NAME

Gtk3 − Perl interface to the 3.x series of the gtk+ toolkit

SYNOPSIS

```
use Gtk3 -init;
my $window = Gtk3::Window->new ('toplevel');
my $button = Gtk3::Button->new ('Quit');
$button->signal_connect (clicked => sub { Gtk3::main_quit });
$window->add ($button);
$window->show_all;
Gtk3::main;
```

ABSTRACT

Perl bindings to the 3.x series of the gtk+ toolkit. This module allows you to write graphical user interfaces in a Perlish and object-oriented way, freeing you from the casting and memory management in C, yet remaining very close in spirit to original API.

DESCRIPTION

The `Gtk3` module allows a Perl developer to use the gtk+ graphical user interface library. Find out more about gtk+ at <http://www.gtk.org>.

The gtk+ reference manual is also a handy companion when writing `Gtk3` programs in Perl: <http://developer.gnome.org/gtk3/stable/>. The Perl bindings follow the C API very closely, and the C reference documentation should be considered the canonical source. The principles underlying the mapping from C to Perl are explained in the documentation of Glib::Object::Introspection, on which `Gtk3` is based.

Glib::Object::Introspection also comes with the `perli11ndoc` program which displays the API reference documentation of all installed libraries organized in accordance with these principles.

Wrapped libraries

`Gtk3` automatically sets up the following correspondence between C libraries and Perl packages:

```
Library        | Package
---------------+----------
Gtk-3.0        | Gtk3
Gdk-3.0        | Gtk3::Gdk
GdkPixbuf-2.0  | Gtk3::Gdk
GdkPixdata-2.0| Gtk3::Gdk
Pango-1.0      | Pango
```

Import arguments

When importing `Gtk3`, you can pass `-init` as in `use Gtk3 -init;` to have `Gtk3::init` automatically called. You can also pass a version number to require a certain version of `Gtk3`.

Customizations and overrides

In order to make things more Perlish or to make porting from `Gtk2` to `Gtk3` easier, `Gtk3` customizes the API generated by Glib::Object::Introspection in a few spots:

- The array ref normally returned by the following functions is flattened into a list:

  Gtk3::ActionGroup::list_actions
  Gtk3::Builder::get_objects
  Gtk3::CellLayout::get_cells
  Gtk3::Container::get_children
  Gtk3::SizeGroup::get_widgets
  Gtk3::TreePath::get_indices
  Gtk3::TreeView::get_columns
  Gtk3::UIManager::get_action_groups

Gtk3::UIManager::get_toplevels
Gtk3::Window::list_toplevels
Gtk3::stock_list_ids
Gtk3::Gdk::Pixbuf::get_formats

- The following functions normally return a boolean and additional out arguments, where the boolean indicates whether the out arguments are valid. They are altered such that when the boolean is true, only the additional out arguments are returned, and when the boolean is false, an empty list is returned.

  Gtk3::TextBuffer::get_selection_bounds
  Gtk3::TreeModel::get_iter
  Gtk3::TreeModel::get_iter_first
  Gtk3::TreeModel::get_iter_from_string
  Gtk3::TreeModel::iter_children
  Gtk3::TreeModel::iter_nth_child
  Gtk3::TreeModel::iter_parent
  Gtk3::TreeModelFilter::convert_child_iter_to_iter
  Gtk3::TreeModelSort::convert_child_iter_to_iter
  Gtk3::TreeSelection::get_selected
  Gtk3::TreeView::get_dest_row_at_pos
  Gtk3::TreeView::get_path_at_pos
  Gtk3::TreeView::get_tooltip_context
  Gtk3::TreeView::get_visible_range
  Gtk3::TreeViewColumn::cell_get_position
  Gtk3::stock_lookup
  Gtk3::Gdk::Event::get_axis
  Gtk3::Gdk::Event::get_button
  Gtk3::Gdk::Event::get_click_count
  Gtk3::Gdk::Event::get_coords
  Gtk3::Gdk::Event::get_keycode
  Gtk3::Gdk::Event::get_keyval
  Gtk3::Gdk::Event::get_scroll_direction
  Gtk3::Gdk::Event::get_scroll_deltas
  Gtk3::Gdk::Event::get_state
  Gtk3::Gdk::Event::get_root_coords
  Gtk3::Gdk::Window::get_origin

- Values of type Gtk3::ResponseType are converted to and from nick names if possible, while still allowing raw IDs, in the following places:

  – For Gtk3::Dialog and Gtk3::InfoBar: the signal `response` as well as the methods `add_action_widget`, `add_button`, `add_buttons`, `response`, `set_default_response` and `set_response_sensitive`.
  – For Gtk3::Dialog: the methods `get_response_for_widget`, `get_widget_for_response`, `run` and `set_alternative_button_order`.

- Values of type Gtk3::IconSize are converted to and from nick names if possible, while still allowing raw IDs, in the following places:

  – Gtk3::Image: the constructors new_from_stock, new_from_icon_set, new_from_icon_name and new_from_gicon, the getters get_stock, get_icon_set, get_icon_name and get_gicon and the setters set_from_stock, set_from_icon_set, set_from_icon_name, set_from_gicon.
  – Gtk3::Widget: the method render_icon.

- The constants `Gtk3::EVENT_PROPAGATE` and `Gtk3::EVENT_STOP` can be used in handlers for event signals like `key-press-event` to indicate whether or not the event should continue propagating through the widget hierarchy.

- The records corresponding to the various Gtk3::Gdk::Event types, like `expose` or `key-release`, are represented as objects blessed into specific Perl packages, like `Gtk3::Gdk::EventExpose` or `Gtk3::Gdk::EventKey`, that all inherit from `Gtk3::Gdk::Event`. This allows you to seemlessly access type-specific fields as well as common fields, as in `$event->window` or `$event->keyval`.

- Gtk3::Gdk::Atom has overloads for the `==` and `!=` operators that check for equality of the underlying atoms.
- For backwards compatibility, the functions `Gtk3::get_version_info` and `Gtk3::GET_VERSION_INFO` are provided, and the functions `Gtk3::CHECK_VERSION`, `Gtk3::check_version`, `Gtk3::init`, `Gtk3::init_check`, `Gtk3::main`, `Gtk3::main_level` and `Gtk3::main_quit` can be called as class-static or as normal functions: for example, `Gtk3->main_quit` and `Gtk3::main_quit` are both supported. Additionally, `Gtk3::init` and `Gtk3::init_check` automatically handle passing and updating `@ARGV` as appropriate.
- A Perl reimplementation of `Gtk3::show_about_dialog` is provided.
- Perl reimplementations of `Gtk3::ActionGroup::add_actions`, `add_radio_actions` and `add_toggle_actions` are provided.
- `Gtk3::Builder::add_objects_from_file` and `add_objects_from_string` also accept a list of objects instead of an array ref.
- `Gtk3::Builder::add_objects_from_string` and `add_from_string` don't take length arguments, as they are computed automatically.
- A Perl reimplementation of `Gtk3::Builder::connect_signals` is provided.
- The default `new` constructors of Gtk3::Button, Gtk3::CheckButton, Gtk3::ColorButton, Gtk3::FontButton and Gtk3::ToggleButton reroute to `new_with_mnemonic` if given an extra argument.
- The default `new` constructor of Gtk3::CheckMenuItem reroutes to `new_with_mnemonic` if given an extra argument.
- The `length` argument of `Gtk3::Clipboard::set_text` is optional.
- Perl reimplementations of `Gtk3::Container::add_with_properties`, `Gtk3::Container::child_get` and `Gtk3::Container::child_set` are provided.
- `Gtk3::Container::find_child_property` and `Gtk3::Container::list_child_properties` are forwarded to the corresponding functions in `Gtk3::ContainerClass`.
- `Gtk3::Container::get_focus_chain` returns a list of widgets, or an empty list.
- `Gtk3::Container::set_focus_chain` also accepts a list of widgets.
- `Gtk3::CssProvider::load_from_data` also accepts a string.
- For Gtk3::Dialog and Gtk3::InfoBar, a Perl implementation of `add_buttons` is provided.
- `Gtk3::Dialog::new` can optionally be called as `Gtk3::Dialog->new (TITLE, PARENT, FLAGS, ...)` where `...` is a series of button text and response id pairs.
- A Perl implementation of `Gtk3::Dialog::new_with_buttons` is provided.
- The `length` argument of `Gtk3::Editable::insert_text` is optional.
- A Perl implementation of `Gtk3::FileChooserDialog::new` is provided.
- `Gtk3::HBox::new` uses the defaults homogeneous = FALSE and spacing = 5.
- The default `new` constructor of Gtk3::ImageMenuItem reroutes to `new_with_mnemonic` if given an extra argument.
- `Gtk3::InfoBar::new` can optionally be called as `Gtk3::InfoBar->new (...)` where `...` is a series of button text and response id pairs.

- A Perl reimplementation of `Gtk3::InfoBar::new_with_buttons` is provided.
- The default `new` constructor of Gtk3::LinkButton reroutes to `new_with_label` if given an extra argument.
- `Gtk3::ListStore::new` also accepts a list of type names.
- Gtk3::ListStore has a `get` method that calls `Gtk3::TreeModel::get` instead of `Glib::Object::get`.
- `Gtk3::ListStore::insert_with_values` also accepts a list of `column => value` pairs and reroutes to `insert_with_valuesv`.
- `Gtk3::ListStore::set` also accepts a list of `column => value` pairs.
- `Gtk3::Menu::popup` reroutes to `popup_for_device` for better callback handling.
- `Gtk3::Menu::popup_for_device` allows the given menu position func to return only x and y coordinates, defaulting `push_in` to FALSE.
- The default `new` constructor of Gtk3::MenuItem reroutes to `new_with_mnemonic` if given an extra argument.
- A Perl reimplementation of `Gtk3::MessageDialog::new` is provided.
- The group handling in the constructors and accessors of Gtk3::RadioAction, Gtk3::RadioButton, Gtk3::RadioMenuItem and Gtk3::RadioToolButton is amended to work correctly when given array refs of group members or single group members.
- Perl reimplementations of `Gtk3::RecentChooserDialog::new` and `new_for_manager` are provided.
- Redirects are provided from `Gtk3::Stock::[function]` to `Gtk3::stock_[function]` for `add`, `add_static`, `list_ids`, `lookup` and `set_translate_func`.
- A Perl reimplementation of `Gtk3::StyleContext::get` is provided.
- An override for `Gtk3::TargetEntry::new` is provided that automatically handles the conversion of the `flags` argument.
- A Perl reimplementation of `Gtk3::TextBuffer::create_tag` is provided.
- The `length` arguments of `Gtk3::TextBuffer::insert`, `insert_at_cursor`, `insert_interactive`, `insert_interactive_at_cursor`, `insert_markup` and `set_text` are optional.
- Perl reimplementations of `Gtk3::TextBuffer::insert_with_tags` and `insert_with_tags_by_name` are provided which do not require a `length` argument.
- A Perl reimplementation of `Gtk3::TreeModel::get` is provided.
- A redirect is added from `Gtk3::TreeModelFilter::new` to <Gtk3::TreeModel::filter_new> so that Gtk3::TreeModelFilter objects can be constructed normally.
- Gtk3::TreeModelFilter has a `get` method that calls `Gtk3::TreeModel::get` instead of `Glib::Object::get`.
- A redirect is added from `Gtk3::TreeModelSort::new_with_model` to <Gtk3::TreeModel::sort_new_with_model> so that Gtk3::TreeModelSort objects can be constructed normally.
- Gtk3::TreeModelSort has a `get` method that calls `Gtk3::TreeModel::get` instead of `Glib::Object::get`.
- `Gtk3::TreePath::new` redirects to `new_from_string` if an additional argument is given.
- A Perl reimplementation of `Gtk3::TreePath::new_from_indices` is provided.

- `Gtk3::TreeStore::new` also accepts a list of type names.
- Gtk3::TreeStore has a `get` method that calls `Gtk3::TreeModel::get` instead of `Glib::Object::get`.
- `Gtk3::TreeStore::insert_with_values` also accepts a list of `column => value` pairs.
- `Gtk3::TreeStore::set` also accepts a list of `column => value` pairs.
- `Gtk3::TreeView::new` redirects to `new_with_model` if an additional argument is given.
- A Perl reimplementation of `Gtk3::TreeView::insert_column_with_attributes` is provided.
- A Perl reimplementation of `Gtk3::TreeViewColumn::new_with_attributes` is provided.
- Perl reimplementations of `Gtk3::TreeViewColumn::set_attributes` and `Gtk3::CellLayout::set_attributes` are provided.
- `Gtk3::UIManager::add_ui_from_string` takes no `length` argument.
- `Gtk3::VBox::new` uses the defaults homogeneous = FALSE and spacing = 5.
- `Gtk3::Widget::add_events` and `Gtk3::Widget::set_events` also accept strings, array references and `Gtk3::Gdk::EventMask` objects for the `events` parameter.
- `Gtk3::Widget::get_events` returns a `Gtk3::Gdk::EventMask` object that can also be compared to numeric values with == and >=.
- `Gtk3::Widget::find_style_property` and `Gtk3::Widget::list_style_properties` are forwarded to the corresponding functions in `Gtk3::WidgetClass`.
- A Perl reimplementation of `Gtk3::Widget::style_get` is provided.
- `Gtk3::Window::new` uses the default type = 'toplevel'.
- A constructor `Gtk3::Gdk::RGBA::new` is provided that can be called as `Gtk3::Gdk::RGBA->new (r, g, b, a)`.
- `Gtk3::Gdk::RGBA::parse` can be called as a function returning a new instance (`$rgba = Gtk3::Gdk::RGBA::parse ($spec)`) or as a method (`$rgba->parse ($spec)`).
- `Gtk3::Gdk::Window::new` optionally computes the `attr_mask` automatically from the given `attr`.
- `Gtk3::Gdk::Pixbuf::get_pixels` returns a byte string.
- `Gtk3::Gdk::Pixbuf::new_from_data` is reimplemented in terms of `new_from_bytes` (with gdk-pixbuf >= 2.32) or `new_from_inline` (with gtk-pixbuf < 2.32) for correct memory management. No `destroy_fn` and `destroy_fn_data` arguments are needed.
- `Gtk3::Gdk::Pixbuf::new_from_inline` does not take a `copy_pixels` argument. It is always set to TRUE for correct memory management.
- `Gtk3::Gdk::Pixbuf::new_from_xpm_data` also accepts a list of XPM lines.
- `Gtk3::Gdk::Pixbuf::save`, `save_to_buffer` and `save_to_callback` also accept `key => value` pairs and invoke the correct C function as appropriate for the current gdk-pixbuf version.
- The `length` arguments of `Pango::Layout::set_text` and `set_markup` are optional.

Perl compatibility

As of 5.20.0, perl does not automatically re-check the locale environment for changes. If a function thus changes the locale behind perl's back, problems might arise whenever numbers are formatted, for example when checking versions. To ensure perl's assumption about the locale are up-to-date, the functions `Gtk3::init`, `init_check`, `init_with_args` and `parse_args` are amended to let perl know of any changes.

Porting from Gtk2 to Gtk3

The majority of the API has not changed, so as a first approximation you can run `s/Gtk2/Gtk3/` on your application. A big exception to this rule is APIs that were deprecated in gtk+ 2.x — these were all removed from gtk+ 3.0 and thus from Gtk3. The migration guide at <http://developer.gnome.org/gtk3/stable/migrating.html> describes what to use instead. Apart from this, here is a list of some other incompatible differences between Gtk2 and Gtk3:

- The call syntax for class-static methods is now always `Gtk3::Stock::lookup` instead of `Gtk3::Stock->lookup`.
- The `%Gtk2::Gdk::Keysyms` hash is gone; instead of `Gtk2::Gdk::Keysyms{XYZ}`, use `Gtk3::Gdk::KEY_XYZ`.
- The Gtk2::Pango compatibility wrapper was not carried over; simply use the namespace "Pango" everywhere. It gets set up automatically when loading Gtk3.
- The types Gtk3::Allocation and Gtk3::Gdk::Rectangle are now aliases for Cairo::RectangleInt, and as such they are represented as plain hashes with keys 'width', 'height', 'x' and 'y'.
- Gtk3::Editable: Callbacks connected to the "insert-text" signal do not have as many options anymore as they had in Gtk2. Changes to arguments will not be propagated to the next signal handler, and only the updated position can and must be returned.
- Gtk3::Menu: In gtk+ < 3.16, the position callback passed to **popup**() does not receive x and y parameters.
- Gtk3::RadioAction: The constructor now follows the C API.
- Gtk3::TreeModel: **iter_next**() is now a method that is modifying the iter directly, instead of returning a new one. **rows_reordered**() and the "rows-reordered" signal are currently unusable.
- Gtk3::TreeSelection: **get_selected_rows**() now returns two values: an array ref containing the selected paths, and the model. **get_user_data**() is not available currently.
- Gtk3::TreeSortable: **get_sort_column_id**() has an additional boolean return value.
- Gtk3::TreeStore, Gtk3::ListStore: **reorder**() is currently unusable.
- Gtk3::Widget: **grab_add**() and **grab_remove**() are methods now: `$widget->grab_add`, `$widget->grab_remove`.
- Gtk3::Gdk::Atom: The constructor **new**() is not provided anymore, and the class function **intern**() must now be called as `Gtk3::Gdk::Atom::intern (name, only_if_exists)`.
- Implementations of Gtk3::TreeModel: Gtk3::TreeIter now has a constructor called **new()** expecting `key => value` pairs; **new_from_arrayref**() does not exist anymore. To access the contents of Gtk3::TreeIter, use **stamp**(), **user_data**(), **user_data2**() and **user_data3**(); **to_arrayref**() does not exist anymore. **GET_ITER**(), **ITER_CHILDREN**(), **ITER_NTH_CHILD**() and **ITER_PARENT**() must return an additional boolean value. **ITER_NEXT**() must modify the iter and return a boolean rather than return a new iter. **GET_VALUE**() must return the value wrapped with `Glib::Object::Introspection::GValueWrapper->new`.
- Implementations of Gtk3::CellLayout: **GET_CELLS**() now needs to return an array ref instead of a list.

Note also that Gtk3::CHECK_VERSION will always fail when passed 2.y.z, so if you have any existing version checks in your code, you will most likely need to remove them.

SEE ALSO

- To discuss Gtk3 and ask questions join gtk−perl−list@gnome.org at <http://mail.gnome.org/mailman/listinfo/gtk−perl−list>.
- Also have a look at the gtk2−perl website and sourceforge project page, <http://gtk2−perl.sourceforge.net>.
- Glib

- Glib::Object::Introspection

AUTHORS

Torsten Schönfeld <kaffeetisch@gmx.de>

COPYRIGHT AND LICENSE

Copyright (C) 2011−2015 by Torsten Schoenfeld <kaffeetisch@gmx.de>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.