

---

Subject: Re: BackgroundTask

Posted by [dolik.rce](#) on Wed, 16 Feb 2011 14:32:13 GMT

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

---

```
Slightly improved version:  
class BackgroundTask{  
    typedef BackgroundTask CLASSNAME;  
#ifdef _MULTITHREADED  
    Thread t;  
#endif  
    bool running;  
    const int id;  
    static Index<int> list;  
  
    void Watch(Callback task);  
    static int AssignId();  
  
    friend bool ProcessForeground(int);  
public:  
    void Start(Callback task);  
    void Stop() {int n=list.Find(id); if(n>=0) list.Remove(id);} {return id;}  
    int GetId() {return id;}  
    bool IsRunning() {return running;}  
    BackgroundTask():running(false),id(AssignId()) {}  
    BackgroundTask(Callback task):id(AssignId()) {Start(task);} {  
};  
    Index<int> BackgroundTask::list;  
  
    void BackgroundTask::Watch(Callback task){  
        running=true;  
        task();  
        Stop(id);  
        running=false;  
    }  
  
    int BackgroundTask::AssignId(){  
        static int sid=0;  
        return sid++;  
    }  
  
    void BackgroundTask::Start(Callback task){  
        if(IsRunning()) Stop();  
        list.Add(sid);  
#ifdef _MULTITHREADED  
        t.Run(THISBACK1(Watch,task));  
#else  
        PostCallback(THISBACK1(Watch,task));  
#endif  
    }  
}
```

```
bool ProcessForeground(int id){  
#ifndef _MULTITHREADED  
Ctrl::ProcessEvents();  
#endif  
return BackgroundTask::list.Find(id)>=0;  
}
```

It allows stopping tasks individually based on unique ID. Also reusing the task instances should work now. The major difference is that ProcessForeground now takes task ID as argument.

Honza

EDIT: Removed file due to contained errors

---