

RemoteActor



,
.
, .



Remote, ,
..
, .

```
akka {  
  actor {  
    provider = "Akka.Remote.RemoteActorRefProvider, Akka.Remote"  
  }  
  
  remote {  
  
    helios.tcp {  
      port = 8001 #bound to a specific port  
      hostname = 127.0.0.1  
    }  
  }  
}
```

App.config , .

- provider : . ()
- helios.tcp : provider , transports tcp / udp .
- ip/port .

DLL

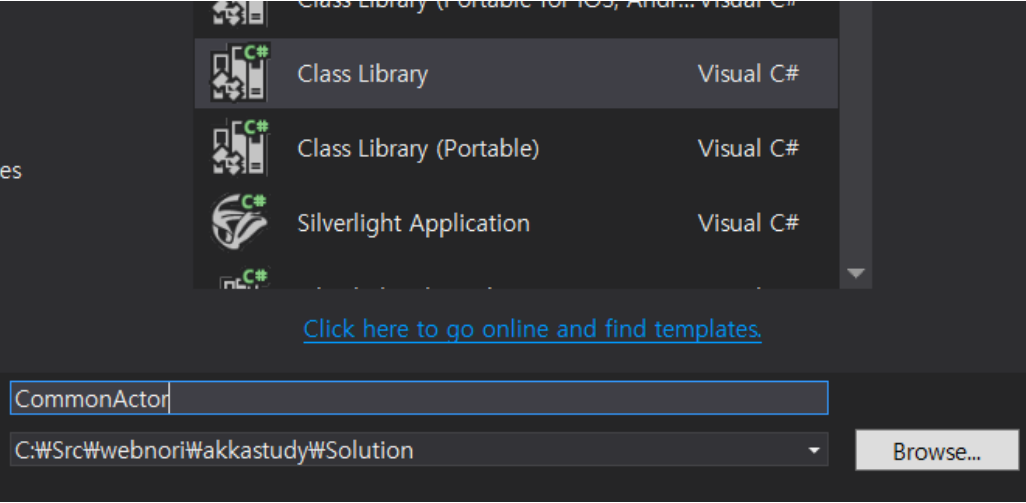
```
public class Hello  
{  
    public Hello(string message)  
    {  
        Message = message;  
    }  
  
    public string Message { get; private set; }  
}  
  
public class EchoActor2 : ReceiveActor  
{  
    public EchoActor2()  
    {  
        Receive<Hello>(hello =>  
        {  
            Console.WriteLine("[{0}]: {1}", Sender, hello.Message);  
            Sender.Tell(hello);  
        });  
    }  
}
```

Class DLL , DLL

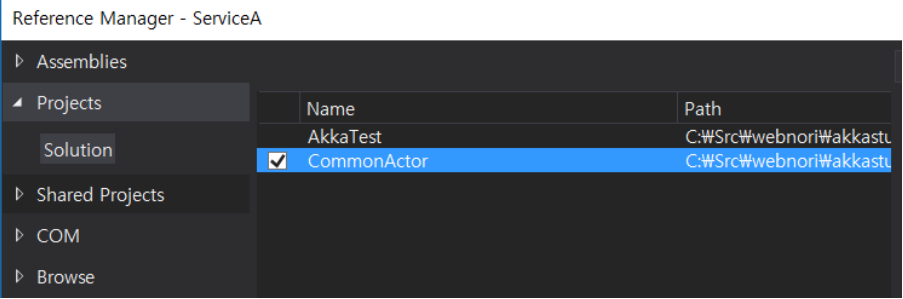
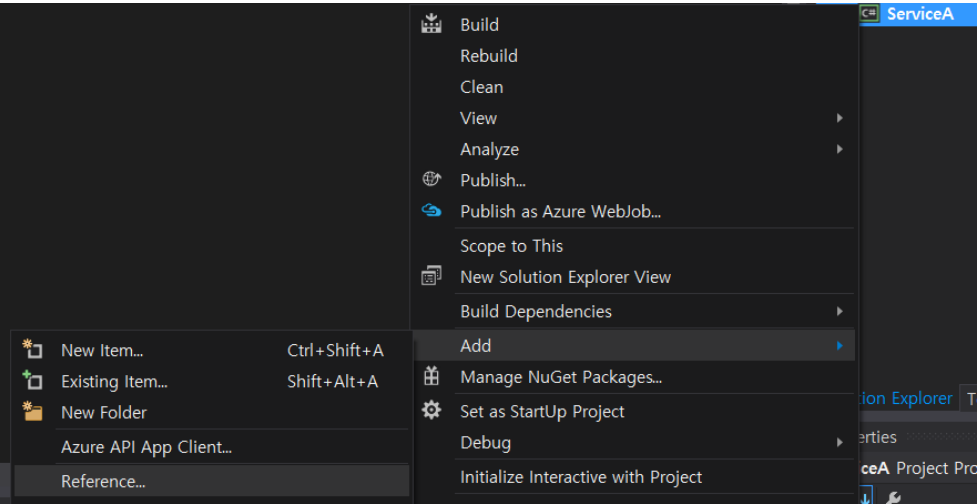
ServiceA / ServiceB

ServiceB ServiceA EcshoActor .

1.Data ClassLib (DLL)



2. CommonActor



DLL . ServiceA , ServiceB

Data .

Actor .

DLL .

Data Data ,

.

JsonSerialization .

Data DLL AKKA.net

ServiceA

```
using (ActorSystem actorSystem = ActorSystem.Create("ServiceA"))
{
    IActorRef myActor = actorSystem.ActorOf<EchoActor>("myactor");
    // ...
} //port 8001
```

ServiceA EchoActor .

ServiceB

```
using (ActorSystem actorSystem = ActorSystem.Create("ServiceB"))
{
    // , ActorSelection .
    ActorSelection otherActor = actorSystem.ActorSelection("akka.tcp://ServiceA@127.0.0.1:8001/user
/myactor");
    Hello msg = new Hello("Hello");
    otherActor.Tell(msg);
    // ...
} // ServiceB 8002
```

All parts form an "ActorPath"



ActorSelection , AkkaSystem .



, (Hello) .

.

, (CPU,) .

(Req) (Res) .