

Elvis Angelaccio

Sviluppatore software

Panoramica Librerie KDE

Develer webinar

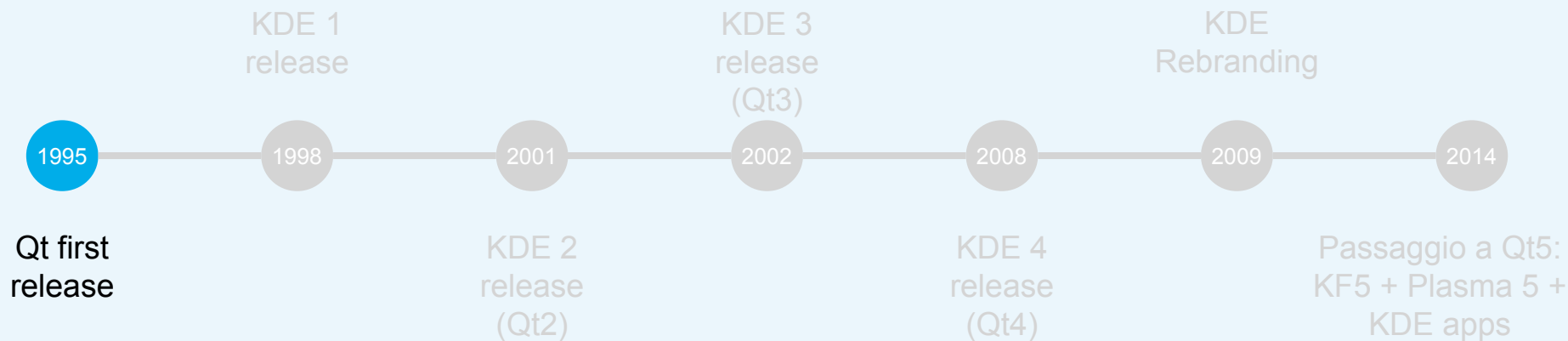
08/06/2022

develer

I COSA VEDREMO

- Introduzione storica e motivazioni
- Licenze disponibili
- Architettura delle librerie
- Come integrare una libreria in un'applicazione C++
- Principali librerie

Un po' di storia



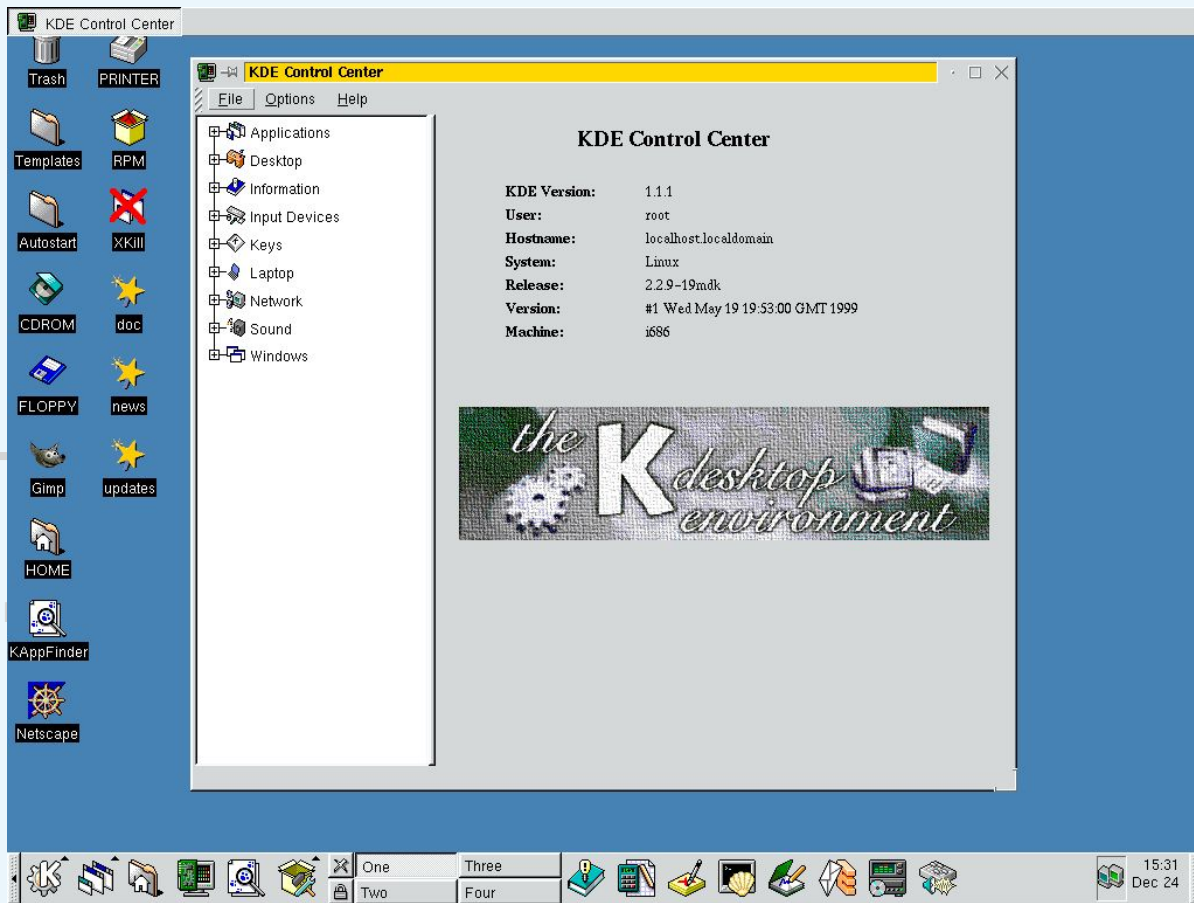
Un po' di storia

KDE 1
release

1995

1998

Qt first
release



Un po' di storia

KDE 1
release

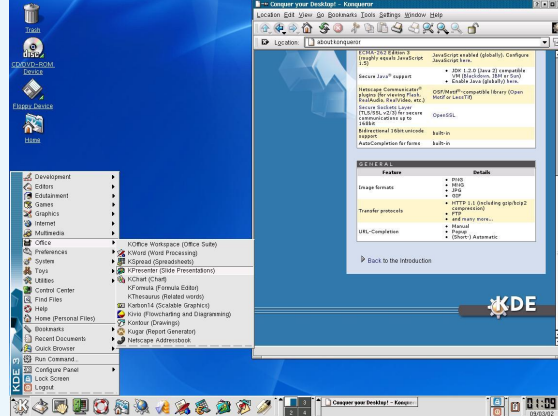
1995

1998

2001

KDE 3
release
(Qt3)

2002

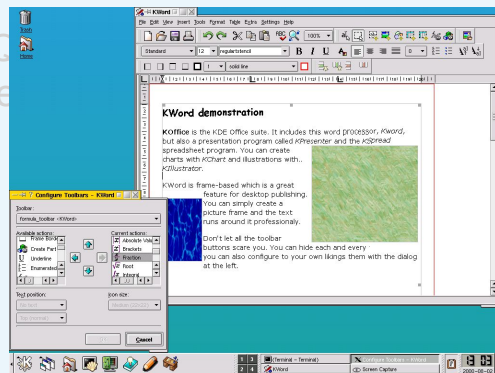


2008

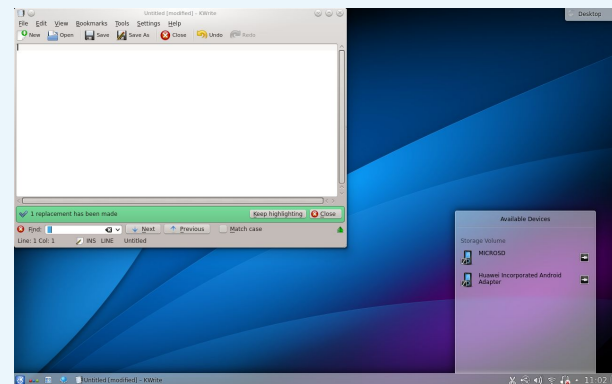
KDE 4
release
(Qt4)

2009

2014



KDE 2
release
(Qt2)



Un po' di storia

Repositioning the KDE Brand

KDE has changed over the past 13 years. The application framework has grown, matured and gone cross-platform, as have the applications. Strong growth in our community has created an increasingly diverse and large set of high-quality applications.

In the process, KDE's identity has shifted from being simply a desktop environment to representing a global community that creates a remarkably rich body of free software targeted for use by people everywhere.

***KDE is no longer software created by people,
but people who create software.***

KDE
Rebranding

1995

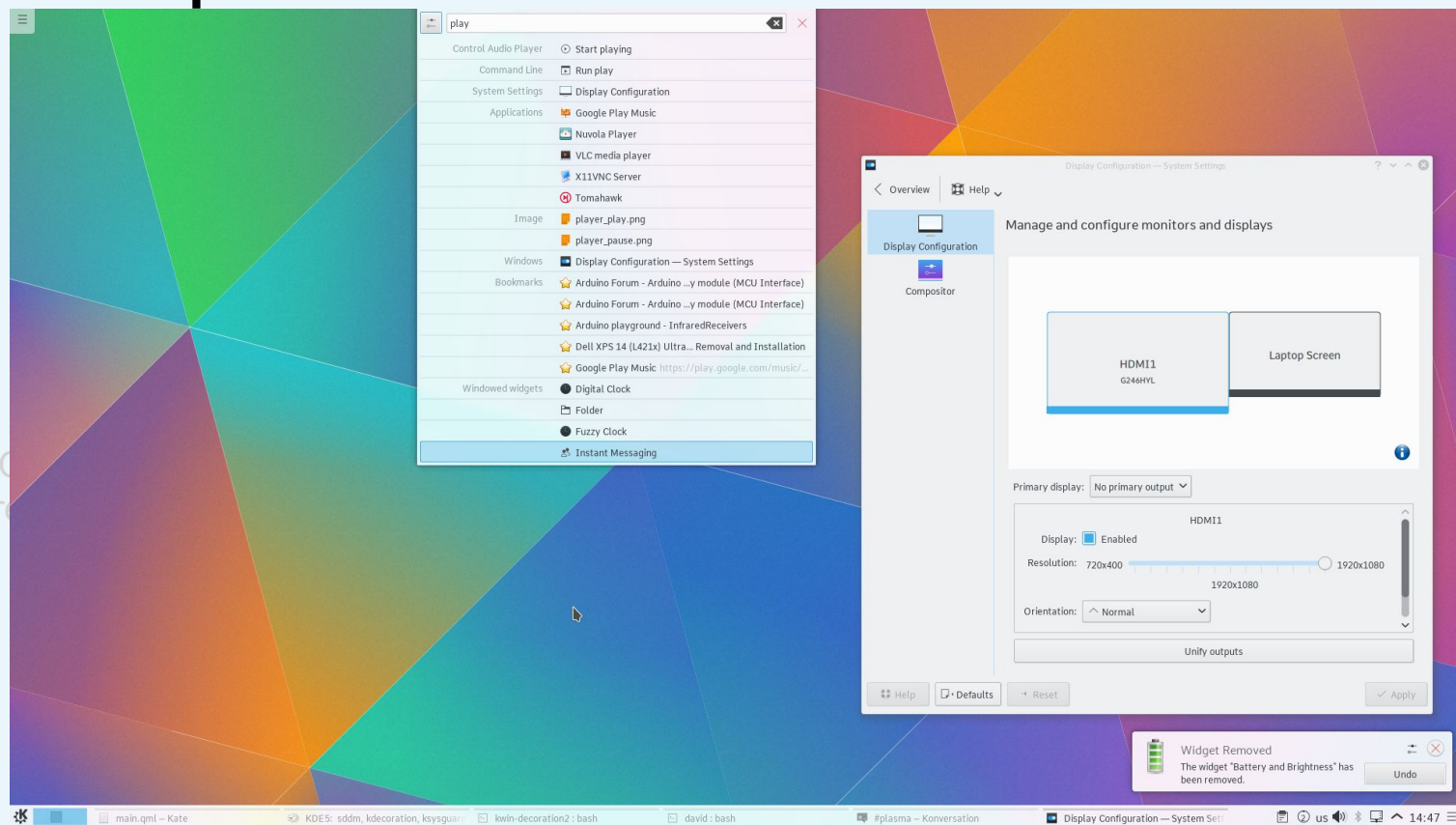
Qt first
release

2009

2014

Passaggio a Qt5:
KF5 + Plasma 5 +
KDE apps

Un po' di storia



2014

Passaggio a Qt5:
KF5 + Plasma 5 +
KDE apps

Librerie KDE 4: kdelibs4

Problematiche in kdelibs4:

- Libreria monolitica
- Interdipendenze ovunque
- Difficile da mantenere
- Difficile distribuire singole applicazioni fuori da Linux

Passaggio a Qt5: nascono i “KDE Frameworks” (KF5)

Split di kdelibs4:

1. Upstream funzionalità in Qt5
2. Divisione in oltre 50 librerie separate
3. Dipendenze tra librerie ben definite
4. SO supportati ben definiti

KDE Frameworks 5 (KF5)

- ABI stabile (come in Qt)
- Rilasci mensili (versione attuale: 5.93)
- Attualmente 83 librerie
- Piattaforme “target”:
 - Linux
 - FreeBSD
 - Windows
 - macOS
 - Android
 - iOS

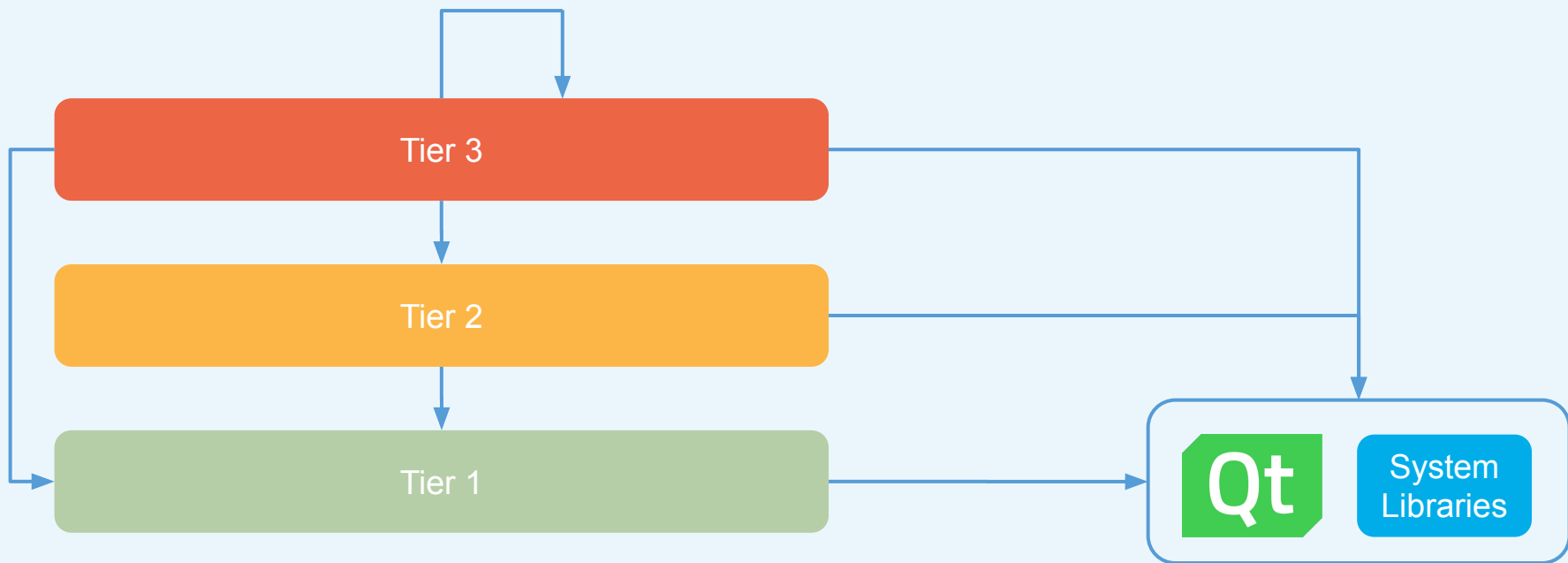
Licenze Disponibili

- LGPL in maggioranza
- Alcune librerie usano MIT o BSD (es. KSyntaxHighlighting)

Architettura KF5

- Ogni libreria ha un *Tier* e un *Type* ben definiti
- Tier e type impongono dei limiti sulle dipendenze

I Tiers



I Types

- **Functional** → no runtime dependencies
- **Integration** → può avere runtime dependencies opzionali
- **Solutions** → richiede runtime dependencies obbligatorie

Come utilizzare un framework in un programma Qt

- CMake o qmake

Esempio CMake minimale

```
project(kf5demo)
set(CMAKE_AUTOMOC ON)

find_package(Qt5 REQUIRED COMPONENTS Core Widgets)
find_package(KF5CoreAddons)
find_package(KF5WidgetsAddons)

add_executable(kf5demo main.cpp mainwindow.cpp)
target_link_libraries(kf5demo PRIVATE Qt5::Core Qt5::Widgets KF5::CoreAddons KF5::WidgetsAddons)
```


Esempio CMake minimale

```
project(kf5demo)
set(CMAKE_AUTOMOC ON)

find_package(Qt5 REQUIRED COMPONENTS Core Widgets)
find_package(KF5CoreAddons)
find_package(KF5WidgetsAddons)

add_executable(kf5demo main.cpp mainwindow.cpp)
target_link_libraries(kf5demo PRIVATE Qt5::Core Qt5::Widgets KF5::CoreAddons KF5::WidgetsAddons)
```

Esempio con extra-cmake-modules (ECM)

```
project(kf5demo)
set(CMAKE_AUTOMOC ON)
```

```
find_package(ECM REQUIRED NO_MODULE)
set(CMAKE_MODULE_PATH ${ECM_MODULE_PATH})
```

ECM: collezione di utilità
cmake (tier 1)

```
find_package(Qt5 REQUIRED COMPONENTS Core Widgets)
```

```
find_package(KF5 REQUIRED COMPONENTS CoreAddons WidgetsAddons)
```

```
add_executable(kf5demo main.cpp mainwindow.cpp)
```

```
target_link_libraries(kf5demo PRIVATE Qt5::Core Qt5::Widgets KF5::CoreAddons KF5::WidgetsAddons)
```

qmake

```
QT += KCoreAddons  
QT += KWidgetsAddons
```

I TIER 1

Tier 1 – Principali Librerie

KCoreAddons

Addons per QtCore

- Classi di utilità strettamente non-GUI

Tier: 1

Type: Functional

Platforms:     

KCoreAddons: KDirWatch

- Permette di essere notificati quando un file o una directory viene creata/modificata
- Rispetto a QFileSystemWatcher, funziona anche su path che (ancora) non esistono

I KDirWatch: esempio di utilizzo

```
auto dirWatch = new KDirWatch(this);
connect(dirWatch, &KDirWatch::dirty, this, [=]{ /* do something */ });
connect(dirWatch, &KDirWatch::created, this, [=]{ /* do something */ });
connect(dirWatch, &KDirWatch::deleted, this, [=]{ /* do something */ });
dirWatch->addDir(someDir);
dirWatch->addFile(someFile);
```

KCoreAddons: KJob

- Permette di lanciare un “job” ed essere notificati quando termina
- Job eseguibili in asincrono (“fire and forget”) oppure in sincrono

I KJob: esempio di utilizzo async

```
void SomeClass::doSomething()
{
    KJob* job = new SomeJob(...); // SomeJob subclass of KJob
    connect(job, &KJob::result, this, &SomeClass::handleResult);
    job->start();
}

void SomeClass::handleResult(KJob *job)
{
    if (job->error()) {
        handleError();
    }

    ...
}
```

I KJob::start() implementation

```
void SomeJob::start()
{
    // Actual work must be triggered by the event loop to get async behavior.
    QTimer::singleShot(0, this, &SomeJob::doWork);
}

void SomeJob::doWork()
{
    // Actual work of the job. If the job is "slow", should be done in another thread or process.
    ...
}
```

I KJob: esempio di utilizzo bloccante

```
void SomeClass::doSomething()
{
    KJob* job = new SomeJob(...); // SomeJob subclass of KJob
    connect(job, &KJob::result, this, &SomeClass::handleResult);
    if (!job->exec()) {
        handleError();
    } else {
        // All good
    }
}
```

KWidgetsAddons

Addons per QtWidgets

- Collezione di svariati QWidget custom.

Tier: 1

Type: Functional

Platforms:     

KWidgetsAddons: KSqueezedTextLabel

- QLabel con supporto per elisione del testo

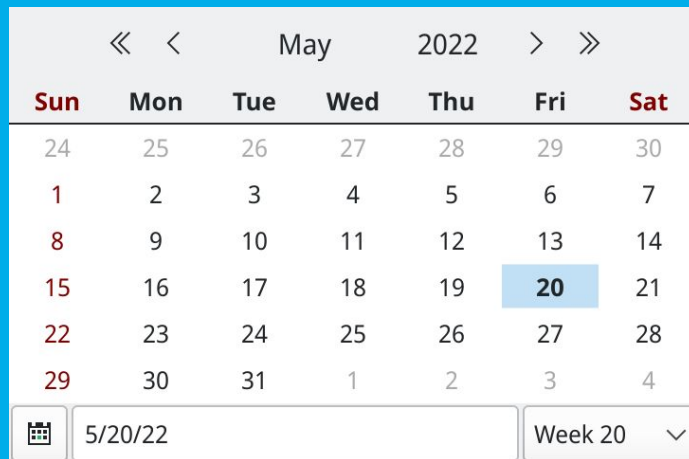
A, B, C, D, E, F, G, ... T, U, V, W, X, Y, Z.

A, B, C, D, E, F, G, H, I, J, K, L, M, N, O...

..., N, O, P, Q, R, S, T, U, V, W, X, Y, Z.

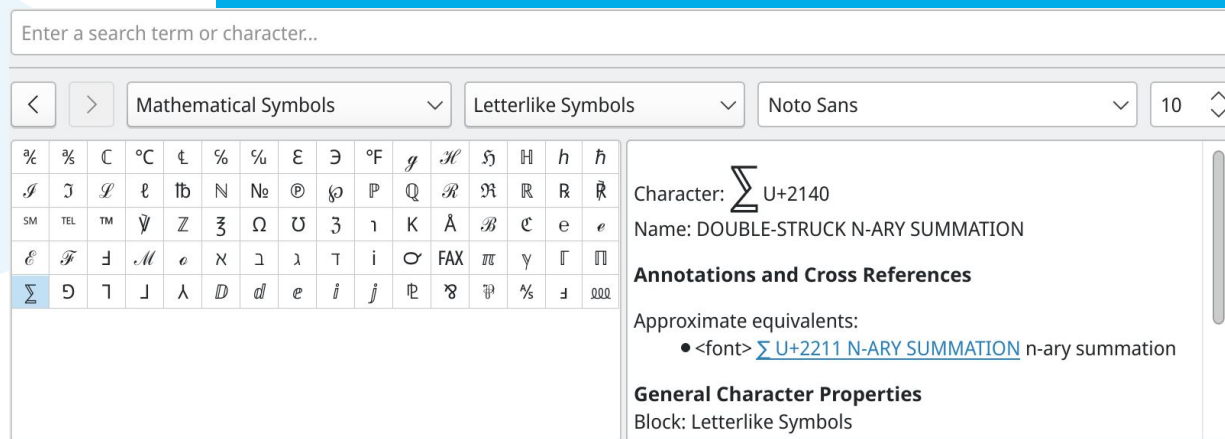
KWidgetsAddons: KDatePicker

- Widget per selezione data



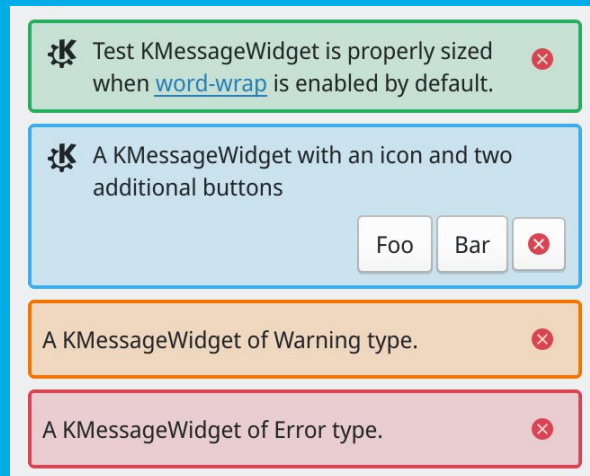
KWidgetsAddons: KCharSelect

- Widget per selezione di caratteri Unicode



KWidgetsAddons: KMessageWidget

- Fornisce l'equivalente di un QMessageBox "inline"



KWidgetsAddons: KCollapsibleGroupBox

- Groupbox “richiudibile”

- > Advanced Options
- > Pick a date

▼ Advanced Options

- ☐ Some text
- ☐ Some text
- ☐ Some text
- ☐ Some text
- ☐ Some text
- ☐ Some text
- ☐ Some really long text that goes on and on and on for ever and ever

Some input field:

- > Pick a date

> Advanced Options

▼ Pick a date

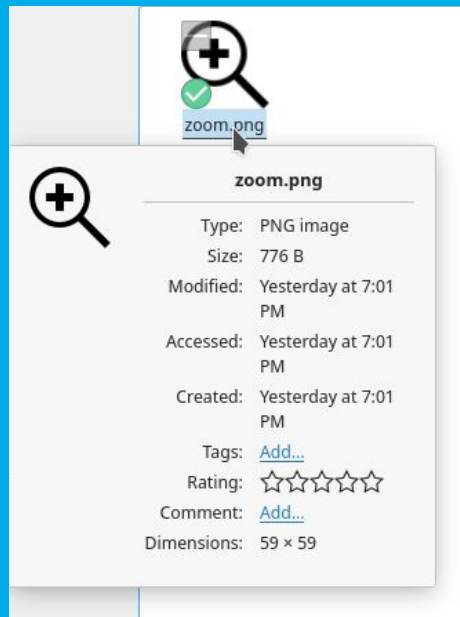
« < May 2022 > »

Sun	Mon	Tue	Wed	Thu	Fri	Sat
24	25	26	27	28	29	30
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

📅 5/19/22 Week 20 ▼

KWidgetsAddons: KTooltipWidget

- Un “tooltip” che può contenere un QWidget



KArchive

Tier: 1

Type: Functional

Platforms:     

Macro-funzionalità:

1. Permette una compressione/decompressione trasparente tramite un QIODevice
2. Permette di leggere/scrivere archivi in formato Zip, 7zip, Tar

■ Compressione trasparente: KCompressionDevice

```
void SomeClass::compressData()
{
    QByteArray data = "This is some data that will be compressed.\n";

    QBuffer buffer;
    KCompressionDevice compressionDevice(&buffer, false, KCompressionDevice::GZip);
    if (compressionDevice.open(QIODevice::WriteOnly)) {
        compressionDevice.write(data);
    }
}
```

I Estrarre un archivio ZIP

```
void SomeClass::extractZip()
{
    KArchive *archive = new KZip("~/test.zip");
    if (archive->open(QIODevice::ReadOnly)) {
        const KArchiveDirectory *rootDir = archive->directory();
        QTemporaryDir destinationDir;
        rootDir->copyTo(destinationDir.path(), true); // true = recursively extract all entries
    }
}
```

Prison

**Libreria per
generare e
scansionare
"barcode" 1D e 2D**

Tier: 1

Type: Functional

Platforms:     

- Formati supportati:
 - Barcode 1D
 - QRCode (richiede qrencode)
 - DataMatrix (richiede libdmtx)
 - Aztec
 - PDF147 (richiede zxing-c++)

■ Esempio generazione QRCode (C++)

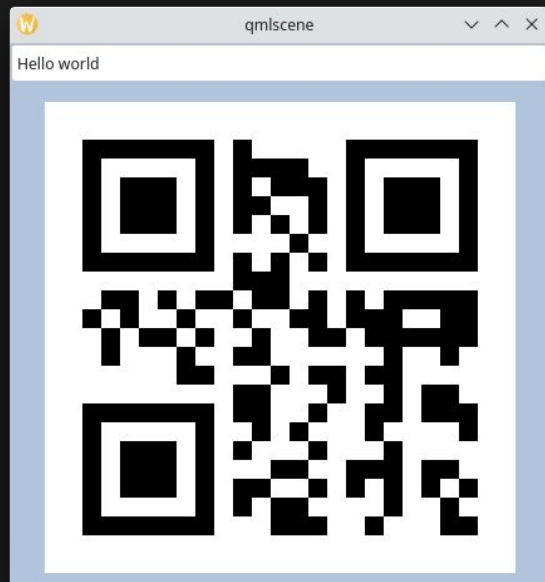
```
void SomeClass::generateQrCode()
{
    QString text = "Hello world";

    std::unique_ptr<Prison::AbstractBarcode> qrcode(Prison::createBarcode(Prison::QRCode));
    qrcode->setData(text);
    QImage image = qrcode->toImage(qrcode->preferredSize(qApp->devicePixelRatio()));

    // do something with the image...
}
```

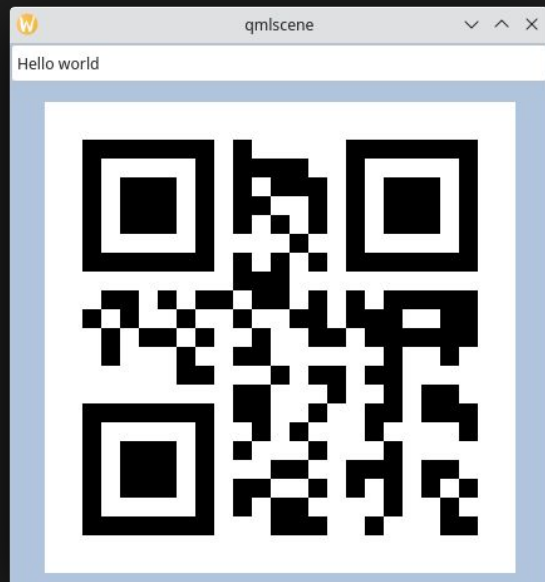
Esempio generazione QRCode (QML)

```
TextField {  
    id: contentEdit  
    Layout.fillWidth: true  
    text: "Hello world"  
}  
  
Prison.Barcode {  
    id: barcode  
    Layout.fillWidth: true  
    Layout.fillHeight: true  
    content: contentEdit.text  
    barcodeType: "QRCode"  
}
```



Esempio generazione QRCode (QML)

```
TextField {  
    id: contentEdit  
    Layout.fillWidth: true  
    text: "Hello world"  
}  
  
Prison.Barcode {  
    id: barcode  
    Layout.fillWidth: true  
    Layout.fillHeight: true  
    content: contentEdit.text  
    barcodeType: "QRCode"  
}
```



I Esempio scansione (QML/Qt5)

```
VideoOutput {
    id: viewFinder
    anchors.fill: parent
    source: camera
    filters: [scanner]
}
Prison.VideoScanner {
    id: scanner
    onResultChanged: {
        if (result.hasText) console.log(result.text, result.format);
        else if (result.hasBinaryData) console.log("<binary content>", result.format);
    }
}
Camera {
    id: camera
}
```

I Esempio scansione (QML/Qt5)

```
VideoOutput {
    id: viewFinder
    anchors.fill: parent
    source: camera
    filters: [scanner]
}
Prison.VideoScanner {
    id: scanner
    onResultChanged: {
        if (result.hasText) console.log(result.text, result.format);
        else if (result.hasBinaryData) console.log("<binary content>", result.format);
    }
}
Camera {
    id: camera
}
```

Kirigami

**Collezione di
componenti
QtQuick**

Tier: 1

Type: Functional

Platforms:      

- Permette di creare applicazioni “convergenti”
- Layout e componenti responsive in base al tipo di piattaforma (desktop/mobile)

Esempio: Form Layout – Android vs Desktop

10:09 32%

Gallery > Form Layout

Label: _____

Loger label: _____

After separator: _____

Combo: First ▾

☒ Option

Section title

Label: _____

Section without line: _____

Section with title without line

Title: _____

Label for radios: ☒ One

Form Layout Info

Label: _____

Longer label: _____

After separator: _____

Combo: First ▾

☒ Option

Section title

Label: _____

Section without line: _____

Expanding Button

Section with title without line

Title: _____

☐ _____

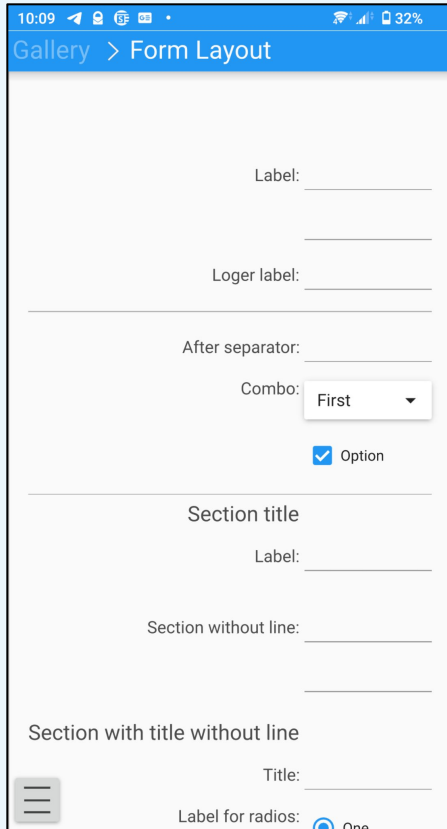
☒ One

☐ Two

Label for radios: ☐ Three

Add Field

Esempio: Form Layout – Android vs Desktop



The screenshot shows the Kirigami Gallery application on an Android device. The status bar at the top displays the time 10:09, signal strength, Wi-Fi, and battery level at 32%. The app's header bar is blue with the text "Gallery > Form Layout". The form layout consists of several sections: a "Label:" section with a text input field; a "Logger label:" section with a text input field; an "After separator:" section with a text input field; a "Combo:" section with a dropdown menu showing "First" and a checked "Option" checkbox; a "Section title" section with a text input field; a "Label:" section with a text input field; a "Section without line:" section with a text input field; a "Section with title without line" section with a text input field; and a "Title:" section with a text input field. At the bottom, there is a "Label for radios:" section with a radio button and the text "One".

Kirigami Gallery
Disponibile su play store.

TIER 1: altri Framework interessanti

- KConfig → gestione strutturata di file di configurazione
- Solid → hardware detection
- Sonnet → spell checking

I TIER 2

Tier 2 – Principali Librerie

KNotifications

**Libreria che astrae
le “notifiche di
sistema”**

Tier: 2

Type: Solution

Platforms:     

- Permette di eseguire una notifica nativa in modo trasparente
- Permette di reagire all'attivazione della notifica (click o tocco dell'utente)

I Esempio minimale di creazione notifica

```
void ChatClient::messageArrived(QString message)
{
    KNotification *notification = new KNotification("messageArrived");
    notification->setText(message);
    notification->setPixmap(...);

    connect(notification, &KNotification::defaultActivated, this, [=]{ /* do something */ });
    // Show the notification!
    notification->sendEvent();
}
```

■ TIER 2: altri Framework interessanti

- KCompletion → widget per “text completion”
- KFileMetadata → estrazione metadati da file
- KImageFormats → plugin per vari formati immagine (es. xcf)

I TIER 3

Tier 3 – Principali Librerie

KIO

I/O “Network Transparent” e file management

Tier: 3

Type: Solutions

Platforms:     

- Permette di copiare file/dati facilmente tra “protocolli” diversi
- Protocolli supportati:
 - FTP, HTTP/Webdav
- Plugin opzionali per ulteriori protocolli:
 - SFTP, SMB, MTP, NFS

KIO

- KIO fornisce un API simile ai comandi UNIX cp, mv, ecc.
- Le operazioni di I/O avvengono in asincrono in processi dedicati (detti “kioslave”)

Esempio: KIO::copy

```
void SomeClass::copy()
{
    QUrl source("file:///home/elvis/foo.txt");
    QUrl dest("ftp:///some/destination/dir");
    KIO::CopyJob* job = KIO::copy(source, dest, KIO::DefaultFlags);
    connect(job, &KJob::result, this, &SomeClass::handleResult);
    // NOTE: all KIO jobs start automatically, so no need to call job->start()
}

void SomeClass::handleResult(KJob *job)
{
    if (job->error()) {
        handleError();
    }
    ...
}
```

Esempio: KIO::copy

```
void SomeClass::copy()
{
    QUrl source("file:///home/elvis/foo.txt");
    QUrl dest("ftp:///some/destination/dir");
    KIO::CopyJob* job = KIO::copy(source, dest, KIO::DefaultFlags);
    connect(job, &KJob::result, this, &SomeClass::handleResult);
    // NOTE: all KIO jobs start automatically, so no need to call job->start()
}

void SomeClass::handleResult(KJob *job)
{
    if (job->error()) {
        handleError();
    }
    ...
}
```


I Principali KIO jobs

- `KIO::copy()`
- `KIO::move()`
- `KIO::del()`
- `KIO::chmod()`

Job “informativi”:

- `KIO::stat()`
- `KIO::listDir()`
- `KIO::directorySize()`

TIER 3: altri Framework interessanti

- KTextEditor → text editor embeddabile
- KXmlGui → QMainWindow configurabile con drag&drop

Link Utili

api.kde.org/frameworks

Elvis Angelaccio

elvis@develer.com

Vuoi rimanere aggiornato sugli eventi Develer?
Seguici nei nostri canali social:



develer

www.develer.com