

History

- xpdf (1995); Poppler and Qt binding (2005)
- hackathon in Dec 2014: Qt wrapper for PDFium (working with Qt 5.4)
- QtPDF available since ~ Qt 5.10 (widgets only)
- https://www.qt.io/blog/2017/01/30/new-qtpdf-qtlabs-module
- build system change (gyp to gn)
- source is now in qtwebengine/src/pdf
- https://www.youtube.com/watch?v=y6-Khqrrz_U Qt Virtual Tech Con 2020
- Tech Preview in 6.3 (mostly the same as in 5.15)
- fully supported and more new features in 6.4

Goals

- make it easy to write a QQ PDF viewer with the usual feature set
- license OK for commercial users
- up-to-date
- QTBUG-77503 (the 'epic' feature list)
- component-based: *not* a whole viewer in an Item subclass
- push the limits with pre-existing Qt Quick components
- enable use of PDF as another scalable image format

22:41 #3/

```
import QtQuick 2.14
                               Simplest possible multi-page viewer
Image {
   id: image
   source: "test.pdf"
   fillMode: Image.PreserveAspectFit
   Shortcut {
          sequence: StandardKey.MoveToNextPage
        sequence: "Ctrl+D"
        enabled: image.currentFrame < image.frameCount - 1</pre>
        onActivated: image.currentFrame++
   Shortcut {
          sequence: StandardKey.MoveToPreviousPage
        sequence: "Ctrl+S"
        enabled: image.currentFrame > 0
        onActivated: image.currentFrame--
```



22:41 # 4 / 13

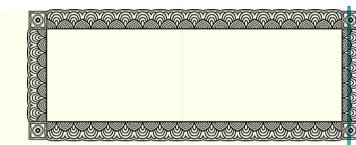
```
import OtQuick
                                                                                           Getting real
import OtOuick.Controls
                                                                                        1. Get started
                                                                                                              nexus
                                                                                        2. Play & explore
import QtQuick.Dialogs
                                                                                        3. Make yourself at home
import QtQuick.Layouts
                                                                                        4. Make Search personal
                                                                                        5. Tune performance
import QtQuick.Pdf
                                                                                        6. Enter & edit text
                                                                                        7. Connect to networks
                                                                                        8. Connect to devices
                                                                                        9. Manage accounts
Rectangle { // ApplicationWindow IRL
                                                                                        10. Secure your tablet
                                                                                        11. Accessibility
     id: root; width: 600; height: 800; color: "lightgrey"
                                                                                        Appendix: Hardware reference
                                                                                                              Guidebook
     PdfDocument {
          id: document
         source: Qt.resolvedUrl("nexus7-manual-page-numbers.pdf")
         onStatusChanged: (status) => {
              // if (status === PdfDocument.Error) errorDialog.open()
              view.document = (status === PdfDocument.Ready ? document
          // onPasswordRequired: passwordDialog.open()
     SplitView {
         anchors.fill: parent; anchors.topMargin: header.height; ancho

y page i : 0 of 106 scale 1.00

          Item {
              SplitView.preferredWidth: sidebarOpenAction.checked ? 250 : 0
              visible: sidebarOpenAction.checked
               TabBar {
                   id: sidebarTabs
                   x: -width
 22:41
                                                                                                                   #5/13
                   rotation: -90
```

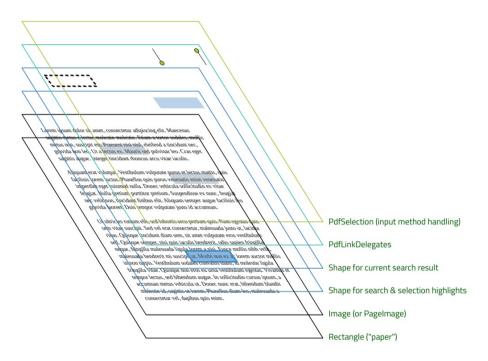
```
import QtQuick 2.14
BorderImage {
    source: "resources/border.pdf"
   border.top: 37
   border.bottom: 37
   border.left: 37
   border.right: 37
   horizontalTileMode: BorderImage.Round
   verticalTileMode: BorderImage.Round
    SequentialAnimation on height {
        loops: Animation. Infinite
        PauseAnimation {
            duration: 500
        NumberAnimation {
            duration: 2300
            to: 700
            easing.type: Easing.InElastic
        NumberAnimation {
            duration: 2200
            to: 400
            easing.type: Easing.InElastic
22:41SequentialAnimation on width {
        loops: Animation. Infinite
```

Getting silly

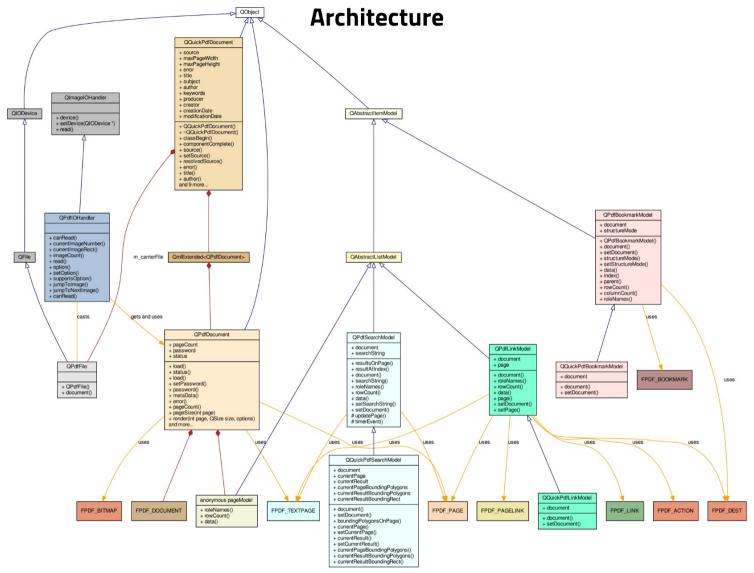


#6/13

Modularity



22:41 # 7 / 13



#8/13 22:41

New stuff in 6.4

- bookmarks
- arbitrary page numbers
- thumbnails
- sharing document instance with PDF image plugin
- password-protected PDFs
- API cleanup
- widget improvements (more coming in 6.5, probably)

22:41 #9/

Stuff left to work on

- image cache improvements in Qt Quick
- Android support
- tiled rendering?
- annotations?

22:41 # 10 / 13

Stuff the community could help with

XFA?

• use Poppler in Qt Quick, same API?

• ?

22:41 # 11 / 13

```
import QtQuick Primitive vector elements in the scene graph (SVG, could be PDF?)
import QtQuick.Svg // experimental hackathon API: not released
                                                                                 Rectangle
Item {
   width: 800; height: 800
   layer.enabled: true
   layer.samples: 8
   SvqView { // experimental hackathon API: not released
       x: 600; y: 300; scale: 2
       source: Application.arguments.length > 2 ?
                   Application.arguments[Application.arguments.length - 1] : Qt.resolvedUrl("primile
       observesViewport: false
       DragHandler { }
       WheelHandler { property: "scale" }
       PinchHandler { }
```

22:41 # 12 / 13



22:41 # 13 / 13