Subject: What happens with the virtual function PreDestroy in Linux?
Posted by nicomesas on Wed, 20 Sep 2006 14:14:11 GMT
View Forum Message <> Reply to Message

I don't know what happens with the virtual function PreDestroy in Linux.

It hoped that when my TopWindow is closed outside call by defect but does not happen thus.

Been I have observed in code in the Windows version and see that the function void Ctrl::WndDestroy() in CtrlCore/Win32Wnd.cpp makes a call to ::DestroyWindow(top->hwnd); this call generates a message that is gathered by function Ctrl::WindowProc in CrtlCore/Win32Proc.cpp

```
 case WM_DESTROY:
    PreDestroy();
```

But the Linux version of Ctrk::WndDestroy() does not invoke to that function of any way.

Well, finally I don't worry because the code to finalize I can put it in the destructor but it is to me rare that any call to that virtual function does not become.

Perhaps could be put a call to PreDestroy() at the end of the function Ctrl::WndDestroy() in linux?

Nico

Subject: Re: What happens with the virtual function PreDestroy in Linux?
Posted by mirek on Thu, 21 Sep 2006 19:44:08 GMT
View Forum Message <> Reply to Message

Please consider PreDestroy as internal implementation feature. You should not normally need to use it.

Subject: Re: What happens with the virtual function PreDestroy in Linux?
Posted by nicomesas on Thu, 21 Sep 2006 22:25:58 GMT
View Forum Message <> Reply to Message

OK
Then, he is correct to use the destructor to make this type of things?

```
class someThing : public TopWindow
{
  MySqlSession mySql ;
  someThing()
  {
```

```
    mySql.Connect(.....) ;
  }
  virtual ~someThing()
  {
    mySql.Close() ;
  }
}
```

I believe that still I have Windows mentality and I do not finish thinking that can do this and be to me so calm, but in fact works.

UPP is great

Nico

Subject: Re: What happens with the virtual function PreDestroy in Linux?
Posted by mirek on Thu, 21 Sep 2006 22:32:58 GMT
View Forum Message <> Reply to Message

nicomesas wrote on Thu, 21 September 2006 18:25OK
Then, he is correct to use the destructor to make this type of things?

```
class someThing : public TopWindow
{
  MySqlSession mySql ;
  someThing()
  {
    mySql.Connect(.....) ;
  }
  virtual ~someThing()
  {
    mySql.Close() ;
  }
}
```

I believe that still I have Windows mentality and I do not finish thinking that can do this and be to me so calm, but in fact works.

UPP is great

Nico

Actually, it is unnecessary. MySqlSession destructor Closes the connection itself, you do not have to care about it.

BTW, this is very basic U++ design: All resources *owned* by object are always released/finalized

when the object is destroyed (but you can release them sooner). Applies e.g. to files, sockets, printer jobs etc, etc.. This is what makes U++ development superior to GC based systems.
