Subject: Gate() or flow control problem
Posted by pntkl on Tue, 10 Jul 2007 00:08:05 GMT
View Forum Message <> Reply to Message

Hi,
I am very new to programming (especially C++). I am having trouble with some code I'm writing to create an XML file that stores user information. (Is XML a preferred method for such a thing?)

Problem:
I'm trying to use Gate() in an if statement, to determine if there exists a duplicate entry in an ArrayCtrl. When I run the application, the flow control never hits the for() loop contained in User::Validate(). Can anyone help me understand what it is I am doing wrong?

Test Case:

User.cpp

```
#include "TrayApp.h"
#include "User.h"


void
User::Add()
{
 if(!IsNull(~modify.EditFName) && !IsNull(~modify.EditLName) && !IsNull(~modify.EditUser) &&
!IsNull(~modify.EditPass) && !IsNull(~modify.DropGroup))
 {
  if(Gate(Validate()))
  {
   sha.Put(~modify.EditPass);
   String hashpass = sha.Finish();
   userarray.Add(~modify.EditFName, ~modify.EditLName, ~modify.EditUser, hashpass,
~modify.DropGroup);
   userarray.GoEnd();
   modify.EditFName <<= modify.EditLName <<= modify.EditUser <<= modify.EditPass <<=
modify.DropGroup <<= Null;
   ActiveFocus(modify.EditFName);
   sha.Put("", 0);
  }
  else
   PromptOK("First and last name or user already exists!");
 }
 else
  PromptOK("Please complete all fields!");
}

bool
User::Validate()
```

```
{
 bool match;
 int rcount = userarray.GetCount();
 if(rcount > 0)
 {
  for(int i = rcount; i >= 1; i--)
  {
   PromptOK("For loop entered");
   String strFirstName = ~modify.EditFName;
   String strLastName = ~modify.EditLName;
   String strUser = ~modify.EditUser;
   String strColFirstName = userarray.GetColumn(i, 0); // Get the First Name
   String strColLastName = userarray.GetColumn(i, 1); // Get the Last Name
   String strColUser = userarray.GetColumn(i, 2); // Get the User Name

   if(CompareNoCase(strFirstName, strColFirstName) && CompareNoCase(strLastName,
strColLastName) || CompareNoCase(strUser, strColUser))
   {
    PromptOK("CompareNoCase match found");
    match = true;
    break;
   }
   else
   {
    PromptOK("CompareNoCase does not match");
    match = false;
   }
  }
 }
 else
 {
  PromptOK("Rcount Else");
  return false;
 }
 if(!match)
 {
  PromptOK("Match return false");
  return false;
 }
 else
 {
  PromptOK("Match return true");
  return true;
 }
}

void
User::Change()
```

```
{
 ;
}

void
User::Delete()
{
 if(userarray.IsCursor() && PromptOKCancel("Are you sure?"))
 {
  int row = userarray.GetCursor();
  userarray.Remove(row);
 }
 else
  PromptOK("Please select which user to delete.");
}

void
User::Open()
{
 ;
}

void
User::Save()
{
 ;
}

void
User::Exit()
{
 ;
}

//void
//User::Serialize()
//{
// ;
//}

User::User()
{
 CtrlLayout(*this, "APR User Management");
 CtrlLayout(modify);
 tab.Add(modify, "Manage");
 userarray.AddColumn("First Name");
 userarray.AddColumn("Last Name");
 userarray.AddColumn("User");
```

```
 userarray.AddColumn("Password");
 userarray.AddColumn("Group");
 modify.EditPass.Password();
 modify.DropGroup.Add("Tech");
 modify.DropGroup.Add("Admin");
 modify.add <<= THISBACK(Add);
 //modify.change <<= THISBACK(Change)
 modify.del <<= THISBACK(Delete);
 //fs.AllFilesType();
}
```

User.h

```
#ifndef _APR_User_h_
#define _APR_User_h_

#include <CtrlLib/CtrlLib.h>
#include <Crypto/Crypto.h>
#include <Report/Report.h>

using namespace Upp;

#define LAYOUTFILE <APR/User.lay>
#include <CtrlCore/lay.h>

class User : public WithUserLayout<TopWindow> {
 WithModifyLayout<ParentCtrl> modify;
 Sha1 sha;
 FileSel fs;
 String filename;

 void Add();
 bool Validate();
 void Change();
 void Delete();
 void Open();
 void Save();
 void Exit();

 typedef User CLASSNAME;

public:
 //void Serialize(Stream& s);

 User();
};
```

```
#endif
```

User.lay

```
LAYOUT(UserLayout, 520, 360)
 ITEM(TabCtrl, tab, LeftPosZ(8, 504).TopPosZ(20, 132))
 ITEM(ArrayCtrl, userarray, LeftPosZ(8, 504).TopPosZ(160, 192))
END_LAYOUT

LAYOUT(ModifyLayout, 496, 96)
 ITEM(Label, dv___0, SetLabel(t_("First Name")).LeftPosZ(4, 56).TopPosZ(20, 19))
 ITEM(Label, dv___1, SetLabel(t_("Last Name")).LeftPosZ(92, 56).TopPosZ(20, 19))
 ITEM(Label, dv___2, SetLabel(t_("User")).LeftPosZ(180, 56).TopPosZ(20, 19))
 ITEM(Label, dv___3, SetLabel(t_("Password")).LeftPosZ(268, 56).TopPosZ(20, 19))
 ITEM(Label, dv___4, SetLabel(t_("Group")).LeftPosZ(356, 56).TopPosZ(20, 19))
 ITEM(EditString, EditFName, LeftPosZ(4, 72).TopPosZ(48, 19))
 ITEM(EditString, EditLName, LeftPosZ(92, 72).TopPosZ(48, 19))
 ITEM(EditString, EditUser, LeftPosZ(180, 72).TopPosZ(48, 19))
 ITEM(EditString, EditPass, LeftPosZ(268, 72).TopPosZ(48, 19))
 ITEM(DropList, DropGroup, LeftPosZ(356, 72).TopPosZ(48, 19))
 ITEM(Button, add, SetLabel(t_("Add")).LeftPosZ(444, 42).TopPosZ(12, 15))
 ITEM(Button, change, SetLabel(t_("Change")).LeftPosZ(444, 42).TopPosZ(40, 15))
 ITEM(Button, del, SetLabel(t_("Delete")).LeftPosZ(444, 42).TopPosZ(68, 15))
END_LAYOUT
```

P.S. - Any pointers on more appropriate methods of writing this? :)