

QtRecorder en Velneo

Me he tomado un descanso en los últimos días, los anteriores artículos fueron realmente intensos y necesitaba un relax. El trabajo que supone hacer todo el proceso para **añadir un nuevo plugin QML es intenso**, y lo cierto es que las dos últimas semanas no hemos parado.

Por eso he elegido esta vez, un plugin light. Por lo menos esa era la teoría.

He partido de **QML Recorder** (<https://github.com/sverzeznassi/qml-recorder>).

Este plugin QML, esencialmente nos permite grabar audio desde QML en distintos formatos y fuentes.

sverzeznassi/qml-recorder is licensed under the **GNU General Public License v3.0**

Permissions of this strong copyleft license are conditioned on making available complete source code of licensed works and modifications, which include larger works using a licensed work, under the same license. Copyright and license notices must be preserved. Contributors provide an express grant of patent rights.

This is not legal advice. [Learn more about repository licenses.](#)

Permissions	Limitations	Conditions
✓ Commercial use	✗ Liability	① License and copyright notice
✓ Modification	✗ Warranty	① State changes
✓ Distribution		① Disclose source
✓ Patent use		① Same license
✓ Private use		

Como a priori parecía muy simple, compilé el plugin e importé los ejemplos a Velneo. El resultado... no funcionaba.

Revisando código, encontré un par de cosas que podían ser las culpables (tratamiento de las rutas y otro detalle menor). Corregidas, compiladas y a funcionar.

Parecía que ya estaba yo contento... hasta que me dio por leer. Terminé leyendo <https://github.com/sverzeznassi/qml-recorder/issues> y me día cuenta de que era una versión anticuada (no mucho la verdad). Navegando... encontré unas fuentes más actualizadas. Tal vez era más correcto partir de ellas y olvidarnos de las antiguas <https://bazaar.launchpad.net/~verzeznassi-stefano/+junk/recorder-test/files>.

Al analizar el código fuente, voy y me encuentro dos problemas .

Primer problema

El código fuente está dividido en dos plugins QML.

~verzegnassi-stefano/+junk/recorder-test

Changes **Files**

~verzegnassi-stefano/+junk/recorder-test : /backend/modules (revision 1)

[view revision](#) [view branch changes](#)

Committer: Stefano Verzeznassi

Date: 2014-09-09 19:00:01 UTC

Revision ID: stefano92.100@gmail.com-20140909190001-vdgadlu5ntks87h7

First commit

Filename	Latest Rev
..	
Recorder	1
StorageManager	1

Loggerhead is a web-based interface for Bazaar branches
Version: 1.18.2, Revision: 18566

Vuelta a empezar... no por favor. No quería hacer nada complicado, tengo que leer mucho más sobre de **Angular 5** y **TypeScript**, tengo que revisar el diseño de Arquitectura de otro mega-proyecto, tengo que ofertar un montaje de un CAS, tengo que evitar los rebotes del 4066.

No tengo tiempo para estas cosas.... pufff... dos plugins en vez de uno... pufff...

Pero no lo pude evitar. No quiero que mis compis **Velneadores** tengan que andar con dos librerías para algo tan sencillo como grabar un audio (conceptualmente están mejor separadas, pero para facilitar el uso, considero que mejor juntas).

El resumen... que he creado un plugin QML que contiene los dos y que hace el trabajo correctamente.

Segundo problema

El ejemplo está desarrollado para Ubuntu.... esto no me lo esperaba.

```
AudioRecorder {
    id: recorder
    input: "default:"
    codec: "audio/PCM"
    quality: 4
    volume: 1

    onRecording: root.lastStatus = "onRecording"
    onResumed: root.lastStatus = "onResumed"
    onStopped: root.lastStatus = "onStopped"
    onErrorChanged: root.lastStatus = "onErrorChanged: " + recorder.errorString
}

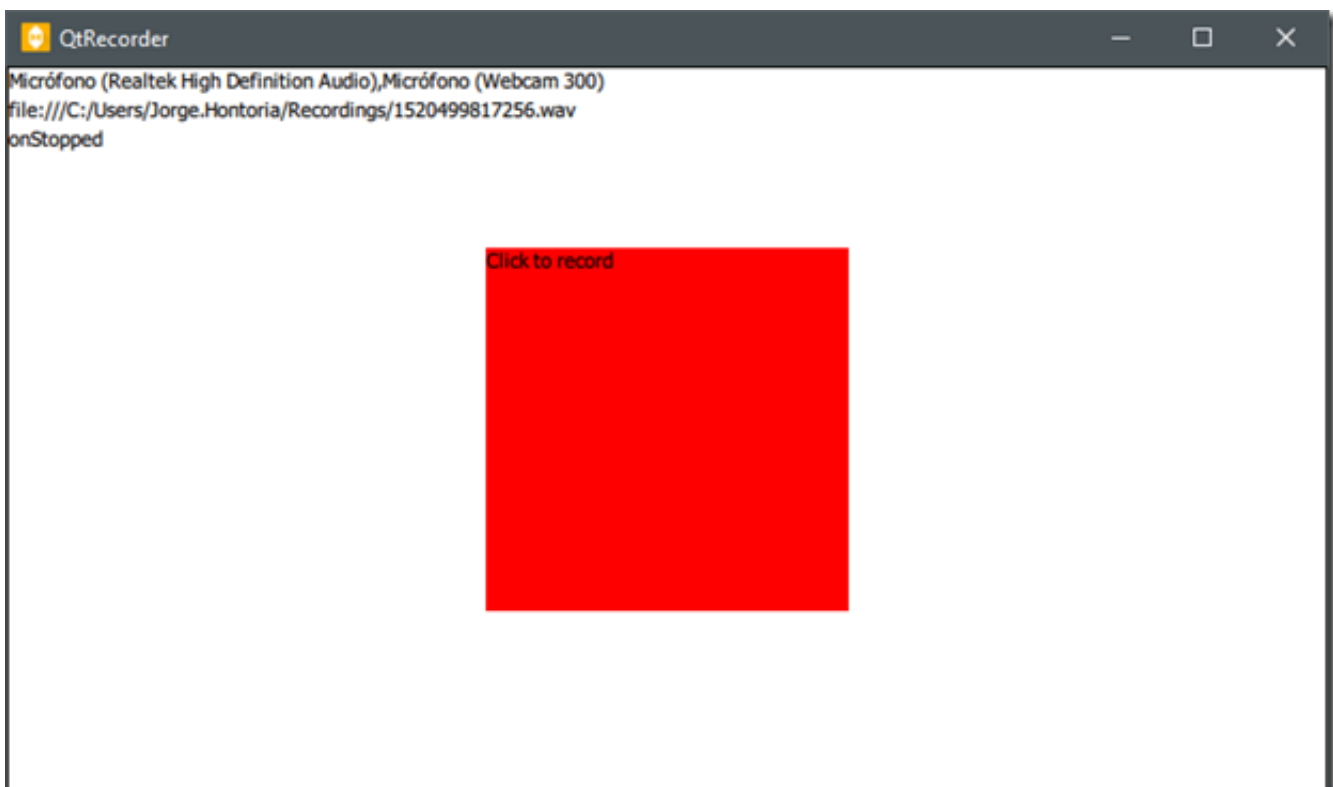
StorageManager {
    id: storage
}

Component.onCompleted: {
    var recsFolder = "file:/// " + storage.getXdgFolder(StorageManager.HomeLocation) + "/Recordings"

    if (!storage.exists(recsFolder))
        storage.mkPath(recsFolder)

    root.recsFolder = recsFolder
}
```

A revisar el ejemplo y retocar las cosas que no son compatibles. Y todo esto para un simple botón que graba audio.



Moralejas

“No es oro todo lo que reluce”, “No hay mal que por bien no venga”, “Lo barato sale caro”, “El que corre pronto para”... aplicaros el que más os guste. A mi me da, que aplican todos y alguno más en este caso.

QML es dado a la fragmentación por culpa de los fabricantes (como siempre):

- BB10
- Ubuntu
- V-Play
- Jolla
- Nemo
- Qt5
- ...

Muchas formas de hacer lo mismo, un sinsentido para un framework tan genial como Qt.

Descargas

Bueno, es hora de publicarlo. Funcionalmente es una librería simple, pero que resuelve una cuestión interesante. Os dejamos disponible todo lo necesario para que podáis disfrutar de ella en Velneo 7.22.1.

[Descargar código fuente del plugin QML de QRecorder para v7.22.1](#)[Descargar código fuente](#)

Recuerda que esta librería tendrás que **compilarla para la versión adecuada** de tu vClient.

Para facilitarte el trabajo, hemos empaquetado la versión para **Windows x32 en un descargable**. Solo tienes que copiar el contenido dentro de la carpeta Velneo (donde está ubicado el ejecutable del vClient.exe). Asegúrate de que el plugin queda dentro de la carpeta qml del directorio de instalación de Velneo.

[Descargar Plugin QML para Velneo v7.22.1](#)[Requiere estar registrado](#)

También os damos acceso al **VIN de todos los ejemplos** funcinando en Velneo V7

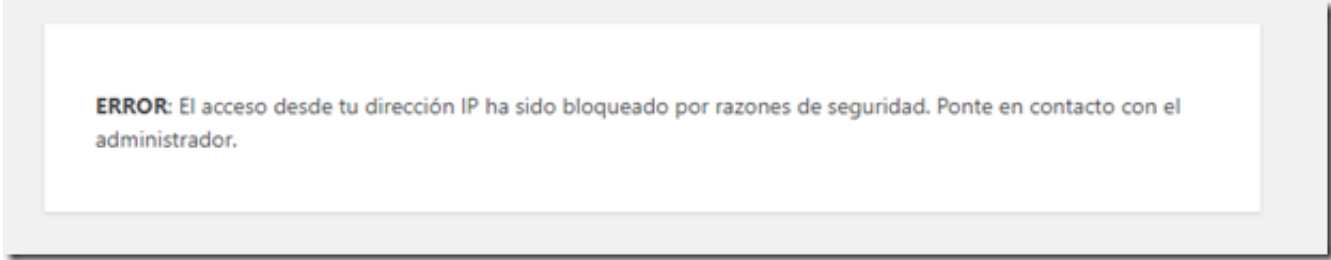
[Descargar VIN con los ejemplos de QtRecorder para Velneo v7.22.1](#)[Requiere estar registrado](#)

La verdad es que no se cual es el uso que la vais a dar, ya me contareis. Recuerda respetar la **licencia GNU** del mismo.

Aún hay más

Si crees que ya has superado todos los problemas es que aún te queda alguno que superar.

Pues eso, que voy a publicar en WordPress y zasca. Ha dicho que hoy no, tal vez mañana.



ERROR: El acceso desde tu dirección IP ha sido bloqueado por razones de seguridad. Ponte en contacto con el administrador.

Un abrazo... y espero que tengáis un camino más sencillo con las pruebas de este plugin.