

Detecting virtual keyboard open and close events in Symbian^3

Applications running on Symbian touch devices may want to react to software input panel (virtual keyboard) state events; for example, to improve user experience by immediately processing the input after virtual keyboard is dismissed. Qt provides `QEvent::CloseSoftwareInputPanel` event, but it's sent only after the target widget has lost focus.

This code snippet uses `QValueSpaceSubscriber` to detect virtual keyboard state changes. The code depends on a `Publish` and `Subscribe` key that is only available on Symbian^3 platform.



Warning: The following code implements a Qt class whose functionality is neither cross-platform nor future-proof. It only works on Symbian^3 platform with the default virtual keyboard in use (no signals are emitted when running on earlier S60 devices or other platforms).

Usage

```
#include <QValueSpaceSubscriber>

QValueSpaceSubscriber* vkbState = new QValueSpaceSubscriber("/vkb/state", this);
QObject::connect(vkbState, SIGNAL(contentsChanged()), this, SLOT(handleStateChanged()));
```

Declaring /vkb/state value state path

Create a new `.qcrml` file, with following content:

```
<?xml version="1.0" encoding="UTF-8"?>
<repository target="RProperty" version="" uidName="KPSUidAknFep" uidValue="0x100056de">
  <key ref="/vkb/state" int="0x00000004"/>
</repository>
```

Changes to .pro file

Add the following definitions to `.pro` file:

```
CONFIG += mobility
MOBILITY += publishesubscribe

symbian: {
  crml.pkg_postrules = "\\pskeydefs.qcrml\" - \"c:/resource/qt/crml/pskeydefs_$$TARGET}.qcrml\"
  DEPLOYMENT += crml
}
```



Warning: While this article shows how the `Publish` and `Subscribe` API can be used to access keys defined and populated by other applications, creating such a `*.qcrml` file raises the deployment problem. In Symbian each file can be owned and deployed by only one package, so if two applications are using this solution and attempt to install the `pskeydefs.qcrml` file, the second app will always fail. The uninstall sequence is also problematic, since one of the apps could remove the file while other installed apps still need it.

Above solution tries to avoid these problems by modifying the deployed `.qcrml` filename to include also the name of the application (`TARGET`), thus making it unique.

Example project

File: [Vkbstatetest qtmobility symbian3.zip](#) (can be self-signed, doesn't use any capabilities)

