

PLASMA 6 IS COMING

What to Expect, UX and under the Hood

Marco Martin Niccolò Venerandi





We also want Plasma 6 to be the most conservative major version change ever.

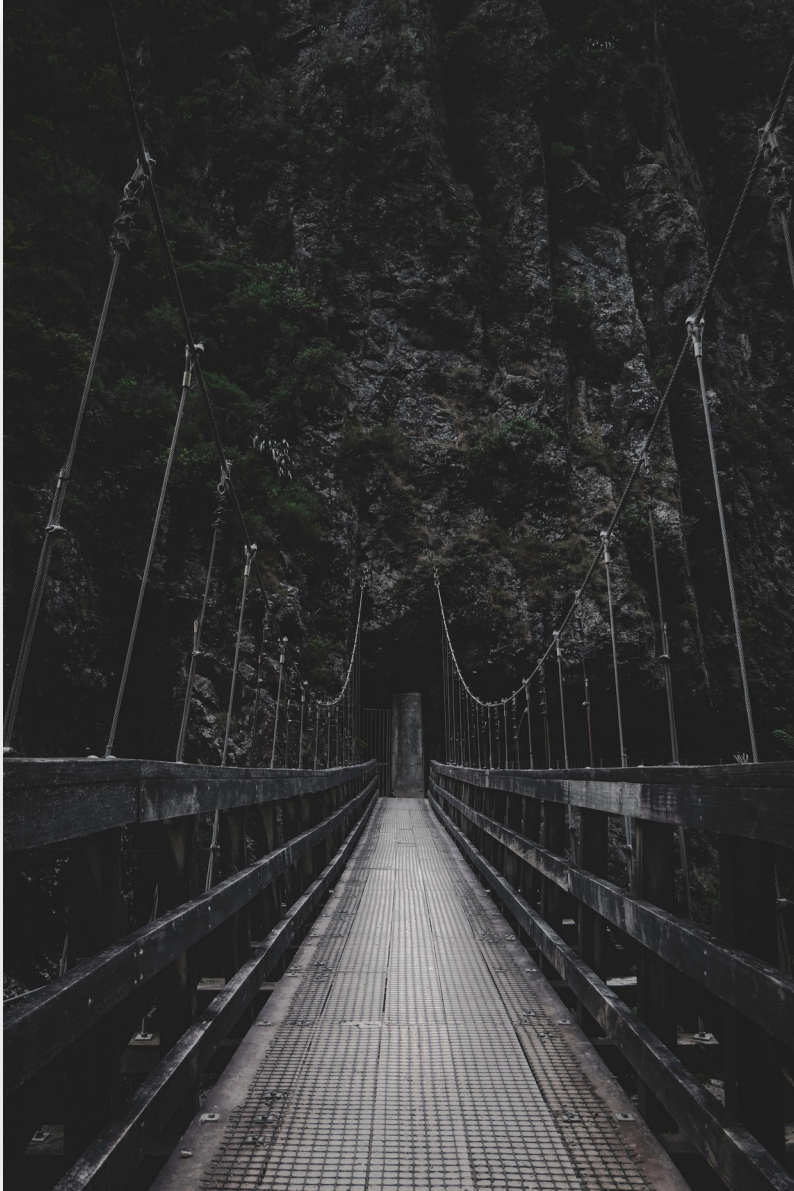
~VDG chat, months ago

We're not actually trying to sell Plasma 6 as a shiny new product.

~VDG chat, still months ago

Ahahah, just kidding, Plasma 6 is actually completely packed with redesigns already

~Me, whilst drafting this presentation

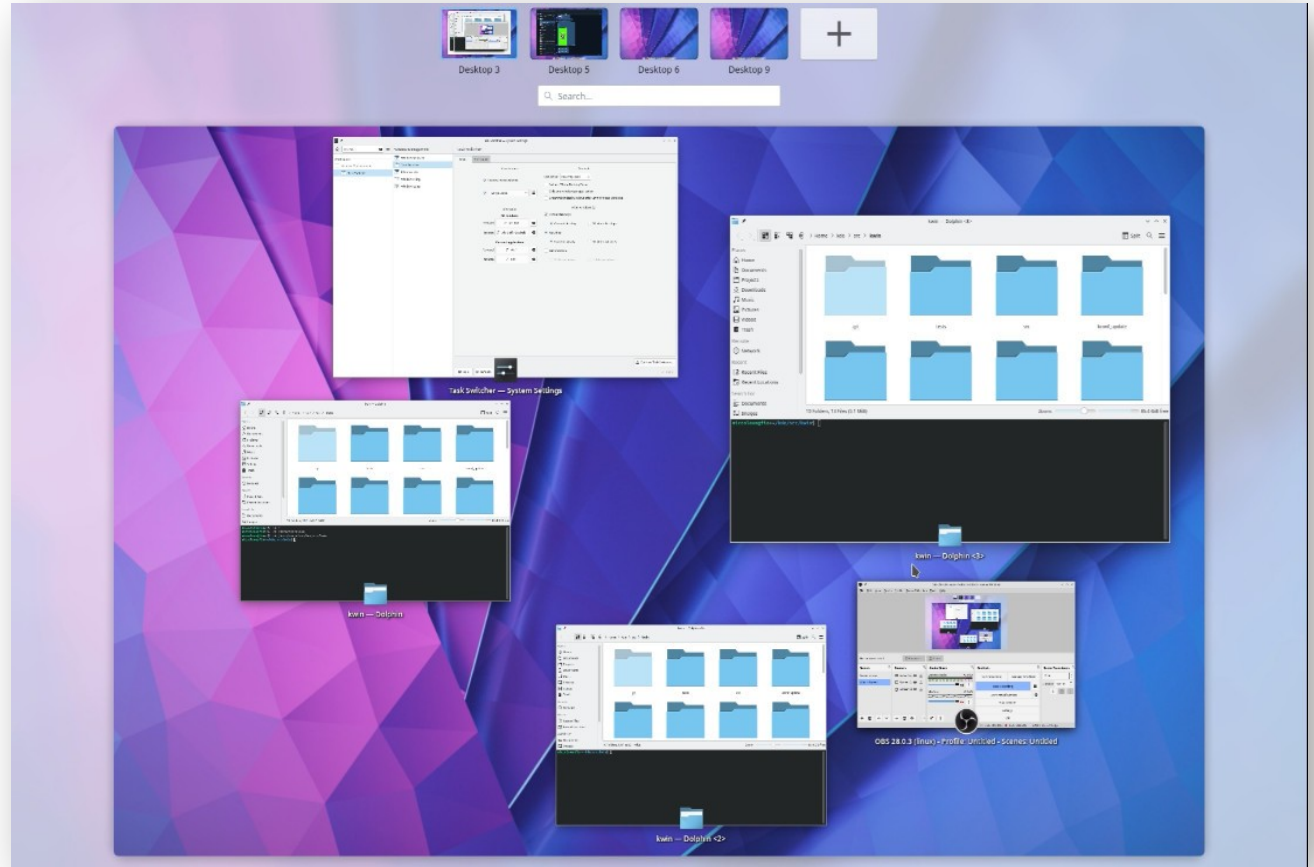


PLASMA STUFF



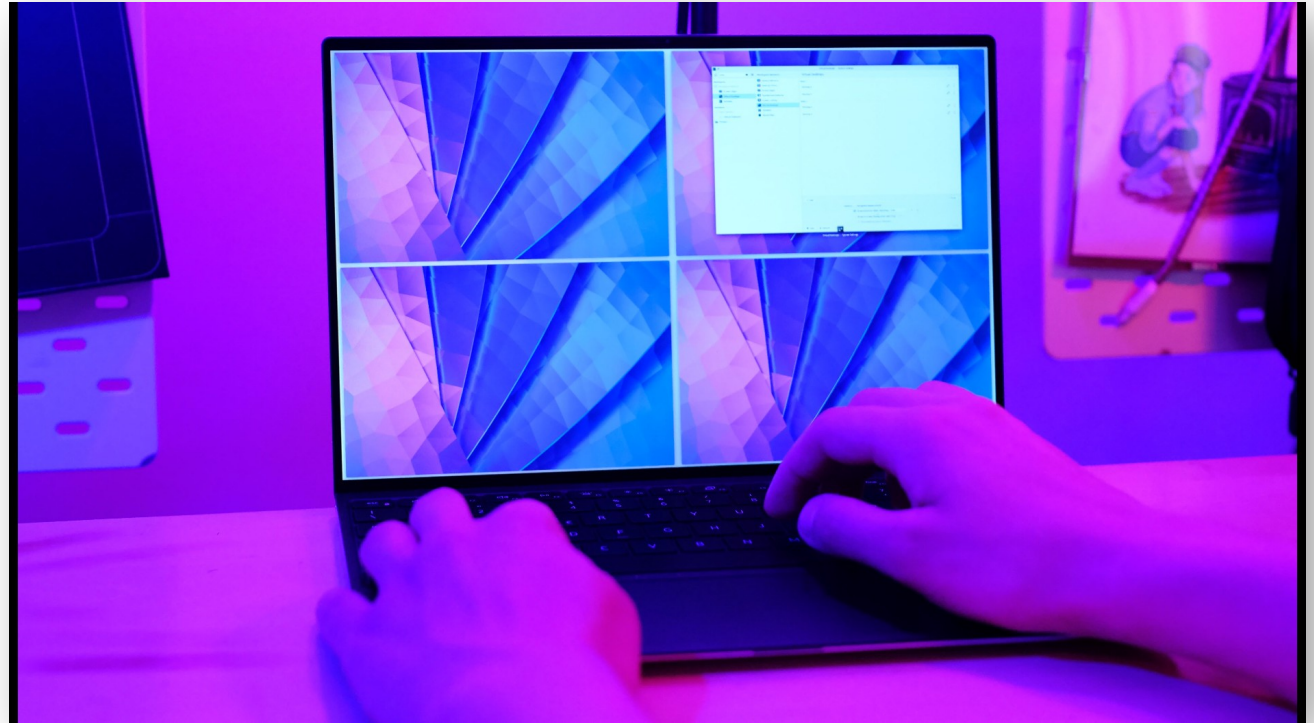
FUN STUFF

COMPLETELY REDESIGNED OVERVIEW



FUN STUFF

COMPLETELY REDESIGNED GRID VIEW



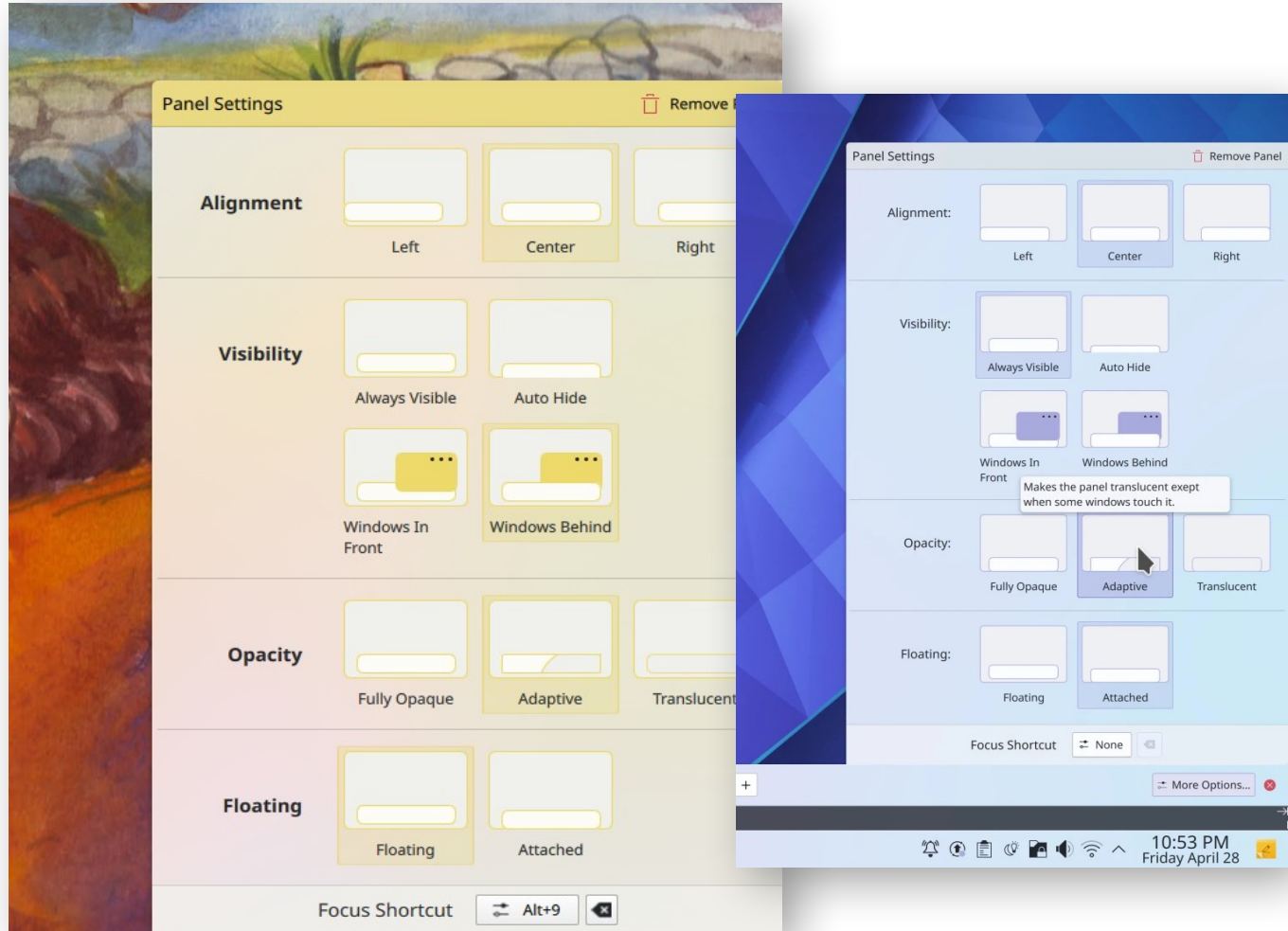
FUN STUFF

1:1 TOUCH PAD/SCREEN GESTURES



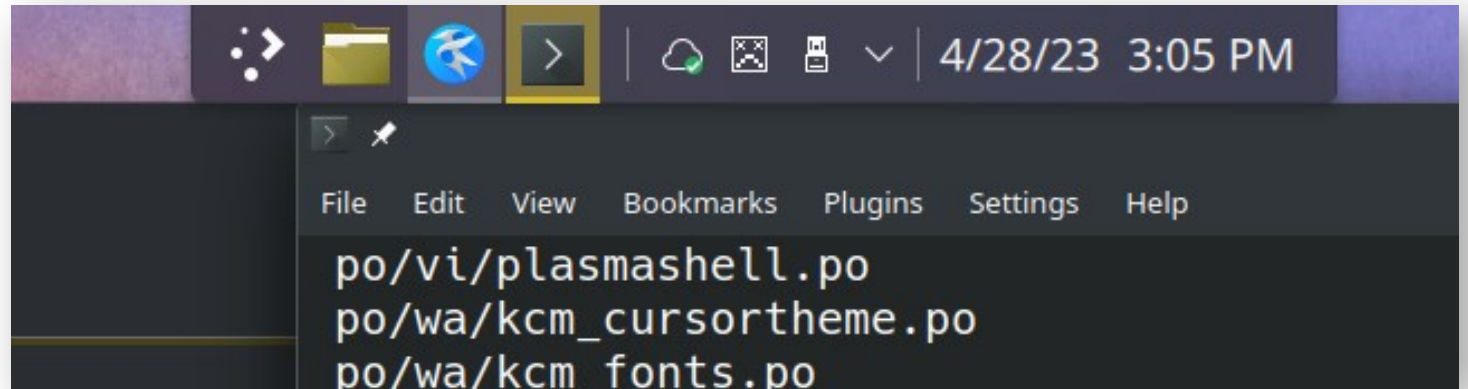
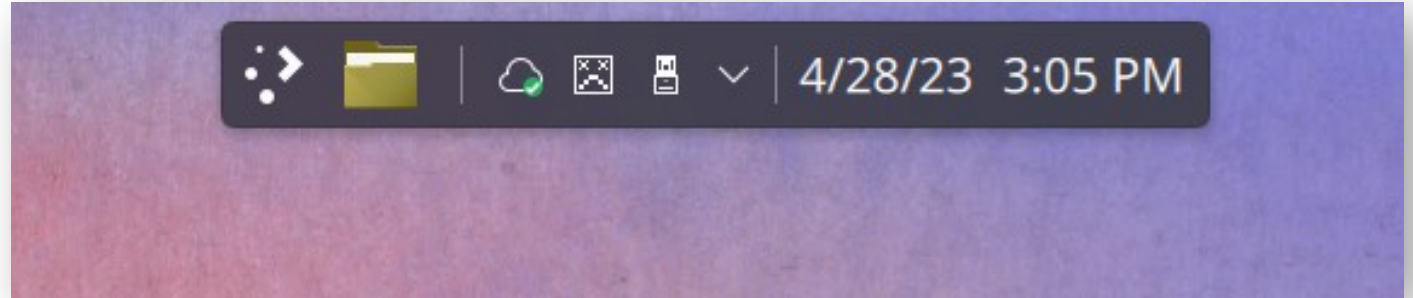
FUN STUFF

COMPLETELY REDESIGNED PANEL SETTINGS



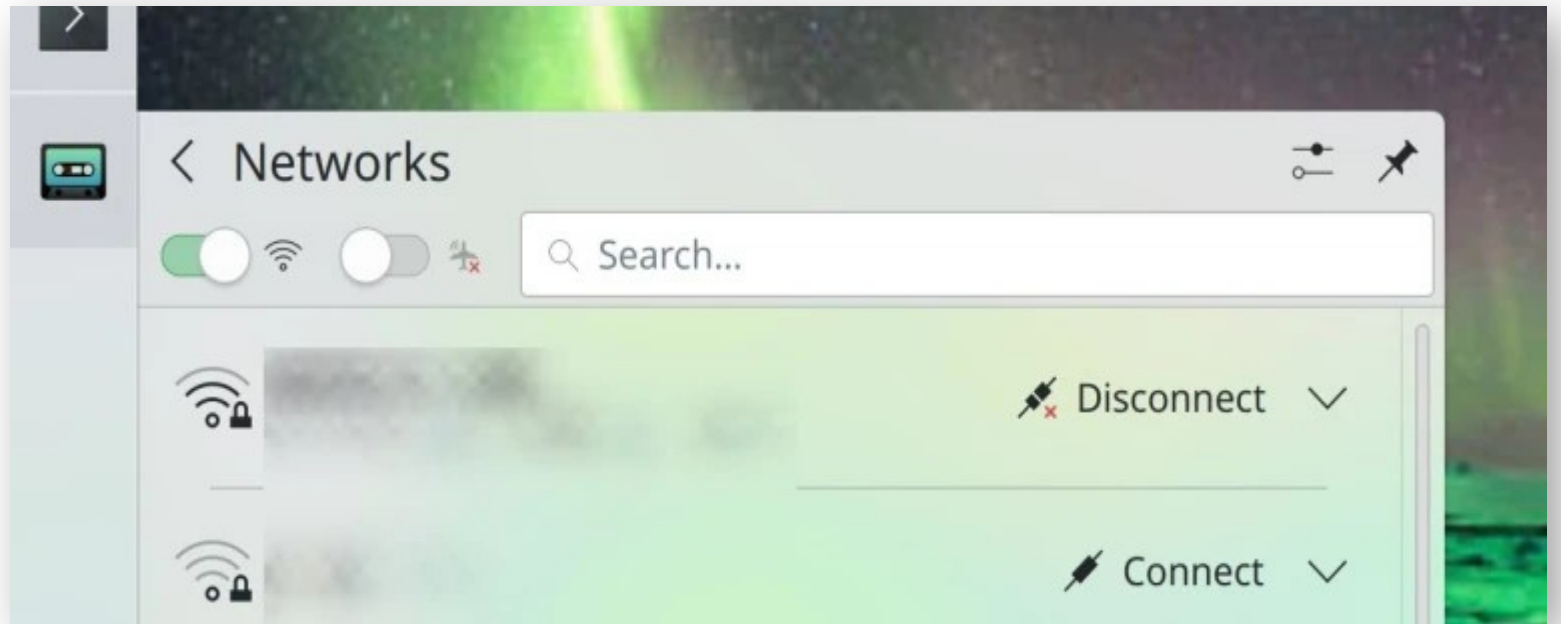
FUN STUFF

REVAMPED (FLOATING) PANELS



FUN STUFF

REVAMPED APPLETS LOOK



FUN STUFF

REDESIGNED TASK SWITCHER

Home — Dolphin

Default: Plasma 6 issues.txt @ ~/Desktop — ...

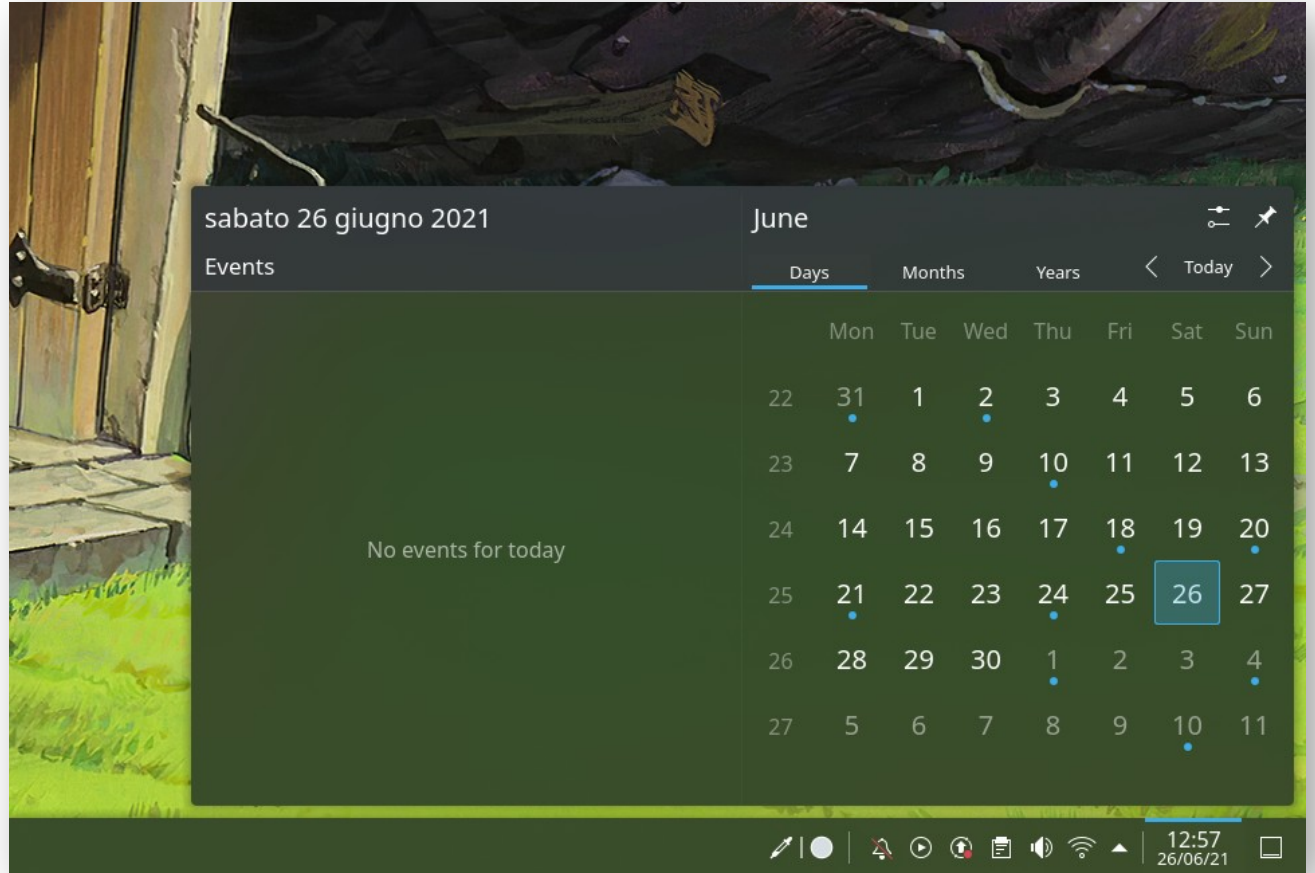
H (She's Not Dead, She Is a Ghost) — Elisa

Inbox - nate@kde.org - Mozilla Thunderbi...

Edit Post < Adventures in Linux and KDE — ...

kio-fuse : elisa — Konsole

FLOATING DIALOGS



**OTHER
STUFF**



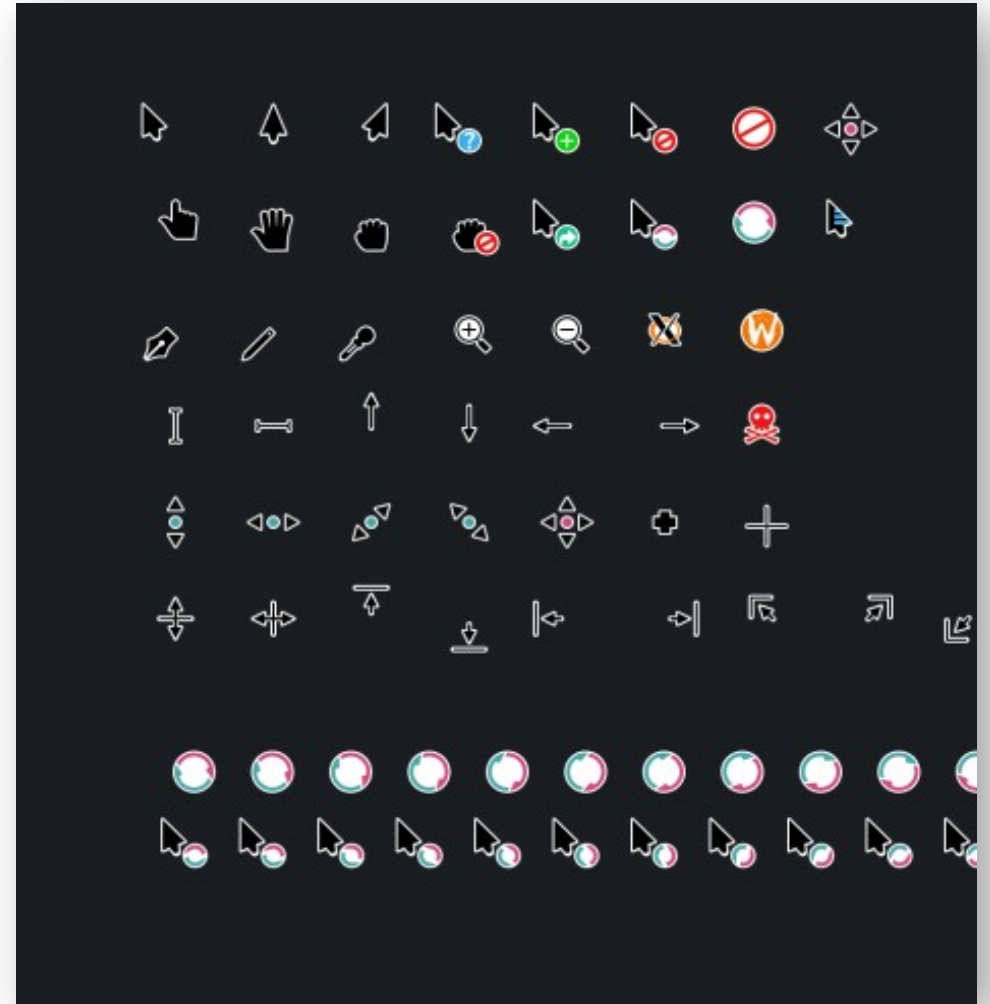
FUN STUFF

REDESIGNED PLACES ICON THEME

















FUN STUFF

REDESIGNED MOUSE ICON THEME



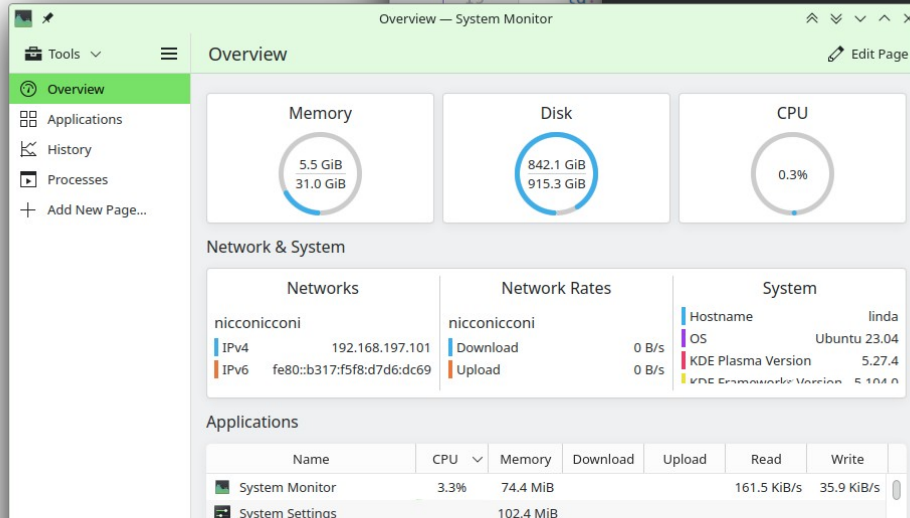
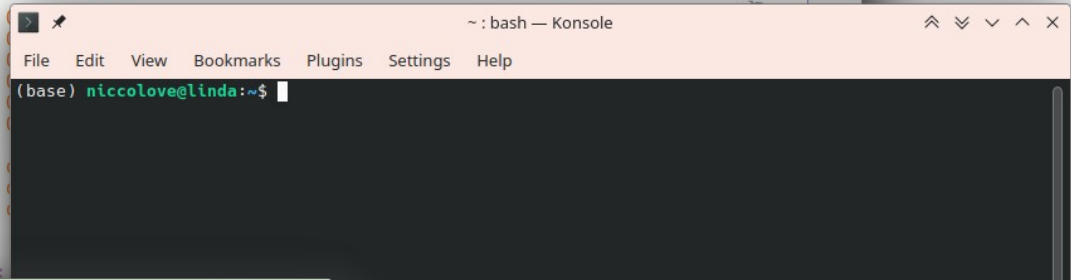
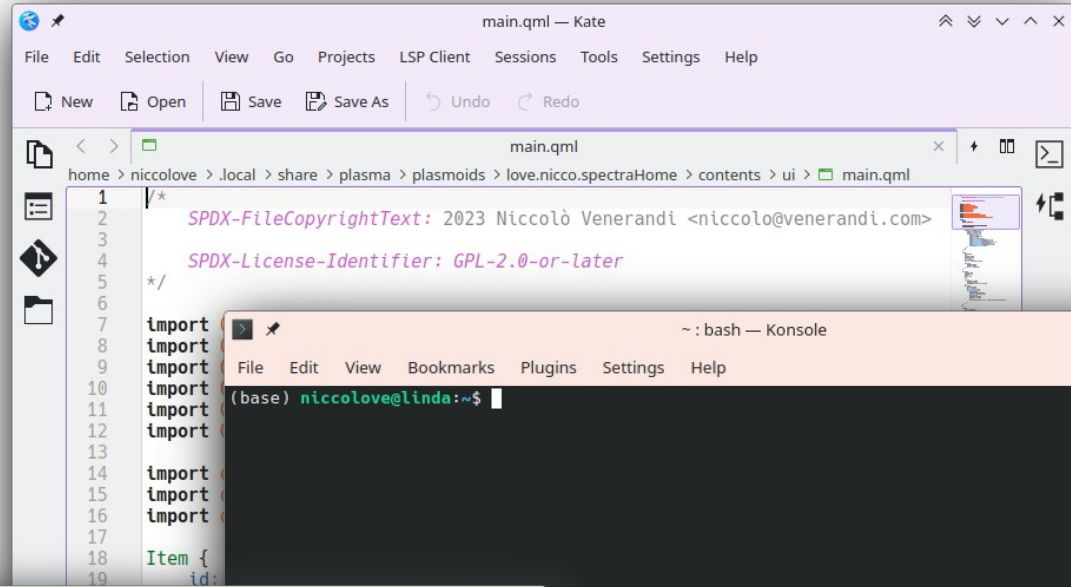
FUN STUFF

REDESIGNED SOUND THEME

Name	Last commit
..	
 audio-volume-change.wav	comply with sound spec folder structure a
 audio-volume-change.wav.license	comply with sound spec folder structure a
 battery-caution.wav	comply with sound spec folder structure a
 battery-caution.wav.license	comply with sound spec folder structure a
 battery-low.wav	comply with sound spec folder structure a
 battery-low.wav.license	comply with sound spec folder structure a
 bell-window-system.wav	Improve bell sound and update project file
 bell-window-system.wav.license	comply with sound spec folder structure a
 completion-fail.wav	comply with sound spec folder structure a
 completion-fail.wav.license	comply with sound spec folder structure a
 completion-partial.wav	comply with sound spec folder structure a
 completion-partial.wav.license	comply with sound spec folder structure a
 completion-rotation.wav	comply with sound spec folder structure a
 completion-rotation.wav.license	comply with sound spec folder structure a

FUN STUFF

COLORFUL WINDOW HEADERS





TECHNICAL DETAILS

- HDR support: games and videos that support HDR will be displayed so on monitors that support it
- Compositor restart: if the wayland compositor crashes and restarts, the applications survive
- New workspace/activities support: ext-workspace protocol which allows to tie virtual desktops to activities

- Was a good api for the old KDE4 imperative javascript plasmoids
- In QML plasmoids is better to use qml types which export the required properties
- Dataengine infra moved to a Plasma5Support lib just as a porting aid, in prevision to be removed completely

- Plasma::Svg and Plasma::FrameSvg were very useful
- Applications that wanted to use them had dependency problems
- All got split in a new framework: KSvg

KSVG

```
import org.kde.plasma 2.0 as PlasmaCore

PlasmaCore.FrameSvgItem {
    imagePath: "widgets/background"
}

PlasmaCore.SvgItem {
    svg: PlasmaCore.Svg {
        imagePath: "widgets/action"
    }
}
```

```
import org.kde.ksvg 1.0 as KSvg

KSvg.FrameSvgItem {
    imagePath: "widgets/background"
}

KSvg.SvgItem {
    imagePath: "widgets/action"
}
```

- In plasmoids is used as it was, no big api changes
- In normal applications with the class `Ksvg::ImageSet` is possible to define where the “themes” for the app are
- For generic apps there is no constraint on the filesystem structure for the “theme”

- “plasmoid” context property is going away
- “Plasmoid” attached property is the Plasma::Applet instance directly
- Everything “graphical” is PlasmoidItem
- It **must** be the root element of the plasmoid qml

WRITE A NEW PLASMOID

```
Item {  
    Plasmoid.fullRepresentation: Item {...}  
    Plasmoid.title: i18n("foo")  
}
```

```
PlasmoidItem {  
    fullRepresentation: Item {...}  
    Plasmoid.title: i18n("foo")  
}
```


ACTIONS

- Plasmoids can have custom context menu actions
- API used to be very imperative, it was done for the KDE4 Javascript plasmoids (when QML didn't exist yet)
- In Plasma6 has been replaced with a way nicer declarative ui, as follows:

ACTIONS

```
Component.onCompleted: {
    Plasmoid.clearActions()
    Plasmoid.setAction("previous", i18nc("Play previous track", "Previous Track"),
        Qt.application.layoutDirection === Qt.RightToLeft ? "media-skip-forward" : "media-skip-backward");
    Plasmoid.action("previous").enabled = Qt.binding(() => root.canGoPrevious)
    Plasmoid.action("previous").visible = Qt.binding(() => root.canControl)
    Plasmoid.action("previous").priority = Plasmoid.LowPriorityAction
}
function action_previous() {
    serviceOp(mpris2Source.current, "Previous");
}
```

```
Plasmoid.contextualActions: [
    PlasmaCore.Action {
        text: i18nc("Play previous track", "Previous Track")
        icon.name: Qt.application.layoutDirection === Qt.RightToLeft ? "media-skip-forward" : "media-skip-backward"
        priority: Plasmoid.LowPriorityAction
        visible: root.canControl
        enabled: root.canGoPrevious
        onTriggered: serviceOp(mpris2Source.current, "Previous")
    }
]
```

KIRIGAMIFICATION

- When Kirigami was designed, we used concepts we learned from Plasma, and refined them
- Plasma::Theme is an api for colors, and Kirigami::Theme derived from it
- The Kirigami version is much more advanced

COLORS

- Kirigami::Theme is an attached property that “inherits”
- There are color “sets” (window, itemviews, header areas etc)
- Theme.textColor will be from the correct “set”
- If an item is set a set, child items will inherit the set
- Colors can be overwritten

COLORS

- Plasma had a similar thing by combining `Plasma::Theme` and `Plasma::ColorScope`
- In Plasma6 we did away with this duplication, and `Kirigami::Theme` is used in plasmoids instead of plasma classes
- Was not possible in plasma5 as the kirigami version only gave system, not plasma colors

UNITS AND ICONS

- Like theme, we had a duplication in the singleton class Units
- Used for grid units, default layout spacing, animation duration etc
- In Plasma6 only the Kirigami version stays
- Same thing for a component to draw icons away from PlasmaCore.IconItem to Kirigami.Icon



QUESTIONS?