DIDE自动测试方案

编写：陈家明

# 自动编译

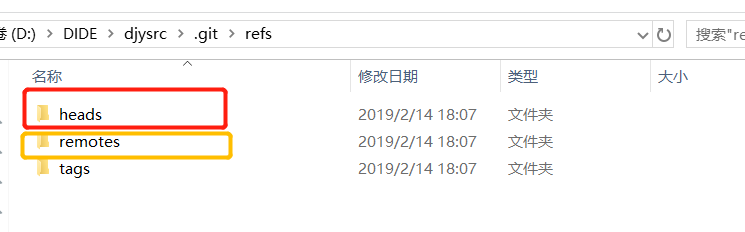


自动编译总流程图

## 1.1、定时检测源码

定时每小时检测一次Djysrc源码，如果检测到源码有更新时，自动更新本地源码。

如何判断源码需要更新：



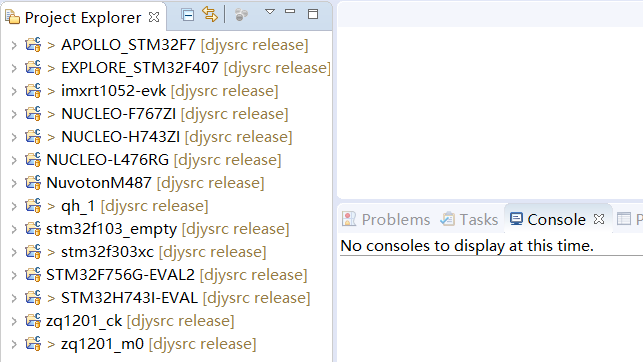
##### 如上图所示，.git/ref/heads下面存储了本地代码的版本号，.git/ref/heads存储了远程git的版本，但不是最新版本的，需要通过gitLocal.fetch().call()可将测目录下存储的版本更新为远程最新的版本，最后通过比较这连个本木下的版本号相同与否就可只带代码需不需需要更新。

## 1.2、自动更新本地源码

在需要更新源码的情况下，即以上的两个版本号不相同的情况下，通过gitLocal.pull().call()即可更新本地源码。

## 1.3、自动编译工作空间的工程

更新完源码之后，IDE获取当前工作空间中的所有工程：



每个工程都遍历所有的编译target，如果是库，则加入编译队列，buildJob = new BuildTarget()；buildJob.schedule(); BuildTarget是我自己写的一个编译target的类，如此即可让需要的库有顺序地进行编译。当每次编译库产生.a后，IDE自动检测，并触发库对应的工程进行编译。

## 1.4、保存编译期间产生的错误工程名称和target

当编译库或者编译工程产生错误时，将错误的工程和对应的target信息存储在IDE/readme/errer\_build.log文件中，写入的方式为追加，全部编译结束并且信息已经发送出去后，删除此文件。

## 1.5、将编译期间产生的错误工程名称和target发送给最后一次提交的用户邮箱

当所有工程的库和工程都编译结束后，系统读取IDE/readme/errer\_build.log文件中的内容，获取最后一次提交代码的用户邮箱，将错误信息发送给用户。

# 自动创建工程

IDE在后台自动模仿用户手动创建工程的过程，包括：自动填写工程名、选择板件、cpu、内核，选择组件，修改组件配置参数的内容，最后将创建好的工程生成在当前工作空间中，并进行编译和分析。