

---

Subject: What does this compiling error mean exactly?  
Posted by [sinpeople](#) on Sun, 01 Nov 2020 16:27:21 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hi folks,

I have a piece of program from this forum. It works fine before I tried to extend it under its current framework.

Please give newbie a help hand.

The code I added is:

```
struct Stage{
    int stageID;
    int stageDuration;
};

struct SignalScheduleMessage : AMessage {
    Time time;
    int nCycleID;
    int nStages;
    Vector<Stage> stages;

    virtual void Serialize(Stream& s) {
        s % time % nCycleID % nStages;
        for(int i=0; i<stages.GetCount(); i++)
            s % stages[i].stageID % stages[i].stageDuration;
    }
};
```

and

```
RegisterMessage<SignalScheduleMessage>(3); // do that for all of your messages
```

The compiling time error message is the following:

The full package of the code (after extension) is here  
`#include <Core/Core.h>`

```
namespace Upp {
    struct AMessage {
        virtual void Serialize(Stream& s) = 0;
    };
};
```

```

    virtual ~AMessage() {}
};

typedef Function< void (One<AMessage>&) > MessageMake;

VectorMap<int, MessageMake> message_maker;

template <class T>
void RegisterMessage(int messageid)
{
    message_maker.Add(messageid, [](One<AMessage>& m) { m.Create<T>(); });
}

// =====
//   Messages definition
// =====
struct TemperatureMessage : AMessage {
    double altitude, temperature;

    virtual void Serialize(Stream& s) {
        s % altitude % temperature;
    }
};

struct WarningMessage : AMessage {
    String text;

    virtual void Serialize(Stream& s) {
        s % text;
    }
};

struct Stage{
    int stageID;
    int stageDuration;
};

struct SignalScheduleMessage : AMessage {
    Time time;
    int nCycleID;
    int nStages;
    Vector<Stage> stages;

    virtual void Serialize(Stream& s) {
        s % time % nCycleID % nStages;
        for(int i=0; i<stages.GetCount(); i++)

```

```

        s % stages[i].stageID % stages[i].stageDuration;
    }
};

// =====
//   Message registration
// =====

INITBLOCK {
    RegisterMessage<TemperatureMessage>(1); // do that for all of your messages
    RegisterMessage<WarningMessage>(2); // do that for all of your messages
    RegisterMessage<SignalScheduleMessage>(3); // do that for all of your messages
};

void ProcessRequest(const String& data)
{
    stringstream ss(data); // error handling for now omitted
    int client_id = ss.Get32();
    int message_id = ss.Get32();
    One<AMessage> m;
    int q = message_maker.Find(message_id);
    if(q < 0)
        return;
    (message_maker[q])(m); // create the required concrete message
    ss % *m; // load data to struct
}

}

using namespace Upp;

CONSOLE_APP_MAIN
{
}

```

Thank you very much!

Best Regards  
David WANG

## File Attachments

1) [Capture.JPG](#), downloaded 414 times

---