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
COMPLETELY REDESIGNED OVERVIEW
COMPLETELY REDESIGNED GRID VIEW
1:1 TOUCH PAD/SCREEN GESTURES
COMPLETELY REDESIGNED PANEL SETTINGS
REVAMPED (FLOATING) PANELS
REVAMPED APPLETS LOOK

FUN STUFF
REDESIGNED TASK SWITCHER
REDESIGNED PLACES ICON THEME

FUN STUFF
REDESIGNED
MOUSE ICON
THEME
### FUN STUFF

**REDESIGNED SOUND THEME**

```
<table>
<thead>
<tr>
<th>Name</th>
<th>Last commit</th>
</tr>
</thead>
<tbody>
<tr>
<td>..</td>
<td></td>
</tr>
<tr>
<td>audio-volume-change.wav</td>
<td>comply with sound spec folder structure</td>
</tr>
<tr>
<td>audio-volume-change.wav.license</td>
<td>comply with sound spec folder structure</td>
</tr>
<tr>
<td>battery-caution.wav</td>
<td>comply with sound spec folder structure</td>
</tr>
<tr>
<td>battery-caution.wav.license</td>
<td>comply with sound spec folder structure</td>
</tr>
<tr>
<td>battery-low.wav</td>
<td>comply with sound spec folder structure</td>
</tr>
<tr>
<td>battery-low.wav.license</td>
<td>comply with sound spec folder structure</td>
</tr>
<tr>
<td>bell-window-system.wav</td>
<td>improve bell sound and update project file</td>
</tr>
<tr>
<td>bell-window-system.wav.license</td>
<td>improve bell sound and update project file</td>
</tr>
<tr>
<td>completion-fail.wav</td>
<td>comply with sound spec folder structure</td>
</tr>
<tr>
<td>completion-fail.wav.license</td>
<td>comply with sound spec folder structure</td>
</tr>
<tr>
<td>completion-partial.wav</td>
<td>comply with sound spec folder structure</td>
</tr>
<tr>
<td>completion-partial.wav.license</td>
<td>comply with sound spec folder structure</td>
</tr>
<tr>
<td>completion-rotation.wav</td>
<td>comply with sound spec folder structure</td>
</tr>
<tr>
<td>completion-rotation.wav.license</td>
<td>comply with sound spec folder structure</td>
</tr>
</tbody>
</table>
```
COLORFUL WINDOW HEADERS

FUN STUFF
• 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
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
• Everyhitng “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")
}
- 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:
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.
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
• 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
• 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