Subject: Skylark and GUI in one app - need some advices
Posted by Peter on Thu, 25 Apr 2013 11:17:34 GMT

View Forum Message <> Reply to Message

Hi.

I need to write a GUI app and have the ability to control it, alter its state or terminate it remotely from a separate website (sort of administration panel). Basically, I need to integrate GUI and Skylark in a single application in some clever and elegant way.

My first thought is to add an object of class derived from SkylarkApp as a field to my GUI class and start Skylark server in a separate thread while starting the whole application. It would be something like this:

App - my GUI class inherited from TopWindow
WebInterface - inherited from SkylarkApp

App class contains WebInterface object as one of its contents.

I start the whole app with App().Run() while WebInterface constructor contains the following line:


Thread().Run(THISBACK(WorkingThread));


This way I start both GUI and Web part of my application in separate threads. I hope you have a general idea of what I intend to do. Is that approach correct or potentially dangerous?

Meanwhile, I have two urgent problems to solve and need your help.

1. Is there any standard way to stop Skylark server from within a program? Basically, I need a clean way to stop the whole application without manually killing it by pressing CRTL+C. If such funtionality is currently not available, could you please add it to U++?

2. In a recent version of U++ (build 5990) on Ubuntu 12.04
an attempt to compile the following code (a single .cpp file) with g++ 4.6 and GUI MT SSE2 flags:


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

GUI_APP_MAIN
{
}
```


results in the following error:

"In file included from /home/piotr/MyApps/SkylarkTest/main.cpp:2:0:
/home/piotr/upp/uppsrc/Skylark/Skylark.h:86:15: error: expected unqualified-id before numeric constant"

For some reason if I change BadRequest to BadRequest1 in the following declaration in Skylark/Skylark.h


virtual void BadRequest(Http& http, const BadRequestExc& e);


the code compiles, but what's the real cause of this error? Looks like some bug in U++.

Regards

Piotr

---

## Subject: Re: Skylark and GUI in one app - need some advices
Posted by Zbych on Sat, 27 Apr 2013 20:05:37 GMT
View Forum Message <> Reply to Message

It is possible to close Skylark nicely, but it requires some changes. This post can give you some clues.

1. You need to add Quit() public method to Skylark:

```
void SkylarkApp::Quit()
{
 quit = true;
#ifdef PLATFORM_POSIX
 Broadcast(SIGTERM);
#endif
 TcpSocket s;
 s.Timeout(100);
 s.Connect("127.0.0.1", port);
}
```


2. Main() also requires some changes, because it waits for all threads in your application to shutdown. If Skylark is run from another thread it will wait for itself in endless loop.
After this change it will wait only for it's own threads:

```
void SkylarkApp::Main()
{
 Buffer<Thread> uwt(threads);
 for(int i = 0; i < threads; i++)
```

```
    uwt[i].Run(THISBACK(ThreadRun));

 /* Wait for threads to shut down */
 for (int i = 0; i < threads; i++)
  uwt[i].Wait();
}
```

3. If you need to wait for Skylark to shutdown you should use Run and Wait methods instead of Start:

```
class MyApp: public Skylark{

    Thread th;

    void Start() {th.Run(THISBACK(Run));}

    void Stop() {Quit(); th.Wait();}
}
```

4. If you need to pass data to skylark handlers via global variables, you should disable forking and use atomic types or mutexes, to protect data integrity.


Mirek, what do you think about those changes (1 & 2)?