# AicCommu开发文档

## 目的

为解决公司软件应用层一些基础网络问题, 如断连检测、重连、跨平台统一接口等问题。

## 需求

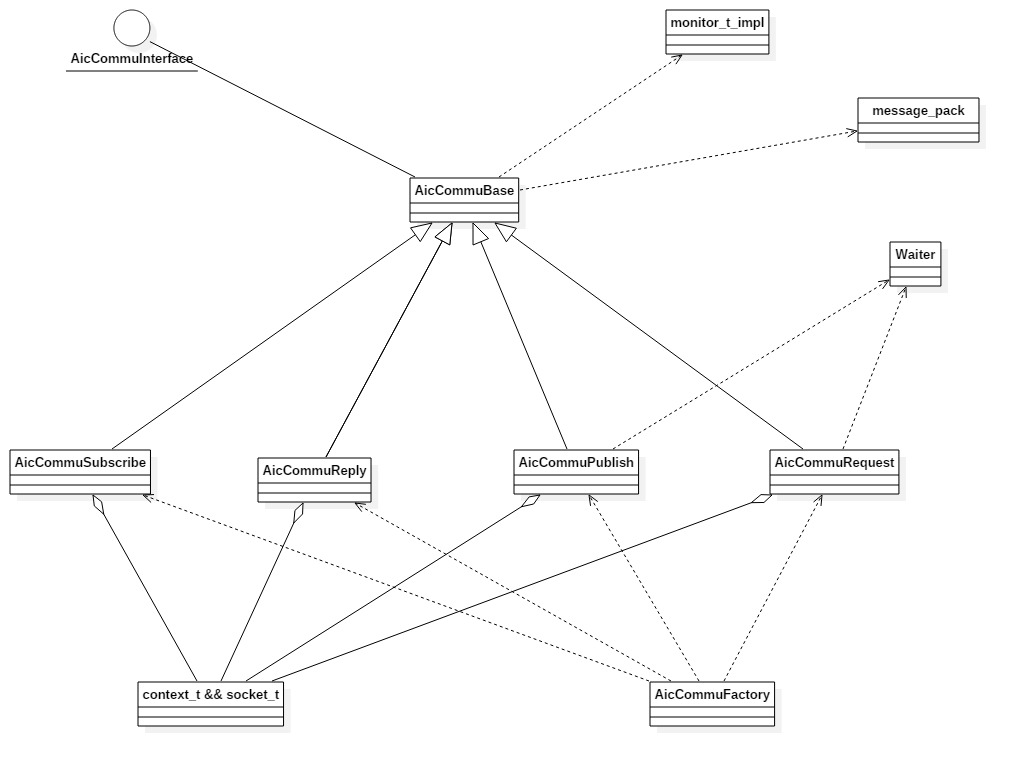
1. 适用平台：linux\windows。
2. 呈现形式：so和dll，两个平台的接口要统一。
3. 设计准则：易用、稳定、规范，使调用者只需关心业务的处理，而不用关心任何网络连接相关问题。
4. 以zmq git最新版库为底层通信实现，protobuf3.5.1为数据封装格式，业务层数据格式可以使用者自定义。需要支持请求应答、推送订阅模式。通讯库能捕获断线和重连消息，通知给调用者。
5. 调用者使用本通讯库只需要依赖zmq，protobuf。
6. 发送和接收数据不能阻塞用户线程，尽量用回调方式，把结果或数据传送给调用者。
7. 定义好各种错误码和错误信息。
8. 调用者可以设置打印函数，库中的各种打印消息都可以通过调用者的函数输出。
9. 调用者可以通过选项设置是否以文本形式打印传输的数据。
10. 每个数据传输（请求包、返回包、通知包）都要有数据头部，库自动生成。
11. 头文件每个函数使用说明注释清楚。
12. 使用c++11标准，不要出现new操作符。
13. 按照google c++开发规范编码。
14. 写一个linux和windows版的使用demo。

## 设计

### 3.1选择模型

因需求指定的底层网络库 libzmq 对底层细节进行抽象，分类出多种套接字模型，为保证使用方无需关心通讯库的具体实现，而采用工厂模式分离接口和实现，为不同的使用目的提供统一的接口。

### 3.2 UML



### 3.3 类

**context\_t、socket\_t、monitor\_t** 类：由于 zmq 库是C语言实现的，作者封装成OOD类型。

**monitor\_t\_impl** 类：继承自 monitor\_t 类，负责监控套接字事件并返回通知。

**message\_pack** 类：protobuf 生成的类，用于封装通讯消息。

**Waiter** 类：封装条件变量用于等待事件发生。

**AicCommuInterface** 类：通讯库接口类，为使用者提供统一接口，纯虚类。

**AicCommuFactory** 类：工厂类，负责创建不同类型的套接字对象，并绑定到接口类。

**AicCommuBase** 类：继承自 AicCommuInterface 类，抽象了各种类型套接字的通用实现部分。

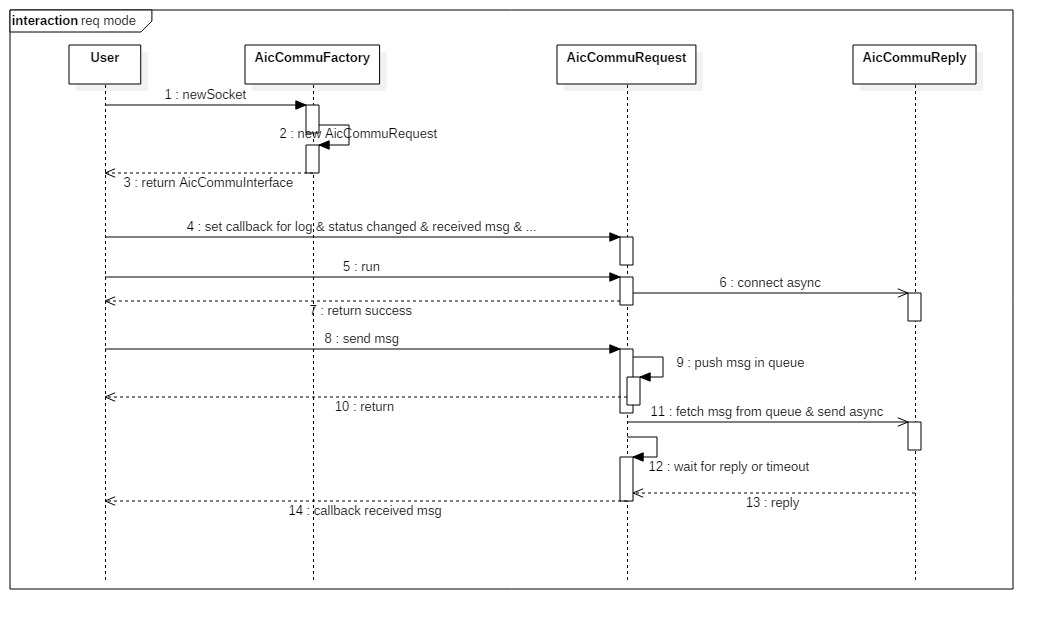
**AicCommuRequest** 类：继承自 AicCommuBase 类，实现请求模式功能。

**AicCommuReply** 类：继承自 AicCommuBase 类，实现应答模式功能。

**AicCommuPublish** 类：继承自 AicCommuBase 类，实现发布模式功能。

**AicCommuSubscribe** 类：继承自 AicCommuBase 类，实现订阅模式功能。

### 3.4 时序

以request模式一次成功通讯为例：  


1. 第1~5步基本是每种类型的套接字通用的。
2. 第4步还可以设置poll超时、心跳、是否打印通讯消息等，非必须故省略。
3. 第5步可能因某些原因失败。
4. 第12步poll超时会**清空待发送队列(仅request模式)**，并通过回调通知用户。

reply模式只是等待请求并回调应答；

publish模式只发布消息，不接收；

subscribe模式只订阅消息，不发送；

这3种模式时序较简单，可自行阅读代码理解。

### 实现

代码有详尽注释，不再赘述。