

---

Subject: A new [Ctrl] timer id strategy

Posted by [huanghuan](#) on Tue, 04 Mar 2008 15:07:32 GMT

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

---

```
Ctrl::~Ctrl() {
```

```
    ...
    KillTimeCallbacks(this, (byte *) this + sizeof(Ctrl));
    [COLOR=red]KillTimeCallbacks(this, (byte *) this + sizeof(this));[/COLOR]
}
```

How can red code implement its mean ? I don't know.

Why? reasons as follow:

1. lazy coder write a new ctrl inherit from some deep inherited ctrl classes. lazyer don't want read the all classes in inherit tree.

```
class ACtrl : public SomeDeepInheritedClassBySomeOthers
{
```

```
    ...
    Ctrl::SetTimeCallback(..., id? );
```

2. In big inherit tree. timer id is managed by increment. But sometimes we forget or mistake the last id. The follow code is safe. So Ctrl::SetTimeCallback is desperate.

```
class A : public Ctrl
```

```
{
    char dummy;
public:
```

```
    ...
    Upp::SetTimeCallback(..., &dummy);
```

```
    ...
};
class AA : public A
```

```
{
    char dummy;
public:
```

```
    ...
    Upp::SetTimeCallback(..., &dummy);
```

```
    ...
};
```

---

---

Subject: Re: A new [Ctrl] timer id strategy

Posted by [mirek](#) on Tue, 04 Mar 2008 16:20:36 GMT

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

---

Yep. This is what TimeCallback class does.

Using "direct id" is "low-level practice"...

Mirek

---