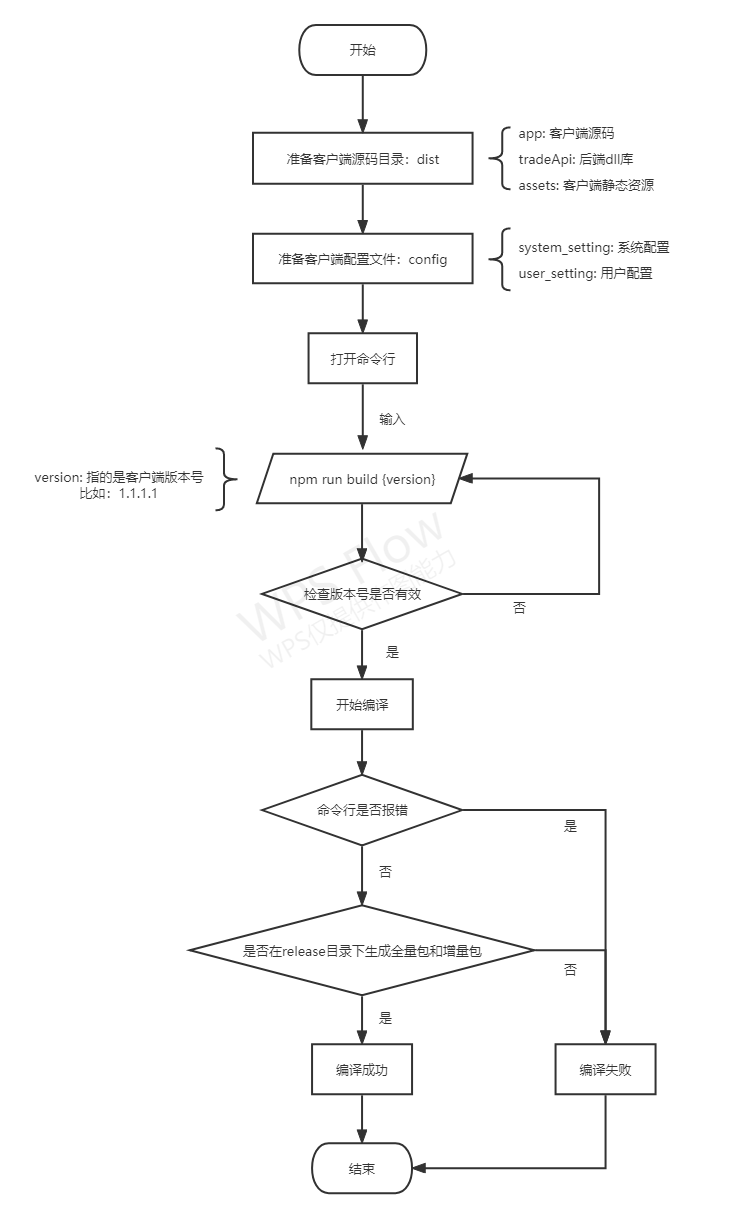
1. 客户端打包

****

打包完成后，当前打包目录会自动生成release目录,。

v1.1.1.1版本为例, 目录结构如下：

release

├── v1.1.1.1/ - 客户端版本目录

├── win32/ - windows 平台

├── AIT\_v1.1.1.1.exe 客户端安装程序

├── AIT\_upgrade\_v1.1.1.1.tgz 客户端增量压缩包

├── macOS/ - macOS 平台（规划）

├── AIT\_v1.1.1.1.dmg 客户端安装程序

├── AIT\_upgrade\_v1.1.1.1.tgz 客户端增量压缩包  
├── upgrade.json - 更新信息

生成的Upgrade.json 是客户端升级版信息，具体内容如下：

{

  "version": "1.1.1.1", // 升级包版本

  "releaseDate": "2022-03-27 00:00:00", // 发布时间

  "file": "AIT\_upgrade\_v1.1.1.1.tgz", // 升级包名

  "fileServer": "www.xxx.com", // 文件服务url

  "description": [ // 更新描述

    "新增xx功能",

    "优化xx性能",

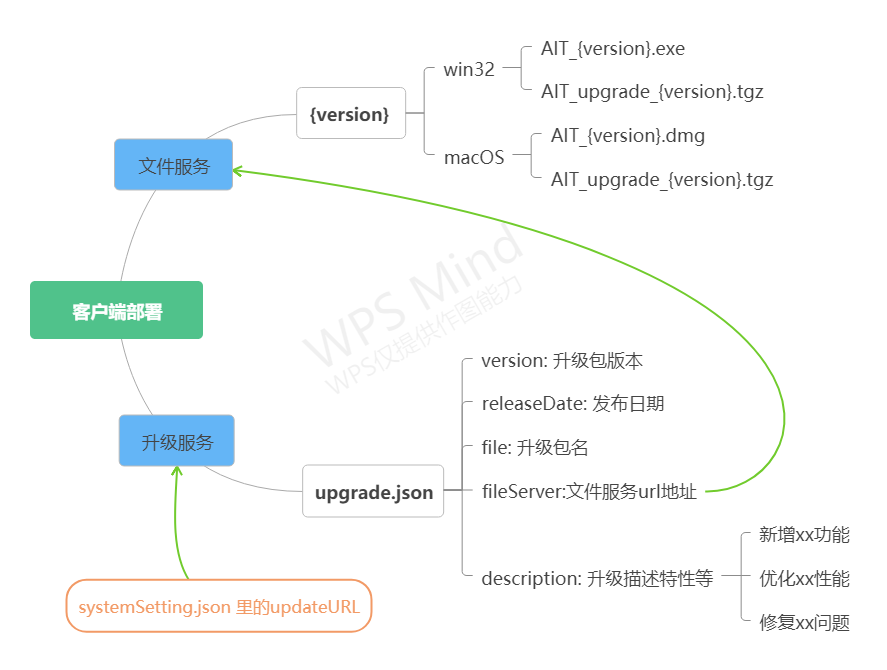
    "修复xx问题"

  ]，

“force”: false

}

1. 客户端部署



1.文件服务url访问规则：{fileServer}/{version}/{platform}/{file}

假设：我们文件服务地址是: http://[www.aaa.com,](http://www.aaa.com,) 通过以下url地址去访问，

windows平台：http://[www.aaa.com/v1.1.1.1/win32/AIT\_v1.1.1.1.exe](http://www.aaa.com/v1.1.1.1/win32/AIT_v1.1.1.1.exe)

macOS平台：<http://www.aaa.com/v1.1.1.1/macOS/AIT_v1.1.1.1.dmg>

2.升级服务url访问规则：{UpgradeServer}/upgrade.json

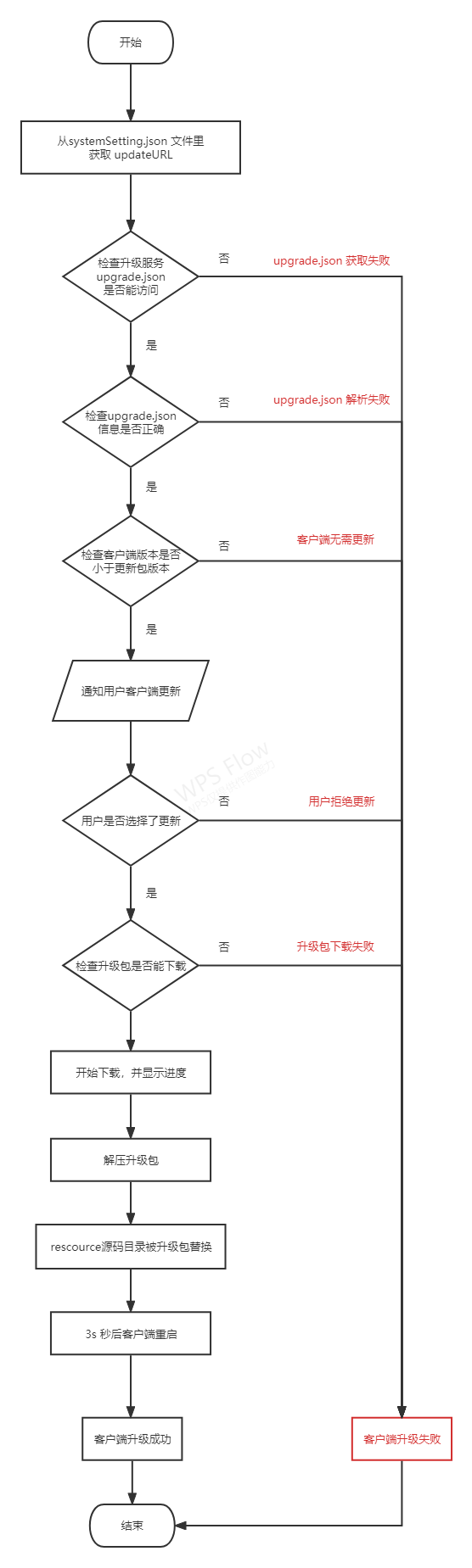
假设：我们升级服务地址是: http://[www.bbb.com,](http://www.aaa.com,) 通过以下url地址去访问，

windows平台：http://[www.bbb.com/](http://www.aaa.com/v1.1.1.1/win32/AIT_v1.1.1.1.exe)upgrade.json

macOS平台：[http://www.bbb.com/upgrade.json](http://www.aaa.com/v1.1.1.1/macOS/AIT_v1.1.1.1.dmg)

注意：无论是windows平台还是macOS平台upgrade.json 的请求地址和更新信息是一样的。因为我们在打包阶段已经做出平台包的分离。客户端升级时会自动判断当前运行平台 和 更新信息，再去下载对应的升级包。

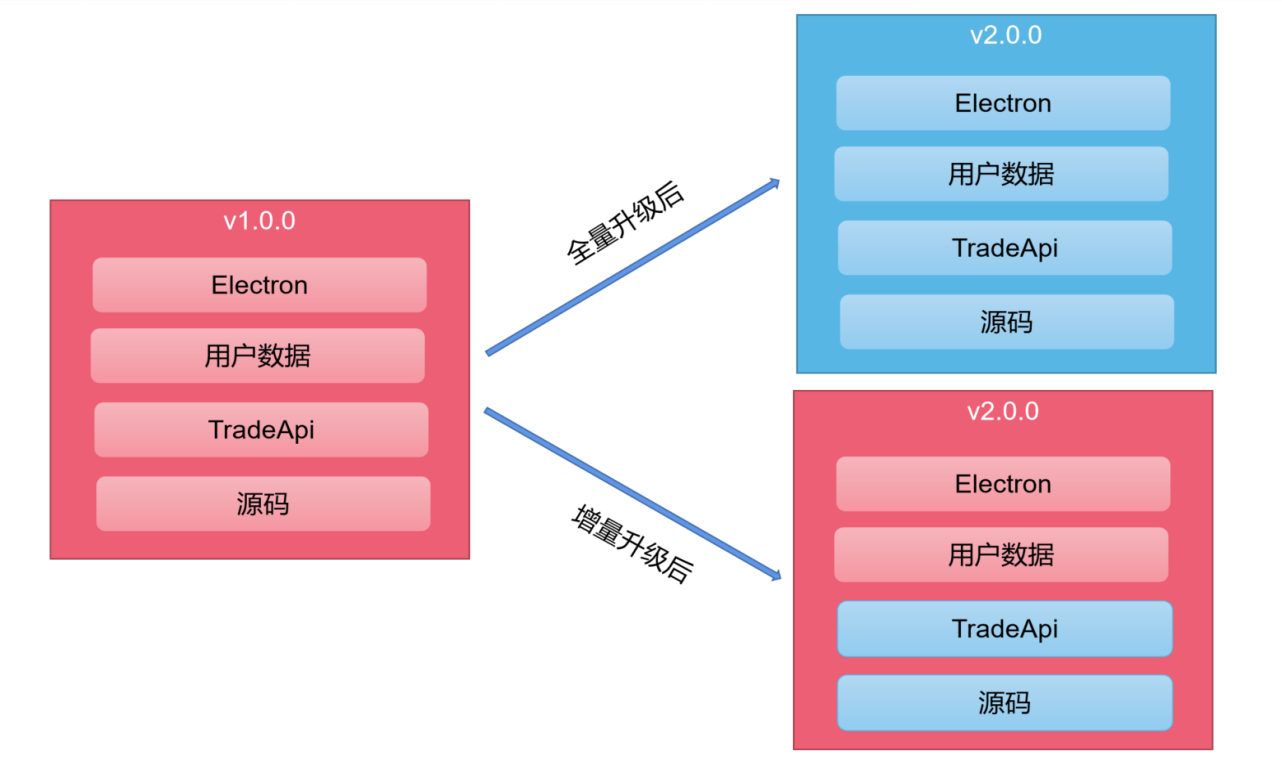
1. 客户端升级



全量包 VS 增量包

|  |  |  |
| --- | --- | --- |
|  | 客户端全量包 | 客户端增量包 |
| 文件大小？ | ≈64MB | ≈16MB |
| 文件后缀？ | exe 或者 dmg | tgz |
| 有哪些模块组成？ | Electron + 源码 + TradeApi | 源码+TradeApi |
| 用户配置是否被修改？ | 重新安装时会被覆盖 | 否 |
| 客户端数据是否被覆盖？ | 重新安装是会被覆盖 | 否 |
| 是否支持手动升级？ | 运行exe安装程序即可 | 下载增量包并解压，打开客户端安装目录，把resources目录替换为增量包即可 |
| 是否支持下载？ | 是 | 是 |
| 是否修改注册表？ | 是 | 否 |

客户端升级先后对比：



Q: 增量升级后 客户端配置文件怎么做更新？

假设：v1.0.0 配置文件原有配置有三个字段： a, b, c, 如果版本迭代，会有如下更新：

v1.1.0 新增d, 删除a

v1.2.0 新增e, 删除b

v1.3.0 新增f, 删除c

v1.4.0 修改d, 修改e, 修改f

那么当用户从v1.0.0 跨版本升级到 v1.4.0 时候 ，怎么确保 客户端配置文件是最新的？

方案1：通过代码控制（随之版本的迭代，需要获取每一个版本的更改记录，维护成本越来越高，代码量不可控）

方案2：通过增量配置（跟方案1类似）

方案3：通过界面UI 方式让用户自己去改？配置文件默认不放开？（跟方案1类似）

方案4：增量更新时，客户端出厂配置覆盖掉用户现有配置，客户端原有配置会修改（那么做增量更新的意义合在？）

方案5：v1.x 系列版本不解决，等有了服务端再解决？

Q：客户端强制更新？