Subject: Thread calls GUI
Posted by Sami on Sat, 14 Feb 2009 18:26:01 GMT
View Forum Message <> Reply to Message

I realize a thread cannot call GUI in upp. It is not however clear how threading should be implemented then. I would ask help for proper solution to the example given below.

```
struct Interface {
  virtual int Ask ( const char * ) = 0;
};

struct Work {
  Interface *gui;
};

struct Library {
  Library ( Work w ) {
    int a = w.gui->Ask ( "Ok?" );
  }
};

void Threading ( Work w ) {
  Library ( w );
}

struct Task
:MyTask<TopWindow>
,Interface {
  typedef Task CLASSNAME;
  Task() {
    CtrlLayout(*this, "Example" );
    Work w;
    w.gui = this;
    Thread().Run ( callback1 ( Threading, w ) );
  }
  volatile Atomic q;
  int Ask_Weird_Hacked ( const char *s, unsigned dummy ) {
    return q = 1 + PromptYesNo ( String().Cat() << s );
  }
  int Ask ( const char *s ) {
    //problem here, cannot call PromptYesNo()
    q = 0;
    PostCallback ( callback2 ( this, &Task::Ask_Weird_Hacked, s, 0 ) );
    while ( !q ) Sleep ( 10 );
    return q - 1;
  }
};
```

So we begin with Task() and our problem is how to implement Ask() call properly. I first understood the Gate-method is what I'm looking for, but I didn't get it to work, can somebody explain what is it? The manual was in my opinion incomplete here.
