
Subject: Re: GUI and printf

Posted by [Sender Ghost](#) on Thu, 21 Jun 2012 18:08:18 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello, Didier.

shmoky wrote on Thu, 21 June 2012 18:01I have made a GUI application under U++ and wants to be able to output printf statements in a console window. How can it be done ?

I am on Win7.

You need to use AllocConsole and FreeConsole functions. Then get console handle and manage it (e.g. to associate with stdout and stderr streams, etc.).

Some example:

Toggle Spoiler

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

```
#include <wincon.h>
```

```
#include <fcntl.h>
```

```
using namespace Upp;
```

```
bool CreateConsole()
```

```
{
```

```
    if (!AllocConsole())  
        return false;
```

```
    int consoleHandle = _open_osfhandle((long)GetStdHandle(STD_OUTPUT_HANDLE),  
    _O_TEXT);
```

```
    if (consoleHandle == -1)  
        return false;
```

```
    *stdout = *(_fdopen(consoleHandle, "w"));
```

```
    setvbuf(stdout, NULL, _IONBF, 0);
```

```
    *stderr = *(_fdopen(consoleHandle, "w"));
```

```
    setvbuf(stderr, NULL, _IONBF, 0);
```

```
    std::ios::sync_with_stdio();
```

```
    return true;
```

```
}
```

```
class App : public TopWindow {
```

```
private:
```

```
    bool opened;
```

```
public:
```

```
    typedef App CLASSNAME;
```

```

App();

EditString text;
FrameRight<Button> btnSend;
Button btnManage;

void ManageConsole();
void ManageLabel();
void Send();

virtual bool Key(dword key, int count);
};

void App::ManageConsole()
{
    if (opened)
        opened = FreeConsole() == 0;
    else
        opened = CreateConsole();

    ManageLabel();
}

void App::ManageLabel()
{
    btnManage.SetLabel(opened ? "Close Console (F11)" : "Open Console (F11)");
}

void App::Send()
{
    if (stdout == NULL)
        return;

    String data = ~text;
    if (!data.IsEmpty())
        fprintf(stdout, "%s\n", ~data);
}

bool App::Key(dword key, int count)
{
    if (key == K_F11) {
        ManageConsole();
        return true;
    }
    if (key == K_ESCAPE)
        Break();

    return TopWindow::Key(key, count);
}

```

```

}

App::App() : opened(false)
{
    Title("From GUI to console");
    SetRect(Size(320, 120));
    MinimizeBox();

    btnSend.SetImage(CtrlImg::smallright());
    btnSend.Tip("Send output to console (Enter)");
    text.NullText(CtrlImg::write(), "Write here..");
    btnSend.WhenPush = text.WhenEnter = THISBACK(Send);
    text.AddFrame(btnSend);

    btnManage.WhenPush = THISBACK(ManageConsole);
    ManageLabel();

    Add(text.HSizePosZ(4, 4).TopPosZ(4, 19));
    Add(btnManage.HSizePosZ(4, 4).TopPosZ(27, 19));
}

GUI_APP_MAIN
{
    App app;
    app.Run();
}

```

Some references:

- "Adding a console window C++ (WinAPI)?".
- "Console in GUI".