Procedure for running AnyStop release build script

Written I. Yulaev 2012-01-01
Contact ivan@yulaev.com for questions

1 Overview

This document outlines the procedure for building the release version of the AnyStop app series. Building the release binaries is a pre-requisite to publishing a new version of the app. We will overview system-level requirements for being able to run the script, setting up the script, running it, and troubleshooting the build process.

2 Script Pre-Requisites

Before running the script, some software must be installed on the build computer. Ensure that have you have installed

1) Cygwin, **if you are not using a linux/osx system.** It is recommended to install all of the Cygwin packages.
2) Python 2.6 with the ElementTree library. For Windows-based PCs, the easiest way to get this is to install Cygwin with all allowed packages. *If you have trouble with the ElementTree library see section 5.1 of this guide*
3) The Java Development Kit. Amount other things this is required for installing the Android SDK (next item).
4) Have the latest Android SDK with all APIs and libraries. Specifically, we will be targeting Android + Google APIs version 14 (4.0).
5) Ant. On windows PCs you can most easily get this by installing WinAnt.

Make sure your PATH contains android-sdk/tools, JDK, and Ant. The last two are added by default by their respective installers; you may have to add android-sdk/tools to your path manually.

3 Script Set-Up

After satisfying the pre-requites, we can begin collecting the files to run the build script. The build script and its auxiliary files are located in the build/ folder of the AnyStop project top-level. It is recommended, although not required, that you move this folder elsewhere on your system before beginning the build process.

Ensure that the following files are present in the build/ folder.

1) busbrothers.keystore. *This file is not distributed with the AnyStop project by default, talk to Brendan or Ivan if you need this file.*
2) copy
3) create.sh
4) libguard.txt
5) parseexec.py
6) stringsBase.xml
7) title.png

If any of these files are not present contact Ivan (see document header).

Make sure that stringsBase.xml has the same format as the current res/values/strings.xml file in the AnyStop project. It is possible that the stringsBase.xml file is out of date and must be updated. If this is the case, just use the current res/values/strings.xml file and replace the string entries for the names called out in the below XML with the below text. This will allow the build script to correctly generate a strings.xml file for each agency that is built.

```
<string name="app_name">AnyStop: agencyLongToken</string>
<!-- Agency settings -->
<string name="agencyName">agencyNameToken</string>
<string name="agencyTable">agencyTableToken</string>
<string name="agencyPredictionType">agencyRealTimeToken</string>
<string name="agencyTag">agencyTagToken</string>
<string name="agencyLong">agencyLongToken</string>
<string name="agencyShort">agencyShortToken</string>
<string name="agencyLat">agencyLatToken</string>
<string name="agencyLon">agencyLonToken</string>
<string name="agencyZoomLevel">11</string>
```

Make sure that create.sh has the correct password for the keystore. Specifically, if the line calling jarsigner reads

```
jarsigner -verbose -keystore ../../busbrothers.keystore -storepass
NOT_A_VALID_PASSWORD $UNSIGNED_APK_FILENAME AnyStop
```

then you probably need to contact Ivan or Brendan to get the correct password.

Finally, copy the entire AnyStop Android project, minus the build folder and any version control system folders (i.e. ".hg"), into build/AnyStopBase/. Make sure that nothing else that you want to keep is anywhere in the build folder.

4 Running the Script and Interpreting Output

Running the script should be as easy as going to the build/ folder in the shell and typing

```
python parseexec.py all
```

The first, and only, argument to parseexec.py tells the build script what agencies to build the binaries for. The "all" argument will build APKs for all agencies. Otherwise, you may manually enter the tag for a particular agency and only that APK will get built.

The script will create and sign the APK for the agency selected. It will create the Android marketplace description for the agency-specific APK. Both will be placed in the build/out/ folder. Make sure that the script outputs the string "BUILD SUCCESSFUL"; if it outputs "BUILD FAILED" then something went wrong and debugging will be necessary.

5 Debugging script runs

This section gives specific advice for overcoming common problems having to do with the build script.

5.1 I have some error related to ElementTree not being imported

Try changing the import directive at the top of parseexec.py. Specifically, in the lines

```
#import elementtree.ElementTree as ET
import xml.etree.ElementTree as ET
```

comment out the second line (add a '#' at the beginning of the line) and un-comment the first. This may solve your issues.

5.2 The Android compilation process (via javac) complains about not being able to find AdView (or something else)

This is a symptom of either the libraries not being found, or the libraries being present but corrupted. For some reason the Android project update command likes to corrupt library JAR files. So, the default build script (create.sh) copies the libraries over from AnyStopBase after android update has been run on the AnyStop project. Make sure that the libraries are being copied into the build folder correctly. More specifically, make sure that the following lines in create.sh point to valid library files in build/AnyStopBase/

```
cp ../AnyStopBase/lib/AdWhirl/AdWhirlSDK_Android_3.1.1.jar
./libs/AdWhirlSDK.jar
cp ../AnyStopBase/lib/FlurryAnalytics/*.jar ./libs
cp ../AnyStopBase/lib/GoogleAdMobAdsSdkAndroid-4.3.1/GoogleAdMobAdsSdk-
4.3.1.jar ./libs/GoogleAdMobAdsSdk.jar
```

Also, make sure that build/AnyStop/proguard.cfg contains the following line:

```
-libraryjars /libs/*.jar
```