
智能APOS离线动态库接口文档

产品名称:

产品版本:

机密等级:

批准:

审核:

拟制:

福建联迪商用设备有限公司

修订说明

日期	修订版本	修改描述	作者
2017-03-28	V1.0	创建	苏初旭
2017-08-17	V2.0	修改2.5文件更新接口的传递参数	苏初旭

目录

1 概述	4
1.1 目的	4
1.2 适用平台	4
1.3 程序提供形式.....	4
2 接口说明.....	4
2.1 获取回调信息接口.....	4
2.2 清除回调信息接口.....	5
2.3 获取解析阶段信息接口.....	5
2.4 获取安装进度信息接口.....	5
2.5 文件更新接口.....	6
2.5.1 文件更新接口 1.....	6
2.5.2 文件更新接口 2.....	6
2.6 获取版本号信息接口.....	7
3 调用示例.....	7

1 概述

1.1 目的

由于离线动态库支持联迪 pkg、uns 等文件的升级，编写《智能 APOS 离线动态库接口文档》这篇文档，主要是供给调用离线动态库进行文件升级的开发者参考，便于能够快速的调用离线动态库相关接口。

1.2 适用平台

主要适用于联迪智能 APOS 平台。

1.3 程序提供形式

提供 FileUpgrade.jar 包和 libNativeDown.so 库，FileUpgrade.jar 包和 libNativeDown.so 库需要导入到开发应用中。

2 接口说明

2.1 获取回调信息接口

接口	String getStrBack ()	
功能	获取回调信息	
参数	入参	无
	出参	无
返回值	返回离线动态库的回调信息	
备注	离线动态库的回调信息格式：阶段信息@进度	

2.2 清除回调信息接口

接口	void clearStrBack ()	
功能	清除离线动态库的回调信息	
参数	入参	无
	出参	无
返回值	无	
备注		

2.3 获取解析阶段信息接口

接口	int getPhase ()	
功能	获取解析阶段信息	
参数	入参	无
	出参	无
返回值	返回解析阶段信息	
备注		

2.4 获取安装进度信息接口

接口	int getProgress ()	
功能	获取安装进度信息	
参数	入参	无
	出参	无
返回值	返回文件安装进度信息	
备注		

2.5 文件更新接口

2.5.1 文件更新接口 1

接口	upgrade(Context mContext,String filelist,byte[] RetInfo)	
功能	文件更新	
参数	入参	mContext: 上下文实例 filelist: 文件路径列表
	出参	RetInfo: 返回安装结果信息
返回值	返回文件更新结果	
备注	<p>1、入参的filelist文件路径列表如果传入多个文件路径时，文件路径之间使用分号隔开，且安装的文件路径需要传入绝对路径，格式举例："/sdcard/1.uns;/sdcard/2.uns;/sdcard/1.pkg"</p> <p>2、当且仅当upgrade返回值为0，且解析阶段信息为3时，才表示文件安装成功。</p>	

2.5.2 文件更新接口 2

接口	String upgradeLV2(Context mContext,int iDebug,String filelist,byte[] RetInfo)	
功能	文件更新	
参数	入参	mContext: 上下文实例 iDebug: 串口是否输出日志信息 filelist: 文件路径列表
	出参	RetInfo: 返回安装结果信息
返回值	返回文件更新结果	
备注	<p>1、入参的filelist文件路径列表如果传入多个文件路径时，文件路径之间使用分号隔开，且安装的文件路径需要传入绝对路径，格式举例："/sdcard/1.uns;/sdcard/2.uns;/sdcard/1.pkg"</p>	

	2、当且仅当upgradeLV2返回值为0，且解析阶段信息为3时，才表示文件安装成功。
--	---

2.6 获取版本号信息接口

接口	String getVersion()	
功能	获取版本号信息	
参数	入参	无
	出参	无
返回值	返回离线动态库和jar包的版本信息	
备注		

3 调用示例

以下示例是使用离线动态库相关接口，安装智能POS终端sdcard路径下的

1.uns、2.uns和1.pkg文件。

```
package com.download.test;

import java.io.UnsupportedEncodingException;

import android.app.Activity;

import android.os.Bundle;

import android.os.Handler;

import android.util.Log;

import android.view.View;

import android.widget.Button;

import android.widget.TextView;

import com.landi.upgrade.jni.JNIInvoker;

public class DownloadActivity extends Activity {
```

```
private static final String TAG = DownloadActivity.class.getSimpleName();

private TextView tvCallBack, tvPhase, tvProgress;

private Button btnUpdate;

private int TIME = 100;

private JNIInvoker jniInvoke = new JNIInvoker();
```

```
@Override
```

```
public void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.main);

    tvCallBack = (TextView) findViewById(R.id.tv_callback);

    tvPhase = (TextView) findViewById(R.id.tv_phase);

    tvProgress = (TextView) findViewById(R.id.tv_progress);

    btnUpdate = (Button) findViewById(R.id.btn_update);

    btnUpdate.setOnClickListener(new OnClickListener());

}
```

```
// 使用handler postDelayed定时器更UI
```

```
Handler handler = new Handler();
```

```
Runnable runnable = new Runnable() {
```

```
    @Override
```

```
    public void run() {
```

```
        try {
```

```
            // 更新UI
```

```
            tvCallBack.setText(jniInvoke.getStrBack());
```

```
            tvPhase.setText(String.valueOf(jniInvoke.getPhase()));
```

```
            tvProgress.setText(String.valueOf(jniInvoke.getProgress()));
```

```
            // 重新设置UI更新定时器
```

```
            handler.postDelayed(this, TIME);
```

```
        } catch (Exception e) {
```



```
// TODO Auto-generated catch block
e.printStackTrace();
}
}
};

private class OnUpdateListener implements View.OnClickListener {
    @Override
    public void onClick(View v) {
        // 设置UI更新定时器
        handler.postDelayed(runnable, TIME);

        // 开启新的线程用于执行文件更新，防止影响主线程的UI更新
        new Thread(){
            @Override
            public void run() {
                // TODO Auto-generated method stub
                super.run();
                byte[] retInfo = new byte[1024];
                String ret = "";
                String strVersion = jniInvoke.getVersion();
                Log.d(TAG, "Current Version:" + strVersion);

                // 文件更新
                ret = jniInvoke.upgradeLV2(DownloadActivity.this, 1,
                "/sdcard/1.uns;/sdcard/2.uns;/sdcard/1.pkg", retInfo);

                Log.d(TAG, "upgradeLV2 return: " + ret);

                if (retInfo != null) {
                    try {
                        String result = new String(retInfo,
```

```
"gb2312").trim();

        Log.d(TAG, "upgradeLV2 return info : " + result);
    } catch (UnsupportedEncodingException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

}

}.start();
}

}

}
```