

ZOSA

中关村智能终端操作系统产业 联盟标准

T/ZOSA 001-2017

智能终端应用生态 高性能应用编程接口（API）规范

XXXX-XX-XX 发布

XXXX-XX-XX 实施

中关村智能终端操作系统产业联盟发布

目 次

前 言.....	II
1 范围.....	1
2 规范性引用文件.....	1
3 术语、定义和缩略语.....	1
3.1 术语和定义.....	1
3.1.1 智能终端.....	1
3.1.2 移动智能终端操作系统 smart mobile terminal operating system.....	1
3.1.3 应用软件 application software.....	1
3.1.4 软件开发工具包 Software Development Kit.....	1
3.1.5 用户.....	1
3.2 缩略语.....	1
4 智能终端应用生态.....	1
4.1 智能终端应用生态.....	2
4.2 应用编程界面（API）规范.....	2
5 API 分类.....	2
5.1 安全管理.....	2
5.2 设备和系统信息管理.....	2
5.3 应用框架管理.....	2
5.4 通短联管理.....	3
5.5 网络管理.....	3
5.6 多媒体.....	3
5.7 消息交互管理.....	3
5.8 MDM 管理.....	3
5.9 其他.....	4
6 参考文献.....	4
附录 A.....	5
A.1 安全管理.....	5
A.2 设备管理.....	13
A.3 应用框架管理.....	16
A.4 通短联管理.....	22
A.5 网络管理.....	23
A.6 多媒体.....	28
A.7 消息交互管理.....	30
A.8 MDM 管理.....	35
A.9 其他.....	39

前 言

本标准按照 GB/T 1.1-2009 给出的规则起草。

本标准由中关村智能终端操作系统产业联盟提出并归口。

本标准起草单位：北京元心科技有限公司，工业和信息化部软件与集成电路促进中心，大唐联诚信息系统技术有限公司，北京航空航天大学，武汉大学，南开大学，北京创原天地科技有限公司，北京虎符信息技术有限公司，启迪国信科技有限公司，苏州国芯科技有限公司，联芯科技有限公司。

本标准起草人：姜哲，史子旺，刘龙庚，刘帅，马士民，陈晖，王勋，梁博，刘建伟，关振宇，杜瑞颖，李涛，张金，宫晓利，尹桦，张宝林，方恒禄，王新军，孙健，李江，高传贵，王建军，张志敏，祝晓平。

智能终端应用生态高性能应用编程接口（API）规范

1 范围

本标准规定了智能终端应用生态中应用程序设计研发时所必须的接口和运行环境，确定了该类自主移动应用生态中设计应用编程接口的一般原则。

本标准适用于移动智能操作系统实现者、工具链实现者和应用程序开发者。

2 规范性引用文件

下列文件中的条款通过本标准的引用而成为本标准的条款。凡是注日期的引用文件，其随后所有的修改单（不包括勘误的内容）或修订版均不适用于本标准，然而鼓励根据本标准达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件，其最新版本适用于本标准。

ISO/IEC 9899:1999 信息技术 程序设计语言 C

ISO/IEC 14882:2011 信息技术 程序设计语言 C++

3 术语、定义和缩略语

3.1 术语和定义

3.1.1 智能终端

具有计算机结构和功能并集成多种功能传感器，能通过多种无线网络为用户提供应用服务的手持式终端设备，在本标准中简称为“智能终端”。

3.1.2 移动智能终端操作系统 smart mobile terminal operating system

运行在移动智能终端上的系统软件，是移动智能终端的组成部分，控制、管理移动终端上的硬件和软件，提供用户操作界面和应用软件编程接口。

3.1.3 应用软件 application software

智能终端操作系统之外，向用户提供服务功能的软件。

3.1.4 软件开发工具包 Software Development Kit

辅助开发应用软件的相关文档、范例和工具的集合。

3.1.5 用户

在智能终端操作系统之外，与智能终端操作系统交互的任何实体（人员用户或外部 IT 实体）。

3.2 缩略语

下列缩略语适用于本文件。

API 应用编程界面（Application Programming Interface）

SDK 软件开发工具包（Software Development Kit）

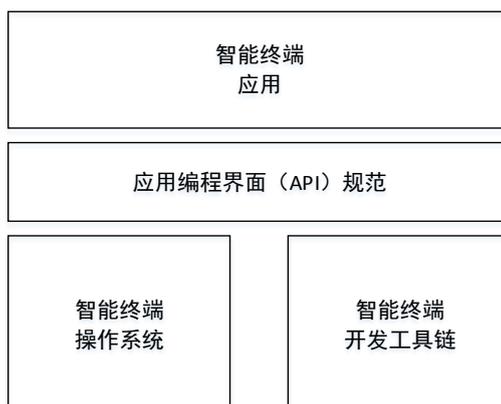
4 智能终端应用生态

4.1 智能终端应用生态

智能终端应用生态由三部分构成：智能终端应用、智能终端操作系统、智能终端应用开发工具链。智能终端应用，由应用开发商提供，负责实现用户的使用逻辑；智能终端操作系统，由操作系统厂商提供，负责管理智能终端硬件资源，为智能终端应用提供运行环境；智能终端应用开发工具链，由开发工具链厂商提供，为智能终端应用提供研发与调试环境。

4.2 应用编程界面（API）规范

应用编程界面 API 规范，是智能终端应用生态的核心，需要智能终端应用、智能终端操作系统、智能终端开发工具链共同遵守。



为满足特殊行业用户对智能终端及智能终端应用对高性能的需求，智能终端应用生态选用编译型语言 Qt 作为官方语言。本规范是在 Qt5.3 所提供的 API 之外，定义了与智能终端相关的 API 规范。

5 API 分类

5.1 安全管理

CUserAuthenManager	提供系统锁屏状态接口，用于获得屏幕状态和屏幕状态变化通知。
CLockScreenManager	提供锁屏管理接口。
CMDMOVManager	提供 VPN 管理接口
CPkgInstallPermissionManager	提供于应用敏感权限管理接口
CPKIManager	提供密码基础运算的接口
CcryptoFile	提供加密文件接口
CCryptoStorageManager	提供加密平台配置接口

5.2 设备和系统信息管理

CArmManager	提供定时器管理接口
CBatteryManager	提供电池管理、查询接口
CFlashManager	提供闪光灯管理接口
CStorageManager	提供系统存储管理接口
CEnvironment	提供系统数据、文件存储路径查询接口
COSInfo	提供系统信息查询的接口，如版本号等

5.3 应用框架管理

CappInfo	提供应用信息查询接口
CHtmlAppInfo	提供 HTML 应用查询接口。
CPackageInfo	提供应用程序包信息查询接口。
CPackageManager	提供应用程序管理接口
CServiceInfo	提供服务信息查询接口
CuiAppInfo	提供应用程序信息查询接口
CProcessManager	提供系统进程管理接口
CProfileManager	提供系统铃声管理接口

5.4 通短联管理

CCallLogManager	提供通话记录管理接口
ContactData	提供联系人数据管理接口
CContactModel	提供联系人列表管理接口
CMessageManager	提供短信管理接口

5.5 网络管理

CNetworkManager	提供网络管理接口
WifiNetWorkTechnology	提供 WiFi 热点管理接口。
WifiNetworkService	提供 Wifi 服务管理接口。
QGeoPositionInfoSource	提供位置更新管理接口。
QGeoCoordinate	提供地理坐标管理接口。
CSystemLocationManager	提供系统定位管理接口。
QBluetoothLocalDevice	提供本地蓝牙设备查询接口。
QBluetoothDeviceDiscoveryAgent	提供查找蓝牙设备接口。
QBluetoothHostInfo	提供蓝牙设备本地管理接口。

5.6 多媒体

CNGFManager	提供了关于播放声音和振动的接口
-------------	-----------------

5.7 消息交互管理

CAlertDialogManager	提供对话框操作接口
CInfoDialogManager	提供信息对话框操作接口
CInputDialogManager	提供输入对话框操作接口
CAbstractNotification	提供系统通知的抽象类型
Cbadge	提供播放声音、振动的接口
CNotification	提供通知接口
CNotificationManager	提供通知服务管理接口
CProgressNotification	提供进度条通知服务接口
CPushManager	提供给第三方应用程序使用的推送接口
CToastManager	提供 Toast 提示框操作接口

5.8 MDM 管理

CMDMPermissionManager	提供用于 MDM 管理敏感权限的接口
CMDMCallLogManager	为 MDM 提供通话记录管理接口
CMDMContactManager	为 MDM 提供联系人操作的接口
CMDMMessageManager	为 MDM 提供短信管理接口
CMDMPushManager	提供 MDM 使用的推送服务接口

5.9 其他

CConfManager	提供三方应用数据中心查询、设置接口
CQRCode	提供生成二维码，解析二维码相关接口

6 参考文献

- [1] 《C Primer 中文版 C API 设计》
- [2] QT 5.3 API reference
- [3] 《C++编码规范》

附录 A

A.1 安全管理

A.1.1 CUserAuthenManager

类说明:

CUserAuthenManager 类提供了关于系统锁屏状态查询接口, 用于获得屏幕状态及屏幕状态变化通知函数, 来完成与屏幕状态相关的功能接口。

A.1.1.1 bool CUserAuthenManager::isLogined()

函数说明:

获取当前系统状态是否已登陆。

参数说明:

输入值: 无

返回值: 如果是已登陆, 返回 true, 否则返回 false。

A.1.1.2 ScreenState CUserAuthenManager::screenState()

函数说明:

获得当前系统锁屏状态。

参数说明:

输入值: 无

返回值: 返回 ScreenState 枚举类型。

ScreenLocked 屏幕已锁屏

ScreenUnLocked 屏幕已解锁

ScreenFrozen 屏幕已冻结, 因口令输入错误次数超限导致

A.1.2 CLockScreenManager

类说明:

CLockScreenManager 类是锁屏管理类, 提供了屏幕锁定和解除屏幕锁定的方法。

A.1.2.1 bool CLockScreenManager::lock()

函数说明:

完成锁定屏幕功能。

参数说明:

输入值: 无

返回值: 成功为 true, 失败为 false

A.1.2.2 bool CLockScreenManager::unlock()

函数说明:

解除屏幕锁定, 如果系统设置密码, 则进入解锁页面, 如果没有密码则直接进入手机首页。

参数说明:

输入值: 无

返回值: 成功为 true, 失败为 false

A.1.3 CPkgInstallPermissionManager

类说明

CPkgInstallPermissionManager 类提供了关于包管理的敏感权限的接口。

A.1.3.1 int CPkgInstallPermissionManager::installAppPermission(QString allPerms)

函数说明:

安装应用的敏感权限。

参数说明:

输入值: 参数 allPerms 格式为: {uid;sopid;perm1:0,perm2:1,perm3:1....}。

返回值: 成功返回 0, 否则返回非 0

注意事项:

uid 为应用程序账户 ID ; sopid 为应用程序 SOP 包标识; perm1、perm2、perm3 等为敏感权限名称; 0 表示没有申请权限, 1 表示有申请权限。

A.1.3.2 int CPkgInstallPermissionManager::removeAppPermission(QString sopid)

函数说明:

删除应用的敏感权限。

参数说明:

输入值: sopid 参数为应用程序 SOP 包标识。

返回值: 成功返回 0, 否则返回非 0。

A.1.4 CPKIManager

类说明

CPKIManager 类提供了密码基础运算的接口。

A.1.4.1 uint CPKIManager::closeHandle(PKIObjectHandle hObj)

函数说明:

关闭对象句柄。

参数说明:

输入值: hObj 参数为对象句柄, 可以关闭摘要、加解密和签验签名对象句柄。

返回值: 成功返回 0, 否则返回非 0 错误码。

A.1.4.2 uint CPKIManager::closeProvider(PKIProviderHandle hProvider)

函数说明:

关闭密码对象句柄。

参数说明:

输入值: hProvider 参数为密钥运算提供者对象句柄。

返回值: 成功返回 0, 否则返回非 0 错误码。

A.1.4.3 uint CPKIManager::createP10(PKIProviderHandle hProvider, QString containerName, QString certDN, int keyBitLen, QString keyPin, uint digestAlg, uint keyUsage, QByteArray & p10)

函数说明:

枚举的密钥运算提供者。

参数说明:**输入值:**

枚举的密钥运算提供者。

hProvider 参数为密钥运算提供者对象句柄。

containerName 参数为容器名称。

certDN 参数为证书 DN，格式

C=CN, ST=beijing, L=dongchengqu, O=yunxin, OU=yanfazhongxin, CN=mmm, E=mmm. email. com,
“=” 和 “,” 为英文符号且两端不能有空格。

keyBitLen 参数为密钥长度，目前支持 1024，2048。

keyPin 参数为密钥的 PIN 码。

digestAlg 参数为摘要算法。

keyUsage 参数为密钥用途，1 签名，2 加密。

p10 参数为证书请求结果。

返回值：成功返回 0，否则返回非 0 错误码

A.1.4.4 uint CPKIManager::decrypt(PKIObjectHandle hObj, const QByteArray & in, QByteArray & out)

函数说明:

单组数据解密。

参数说明:

输入值：hObj 参数为解密对象句柄。

in 参数为待解密数据。

out 参数为解密后数据。

返回值：成功返回 0，否则返回非 0 错误码。

A.1.4.5 uint CPKIManager::decryptFinal(PKIObjectHandle hObj, QByteArray & out)

函数说明:

结束解密。

参数说明:**输入值:**

hObj 参数为解密对象句柄。

out 参数为解密后数据。

返回值：成功返回 0，否则返回非 0 错误码。

A.1.4.6 uint CPKIManager::decryptInit(PKIProviderHandle hProvider, uint uAlgID, const QByteArray & key, const QByteArray & iv, PKIObjectHandle * phObj)

函数说明:

解密初始化。

参数说明:**输入值:**

hProvider 参数为密钥运算提供者对象句柄。

uAlgID 参数为对称算法标识。

key 参数为对称密钥。

iv 参数为初始向量。

phObj 参数为解密对象句柄。

返回值：成功返回 0，否则返回非 0 错误码。

注意事项：

phObj 使用结束后，使用 closeHandle 释放内存。

A.1.4.7 uint CPKIManager::decryptUpdate(PKIObjectHandle hObj, const QByteArray & in, QByteArray & out)

函数说明：

多组数据解密。

参数说明：

输入值：

hObj 参数为解密对象句柄。

in 参数为待解密数据。

out 参数为解密后数据。

返回值：成功返回 0，否则返回非 0 错误码。

A.1.4.8 uint CPKIManager::digest(PKIObjectHandle hObj, const QByteArray & in, QByteArray & out)

函数说明：

单组数据摘要。

参数说明：

输入值：

hObj 参数为摘要对象句柄。

in 参数为数据。

out 参数为摘要结果。

返回值：成功返回 0，否则返回非 0 错误码。

A.1.4.9 uint CPKIManager::digestFinal(PKIObjectHandle hObj, QByteArray & out)

函数说明：

结束摘要。

参数说明：

输入值：

hObj 参数为摘要对象句柄。

out 参数为摘要结果。

返回值：成功返回 0，否则返回非 0 错误码。

A.1.4.10 uint CPKIManager::digestInit(PKIProviderHandle hProvider, uint uAlgID, PKIObjectHandle * phObj)

函数说明：

摘要初始化。

参数说明：

输入值：

hProvider 参数为密钥运算提供者对象句柄。

uAlgID 参数为算法标识。

phObj参数为摘要对象句柄。

返回值：成功返回 0，否则返回非0错误码。

注意事项：

phObj 使用结束后，使用 closeHandle 释放内存。

A.1.4.11 uint CPKIManager::digestUpdate(PKIObjectHandle hObj, const QByteArray & in)

函数说明：

多组数据摘要。

参数说明：

输入值：

hObj参数为摘要对象句柄。

in参数为数据。

返回值：如果成功，则返回 0；否则返回错误码。

A.1.4.12 uint CPKIManager::encrypt(PKIObjectHandle hObj, const QByteArray & in, QByteArray & out)

函数说明：

单组数据加密。

参数说明：

输入值：

hObj参数为加密对象句柄。

in参数为原文数据。

out参数为密文数据。

返回值：如果成功，则返回 0；否则返回错误码。

A.1.4.13 uint CPKIManager::encryptFinal(PKIObjectHandle hObj, QByteArray & out)

函数说明：

结束加密。

参数说明：

输入值：

hObj参数为加密对象句柄。

out参数为加密后的数据缓冲区。

返回值：如果成功，则返回 0；否则返回错误码。

A.1.4.14 uint CPKIManager::encryptInit(PKIProviderHandle hProvider, uint uAlgID, const QByteArray & key, const QByteArray & iv, PKIObjectHandle * phObj)

函数说明：

加密初始化。

参数说明：

输入值：

hProvider参数为密钥运算提供者对象句柄。

uAlgID参数为对称算法标识。

key参数为对称密钥。

iv参数为初始向量。

phObj参数为加密对象句柄。

返回值：如果成功，则返回 0；否则返回错误码。

注意事项：

phObj使用结束后，使用closeHandle释放内存。

A.1.4.15 uint CPKIManager::encryptUpdate(PKIObjectHandle hObj, const QByteArray & in, QByteArray & out)

函数说明：

多组数据加密。

参数说明：

输入值：

hObj参数为加密对象句柄。

in参数为一组原文数据。

out参数为一组密文数据。

返回值：如果成功，则返回 0；否则返回错误码。

A.1.4.16 uint CPKIManager::enumCert(PKIObjectHandle hObj, QList & certList)

函数说明：

枚举证书。

参数说明：

输入值：

hObj参数为签名或验签对象句柄。

certList参数为证书。

返回值：如果成功，则返回 0；否则返回错误码。

A.1.4.17 uint CPKIManager::enumProvider(QStringList & providerNames)

函数说明：

枚举的密钥运算提供者。

参数说明：

输入值：

providerNames参数为密钥运算的提供者名称列表。

返回值：如果成功，则返回 0；否则返回错误码。

A.1.4.18 uint CPKIManager::openProvider(QString providerName, PKIProviderHandle * phProvider)

函数说明：

打开指定的密钥运算提供者。

参数说明：

输入值：

providerName参数为密钥运算的提供者名称。providerName为空字符串，打开默认提供者。

phProvider参数为密钥运算提供者对象句柄。

返回值：如果成功，则返回 0；否则返回错误码。

A.1.4.19 uint CPKIManager::password2Key(const QString & pass, const QByteArray & salt, int keylen, QByteArray & out)

函数说明:

由密码生成对称密钥。

参数说明:

输入值:

pass参数为密码。

salt参数为混淆值。

keylen参数为密钥长度（字节数）。

out参数为密钥。

返回值: 如果成功, 则返回 0; 否则返回错误码。

A.1.4.20 uint CPKIManager::selectCert(PKIObjectHandle hObj, const CPKICertInfo & cert)

函数说明:

选择签名证书。

参数说明:

输入值:

hObj参数为签名或验签对象句柄。

cert参数为证书。

返回值: 如果成功, 则返回 0; 否则返回错误码。

A.1.4.21 uint CPKIManager::setCertFilter(PKIObjectHandle hObj, uint uType, QString strFilter)

函数说明:

设置证书的过滤条件。

参数说明:

输入值:

hObj参数为签名或验签对象句柄。

uType参数为过滤条件类型。

strFilter参数为过滤条件。

返回值: 如果成功, 则返回 0; 否则返回错误码。

A.1.4.22 uint CPKIManager::signature(PKIObjectHandle hObj, const QByteArray & in, QByteArray & out)

函数说明:

单组签名。

参数说明:

输入值:

hObj参数为签名对象句柄。

in参数为签名原文。

out参数为签名结果。

返回值: 如果成功, 则返回 0; 否则返回错误码。

A.1.4.23 uint CPKIManager::signatureFinal(PKIObjectHandle hObj, QByteArray & out)

函数说明:

分组签名结束。

参数说明:

输入值:

hObj参数为签名对象句柄。

out参数为签名结果。

返回值：如果成功，则返回 0；否则返回错误码。

A.1.4.24 uint CPKIManager::signatureInit(PKIProviderHandle hProvider, uint uAlgID, PKIObjectHandle * phObj)

函数说明：

签名初始化

参数说明：

输入值：

hProvider参数为密钥运算提供者对象句柄。

uAlgID参数为签名使用的摘要算法。

phObj参数为签名对象句柄。

返回值：如果成功，则返回 0；否则返回错误码。

注意事项：

phObj使用结束后，使用closeHandle释放内存。

A.1.4.25 uint CPKIManager::signatureUpdate(PKIObjectHandle hObj, const QByteArray & in)

函数说明：

多组签名。

参数说明：

输入值：

hObj参数为签名对象句柄。

in参数为签名原文。

返回值：如果成功，则返回 0；否则返回错误码。

A.1.4.26 uint CPKIManager::verifyBlock(PKIObjectHandle hObj, const QByteArray & in, const QByteArray & out)

函数说明：

单组验证签名。

参数说明：

输入值：

hObj参数为验签名对象句柄。

in参数为签名数据。

out参数为签名结果。

返回值：如果成功，则返回 0；否则返回错误码。

A.1.4.27 uint CPKIManager::verifyFinal(PKIObjectHandle hObj, const QByteArray & out)

函数说明：

单组验证签名。

参数说明：

输入值：

hObj参数为验签名对象句柄。

out参数为签名结果。

返回值：如果成功，则返回 0；否则返回错误码。

A.1.4.28 `uint CPKIManager::verifyInit(PKIProviderHandle hProvider, uint uAlgID, PKIObjectHandle * phObj)`

函数说明:

验签名初始化。

参数说明:

输入值

`hProvider`参数为密钥运算提供者对象句柄。

`uAlgID`参数为签名使用的摘要算法。

`phObj`参数为验签名对象句柄。

返回值: 如果成功, 则返回 0; 否则返回错误码。

注意事项:

`phObj`使用结束后, 使用`closeHandle`释放内存。

A.1.4.29 `uint CPKIManager::verifyPin(PKIObjectHandle hObj, QString strPIN)`

函数说明:

验证 PIN。

参数说明:

输入值

`hObj`参数为签名对象句柄。

`strPIN`参数为Pin码。

返回值: 如果成功, 则返回 0; 否则返回错误码。

A.1.4.30 `uint CPKIManager::verifyUpdate(PKIObjectHandle hObj, const QByteArray & in)`

函数说明:

验证 PIN。

参数说明:

输入值:

`hObj`参数为验签名对象句柄。

`in`参数为签名原文。

返回值: 如果成功, 则返回 0; 否则返回错误码。

A.2 设备管理

设备管理模块对设备的硬件设备进行管理。闹钟提醒管理、电量管理、闪光灯、屏幕设置、存储设备管理、时间、USB管理接口。包含如下类:

A.2.1 CAlarmManager

类说明:

CAlarmManager 为定时器设置, 该类提供了设置和取消定时提醒接口。

A.2.1.1 `int CAlarmManager::setAlarm(QDateTime firstTime, int interval, bool repeat, QString notifyId = "")`

函数说明:

设置定时提醒接口。

参数说明:

输入值:

`firstTime`为从何时开始。

`interval`为从`firstTime`开始, 过`interval`时间后开始提醒, 单位分钟。当`repeat`为`true`时, 该参数也表示该alarm将会以`interval`为周期提醒, 即自动设置一个每隔`interval`是长都提醒的定时器。

repeat参数的取值：true重复提醒，间隔为interval；false单次提醒。

notifyId为定时器标识；不传该参数时，提取sopId代替，service无sopId的必须传参。

返回值：

返回一个id给调用者。

注意事项：

需要申请OS.permission.ACCESS_ALARM 敏感权限才可以调用。

A.2.2 CBatteryManager

类说明：

CBatteryManager 类提供了关于电池状态和电量大小的查询接口。当电池可用时，电池电量和状态查询的接口才是有效的，否则无效。

A.2.2.1 int CBatteryManager::capacity() const

函数说明：

获取电池的总容量。

参数说明：

输入值：

无

返回值：

电池电量的总容量，单位是mAh。

A.2.2.2 bool CBatteryManager::isValid() const

函数说明：

判断电池是否有效。

参数说明：

输入值：

无

返回值：

电池可用返回true，否则返回false。

A.2.3 CFlashManager

类说明：

CFlashManager类提供了关于相机闪光灯状态的设置接口。

A.2.3.1 bool CFlashManager::openFlash(bool active) const

函数说明：

设置闪光灯的开关状态：

参数说明：

输入值：

active为true时，打开闪光灯；为false时，为关闭闪光灯。

返回值：

成功为true，失败为false。

注意事项：

调用接口时，需要申请OS.permission.ACCESS_FLASHLIGHT 敏感权限。

A.2.3.2 bool CFlashManager::setFlashActive(bool active)

函数说明：

设置闪光灯的使能状态

参数说明：

输入值

active为true时，闪光灯正常使用，为false时，闪光灯失效。

返回值：

成功为true，失败为false。

注意事项：

将闪光灯设置为失效的状态，是为了避免多个应用同时使用闪光灯。如：相机启动的时候，闪光灯是为失效状态。调用接口时，需要申请OS.permission.ACCESS_FLASHLIGHT敏感权限。

A.2.4 CStorageManager

类说明：

CStorageManager类提供了关于设备存储大小的查询接口，包括系统内存、内置存储和外置存储。

A.2.4.1 qint64 CStorageManager::memFreeSize()

函数说明：

获取内存的可用空间大小。

参数说明：

输入值：无

返回值：单位为字节。

A.2.4.2 qint64 CStorageManager::memTotalSize()

函数说明：

获取内存的总空间大小。

参数说明：

输入值：无

返回值：单位是字节。

A.2.4.3 qint64 CStorageManager::storageFreeSize(const StorageFS & fs)

函数说明：

获取设备的可用空间大小

参数说明：

输入值：

fs为系统设备。

返回值：

单位为字节。当获取无效或设备不存在时，返回 -1。

A.2.4.4 qint64 CStorageManager::storageTotalSize(const StorageFS & fs)

函数说明：

获取设备的总空间大小。

参数说明：

输入值：

fs为系统设备。

返回值：

单位为字节。当获取无效或设备不存在时，返回 -1。

A.3 应用框架管理

系统管理对整个系统的应用、进程、事件等管理。运行应用程序管理、应用管理（安装、更新、卸载）、进程管理、窗口事件分发、截屏、存储格式化、系统升级、电源管理、系统提示音配置与播放等。

A.3.1 CAppInfo

类说明：

CAppInfo类是 CUiAppInfo 类、CServiceInfo 类和 CHtmlAppInfo 的基类，提供了 uiapp、service 和 htmlapp 通用的信息。

A.3.1.1 QStringList CAppInfo::background() const [virtual]

函数说明：

获取应用程序的 background 设置列表。

参数说明：

输入值：

无。

返回值：

background 值列表。

注意事项：

background用于设置应用程序转入后台后是否暂停，目前取值范围为{continue}。该值取的是 sopconfig.xml 中定义在uiapp或者htmlapp标签下的移动智能终端安全操作系统API接口技术规范:background属性。如果该值没有设置，则默认应用程序转入后台后会暂停其运行。

A.3.1.2 QList CAppInfo::docTypeInfoList() const

函数说明：

获取应用程序或者服务支持的 doctype 信息列表。

参数说明：

输入值：无。

返回值：doctype信息列表。

注意事项：

doctype取sopconfig.xml中定义在uiapp、service或者htmlapp标签中的doctype标签属性。

A.3.1.3 QString CAppInfo::exec() const

函数说明：

获取应用程序或者服务的可执行程序路径，该路径为绝对路径。

参数说明：

输入值：无。

返回值：可执行程序绝对路径。

注意事项：

可执行程序路径取sopconfig.xml中定义在uiapp、service或者htmlapp标签下的移动智能终端安全操作系统API接口技术规范:exec属性。移动智能终端安全操作系统API接口技术规范:exec属性是相对路径，经过转换后成为绝对路径。

A.3.1.4 QList CAppInfo::mutiDocTypeInfoList() const

函数说明：

获取应用程序或者服务支持的multidoc信息列表。

参数说明:

输入值: 无。

返回值: multidoc信息列表。

注意事项:

multidoc取sopconfig.xml中定义在uiapp、service或者htmlapp标签中的multidoc标签属性。

A.3.1.5 QList CAppInfo::receiverInfoList() const

函数说明:

获取应用程序或者服务支持的receiver信息列表。

参数说明:

输入值: 无。

返回值: receiver 信息列表。

注意事项:

receiver取sopconfig.xml中定义在uiapp、service或者htmlapp标签中的receiver标签属性。

A.3.1.6 QString CAppInfo::sopid() const

函数说明:

获取应用程序的SOP包标识字符串, 对所有应用程序而言, SOP包标识字符串非空且唯一。

参数说明:

输入值: 无

返回值: SOP包标识字符串。

注意事项:

SOP包标识字符串取sopconfig.xml中定义在sopconfig标签下的移动智能终端安全操作系统API接口技术规范:sopid属性。

A.3.2 CHtmlAppInfo

类继承自 CAppInfo 类, 它提供了应用程序的相关信息。

A.3.2.1 QStringList CHtmlAppInfo::background() const [virtual]

函数说明:

获取htmlapp应用程序的background设置列表。

参数说明:

输入值: 无

返回值: background 值列表。

注意事项:

background用于设置应用程序转入后台后是否暂停, 目前取值范围为{continue}。该值取的是sopconfig.xml中定义在htmlapp标签下的移动智能终端安全操作系统API接口技术规范:background属性。如果该值没有设置, 则默认应用程序转入后台后会暂停其运行。

A.3.2.2 bool CHtmlAppInfo::launcher() const [virtual]

函数说明:

获取htmlapp应用程序是否显示在主界面。

参数说明:

输入值: 无

返回值: true 表示显示在主界面, false 表示不显示在主界面。

注意事项:

launcher是取sopconfig.xml中定义在htmlapp标签下的OS:launcher属性。

A.3.2.3 QString CHtmlAppInfo::splash() const [virtual]

函数说明:

获取htmlapp应用程序的splash图片绝对路径。

参数说明:

输入值: 无

返回值: 图片的绝对路径。

注意事项:

splash是取sopconfig.xml中定义在htmlapp标签下的OS:splash属性。

A.3.2.4 QList CHtmlAppInfo::urlInfoList() const [virtual]

函数说明:

获取sopconfig.xml中定义在htmlapp标签中的url信息列表。

参数说明:

输入值: 无

返回值: url 信息列表。

注意事项:

url是取sopconfig.xml中定义在htmlapp标签下的url标签段数据。

A.3.3 CPackageInfo

CPackageInfo 类提供应用程序包信息。

PackageError枚举类型用来描述应用程序的安装错误信息，其值如下表所示：

变量名	值	描述
CPackageInfo::NoError	1000	安装成功
CPackageInfo::ErrorDownload	1001	下载错误
CPackageInfo::ErrorDownloadTimeout	1002	下载超时
CPackageInfo::ErrorDownloadFileNotProvided	1003	无法获取下载文件
CPackageInfo::ErrorDownloadUnknow	1004	未知下载失败错误
CPackageInfo::ErrorInstallFailed	1005	安装失败
CPackageInfo::ErrorInstallUnknow	1006	未知安装错误
CPackageInfo::ErrorCancelFailed	1007	安装中任务无法取消
CPackageInfo::ErrorInstallAppNotInstall	1008	应用程序未安装
CPackageInfo::ErrorInstallPresetSystemAppCanNotRemove	1009	预装系统应用无法卸载
CPackageInfo::ErrorInstallInstallerProgramException	1010	安装程序异常
CPackageInfo::ErrorInstallPackageNotExist	1011	安装包不存在
CPackageInfo::ErrorInstallUnrecognizedPkgType	1012	无法识别的包类型
CPackageInfo::ErrorInstallCertVerifyFail	1013	签名验证失败
CPackageInfo::ErrorInstallAppConfigXmlNotExist	1014	包配置文件不存在
CPackageInfo::ErrorInstallIllegalInstalledFiles	1015	安装包有非法文件
CPackageInfo::ErrorInstallDecompressPkgFail	1016	安装失败

1		
CPackageInfo::ErrorInstallCheckSopidLengthFail	1017	应用程序身份标识长度检查失败
CPackageInfo::ErrorInstallStorageNotEnough	1018	磁盘空间不足
CPackageInfo::ErrorInstallReleaseCertAppInstallFail	1019	Release签名认证失败
CPackageInfo::ErrorInstallTimeout	1020	安装超时
CPackageInfo::ErrorInstallAppNotInWhitelist	1021	应用程序不在白名单

PackageStatus

本枚举类型用来描述应用程序安装包的下载，其值如下表所示：

变量名	值	描述
CPackageInfo::DownloadNotReady	0	下载未开始
CPackageInfo::DownloadReady	1	下载准备就绪
CPackageInfo::DownloadOngoing	2	正在下载
CPackageInfo::DownloadPaused	3	暂停下载
CPackageInfo::DownloadResumed	4	恢复下载
CPackageInfo::DownloadFinished	5	完成下载
CPackageInfo::DownloadCancelled	6	取消下载
CPackageInfo::InstallOngoing	7	正在安装
CPackageInfo::InstallFinished	8	完成安装
CPackageInfo::DeleteFinished	9	取消安装
CPackageInfo::UpdateFinished	10	更新安装
CPackageInfo::UpdateReady	11	更新准备状态
CPackageInfo::UpdateOngoing	12	正在更新
CPackageInfo::UpdatePaused	13	更新暂停

A.3.3.1 QString CPackageInfo::backupExec() const

函数说明：

获取sopconfig.xml中定义在application标签下的backup标签字段值。

参数说明：

输入值：无

返回值：可执行程序绝对路径。

A.3.3.2 QString CPackageInfo::configFilePath() const

函数说明:

获取应用程序的sopconfig.xml文件绝对路径。

参数说明:

输入值: 无

返回值: sopconfig.xml文件绝对路径。

A.3.3.3 CDownloadInfo CPackageInfo::downloadInfo() const

函数说明:

获取应用程序的下载状态信息。

参数说明:

输入值: 无

返回值: 应用程序下载信息数据结构 CDownloadInfo 。

注意事项:

下载状态信息包括: 应用程序安装包大小、已下载大小、下载百分比、下载速度、应用程序名称和应用程序图标。

A.3.3.4 QStringList CPackageInfo::permissionList() const

函数说明:

获取应用程序申请的敏感权限列表。

参数说明:

输入值: 无

返回值: 敏感权限列表。

注意事项:

敏感权限取sopconfig.xml中定义在uses-permission标签的属性。

A.3.4 CUiAppInfo

CUiAppInfo类继承自 CAppInfo 类, 它提供了应用程序的相关信息。

A.3.4.1 QStringList CUiAppInfo::background() const [virtual]

函数说明:

获取uiapp应用程序的background设置列表。

参数说明:

输入值: 无

返回值: background值列表。

注意事项:

background用于设置应用程序转入后台后是否暂停, 目前取值范围为{continue}。该值取的是sopconfig.xml中定义在uiapp标签下的OS:background属性。如果该值没有设置, 则默认应用程序转入后台后会暂停其运行。

A.3.4.2 bool CUiAppInfo::launcher() const [virtual]

函数说明:

获取uiapp应用程序是否显示在主界面。

参数说明:

输入值: 无

返回值: true 表示显示在主界面, false 表示不显示在主界面。

注意事项:

launcher是取sopconfig.xml中定义在uiapp标签下的OS:launcher属性。

A.3.4.3 QString CUiAppInfo::splash() const [virtual]

函数说明:

获取uiapp应用程序的splash图片绝对路径。

参数说明:

输入值: 无

返回值: 图片的绝对路径。

注意事项:

splash是取sopconfig.xml中定义在uiapp标签下的OS:splash属性。

A.3.5 CProcessManager

CProcessManager为进程操作类。

A.3.5.1 int CProcessManager::killProcessByPid(int pid)

函数说明:

根据进程标识符杀进程。

参数说明:

输入值: pid为进程标识符。

返回值: 成功为0, 失败为非0错误码。

A.3.5.2 QList CProcessManager::processList()

函数说明:

取得pm管理的所有应用进程列表。

参数说明:

输入值: 无

返回值: 进程标识符列表。

A.3.5.3 QString CProcessManager::sopidByPid(int pid)

函数说明:

取得指定进程的SOP包标识。

参数说明:

输入值: pid为进程标识符。

返回值: SOP包标识。

A.3.6 CProfileManager

CProfileManager类提供了关于第三方应用铃声的查询、设置接口。

A.3.6.1 QString CProfileManager::ringTone()

函数说明:

获取铃声音乐路径。

参数说明:

输入值：无
返回值：返回铃声音乐路径。

A.3.6.2 `int CProfileManager::setRingtone(const QString & path)`

函数说明：

根据铃声音乐路径设置铃声。

参数说明：

输入值：path为铃声音乐路径。
返回值：成功为0，失败为非0错误码。

A.4 通短联管理

通短联管理对系统的通话、短信、联系人进行管理。黑白名单管理、通过记录、联系人、短信（垃圾短信）管理。

A.4.1 CCallLogManager

CCallLogManager是通话记录管理器的基类。

A.4.1.1 `CCallLogManager * CCallLogManager::instance() [static]`

函数说明：

单例实例化。

参数说明：

输入值：无。
返回值：返回CCallLogManager的实例。

A.4.2 ContactData。

为联系人节点类

A.4.2.1 `ContactData::ContactData(quint32 contactId, QContact & contact)`

函数说明：

构造函数。

参数说明：

输入值：
contactId为联系人ID。
contact为联系人数据。

返回值：
ContactData

A.4.3 CMessageManager

提供垃圾短信获取，删除，恢复接口。

A.4.3.1 `CMessageManager * CMessageManager::instance() [static]`

函数说明：

获取CMessageManager实例。

函数。

参数说明：

输入值：无

返回值：CMessageManager指针。

A.5 网络管理

网络管理对系统的网络通信模块进行管理。定位、2g/3g/4g移动蜂窝网络信息（信号强度等）、wifi连接等。

A.5.1 CNetworkManager

CNetworkManager 类提供网络状态获取的相关接口。

A.5.1.1 CNetworkManager::NetworkType CNetworkManager::currNetworkType()

函数说明：

查询当前网络类型

参数说明：

输入值：无。

返回值：网络类型。

注意事项：

需要申请OS.permission.READ_NETWORK_STATE敏感权限才可以调用。

A.5.1.2 bool CNetworkManager::isNetworkAvailabe()

函数说明：

查询当前网络是否可用。

参数说明：

输入值：无。

返回值：可用返回 true，不可用返回 false。

注意事项：

需要申请OS.permission.READ_NETWORK_STATE敏感权限才可以调用。

A.5.1.3 bool CNetworkManager::isWifiConnected()

函数说明：

查询返回当前WiFi是否处于连接状态。

参数说明：

输入值：无。

返回值：WiFi连接返回 true，否则返回 false。

注意事项：

需要申请OS.permission.READ_NETWORK_STATE敏感权限才可以调用。

A.5.1.4 bool CNetworkManager::setCellularEnabled(const bool flag)

函数说明：

设置数据网络状态

参数说明：

输入值：flag: true 表示可用， false 表示禁用。

返回值：设置成功返回 true，失败返回 false。

注意事项：

需要申请OS.permission.ADMIN_DATA_NETWORK敏感权限才可以调用。

A.5.2 WifiNetworkTechnology

WifiNetworkTechnology类包含了判断wifi状态、扫描wifi热点等。

A.5.2.1 bool WifiNetworkTechnology::powered() const

函数说明：

查询当前wifi状态

参数说明：

输入值：无

返回值：true为wifi已打开，false为未打开。

A.5.2.2 bool WifiNetworkTechnology::connected() const

函数说明：

返回当前是否连接wifi热点

参数说明：

输入值：无

返回值：true为已连接，false为未连接。

A.5.3 WifiNetworkService

类说明：

类WifiNetworkService中包含了连接某热点、断开与某热点的连接状态

A.5.3.1 bool WifiNetworkService::requestConnect()

函数说明：

连接当前选中的wifi热点

参数说明：

输入值：无

返回值：true为成功，false为失败。

A.5.3.2 bool WifiNetworkService::requestDisconnect()

函数说明：

断开已连接的wifi热点

参数说明：

输入值：无

返回值：true为成功，false为失败。

A.5.4 QBluetoothLocalDevice

QBluetoothLocalDevice用于访问本地蓝牙设备，提供函数去获取或者设置本地蓝牙设备的状态。

A.5.4.1 QList<QBluetoothHostInfo> QBluetoothLocalDevice::allDevices()

函数说明：

获取所有本机蓝牙设备信息

参数说明:

输入值: 无

返回值: 蓝牙设备列表

A. 5.4.2 QString QBluetoothLocalDevice::name() const

函数说明:

返回使用的蓝牙设备的名称

参数说明:

输入值: 无

返回值: 设备名字符串

A. 5.5 QBluetoothDeviceDiscoveryAgent

QBluetoothDeviceDiscoveryAgent用于发现近处的蓝牙设备。如果需要异步获取设备列表,可以连接deviceDiscovered()信号,如果需要同步获取一个完整的设备列表,可以连接finished()信号,并调用discoveredDevices()方法。

A. 5.5.1 bool QBluetoothDeviceDiscoveryAgent::start()

函数说明:

开始扫描蓝牙设备参数说明:

参数说明:

输入值: 无

返回值: 成功返回true,失败返回false

注意事项:

当发现设备时,即会发出信号deviceDiscovered(),当扫描完成则发送信号finished()。

A. 5.5.2 bool QBluetoothDeviceDiscoveryAgent::stop()

函数说明:

停止扫描蓝牙设备

参数说明:

输入值: 无

返回值: 成功返回true,失败返回false

注意事项:

当扫描停止则发出信号cancel(),如果在cancel()信号调用之前调用了start()函数,则start()函数会被丢弃。

A. 5.6 QBluetoothHostInfo

QBluetoothHostInfo封装本地蓝牙设备的静态详情信息(名称/地址),并且有提供修改名称地址的接口。

A. 5.6.1 QBluetoothAddress QBluetoothHostInfo::address() const

函数说明:

获取本地蓝牙地址

参数说明:

输入值: 无

返回值：返回一个QBluetoothAddress类型的蓝牙地址。

A. 5.6.2 QString QBluetoothHostInfo::name() const

函数说明：

获取蓝牙设备名称

参数说明：

输入值：无

返回值：返回主机信息对象的可见名字。

A. 5.7 QGeoPositionInfoSource

QGeoPositionInfoSource类是一个描述位置更新的抽象基类。

A. 5.7.1 bool QGeoPositionInfoSource::requestUpdate(int timeout = 0)

函数说明：

请求当前位置信息

参数说明：

输入值：无

返回值：成功返回true，失败返回false。

A. 5.7.2 bool QGeoPositionInfoSource::startUpdates()

函数说明：

启动正常位置更新

参数说明：

输入值：无

返回值：成功返回true，失败返回false。

A. 5.7.3 bool QGeoPositionInfoSource::stopUpdates()

函数说明：

停止当前位置更新

参数说明：

输入值：无

返回值：成功返回true，失败返回false。

A. 5.8 QGeoCoordinate

A. 5.8.1 double QGeoCoordinate::altitude() const

函数说明：

查询海拔高度，如果海拔高度没有设置，返回未定义的值

参数说明：

输入值：无

返回值：海拔高度

A. 5.8.2 bool QGeoCoordinate::setAltitude(double altitude)

函数说明:

设置海拔高度

参数说明:

输入值: 无

返回值: 成功返回true, 失败返回false。

A.5.8.3 double QGeoCoordinate::latitude () const

函数说明:

查询纬度, 在十进制度。如果该纬度没有被设置, 返回值是未定义的。一个正纬度指示北半球, 负纬度表明南半球

参数说明:

输入值: 无

返回值: 纬度值

A.5.8.4 bool QGeoCoordinate::setLatitude(double latitude)

函数说明:

设置纬度(以十进制度)为纬度。该值应在WGS84基准。如果要纬度有效, 纬度必须介于-90至90之间

参数说明:

输入值: 无

返回值: 成功返回true, 失败返回false。

A.5.8.5 double QGeoCoordinate::longitude()const

函数说明:

查询经度

参数说明:

输入值: 无

返回值: 经度值

A.5.8.6 bool QGeoCoordinate::setLongitude(double longitude)

函数说明:

设置经度(在十进制度)到经度。该值应在WGS84基准。如果要经度有效, 经度必须介于-180至180之间

参数说明:

输入值: 无

返回值: 成功返回true, 失败返回false。

A.5.9 CSystemLocationManager

OS提供CSystemLocationManager类提供定位服务的相关接口。

A5.9.1 CSystemLocationManager::LocationMode CSystemLocationManager::locationMode()

函数说明:

查询当前定位方式

参数说明:

输入值：无

返回值：

属性名	类型	描述
CSystemLocationManager::NoneMode	0	无定位
CSystemLocationManager::SatelliteMode	1	卫星定位
CSystemLocationManager::NetworkMode	2	网络定位
CSystemLocationManager::SatelliteNetworkMode	3	卫星+网络定位

A.5.9.2 bool CSystemLocationManager::setLocationMode(CSystemLocationManager::LocationMode mode)

函数说明：

设置定位方式参数说明：

参数说明

输入值：mode，定位方式。

返回值：设置成功返回true，失败返回false。

A.6 多媒体

多媒体完成没有界面的音乐播放。包含如下类：

A.6.1 CNGFManager

CNGFManager类提供了关于播放声音和振动的接口。

Ngf服务支持的事件类型：

变量名	值	描述
CNGFManager::NGFDefault	0	默认事件
CNGFManager::NGFRingtone	1	来电铃声事件
CNGFManager::NGFVoipRingtone	2	VoIP铃声事件
CNGFManager::NGFSms	3	收到短信事件
CNGFManager::NGFSmsDelivered	4	发送短信事件
CNGFManager::NGFSmsFg	5	前台收到短信事件
CNGFManager::NGFClock	6	闹钟提醒事件
CNGFManager::NGFCalendar	7	日历事件
CNGFManager::NGFEmail	8	收到邮件事件
CNGFManager::NGFEmailSend	9	发送邮件事件
CNGFManager::NGFNotifier	10	通知事件
CNGFManager::NGFTouch	11	触摸按键事件
CNGFManager::NGFVibrating	12	连续振动事件
CNGFManager::NGFBatteryLow	13	低电事件
CNGFManager::NGFVibrator	14	短轻振动事件
CNGFManager::NGFBluetooth	15	蓝牙事件
CNGFManager::NGFCall	16	通话事件
CNGFManager::NGFDialpad	17	拨号键盘事件
CNGFManager::NGFChargingStarted	18	开始充电事件

CNGFManager::NGFRechargeBattery	19	重新充电事件
CNGFManager::NGFWrongCharger	20	充电错误事件
CNGFManager::NGFChatFg	21	前台聊天事件
CNGFManager::NGFChat	22	聊天事件
CNGFManager::NGFAccessoryConnected	23	附件连接事件
CNGFManager::NGFAccessoryDisconnected	24	附件断开事件
CNGFManager::NGFInformationSnd	25	信息发送事件
CNGFManager::NGFInformationStrong	26	紧急信息事件
CNGFManager::NGFInformationTaction	27	信息接触事件
CNGFManager::NGFMultimedia	28	多媒体事件
CNGFManager::NGFScreenLock	29	锁屏事件
CNGFManager::NGFSystem	30	系统事件
CNGFManager::NGFWarningSnd	31	发送警告事件
CNGFManager::NGFWarningStrong	32	紧急警告事件
CNGFManager::NGFWarningTaction	33	警告接触事件
CNGFManager::NGFTypeMax	34	Ngf事件总数

A. 6.1.1 bool CNGFManager::connect() [virtual]

函数说明:

连接Ngf服务。

参数说明:

输入值: 无

返回值: true: 连接Ngf服务成功, false: 连接Ngf服务失败。

A. 6.1.2 bool CNGFManager::disconnect() [virtual]

函数说明:

断开与Ngf服务的连接。

参数说明:

输入值: 无

返回值: true: 连接Ngf服务成功, false: 连接Ngf服务失败。

A. 6.1.3 bool CNGFManager::pause(const quint32 & eventId) [virtual]

函数说明:

根据事件标识符向Ngf服务申请暂停播放。

参数说明:

输入值: eventId为事件标识符。

返回值: true: 成功, false: 失败。

A. 6.1.4 quint32 CNGFManager::play(NGFEventType event) [virtual]

函数说明:

根据事件类型向Ngf服务申请播放。

参数说明:

输入值: event为事件类型。

返回值: 事件标识符如果为0, 则表示未连接Ngf服务或不支持的事件类型。

A.6.1.5 `bool CNGFManager::resume(const quint32 & eventId) [virtual]`

函数说明:

根据事件标识符向Ngf服务申请恢复播放。

参数说明:

输入值: `eventId`为事件标识符。

返回值: `true`: 成功, `false`: 失败。

A.6.1.6 `bool CNGFManager::stop(const quint32 & eventId) [virtual]`

函数说明:

根据事件id向Ngf服务申请停止播放。

参数说明:

输入值: `eventId`为事件标识符。

返回值: `true`: 成功, `false`: 失败。

A.7 消息交互管理

消息通知管理包含通知、角标、对话框、提示信息 (Toast)、推送 (push)。包含如下类:

A.7.1 CAlertDialogManager

CAlertDialogManager为警示对话框操作类。

A.7.1.1 `bool CAlertDialogManager::close()`

函数说明:

关闭警示对话框。

参数说明:

输入值: 无

返回值: `true`: 成功, `false`: 失败。

A.7.1.2 `int CAlertDialogManager::show(const QString & title, const QString & text, const QStringList & buttons)`

函数说明:

显示一个警示对话框。

参数说明:

输入值:

`title`: 设置对话框标题。

`text`: 设置对话框内容。

`buttons`: 设置对话框需要显示的按钮列表。

返回值:

用户点击的第几个按钮。

A.7.2 CInfoDialogManager

CInfoDialogManager为系统信息对话框操作类。

A.7.2.1 `bool CInfoDialogManager::close()`

函数说明:

关闭系统信息对话框。

参数说明：

输入值：无

返回值：true：成功，false：失败。

A.7.2.2 int CInfoDialogManager::show(const QString & title, const QString & text, const QStringList & buttons)

函数说明：

显示一个系统信息对话框。

参数说明：

输入值：

title：设置对话框标题。

text：设置对话框内容。

buttons：设置对话框需要显示的按钮列表。

返回值：用户点击的第几个按钮。

A.7.3 CInputDialogManager

CInputDialogManager为输入对话框操作类。

A.7.3.1 int CInputDialogManager::show(const QString & title, char * result, const QStringList & buttons)

函数说明：

显示一个输入对话框。

参数说明：

输入值：

title：设置对话框标题。

result：取得输入内容。

buttons：设置对话框需要显示的按钮列表。

返回值：

用户点击的第几个按钮。

A.7.4 CAbstractNotification

CAbstractNotification是各种通知的抽象类型。

enum Attribute通知允许设置特定的属性，来控制通知发送时的具体表现。其值如下表显示：

变量名	值	描述
CAbstractNotification::StayOnStatusBar	0x01	通知停留在状态栏上(兼容方法，目前无具体意义)
CAbstractNotification::HideNotification	0x02	通知不显示在状态栏下拉列表中
CAbstractNotification::HideDisplayTime	0x04	通知发送时间不显示在下拉列表上
CAbstractNotification::DeleteOnFinished	0x08	通知在状态栏显示完成后，自动删除
CAbstractNotification::DeleteOnReboot	0x10	通知在手机重启后，自动删除

A.7.4.1 bool CAbstractNotification::setAction(const QString & action)

函数说明:

这个方法允许你设置一个自定义的触发参数。

当用户在状态栏触发某个通知的时候，会启动对应的应用界面，而触发参数将作为传入参数一同执行。

参数说明:

输入值: action: 自定义的字符串参数。

返回值: true: 成功, false: 失败。

A.7.4.2 bool CAbstractNotification::setAttribute(Attribute attribute, bool on)

函数说明:

此方法允许设置指定属性是否启用。

参数说明:

输入值:

attribute: 属性的枚举。

on: 决定属性是否启用的开关。

返回值:

true: 成功, false: 失败。

A.7.4.3 bool CAbstractNotification::setAttributes(Attributes attributes)

函数说明:

此方法允许设置一个属性集合，当前属性集合将会替换之前的属性集合。

参数说明:

输入值: attributes: 属性集合。

返回值: true: 成功, false: 失败。

A.7.5 CBadge

CBadge用于设置主页角标值。

A.7.5.1 bool CBadge::setUiValue(const QString & uiAppId, int value)

函数说明:

此方法用于设置界面的角标值。

参数说明:

输入值:

uiAppId: UiAppId。

value: 角标值。如果 value <= 0, 表示移除角标值。

返回值:

true: 成功, false: 失败。

A.7.6 CNotification

CNotification为通知类。提供发送给系统通知的功能，开发者可以通过这个类，实体化一个“通知”对象，并将这条通知通过CNotificationManager发送出去。

A.7.6.1 bool CNotification::setDisplayTime(const QDateTime & time)

函数说明:

设置成指定的显示时间。

参数说明:

输入值: time: 时间。

返回值: true: 成功, false: 失败

注意事项:

默认情况下, 状态栏显示的每个消息中, 都会显示一个消息发出的时间。这个时间默认使用通知发出时的系统时间。在某些特殊情况下需要指定显示的发送时间的, 可以通过这个方法设置一个自定义的时间。这个时间将替代系统时间, 显示在状态栏通知上。

A.7.6.2 bool CNotification::setMarqueeText(const QString & marqueeText)

函数说明:

设置动态提示文字。

参数说明:

输入值: marqueeText: 动态提示文字。

返回值: true: 成功, false: 失败

注意事项:

当设置动态提示文字的时候, 状态栏会用动态滚动效果(跑马灯)显示设置的问题。如果不设置, 不会启用动态滚动提示。

A.7.6.3 bool CNotification::setSound(const QString & fileName)

函数说明:

设置通知播放声音。如果不设置声音文件, 则静音。

参数说明:

输入值: fileName: 声音的文件名。

返回值: true: 成功, false: 失败。

A.7.6.4 bool CNotification::setTitle(const QString & title)

函数说明:

设置标题。

参数说明:

输入值: title: 标题。

返回值: true: 成功, false: 失败。

A.7.7 CNotificationManager

类说明:

CNotificationManager为通知服务发送类。

注意事项: CNotificationManager的所有方法都需要申请OS.permission.ACCESS_NOTIFICATION敏感权限才可以调用。

A.7.7.1 bool CNotificationManager::removeAllNotifications()

函数说明:

此方法可以删除当前应用的全部通知。

参数说明:

输入值: 无

返回值: 成功执行返回 true; 执行失败返回 false。

A.7.7.2 bool CNotificationManager::removeNotifications(const QStringList & updateIds)

函数说明:

删除指定的通知列表输入值:

参数说明:

输入值: updateIds: 通知 ID 列表。

返回值: 成功执行返回 true; 执行失败返回 false。

A.7.7.3 bool CNotificationManager::removeOneNotification(const QString & updateId)

函数说明:

此方法可以删除指定的通知。

输入值:

updateId: 通知ID。

返回值:

成功执行返回 true; 执行失败返回 false。

A.7.7.4 QString CNotificationManager::sendNotification(const CAbstractNotification & notification)

函数说明:

发送一条通知

参数说明:

输入值: notification: 通知。

返回值: 如果成功, 返回 notification 的更新ID; 失败返回空字符串。

A.7.8 CProgressNotification

CProgressNotification为进度通知类。提供发送给系统进度通知的功能, 进度通知有别于普通通知, 会在状态栏通知面板中显示一个进度条。

A.7.8.1 bool CProgressNotification::setDisplayTime(const QDateTime & time)

函数说明:

设置成指定的显示时间。

参数说明:

输入值: time: 时间。

返回值: 成功执行返回 true; 执行失败返回 false。

注意事项:

默认情况下, 状态栏显示的每个消息中, 都会显示一个消息发出的时间。这个时间默认使用通知发出时的系统时间。在某些特殊情况下需要指定显示的发送时间的, 可以通过这个方法设置一个自定义的时间。这个时间将替代系统时间, 显示在状态栏通知上。

A.7.8.2 bool CProgressNotification::setProgress(int value, int maximum = 100)

函数说明:

设置进度通知的值value和最大值maximum。默认最大值是100。

参数说明:

输入值: 如果是文件传输等操作, 也可以将文件大小作为最大值, 处理字节数作为进度值。

返回值: 成功执行返回 true; 执行失败返回 false。

A.7.8.3 bool CProgressNotification::setTitle(const QString & title)

函数说明:

设置标题。

输入值:

title: 标题。

返回值: 成功执行返回 true; 执行失败返回 false。

A.7.9 CPushManager

CPushManager 是提供给第三方应用程序使用的推送相关接口类。

A.7.9.1 bool CPushManager::regist(const QString & bundleId)

函数说明:

向服务器注册应用程序。

参数说明:

输入值: bundleId: 应用程序身份标识字符串。

返回值: 成功执行返回 true; 执行失败返回 false。

A.7.9.2 bool CPushManager::unregist(const QString & token)

函数说明:

向服务器注销应用程序。

参数说明:

输入值: token: 服务器返回的令牌。

返回值: 成功执行返回 true; 执行失败返回 false。

A.7.10 CToastManager

CToastManager为toast提示框操作类。

A.7.10.1 bool CToastManager::notify(const QString & content)

函数说明:

显示Toast。

参数说明:

输入值: content: Toast内容。

返回值: 成功执行返回 true; 执行失败返回 false。

A.8 MDM 管理

A.8.1 CMDMOVManager

类说明:

CMDMOVManager提供了管理、连接OpenVPN客户端的相关接口支持配置OpenVPN服务器的IP、传输协议、端口、根证书、客户端证书、是否压缩。需要具有移动设备管理权限才可使用。

A.8.1.1 CMDMOVManager::OVStatus CMDMOVManager::Status()

函数说明:

获取OpenVPN的连接状态。

参数说明:

输入值: 无
 返回值:
 OVConnected OpenVPN已连接
 OVConnecting OpenVPN正在连接
 OVConnectsFailed OpenVPN连接失败

A.8.1.2 int CMDMOVManager::disconnect()

函数说明:

断开OpenVPN连接。

参数说明:

输入值: 无
 返回值: 断开成功返回 0, 失败返回非0。

A.8.1.3 QString CMDMOVManager::getCSR()

函数说明:

获取证书签名请求(CSR)

参数说明:

输入值: 无
 返回值: 成功返回一个证书请求文件内容, 失败返回空字符串。

A.8.1.4 int CMDMOVManager::remove()

函数说明:

删除OpenVPN。

参数说明:

输入值: 无
 返回值: 删除成功返回 0, 失败返回非

注意事项:

调用此api删除时如果该OpenVPN是连接状态会先断开后删除配置。

A.8.1.5 bool CMDMOVManager::setCert(const QString & cert, CMDMOVManager::CertType type)

函数说明:

配置证书。

参数说明:

输入值:
 cert参数为表示证书内容。
 type参数为表示证书类型返回值: 删除成功返回 0, 失败返回非。

返回值:
 设置成功返回 true, 设置失败返回 false

注意事项:

支持设置根证书, 客户端p10/p12证书。

A.8.1.6 int CMDMOVManager::setConfig(QMap argMap)

函数说明:

配置OpenVPN的参数。

参数说明:

输入值: argMap参数为要设置的OpenVPN参数。

返回值: 返回值: 成功返回 0, 否则返回非0

QMap key 说明:

值	描述
Host	OpenVPN 服务器 ip 地址(合法 ip 地址)
OpenVPN.Proto	传输层协议(tcp/udp)
OpenVPN.Port	端口(0-65535)
OpenVPN.ComplZO	是否开启压缩 (disable/enable)

A. 8.1.7 int CMDMOVManager::updateConfig(QMap argMap)**函数说明:**

更新OpenVPN的参数。

参数说明:

输入值: argMap参数为要设置的OpenVPN参数。

返回值: 返回值: 成功返回 0, 否则返回非0

注意事项:

更新只是在原有OpenVPN基础上创建一个新的OpenVPN, 待系统重新启动后才会使用更新后的OpenVPN, 旧的都删除。

A. 8.2 CMDMPermissionManager

CMDMPermissionManager 类提供了关于MDM管理的敏感权限的接口。此类需要具有MDM权限应用使用。

A. 8.2.1 bool CMDMPermissionManager::setAppPermissionbyPriority(QString sopid, QString perm, int priority)**函数说明:**

MDM配置应用敏感权限。

参数说明:

输入值:

sopid参数为应用程序SOP包标识。

perm参数为敏感权限的名称。

priority:

值	描述
0	非强制
1	强制 on(MDM 模式)
2	强制 off (MDM 模式)

返回值:

成功执行返回 true; 执行失败返回 false。

A. 8.2.2 bool CMDMPermissionManager::setGlobalPermissionMode(QString perm, int mode)**函数说明:**

设置全局的敏感权限。

参数说明:

输入值:

perm参数为敏感权限名称。

mode:

值	描述
0	表示忽略
1	表示无权限
2	表示无权限

返回值: 成功为true, 失败为false。

A.8.3 CMDMCallLogManager

CMDMCallLogManager为MDM提供操作通话记录的接口, 派生自CCallLogManager。

A.8.3.1 bool CMDMCallLogManager::clearAllCallLog()

函数说明:

删除所有的通话记录。

参数说明:

输入值: 无

返回值: 成功为true, 失败为false。

A.8.4 CMDMMessageManager

CMDMMessageManager提供MDM的短信接口。

A.8.4.1 bool CMDMMessageManager::clearAllMessages()

函数说明:

删除所有短信数据。

参数说明:

输入值: 无

返回值: 成功为true, 失败为false。

A.8.5 CMDMWindowManager

CMDMWindowManager类用来完成MDM相关的功能

A.8.5.1 QString CMDMWindowManager::captureScreen()

函数说明:

截取当前屏幕

参数说明:

输入值: 无

返回值: 截取后图片的绝对路径。

注意事项:

需要申请“OS.permission.CAPTURE_SCREEN”敏感权限才可以调用该接口。

A.8.6 CMDMPushManager

A.8.6.1 bool CMDMPushManager::loginStatus()

函数说明:

获取登录状态。

参数说明:

输入值: 无

返回值: 成功执行返回 true; 执行失败返回 false。

A.9 其他

A.9.1 CConfManager

CConfManager类提供了关于第三方应用数据中心的查询、设置接口。

A.9.1.1 bool CConfManager::getValue(const QString & key, int & value)

函数说明:

获取指定key存储的int类型value值。

参数说明:

输入值:

key: 数据键。

value: 获取的对应数据值。

返回值:

获取成功返回true; 否则返回false。

A.9.1.2 bool CConfManager::getValue(const QString & key, double & value)

函数说明:

获取指定key存储的double类型值。

参数说明:

输入值:

key: 数据键。

value: 获取的对应数据值。

返回值:

获取成功返回true; 否则返回false。

A.9.2 CQRCode

CQRCode提供生成二维码, 解析二维码相关接口。

A.9.2.1 bool CQRCode::decodeQR(const QString & imgPath, QString & output, int & type)

函数说明:

解析二维码, 将指定二维码图片解码为输出内容。

参数说明:

输入值:

imgPath: 二维码图片路径。

output: 输出内容。

type: 二维码类型。

返回值:

解码成功返回true；否则返回false。

A.9.2.2 bool CQRCode::encodeQR(const QString & input, const QString & output, bool micro = false)

函数说明：

生成二维码，将输入内容生成二维码，并将二维码图片保存到指定输出路径。

参数说明：

输入值：

input: 输入内容。

output: 二维码图片输出路径。

micro: 生成二维码类型，true为微型，false为普通，默认为false。

返回值：

加密成功返回true；否则返回false。