```
    &script Variant))
387         gLogI.log(instance, kError_ANPLogType, " ------ %p Unable
            to eval the JS.", instance);
388
389     if (scriptVariant.type == NPVariantType_Int32) {
390         if (scriptVariant.value.intValue != 1234)
391             gLogI.log(instance, kError_ANPLogType, " ------ %p
              Invalid Value for JS Return: %d,1234", instance, scriptVariant.
              value.intValue);
392     } else {
393         gLogI.log(instance, kError_ANPLogType, " ------ %p Invalid
            Variant type for JS Return: %d,%d", instance, scriptVariant.
            type, NPVariant Type_Int32);
394     }
395
396     // free the memory allocated within the browser
397     browser->memfree(stringMem);
398 }
```

BackgroundPlugin.h 文件代码：

```
/*
 * Copyright 2008, The Android Open Source Project
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *  * Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 *  * Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS ``AS IS'' AND ANY
 * EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
 * PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL APPLE COMPUTER, INC. OR
 * CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
 * EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
 * PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
 * PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY
 * OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
 * OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */
```

```
001 #include "PluginObject.h"
002
003 #ifndef backgroundPlugin__DEFINED
004 #define backgroundPlugin__DEFINED
005
006 class BackgroundPlugin : public SurfaceSubPlugin {
007 public:
008     BackgroundPlugin(NPP inst);
009     virtual ~BackgroundPlugin();
010     virtual bool supportsDrawingModel(ANPDrawingModel);
011     virtual int16 handleEvent(const ANPEvent* evt);
012     virtual void surfaceCreated(JNIEnv* env, jobject surface);
013     virtual void surfaceChanged(int format, int width, int height);
014     virtual void surfaceDestroyed();
015     virtual bool isFixedSurface();
016
017     // Timer Testing Variables
018     uint32_t mStartTime;
019     uint32_t mPrevTime;
020     int      mTimerRepeatCount;
021     int      mTimerLatencyCount;
022     int      mTimerLatencyCurrentCount;
023
024     // Bitmap Transparency Variables
025     bool mFinishedStageOne;   // check default & set transparent
026     bool mFinishedStageTwo;   // check transparent & set opaque
027     bool mFinishedStageThree; // check opaque
028
029 private:
030     void drawPlugin(int surfaceWidth, int surfaceHeight);
031
032     jobject    m_surface;
033     JavaVM*    m_vm;
034
035     void test_logging();
036     void test_timers();
037     void test_bitmaps();
038     void test_bitmap_transparency(const ANPEvent* evt);
039     void test_domAccess();
040     void test_javascript();
041
042 };
043
044 #endif // backgroundPlugin__DEFINED
```

11.3.3 BrowserPlugin 中的 ANPInterface

为了弥补 NPAPI 在 Android 上的不足，Google 在 Android 的浏览器上实现了 ANPInterface，就是一系列的操作接口（函数），提供了一些 NPAPI 没有的东西。插件可以在初始化的时候获取这些 ANPXXXInterface，并在运行过程中使用。

其实，ANPInterface 提供的接口大多来自 webkit 的底层库（external/webkit/WebKit/ android/ Plugins）。

BrowserPlugin 中的 ANPInterface 列表如下：

```
ANPBitmapInterfaceV0      gBitmapI;
ANPCanvasInterfaceV0      gCanvasI;
ANPLogInterfaceV0         gLogI;
ANPPaintInterfaceV0       gPaintI;
ANPPathInterfaceV0        gPathI;
ANPSurfaceInterfaceV0     gSurfaceI;
ANPSystemInterfaceV0      gSystemI;
ANPTypefaceInterfaceV0    gTypefaceI;
ANPWindowInterfaceV0      gWindowI;
```

11.3.4 BrowserPlugin 的工作流程

BrowserPlugin 的工作流程如图 11-3 所示，具体描述如下。

浏览器解析页面时，遇到插件的 MIME 类型，就去检查插件注册表，如果有，就加载插件。

在插件加载之后，插件先会进行 API 映射，即把各种调配资源的 API 映射到 NPNetscapeFuncs 的结构体指针上，然后作为输入参数调用 NP_Initialize()，NP_Initialize()只被调用一次。

初始化 API 返回成功后（一个 NPPetscapeFuncs 结构体指针），插件入口 API 将被调用，它允许浏览器不必常规地来调用插件端的 APIs。这样入口 API 返回成功后，浏览器的 NPPetscapeFuncs 结构体将被插件端有效的 APIs 指针填充（根据适当的内部流程），并将立即按需被调用。

调用 NPP_New()，实例化插件，如上面的实例 0x312a10 和 0x420f18；每个实例都会被分配给一个数据块，每个实例根据插件的定义填充参数。

调用 NPP_SetWindow()，显示插件。

如果在插件上单击鼠标之类的，就会调用 NPP_HandleEvent()。

关闭此页面，会先调用 NPP_SetWindow()，然后调用 NPP_Destroy()，释放实

例的资源。

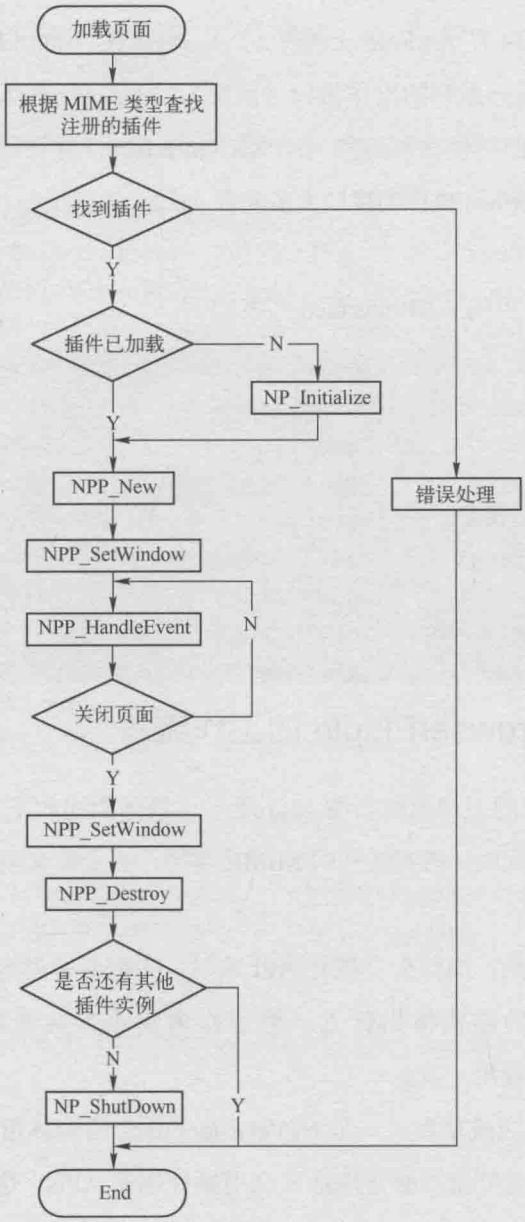

图 11-3　BrowserPlugin 工作流程图

全部实例被 Destroy 后，调用 NP_ShutDown()，释放全局资源。

11.4　编译和运行浏览器插件

修改 jni/main.cpp 文件之后编译（主要是增加 LOGCAT 调试信息），以方便后面分析插件加载流程。

进入源码根目录下，运行 make SampleBrowserPlugin。

运行“adb install [apk_file]”，把编译好的插件 apk 安装到设备或模拟器中。

安装成功后，可以通过“Settings→Applications→Manage applications”管理插件。

用包含以下内容的 HTML 页面测试浏览器插件：

```
<object type="application/x-testbrowserPlugin" height=50 width=250>
    <param name="DrawingModel" value="Surface" />
    <param name="PluginType" value="Background" />
</object>
```

用浏览器打开测试网页，将会打印类似以下 log：

```
D/Plugin  (  366): *** NP_Initialize ***
D/Plugin  (  366): *** 0x420f18 START NPP_New ***
D/Plugin  (  366): ------ 0x420f18 DrawingModel is 1
D/Plugin  (  366): Application data dir is /data/data/com.android.
          browser/ app_plugins
E/Plugin  (  366):  ------ 0x420f18 Testing Log Error
W/Plugin  (  366):  ------ 0x420f18 Testing Log Warning
D/Plugin  (  366):  ------ 0x420f18 Testing Log Debug
D/Plugin  (  366): pixel format [0] unknown has no packing
D/Plugin  (  366): pixel format [1] 8888 has packing ARGB [24 8] [0 8]
          [8 8] [16 8]
D/Plugin  (  366): pixel format [2] 565 has packing ARGB [0 0] [11 5]
          [5 6] [0 5]
D/Plugin  (  366):  ------ 0x420f18 Testing DOM Access
D/Plugin  (  366):  ------ 0x420f18 Testing JavaScript Access
E/Plugin  (  366):  ------ 0x420f18 Invalid Variant type for JS Return: 4,3
D/Plugin  (  366): ------ 0x420f18 PluginType is 3
D/Plugin  (  366): *** 0x420f18 END NPP_New ***
D/Plugin  (  366): *** 0x312a10 START NPP_New ***
D/Plugin  (  366): ------ 0x312a10 DrawingModel is 1
D/Plugin  (  366): Application data dir is /data/data/com.android.
          browser/ app_plugins
D/Plugin  (  366): ------ 0x312a10 PluginType is 6
D/Plugin  (  366): *** 0x312a10 END NPP_New ***
D/Plugin  (  366): *** 0x420f18 NPP_SetWindow ***
D/dalvikvm(  366): Trying to load lib /data/data/com.android.
          samplePlugin/ lib/libsamplePlugin.so 0x43c2e448
D/dalvikvm(  366): Added shared lib /data/data/com.android.samplePlugin/
lib/libsamplePlugin.so 0x43c2e448
D/Plugin  (  366): *** 0x312a10 NPP_SetWindow ***
```

```
D/Plugin  (  366): -------- repeat timer 5
D/Plugin  (  366): -------- latency test: [1937207155] interval 421
expected 50, total 421 expected -1923890058, drift 1923890479avg 0
D/Plugin  (  366): -------- oneshot timer
D/Plugin  (  366): -------- repeat timer 4
D/Plugin  (  366): -------- latency test: [1937207156] interval 473
expected 50, total 894 expected -1923890008, drift 1923890902 avg 0
E/Plugin  (  366): ----0x312a10 Invalid Surface Dimensions (300,150):
(120,60)
D/Plugin  (  366): -------- repeat timer 3
D/Plugin  (  366): -------- latency test: [1937207157] interval 73
expected 50, total 967 expected -1923889958, drift 1923890925 avg 0
D/Plugin  (  366): -------- repeat timer 2
D/Plugin  (  366): -------- latency test: [1937207158] interval 130
expected 50, total 1097 expected -1923889908, drift 1923891005 avg 0
D/Plugin  (  366): *** 0x420f18 NPP_HandleEvent ***
D/Plugin  (  366):  ------ 0x420f18 the plugin received an onLoad event
D/Plugin  (  366): *** 0x312a10 NPP_HandleEvent ***
D/Plugin  (  366): -------- repeat timer 1
D/Plugin  (  366): -------- latency test: [1937207159] interval 90
expected 50, total 1187 expected -1923889858, drift 1923891045 avg 0
D/Dalvikvm(   54): GC freed 8773 objects / 568952 bytes in 169ms
D/Plugin  (  366): *** 0x312a10 NPP_HandleEvent ***
D/Plugin  (  366): *** 0x312a10 NPP_HandleEvent ***
D/Plugin  (  366): *** 0x312a10 NPP_HandleEvent ***
D/Plugin  (  366): *** 0x420f18 NPP_HandleEvent ***
D/Plugin  (  366): *** 0x420f18 NPP_HandleEvent ***
D/PowerManagerService(   54): setPowerState: mPowerState=3 newState=7
noChangeLights=false
D/PowerManagerService(   54):   oldKeyboardBright=false newKeyboard
Bright=false
D/PowerManagerService(   54):   oldScreenBright=true newScreenBright=
true
D/PowerManagerService(   54):   oldButtonBright=false newButton
Bright= true
D/PowerManagerService(   54):   oldScreenOn=true newScreenOn=true
D/PowerManagerService(   54):  oldBatteryLow=false newBatteryLow=false
W/KeyCharacterMap(  366): No keyboard for id 0
W/KeyCharacterMap(  366): Using default keymap: /system/usr/keychars/
 qwerty.kcm.bin
D/Plugin  (  366): *** 0x420f18 NPP_SetWindow ***
D/Plugin  (  366): *** 0x420f18 NPP_Destroy ***
D/Plugin  (  366): *** 0x312a10 NPP_SetWindow ***
D/Plugin  (  366): *** 0x312a10 NPP_Destroy ***
```

```
D/Plugin  (  366): *** NP_Shutdown ***
```

课后习题

1. 什么是 NPAPI？试描述 NPAPI 的框架结构。
2. Android 浏览器插件开发过程中是如何实现 NPAPI 的？
3. 简述 Android 浏览器插件的工作流程。

参 考 文 献

1. 维基百科：
   http://en.wikipedia.org/wiki/Android_(operating_system)
2. Android 官方文档：
   http://developer.android.com/guide/basics/what-is-android.html
3. Dalvik 虚拟机线识：
   http://hi.baidu.com/carvencao/blog/item/c3672064d7f2f031aa184cf2.html
4. WebKit 内核优点：
   http://bbs.zlsoft.com/home.php?mod=space&uid=18026&do=blog&id=1247
5. Android 系统魅力何在？：
   http://mobile.zol.com.cn/153/1538969.html
6. Android 官方文档，Views Tutorial：
   http://developer.android.com/resources/tutorials/views/index.html
7. Android 菜单：
   http://blog.csdn.net/hellogv/article/details/6168439
8. Android 对话框：
   http://www.cnblogs.com/salam/archive/2010/11/15/1877512.html
9. android-aidl-ipc-rpc-example：
   http://code.google.com/p/android-aidl-ipc-rpc-example/
10. Android ContentProvider：
    http://xuyuanshuaaa.iteye.com/blog/973755
11. 开放源码嵌入式数据库 SQLite 简介：
    http://www.ibm.com/developerworks/cn/opensource/os-sqlite/
12. Android SQLite 简介：
    http://gy890725.iteye.com/blog/782485
13. Android 官方文档 Multimedia and Camera：
    http://developer.android.com/guide/topics/media/index.html
14. Android 官方示例 ApiDemos/Graphics/PathEffects：

http://developer.android.com/resources/samples/ApiDemos/src/com/example/android/apis/graphics/PathEffects.html

15. Android 官方文档 Graphics→OpenGL：
http://developer.android.com/guide/topics/graphics/opengl.html
16. OpenGL ES Tutorial for Android：
http://blog.jayway.com/2009/12/03/opengl-es-tutorial-for-android-part-i/
17. Android 网络编程之 Http 通信：
http://52android.blog.51cto.com/2554429/496621
18. 深入探讨 Android 传感器：
http://www.ibm.com/developerworks/cn/opensource/os-android-sensor/index.html
19. 硬件传感器：
http://dev.10086.cn/cmdn/bbs/thread-41843-1-1.html
20. Android 官方文档 Sensors：
http://developer.android.com/guide/topics/sensors/index.html
21. 构建 Android 平台 Google Map 应用：
https://developers.google.com/maps/documentation/android/
22. Android 官方文档 Location and Maps：
http://developer.android.com/guide/topics/location/index.html
23. Android 浏览器插件开发：
http://blog.csdn.net/qyqzj/article/details/5617220

反侵权盗版声明