Constants

Callback Data Types

uex.cText = 0;

uex.cJSON = 1;

uex.cInt = 2;

Device Info Types

|  |  |  |  |
| --- | --- | --- | --- |
| name | value | 说明 | 返回值 |
| uex.cCPU | 0 | 此key对应的Value是一个描述CPU频率的字符串，eg：“1024MHZ”。IOS平台获取不到时，返回“0”。 | {cpu:xxx} |
| uex.cOS | 1 | 此key对应的Value是一个描述系统版本的字符串, eg：“Android2.3.4” | {os:xxx} |
| uex.cManufacturer | 2 | 此key对应的Value是一个标书设备制造商的字符串，eg:“htc” | {manufacturer:xxx} |
| uex.cKeyboard | 3 | 此key对应的Value是一个代表是否支持键盘的字符串0（不支持）或1（支持), uex.jvDeviceSupport= 1;  uex.jvDeviceUnsupport= 0; | {keyboard:1} |
| uex.cBlueTooth | 4 | 代表是否支持蓝牙的字符串0（不支持）或1（支持）,uex.jvDeviceSupport= 1;uex.jvDeviceUnsupport= 0;当设备有蓝牙功能时，即使蓝牙关闭，返回信息仍然是支持蓝牙，即value值为字符串1。在iOS上的蓝牙功能只支持同一应用间使用，和普遍人们理解的不同，视为不支持。 | {blueTooth:1} |
| uex.cWIFI | 5 | 此Key的Value是一个代表是否支持WIFI的字符串0（不支持）或1（支持），uex.jvDeviceSupport= 1;uex.jvDeviceUnsupport= 0;当设备有wifi功能时，即使wifi关闭，返回信息仍然是支持wifi，即value值为字符串1。 | {wifi:1} |
| uex.cCamera | 6 | 此Key的Value是一个代表是否支持摄像头的字符串0（不支持）或1（支持）,uex.jvDeviceSupport= 1;uex.jvDeviceUnsupport= 0; | {camera:1} |
| uex.cGPS | 7 | 此Key的Value是一个代表是否支持GPS的字符串0（不支持）或1（支持）,uex.jvDeviceSupport= 1;uex.jvDeviceUnsupport= 0;当设备有gps功能时，即使gps关闭，返回信息仍然是支持gps，即值为字符串1。 | {gps:1} |
| uex.cGPRS | 8 | 此Key的Value是一个代表当前移动网络数据连接是否可用（不含WIFI）的字符串0（不可用）或1（可用）uex.jvDeviceSupport= 1;uex.jvDeviceUnsupport= 0; | {gprs:1} |
| uex.cTouch | 9 | 此Key的Value是一个代表设备是否支持触屏的字符串0（不支持）或1 （支持）uex.jvDeviceSupport= 1;uex.jvDeviceUnsupport= 0; | {touch:1} |
| uex.cIMEI | 10 | 此Key的Value是一个代表此设备IMEI（国际移动设备唯一标识码）号的15位字符串，eg：“356357046156042”。在iOS上，获得是openUDID, 开源的一个UDID替代方案，原理是利用应用间的剪贴板共享和本地一些必要的缓存信息，让多个应用间共享一个UUID。OpenUDID在官方废弃UDID接口之后，受到广泛的欢迎！可以说是现在大多数应用的UDID替代方法。OpenUDID在刷机、还原设备后就会产生新的UDID，事实上，由于剪贴板的特殊性，如果所有使用了OpenUDID的应用被全部卸载之后，再次安装的应用取到的OpenUDID将会是一个全新的值！iOS7中，不同组的应用（即不同厂商）的应用之间不再能共享剪贴板间的数据！同组（即同一厂商）应用的定义为：Info.plist中关于软件唯一标示符的字段CFBundleIdentifier中的前两段标识符（例如com.mycompany）相同。固在iOS7中，OpenUDID也将慢慢失去它的意义。 | {imei:xxx} |
| uex.cDeviceToken | 11 | 此Key的value是推送服务器需要的一个代表此设备的唯一令牌的字符串。eg：“98d264a3 77689b33 6f1215e6 264ab0c5 55f45b4a ab61e6ff f667883a ef829ccb”,没有时返回空字符串。Android的deviceToken是softToken。 | {deviceToken:xxx} |
| uex.cDeviceType | 12 | 此key的value是一个设备类型，用来判断当前的设备是phone或者pad。uex.jvDeviceTypeIPhone = 0;uex.jvDeviceTypeIPad = 1;uex.jvDeviceTypeIPodTouch =2; | {deviceType:0} |
| uex.cConnectStatus | 13 | 此key的value表示当前联网的方式(uex.jvConnectStatusUnreachability（无网络连接）uex.jvConnectStatusWifi（wifi连接） ,uex.jvConnectStatus3G（3g连接）,uex.jvConnectStatusGPRS（gprs连接）),uex. jvConnectStatus4G（4g网络） | {connectStatus:-1} |
| uex.cRestDiskSize | 14 | 此key的value表示当前设备剩余的磁盘空间大小的字符串，eg：“12345678”单位：字节 | {restDiskSize:10000} |
| uex.cMobileOperatorName | 15 | 此key的value表示当前移动网络运营商的名称，比如”中国联通”,如果获取不到返回空字符串 | {mobileOperatorName:xxx} |
| uex.cMacAddress | 16 | 此key的value表示表示当前设备的WIFI mac地址 ，可作为设备的唯一标识，IMEI可能在某些不具备移动通讯的android平板或MP4上获取不到，但是android系统设备一般都会具有WIFI功能，所以mac地址作为设备唯一标识比IMEI更可靠 | {macAddress:xxx} |
| uex.cModel | 17 | 此key的value表示当前设备的型号名称，如“Galaxy Nexus” | {model:xxx} |
| uex.cResolutionRatio | 18 | 此key的value表示获得屏幕的分辨率回调，例如：iphone4的分辨率为640\*960(格式固定，宽\*高) | {resolutionRatio:宽\*高} |
| uex.cSimSerialNumber | 19 | 此key的value表示当前设备的sim卡的序列号。只支持Android | {simSerialNumbers:序列号} |
| uex.cUUID | 20 | iPhone返回UUID，Android返回空。 此key的value表示当前设备的uuid。在iOS5将UDID标为废弃之后，官方提供的替代方案。即使用CFUUIDCreate生成一个UUID，并将之保存在NSUserDefault中，用它作为设备标识符。在iOS6之后，苹果更推出NSUUID来替代CFUUIDCreate，但本质是一样的。UUID每次都会生成一个新的字符串，也就是说应用被卸载之后，就会被认为是一个新的设备，更不用提刷机、还原设备了。故基本无人采用UUID的方案。 | {uuid:xxx} |

File Open Modes：

uex.cRead = 1;

uex.cWrite = 2;

uex.cNew = 4;

uex.cReader =8

File Write Modes

uex.cAppend = 1;

Callback Int Values：

uex.cTrue = 1;

uex.cFalse = 0;

uex.cSuccess = 0;

uex.cFailed = 1;

uex.cExist = 0;

uex.cNotExist = 1;

uex.cPaySuccess = 0;

uex.cPaying = 1;

uex.cPayFailed = 2;

uex.cPayPlugInError = 3;

uex.cOrientationPortraitUp = 1;

uex.cOrientationLandscapeLeft = 2;

uex.cOrientationPortraitDown = 4;

uex.cOrientationLandscapeRight = 8;

uex.cFile = 0;

uex.cFolder =1;

Sensor Type：

uex.cAccelerometer = 1;

uex.cOrientation = 2;

uex.cMagnetic = 3;

uex.cTemperature = 4;

uex.cPressure = 5;

uex.cLight = 6;

Sensor Rate

uex.cRateFastest = 0

uex.cRateGame = 1

uex.cRateUI = 2

uex.cRateNormal = 3

Window Types

uex.cWindowTypeNormal = 0

uex.cWindowTypeTop = 1

uex.cWindowTypeBottom = 2

Window State

uex.cWindowStateForeGround =0

uex.cWindowStateBackGround =1

Window Source Types

uex.cWindowSrcTypeUrl = 0

uex.cWindowSrcTypeData = 1

uex.cWindowSrcTypeUrlAndData = 2

Window Flags

uex.cWindowFlagNone = 0

//标记被open的window为普通window。

uex.cWindowFlagOAuth = 1

//标记被open的window为专用于OAuth验证的window。

uex.cWindowFlagObfuscation = 2

//标记被open的window要加载的网页为加密的网页。

uex.cWiondowFlagReload =4

//标记被open的window无论是否已存在都将强行刷新页面。

uex.cWiondowFlagDisableCrossdomain =8

//标记被open的window当中的任何url都将调用系统浏览打开。

uex.cWiondowFlagOpaque =16

//标记被open的window当中的view为不透明的。

uex.cWiondowFlagHidden = 32

//标记被open的window为隐藏的。隐藏的window不会显示到屏幕上，只存在于后台。隐藏的window不可以再调用open window。

uex.cWiondowFlagPreOpen = 64

//标记被open的window将有一个或n个popOver的预加载，且只有此window中的这些popOver都加载完毕后，此window才会显示到屏幕上。

uex.cWiondowFlagEnableScale = 128

//标记被open的window或popOver将支持手势缩放。

uex.cWindowFlagPreNotHidden = 256

//标记被open的window的上一个window在屏幕上不隐藏。如果当前window可能会调用uexWindow. setWindowFrame接口，则在open当前窗口时需传入此flag。

该flag设置原因：Android平台系统的渲染机制不同于IOS，不在屏幕上的View必须要设置隐藏，否则只要存在于屏幕上的View，系统都会进行渲染，将会消耗非常大的CPU或者GPU资源，导致渲染变慢，卡顿。而IOS上则不用隐藏，IOS系统只会渲染屏幕最上层的View，位于底层的View，虽然也在屏幕上，但不会被渲染。

当调用uexWindow. setWindowFrame将最上层的View移动位置后，IOS平台中，底下的View会露出来，此时系统会对其进行渲染；而Android平台的底层View此时是处于隐藏状态的，无法渲染，所以需要增加此flag，标志当前window的上一个window不隐藏。

uex.cWindowFlagWebApp = 512

//标记被open的浮动窗口将用于打开WebApp。此浮动窗口将不进行默认字体的设置，缩放比例的设置等，WebApp的排版适配交由系统处理。注意：此flag仅用于open浮动窗口。

Window Animi ID

uex.cWindowAnimiIDNone = 0

//无动画

uex.cWindowAnimiIDLeftToRight = 1

//由左往右推入

uex.cWindowAnimiIDRightToLeft = 2

//由右往左推入

uex.cWindowAnimiIDUpToDown = 3

//由上往下推入

uex.cWindowAnimiIDDownToUp = 4

//由下往上推入

uex.cWindowAnimiIDFadeOutFadeIn = 5

//淡入淡出

uex.cWindowAnimiIDLeftFlip = 6

//左翻页（android暂不支持）

uex.cWindowAnimiIDRigthFlip = 7

//右翻页（android暂不支持）

uex.cWindowAnimiIDRipple = 8

//水波纹（android暂不支持）

uex.cWindowAnimiIDLeftToRightMoveIn = 9

//由左往右切入

uex.cWindowAnimiIDRightToLeftMoveIn = 10

//由右往左切入

uex.cWindowAnimiIDTopToBottomMoveIn = 11

//由上往下切入

uex.cWindowAnimiIDBottomToTopMoveIn = 12

//由下往上切入

**//以下为close专用，与9，10，11，12对应：**

uex.cWindowAnimiIDLeftToRightReveal= 13

//由左往右切出，与10对应

uex.cWindowAnimiIDRightToLeftReveal= 14

//由右往左切出，与9对应

uex.cWindowAnimiIDTopToBottomReveal= 15

//由上往下切出，与12对应

uex.cWindowAnimiIDBottomToTopReveal= 16

//由下往上切出，与11对应

Window AnimCurveType

uex.cViewAnimaCurveNone = 0;

//无运动曲线,做线性平滑运动

uex.cViewAnimaCurveEaseInOut = 1;

//先加速后减速运动

uex.cViewAnimCurveEaseIn = 2;

//加速运动

uex.cViewAnimCurveEaseOut = 3;

//减速运动

uex.cViewAnimCurveLinear = 4;

//动画线性平滑运动

Window Toast Location

uex.cToastLocationLeftTop = 1

uex.cToastLocationTop = 2

uex.cToastLocationRightTop = 3

uex.cToastLocationLeft = 4

uex.cToastLocationMiddle = 5

uex.cToastLocationRight = 6

uex.cToastLocationBottomLeft = 7

uex.cToastLocationBottom = 8

uex.cToastLocationRightBottom = 9

Window BounceViewTypes

uex.cBounceViewTypeTop = 0

uex.cBounceViewTypeBottom = 1

Window BounceView State

uex.cBounceViewStatePullToReload = 0

uex.cBounceViewStateReleaseToReload = 1

uex.cBounceViewStateLoading = 2

Window BounceView Parm Key

uex.cBounceParmKeyImagePath = “imagePath”

//下拉状态小图标的路径，只支持res://格式。

uex.cBounceParmKeyTextColor = “textColor”

//展示下拉状态文字的颜色。支持的格式为“#”3位，或者7位，以及,rgba()格式。

uex.cBounceParmKeyLevelText = “levelText”

//显示的二级文字，如：“上次更新时间：xxxxx”。

uex.cBounceParmKeyPullToReloadText= “pullToReloadText”

//开始拖动直到超过刷新临界线之前显示的文字，如：“拖动刷新”。

uex.cBounceParmKeyreleaseToReloadText = “releaseToReloadText”

//拖动超过刷新临界线后显示的文字，如：“释放刷新”。

uex.cBounceParmKeyLoadingText =“loadingText”

//拖动超过刷新临界线并且释放拖动，进入刷新状态时显示的文字，如：“加载中，请稍等”。

uex.cBounceParmKeyLoadingImagePath = “loadingImagePath”

//拖动超过刷新临界线并且释放拖动，进入刷新状态时显示的loading状态小图标，默认为系统的小圈圈。此字段在东航项目中起作用。

UpLoadStatus

Uex.cUpLoading =0

Uex.cFinishUpLoad =1

Uex.cUpLoadError =2

DownLoadStatus

Uex.cDownLoading =0

Uex.cFinishDownLoad =1

Uex.cDownLoadError =2

Download mode

uex.cNot Breakpoint =0

uex. cBreakpoint =1

XmlHttpRequest Method

uex.cXmlHttpRequestMethodGet = get

uex.cXmlHttpRequestMethodPost = post

XmlHttpRequestStatus

uex.cXmlHttpRequestStatusReceive =0

uex.cXmlHttpRequestStatusFinish = 1

uex.cXmlHttpRequestStatusError = -1

XmlHttpRequestDataType

uex.cXmlHttpRequestPostText =0

uex.cXmlHttpRequestPostBinary = 1

Platform Key Code

uex.cKeyCodeBack = 0

uex.cKeyCodeMenu = 1

Platform Info

uex.cPlatformIOS = 0;

uex.cPlatformAndroid = 1;

uex.cPlatformChrome = 2;

BaiduDataAnalysis Strategy

BaiduMobStatLogStrategyAppLaunch = 0, //每次程序启动

BaiduMobStatLogStrategyDay = 1, //每天的程序第一次进入前台

BaiduMobStatLogStrategyCustom = 2, //根据开发者设定的时间间隔接口发送

MapStateCode

uex.cMapStateStart = 0, //开始移动

uex.cMapStateMove = 1, //正在移动

uex.cMapStateStop = 2, //停止移动

Keyboard Types

uex.StandardKB = 0;

uex.NumberKB = 1;

uex.EmailKB= 2;

uex.URLKB = 3;

uex.PasswordKB= 4;

Path Types

|  |  |  |
| --- | --- | --- |
| 协议头 | Android对应路径  (其中"/sdcard/"等同于"/storage/emulated/0/") | iOS对应路径 |
| res:// | widget/wgtRes/ | widget/wgtRes |
| wgt:// | /storage/emulated/0/widgetone/apps/*xxx(widgetAppId)/* |  |
| wgts:// | /storage/emulated/0/widgetone/widgets/ |  |
| file:///sdcard/ | /storage/emulated/0/ |  |