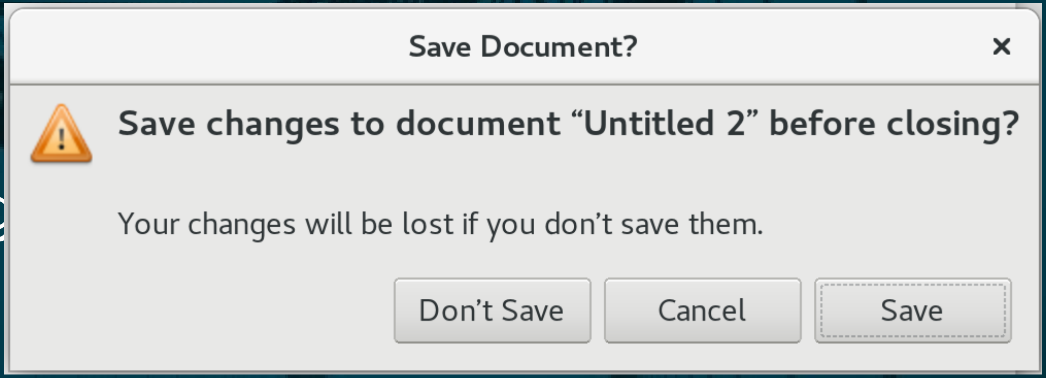
Native GTK UI

Native in 6.0

- File Dialog
  - Native for years
- Tooltips
  - Tell GTK the area the tooltip is for and GTK positions it
- Popovers
  - Formula typeahead indicator in calc
  - Page indicator in impress slide pane
- Menubar and menus
  - Context menus too

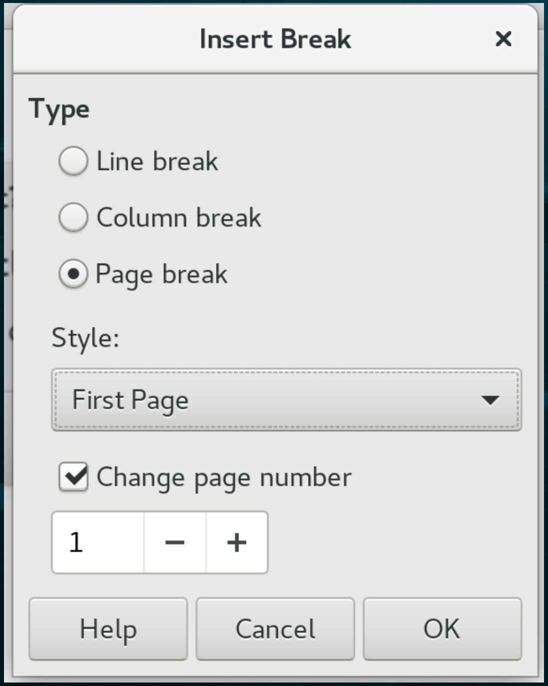
Native in 6.1

- Message Dialogs
- Some simple dialog



Save changes to document “Untitled 2” before closing?

Your changes will be lost if you don’t save them.



Type

○ Line break

○ Column break

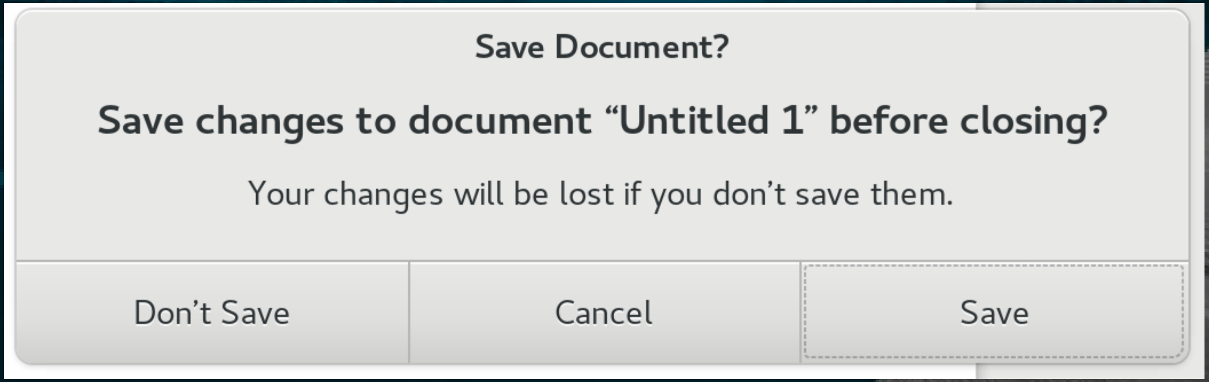
● Page break

Style:

First Page

☑ Change page number

1

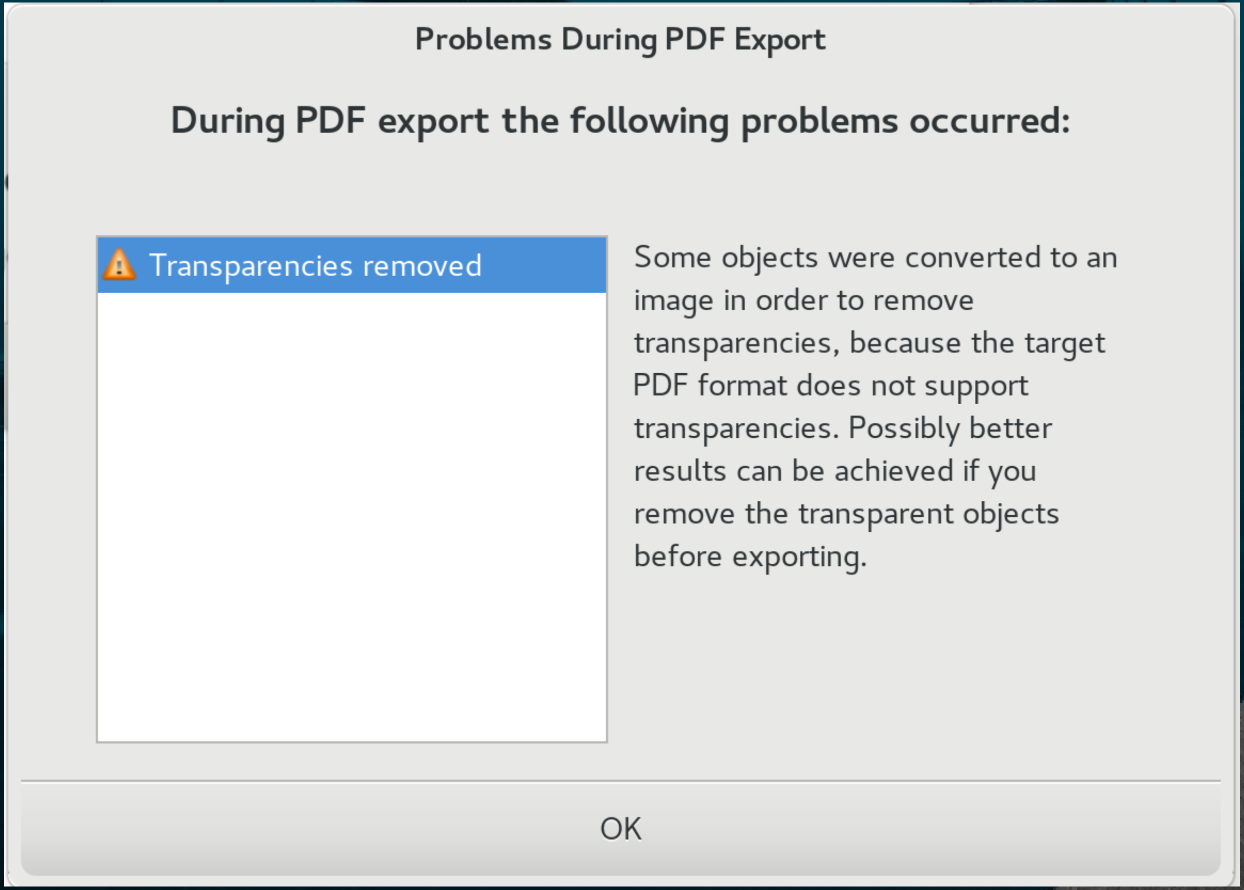


Save changes to document “Untitled 1” before closing?

Your changes will be lost if you don’t save them.

Native in 6.1

- Sample warning dialog with extra widgetry



Problems During PDF Export

During PDF export the following problems occurred:

⚠ Transparencies removed

Some objects were converted to an image in order to remove transparencies, because the target PDF format does not support transparencies. Possibly better results can be achieved if you remove the transparent objects before exporting.

Native towards 6.2

- 190+ GenericDialogControllers
- 80+ Tab Pages
  - Tab pages get reused in multiple dialogs, so...
  - Detect whether in a native SfxTabDialogController or a vcl-based SfxTabDialog and react accordingly
- 26+ SfxTabDialogControllers
  - Including the big ones, format character, format paragraph and format character
  - and the format area “six tab pages in a tab page”

Walk Through of Native GTK components

Walkthrough

- Tooltips, Popovers
- Native Message Dialog example
- Animated effects, e.g. radio/check buttons
- Color Menu Button/Line Style Menu Button
- Overlay Scrollbar
- Interactive Custom Widget
- Tabbed Dialog
  - A mega Tabbed Dialog with area tab, e.g Format Page
- GtkComboBox[Text] with images
- Password Caps Lock Indicator

UI Descriptions

Original UI Descriptions

- src file format
- Fixed positions
  - Measured in average character cell widths
  - Arbitrary language-based guesstimate multiplier
- Manually sized to longest translation strings
- No GUI Editor

Widget Builder UI Descriptions

- Gtk Builder file format
- Described in terms of Gtk Widgets
  - **Mapped to VCL Widgets**
- New VCL GtkGrid/GtkBox equivalents
- Dynamically sized and positioned
- Glade GUI Editor
- Resulted in 977 .ui files

redhat.

Translations

Old Translation Format

- src file input format
- Custom .res binary output format
- Each translation as a unique id number
- #define in .hrc included by .cxx and .src
- Custom tooling to convert .src <-> .po

Current Gettext Format

- Direct from .ui and c++ source files
- Standard .mo binary output format
- Each translation now a “Context”, “English source” pair #defined in .hrc
- Standard tooling to extract to .po and output .mo
  - Write .mo files with gettext tooling
  - Read .mo files with boost::gettext
- 22162 translations

Native GTK LibreOffice UI

Native UI Loading

- Load the .ui files natively with GTK own GtkBuilder API
- Let GTK load the .mo files by itself for translations
- Bind (weld) to those native GTK widgets from LibreOffice
- Current LibreOffice .ui loading code is fallback implementation for the non-GTK case

Native UI Loading

- A half-way house API, nudge a few places to behave more like the other.
- A GTK implementation in terms of the GTK API
- Fallback VCL implementation in terms of existing VCL API
- New “Custom” widget with Paint callback providing a VirtualDevice, etc.
  - In vcl case blit to VCL Windows OutputDevice
  - In GTK case blit VirtualDevices underlying cairo surface during draw signal

File Format details #1

- GtkRadioButton groups have to all link to the active entry, which itself has to link to nothing
  - We typically linked them around in a circle, a grouped to b, b to c, c to d and d to a.
- Have to have a different GtkAdjustment for each GtkSpinButton
  - We often reused the same GtkAdjustment to describe starting conditions of multiple widgets
- GtkSpinButton “output” signal to format value, i.e. support LibreOffice some what unusual “10.00 cm”, “20%” formatting
  - Remove “:UNIT” naming hack, move unit to code as bind time argument

File Format details #2

- Vertical action areas where we have vertical buttons on the right of a dialog have to made horizontal instead

- GtkTreeView have to have an associated GtkTreeViewColumn with a GtkCellRendererText set for column 0

- GtkComboBox using liststores can typically be converted to GtkComboBoxText

- All GtkTreeViews (and GtkComboBoxes) have to have the same model/renderer layouts

Custom Widgets

- Three main types
- Simple wrappers
  - e.g. Numbering List box, just populates a ComboBox with available numering types
- Preview widgets
  - e.g. format character preview
- Interactive widgets
  - e.g. color selector ValueSet, anchoring selector RectCTL

Custom Widgets

- Custom Widgets typically inherited from vcl::Window or Control
- Now inherit from CustomWidgetController, which provides mostly the same API, GetFocus, LoseFocus, MouseButtonDown, etc
- Doesn’t inherit from vcl::Window, so finds some missing places of the double-buffering work
- The .ui element is a GtkDrawingArea as its canvas
  - VCL_BUILDER_FACTORY dlsym hack gets removed
  - Custom widget description in glade catalog gets removed
  - Borders now in .ui via a surrounding GtkScrolledWindow
- Connect a CustomWidgetController to its canvas via a CustomWeld, which takes as arguments the CustomWidgetController and the name of the GtkDrawingArea

Custom Widget Accessibility

- Like as when inheriting from vcl::Window, a custom widget can implement CreateAccessible() which returns a uno object implenting the a11y apis.
- The LibreOffice GTK<->libreoffice-ally bridge, previously used for the “mega-widget” is reused to connect to the underlying GTK individual canvas widget’s a11y.
- So, the native GTK widgets use their own native a11y, the custom widgets hook up to our preexisting a11y

Potential Gotchas

- Parents of dialogs sometimes described as awt::XWindow
  - Currently just enough of XWindow implemented to smuggle a weld::Window through as an argument to make that work
- Minimum targeted version of GTK is GTK 3.18, so can’t use any properties not available there or it crashes
  - Glade **typically** warns about these, but not always
- Pretty popovers can only escape dialogs under wayland, so boring alternative used under X
- No Typeahead in Entry-less ComboBoxes
- GtkNotebook scrolling tabs, not double-decker rows

End, Thanks