**NETCA CRYPTO macOS版使用说明**

## 简介

在 KeyX 开发完成发布前，需要把代码打包成 .app 。对于用户来说，.dmg方式直接的拖拉复制是最简单的安装方式，且不会对用户系统有过多的侵入，相比较 .pkg 来说，更容易被接受。虽然说 dmg 包会比 pkg 包的体积更大些，但考虑到 KeyX 使用的依赖包体积相对较小，最后打包完成的产物体积应该也不会很大。综上考虑，使用 dmg 来打包 KeyX 。

下面简要说明 Crypto 各库的处理方式和和相关配置文件的存放路径。

## 查看库依赖路径是否正确

首先，所有库存放在 /Applications/KeyX.app/Contents/Frameworks/ 或其他自定义目录下。

在 macOS 下是使用 dylib 动态库的形式来链接的，在运行过程中若需要链接其他库，是通过编译时指定的其他库的 install\_name 来寻找被依赖库的。其中，UTIL库需要链接gmp.dylib ，URL库需要链接ssl.dyllib 和 crypto.dylib(区别于netca\_crypto)，CRYPTO需要链接 sqlite3.dylib（有些系统自带）。

运行时寻找被依赖库的路径是在编译时指定，因此在运行过程中必定会出现找不到库的问题，这时候就需要手动指定 Crypto 库在运行时去哪里找到所需的动态库。

1. 可以使用命令 otool -L 查看 dylib 依赖的库。
2. 通过命令 install\_name\_tool [-change old new] dylib来修改 dylib 依赖的库的寻找路径。
3. 通过 ln -s 命令创建软链接或将crypto库改名，去掉后缀的版本号。

由此来确保所有库都可以在运行时被找到。

## 相关配置文件

配置文件存放在 /Applications/KeyX.app/Contents/NETCA/ 下。

CRYPTO配置分别有设备库和全局配置文件，详见《NETCA PKI API 参考手册》4.4。

关于UI配置，还有一个配置文件：NetcaUIExtensionEncodeConfigure，是告诉UI如何去解析未知扩展，它有下表列出的属性：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| EncodeMethod | 字符串 | 字符编码类型：UTF8String、PrintableString、IA5String、VisibleString、BMPString、OID、NULL、OctetString、Integer、Boolean |
| isNetca | 数字 | 1代表是Netca的扩展，0代表否 |

该文件为UTF-8格式的，可以包含BOM也可以不包含。文件格式如下：每个配置项以 [扩展的OID] 这样的行开头，然后，每个配置项一行，直到文件结束或者遇到另外一个配置项为止。每个配置项的格式为“名称”=“值”，不包含引号。名称和上面的表格的一致，数字类型的值转换为10进制整数表示。