*id:style* [*prop:val*; *prop:val ...*]" sets the label's style. See *Style*.

Feedback none

GtkMenuItem

Commands "*id:set_sensitive* {**0**|**1**}" makes the item grayed out (**0**) or responsive (**1**).

"*id:set_visible* {**0**|**1**}" hides (**0**) the item, or makes it visible (**1**).

"*id:block* {**0**|**1**}" blocks (**1**) or unblocks (**0**) feedback messages. Initially, the item is unblocked.

"*id:set_tooltip_text* [*text*]" sets the item's tooltip. Default is disabling the tooltip.

"*id:force*" simulates a click on the menu item.

"*id:set_size_request* [*width height*]" sets the item's minimum size to *width* x *height*. Default is the item's natural size.

"*id:style* [*prop:val*; *prop:val ...*]" sets the item's style. See *Style*.

Feedback A **GtkMenuItem** with id *foo_invoke* will invoke the *GtkDialog* or *GtkFileChooserDialog*

with *id foo* if it exists. If there isn't any dialog attached to the **GtkMenuItem**, it reports "*id:active label*".

GtkNotebook

Commands "*id:set_current_page n*" switches to zero-based page number *n*.

"*id:grab_focus*" puts the keyboard focus onto the notebook.

"*id:set_sensitive {0|1}*" makes the notebook grayed out (**0**) or responsive (**1**).

"*id:set_visible {0|1}*" hides (**0**) the notebook, or makes it visible (**1**).

"*id:set_tooltip_text [text]*" sets the notebook's tooltip. Default is disabling the tooltip.

"*id:snapshot file{.ps|.eps|.epsf|.pdf|.svg}*" writes a snapshot of the notebook to an image file of the specified type.

"*id:set_size_request [width height]*" sets the notebooks's minimum size to *width* x *height*. Default is the widget's natural size.

"*id:style [prop:val[; prop:val ...]]*" sets the notebook's style. See *Style*.

Feedback none

GtkOverlay

Commands "*id:set_sensitive {0|1}*" makes the overlay grayed out (**0**) or responsive (**1**).

"*id:set_visible {0|1}*" hides (**0**) the overlay, or makes it visible (**1**).

"*id:set_tooltip_text [text]*" sets the overlay's tooltip. Default is disabling the tooltip.

"*id:snapshot file{.ps|.eps|.epsf|.pdf|.svg}*" writes a snapshot of the overlay to an image file of the specified type.

"*id:set_size_request [width height]*" sets the overlay's minimum size to *width* x *height*. Default is the overlay's natural size.

"*id:style [prop:val[; prop:val ...]]*" sets the style. See *Style*.

Feedback none

GtkPlug

The main window can be turned into a **GtkPlug** and embedded into the XEmbed socket of another

process; see *OPTIONS*, option **-e**.

GtkPrintUnixDialog

Commands "**id:print** *file.ps*" opens the print dialog. Pressing the 'Print' button sends *file.ps* to the printer the user selected in the dialog.

"**id:set_sensitive** {**0|1**}" makes the dialog grayed out (**0**) or responsive (**1**).

"**id:set_visible** {**0|1**}" hides (**0**) the dialog, or makes it visible (**1**).

"**id:block** {**0|1**}" blocks (**1**) or unblocks (**0**) feedback messages. Initially, the dialog is unblocked.

"**id:set_tooltip_text** [*text*]" sets the dialog's tooltip. Default is disabling the tooltip.

"**id:snapshot** *file*{**.ps|.eps|.epsf|.pdf|.svg**}" writes a snapshot of the dialog to an image file of the specified type.

"**id:set_size_request** [*width height*]" sets the dialog's minimum size to *width* x *height*. Default is the dialog's natural size.

"**id:style** [*prop:val*; *prop:val ...*]" sets the widget style. See *Style*.

Feedback "**id:closed**" if the **GtkPrintUnixDialog** window was closed by the window manager.

GtkProgressBar

Commands "**id:set_fraction** *x*" moves the progress bar to *x* (between 0 and 1).

"**id:set_text** [*string*]" replaces the text of the progress bar with *string*. Default is the progress percentage.

"**id:set_visible** {**0|1**}" hides (**0**) the progress bar, or makes it visible (**1**).

"**id:set_tooltip_text** [*text*]" sets the tooltip of the progressbar. Default is disabling the tooltip.

"**id:snapshot** *file*{**.ps|.eps|.epsf|.pdf|.svg**}" writes a snapshot of the progress bar to an image file of the specified type.

"**id:set_size_request** [*width height*]" sets the minimum size of the progress bar to *width* x *height*. Default is the natural size.

"**id:style** [*prop:val*; *prop:val ...*]" sets the style. See *Style*.

Feedback none

GtkRadioButton

Commands "**id:set_active 1**" switches the button on. All other buttons of the same group will go off automatically.

"**id:set_label** *string*" replaces the button text with *string*.

"**id:grab_focus**" puts the keyboard focus onto the button.

"**id:set_sensitive** {0|1}" makes the button grayed out (0) or responsive (1).

"**id:set_visible** {0|1}" hides (0) the button, or makes it visible (1).

"**id:block** {0|1}" blocks (1) or unblocks (0) feedback messages. Initially, the button is unblocked.

"**id:set_tooltip_text** [*text*]" sets the button's tooltip. Default is disabling the tooltip.

"**id:force**" simulates a click on the button.

"**id:snapshot file**{.ps|.eps|.epsf|.pdf|.svg}" writes a snapshot of the button to an image file of the specified type.

"**id:set_size_request** [*width height*]" sets the button's minimum size to *width* x *height*. Default is the button's natural size.

"**id:style** [*prop:val*; *prop:val ...*]" sets the style. See *Style*.

Feedback "**id:1**" if switched on, or "**id:0**" otherwise.

GtkScale

Commands "**id:set_value** *x*" moves the slider to value *x*.

"**id:set_fill_level** [*x*]" moves the fill level indicator to value *x*. Default is hiding the fill level indicator.

"**id:set_range** *min max*" changes the range.

"**id:set_increments** *step page*" sets the steps the slider moves while pressing arrow keys and 'Page_Up' / 'Page_Down', respectively.

"**id:grab_focus**" puts the keyboard focus onto the scale.

"*id:set_sensitive* {0|1}" makes the scale grayed out (0) or responsive (1).

"*id:set_visible* {0|1}" hides (0) the scale, or makes it visible (1).

"*id:block* {0|1}" blocks (1) or unblocks (0) feedback messages. Initially, the scale is unblocked.

"*id:set_tooltip_text* [*text*]" sets the scale's tooltip. Default is disabling the tooltip.

"*id:force*" simulates fresh selection of the current value.

"*id:snapshot file*{.ps|.eps|.epsf|.pdf|.svg}" writes a snapshot of the scale to an image file of the specified type.

"*id:set_size_request* [*width height*]" sets the scale's minimum size to *width* x *height*. Default is the natural size.

"*id:style* [*prop:val*[: *prop:val* ...]]" sets the style. See *Style*.

Feedback "*id:value number*"

GtkScrolledWindow

Commands "*id:hscroll position*" and "*id:vscroll position*" scroll *position* to the left or top edge of the **GtkScrolledWindow**, respectively.

"*id:hscroll_to_range pos0 pos1*" and "*id:vscroll_to_range pos0 pos1*" scroll, if necessary, the range between *pos0* and *pos1* into the **GtkScrolledWindow**. If the range is greater than the window, the initial part of the range will be visible.

"*id:grab_focus*" puts the keyboard focus onto the widget.

"*id:set_sensitive* {0|1}" makes the widget grayed out (0) or responsive (1).

"*id:set_visible* {0|1}" hides (0) the widget, or makes it visible (1).

"*id:set_tooltip_text* [*text*]" sets the widget's tooltip. Default is disabling the tooltip.

"*id:snapshot file*{.ps|.eps|.epsf|.pdf|.svg}" writes a snapshot of the widget to an image file of the specified type.

"*id:set_size_request* [*width height*]" sets the minimum size to *width* x *height*. Default is the widget's natural size.

"*id:style* [*prop:val*]; [*prop:val* ...]" sets the style. See *Style*.

Feedback none

GtkSeparator

Commands "*id:set_sensitive* {**0**|**1**}" makes the separator insensitive (**0**) or responsive (**1**).

"*id:set_tooltip_text* [*text*]" sets the tooltip of the separator. Default is disabling the tooltip.

"*id:set_visible* {**0**|**1**}" hides (**0**) the separator, or makes it visible (**1**).

"*id:snapshot file*{**.ps|.eps|.epsf|.pdf|.svg**}" writes a snapshot of the separator to an image file of the specified type.

"*id:set_size_request* [*width height*]" sets the minimum size to *width* x *height*. Default is the natural size of the separator.

"*id:style* [*prop:val*]; [*prop:val* ...]" sets the widget style. See *Style*.

Feedback none

GtkSeparatorMenuItem

Commands "*id:set_sensitive* {**0**|**1**}" makes the menu item insensitive (**0**) or responsive (**1**).

"*id:set_visible* {**0**|**1**}" hides (**0**) the menu item, or makes it visible (**1**).

"*id:set_tooltip_text* [*text*]" sets the tooltip of the menu item. Default is disabling the tooltip.

"*id:snapshot file*{**.ps|.eps|.epsf|.pdf|.svg**}" writes a snapshot of the menu item to an image file of the specified type.

"*id:set_size_request* [*width height*]" sets the minimum size to *width* x *height*. Default is the natural size of the menu item.

"*id:style* [*prop:val*]; [*prop:val* ...]" sets the widget style. See *Style*.

Feedback none

GtkSocket

GtkSocket may be unsupported by Glade, but its definition can be inserted manually into the GtkBuilder (*.ui*) file:

```

<child>
  <object class="GtkSocket" id="socket1">
    <property name="visible">True</property>
    <property name="can_focus">True</property>
  </object>
  <packing>
    <property name="expand">True</property>
    <property name="fill">True</property>
    <property name="position">1</property>
  </packing>
</child>

```

Commands "**id:id**" requests a feedback message containing the socket *xid*.

"**id:grab_focus**" puts the keyboard focus onto the socket.

"**id:set_sensitive** {**0|1**}" makes the socket grayed out (**0**) or responsive (**1**).

"**id:set_visible** {**0|1**}" hides (**0**) the socket, or makes it visible (**1**).

"**id:set_tooltip_text** [*text*]" sets the socket's tooltip. Default is disabling the tooltip.

"**id:snapshot file**{**.ps|.eps|.epsf|.pdf|.svg**}" writes a snapshot of the socket to an image file of the specified type.

"**id:set_size_request** [*width height*]" sets the socket's minimum size to *width* x *height*. Default is the socket's natural size.

"**id:style** [*prop:val*; *prop:val ...*]" sets the style. See *Style*.

Feedback "**id:id xid**" can be used by another process to XEmbed its widgets into the **GtkSocket**.

"**id:plug-added**", "**id:plug-removed**". Notification that the other process has inserted its widgets into or removed them from the **GtkSocket**.

GtkSpinButton

Commands "**id:set_text** *string*" sets the selected value to *string*.

"**id:set_range** *min max*" changes the range.

"**id:set_increments** *step page*" sets the steps the value changes while pressing arrow keys/left mouse button, and 'Page_Up' / 'Page_Down' /middle mouse button, respectively.

"**id:grab_focus**" puts the keyboard focus onto the button.

"**id:set_sensitive** {0|1}" makes the button grayed out (**0**) or responsive (**1**).

"**id:set_visible** {0|1}" hides (**0**) the button, or makes it visible (**1**).

"**id:block** {0|1}" blocks (**1**) or unblocks (**0**) feedback messages. Initially, the button is unblocked.

"**id:set_tooltip_text** [*text*]" sets the button's tooltip. Default is disabling the tooltip.

"**id:force**" simulates fresh selection of the current value

"**id:snapshot file**{.ps|.eps|.epsf|.pdf|.svg}" writes a snapshot of the button to an image file of the specified type.

"**id:set_size_request** [*width height*]" sets the button's minimum size to *width* x *height*. Default is the widget's natural size.

"**id:style** [*prop:val*; *prop:val ...*]" sets the button's style. See *Style*.

Feedback "**id:text** *text*"

GtkSpinner

Commands "**id:start**" and "**id:stop**" start and stop the spinner.

"**id:set_sensitive** {0|1}" makes the spinner grayed out (**0**) or clearly visible (**1**).

"**id:set_visible** {0|1}" hides (**0**) the spinner, or makes it visible (**1**).

"**id:set_tooltip_text** [*text*]" sets the spinner's tooltip. Default is disabling the tooltip.

"**id:snapshot file**{.ps|.eps|.epsf|.pdf|.svg}" writes a snapshot of the spinner to an image file of the specified type.

"**id:set_size_request** [*width height*]" sets the spinner's minimum size to *width* x *height*. Default is the natural size.

"**id:style** [*prop:val*; *prop:val ...*]" sets the style. See *Style*.

Feedback none

GtkStatusbar

The context parameter *c* is an arbitrary non-whitespace string.

Commands "**id:push_id** *c string*", "**id:push** *string*" associate *string* with context parameter *c* or '0', respectively, and display it in the statusbar.

"**id:pop_id** *c*", "**id:pop**" remove the latest entry associated with context parameter *c* or '0', respectively, from the statusbar.

"**id:remove_all_id** *c*", "**id:remove_all**" remove the entries associated with context parameter *c* or '0', respectively, from the statusbar.

"**id:set_sensitive** {0|1}" makes the statusbar grayed out (0) or clearly visible (1).

"**id:set_visible** {0|1}" hides (0) the statusbar, or makes it visible (1).

"**id:set_tooltip_text** [*text*]" sets the statusbar's tooltip. Default is disabling the tooltip.

"**id:snapshot** *file*{.ps|.eps|.epsf|.pdf|.svg}" writes a snapshot of the statusbar to an image file of the specified type.

"**id:set_size_request** [*width height*]" sets the statusbar's minimum size to *width* x *height*. Default is the natural size.

"**id:style** [*prop:val*; *prop:val* ...]" sets the statusbar's style. See *Style*.

Feedback none

GtkSwitch

Commands "**id:set_active** {0|1}" turns the switch off (0) or on (1).

"**id:grab_focus**" puts the keyboard focus onto the switch.

"**id:set_sensitive** {0|1}" makes the switch grayed out (0) or responsive (1).

"**id:set_visible** {0|1}" hides (0) the switch, or makes it visible (1).

"**id:block** {0|1}" blocks (1) or unblocks (0) feedback messages. Initially, the switch is unblocked.

"**id:set_tooltip_text** [*text*]" sets the tooltip of the switch. Default is disabling the tooltip.

"**id:force**" simulates a click on the switch.

"*id:snapshot file*{*.ps|.eps|.epsf|.pdf|.svg*}" writes a snapshot of the switch to an image file of the specified type.

"*id:set_size_request* [*width height*]" sets the switch's minimum size to *width* x *height*. Default is the natural size.

"*id:style* [*prop:val*[:*prop:val* ...]]" sets the style. See *Style*.

Feedback "id:1" if switched on, or "id:0" otherwise.

GtkTextView

There should be a dedicated *GtkButton* for sending (parts of) the text. If the id of the **GtkTextView** is *foo*, a *GtkButton* with id *foo_send_text* will send the content of the **GtkTextView**; a *GtkButton* with id *foo_send_selection* will send the highlighted part the **GtkTextView**.

Commands "*id:set_text string*" replaces the user-editable text with (potentially empty) *string*.

"*id:delete*" deletes the text.

"*id:insert_at_cursor string*" inserts *string* at cursor position.

"*id:place_cursor* {*position* | **end**}" places the text cursor at *position* or at the end of the text.

"*id:place_cursor_at_line line*" places the text cursor at the beginning of *line*.

"*id:scroll_to_cursor*" scrolls to the cursor position if necessary.

"*id:save file*" stores in *file* a **pipeglade** command containing the text.

"*id:grab_focus*" puts the keyboard focus into the textview.

"*id:set_sensitive* {**0**|**1**}" makes the text view grayed out (**0**) or responsive (**1**).

"*id:set_visible* {**0**|**1**}" hides (**0**) the text view, or makes it visible (**1**).

"*id:set_tooltip_text* [*text*]" sets the tooltip of the text view. Default is disabling the tooltip.

"*id:snapshot file*{*.ps|.eps|.epsf|.pdf|.svg*}" writes a snapshot of the text view to an image file of the specified type.

"*id:set_size_request* [*width height*]" sets the minimum size to *width* x *height*. Default is

the natural size.

"*id:style* [*prop:val*[: *prop:val* ...]]" sets the style. See *Style*.

Feedback "*button_id:text text*", *button_id* being the id of the *GtkButton*. Line endings in *text* are replaced by `\n`, and backslashes are replaced by `\\`.

GtkToggleButton

Commands "*id:set_active* {**0**|**1**}" switches the button off (**0**) or on (**1**).

"*id:set_label string*" replaces the button text with *string*.

"*id:grab_focus*" puts the keyboard focus onto the button.

"*id:set_sensitive* {**0**|**1**}" makes the button grayed out (**0**) or responsive (**1**).

"*id:set_visible* {**0**|**1**}" hides (**0**) the button, or makes it visible (**1**).

"*id:block* {**0**|**1**}" blocks (**1**) or unblocks (**0**) feedback messages. Initially, the button is unblocked.

"*id:set_tooltip_text [text]*" sets the button's tooltip. Default is disabling the tooltip.

"*id:force*" simulates a click on the button.

"*id:snapshot file*{`.ps`|`.eps`|`.epsf`|`.pdf`|`.svg`}" writes a snapshot of the button to an image file of the specified type.

"*id:set_size_request [width height]*" sets the button's minimum size to *width* x *height*. Default is the button's natural size.

"*id:style* [*prop:val*[: *prop:val* ...]]" sets the button's style. See *Style*.

Feedback "*id:1*" if switched on, or "*id:0*" otherwise.

GtkTreeView

pipeglade can deal with columns of type **gboolean**, **gint**, **guint**, **glong**, **gulong**, **gint64**, **guint64**, **gfloat**, **gdouble**, and **gchararray**.

row and *column* refer to the underlying model (**GtkListStore** or **GtkTreeStore**). *row* is a sequence of one or more colon-separated integers, e.g. '3' or '0:0:1'.

Commands "*id:set row column data*" replaces the content at (*row*, *column*) with *data* (which should

be compatible with the type of *column*). If necessary, new tree nodes are created to obtain the minimal tree structure needed to support *row*.

"**id:insert_row** {*row* [**as_child**] | **end**}" inserts a new, empty row; either as a sibling of *row* at position *row*, or as a child of *row*, or at the end of the list, respectively.

"**id:move_row** *origin* {*destination* | **end**}" moves the row at *origin* within its current level to *destination* or to the end.

"**id:remove_row** *row*" removes the row at position *row*.

"**id:clear**" removes all rows.

"**id:expand** *row*" expands one level of the subtree below *row*.

"**id:expand_all** [*row*]" expands the subtree below *row*, or the whole tree.

"**id:collapse** [*row*]" collapses the subtree below *row*, or the whole tree.

"**id:set_cursor** [*row*]" sets the cursor to *row*, or unsets it.

"**id:scroll** *row* *column*" scrolls the cell at (*row*, *column*) into view.

"**id:save_file**" stores the content of the underlying model as a sequence of **pipeglade** commands into *file*.

"**id:grab_focus**" puts the keyboard focus onto the tree view.

"**id:set_sensitive** {**0**|**1**}" makes the tree view grayed out (**0**) or responsive (**1**).

"**id:set_visible** {**0**|**1**}" hides (**0**) the tree view, or makes it visible (**1**).

"**id:block** {**0**|**1**}" (with *id* from the underlying **GtkTreeSelection**) blocks (**1**) or unblocks (**0**) feedback messages. Initially, the widget is unblocked.

"**id:set_tooltip_text** [*text*]" sets the tooltip of the tree view. Default is disabling the tooltip.

"**id:snapshot** *file* {**.ps**|**.eps**|**.epsf**|**.pdf**|**.svg**}" writes a snapshot of the tree view to an image file of the specified type.

"**id:set_size_request** [*width* *height*]" sets the minimum size of the tree view to *width* x *height*. Default is the widget's natural size.

"**id:style** [*prop:val*[: *prop:val* ...]]" sets the style. See *Style*.

Feedback "**id:clicked**"

"**id:column_type** *row column value*", one message per cell in the underlying model for each selected row; or

"**id:column_type** *row column new_value*", if the cell at (*row*, *column*) has been edited.

GtkTreeViewColumn

Commands "**id:block** {**0**|**1**}" blocks (**1**) or unblocks (**0**) feedback messages. Initially, the widget is unblocked.

Feedback "**id:clicked**"

GtkViewport

Commands "**id:set_sensitive** {**0**|**1**}" makes the viewport insensitive (**0**) or responsive (**1**).

"**id:set_visible** {**0**|**1**}" hides (**0**) the viewport, or makes it visible (**1**).

"**id:set_tooltip_text** [*text*]" sets the tooltip of the viewport. Default is disabling the tooltip.

"**id:snapshot** *file*{*.ps*|*.eps*|*.epsf*|*.pdf*|*.svg*}" writes a snapshot of the viewport to an image file of the specified type.

"**id:set_size_request** [*width height*]" sets the minimum size to *width x height*. Default is the natural size of the viewport.

"**id:style** [*prop:val*[: *prop:val* ...]]" sets the widget style. See *Style*.

Feedback none

GtkWindow

Commands "**id:set_title** *string*" replaces the text in the title bar with *string*.

"**id:resize** [*width height*]" changes the window size to *width x height* pixels if specified, or to the default size.

"**id:move** *x y*" moves the window to position (*x*, *y*).

"**id:fullscreen**" and "**id:unfullscreen**" switch fullscreen mode on and off.

"*id:set_sensitive* {0|1}" makes the widget grayed out (0) or responsive (1).

"*id:set_visible* {0|1}" hides (0) the widget, or makes it visible (1).

"*id:block* {0|1}" blocks (1) or unblocks (0) feedback messages. Initially, the widget is unblocked.

"*id:set_tooltip_text* [*text*]" sets the window's tooltip. Default is disabling the tooltip.

"*id:snapshot file*{.ps|.eps|.epsf|.pdf|.svg}" writes a snapshot of the window to an image file of the specified type.

"*id:set_size_request* [*width height*]" sets the window's minimum size to *width x height*. Default is the widget's natural size.

"*id:style* [*prop:val*; *prop:val* ...]" sets the style. See *Style*.

Feedback "*id:closed*" if the **GtkWindow** was closed by the window manager.

SPECIAL SYNTAX

Style

"*id:style* [*prop:val*; *prop:val* ...]" sets properties *prop* of the widget style to *val*. Properties not explicitly specified are reset to their default values. Examples:

- 'label1:style font:Bold 11; background-color:green'
- 'label1:style font-style:italic; font-weight:bold; color:blue'
- 'button1:style background-image:url("q.png")'
- 'treeview1:style background-image:linear-gradient(45deg, yellow, blue);'
- 'frame1:style border-color:red yellow; border-width:5px'
- 'frame1:style border-radius:10px; transition:10s'

For color notations see *Color*, below.

Color

Possible *color* notations:

- standard X11 colors names like 'blue', 'Dark Sea Green', or 'Red';
- 'transparent';
- hexadecimal values in the form #*rgb*, #*rrggbb*, #*rrrrggbbb*, or #*rrrrggggbbbb*;
- RGB colors in the form **rgb**(*red,green,blue*) or RGBA colors in the form **rgba**(*red,green,blue,alpha*); *red*, *green*, *blue* between 0 and 255 and *alpha* between 0 and 1;
- color expressions like 'darker(blue)' or 'lighter(currentColor)'.

EXIT STATUS

The **pipeglade** utility exits 0 on success, and >0 if an error occurs.

EXAMPLES

Discovering Pipeglade Interactively

Suppose the interface in `./pipeglade.ui` has a `GtkLabel` 'label1' and a `GtkButton` 'button1'. After invoking

```
pipeglade
```

and clicking the `GtkButton`, 'button1:clicked' will be reported on the terminal. Typing

```
label1:set_text The Button
```

will change the text shown on the label into 'The Button'.

One-Shot File Dialog

Suppose the interface in `./simple_open.ui` contains a `GtkFileChooserDialog` with an 'OK' `GtkButton` whose id is 'main_ok'. Invoking

```
pipeglade -u simple_open.ui
```

will open the dialog; pressing 'OK' will close it after sending the selected filename to `stdout`.

One-Shot User Notification

If the interface in `./simple_dialog.ui` contains a `GtkLabel` 'label1', then

```
pipeglade -u simple_dialog.ui <<<< \  
"label1:set_text NOW READ THIS!"
```

will set the label text accordingly and wait for user input.

Continuous Input

The following shell command displays a running clock:

```
while true; do  
  echo "label1:set_text `date`";  
  sleep 1;  
done | pipeglade -u simple_dialog.ui
```

Continuous Input and Output

The following shell script fragment sets up **pipeglade** for continuous communication with another program, `main_prog`:

```
pipeglade -i in.fifo -o out.fifo -b  
main_prog <out.fifo >in.fifo
```

SEE ALSO

dialog(1), glade(1), gxmmessage(1), kdialog(1), whiptail(1), xmessage(1), zenity(1)

AUTHOR

pipeglade was written by Bert Burgemeister <trebbu@googlemail.com>.

BUGS

Due to what appears to be a bug in cairo v1.14.0, **pipeglade** used with this library version occasionally crashes on **GtkDrawingArea** commands.