#### LabVIEW Actor Framwork (1)\_\_\_\_\_\_\_\_ 边学边做server&client

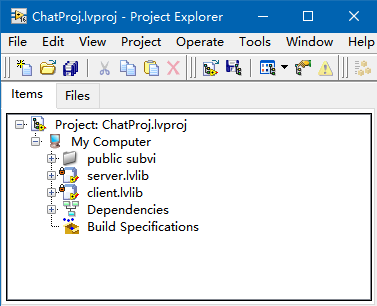
初始需求：

现在要做一个类似聊天的demo，一个server端，若干个client端；首先是server启动，通过server可以打开若干个client端，然后每个client可以独立给server发送数据。server收到数据打印出来带有client id的信息。

必然想到，这里出现了2个Actor，一个server，一个client；

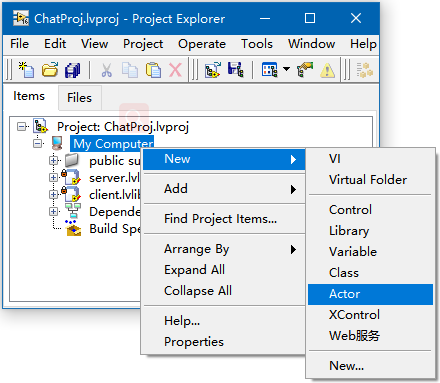
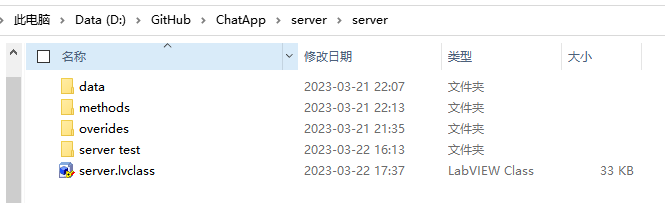
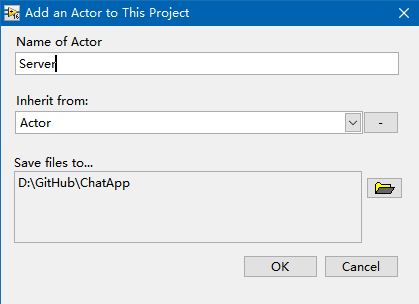
下面，我们来编写和测试第一个actor——server

### 一、打开LabVIEW2016，新建一个proj：



 并保存到硬盘文件夹。

### 二、右键项目中的My Computer新建Actor：

 最后建立好的文件目录如上图，硬盘存储文件夹的定义如下：

data——存放定义该Actor的私有数据类型；

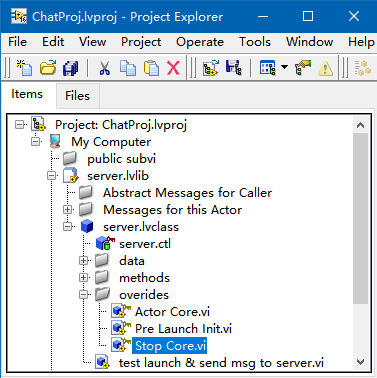
mothods——方法；

overides——重写的方法；

server test——用于自测试的vi；

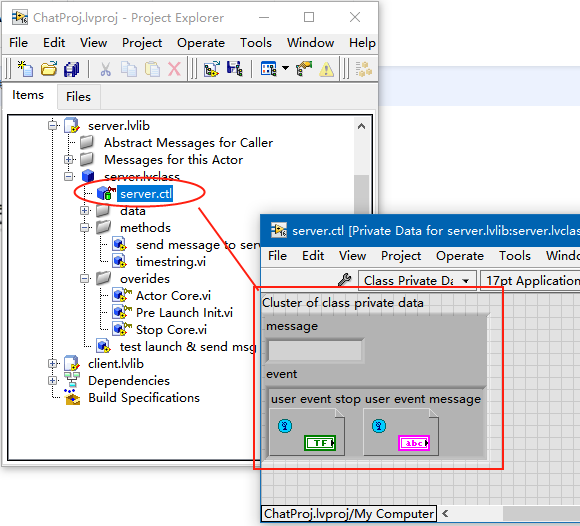
note：当然，你也可以不这么存储和放置，但最好这么做，清晰不会为后面产生混乱与冲突；

此时项目的如下：



## 定义私有数据：

一般将私有数据放到一个簇里面进行管理，这样做是为了方便以后拓展；



如上图：私有数据有分别为:

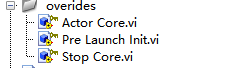
message:字串类型，用来存储server收到的信息，注意，这个信息可以是外部发来的，也可以是自己发送给自己的；比如，我们在写好了这个Acotor server后，可以行单元测试，此时就是自己给自己的actor发送一个消息，看看能否正常收到。

User event top：停止的用户事件

User event message：发送信息的用户事件

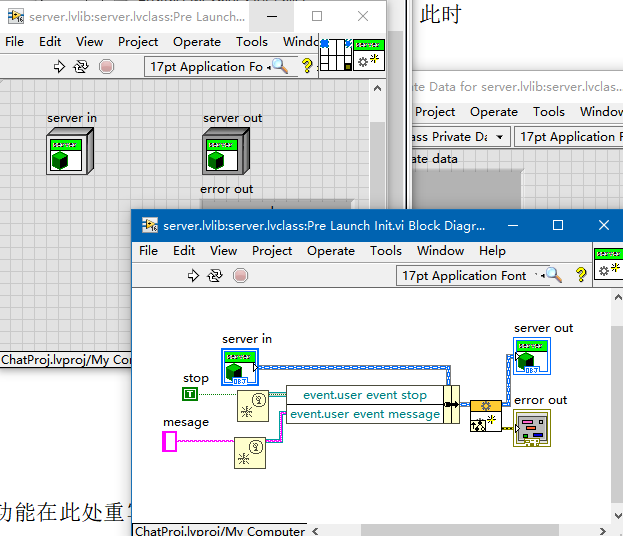
## 重写方法：

## 一般就重写的vi就是这三个：

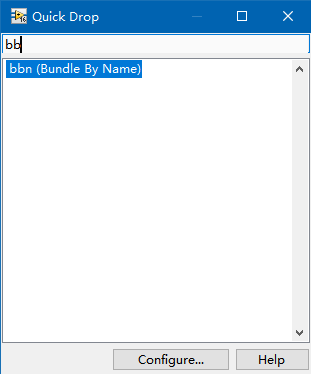


 初始化——core核心（也就是你要实现的功能在此处重写）——stop停止机制

1.Pre Launch Init.vi

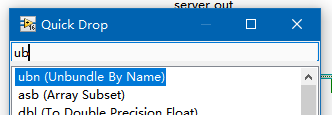


将对象进行初始化，采用Bundle By Name节点进行捆绑；

截图_20230323114825

1. Stop Core.vi

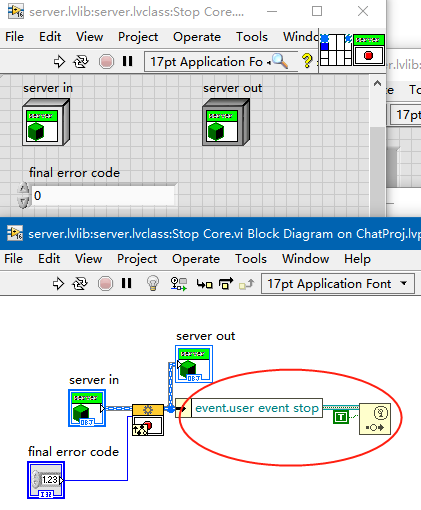
在server out后面采用

截图_20230323115143

来解捆绑，得到用户事件句柄，然后此处产生一个停止事件，这样，该类中所有有响应这个用户事件的都能做出自己的处理。

目前这个例子中，此处的stop用户事件是为了到时候停止Actor Core.vi 中的while循环；

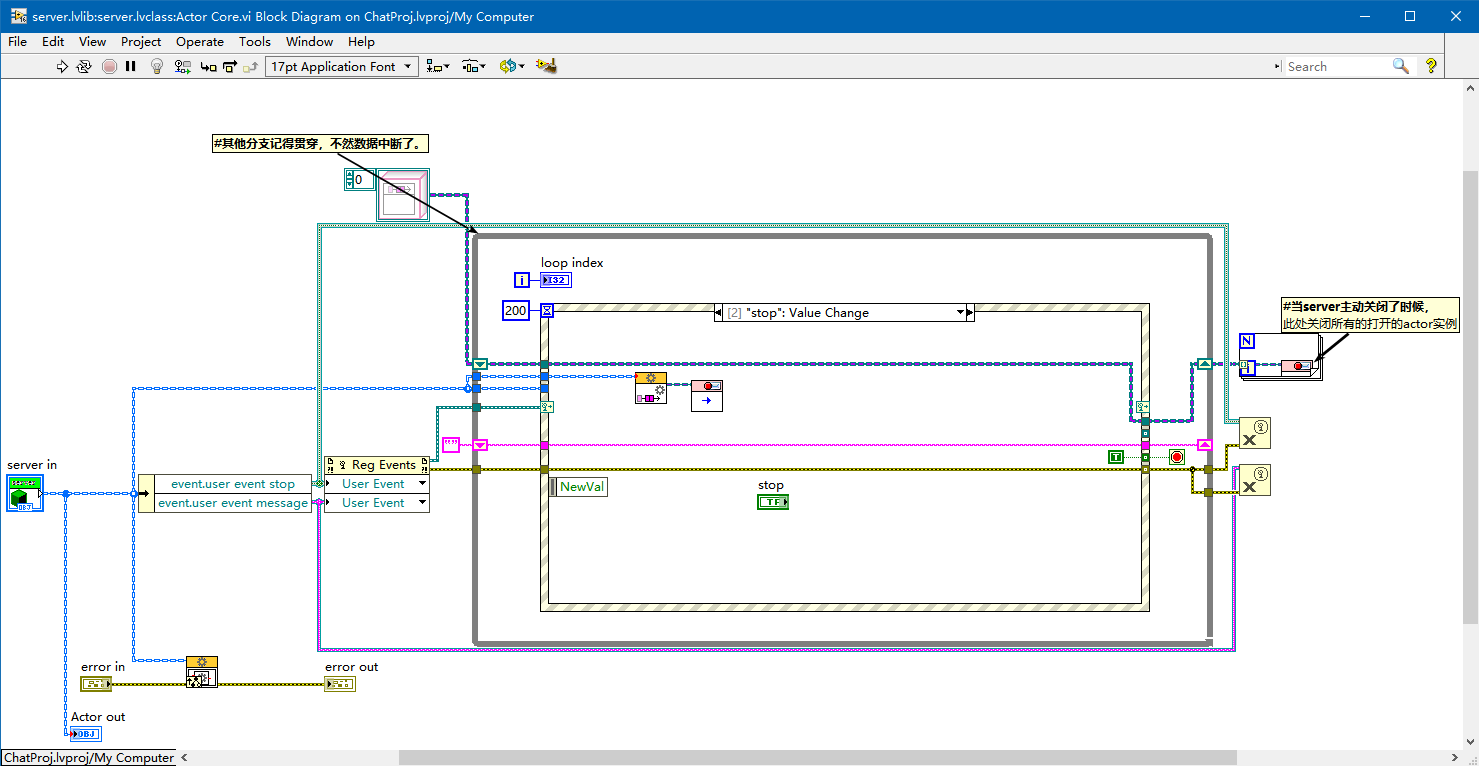
（如果你此处还没有理解， 不用着急，看到后面再回来就能恍然大悟。）

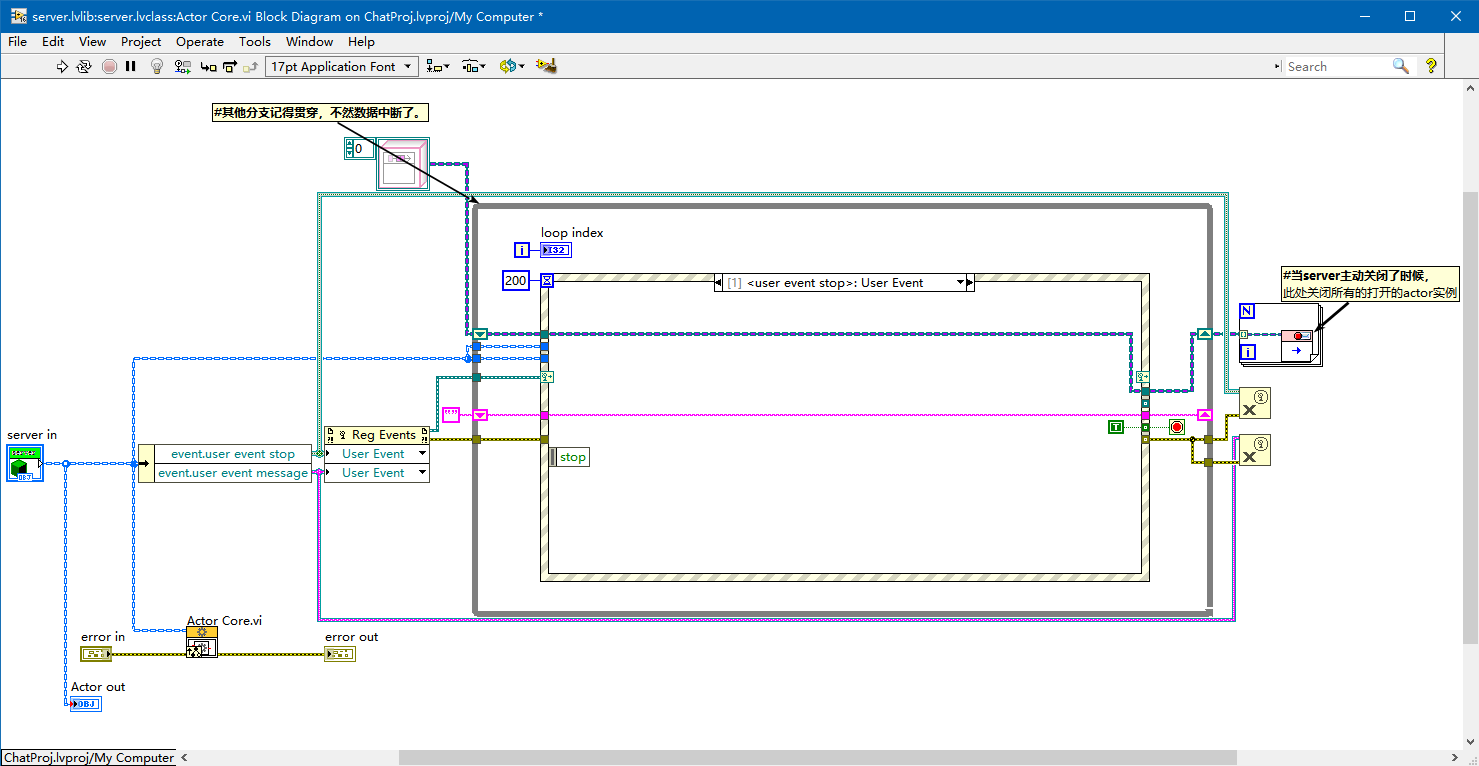


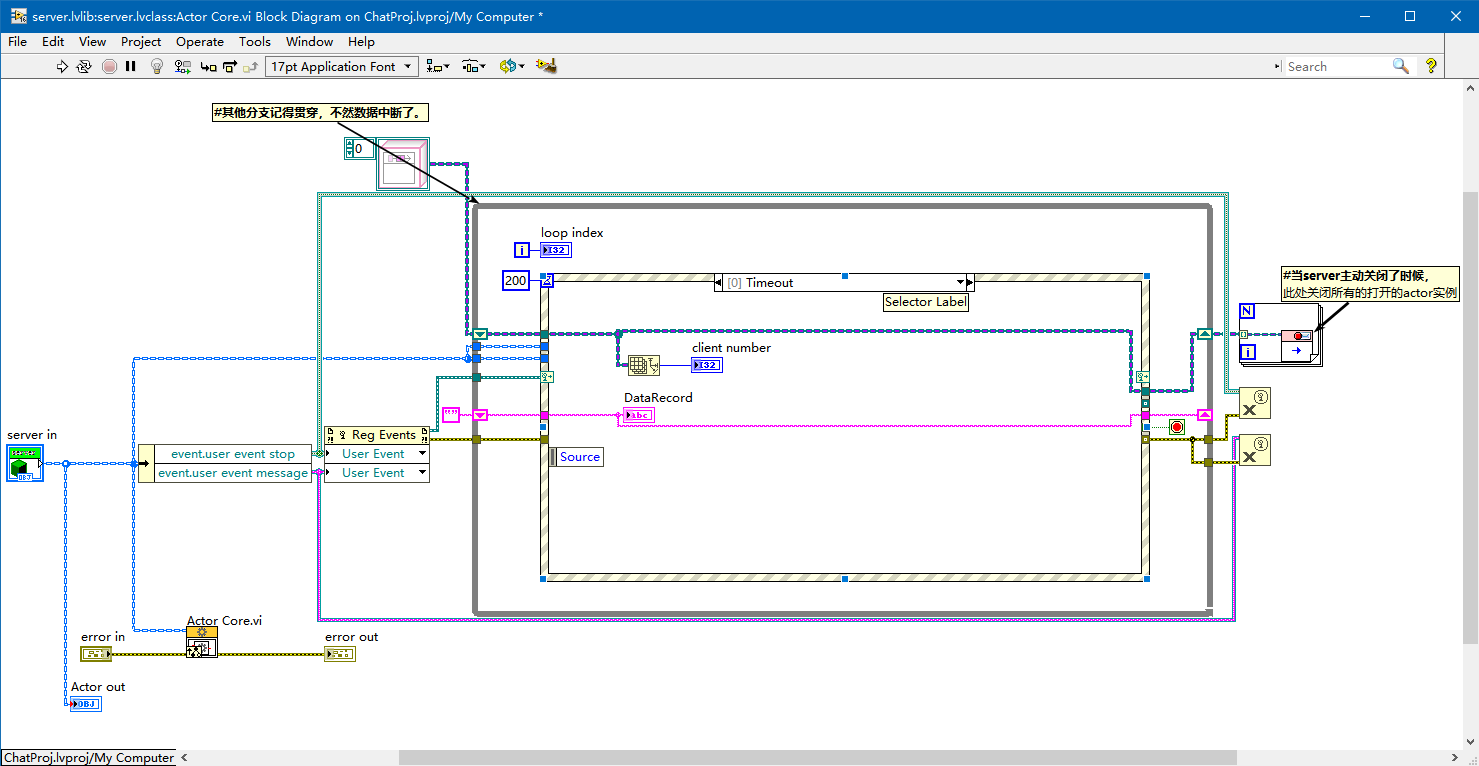
1. Actor Core.vi

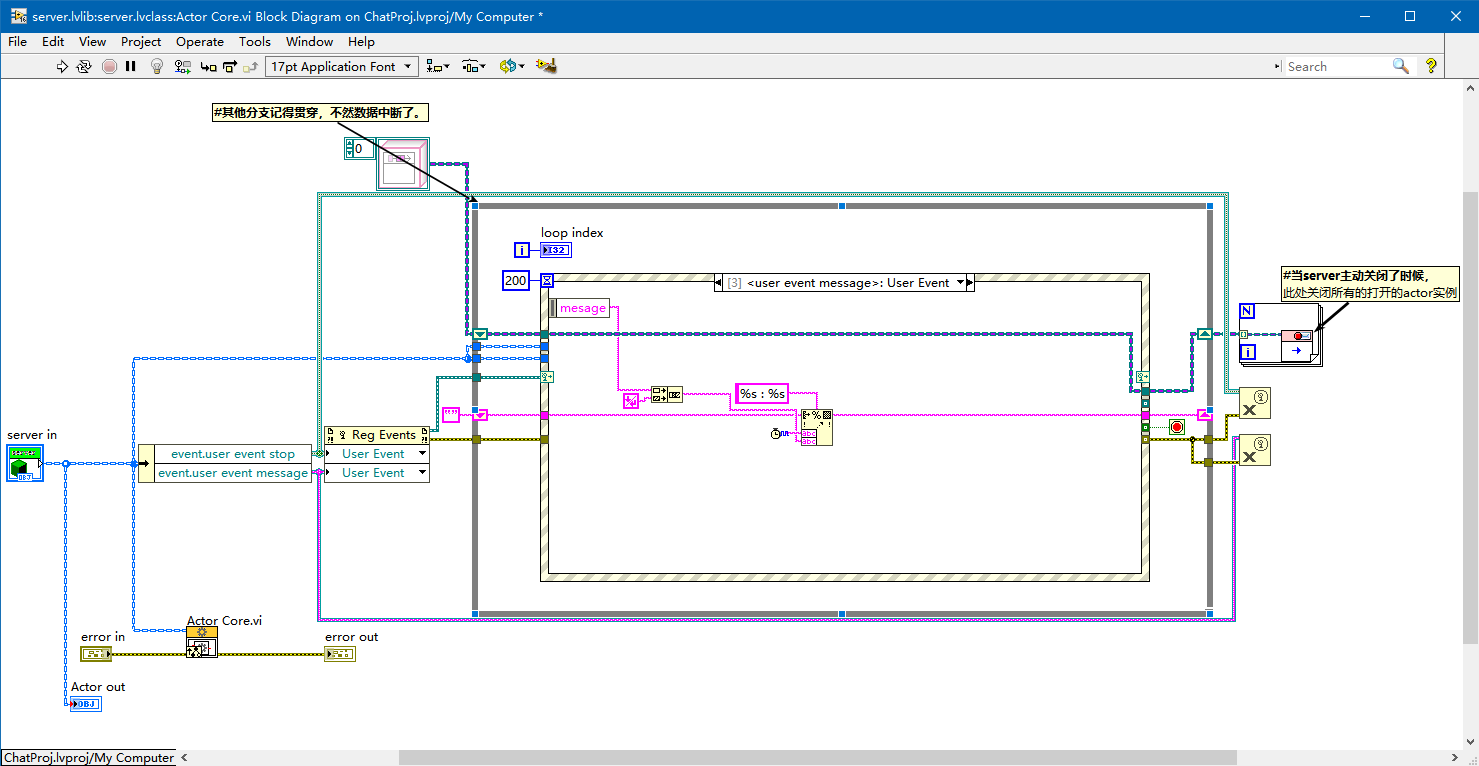
这个便是咱们的actor要实现功能所在的地方。我们需求是要接收到一条消息，然后把它显示记录下来。那么我需要建立一个简单的循环+事件结构，用来响应用户事件，这里有两个用户时间，一个是message，一个是stop；

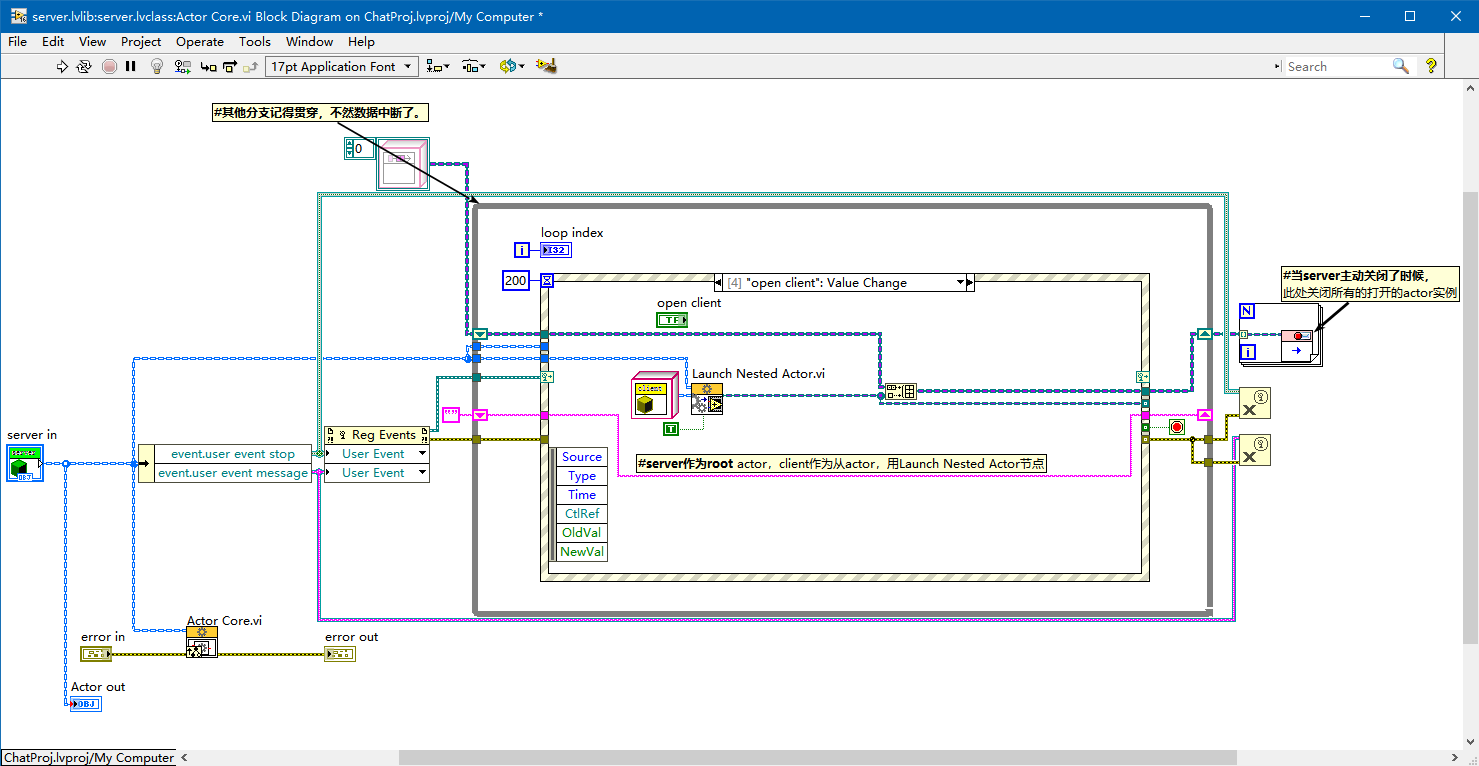
程序框图如下：

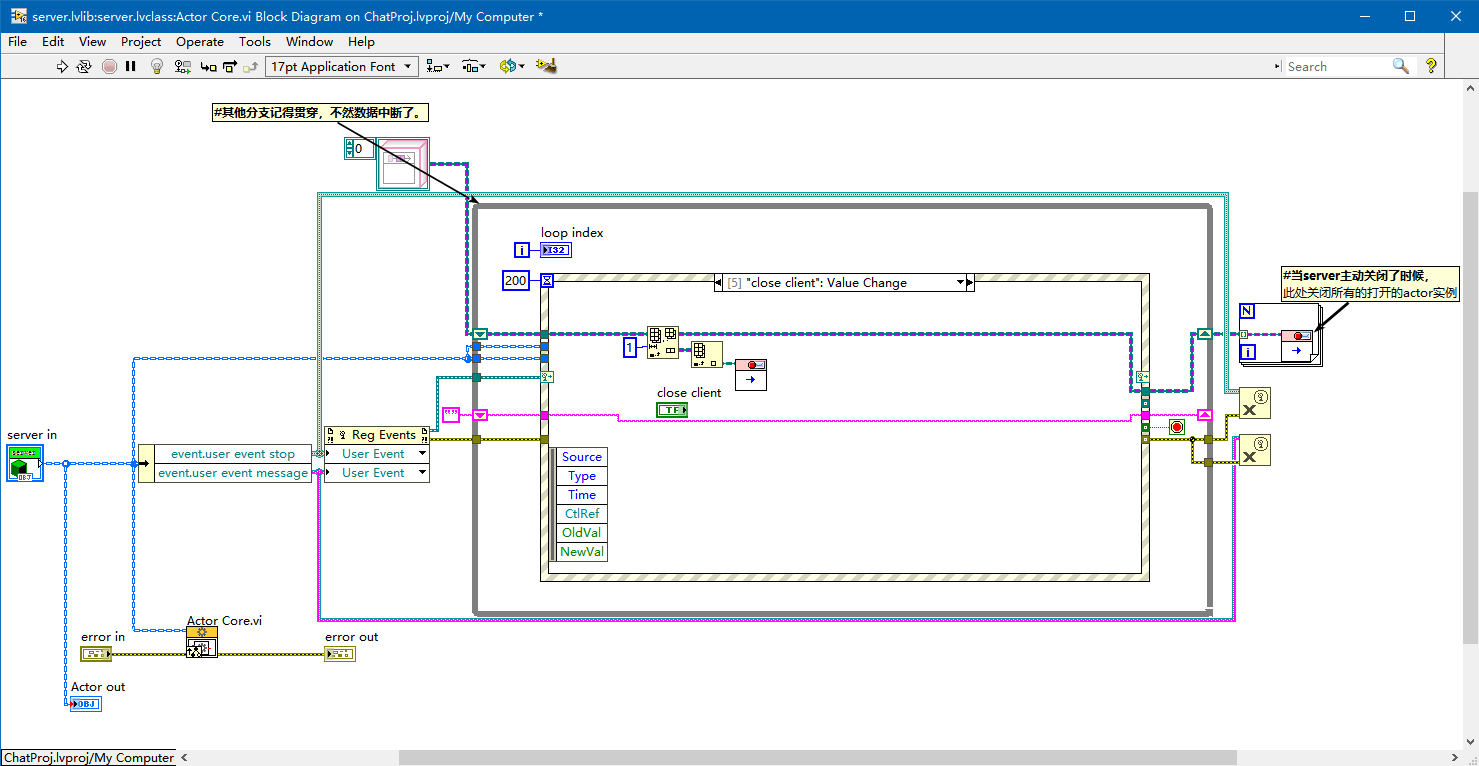












注意的几点：

* 用户事件在使用之前请先行注册，最后程序退出时销毁事件。
* 所有的Actor对象的使用请连线在Actor Framework.lvlib:Actor.lvclass:Actor Core.vi

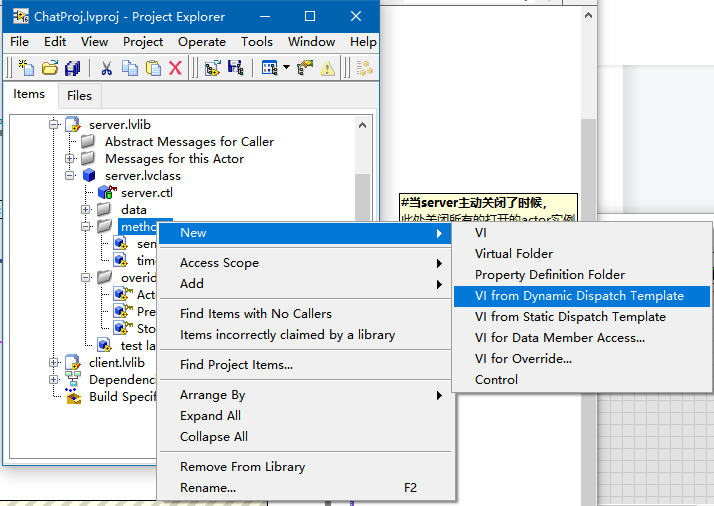
之前，因为父类的这个core也在一直运行，数据流驱动的LabVIEW编程决定这样编写；

* 用户事件<user event message>: User Event ，此处直接可以从事件结构元素message端得到内容的。

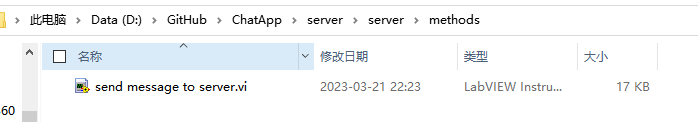
那么message是怎么发送的呢？接下来说明如何**“创建方法的消息队列”**？

## 创建类的消息队列：

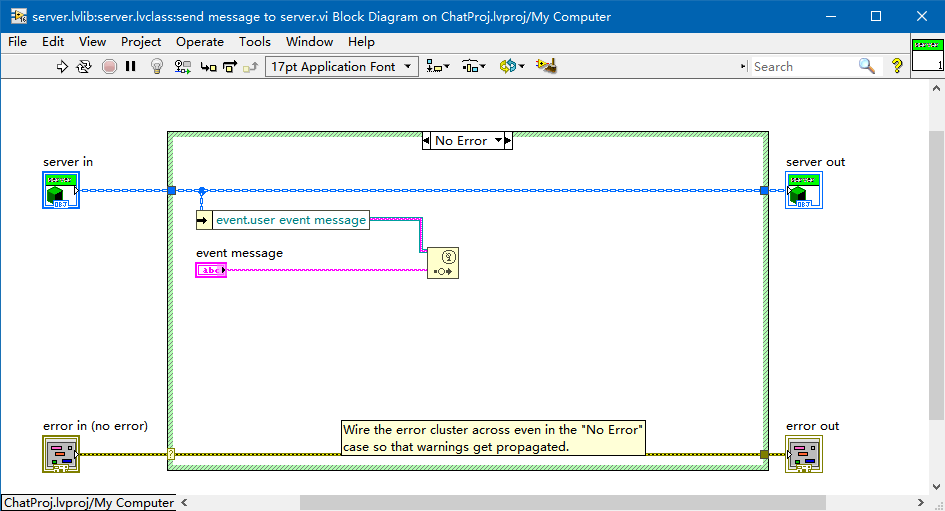
首先在server 类的method虚拟文件夹下，创建动态方法，操作如下图：



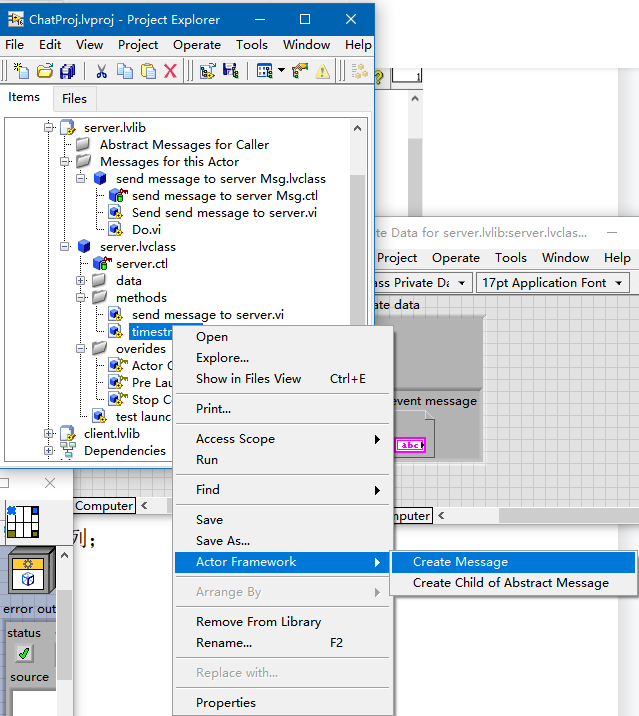
保存自动生成的vi到硬盘实际文件夹的mothods目录：



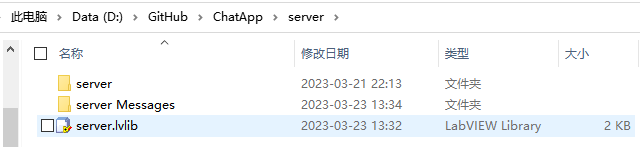
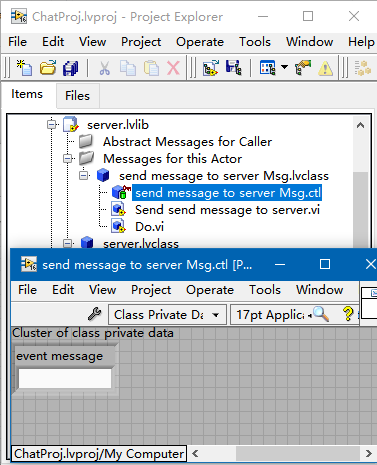
接下来编辑此方法vi，如下：



保存，然后在project中对此方法点击右键创建消息队列；



 创建好了以后，保存到硬盘。一般会在server类同级目录：



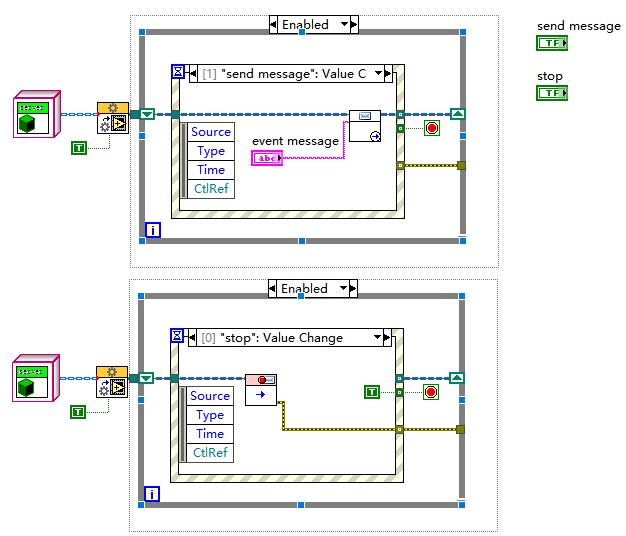
然后在send message to server Msg.lvclass定义私有数据，此处就一个message，我们用一个string输入控件即可；至此，创建send message的消息队列完成。

send message to server Msg.lvclass->Send send message to server.vi便是我们用来给server 这个Actor发送消息的方法。

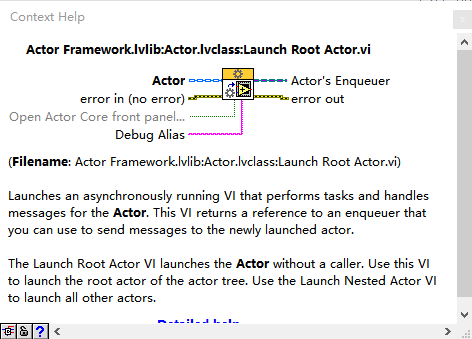
至此，整个server的actor开发完成。

接下来我们编写一个测试vi，用来启动server actor，停止，和发送一个消息测试是否正常？

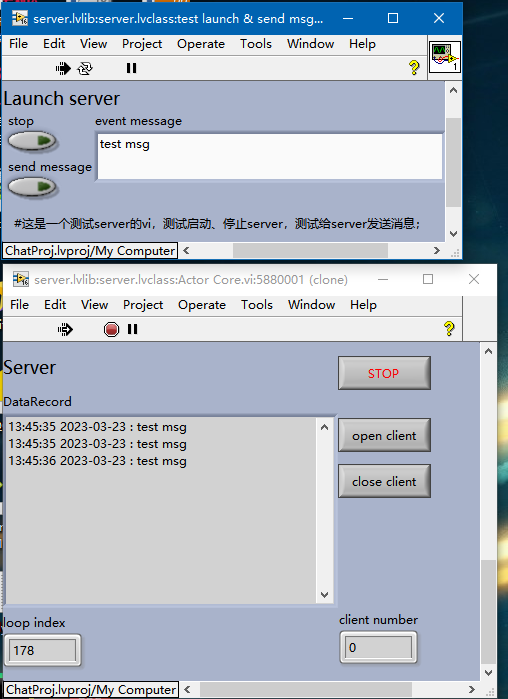
测试vi取名test launch & send msg to server.vi存放在server类的下面：



注意，server作为root来启动，所以采用：



分别测试上述3个功能，这样便能每个Actor自行保证OK；



至此，Actor server测试完成，如果是多人协同开发的时候，此时就可以将自己编写的这个actor发布给别人使用。

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

接下来，我们再来编写client，这部分内容我们下节再讲。