|  |  |  |
| --- | --- | --- |
| 华为技术有限公司 | 产品版本 | 密级 |
|  | 外部公开 |
| 产品名称 | 页数：共14页 |

【华为SSL VPN集成指导文档-iOS】

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **拟制:** |  |  | **日期：** | **2019-9-21** |
| **审核:** |  |  | **日期：** | **年-月-日** |
| **审核:** |  |  | **日期：** | **年-月-日** |
| **批准:** |  |  | **日期：** | **年-月-日** |



修订记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **修订版本** | **描述** | **作者** |
| 2019-06-18 | 1.0 | 初稿 |  |
| 2019-12-17 | 2.0 | 第一版 |  |
|  |  |  |  |

目录

[第一章 SDK介绍 4](#_Toc27506553)

[第二章 开发前准备 4](#_Toc27506554)

[1.1 创建工程的Provisioning Profile 4](#_Toc27506555)

[1.2 配置工程开发环境 5](#_Toc27506556)

[第三章SDK使用说明 10](#_Toc27506557)

[3.1 SecoSDKTunnel.framework的使用 10](#_Toc27506558)

[3.2 SecoSDK.framework的使用 10](#_Toc27506559)

[3.3 SecoSDK.framework中的接口使用 12](#_Toc27506560)

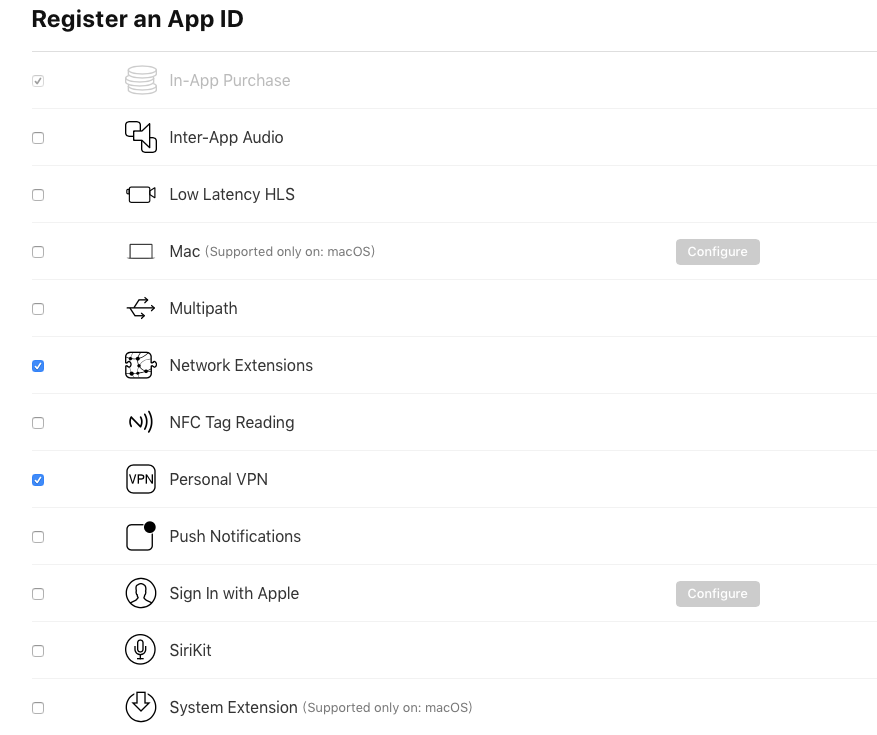
# 第一章 SDK介绍

用户可通过集成SDK实现SSL VPN业务，SDK分为iOS和安卓两种平台，此文档仅限于iOS平台使用。此SDK包含两部分，SecoSDK.framework用于主程序调用的API，SecoSDKTunnel.framework用于插件集成，详细使用见以下操作说明。

# 第二章 开发前准备

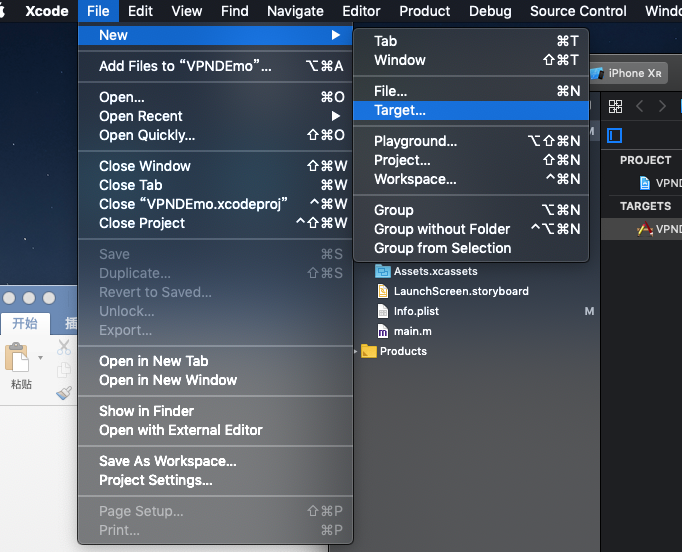
## 创建工程的Provisioning Profile

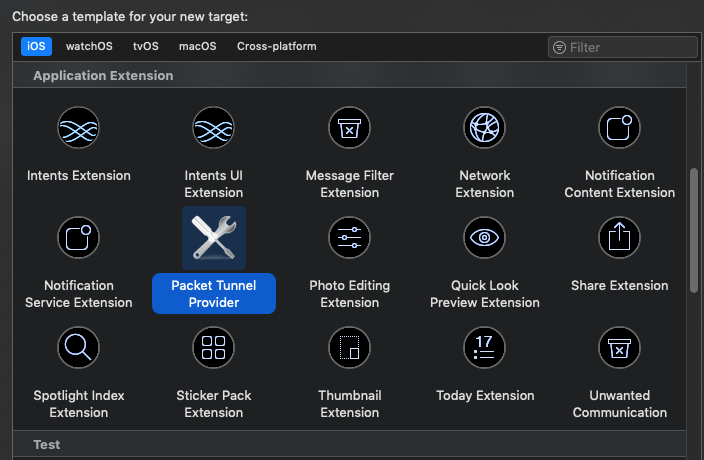
在苹果开发者账号网站创建扩展的Provisioning Profile，在创建appid时需开启Network Extension、Personal VPN，如下图示。注意，app及扩展都需要开启Network Extension、Personal VPN，且扩展的appid以APP的appid为前缀。比如APP的appid为com.testclient.client，那么扩展的appid可以设置为com.testclient.client.vpnextension。



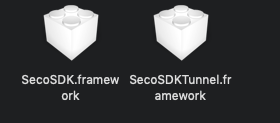
## 配置工程开发环境

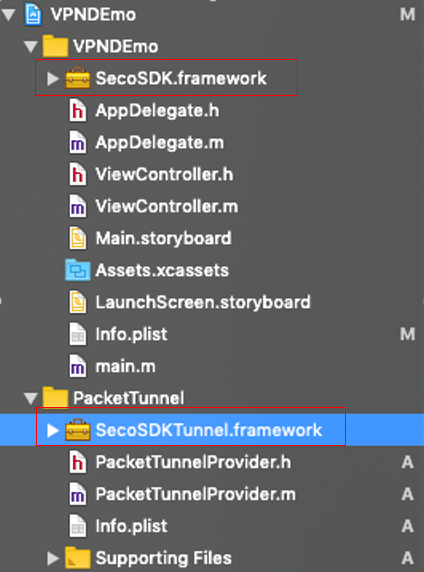
（1）创建工程后新建一个Target，选择File->New->Target，选择Packet Tunnel Provider



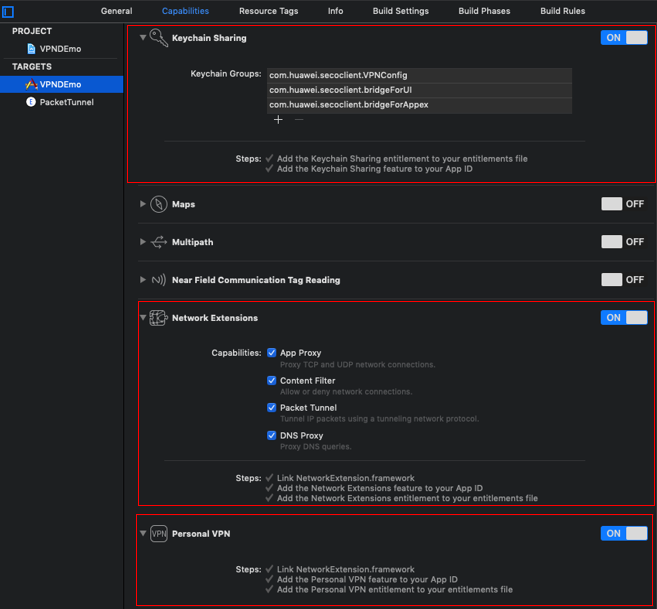


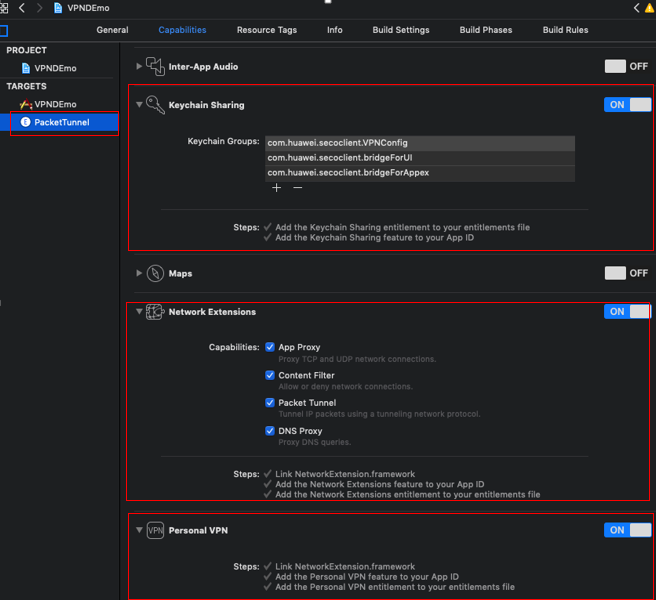
（2）将提供的静态库SecoSDK.framework、SecoSDKTunnel.framework分别引入到工程和新建的Packet Tunnel Provider中





（3）在两个Target（VPNDemo和PacketTunnel）中配置申请好的appid，在xcode的Capabilities中开启相应权限





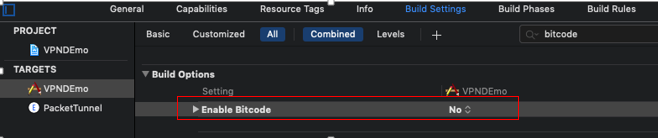
**此处注意开启Keychain Sharing时需要添加以下三个Keychain Groups**：

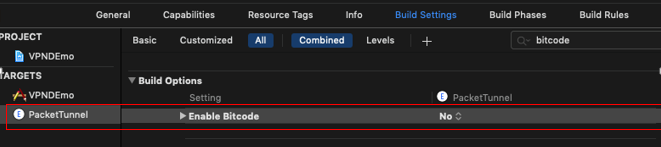
com.huawei.secoclient.VPNConfig

com.huawei.secoclient.bridgeForUI

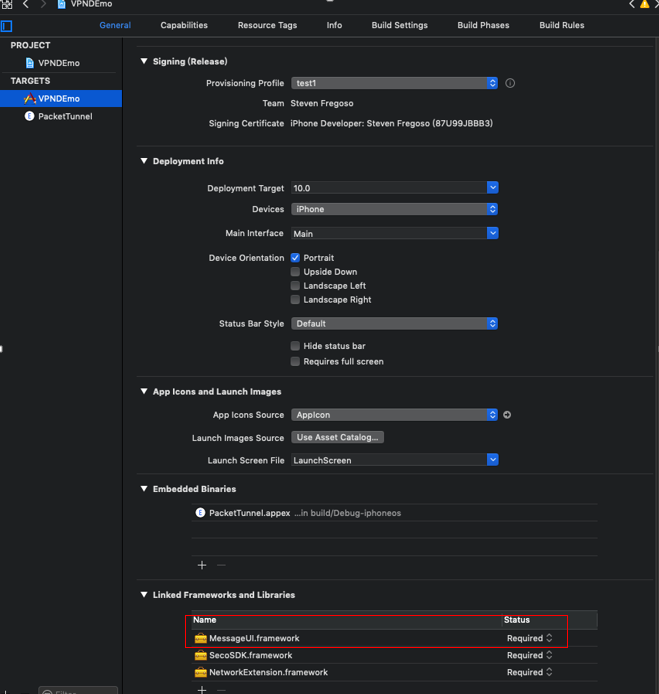
com.huawei.secoclient.bridgeForAppex

（4）在Build Settings里面设置关闭 Bitcode





（5）引入MessageUI.framework依赖库



# 第三章SDK使用说明

## 3.1 SecoSDKTunnel.framework的使用

（1） PacketTunnelProvider.h文件中引入 “<SecoSDKTunnel/SecoSDKTunnel.h>”，并且使PacketTunnelProvider.h继承于TunnelManager

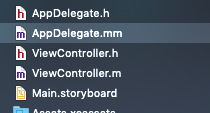
（2）在PacketTunnelProvider.m 中的三个代理中分别自调用各自代理方法

通过以上两个步骤，Packet Tunnel Provider已经配置完成，不需要再做其他操作。

## 3.2 SecoSDK.framework的使用

（1）在AppDelegate.m中引入 “import <SecoSDK/SSLVPN.h>”，在 “didFinishLaunching”中调用

“SSLVPNInitWithPacketTunnelBundleIdentify”方法进行初始化，**注意：由于静态库含有c++的代码，此时需要将“AppDelegate.m”改为 “AppDelegate.mm”。**



（2）在使用连接VPN接口的控制器中需要遵守代理“VPNStatusDelegate”，并在相应的代理方法中获取VPN状态：

状态一：VPN未连接

**- (void)whenTheVPNStatusDisconnected;**

状态二：VPN正在连接，并实时返回连接状态

|  |  |
| --- | --- |
| **-(void)whenTheVPNStatusConnectingWithConnectErrorCode:**  **(VPN\_CONNECTING\_ERROR)connectErrorCode;** | |
| **错误码** | **说明** |
| VPN\_TUNNEL\_CONNECTING | 正在连接 |
| VPN\_WAITE\_FOR\_RESULT | 等待结果 |
| VPN\_TUNNEL\_FIRSTCONN\_FAIL | 无法建立VPN连接，VPN服务器可能无法到达 |
| VPN\_TUNNEL\_CONNECT\_LOGIN\_OHERTYPE | 其他认证方式，暂不支持 |
| VPN\_TUNNEL\_CONNECT\_SUCCEED | vpn隧道建立成功 |
| LOGIN\_CNEM\_START\_SUCCEED | 登陆成功，网络扩展启动成功 |
| LOGIN\_CNEM\_START\_FAIL | 登陆成功，但网络扩展启动失败 |
| LOGIN\_AUTH\_FAIL | 认证失败 |
| LOGIN\_USEROVERTOP\_ERROR | 用户连接数已达到上限 |
| LOGIN\_USER\_FIRST\_LOGIN\_ERROR | 您的密码为初始状态，请修改密码后再开始业务 |
| LOGIN\_USEREXPIREWEAK\_ERROR | 您的密码已经过期，请和管理员联系 |
| VPN\_CONNECTING\_SUCCESS | 连接成功 |

状态三：VPN已连接，并且随时接收网关返回的信息

|  |  |
| --- | --- |
| **- (void)whenTheVPNStatusConnectedWithConnectedErrorCode:**  **(VPN\_CONNECTED\_RESULT\_CODE)connectedErrorCode;** | |
| 错误码 | 说明 |
| VPN\_RECONNECT\_FAIL | 重连失败 |
| VPN\_KICKOUT\_SUCCESS | 被网关踢下线 |
| VPN\_DHCP\_ERROR | DHCP错误 |

**注意：VPN状态获取需要刷新UI时，请在主线程进行操作**

## 3.3 SecoSDK.framework中的接口使用

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | - (void)connectVPNWithConnectName:(NSString \*)connectName serverAddress:(NSString \*)serverAddress serverPort:(NSString \*)serverPort isGMSSL:(BOOL)isGMSSL transferMode:(TUNNEL\_MODE)transferMode username:(NSString \*)username password:(NSString \*)password connectResult:(nullable void (^)(VPN\_TUNNEL\_CONNECT\_ERROR))connectResult; | | |
| 功能 | 创建SSLVPN连接 | | |
|  | | | |
| 参数 | 参数名 | 类型 | 说明 |
| connectName | NSString | 连接名称 |
| serverAddress | NSString | 连接网关地址 |
| isGMSSL | BOOL | 是否使用国密加密算法 |
| transferMode | int | 传输模式 |
| username | NSString | 用户名 |
| password | NSString | 密码 |
| connectResult | int | 连接结果回调 |
| 结果回调说明 | SAVE\_CONNECT\_SUCCEED | | 配置保存成功 |
| SAVE\_CONNECT\_FAILED | | 配置保存失败 |
| SAVE\_CONNECT\_FAILED\_REPEAT | | 配置不全或格式有误 |
| SAVE\_CONNECT\_FAILED\_USER\_REFUSE | | 首次保存配置到系统vpn配置时，需要用户同意并输入终端锁屏密码 |
| 备注 | | **注意：VPN创建连接结果需要刷新UI时，请在主线程进行操作** | | | |

|  |  |
| --- | --- |
| 名称 | + (SSLVPN \*)shareInstance; |
| 功能 | 获取管理类单例对象 |
| 备注 | 管理类单例对象，调用接口，设置代理均要使用 |

|  |  |
| --- | --- |
| 名称 | - (void)disconnectVPN; |
| 功能 | 关闭VPN连接 |

|  |  |
| --- | --- |
| 名称 | - (void)collectLog; |
| 功能 | 收集日志 |
| 添加日志 | 需要在代码中添加日志需要引入 “import <SecoSDK/WriteLog.h>”,直接使用 “LOGWSERR(format,...)、LOGWSWARN(format,...)、LOGWSNOTICE(format,...)、LOGWSDEBUG(format,...)”进行添加 |
| 备注 | 日志文件是以邮件发出的，使用时需要确保手机邮件可以正常发送邮件 |

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | - (void)modifyPasswordWithOldPassword:(NSString \*)oldPassword AndNewPassword:(NSString \*)newPassword CompletionHandler:(nullable void (^)(VPN\_MODIFYPASSWORD\_RESULT\_CODE errorCode))completionHandler; | | |
| 功能 | 修改登录密码 | | |
| 参数 | 参数名 | 类型 | 说明 |
| oldPassword | NSString | 旧密码 |
| newPassword | NSString | 新密码 |
| completionHandler | int | 修改结果回调 |
| 结果回调说明 | MODIFYPWD\_SUCCESS | | 修改成功 |
| MODIFYPWD\_FORMAT\_ERROR | | 密码格式错误 |
| MODIFYPWD\_FAIL\_BECAUSE\_SAVING | | 网关侧正在保存,修改密码失败 |
| 备注 | 只能是在连接成功状态下才可以进行密码修改 | | |