# 时代亿信邮件移植交流

2013/7/16

中科院信息工程研究所

# 邮件交流介绍

由于时代亿信没有保密环境计算机，不了解开发环境，移植邮件系统存在一定困难。故对保密环境计算机环境做如下介绍。

保密环境的计算机硬件主要采用国产龙芯处理器3号系列, 保密环境的计算机操作系统采用中标麒麟安全操作系统。注意这里龙芯架构与通常X86架构计算机不同，因此在X86环境下动态链接库、可执行程序在龙芯平台必须重新编译。

中标麒麟系统拥有完善的开发环境：基本涵盖了目前主流的开发发语言及相关的工具、库等。它们包括：C/C++、JAVA、PHP、Python、Perl、Ruby、Qt等。此外，还有大量图形IDE工具、项目管理工具、辅助工具等。中标麒麟桌面办公的office产品有：中标普华、无锡永中、金山WPS等。中标麒麟V6采用的桌面环境为Gnome 2.30.2，Firefox 版本为4.0。

中科院信息工程研究所应用移植组针对邮件移植中存在的数据加解密以及附件的上传/下载问题提供相应支持。其中：1、数据加解密，应用移植组已经完成一种Firefox解决方案的测试，并完成x86环境到龙芯环境的移植，验证了该方案的可行性。2、附件上传/下载，应用移植组在保密计算机环境下测试了Firefox浏览器对其他邮件的支持情况（新浪邮箱，网易邮箱等），并实际测试了一组代码。

# 数据加解密（Firefox插件开发）

## 2.1 linux环境下Firefox插件开发

Firefox插件开发采用跨浏览器的API---NPAPI（NetscapePlugin Application Programming Interface），NPAPI主要依据网页排版引擎Gecko的插件API，NPAPI 原本是由 Netscape 所制定的一组单纯的 C Plugin API，起初是无法支持 Scriptability；于是到了 2004年底时，各家浏览器 (IE, Opera, Mozilla 等) 都同意支援 npruntime 延伸NPAPI以支援 Scriptability，所以目前若是想写插件则应该以npruntime API才能跨不同的 Browsers。

（何谓Scriptability？就是Plugin实例可以作为一个脚本对象供外部访问，反之，Plugin实例也可通过接口访问外部脚本。Netscape 时代的Scripting技术就是LiveConnect是基于Java的，当时还没有JNI，Netscape自己搞了一个JRI（不知当时的 Netscape是否是内置了JVM的，因为没有JVM支持的话，页面脚本也是无从执行的），后来Sun主推了JNI，JRI就废弃了。Mozilla只 好又联合若干家相关Vendors搞了一套新的Scripting技术。相当好，独立于Java，简洁实惠。能与脚本互访，插件的灵活性和实用性大为提高。）

## 2.2 开发框架和工具

参考：<https://developer.mozilla.org/en-US/docs/Plugins/External_resources_for_plugin_creation>

1、**FireBreath**

FireBreath是一个基于C++开发的跨平台浏览器插件框架。基于FireBreath框架所开发的插件可运行在包括FireFox 3.0以上版本的很多主流浏览器中。

其支持的平台包括：Windows， Linux，以及Mac OS X。其支持的开发环境包括：Visual Studio 2005及以上版本（Windows），XCode（Mac OS），GCC（Linux）。所有平台均依赖于CMake进行跨平台的编译。

参考：<http://www.firebreath.org/>

2、**Nixysa**

Nixysa是一个基于Python所写的自动生成NPAPI插件（包括支持Chrome和FireFox的插件）源码的框架。它能够是开发人员能够简单的通过IDL描述将C++类暴露给JavaScript。

参考：<http://code.google.com/p/nixysa/>

3、**QtBrowserPlugin**

QtBrowserPlugin 解决方案能够简单地创建支持包括Mozilla FireFox, Safari, Opera, Google Chrome, QtWebKit等支持NPAPI的主流浏览器的插件。

参考：<http://doc.qt.digia.com/solutions/4/qtbrowserplugin/developingplugins.html>

# 上传、下载

经过测试发现市场上绝大多数邮箱如Sina邮箱, QQ邮箱，网易邮箱等都能够在国产平台上正常使用附件的上传、下载功能。如表2所示代码可以在国产环境运行。

表2 Firefox支持file控件代码

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >

<head>

<title>Untitled Page</title>

</head>

<body>

<input id="File1" type="file" /><input id="btAdd" type="button" value="Add" onclick="alert(document.getElementById('File1').value);" />

</body>

</html>

其运行结果如图1，图2。

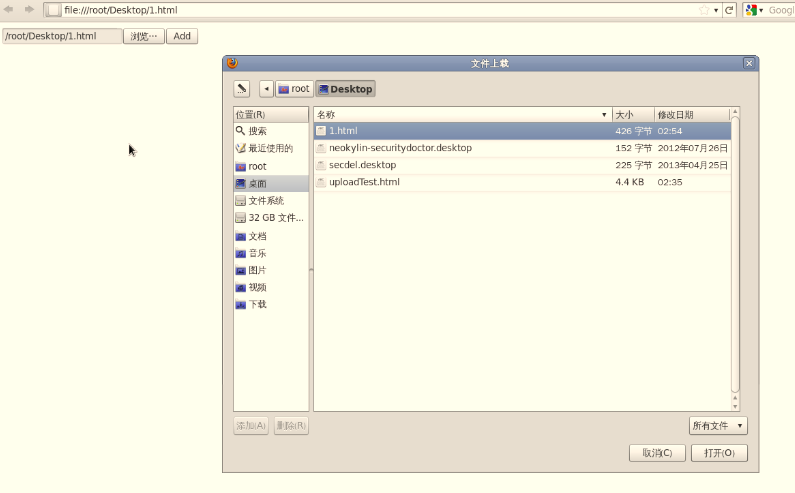


图1 点击浏览时显示效果

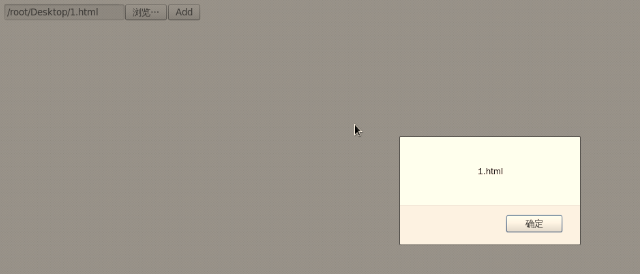


图2 点击Add按钮时效果

# 四、Firebreath框架使用简介

## 4.1 FireBreath项目简介

FireBreath是一个开源的跨平台、跨浏览器插件开发框架，其主要的目标就是提供一个能够快速而简单地创建跨平台、多浏览器支持的插件的解决方案。通过该框架创建的插件能够运行在不同平台(Windows, Linux, Mac os)的支持NPAPI或者ActiveX技术的不同浏览器上。

其浏览器支持列表如下：

* NPAPI浏览器(支持平台包括windows, mac os以及 linux)：
  + Firefox 4.0及以上（已验证可行）
  + Chrome 2及以上（已验证可行）
  + Safari（支持，未验证）
  + Opera（支持，未验证）
* ActiveX技术浏览器：
  + IE 6, 7, 8, 9, 10及以上（已验证可行）

## 4.2 FireBreath框架的环境搭建与使用

#### 4.2.1 Windows下框架环境搭建及开发

##### 框架依赖项列表

* CMake （必要）

CMake 是一个开源的跨平台自动化建构系统，它并不直接建构出最终的软件，而是产生标准的建构档（如Unix的Makefile或Windows Visual C++的projects/workspaces）。框架中使用CMake自动构建不同平台下的建构档，如在Linux下构建Makefile，以及在Windows下构建VS工程和解决方案（solution）。

* Python （必要）
* Visual Studio 2005, 2008或以上版本（必要）

##### 环境搭建

* 当前使用操作系统及相关依赖项版本
  + 操作系统：Windows 7
  + 编译器： Visual studio 2012

并非必须使用VS 2012，在使用其他版本的VS时，只需要运行相应的工程生成脚本即可（后边会提到）。

* + Python版本号：2.7.5

下载地址：

<http://www.python.org/ftp/python/2.7.5/python-2.7.5.msi>

之所以使用Python 2.7.5是由于FireBreath工程创建脚本（fbgen.py）在Python 3.X版本上运行会出现问题，故建议使用Python 2.X版。

* + CMake版本号：2.8.11.2

下载地址：

<http://www.cmake.org/files/v2.8/cmake-2.8.11.2-win32-x86.exe>

* + FireBreath版本号：1.7

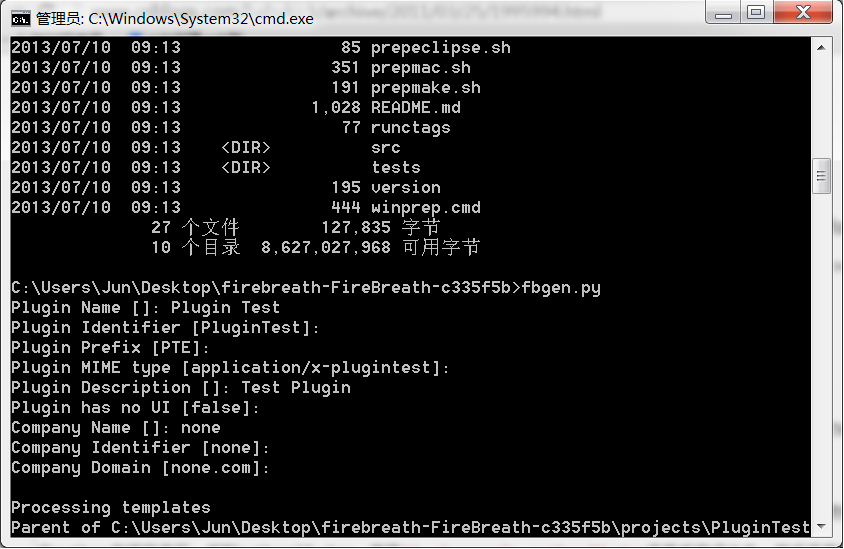
下载地址：

<https://github.com/firebreath/FireBreath/zipball/firebreath-1.7>

##### Windows下插件开发

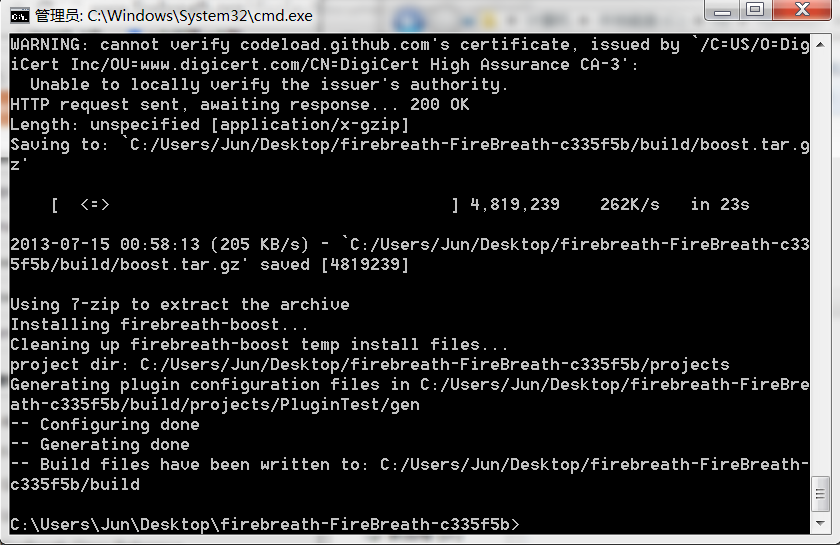
###### 创建解决方案

1. 解压缩FireBreath1.7压缩包。
2. 打开windows命令行操作窗口。
3. 进入解压后的FireBreath目录下。
4. 运行fbgen.py脚本，输入项目名称等信息。



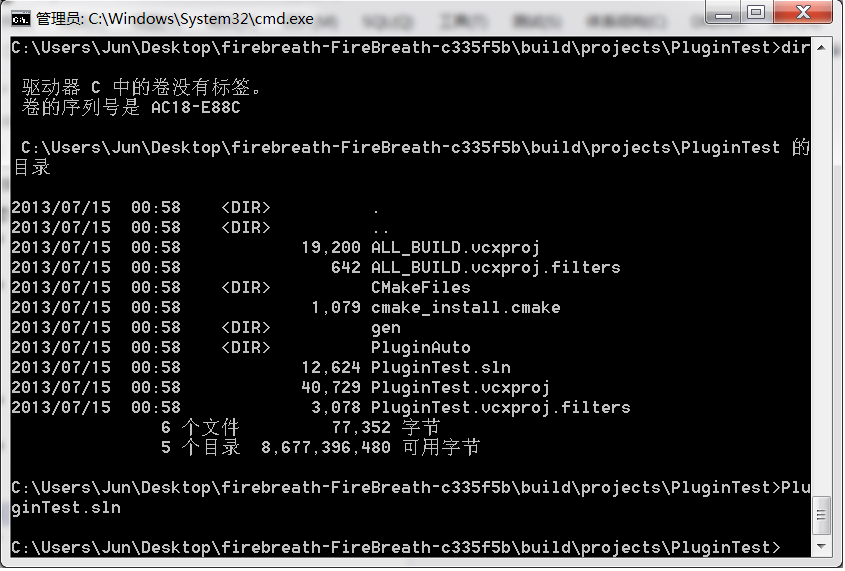
1. 运行与编译器相应的解决方案生成脚本（这里运行的是prep2012.cmd）。

C:\Users\Jun\Desktop\编程软件\MyFiles\FB\5.bmp

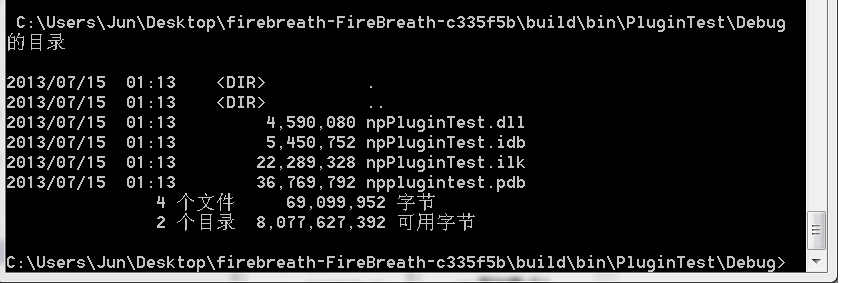


###### 编译与安装

1. 进入FireBreath目录下新增的build文件夹下的projects文件夹中，打开刚刚新建的解决方案。

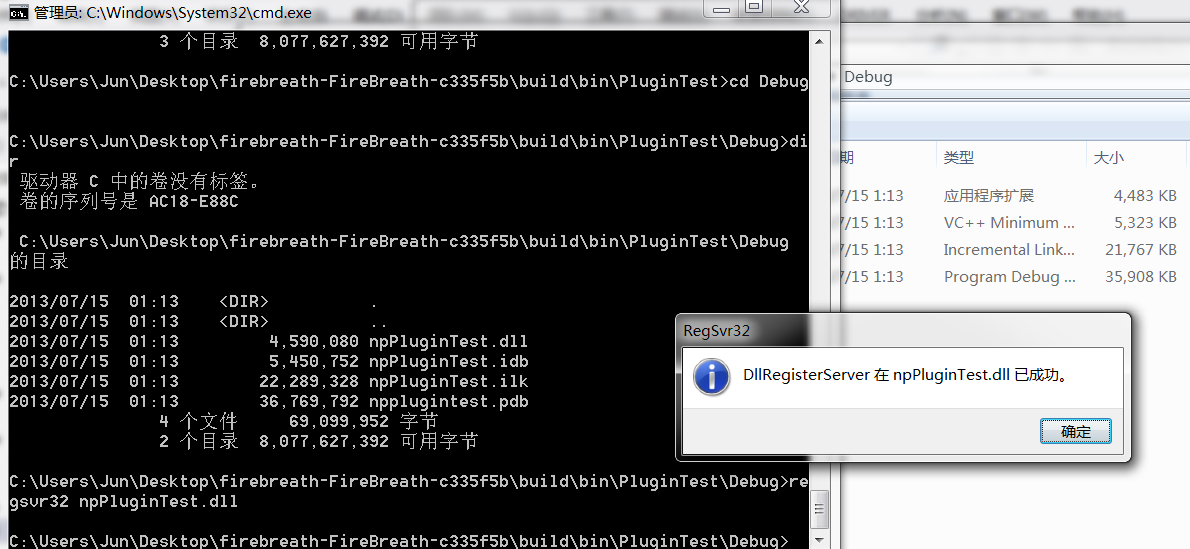


1. 在VS 2012中编译解决方案。
2. 进入build文件夹下的bin文件夹中，在对应的工程文件夹中可找到编译好的dll文件。



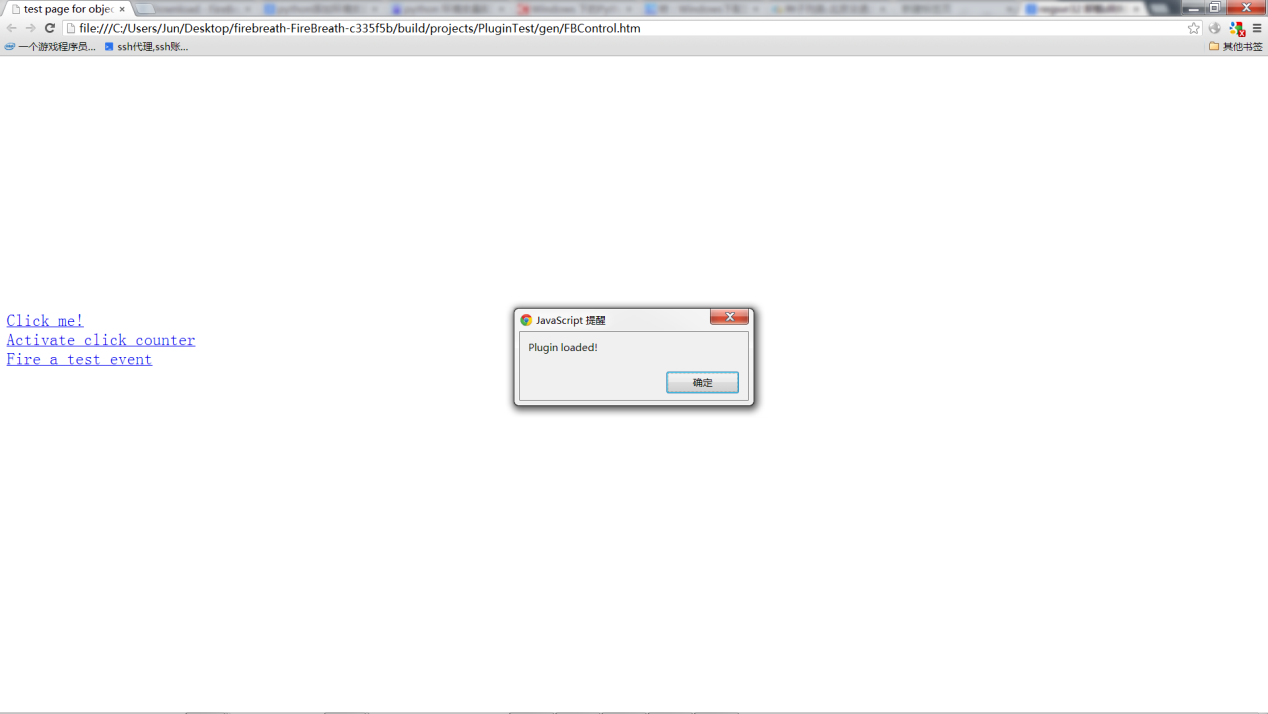
1. 将dll文件在系统中注册。

在命令行中使用命令：regsvr32 XXXX.dll对编译好的dll文件进行注册，即可在浏览器中使用插件。



1. 进入“build/projects/项目名称/gen”文件夹中，在不同浏览器中打开插件测试文件FBControl.htm。

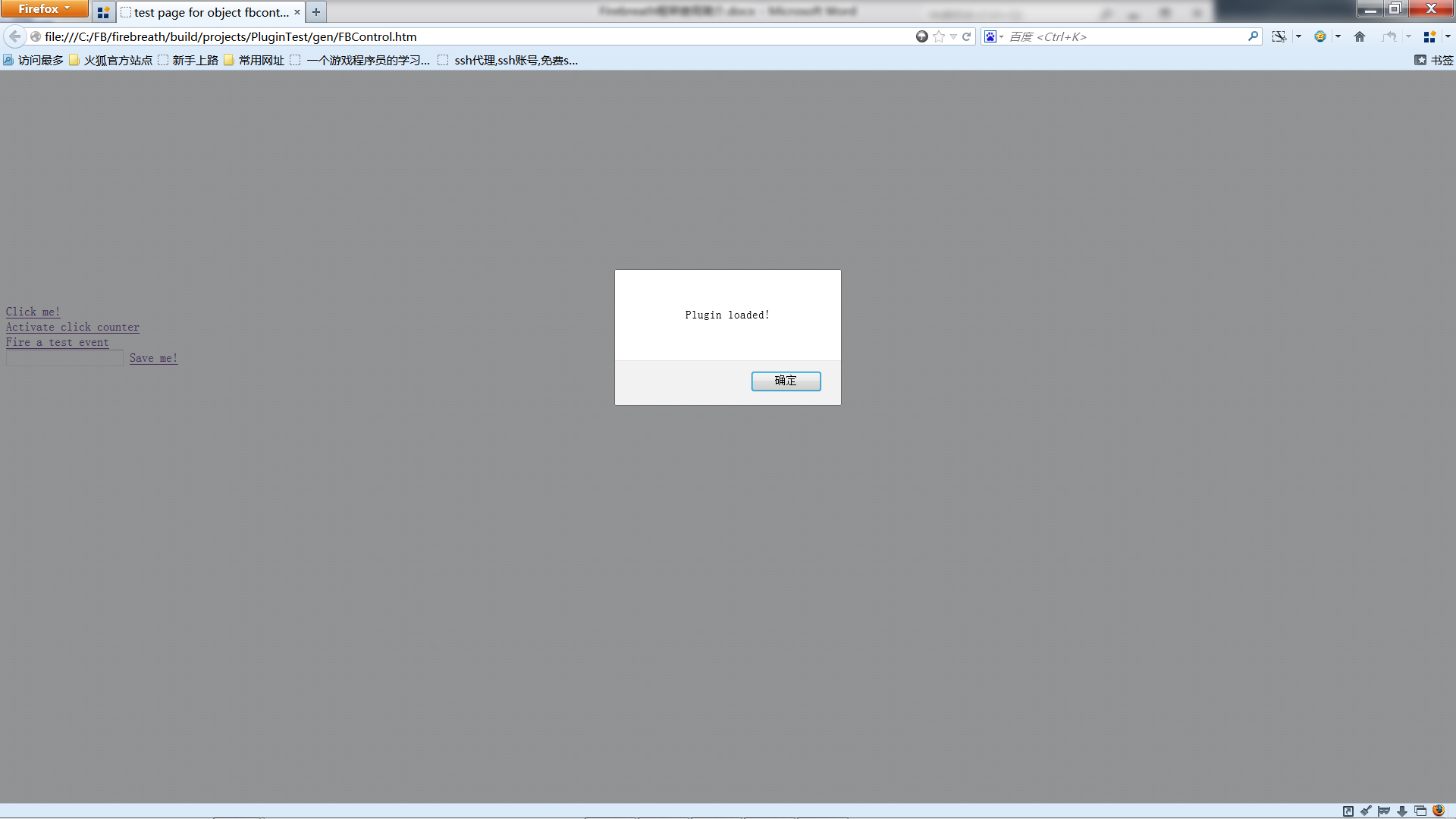
测试文件FBControl.htm中提供了几个自动生成的测试接口的调用，以下为在Chrome浏览器中打开测试文件的效果图。



###### 插件调试

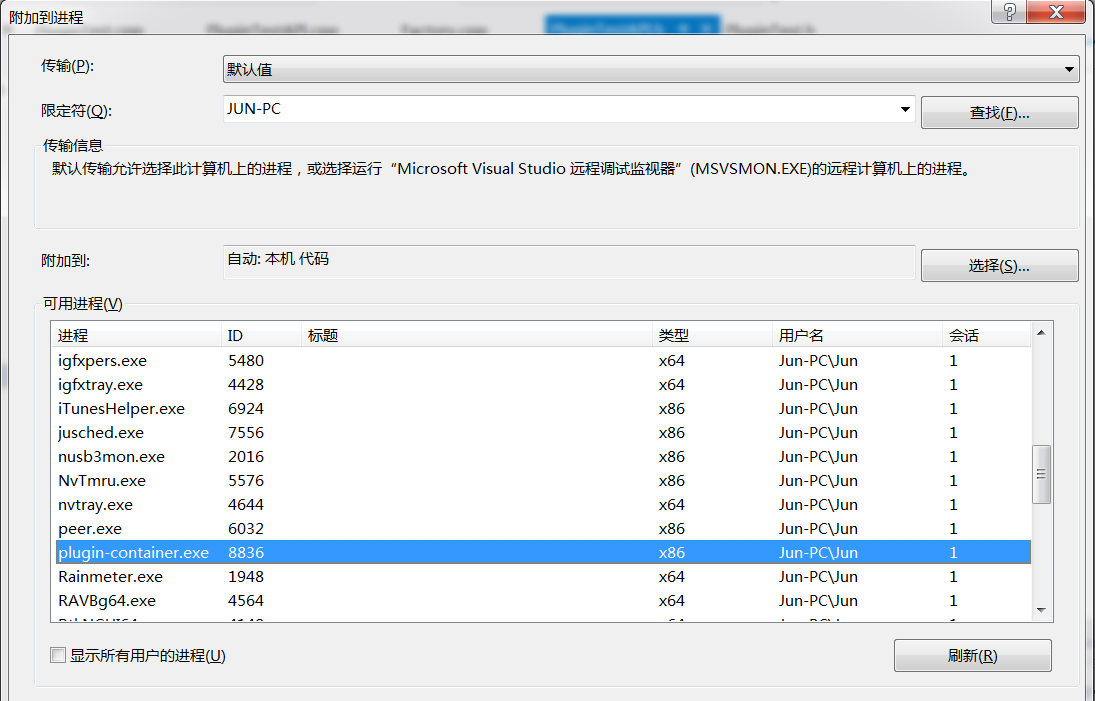
对于Windows下FireBreath插件的调试，采用进程附加进行调试，这里以Firefox为例。

1. 首先需要打开Firefox，这里直接使用Firefox打开了测试文件FBControl.htm。

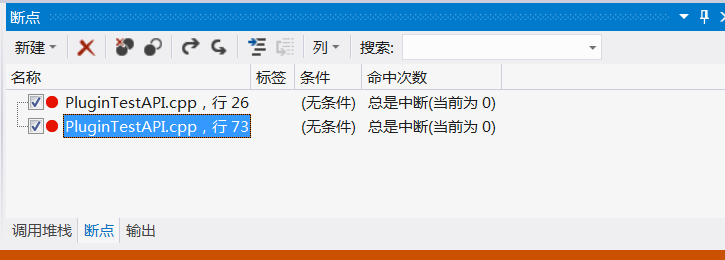


1. 对Firefox进程进行附加。

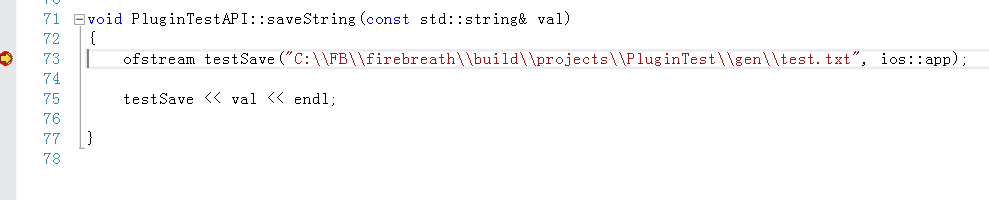
一般情况下Firefox浏览器会有两个以上的进程，只要附加到进程plugin-container即可。对于其他浏览器，若不知道插件调用是在哪一个进程中，则需全部附加，这里建议全部附加。



附加进程后便会看到先前设置的断点为可命中状态：



1. 点击在浏览器中打开的测试文件的调用相关接口的相应按钮，则可发现成功命中了断点。



#### 4.2.2 Linux下框架环境搭建及开发

##### 框架依赖项列表

* CMake（必要）
* gtk2.0-devel（非必要）
* Python（必要）
* Shell（必要）
* git（非必要）
* 编译器（必要，如gcc，gcc-c++）

##### 环境搭建

* 当前使用操作系统及相关依赖项版本
  + 操作系统：Fedora
  + 编译器： gcc 4.8.1, gcc-c++ 4.8.1

安装命令：sudo yum install gcc gcc-c++

* + Python版本号：2.7.5

安装命令：sudo yum install python2.7

使用Python 2.7的理由同前。

* + CMake版本号：2.8.11

安装命令：sudo yum install cmake

* + gtk版本号：2.0

安装命令：sudo yum install gtk2.0-devel

* + Shell版本号：4.2
  + git版本号：1.8.3.1

安装命令：sudo yum install git

* + FireBreath版本号：1.7

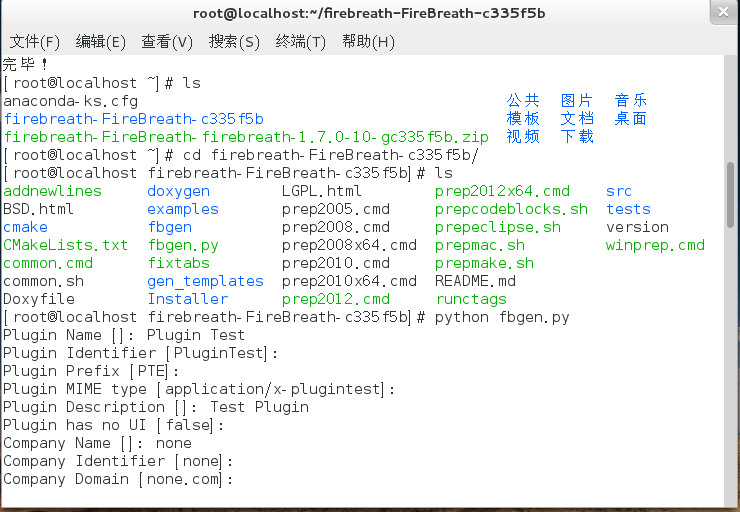
下载地址：

<https://github.com/firebreath/FireBreath/tarball/firebreath-1.7>

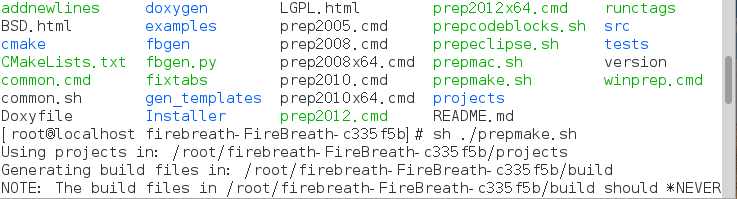
##### Linux下插件开发

###### 创建工程

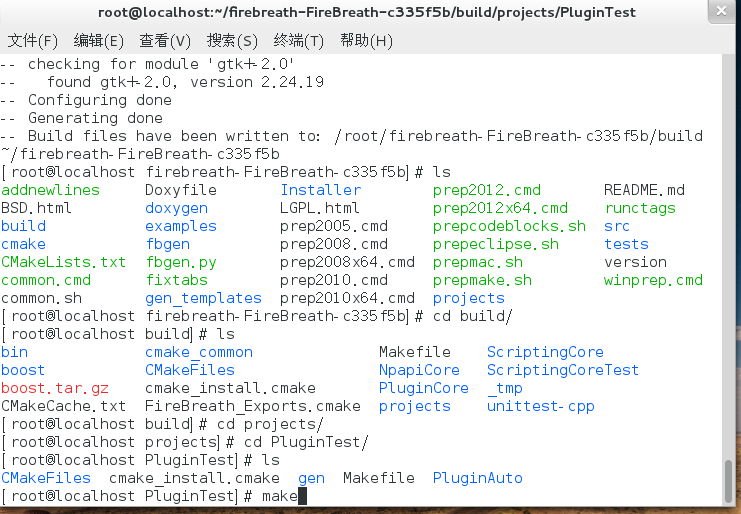
1. 解压缩FireBreath1.7压缩包。
2. 打开终端，进入解压后的FireBreath目录。
3. 运行fbgen.py脚本，输入项目名称等信息。



1. 运行与编译器相应的解决方案生成脚本（这里运行的是prepmake.sh）。

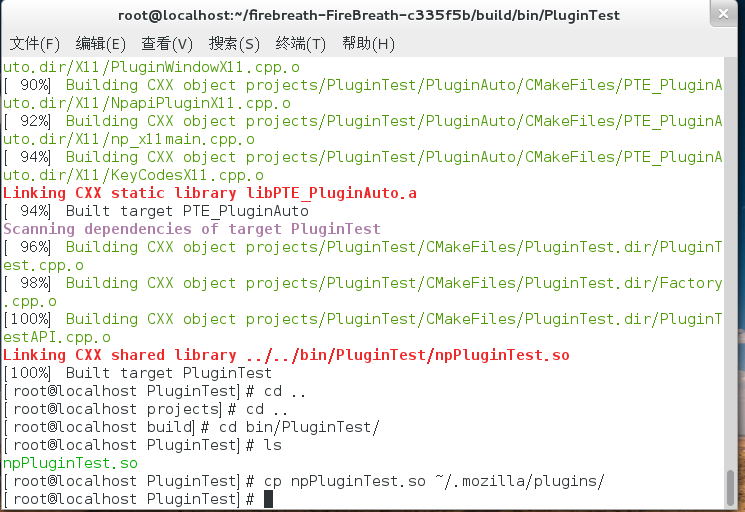


成功生成makefile文件：



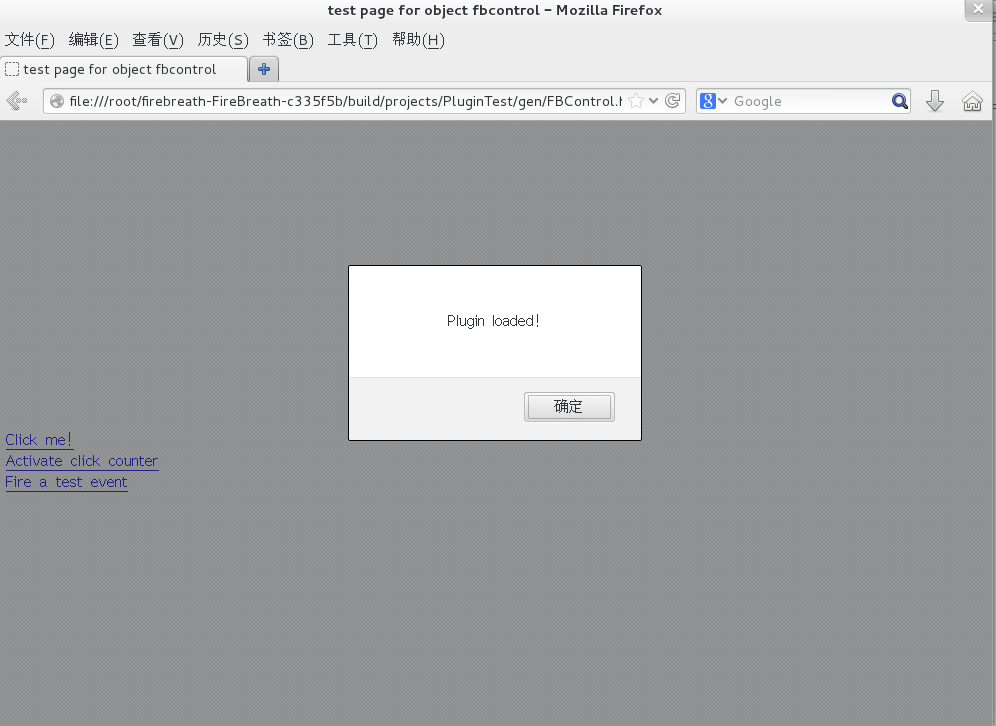
###### 编译与安装

1. 进入FireBreath目录下新增的build文件夹下的“/projects/项目名称”文件夹中。
2. make编译工程。
3. 进入build文件夹下的bin文件夹中，在对应的工程文件夹中可找到编译好的.so文件。



1. 将文件复制到浏览器插件目录下，这里是~/.mozilla/plugins。
2. 进入“build/projects/项目名称/gen”文件夹中，在不同浏览器中打开插件测试文件.htm。

在firefox下打开测试文件效果：



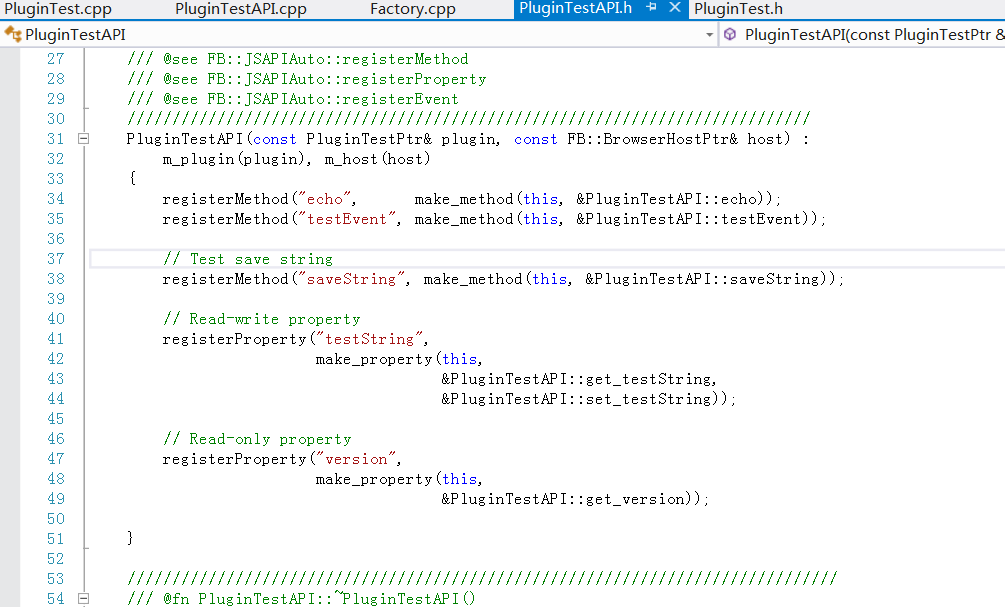
#### 4.2.3 插件自定义接口创建示例

1. 在工程中注册接口

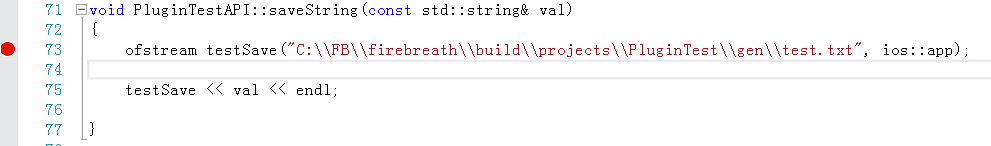
一般情况下，接口的注册需要在类XXXXXXAPI的构造函数中进行。如图所示，示例工程中在PluginTestAPI的构造函数中注册了如下几个接口：

Method类型：echo，testEvent，saveString

Property类型：testString，version



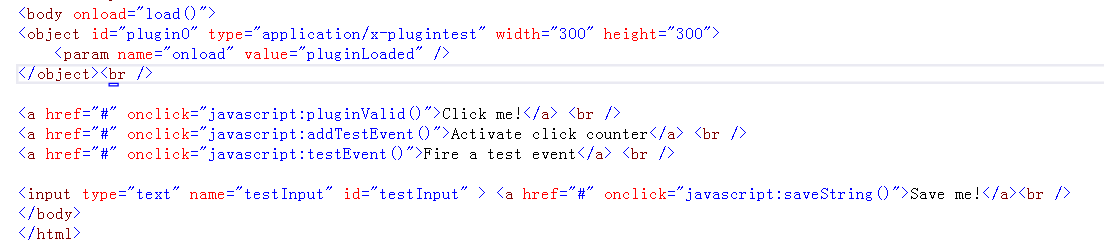
其中，saveString为自定义的接口，其功能为接受一个字符串，并把它保存到一个文件中，这里为了方便起见，文件路径为绝对路径，实际开发中可使用相对路径。其实现如下：



FireBreath接口类型有三种，分别为Method，Property，Event，对于这三种类型接口的详细说明请参考：参考资料8。

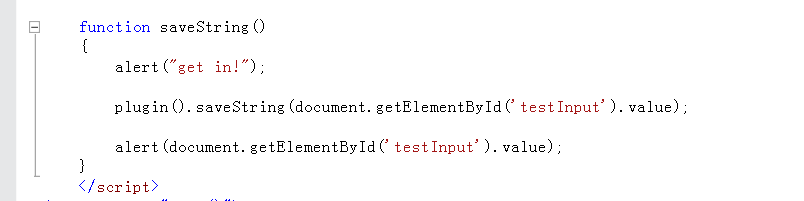
1. 页面中添加插件

在页面中，使用object标签将插件引入页面。示例中同时定义了一个输入框，可通过保存将输入框中的值保存到文件中。



1. 使用JS调用接口

示例中，“Save me”触发了一个JS的调用，在这一JS函数中，调用了保存接口saveString：



1. 解决方案工程结构简介

解决方案中主要使用的工程为创建插件时所创建的自定义工程，示例中为PluginTest工程，该工程中有两个类，分别为PluginTest和PluginTestAPI，其中PluginTest中定义了几个鼠标、窗口事件，其实现中默认只是进行了简单的返回操作，实际开发中可对其进行修改，以进行更加复杂的事件响应操作。在PluginTestAPI中可进行自定义接口的创建操作。

详细请参考：参考资料7。

## 4.3 其他

#### 注意事项

1. 插件可在windows下进行开发，若需要在Linux下运行则需要在Linux下重新编译。在Linux下重新编译之前，需要先将build目录删除，并重新运行工程生成脚本（Linux下可用prepmake.sh），然后再按相应步骤进行编译和安装。

#### 参考资料

1. 跨浏览器插件框架FireBreath安装与使用之一——Windows下的插件 ：

<http://blog.csdn.net/cheungmine/article/details/7941167>

1. 跨浏览器插件框架FireBreath安装与使用之二——Linux下的插件：

<http://blog.csdn.net/cheungmine/article/details/7982220>

1. FireBreath学习之一 开发环境搭建

<http://cuihao0532.blog.163.com/blog/static/14091811520121118700548/>

1. FireBreath学习之二 Summary--方法 属性 Attributes 事件和js的交互等

<http://cuihao0532.blog.163.com/blog/static/14091811520121124422048/>

1. FireBreath学习之三 JSAPIAuto类的成员函数说明和插件调用DOM方法/属性一般形式

<http://cuihao0532.blog.163.com/blog/static/140918115201211251420643/>

1. FireBreath学习之四 FB::Variant类 FB::VariantList类 和一些建议

<http://cuihao0532.blog.163.com/blog/static/140918115201211269300441/>

1. FireBreath Class Reference

<http://www.firebreath.org/display/documentation/FireBreath+Class+Reference>

1. [Using FireBreath](http://www.firebreath.org/display/documentation/Using+FireBreath)

<http://www.firebreath.org/display/documentation/Using+FireBreath>

1. [FireBreath Home](http://www.firebreath.org/display/documentation/FireBreath+Home) Page

<http://www.firebreath.org/display/documentation/FireBreath+Home>