

---

Subject: Re: Callbacks with 'Disable' feature  
Posted by [kohait00](#) on Thu, 07 Oct 2010 09:39:07 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

maybe sth as simple as this..

```
template <class P1>
class Callback1 : Moveable< Callback1<P1> > {
    Callback1Action<P1> *action;
    bool act;

    void Retain() const { if(action ) AtomicInc(action->count); }
    void Release()      { if(action && AtomicDec(action->count) == 0) delete action; }

    bool operator==(const Callback1&);
    bool operator!=(const Callback1&);

public:
    typedef Callback1 CLASSNAME;

    Callback1& operator=(const Callback1& c);
    Callback1(const Callback1& c);
    void Clear()      { Release(); action = NULL; }

    void Enable(bool b = true) { act = b; }
    void Disable() { Enable(false); }
    bool Enabled() const { return act; }

    operator bool() const      { return action && action->IsValid(); }
    void Execute(P1 p1) const   { if(act && action) action->Execute(p1); }
    void operator()(P1 p1) const { Execute(p1); }

    explicit Callback1(Callback1Action <P1> *newaction) { act = true; action = newaction; }
    Callback1() { act = true; action = NULL; }
    Callback1(_CNULL) { act = true; action = NULL; }
    ~Callback1();

    static Callback1 Empty() { return CNULL; }

};
```

---