
Subject: Uniq : single instance applications made easy
Posted by [mdelfede](#) on Sat, 05 Feb 2011 23:27:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

I started Uniq class, by now working just for Linux, windows part in development. It allows to have a single application instance running and other instances passing their command line to it.
Example code :

UniqTest.lay :

```
LAYOUT(UniqTestLayout, 404, 240)
  ITEM(LineEdit, edit, SetEditable(false).WantFocus(false).LeftPosZ(8, 388).TopPosZ(8, 220))
END_LAYOUT
```

UniqTest.h :

```
#ifndef _UniqTest_UniqTest_h
#define _UniqTest_UniqTest_h

#include <CtrlLib/CtrlLib.h>

#include <Uniq/Uniq.h>

using namespace Upp;

#define LAYOUTFILE <UniqTest/UniqTest.lay>
#include <CtrlCore/lay.h>

class UniqTest : public WithUniqTestLayout<TopWindow>
{
private:
public:
  typedef UniqTest CLASSNAME;
  UniqTest();

  // handles news app instances
  void handleInstance(Vector<String> const &v);
};
```

UniqTest.cpp :

```
#include "UniqTest.h"

UniqTest::UniqTest()
{
```

```

CtrlLayout(*this, "Window title");
}

// handles news app instances
void UniqTest::handleInstance(Vector<String> const &v)
{
String s = edit.Get() + "NEW INSTANCE LAUNCHED\n Command line : ";
for(int i = 0; i < v.GetCount(); i++)
    s += v[i] + ",";
if(s[s.GetCount() - 1] == ',')
    s = s.Left(s.GetCount()-1);
s += "\n";

edit <<= s;
ProcessEvents();
}

GUI_APP_MAIN
{
// create Uniq object
Uniq uniq;

// if not inside first instance, send commandline
// to first instance and leave
if(!uniq)
    return;

UniqTest uniqTest;

// setup callback handling other app's instances
uniq.WhenInstance = callback(&uniqTest, &UniqTest::handleInstance);

uniqTest.Run();
}

```

This code allows first app instance to start. If you start other instances, their Command Line is routed to it with a posted callback, allowing opened instance, for example, to open a file. In this small example, it just echoes command line of newly started apps.

Ciao

Max
