Subject: login and main window
Posted by BetoValle on Sun, 01 Nov 2020 16:11:06 GMT
View Forum Message <> Reply to Message

There are 2 situations with different startup. What is the reason for the second to generate exception error?

(1) ok

```
String saida="";

struct Snh : public WithsnhLayout<TopWindow> {
  typedef Snh CLASSNAME;

   Snh(){
      CtrlLayout(*this, "Acesso inicial");
       ok.SetImage(ClassImg::btEdita);
       sai.SetImage(ClassImg::btSai);
        ok <<= THISBACK( fx );
       sai << [=] {Exit(); };

   }

   void fx(){
      saida="okxxx";
      Close();
   }

};


struct MainSig : public WithmenuLayout<TopWindow> {
 MainSig(){

    Snh se;
    se.Run();

    if( saida=="okxxx"){

        CtrlLayout(*this, "Tela Principal");
        menu.Set([=](Bar& bar) {

  bar.Sub("File", [=](Bar& bar) {
   bar.Separator();
  });

  bar.Sub("File", [=](Bar& bar) {
    bar.Add("Item 1", [&] {
                        Exit(); });
```

```
  });

 });
   }else{
       Exit();
   }
}

};

GUI_APP_MAIN
{
      MainSig sg;
      sg.Run();
}

(2) error
String saida="";
struct Snh : public WithsnhLayout<TopWindow> {
  typedef Snh CLASSNAME;

   Snh(){
      CtrlLayout(*this, "Acesso inicial");
       ok.SetImage(ClassImg::btEdita);
       sai.SetImage(ClassImg::btSai);
           ok <<= THISBACK( fx );
       sai << [=] {Exit(); };

   }

   void fx(){
      saida="okxxx";
      Close();
   }

};


struct MainSig : public WithmenuLayout<TopWindow> {
 MainSig(){

        CtrlLayout(*this, "Tela Principal");
        menu.Set([=](Bar& bar) {

  bar.Sub("File", [=](Bar& bar) {
   bar.Separator();
  });
```

```
  bar.Sub("File", [=](Bar& bar) {
    bar.Add("Item 1", [&] {
                          Exit(); });
  });

 });

}

};

GUI_APP_MAIN
{

    Snh se;
    se.Run();

    if( saida=="okxxx"){
       MainSig sg;
       sg.Run();
    }

}
```

Subject: Re: login and main window
Posted by Novo on Sun, 01 Nov 2020 16:43:27 GMT
View Forum Message <> Reply to Message

A hint: try to use a "code" tag to make code look better inside of a message.
An example:

```
Snh(){
  CtrlLayout(*this, "Acesso inicial");
  ok.SetImage(ClassImg::btEdita);
  sai.SetImage(ClassImg::btSai);
  ok <<= THISBACK( fx );
  sai << [=] {Exit(); };
}
```

Subject: Re: login and main window
Posted by BetoValle on Sun, 01 Nov 2020 18:30:28 GMT
View Forum Message <> Reply to Message

There are 2 situations with different startup. What is the reason
for the second to generate exception error ?

(1)
 ok
 String saida = "";

```
struct Snh : public WithsnhLayout<TopWindow>
{
 typedef Snh CLASSNAME;

 Snh()
 {
  CtrlLayout ( *this, "Acesso inicial" );
  ok.SetImage ( ClassImg::btEdita );
  sai.SetImage ( ClassImg::btSai );
  ok <<= THISBACK ( fx );
  sai << [=] {Exit();
       };

 }

 void fx()
 {
  saida = "okxxx";
  Close();
 }

};


struct MainSig : public WithmenuLayout<TopWindow>
{
 MainSig()
 {

  Snh se;
  se.Run();

  if ( saida == "okxxx" )
  {

   CtrlLayout ( *this, "Tela Principal" );
   menu.Set ( [=] ( Bar & bar )
   {
```

```
    bar.Sub ( "File", [=] ( Bar & bar )
    {
     bar.Separator();
    }

      );

    bar.Sub ( "File", [=] ( Bar & bar )
    {
     bar.Add ( "Item 1", [&]
     {
      Exit();
     } );
    }

      );

   }

      );
  }

  else
  {
   Exit();
  }
 }

};

GUI_APP_MAIN
{
 MainSig sg;
 sg.Run();
}

( 2 ) error
String saida = "";

struct Snh : public WithsnhLayout<TopWindow>
{
 typedef Snh CLASSNAME;

 Snh()
 {
  CtrlLayout ( *this, "Acesso inicial" );
  ok.SetImage ( ClassImg::btEdita );
  sai.SetImage ( ClassImg::btSai );
```

```
  ok <<= THISBACK ( fx );
  sai << [=] {Exit();
       };

 }

 void fx()
 {
  saida = "okxxx";
  Close();
 }

};


struct MainSig : public WithmenuLayout<TopWindow>
{
 MainSig()
 {

  CtrlLayout ( *this, "Tela Principal" );
  menu.Set ( [=] ( Bar & bar )
  {

   bar.Sub ( "File", [=] ( Bar & bar )
   {
    bar.Separator();
   }

     );

   bar.Sub ( "File", [=] ( Bar & bar )
   {
    bar.Add ( "Item 1", [&]
    {
     Exit();
    } );
   }

     );

  }

     );

 }

};
```

```
GUI_APP_MAIN
{

 Snh se;
 se.Run();

 if ( saida == "okxxx" )
 {
  MainSig sg;
  sg.Run();
 }

}
```

Subject: Re: login and main window
Posted by Klugier on Sun, 01 Nov 2020 19:55:53 GMT
View Forum Message <> Reply to Message

Hello,

Good job with improved formatting! Now I can answer For the next time you could improve formatting as well I don't know what causing the issues, but I could tell that you used some strange construction. Opening new window in constructor is the example. Moreover you use global to read the Run() status of the window. This is also something that is not optimal and may cause problems. You could overcome this situation by using Acceptors and Rejestros. For more information please read "20. Breaking the modal loop" section of GUI tutorial.

In case of any problems - please let us know. Also I updated GUI Tutorial with additional info in "Breaking the modal loop" section, so you should see changes tomorrow.

Klugier

Subject: Re: login and main window
Posted by BetoValle on Sun, 01 Nov 2020 21:30:13 GMT
View Forum Message <> Reply to Message

Hi Klugier

The example you mentioned doesn't answer, because the goal is to build a login screen, normal and as I'm new, and I also didn't find an example of this type (evolving from the login screen to a

main screen), I was having difficulties and in error and attempt I came across 2 situations above. I know it is not elegant / productive to use a variable to control the situation, but it is only a test in the 1st case.
How would you go about building the login screen and the option to navigate to the main screen?

Subject: Re: login and main window
Posted by Klugier on Sun, 01 Nov 2020 21:46:22 GMT
View Forum Message <> Reply to Message

Hello,

In context of login screen I would definitely go to option number 2 (Anyway why do not call this class elegant LoginWindow?):

```
GUI_APP_MAIN
{
 LoginWindow login_window;

   if (login_window.Run() != IDOK)
 {
            // The user click "Cancel" or "System close" button. So, we should finish app execution
here.
            // I strongly encourage to use notation from 20 GUI Tutorial with Acceptor and Rejector.
            return;
 }

      // We should obtain information about the user somehow...
      // Store credentials when user press "Enter" or click on "OK" button in login screen...
      Credentials credentials = login_window.GetCrdentials();
      MainWindow(credentials).Run(); // <- Run main with authenticated user
}
```

// In any case there is no need for global variables, which in most cases are not optimal and cause bugs.

Is it fine or you need more info in context of Window infrastructure? Optimally it is the good practice to name variables, classes etc. using English - it will be much easier to us to understand your code and you could easily share it outside your country.

Klugier

Subject: Re: login and main window
Posted by BetoValle on Sun, 01 Nov 2020 21:57:09 GMT
View Forum Message <> Reply to Message

Thank you for your help. Your example is much better!
I was already thinking something
example gui-20C tutorial

Quote:

```
GUI_APP_MAIN
{
        Snh se;
 switch(se.Run()) {
 case IDOK:
    {
     se.Close();
     MainSig sg;
     sg.Run();}
  break;
 case IDCANCEL:  //not necessary
  return;
 }
}
```

Subject: Re: login and main window
Posted by BetoValle on Sun, 01 Nov 2020 22:05:32 GMT
View Forum Message <> Reply to Message

like this: work! Thanks.
Quote:

```
struct Snh : public WithsnhLayout<TopWindow> {
  typedef Snh CLASSNAME;

   Snh(){
    CtrlLayout(*this, "Acesso inicial");
    ok.SetImage(ClassImg::btEdita);
    sai.SetImage(ClassImg::btSai);
         ok.Ok() <<= Acceptor(IDOK);
  sai.Cancel() <<= Rejector(IDCANCEL);

   }


};
```

Subject: Re: login and main window
Posted by Klugier on Sun, 01 Nov 2020 22:22:51 GMT
View Forum Message <> Reply to Message

Hello,

You could simplify it even more according to GUI tutorial:

```
struct Snh : public WithsnhLayout<TopWindow> {
  typedef Snh CLASSNAME;

  Snh(){
   CtrlLayoutOKCancel(*this, "Acesso inicial");
  }
};
```

Please rename "sai" to "cancel" and everything should be OK. Also tomorrow there will be update to tutorail and you should see more information about "CtrlLayoutOKCancel" construction.

Also, here is some power tip - write application using English with t_ macro

```
CtrlLayoutOKCancel(*this, t_("Initial access"));
```

Then all you need to do is translate the app to your language in .t file and it should work like a charm. For more information please read following document. At the beginning it was hard to me to write using English only (code and text), but in the end it paid off.

Klugier