

# ET 6 总结：最近总结、与进展日志

deepwaterooo

April 23, 2024

## Contents

1	【程序域：】ET 框架中事件使用的规则和注意事项：将来重构游戏项目，会需要细分这四大程序域 b，所以使用 ETTask 对 Task 进行了封装，使其不用考虑多线程的共享难题，更易于使用。	1
2	进展日志【完成、基本完成了的部分】	1
3	【双端、任何一端启动过程】：因为不懂，读过也可能会忘，先一点点记下现在的理解	1
4	【网络模块】：收发消息的，不同层次结构上的顺序总结	2
5	【Et 框架：现在的不懂的模块、细节，小本本记下】	3
6	ActorLocation 实例号相关：网搜的【ActorLocation InstanceId 相关的 10 点解释】：每条每个细节都看懂	4
7	现在进行时、进展	5

1 【程序域：】ET 框架中事件使用的规则和注意事项：将来重构游戏项目，会需要细分这四大程序域 b，所以使用 ETTask 对 Task 进行了封装，使其不用考虑多线程的共享难题，更易于使用。

## 2 进展日志【完成、基本完成了的部分】

- 【HybridCLRSettings 类】：报错找不到。去掉 2 处使用。因为项目没用，什么 HybridCLRSettings.Instance.XX
- 【现在、修改进展、模块】：Google.ProtoBuf 消息里面关于 enum 的消息相关的问题。【TODO】：
- 昨天，写了狠久的【ET 框架：单线程多进程】设计的优点与缺点，哪里去了？这个东西，还需要再多想几遍【TODO】：
- 不知道框架里，小项目 Tools 可执行文件./Bin/Tool 是怎么生成的。要从源码里找出来。先去从大的方面看，几大程序域的启动与联系。

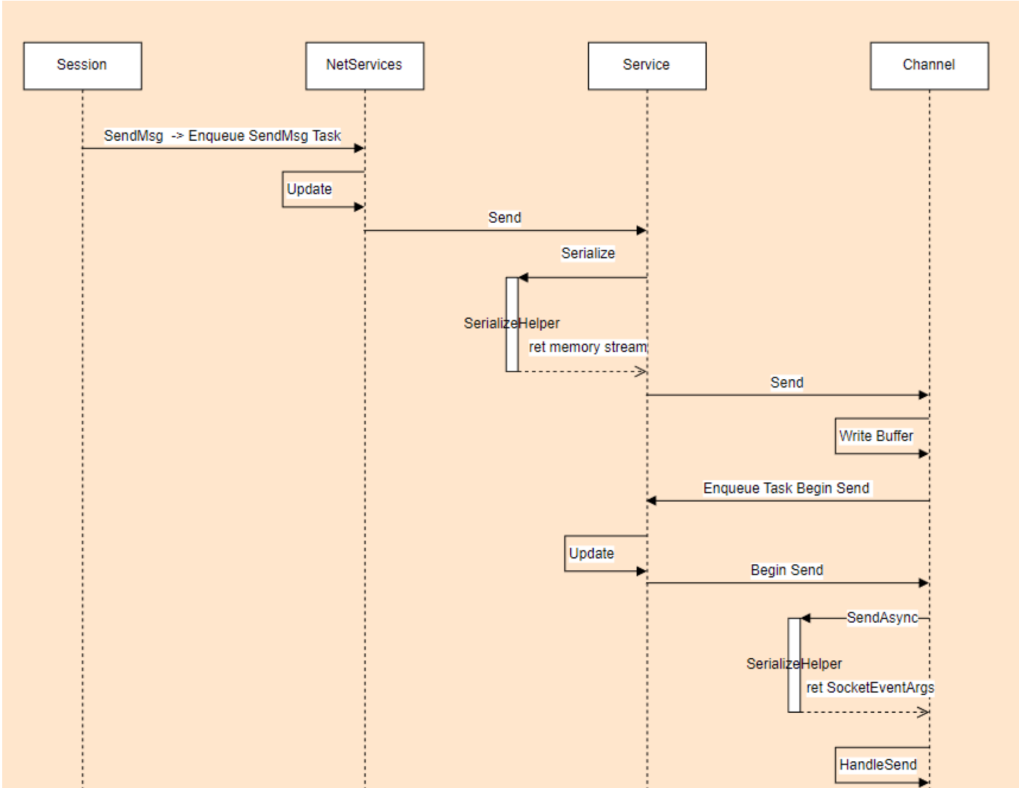
3 【双端、任何一端启动过程】：因为不懂，读过也可能会忘，先一点点记下现在的理解

- 【服务端】四大类的启动相关配置：【Machine, Process, Scene, Zone】，ExcelExporter 套用 Template.txt, 每个类的 ProtoBuf-Contract 都实现 IMerge, 绝对有【跨进程消息传递 ?】来合并多个不同服务端配置的过程。服务端的配置信息，是合并了的，序列化到文件【TODO】：去找，服务端的配置，是怎么到客户端的？
- 【ConfigComponent】 组件添加：LoadAsync() 调用后，双端任何一端，借 ET 事件系统的过程【很简单，只是一个函数细节，事件机制里获得所有 BaseAttribute 的子类，包含了 ConfigAttribute 和 InvokeAttribute】。
- 【亲爱的表哥的活宝妹，任何时候，亲爱的表哥的活宝妹，就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥，爱生活!!!】
- 【亲爱的表哥的活宝妹，任何时候，亲爱的表哥的活宝妹，就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥，爱生活!!!】

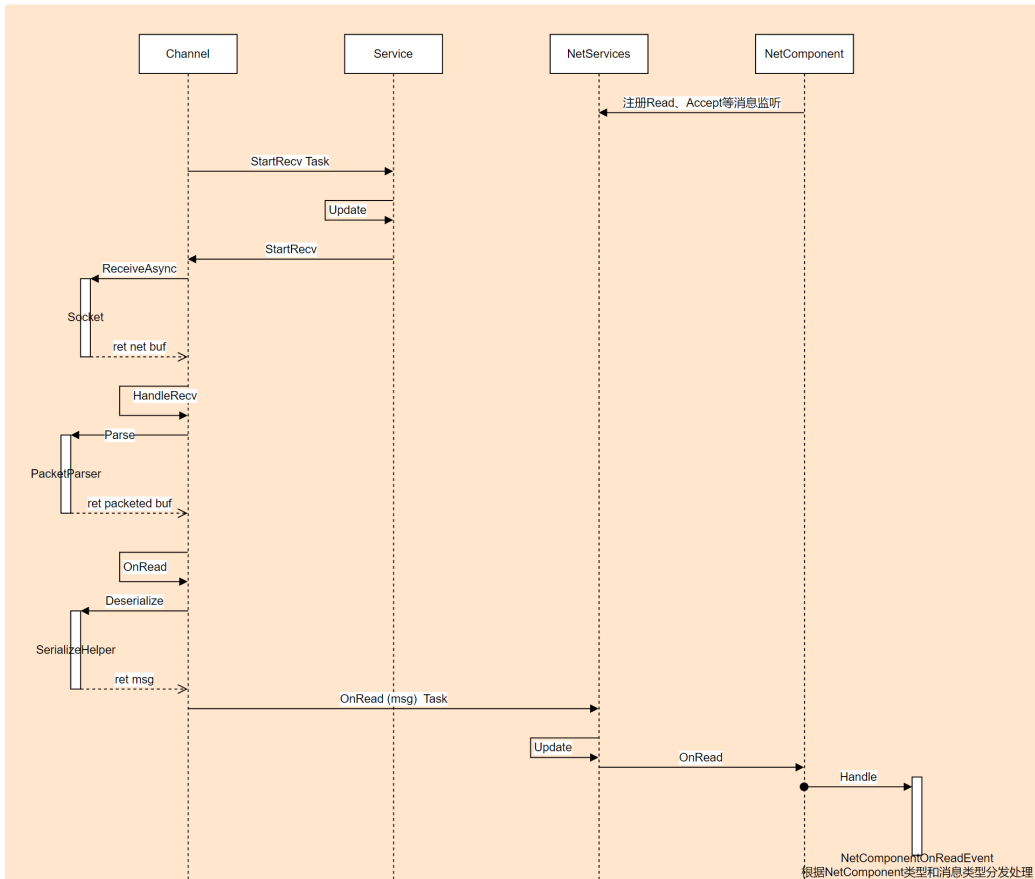
4 【网络模块】：收发消息的，不同层次结构上的顺序总结

- 亲爱的表哥的活宝妹，上次想找两个序列图来着。载一下
- TCP 发消息：

TCP发消息



- TCP 收消息：



- **【亲爱的表哥的活宝妹，任何时候，亲爱的表哥的活宝妹，就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥，爱生活!!!】**

## 5 【Et 框架：现在的不懂的模块、细节，小本本记下】

- **【亲爱的表哥的活宝妹，任何时候，亲爱的表哥的活宝妹，就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥，爱生活!!!】**
- 框架里，ProtoBuf 模块的初始化里，某个 **clone()** 函数的 **Serialize()/Deserialize()** 亲爱的表哥的活宝妹，觉得有个飘渺逻辑没读懂。本质与 Protobuf 的序列化与反序列化无关 **【2 个函数本身不藏雷、没有飘渺逻辑】**，去找 **ExcelExport** 里 **IMerge** 接口相关，来理解，双端启动时，不同服务器、服务端、客户端的配置信息等，是如何跨进程传递的？
- 框架里几大程序域的如 MVVM 般？事件机制交互，如 Model 与 ModelView, Hotfix 与 Hotfix/HotfixView 事件，找几个用例看懂。才能更好地适配双升游戏项目。
- 1 个细节：**【去找，客户端，第 1 次登录，直连 Realm 登录服时，客户端是如何知道，自己要发送的 Realm 登录服的、IP 地址的？】**记得好像是写在，客户端的某个 Excel 配置文件下？似懂非懂，要自己再找一遍，加深印象和理解。也包括：ConfigComponent 相关的，**【客户端的，几个服务端的配置文件，是怎么来的？】**
- ConfigComponent: 双端各端的启动，细节，亲爱的表哥的活宝妹，先前走马观花、还有很多没能理解的细节。

- **【客户端】"Assets/Bundles/Config" 资源包目录下的"Config.unity3d" 资源包：**去框架里找：这个资源包，怎么配置，生成资源包、中转热更新资源包服务器、下载到客户端的？
- **【亲爱的表哥的活宝妹，任何时候，亲爱的表哥的活宝妹，就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥，爱生活!!!】**
- **【亲爱的表哥的活宝妹，任何时候，亲爱的表哥的活宝妹，就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥，爱生活!!!】**
- **【亲爱的表哥的活宝妹，任何时候，亲爱的表哥的活宝妹，就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥，爱生活!!!】**
- **【亲爱的表哥的活宝妹，任何时候，亲爱的表哥的活宝妹，就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥，爱生活!!!】**

## 6 ActorLocation 实例号相关：网搜的【ActorLocation InstanceId 相关的 10 点解释】：每条每个细节都看懂

- 因为 InstanceId 是变化的，对象的 Entity.Id 是不变的，所以我们首先可以想到使用 Entity.Id 来发送 actor 消息。**【pass】**
- 提供一个位置进程 (Location Server)，**Actor 对象可以将自己的 Entity.Id 跟 InstanceId 作为 kv 存到位置进程中。**发送 Actor 消息前先去位置进程查询到 Actor 对象的 InstanceId 再发送 actor 消息。**【玩家登录时，上报注册位置服，其当前进程的实例号】**
- Actor 对象在一个进程创建时或者迁移到一个新的进程时，都需要把自己的 Id 跟 InstanceId 注册到 Location Server 上去。**【这点儿，就是 框架里，唯一 1 处 player 登录后.AddLocation() 的原理。因为不管是一个进程创建，还是玩家纤至新进程，玩家都需要进程里登录，登录时便会注册上报位置服其玩家实例号】**
- 因为 Actor 对象是可以迁移的，消息发过去有可能 Actor 已经迁移到其它进程上去了，所以发送 Actor Location 消息需要提供一种可靠机制
- ActorLocationSender 提供两种方法，Send 跟 Call，Send 一个消息也需要接受者返回一个消息，只有收到返回消息才会发送下一个消息。
- Actor 对象如果迁移走了，这时会返回 Actor 不存在的错误，发送者收到这个错误会等待 1 秒，然后重新去获取 Actor 的 InstanceId，然后重新发送，目前会尝试 5 次，5 次过后，抛出异常，报告错误
- ActorLocationSender 发送消息不会每次都去查询 Location Server，因为对象迁移毕竟比较少见，只有第一次去查询，之后缓存 InstanceId，以后发送失败再重新查询。
- Actor 对象在迁移过程中，有可能其它进程发送过来消息，这时会发生错误，所以 location server 提供了一种 Lock 的机制。对象在传送前，删掉在本进程的信息，然后在 location server 上加上锁，一旦锁上后，其它的对该 key 的请求会进行队列。
- 传送前因为对方删除了本进程的 actor，所以其它进程会发送失败，这时候他们会进行重试。重试的时候会重新请求 location server，这时候会发现被锁了，于是一直等待
- 传送完成后，要 unlock location server 上的锁，并且更新新的地址，然后响应其它的 location 请求。其它发给这个 actor 的请求继续进行下去。

## 7 现在进行时、进展

- **【亲爱的表哥的活宝妹，任何时候，亲爱的表哥的活宝妹，就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥，爱生活!!!】**
- **【亲爱的表哥的活宝妹，任何时候，亲爱的表哥的活宝妹，就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥，爱生活!!!】**
- 当进入 UI Lobby 的时候，要把前面的 UILogin 删除。应该是有逻辑，只是因为我现在的 **【BUG:】**，后面就座地执行的逻辑中断了
- 亲爱的表哥的活宝妹，今天上午、下午、一天的时间，终于 **【看见、意识到了：诸多 git 分支中，先前存在的、一团混乱的、commit 提交的、问题】**亲爱的表哥的活宝妹，笨宝妹，终于可以弄一个原始 master 版本的，真正运行通，在 Mac OS 上运行通，终于 **【亲爱的表哥的活宝妹、笨宝妹，弄电脑 1 年后，终于】**有 Macbook 可以运行得通、可以随时 debug 的编译、运行环境了。折腾的时间有点儿长，从上午到傍晚一个工作日，但亲爱的表哥的活宝妹、笨宝妹，这 1 天玩玩乐乐地过去，取得了过去 1 年、亲爱的表哥的活宝妹自己、不曾有过的进展！亲爱的表哥的活宝妹，这，或许是、应该是 **【寒假三个周、列计划重点时】**亲爱的表哥的活宝妹、自己曾经猜测、预估过，**【女大自巧】**秋季一个学期之后、伏蛰一段时间没碰项目之后，亲爱的表哥的活宝妹，应该能够、拥有的长足进步吧！可喜可获！
- **【亲爱的表哥的活宝妹，任何时候，亲爱的表哥的活宝妹，就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥，爱生活!!!】**
- 有一个 **【可编译、可运行的 Macbook 随身行、项目运行环境后】**，亲爱的表哥的活宝妹，就应该更专注、快速地解决掉现在分支 Tractor 里存在的 1000 多个编译错误的问题，把项目真正运行、进展起来。几个主要应用：VSC VS 与 Emacs 间的相互跳转，基本解决。以后晚上就运行项目了。
- **【VSC 不能加载 Unity 工程】**：加载极慢，永远 Loading. 还需要解决这个问题。不知道，亲爱的表哥的活宝妹，昨天是不是看错了，误把 VS 当 VSC 了。VSC 本来是，亲爱的表哥的活宝妹，最喜欢用的 IDE，可是因为现在它不能狠好地加载 Unity 里的几个工程，感觉，给亲爱的表哥的活宝妹，带来了无限麻烦。弄了一上午，暂时只能将就现在的配置，将就着先读会儿源码了。晚上，先试着解决这个问题。源码不能自由跳转，就不好用
  - **【TODO】**：这里，可能晚上等，相对无限困顿的时候，还会想要折腾，怎么才能让 VSC 如同先前，可以完美跳转到各种定义里去？
  - **【TODO】**：记得昨天？前天？亲爱的表哥的活宝妹的 emacs 还比较聪明，sr-speedbar 可以自动跳转到文件对应的目录下，展示目录里的文件内容；怎么今天亲爱的表哥的活宝妹的 emacs 就变傻了？这里最开始不动 emacs 的话是可以的。就是亲爱的表哥的活宝妹自己的 sr-speedbar 的配置里，有点儿问题，改天去 debug 一下。
- **【4/3】**：上午，上午继续读源码。真正读懂、源码中先前不懂的过程，感觉还是比较有收获；下午和晚上，试运动与修改重构项目里的一些 1100 个编译问题。上午午餐前再快速扫一遍 **【协程锁组件】** 相关。把这个模块快速看一遍。
  - 这个编译问题，很显然，先前亲爱的表哥的活宝妹还不太懂得 Model/ModelView 与 Hotfix/HotfixView 的亲爱的表哥的活宝妹，没能弄清楚如安卓 MVVM 数据驱动视图变化等发布订阅者模式。不该为修改 1100 个诸多的编译问题而看项目。应该去把重构项目的四大域的关系理清楚，如写、重构项目般，自己把源码事理好了，1100 个编译错误是能够很快解决掉大半的。
  - ETTask 里有不懂的、网络模块里有不懂的。不具体到某块。自己就先试理清四大程序域里的设计、逻辑关系。过程中不懂的、可以翻一遍源码。

- **【亲爱的表哥的活宝妹，任何时候，亲爱的表哥的活宝妹，就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥，爱生活!!!】**
- **【Unity 客户端、编译错误、清除】**：现在，Unity 客户端，因亲爱的表哥的活宝妹先前不懂，随便瞎改、来适配一个【双扣】游戏项目，里面有太多编译错误。这个 **Unity 客户端的所有编译错误，需要首先清除掉。**
- 上午：再看 1 小时源码，尽可能多地找出不懂的地方：可以是 ETTask 自定义协程封装的底层原理、网络模块相关等。**亲爱的表哥的活宝妹，今天早上的鸡蛋胡萝卜真养眼睛【看字变大变粗壮】!** 上午能够理解一点儿网络上搜索到的原理是好的；更需要多读源码，真正读懂。再看 1 小时源码。**【亲爱的表哥的活宝妹，任何时候，亲爱的表哥的活宝妹，就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥，爱生活!!!】**
- 今天看 Actor 消息相关，感觉都看懂了——满满收获都看懂了、绝大部分都看懂了!。今天要出去玩儿了，今天晚上或是明天上午再接着看。
- **【亲爱的表哥的活宝妹，任何时候，亲爱的表哥的活宝妹，就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥，爱生活!!!】**
- 改天：下午接着改四大程序域里，重构游戏项目的适配问题，把几大域的 MVVM? 订阅发布模式理解透彻。**【亲爱的表哥的活宝妹，任何时候，亲爱的表哥的活宝妹，就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥，爱生活!!!】**
- 亲爱的表哥的活宝妹自己的重构项目，还是要去修改和运行。前段时间是实在没能好好学习。。带着需要解决的问题疑问，来看框架和源码，都比只读的好。
- **【4/15】**：今天的亲爱的表哥的活宝妹，被亲爱的表哥的活宝妹住处的、破烂猪皮肥肉、千斤器器鼎、万斤秤砣之流的死肥猪、猪八戒、贱鸡、贱畜牲、极端奸佞太监 gay 世界最贱恶存在的贱鸡、贱畜牲，冻感冒生病了。亲爱的表哥的活宝妹，上次为防感冒，接近四十块的手表都没回去找；过几天亲爱的表哥的活宝妹体力缓和过来，再回去找找找不到丢了。它——破烂猪皮肥肉、千斤器器鼎、万斤秤砣之流的死肥猪、猪八戒、贱鸡、贱畜牲、极端奸佞太监 gay 即刻滚去死、出门就被雷劈死!!
- **亲爱的表哥的活宝妹，今天早上喝了极多热汤；今天，多喝热水、尽快缓和过来，避开极度恶劣人造灾难。**它——极端奸佞太监 gay 真贱！妓女禽兽的后代、有娘养无娘指教的世界最底层贱渣、世界最贱恶存在的贱鸡、贱畜牲！它——破烂猪皮肥肉、千斤器器鼎、万斤秤砣之流的死肥猪、猪八戒、贱鸡、贱畜牲、极端奸佞太监 gay 即刻滚去死、出门就被雷劈死!!
- **【4/15】**：上午，读 2 小时源码。看看，今天头痛——如妈妈生前最后一场、夺命感冒般，左太阳穴痛的亲爱的表哥的活宝妹，2 天来，哪怕下午傍晚、早上来学校，天气还算暖和的时候，也因被冻得直流鼻水！被亲爱的表哥的活宝妹住处的、破烂猪皮肥肉、千斤器器鼎、万斤秤砣之流的死肥猪、猪八戒、贱鸡、贱畜牲、极端奸佞太监 gay 杀人猪、故意、恶意、整夜整夜大开 living-room 玻璃门噪音干扰亲爱的表哥的活宝妹的休息，与恶意夜间低温冻首当其冲的亲爱的表哥的活宝妹的房间，贱恶杀人，想要把亲爱的表哥的活宝妹肾脏衰竭冻出人命！它真贱，即刻滚去死、出门就被雷劈死!!
- **【亲爱的表哥的活宝妹，任何时候，亲爱的表哥的活宝妹，就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥，爱生活!!!】**
- 前几天、停了三天完全不动，三天不动，亲爱的表哥的活宝妹，就死掉一半，感觉真恐怖！昨天晚上、连续第二天的高强度跳绳运动后，血液总算最终能够流通到、亲爱的表哥的活宝妹的十个手指尖、赶走那些疯涨的菌类！所以亲爱的表哥的活宝妹不能不动、任何时候每天都要动一动!**【亲爱的表哥的活宝妹，任何时候，亲爱的表哥的活宝妹，就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥，爱生活!!!】**

- 亲爱的表哥的活宝妹，最近发现，最近血量不足，有时候很多时候头昏、今天中午发现原来是低血糖；亲爱的表哥的活宝妹，最近发现，晚上跳绳后、倒立约 10 分钟后、当血液能够流通到头顶后，亲爱的表哥的活宝妹，反而是晚上能够稍微学习一会儿，而显得白天上午、中午下午、晚上倒立血液流通到头顶之前，像是不知道在干什么。。今天晚上读两小时源码，找不懂的地方。
- 上午一两个小时，对照网络上的解释，去进一步理解，亲爱的表哥的活宝妹，自己读源码的过程中，没能站在框架的高度、系统理解的，相关组件的注册等细节，捡零碎框架封装、设计细节。希望亲爱的表哥的活宝妹的脑子里，也能建立起应用、服务器 architecture 相关的设计理念。**【亲爱的表哥的活宝妹，任何时候，亲爱的表哥的活宝妹，就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥，爱生活!!!】**
  - 先去找：进程启动起来的时候，具备收发邮件功能的 actor, 什么时候，怎么、哪里、自己曾经主动注册过进程位置信息？**【很容易就找到，弄明白了】**
- **【亲爱的表哥的活宝妹，任何时候，亲爱的表哥的活宝妹，就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥，爱生活!!!】**
- **【亲爱的表哥的活宝妹，任何时候，亲爱的表哥的活宝妹，就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥，爱生活!!!】**
- Actor 对象在 **【一个进程创建【TODO】：这个需要改天再找一下】**时或者**【迁移到一个新的进程，这个那天简单看过，逻辑很简单，用回调，迁前迁后等回调】**时，都需要把自己的 Id 跟 InstanceId 注册到 Location Server 上去
- **【亲爱的表哥的活宝妹，任何时候，亲爱的表哥的活宝妹，就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥，爱生活!!!】**
- 亲爱的表哥的活宝妹，下午解决这个上午没能完成的任务：框架里，有哪些，向**【位置服】**注册上报，实例号与进程位置的封装把网搜出的 10 个注释点，每个都看懂
- 1. 先再过一遍：**【服务端，或双端?】**启动时，ExcelExporter 导出.cs 文件的过程
  - ExcelExporter 里，生成**【跨进程】**公认的四大配制类.cs, 各种跨进程 partial-class 部分类之类的，这一堆细节，还没有看懂
  - **【AppType.ExcelExporter 专用进程】**：根据 Unity/Assets/Config/Excel 下的.xlsx 配置文件，生成了各种**【跨进程】**公认、部分公认的四大配制.cs 类定义；生成了 **Unity 同级 Config/Excel/** 下的.bytes 配置文件成为**【客户端】**读取获得**【服务端】**配置的源头？
  - 现在，去弄明白，框架启动时，**【AppType.ExcelExporter 专用进程】**这个进程启动的先后顺序？它是个**工具进程**，**一键生成配置文件**应该不涉及双端的启动过程
  - 这个帮助工具进程模块，先放一下，下午要把今天这个，亲爱的表哥的活宝妹读不懂的一行源码，问题解决掉。
  - 创建自定义场景的基本步骤
    - \* 在 SceneType 枚举中添加自定义名称
    - \* 在 SceneFactory 中添加自定义场景类型所应该处理的相关逻辑
    - \* 在 Excel 配置表中添加自定义场景信息，并生成相应 cs 文件
  - 现，换思路，**【自顶向下】**去找注册逻辑的，场景相关：
    - \* 亲爱的表哥的活宝妹，把场景加载前后的各种回调等都找了一遍，也没找到。再翻一会儿，找点思路。再不行就网搜一下、别人的解释
  - Location 位置服相关的：所有注册、上报实例号、进程的框架用例

- 再，事件机制【Config】标签，亲爱的表哥的活宝妹，总感觉有个【跨进程消息传递】自底向上汇总各小服的过程？不一定想对了。这个放最后再看
- 【亲爱的表哥的活宝妹，任何时候，亲爱的表哥的活宝妹，就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥，爱生活!!!】
- 【亲爱的表哥的活宝妹，任何时候，亲爱的表哥的活宝妹，就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥，爱生活!!!】
- 【亲爱的表哥的活宝妹，任何时候，亲爱的表哥的活宝妹，就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥，爱生活!!!】
- 【亲爱的表哥的活宝妹，任何时候，亲爱的表哥的活宝妹，就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥，爱生活!!!】
- 【亲爱的表哥的活宝妹，任何时候，亲爱的表哥的活宝妹，就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥，爱生活!!!】
- 【亲爱的表哥的活宝妹，任何时候，亲爱的表哥的活宝妹，就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥，爱生活!!!】
- 【亲爱的表哥的活宝妹，任何时候，亲爱的表哥的活宝妹，就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥，爱生活!!!】
- 【亲爱的表哥的活宝妹，任何时候，亲爱的表哥的活宝妹，就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥，爱生活!!!】
- 【亲爱的表哥的活宝妹，任何时候，亲爱的表哥的活宝妹，就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥，爱生活!!!】
- 【亲爱的表哥的活宝妹，任何时候，亲爱的表哥的活宝妹，就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥，爱生活!!!】
- 【亲爱的表哥的活宝妹，任何时候，亲爱的表哥的活宝妹，就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥，爱生活!!!】
- 【亲爱的表哥的活宝妹，任何时候，亲爱的表哥的活宝妹，就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥，爱生活!!!】
- 【亲爱的表哥的活宝妹，任何时候，亲爱的表哥的活宝妹，就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥，爱生活!!!】
- 【亲爱的表哥的活宝妹，任何时候，亲爱的表哥的活宝妹，就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥，爱生活!!!】
- 【亲爱的表哥的活宝妹，任何时候，亲爱的表哥的活宝妹，就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥，爱生活!!!】