



Application Notes

OTA 用户手册
Revision 0.2

Amlogic Confidential!

Amlogic, Inc.
3930 Freedom Circle
Santa Clara, CA 95054
U.S.A.
www.amlogic.com

Legal Notices

© 2013 Amlogic, Inc. All rights reserved. Amlogic® is registered trademarks of Amlogic, Inc. All other registered trademarks, trademarks and service marks are property of their respective owners.

This document is Amlogic Company confidential and is not intended for any external distribution.

目录

1. 简介.....	4
2. 配置 OTA 服务器.....	5
2.1 搭建 OTA 服务器的环境.....	5
2.2 OTA 服务器的配置.....	8
2.3 配置之后的服务器测试.....	15
3. 生成 OTA 差分包.....	16
3.1 目标.....	16
3.2 正常编译生成当前版本的 zip 包.....	16
3.3 生成新版本与某个旧版本之间的差分包.....	16
4. 配置 OTA 客户端.....	17
4.1 更新附件文件.....	17
4.2 在 system.prop 中添加.....	17
4.3 将 OTAUpgrade.apk 放到/system/app 下.....	17
5. 实例.....	18
5.1 目标.....	18
5.2 配置 parameter.conf.....	18
5.3 配置 debug.conf.....	18
5.4 配置 ota_update_example.xml.....	19
5.5 准备升级包.....	19

Amlogic Application Notes

修改记录

Revision	Date	Owner	Changes
0.1	March 13, 2013	Yu Cheng	Draft
0.2	Jul 25, 2013	Jiquan Liu	<ul style="list-style-type: none">● Add some remarks● Enrich chapter 5.5● Update index

Amlogic Confidential!

1. 简介

OTA 升级的作用是通过网络, 将放在 OTA 服务器上的升级文件, 升级到 OTA 客户端, 其过程如下:

配置 OTA 服务器和 OTA 客户端

- 在 OTA 客户端发送下载请求
- 服务器端通过请求后, 实现下载 update.zip 到 OTA 客户端的板子上
- OTA 客户端升级 update.zip

本用户手册主要阐述以下四个步骤:

- 配置 OTA 服务器
- 生成 OTA 差分包
- 配置 OTA 客户端
- 实例

Amlogic Confidential!

2. 配置 OTA 服务器

2.1 搭建 OTA 服务器的环境

安装 java 虚拟机。JDK1.7 或 1.6 都可以。

下载并安装 Apache tomcat 服务器(JDK1.6 以上)

- 在命令行模式下 cd 进入到解压目录下的 bin 目录, 如:
`cd /d E:\apache- tomcat-6.0.32\bin`
- 安装服务:
`service.bat instal`
- 双击 apache-tomcat-6.0.32\bin 下的 tomcat6.exe, 启动服务器。
- 配置该 tomcat 服务器 conf 目录下的 web.xml 使其支持后缀名称为 .img .aml 以及无后缀名称的脚本的下载。其配置方式如下:

```
<mime-mapping>
  <extension></extension>
  <mime-type>application/octet-stream</mime-type>
</mime-mapping>
<mime-mapping>
  <extension>aml</extension>
  <mime-type>text/plain</mime-type>
</mime-mapping>
<mime-mapping>
  <extension>img</extension>
  <mime-type>application/zip</mime-type>
</mime-mapping>
```

验证 Apache tomcat 服务器

在 IE 浏览器里输入 <http://localhost:8080/>，如果出现图 2.1 所示界面就代表服务器搭建成功。

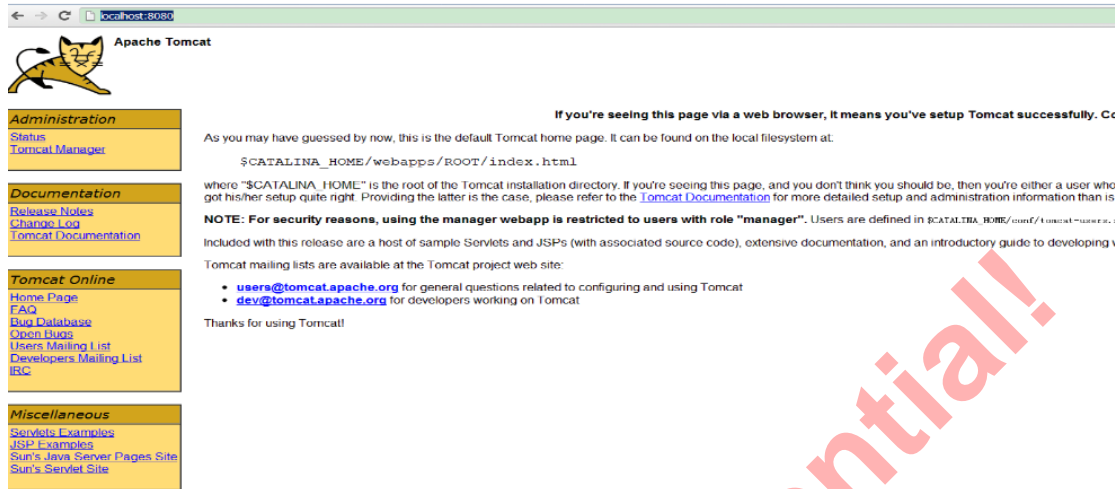


图 2.1 - Tomcat 服务器验证成功界面

部署

附件中的 otaupgrade.war 文件是通过附件“OTA 升级发布流程解析”的服务器端实现的一个演示。在图 2.1 中选择“tomcat manager”进入图 2.2 所示界面，然后在“WAR or Directory URL”栏中选择附件中提供的 otaupgrade.war 进行部署。

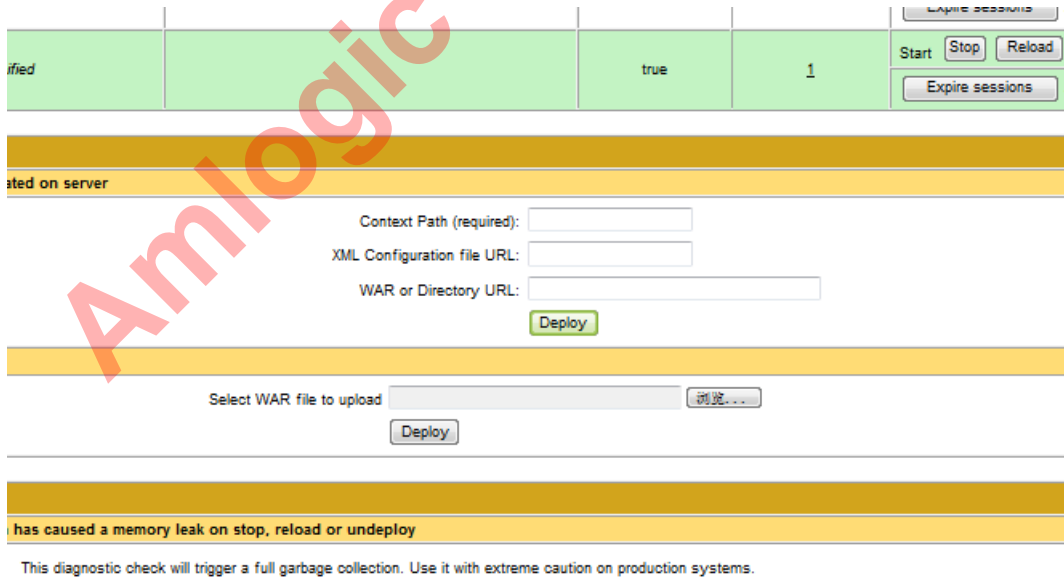
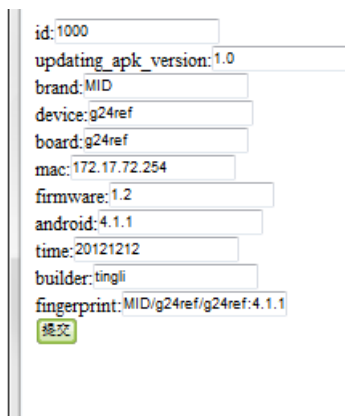


图 2.2 - 部署界面

验证部署

访问 <http://127.0.0.1:8080/otaupdate/form.jsp>，如果出现图 2.3 所示界面，就表示部署成功。



The screenshot displays a web form with the following fields and values:

id:	1000
updating_apk_version:	1.0
brand:	MID
device:	g24ref
board:	g24ref
mac:	172.17.72.254
firmware:	1.2
android:	4.1.1
time:	20121212
builder:	tingli
fingerprint:	MID/g24ref/g24ref:4.1.1

At the bottom of the form, there is a green button labeled "提交" (Submit).

图 2.3 - 部署验证成功界面

〈注〉：至此表明 OTA 服务器的环境已经搭建完毕。

2.2 OTA 服务器的配置

OTA 服务器端的 parameter.conf 配置文件

用于验证客户端身份，对于符合 android 属性的客户端要在此配置文件中检索更新。

```
device=g24ref  
board=g24ref  
android=4.1.1
```

OTA 服务器端的 update.conf 配置文件

目前 OTA 服务器端配置用的是 debug.conf，请配置这个文件

- 同一系统时的配置情况：

```
00411001=http://192.168.0.124:8080/update/xml/update/ota_test_1.xml  
00411002=http://192.168.0.124:8080/update/xml/update/ota_test_2.xml  
lastest=00411003
```

- 整个 android 系统大版本更新 4.1.1-4.2.0，*为通配。

```
0041104*=http://192.168.0.124:8080/update/xml/update/ota_test_1.xml  
004113**=http://192.168.0.124:8080/update/xml/update/ota_test_1.xml  
.....  
00420001=http://192.168.0.124:8080/update/xml/update/ota_test_3.xml  
00420002=http://192.168.0.124:8080/update/xml/update/ota_test_4.xml  
lastest=00411003
```

<注>:

- 等号左边的是固件版本号
- 右边是对应的 xml 指令文件的连接地址
- lastest=00411003 表示发布的新版本是 00411003 版本

脚本情况下 OTA 服务器端的 xml 指令文件的配置方式

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<updata>
  <command force="false" name="update_with_script">
    <url name="update.zip" locattr="/storage/external_storage/sdcard1/update.zip"
updatezip="true">http://10.18.19.10:8080/otaupdate/xml/download/zip/02_03_update.zip</
url>
    <md5 name="update.zip">cc4a842ce8e50aacc02bab6e234cb7dc</md5>
    <url name="uImage_recovery"
locattr="/storage/external_storage/sdcard1/recovery.img"
updatezip="false">http://10.18.19.10:8080/otaupdate/xml/download/zip/recovery.img</url
>
    <md5 name="uImage_recovery">46BC1F23569D90EB3DE6CD78FB6154BB</md5>
    <url name="factory_update_param.aml"
locattr="/storage/external_storage/sdcard1/factory_update_param.aml"
updatezip="false">http://10.18.19.10:8080/otaupdate/xml/download/script/factory_update
_param.aml</url>
    <md5 name="factory_update_param.aml">75ADBC35C5FB655142B3514BB29FD9CC</md5>
    <storagemem>3145728</storagemem>
    <!--语言从上往下选择，如何没有相对应的语言，就选择英文。country 的值是按照
android 标准获取的国家代码-->
    <description country="CN" language="zh">1. 修正鼠标有时无法使用的 bug. 2. 优化视频
播放效果</description>
    <description country="TW" language="zh">1. 修正鼠標有時無法使用的 bug. 2. 優化視頻
播放效果.</description>
    <description country="ELSE" language="en">1. fix a bug that the mouse can't be
used sometimes. 2. perfect the video play effect.</description>

  </command>
</updata>
```

<注>：md5 后面的一串值是通过 MD5.exe 工具计算出的所要升级文件的希哈值，用来检验下载后的文件大小是否相同，文件是否在下载过程中有损坏。

非脚本情况下 OTA 服务器端的 xml 指令文件的配置方式

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<updata>
  <command force="false" name="update_with_inc_ota">
    <url name="update.zip" locattr=""
updatezip="true">http://10.18.19.10:8080/otaupdate/xml/download/zip/02_03_update.zip</
url>
    <md5 name="update.zip">cc4a842ce8e50aacc02bab6e234cb7dc</md5>
    <storagemem>3145728</storagemem>
    <!--语言从上往下选择，如何没有相对应的语言，就选择英文。country 的值是按照
android 标准获取的国家代码-->
    <description country="CN" language="zh">1. 修正鼠标有时无法使用的 bug. 2. 优化视频
播放效果</description>
    <description country="TW" language="zh">1. 修正鼠标有时无法使用的 bug. 2. 優化視頻
播放效果.</description>
    <description country="ELSE" language="en">1. fix a bug that the mouse can't be
used sometimes. 2. perfect the video play effect.</description>
  </command>
</updata>
```

<注>：md5 后面的一串值是通过 MD5.exe 工具计算出的所要升级文件的希哈值，用来检验下载后的文件大小是否相同，文件是否在下载过程中有损坏。

OTA 服务器端的 xml 指令文件的配置配置说明

- 升级文件放置的位置是由服务器端<url>标签下的<locattr>属性指定的。若该属性值为空，则会放置到 data 区进行升级。
- <storagemem>标签记录了将要下载文件所需要的总空间的大小，表明对外部 sdcard 容量（脚本大小+升级包大小）的要求，在 storage 为 true 的情况下有效。
- Command namd= “update_with_script” 表明该情况为脚本升级，需要同时支持脚本，升级包以及对 sdcard 设备的要求。该情况下客户端只是对升级包进行 md5 验证，并不验证安装包是否可以正确安装，所有安装过程交给脚本执行
- MD5 值可以通过附件中生成（图 2.4）

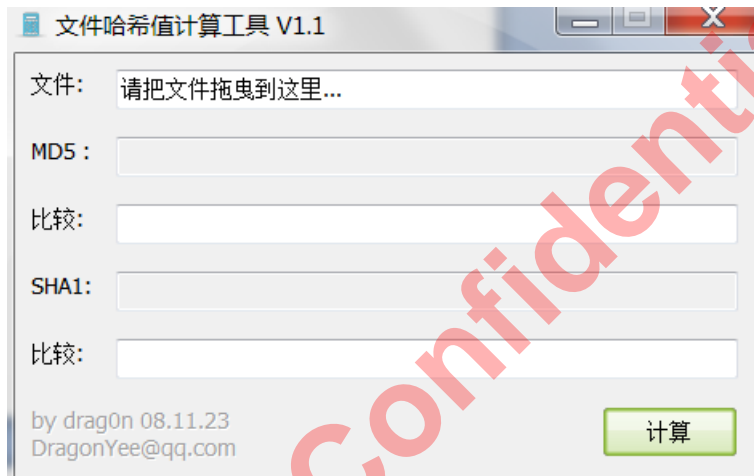


图 2.4 - 生成 MD5 的工具界面

- IP 地址可以在这里查看（图 2.5）

Server Information							
Tomcat Version	JVM Version	JVM Vendor	OS Name	OS Version	OS Architecture	Hostname	IP Address
Apache Tomcat/7.0.23	1.6.0_21-b07	Sun Microsystems Inc.	Windows XP	5.1	x86	aml-fa607489aa2	10.28.19.44

图 2.5 - 查看 IP 地址的界面

OTA 客户端的信息列表

在 OTA 客户端向 OTA 服务器请求更新的时候，OTA 客户端会向 OTA 服务器发送信息列表

```
<id>188</id>
<updating_apk_version>1</updating_apk_version >
<brand>MID</brand>
<device>g24ref</device>
<board>g24ref</board>
<mac>00. 11. 22. 33. 44. 55</mac>
<firmware>1. 0</firmware>
<android>4. 1. 1</android>
<time>20120301. 092251</time>
<builder>buildernam</builder>
<fingerprint>00421003</fingerprint>
```

〈注〉:

- OTA 客户端的变量名要和 OTA 服务器的 parameter.conf, debug.conf 和 update.conf 三个配置文件里面的变量相对应。
- 附件中 Demo 的服务器对 brand, board, firmware 和 android 进行了要求。

服务器执行流程（图 2.6）

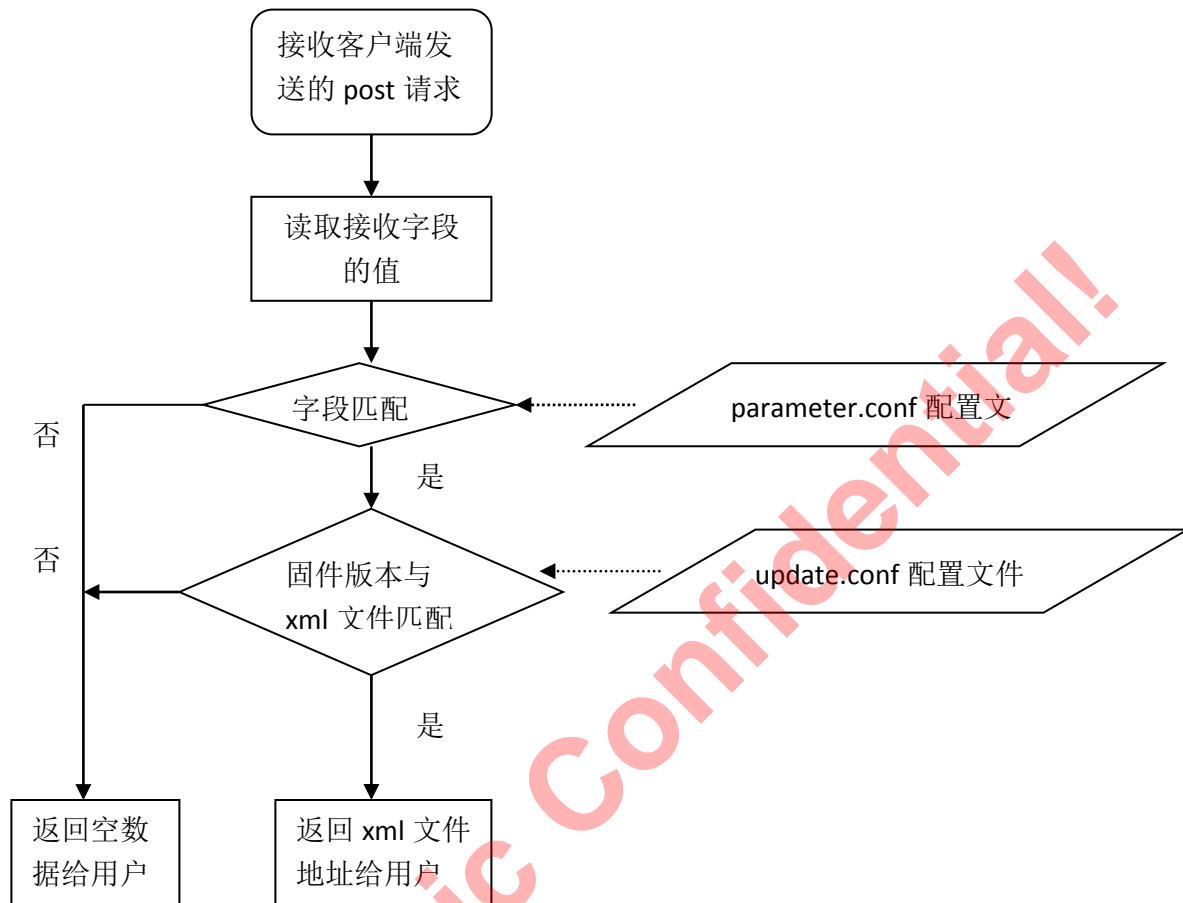


图 2.6 - 服务器执行流程

服务器配置的注意事项

- 每次放置好升级包后，需要<reload>该 webapp，配置才可以生效（图 2.7）。

<u>/otaupdate</u>	None specified		true	3	Start Stop Reload Undeploy
					Expire sessions with idle <input type="text" value="30"/> minutes

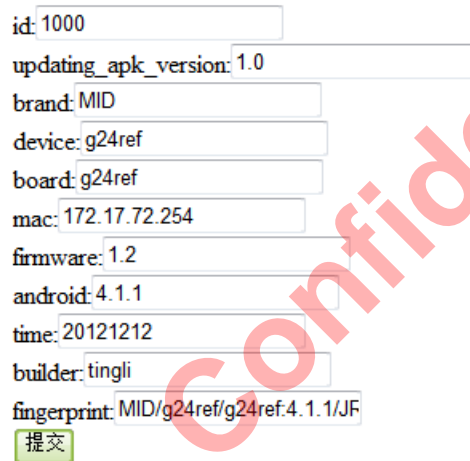
图 2.7 - 重载 webapp 的界面

- 每个文件的 md5 值要计算正确。
- 差分包在发布之前，先要在本地升级，以确定其是否正确。

Amlogic Confidential!

2.3 配置之后的服务器测试

- 服务端配置了一个用于测试的 form.jsp
- 配置好服务器之后，在浏览器输入 192.168.0.124/update/form.jsp 进入测试界面（图 2.8）
- 界面显示了客户端提交的信息列表，根据服务器实际情况加以修改，然后提交，便可模拟客户端发送的 post 请求
- 如果信息匹配正确且 firmware 版本有对应的 xml 文件，那么将返回 url=对应 xml 文件链接地址。例如：url=http://192.168.0.124/update/xml/debug/v0.3.xml。
- 如果信息不匹配，那么返回是空的，即浏览器显示是空的。
- 如果信息匹配了，但是 firmware 版本不对，那么返回的是 url=null，即浏览器显示的是“url=null”。



The screenshot shows a web form with the following fields and values:

- id: 1000
- updating_apk_version: 1.0
- brand: MID
- device: g24ref
- board: g24ref
- mac: 172.17.72.254
- firmware: 1.2
- android: 4.1.1
- time: 20121212
- builder: tingli
- fingerprint: MID/g24ref/g24ref.4.1.1/JF

At the bottom of the form is a green button labeled "提交" (Submit).

图 2.8 - form.jsp 测试界面

<注>:

- 测试的 id 号是 1000
- 1000 调用的是 debug.conf, 非 1000 调用的是 update.conf
- Updating_apk_version 有最新的 1.3 版本，用最新的 1.3 版本
- <firmware>要填写 update.conf 里面的版本号

3. 生成 OTA 差分包

3.1 目标

需要生成多个差分包。例如现在需要发布 0.4 版本，之前已经发布了 0.1，0.2 和 0.3 三个版本，那么需要生成三个差分包：

- 0.4 与 0.1 之间的差分包
- 0.4 与 0.2 之间的差分包
- 0.4 与 0.3 之间的差分包

3.2 正常编译生成当前版本的 zip 包

- 执行以上命令后，在目录 `out/target/product/g**ref*/obj/PACKAGING/target_files_intermediates` 下生成 `xxx-target_files-xxxxxxx.zip`：
`$ make otapackage -j`
- 该文件即当前版本的 zip 包 `target_files_package`，需要做备份。
- 当有新版本升级时，需要用到该文件参与生成新版本的差分包。

3.3 生成新版本与某个旧版本之间的差分包

- 将新版本的 zip 包 `target_files_package` 命名为 `new.zip`
- 将备份的旧版本的 zip 包 `target_files_package` 命名为 `old.zip`
- 将 `new.zip` 和 `old.zip` 放在 android 根目录的自定义文件夹 `ota` 下（如 `android4.1.1/ota`）
- 执行以下命令生成 `ota/update.zip`
`$./build/tools/releasetools/ota_from_target_files -i ota/old.zip ota/new.zip ota/update.zip`

<注>：

- 生成差分包之后，可以通过重新命名方便记忆每个差分包对应的是哪两个版本的。例如：
`v0.1-v0.4.zip`，`v0.2-v0.4.zip` 和 `v0.3-v0.4.zip`。
- 生成该差分包之后，可以将该文件复制到本机，做本地升级验证。
- 本地升级验证成功后，如客户需要，则申请放到服务器上实现在线升级的功能。

4. 配置 OTA 客户端

4.1 更新附件文件

```
Android 4.1----4.1\0001-make-ota-script-work.patch  
Android 4.2----4.2\0001-make-ota_script-work.patch
```

〈注〉：现在 4.1 系统及以上版本不需要增加这个 patch

4.2 在 system.prop 中添加

```
#support for OTAUpdate  
ro.product.firmware=00421001  
ro.product.otaupdateurl=http://10.28.19.44:8080/otaupdate/update
```

〈注〉：IP 地址根据自己的设置

4.3 将 OTAUpgrade.apk 放到/system/app 下

```
cp /storage/external_sdcard/sdcard1/OTAUpgrade.apk /system/app/OTAUpgrade.apk  
chmod 644 /system/app/OTAUpgrade.apk  
reboot
```

〈注〉：android4.1以后 OTA 升级的 OTAUpgrade.apk 已经加入到系统中无需此步骤。

5. 实例

5.1 目标

配置 Tomcat 下面的 parameter.conf, debug.conf 和 ota_update_example.xml。

5.2 配置 parameter.conf

```
android=4.1.1  
device=g24ref  
board=g24ref
```

5.3 配置 debug.conf

```
00421001=http://10.28.19.44:8080/otaupdate/xml/update/ota_update_example.xml  
lastest=00421002
```

5.4 配置 ota_update_example.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<update>
  <command force="false" name="update_with_inc_ota">
    <url name="update.zip" locattr=""
updatezip="true">http://10.28.19.44:8080/otaupdate/xml/download/zip/update.zip</url>
    <md5 name="update.zip">461C4C706F6252F29132E746833FB87B</md5>
    <storagemem>1394606</storagemem>
    <!--语言从上往下选择，如何没有相对应的语言，就选择英文。country 的值是按照
android 标准获取的国家代码-->
    <description country="CN" language="zh">1. 修正鼠标有时无法使用的 bug. 2. 优化视频
播放效果</description>
    <description country="TW" language="zh">1. 修正鼠标有时无法使用的 bug. 2. 優化視頻
播放效果.</description>
    <description country="ELSE" language="en">1. fix a bug that the mouse can't be
used sometimes. 2. perfect the video play effect.</description>
    <!--当前差分包的 fingerprint:
softwinners/crane_evb_v12r/crane-evb-v12r:4.0.4/IMM76D/20120628:eng/test-keys
当差分包对应升级版本的 fingerprint:
softwinners/crane_evb_v12r/crane-evb-v12r:4.0.4/IMM76D/20120628:eng/test-keys
-->
  </command>
</update>
```

5.5 准备升级包

- 拷贝升级包 update.zip, recovery.img 文件到:
Tomcat 7.0\webapps\otaupdate\xml\download\zip\update.zip
- 拷贝 factory_update_param.aml 升级脚本文件到:
Tomcat 7.0\webapps\otaupdate\xml\download\script\factory_update_param.aml
- 配置对应的 conf 文件，在机器上打开 OTAUpgrade.apk 进行升级。