**PPserver 目录结构、编译使用说明**

# 一、代码目录结构

## 1、工程代码第一层目录结构说明

[root@localhost ppserver-mysql]# tree -L 1

.

├── bin #编译生成的所有可执行都在bin目录下

├── build.sh #命令脚本

├── cron.sh #crontab监听脚本

├── etc #所有配置文件存放目录

├── lib #本程序指定lib库文件目录，脚本自动添加库，查找库

├── make #其他目录下makefile源文件目录

├── makefile #编译时makefile，最初的make

├── makefile.mk #编译时的全局环境变量

├── obj #编译时生成的\*.o 和lib/\*.so存放目录

├── src #源码目录

└── log #日志目录

8 directories, 4files

[root@localhost ppserver-mysql]#

## 2、工程代码第一层结构lib目录说明

[root@localhost ppserver-mysql]# tree -L 1 lib/

lib/

├── libconf.so #配置文件lib库，源码路径src/config 需要单独编译

├── libhsm.so #

0 directories, 3 files

## 3、工程代码第一层结构模块文件夹的第一层目录说明

[root@localhost ppserver-mysql]# tree -L 1 make/

make/

├── makefile-bin #生成可执行目标文件makefile

├── makefile-lib #生成lib库目标文件makefile

└── makefile-src #模块源码生成.o文件makefile

阅读make目录下readMe.txt

## 4、工程代码第一层结构obj目录说明

0 directories, 4 files

[root@localhost ppserver-mysql]# tree -L 1 obj/

obj/

0 directories, 0 files

## 5、工程代码第一层结构src目录说明

[root@localhost ppserver-mysql]# tree -L 1 src/

src/

├── config #配置文件解析源码,这个模块需要单独编译

├── DlApi #动态加载动态库模块

├── include #头文件

├── log

├── mainFrm #PPserver　main函数

├── makefile

├── PlatformUtil

├── readMe.txt

├── testBin

└── UnionDBModual

8 directories, 2 files

[root@localhost ppserver-mysql]#

如果不想编译那个目录模块，可以查看当前makefile 参照想关配置ELIMINATE\_LIST。

添加模块查看src目录下readMe.txt

## 6、工程代码第一层结构build.sh脚本说明

* 编译mysql版本命令

./build.sh makefile　或

./build.sh makefile mysql

* 编译oracle版本命令

./build.sh makefile oracle

* 清除编译结果命令

./build.sh clean

* 启动服务

./build.sh start

* 停止服务

./build.sh stop

* 重启服务

./build.sh restart

* 编译指定目录mysql版本命令

./build.sh makefile　dir或

./build.sh makefile dir mysql

# 二、添加应用模块

添加的源码都在src目录下。

## 1、添加基础公共代码

第一步：在src目录查找是否有合适的目录模块，

　　有，直接添加\*.cpp文件，无，在src新建目录模块。

第二步：

　　复制make/makefile-src 到makefile。

## 2、添加应用生成可执行文件

在src新建目录

添加\*.cpp文件。

复制make/makefile-src 到src/新建目录/makefile。

Make/makefile-bin 到src/新建目录/makefile-bin。

注：makefile生成默认目标名以新建目录为主。可以修改其他指定目标名

默认DEST=$(notdir $(shell pwd))

改为DEST=指定目标名

修改方式参数其他现有模块方式

## 3、添加应用生成lib库

在src新建目录

添加\*.cpp文件。

复制make/makefile-src 到src/新建目录/makefile。

Make/makefile-lib 到src/新建目录/makefile-lib。

注：makefile生成默认目标名以新建目录为主。可以修改其他指定目标名

默认DEST=$(notdir $(shell pwd))

改为DEST=指定目标名

修改方式参数其他现有模块方式

# 三、本地环境配置

移交本地配置环境：

## 硬件：

　　　处理器：ＡＭＤ

　　　内存　：８Ｇ

## win：

　　　系统　：win 7 64位

　　　虚拟机：VMware Workstation 12 Player

　　　ＩＤＥ：Source Insight 3

　　　数据库：oralce mysql

　　　数据库工具：SQLyog - 32 bit　和Navicat Premium

## Linux：

　　　系统：VMware Cenos 64位

　　　数据库：mysql

　　　Smb　：service smb start

# 编译运行配置示例

## 添加日志配置环境变量

## 1、进入工程目录 cd /home/workspace/ppserver-mysql/

## 2、执行编译脚本 ./build.sh makefile mysql

生成错误日志在 ../error.txt文件中（可以在build.sh中修改编译时错误日志生成路径）

## 3、编译结束生成obj bin目录结构

.

├── bin

│   ├── PPServer #服务程序

│   └── tranStreeTest #压力测试客户端

├── obj

│   ├── Config.o

│   ├── DBOperation.o

│   ├── Dictate.o

│   ├── DlApi

│   │   ├── dlapi.o

│   │   └── makefile

│   ├── lib #在obj目录生成的动态库文件，可以复制到上级lib目录

│   │   ├── libDlApi.so

│   │   ├── libSample.so

│   ├── makefile

│   ├── multiThreadVariable.o

│   ├── PPServer

│   │   ├── MainFrm.o

│   │   ├── makefile

│   │   ├── Server.o

│   │   └── SMSOperation.o

│   ├── PublicModule.o

│   ├── RModule.o

│   ├── Sample

│   │   ├── makefile

│   │   └── Sample.o

│   ├── tcpSocket.o

│   ├── Text.o

│   └── UnionPayCommunicate.o

## 4、启动服务　./build.sh restart

配置文件中不能有下划线‘\_’，如果在数据库用户名中有可以用‘-’替换。

Eg:　　　# DBUser: tposdb\_user

改为　# DBUser: tposdb-user

其他配置需要使用下划线时，在CCofnig.cpp中有函数void switch\_to\_s(char \*src)。实现配置中‘-’到‘\_’的替换。

## 5、查看服务　./cron.sh checkproc

root 110702 9.6 0.1 159604 2776 pts/2 Sl 08:05 0:01 ./PPServer ../etc/conf.yaml

root 110709 0.0 0.0 103256 832 pts/2 S+ 08:05 0:00 grep PPServer

[root@localhost ppserver-mysql]#