

GSI
Online
Offline
Object
Oriented

Go4 GUI and GSI's QtROOT interface

ROOT 2002

M. Al-Turany, D. Bertini, J. Adamczewski, H.G.Essel, S.Linev



Go4 GUI and QtROOT

- **Features of Qt ROOT Interface**
- **Using the Qt Designer**
- **Go4 GUI Singleton**
- **Features of the Go4 GUI**
- **Demo**

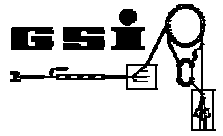


Qt Root Interface

Features:

- **Support only Unix.**
- **No changes in Root or Qt are needed**
- **Maintainable with minimum effort**
- **Can be used in parallel to the ROOT native GUI (TGxx classes).**
- **Full Root canvas functionality embedded in a QWidget.**
- **Drag and Drop support for canvas and pads.**
- **SelectedPadChanged(TPad *) signal, emitted from TPad**
- **The interface is tested and it works with ROOT ≥ 3.0 and Qt $\geq 2.3.0$**





Qt (Main.cxx)

The designer generated Main.cpp will looks like:

```
#include <qapplication.h>
#include "MyWidget1.h"
int main( int argc, char ** argv )
{
    QApplication a( argc, argv );
    MyWidget1 *w = new Mywidget1;
    w->show();
    a.connect( &a, SIGNAL( lastWindowClosed() ), &a, SLOT( quit() ) );
    return a.exec();
}
```



Qt Root (Main.cxx)

To use the QtRoot Interface you need:

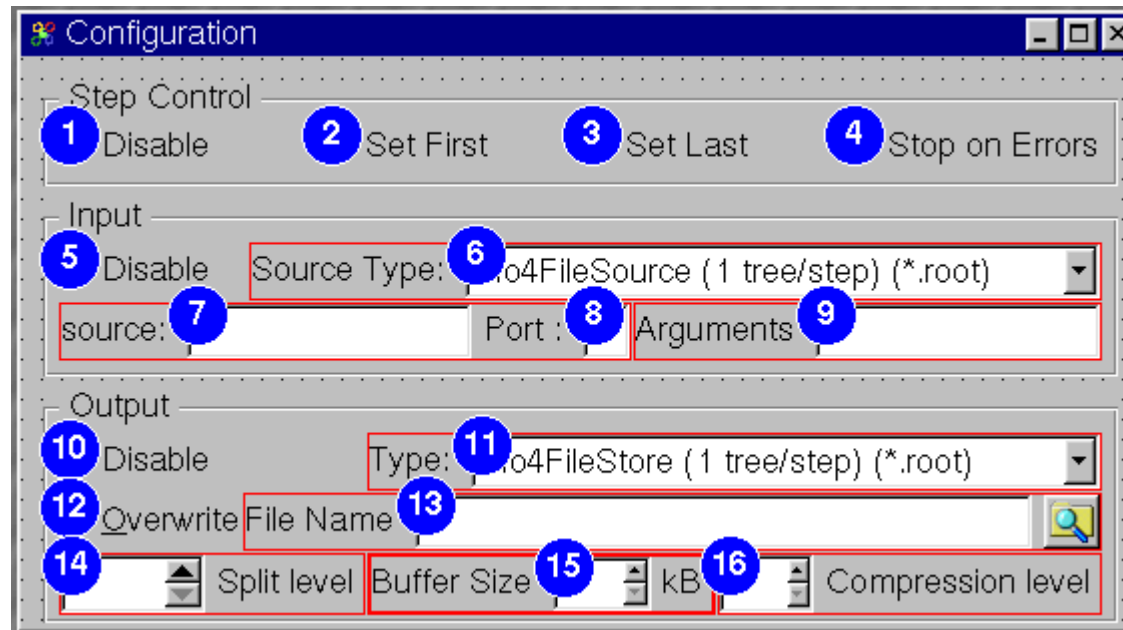
```
#include "YOUR QT-ROOT INTERFACE DIR/tqapplication.h"
#include "YOUR QT-ROOT INTERFACE DIR/qrootapplication.h"
#include "MyWidget1.h"
int main( int argc, char ** argv )

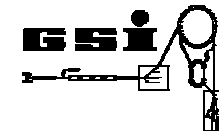
{
    QRootApplication a( argc, argv, 0);
    TQApplication app("uno",&argc,argv);
    MyWidget1 *w = new Mywidget1;
    w->show();
    a.connect( &a, SIGNAL( lastWindowClosed() ), &a, SLOT( quit() ) );
    return a.exec();
}
```



Qt Designer

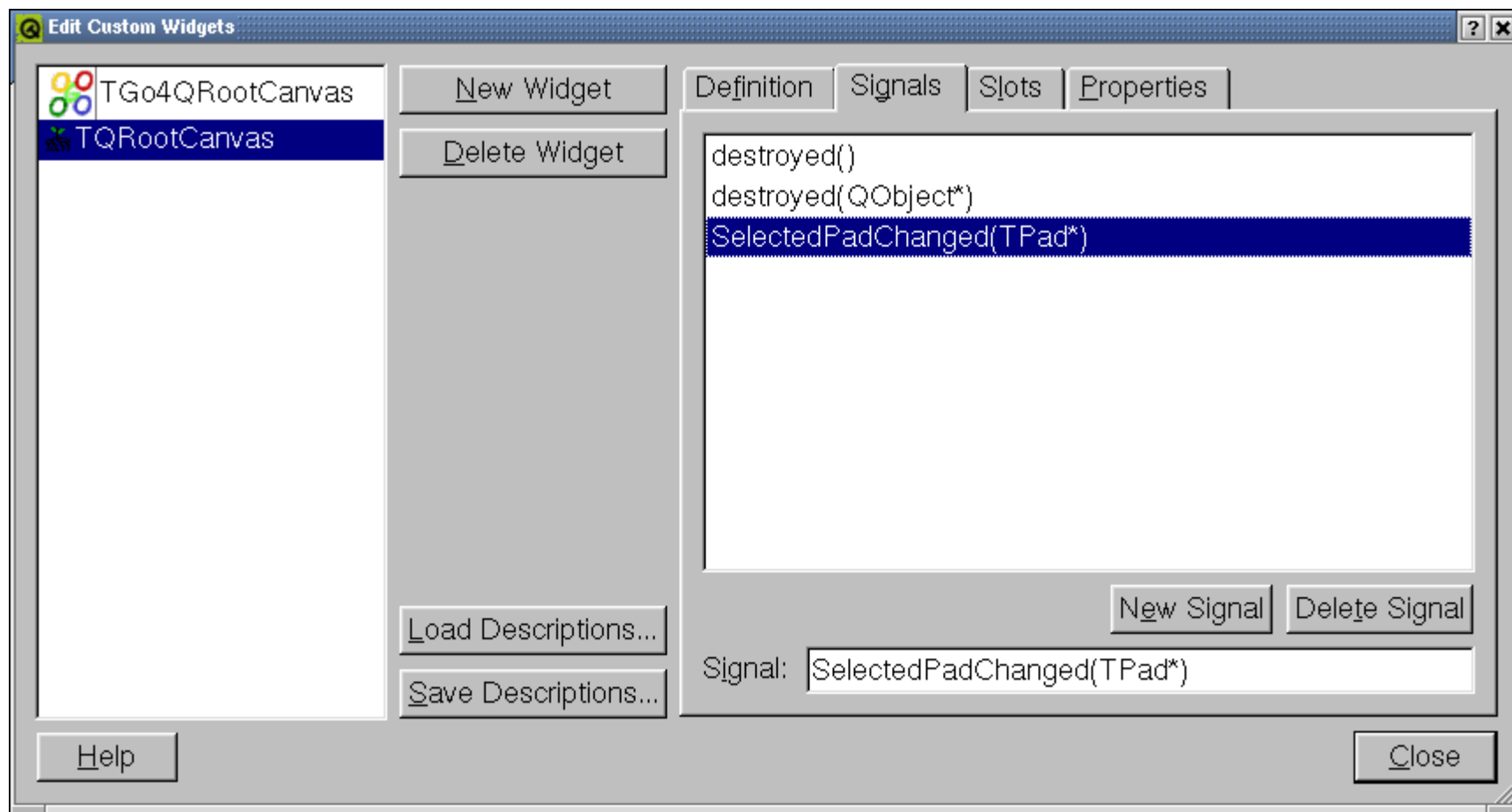
A professional tool to create complex GUI's very easily

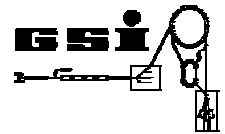




Qt Designer

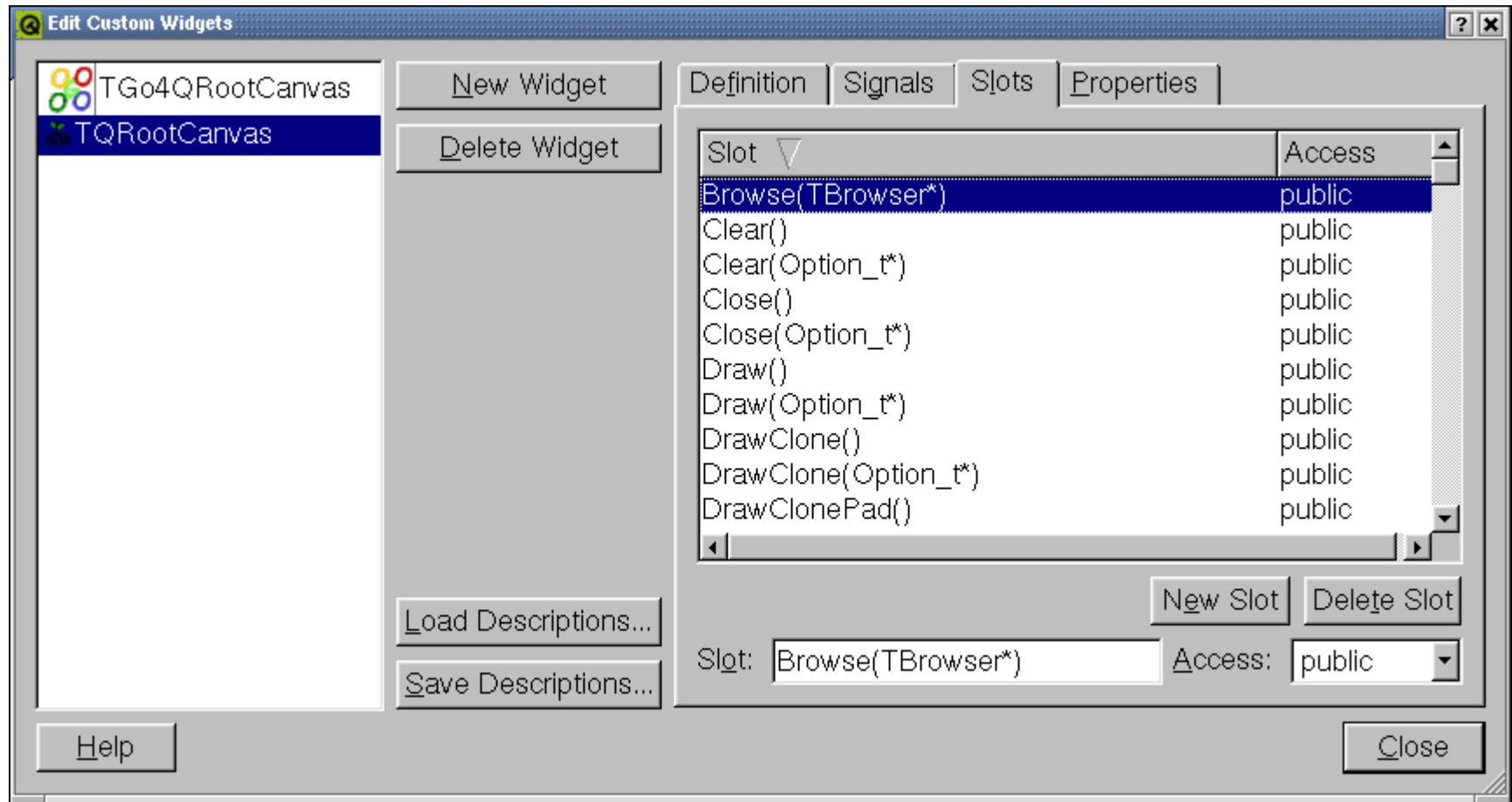
(Signals of QtRootCanvas)





Qt Designer

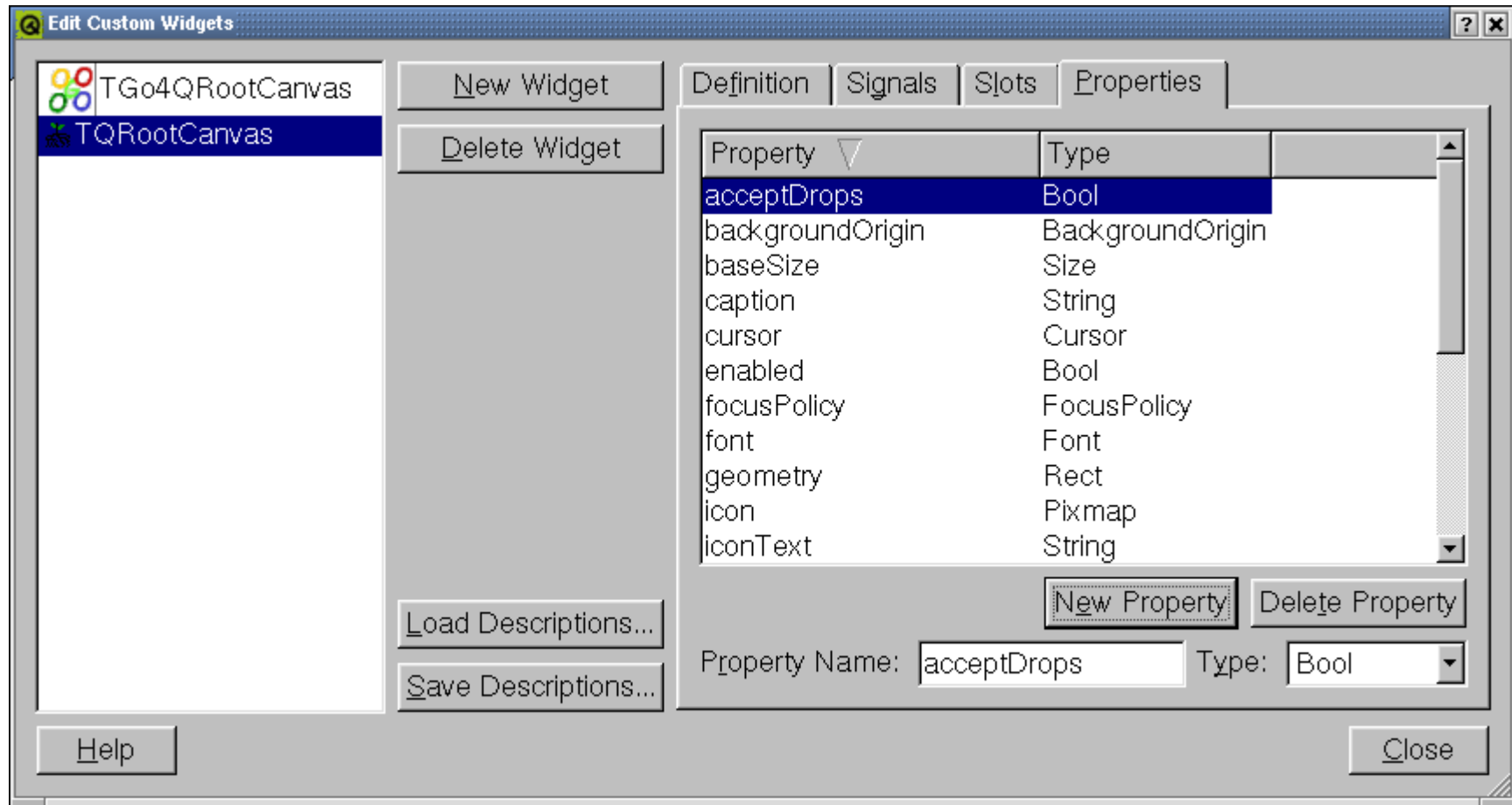
(Slots of QtRootCanvas)





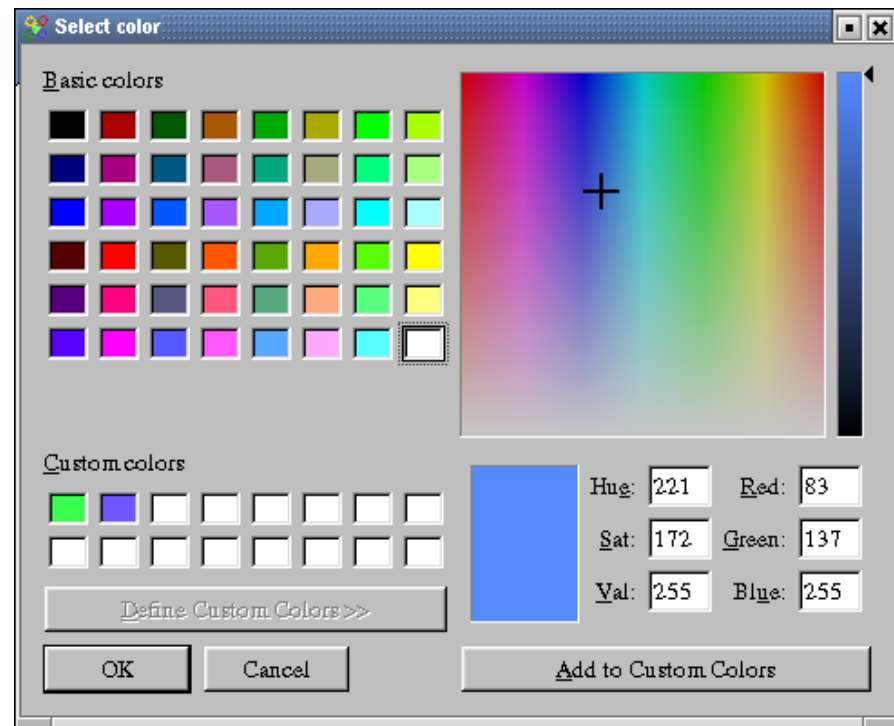
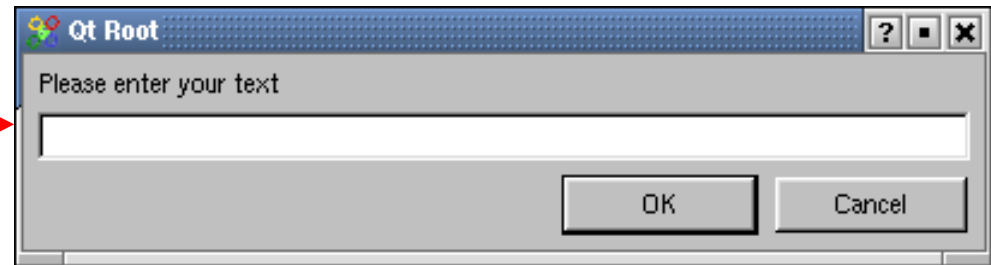
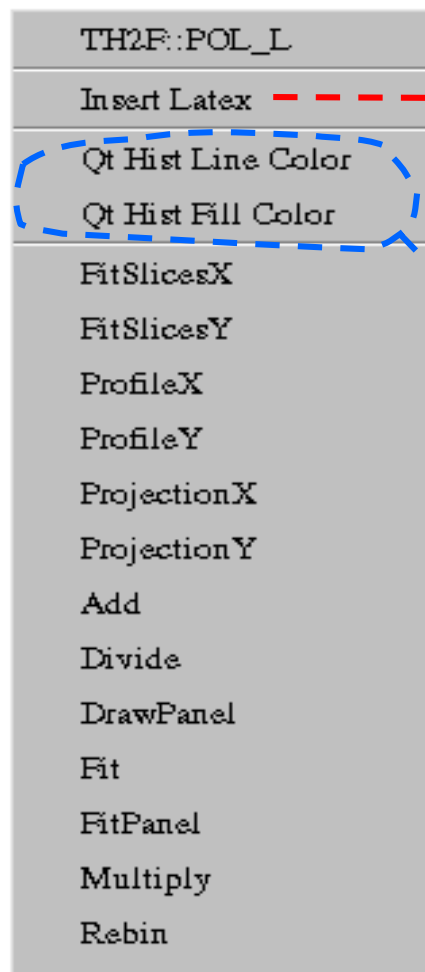
Qt Designer

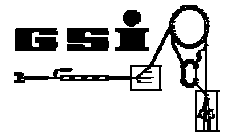
(Properties of QtRootCanvas)



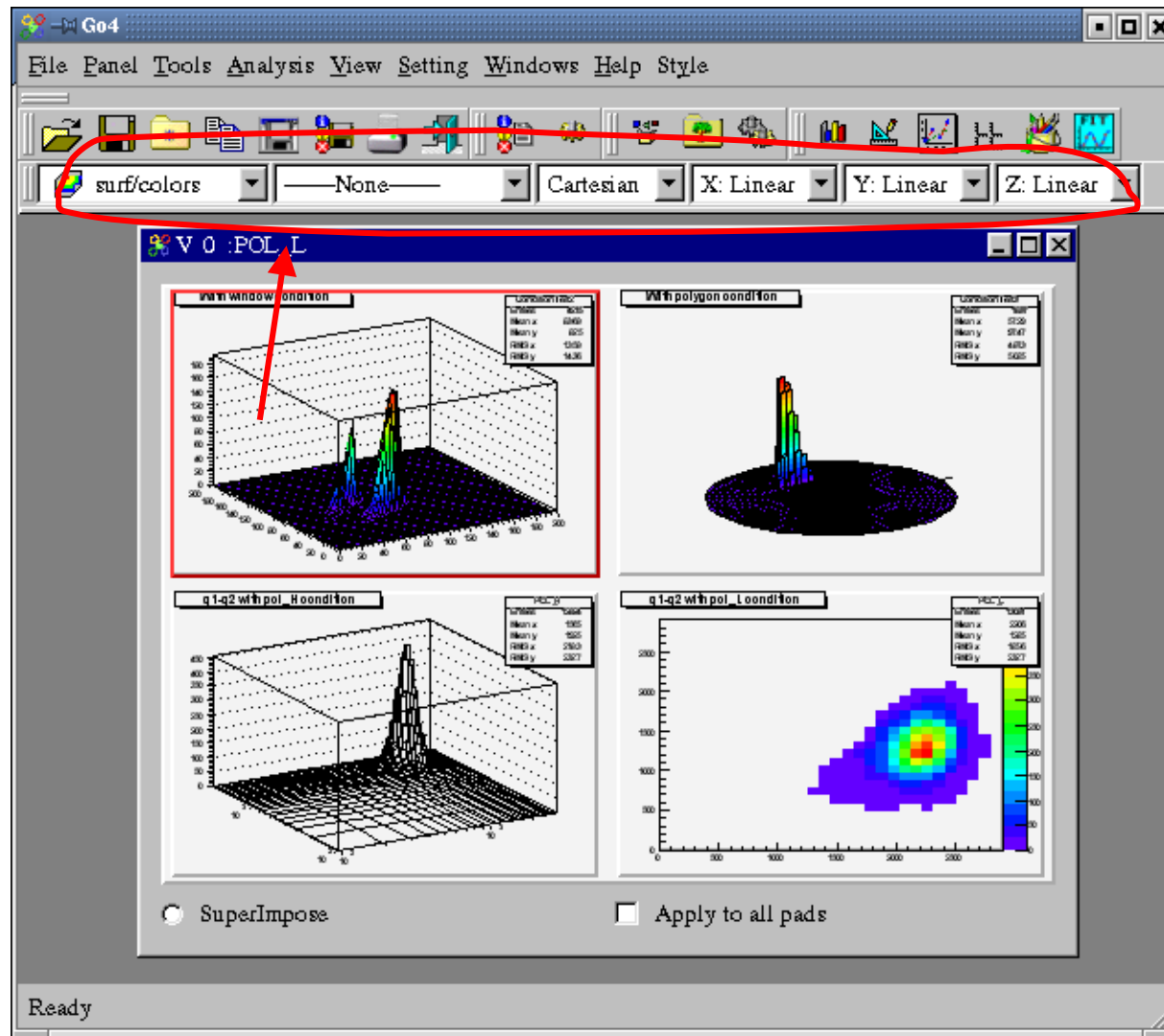


Modified Popup Menu



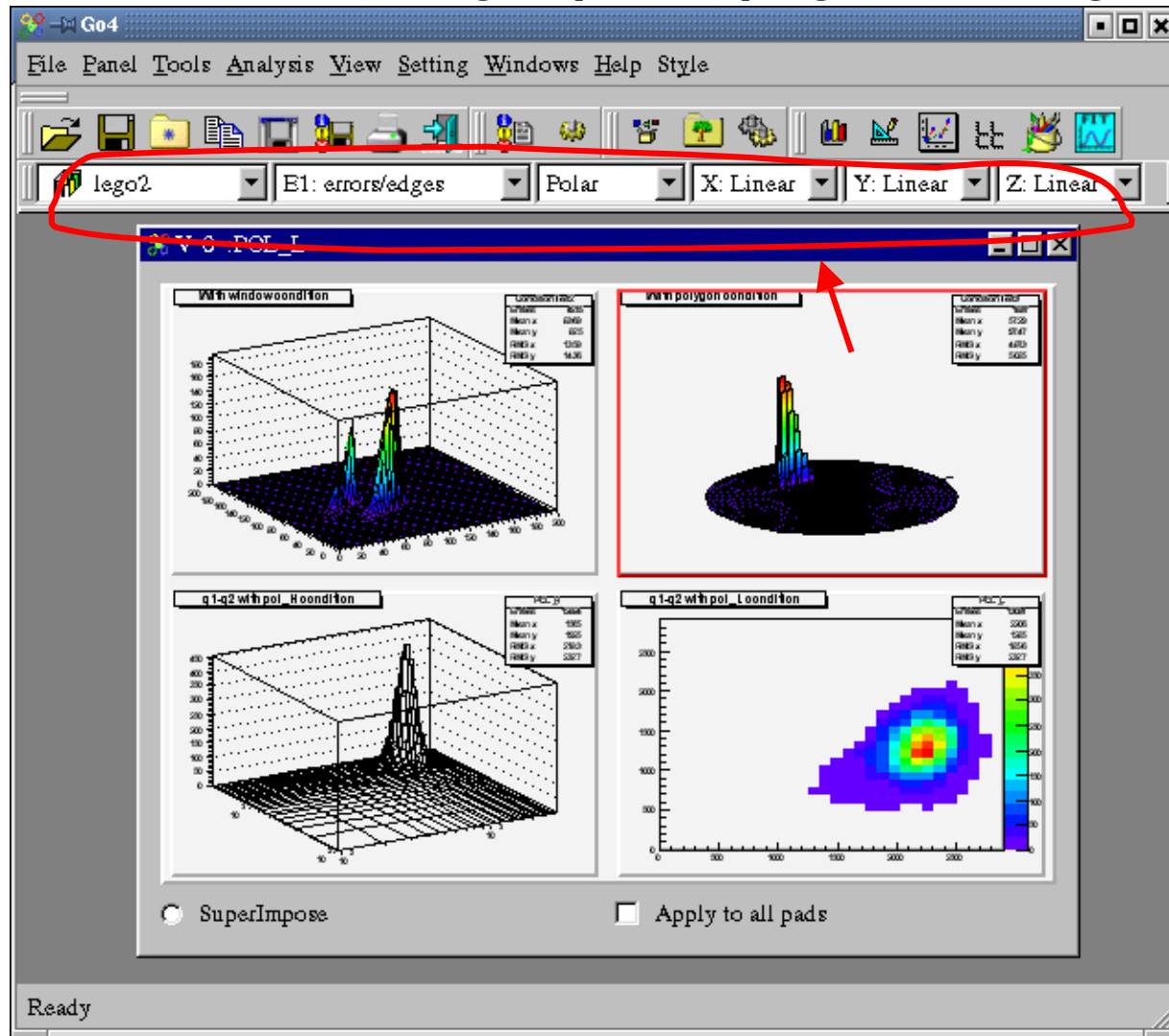


Use of SelectedPadChanged (TPad *) signal to change settings





Use of SelectedPadChanged (TPad *) signal to change settings





Go4 GUI Description

- **Complex GUI**
- **Flexible**
- **Rapid prototyping**
- **Easily extended**
- **Use ROOT and Qt together!**





Go4 GUI Architecture

- MDI Application
- **Remote** File browser and Tree viewer
- Signal-slot mechanism (both Qt and ROOT)
- Gui singleton: registry + dispatcher
(modified mediator pattern)
- Persistent status objects per window
- Well defined communication channels between the different parts of the GUI and external packages (e.g. Analysis, User GUI)



Gui Singleton: Registry + Dispatcher

- **A static Singleton which acts as a registry for all GUIs in the Go4**
- **All Commands sent to the analysis via this registry, also all objects sent from the analysis are delivered first here (dispatcher).**

[illegible]



Go4 GUI Features:

- Easy to use and self explaining GUI
- Simplify use of ROOT standard methods
- Drag and Drop support
- Online help is integrated
- User extensions are supported as:
 - Integrated in the Workspace and/or
 - Outside of the Go4 main window
- Look & Feel support (fonts, styles)

