Subject: Distorted GUI / memory leak
Posted by galious on Mon, 01 Mar 2010 21:33:47 GMT
View Forum Message <> Reply to Message

Hi,

I'm trying to create a 'static-linked plugin system' like is done for e.g. plugin/bmp and plugin/png. The plugins should provide a GUI to allow the users to set some configuration. Providing the controls works for buttons, but adding a treectrl results in a distorted UI (the text in the provided is not centered anymore) and a memory leak.

What am I doing wrong? Is this not supported or is this a bug somewhere in the upp code?

For an example see the code below.

Best regards,

Martin

```
#include <CtrlLib/CtrlLib.h>
using namespace Upp;

/* Plugins */
static StaticCriticalSection sAnyStoragePlugin;
class StoragePlugin {
 typedef StoragePlugin *(*StorageFactory)();
 template <class T> static StoragePlugin *FactoryFn() { return new T; }
 static void AddPlugin(StorageFactory f) { INTERLOCKED_(sAnyStoragePlugin)Map().Add(f()); }
 static Array<StoragePlugin>& Map() { static Array<StoragePlugin> x; return x; }

public:
 template <class T> static void Register() { AddPlugin(&StoragePlugin::FactoryFn<T>); }
};

class DiskStorage : public StoragePlugin {
 TreeCtrl  c;  // <<<< Comment me to get a correct GUI and remove the memory leak
 Button    b;
};

INITBLOCK {
 StoragePlugin::Register<DiskStorage>();
}

/* App */
class DeBunny : public TopWindow {
 Button button;
```

```
public:
 typedef DeBunny CLASSNAME;
 DeBunny() {
          Add(button.SetLabel("&I'm an Ultimate++ button!").VCenterPos(20).HCenterPos(200));
 };
};

GUI_APP_MAIN
{
 DeBunny().Run();
}
```

---

Subject: Re: Distorted GUI / memory leak
Posted by mirek on Tue, 02 Mar 2010 04:41:29 GMT
View Forum Message <> Reply to Message

```
#include <CtrlLib/CtrlLib.h>

using namespace Upp;

/* Plugins */
static StaticCriticalSection sAnyStoragePlugin;
class StoragePlugin {

 typedef StoragePlugin *(*StorageFactory)();
 template <class T> static StoragePlugin *FactoryFn() { return new T; }

 static void AddPlugin(StorageFactory f) { INTERLOCKED_(sAnyStoragePlugin)Map().Add(f); }
 static Array<StorageFactory>& Map() { static Array<StorageFactory> x; return x; }

public:
 template <class T> static void Register() { AddPlugin(&StoragePlugin::FactoryFn<T>); }
};

class DiskStorage : public StoragePlugin {
 TreeCtrl  c;  // <<<< Comment me to get a correct GUI and remove the memory leak
 Button    b;
};

INITBLOCK {
 StoragePlugin::Register<DiskStorage>();
}

/* App */
```

```
class DeBunny : public TopWindow {
 Button button;

public:
 typedef DeBunny CLASSNAME;
 DeBunny() {
        Add(button.SetLabel("&I'm an Ultimate++ button!").VCenterPos(20).HCenterPos(200));
 };
};

GUI_APP_MAIN
{
 DeBunny().Run();
}
```

---

Subject: Re: Distorted GUI / memory leak
Posted by galious on Wed, 03 Mar 2010 11:04:39 GMT
View Forum Message <> Reply to Message

Hi,

the 'problem' with replacing Plugin by Factory was I've to use something like the following code:

```
One<StoragePlugin> GetPlugin(int i) {
   One<StoragePlugin> plugin = (*StorageFactory(Map()[i]))();
   return plugin;
}
```

this will, I think, create a new StoragePlugin object each time I iterate through the plugins. In this way I won't able to store the state in the plugin, as each invocation will return a default plugin object.

I think I've to rewrite this a bit to iterate through all factories at initialisation and create an array of plugins (by using the method shown above) and afterwards use this list, or I'll create the plugins as singletons. I'll try these possibilities tonight.

One question still stands though, why did a Button work and a TreeCtrl didn't, just coincidence?

Best regards,

Martin

---

Subject: Re: Distorted GUI / memory leak
Posted by mirek on Thu, 04 Mar 2010 06:32:05 GMT
View Forum Message <> Reply to Message

galious wrote on Wed, 03 March 2010 06:04

One question still stands though, why did a Button work and a TreeCtrl didn't, just coincidence?

Yes. Generally, there is a rule saying "no widgets before GUI_APP_MAIN". (Means: no widgets should be constructed in global constructors).

Mirek

Subject: Re: Distorted GUI / memory leak
Posted by galious on Thu, 04 Mar 2010 07:13:14 GMT
View Forum Message <> Reply to Message

Ahhh, thanks. I've rewritten to part to register the factory and afterwards I initialize an Array of plugins.

The memory leak was removed by added a virtual destructor (stupid me to forget it).

Thanks for your help,

Martin