Autodesk®
Scaleform®

Mobile SDK ReadMe for Android

Autodesk®
GAMEWARE

Copyright Notice

Autodesk® Scaleform® 4.x

Disclaimer

THIS PUBLICATION AND THE INFORMATION CONTAINED HEREIN IS MADE AVAILABLE BY AUTODESK, INC. “AS IS.” AUTODESK, INC. DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE REGARDING THESE MATERIALS.

Prerequisites

The following must be installed to use the Autodesk Scaleform Mobile SDK for Android:

PC & Mac

Android SDK (recommended minimum API level 10)
Android NDK (recommended r9d)
Java JDK 1.6 (JRE alone is not sufficient and 1.7 not recommended by Google)
Apache Ant 1.8 or later

PC

Cygwin (The default install along with the following packages)

- binutils
- make
- perl

Displaying a Flash File on your Android Device

In order to run a flash file, you can use one of the two player packages shipped with this package.

License Key

A license key is necessary to run our tools (Amp and Exporter), our players (FxPlayerTiny, FxPlayerMobile and ShippingMobile Player) and/or use our libraries.

Creating a License Key

You should have received a key from the Gameware Website or Autodesk E-Store that looks something like this:

"ABCDEFGHIJKLMNOPQRSTUVWXYZ12345A67890ABCDEFGHIJKLMNOPQRSTUVWXYZ"

That key must be copied into a text file named "sf_consumer_license_android_mobile.txt".

Using the License Key

To use the license key with our PC/Mac based tools, place "sf_consumer_license_android_mobile.txt" in your SDKs "Bin" directory.

To use the license key with our players on actual devices there are two approaches:

1. During development, the easiest approach is to simply copy the "sf_consumer_license_android_mobile.txt" to whatever gets mounted as /sdcard on your Android device. You can accomplish this from a Cygwin prompt in the directory containing your license file by issuing the following command:

   ```
   $ adb push sf_consumer_license_android_mobile.txt /sdcard/
   ```

2. Copy the license key into the code itself (This is the method you should use for your final Shipping App). We provide a function called "GFx_SetEvalKey" which can be used to pass the key to Scaleform. This function must be called before any Scaleform related classes are loaded, otherwise Scaleform initialization will not take place. For your convenience, we have placed the function (commented out by default) in the main Platform implementation for both the FxPlayerMobile and ShippingMobile Players (Src/Platform/Platform_Android.h).

FxPlayerMobile

This is the mobile version of the GFxPlayer that can be used to play SWFs located on the SD card of your Android device as well as view performance statistics. A prebuilt release player is included in this package and located in the following location (the "-debug" indicates it was signed with a debug key):

Bin/Android/FxPlayerMobile_Release_NoRTTI-debug.apk

In order to run the player on your device, do the following:

1. Install the prebuilt player onto your device with the following command:
   ```
   $ adb install Bin/Android/FxPlayerMobile_Release_NoRTTI-debug.apk
   ```

2. Place some flash content in the /sdcard/ location for the device:
   ```
   $ adb push <SOME_FLASH_FILE>.swf /sdcard/flash.swf
   ```

3. Please note that you may also need to place the license file "sf_consumer_license_android_mobile.txt" in /sdcard (please see the licensing section above for more details):
   ```
   $ adb push sf_consumer_license_android_mobile.txt /sdcard/sf_consumer_license_android_mobile.txt
   ```

4. Please also note that when FxPlayerMobile first initializes it looks for a special flash file called "flash.swf" in the root of /sdcard. This file will always be the first file loaded. You can use the command above to push any arbitrary flash file to "/sdcard/flash.swf". Once the player has loaded, you can switch between additional SWF files via the HUD, accessible via the "Menu" button on your device.

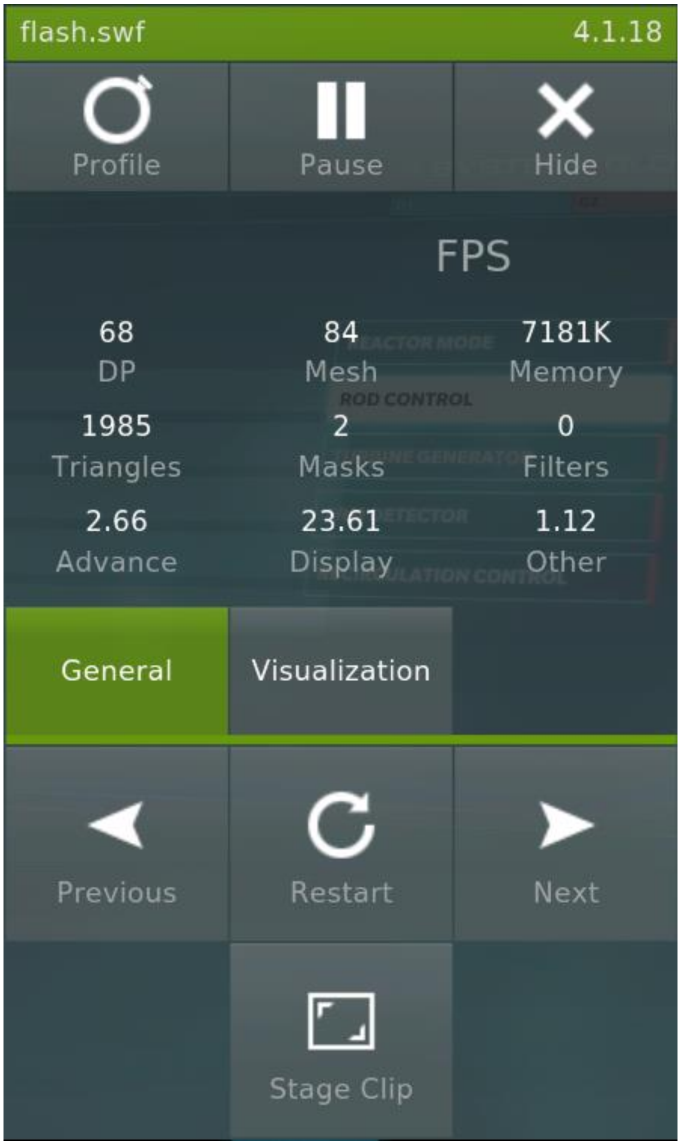
   To push additional flash content to the device without renaming, use the following command:
   ```
   $ adb push Bin/Data/AS3/Samples/3DGenerator_AS3.swf /sdcard/
   ```

5. Launch the player by selecting its icon in the application drawer. You should now see the initial flash.swf file running on your device.

Using the HUD

In order to make it easy for users to cycle between SWFs and view performance statistics, FxPlayerMobile comes with a HUD (which is itself a swf file that is baked in to the FxPlayerMobile executable). Once you have FxPlayerMobile running, press the "Menu" button on the Android to bring out the HUD. The main page of the HUD has the following buttons:

Profile – Switches the Profiling Mode (Profile (FPS) -> Fast Forward (Full Advance) -> Off)
Pause – Toggles pausing the current SWF Movie
Hide – Closes the HUD
Previous – Cycles to the previous SWF Movie
Restart – Restarts the current SWF Movie
Next – Cycles to the next SWF Movie
Stage Clip – Toggles cropping the SWF Movie to the dimensions of the stage

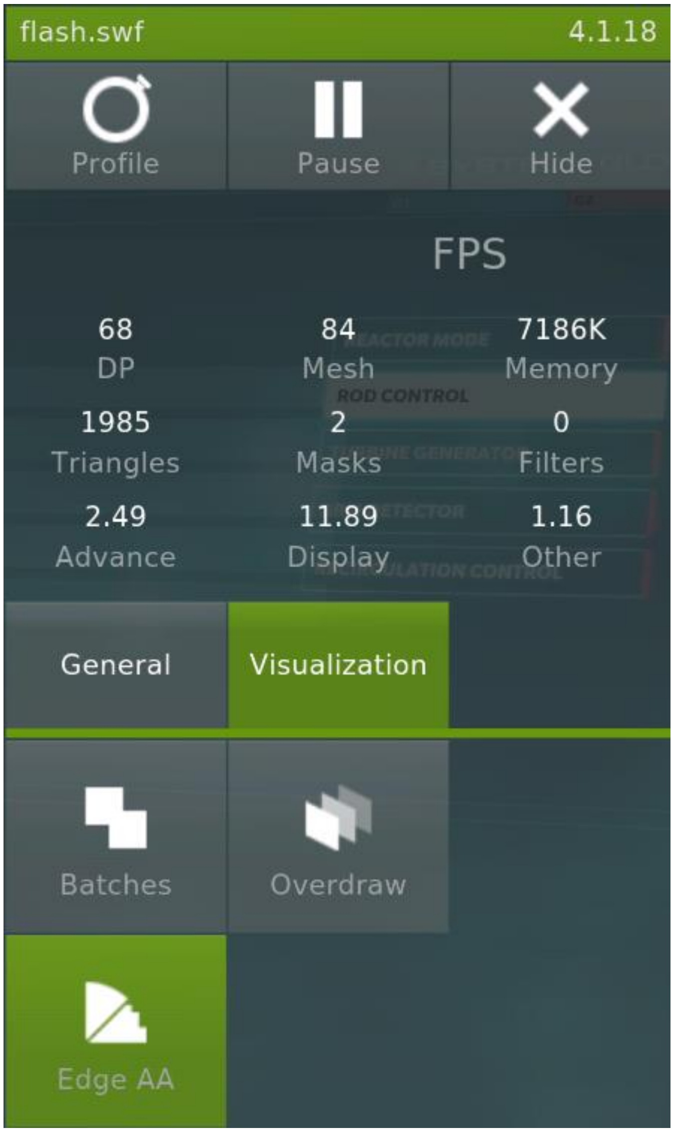


FPS

68 DP
84 Mesh
7181K Memory
1985 Triangles
2 Masks
0 Filters
2.66 Advance
23.61 Display
1.12 Other

Within the “Visualization tab” you will find the following buttons:

- Batches – Shows the textures that have been batched into one draw call
- Overdraw – Shows the amount of overdraw (Darker = more overdraw, Red = Masks, Blue = Filters)
- EdgeAA – Turns EdgeAA on or off



```
                  FPS
68          84          7186K
DP          Mesh        Memory
1985        2           0
Triangles   Masks       Filters
2.49        11.89       1.16
Advance     Display     Other
```

Shipping Mobile Player

The Shipping Mobile Player offers customers a way to package their application and content within a shippable APK which after proper signing, can be uploaded to the Android Market.

The Shipping Mobile Player project is created based on a DEF (definition) file that our build system translates into a complete, standalone Android project. One of our sample DEF files is StarforceTD.def (located in the root of LocalApps). An example shipping player named StarforceTD.apk is located in "LocalApps/StarforceTD/StarforceTD.apk". Starforce is a tower defense game that provides a best-practices sample implementation of a mobile, touch-based game. More information can be found in the sf_4.x_mobile_game_kit_overview.pdf document located in /Doc.

In order to deploy this game, please use:

```
$ adb install LocalApps/StarforceTD/StarforceTD.apk
```

This will install the StarforceTD application on your Android device. In order to run it, please tap on the corresponding icon.

Manually Modifying Build Settings for the Shipping Player

The build settings for the shipping player can be modified by editing a DEF file which defines all of the necessary settings for the APK including the Application's display name and content.

For an example on how to build a Shipping Mobile Player based on a DEF file, let's take a look at the StarforceTD.def file located in the root of the LocalApps folder.

With "DISPLAY_NAME", you can set the name of your app as you would like it to appear on your device:

DISPLAY_NAME := Starforce TD

You can also set the package prefix for your application:

PACKAGE := com.scaleform

Like FxPlayerMobile, the Shipping Mobile Player needs a startup swf named "flash.swf". Instead of renaming your content, you can set "STARTUP_FLASH" to your main swf and the build system will automatically rename it when it generates the APK:

STARTUP_FLASH := Bin/Data/AS3/Kits/StarforceTD/StarforceTD.swf

The rest of your application's resources (including entire folders) can be added by setting the "RESOURCES" variable:

```
RESOURCES := Bin/Data/AS3/Kits/StarforceTD/StarforceMenu.swf \
             Bin/Data/AS3/Kits/StarforceTD/LoadingView.swf \
             Bin/Data/AS3/Kits/StarforceTD/audio \
             Bin/Data/AS3/Kits/StarforceTD/conf
```

Orientation for your application can be set with the "ORIENTATIONS" variable. If only one orientation is set, your app will be locked to that orientation. With no orientation set, your application will be able to rotate across all orientations.

```
ORIENTATIONS := landscape
```

EdgeAA and FMOD sound support can be turned on or off with their respective variables:

```
EDGEAA := 1
USE_FMOD := 1
```

The generated APK can be automatically installed on your device when building with the AUTODEPLOY variable and automatically launched with the AUTORUN variable:

```
AUTODEPLOY := 1
AUTORUN := 1
```

Manually Building the Shipping Mobile Player

Now that your DEF file is set, it's time to build the Shipping Mobile Player (in this case, StarforceTD). If this is the first time you've built an Android application on this machine, there are 2 additional steps you must perform this first time building only:

1. Create a debug.keystore file to be able to sign your applications for development. To do so open up Cygwin or the Mac terminal and run the following commands one at a time (using copy/paste is recommended):

```
$ mkdir -p ~/.android
$ cd ~/.android
$ /cygdrive/c/Android/jdk/bin/keytool -genkeypair -alias androiddebugkey -keypass android -keystore
debug.keystore -storepass android -dname "CN=Android Debug,O=Android, C=US" -validity 9999
$ cd -
```

More information about the debug.keystore file and application signing in general can be found at: http://developer.android.com/tools/publishing/app-signing.html

2. While still in the console, change directories to the root of your Scaleform SDK. A Makeconfig file needs to be created in the root of the Scaleform SDK. The Makeconfig tells the build system where your Android NDK, Android SDK, Java SDK and Ant binaries are located. This package comes with a sample Makeconfig "Makeconfig_Example" for you to use as a reference. Please copy this file to a new file called "Makeconfig" and edit the Android section accordingly paying particular attention to the comments therein.

Now that that is done we can actually initiate a build. This is the command you would use for iteration each time you make source code changes but is not necessary for Flash only content changes. It should be run from the root directory of the Scaleform SDK in either a Cygwin or Mac Terminal:

```
$ make P=Android C=Release+NoRTTI
```

Note: This will build the "Release" configuration (`C=Release+NoRTTI`) which is suitable for use during development as it will perform close to the final "Shipping" (`C=Shipping+NoRTTI`) configuraiton but also allow profiling with our AMP tool (see below). There is also the "Debug" (`C=Debug+NoRTTI`) configuration which will allow for debugging your application code within the appropriate IDE.

Deployer

The Mobile SDK comes with a graphical tool called the Deployer (located in the Bin directory), which will configure a DEF file and automatically build an application based on it without the user ever having to open up the terminal. Please see the Deployer User Guide for more information on this tool.

AMP

The SDK comes with a memory and performance tool (AMP) that can be used to profile your application while it is running on the device. To connect AMP to your device, both the device and the system running AMP must be on the same wireless network (direct connection via usb is not supported at this time). Once your device has an IP address, put that address into AMP's connection dialog box and press connect. For more detailed information on AMP, please see our AMP User Guide.

Next Steps

Scaleform Mobile SDK comes with an extensive set of documentation. If you are unfamiliar with the SDK, please read the "sf_4.x_getting_started_with_scaleform" document. If you are already familiar with our SDK, then play around with the FxPlayerMobile HUD, go through the source code for StarforceTD mobile game and read the other documents containing a wealth of information about all the various aspects of using Scaleform.

If you have any questions or feedback, please post them at:
http://area.autodesk.com/forum/game-developer-zone/scaleformmobile-development/