**Core Engine自动化测试程序使用指南**

**[该程序会测试什么]**

该程序会通过执行以下步骤，模拟用户的操作，以便测试Core Engine是否能正确处理用户请求：

1. 通过Thrift接口向Arbiter添加设备，并判断设备是否正确添加到数据库和Device Server
2. 视频直播是否正常
3. 设置视频存储时每段视频的时间长度
4. 视频存储是否正常
5. 图片存储是否正常
6. 事件视频存储是否正常
7. 更新设备是否正确
8. 删除设备，查看数据库和DS是否删除正确

注意：

1. 要添加的设备的详细信息在配置文件scripts/configuration.cfg中
2. 配置文件configuration.cfg的配置介绍请看文档

《Automated-Testing-Config-File-Description》

1. 测试程序的详细测试策略请看文档《Automated-Testing-Detailed-Description》

**[执行测试前的必备条件]**

1. 测试程序和Core Engine安装在同一台服务器（如果不在同一台服务器，就需要Core Engine服务器对测试程序所在的服务器开放MySQL的远程访问权限）
2. 测试程序所在的服务器安装了Python环境
3. 测试程序所在的服务器安装了python MySQLdb模块

检查方法：

运行python，在终端输入import MySQLdb ，如果不报错，说明安装了该模块；否则，退出Python终端，执行sudo apt-get install python-mysqldb

1. Core Engine必须被正确安装了，并且Arbiter、Recording Server、MySQL和AMTK类型的Device Server正在运行

注意：

1. Python 采用2.7.3版本

**[如何运行测试程序]**

1. 将测试程序的文件夹目录拷贝到测试服务器的任意目录下
2. 修改配置文件scripts/configuration.cfg
3. 将要测试版本的Core Engine的juzz4v2.sql拷贝到sql/目录下（测试程序在每次运行时都会将最新的sql文件导入数据库）
4. 执行脚本scripts/run-all.sh进行测试
5. 一次测试流程如果不发生异常情况，所需时间大概是30~40分钟，请耐心等待

**[如何查看测试结果]**

1. 运行结束后，会生成两个日志文件，分别是scripts/log目录下的core-test-debug.log和core-test-info.log
2. 先查看core-test-info.log文件，其内有对各个测试点的结果的说明，用OK/False显示.
3. 在文档的最后部分，如果有” Congratulations!!! Your testing is successful.”则测试通过，表明测试结果和预期结果无误差或者误差在可接受范围之内
4. 如果没有上面那句话，则需查找info日志，定位出现False信息的位置。如果False信息仅仅出现在测试视频直播的时候判断实时帧率的地方，可认为这种情况依旧为True，判定为测试通过
5. 如果其他位置出现False信息，可查看core-test-debug.log日志文件，获得出现错误的日志信息，并定位出现错误的原因