Lombard Beta Release Notes

Flash Player 14 and AIR 14 Release Notes

Welcome to Adobe® Flash® Player 14 and Adobe AIR 14!

Last Updated: July 23rd, 2014

This beta release includes new features as well as enhancements and bug fixes related to security, stability, performance, and device compatibility for Flash Player 14 and AIR 14. This document may be updated periodically as more information becomes available.

As always, we appreciate all feedback. We encourage you to post in our beta forums or create bug reports or feature requests on our public bug database.

Flash Player Beta Forum - http://forums.adobe.com/community/labs/flashruntimes/flashplayer
AIR Beta Forum - http://forums.adobe.com/community/labs/flashruntimes/air
Bug Database - https://bugbase.adobe.com

NOTE:

- The ActiveX Flash Player in this release is not compatible with Windows® 8
  Flash Player for Windows® 8 is available as part of the generally available Windows® 8 update

New and Updated Features

- **Preview: PPAPI Debugger for Mac OS X and Windows**
  An early preview of the PPAPI content debugger is available for Mac OS X and Windows. Flash Developers using Opera (dev) and Google Chrome (dev and Canary) can now try out the PPAPI-based content debugger.

  Download plug-in for Mac OS X - for all PPAPI based browsers
  Download plug-in for Windows – for all PPAPI based browsers

  For Opera (Mac OS X and Windows):
    - Download the latest Opera Developer from http://www.opera.com/developer
      Enable the PPAPI flag in settings:
        - opera://flags/#find-pepper-flash-plugin •

  For Chrome (Windows):
    - Download the latest Chrome dev or Canary from http://dev.chromium.org/getting-involved/dev-channel
      Enable the PPAPI flag in plugins:
        - chrome://plugins
        - Expand the 'Details'
        - Ensure the PPAPI plugin in the \\Macromed\\Flash location is enabled and the PPAPI plugin in the Google Chrome location is disabled

- **AnisotropicFiltering**
  This new texture sampling filter can enhance the image quality of textures on surfaces that are at oblique viewing angles. There are two ways to enable this feature:
  1. In AGAL, set one of the values – "anisotropic2x", "anisotropic4x", "anisotropic8x", or "anisotropic16x" to the filter option in the sampling instructions.
  2. Call Context3D::setSamplerStateAt with the 3rd parameter "filter" being one of the values defined in Context3DTextureFilter - "ANISOTROPIC2X", "ANISOTROPIC4X", "ANISOTROPIC8X", or "ANISOTROPIC16X"

- **New Stage3D "Standard" Profile**
  Developers can now request this high level profile when creating Context3D. Three new features are available in this profile:
  - Multiple render target allows to you to draw geometry to multiple outputs (up to 4) during one drawing
  - Floating point texture allows you to create Texture, RectangleTexture and CubeTuxture with the RGBA16F folder.
  - AGAL v2 contains these improvements:
    - Increased register size
    - Partial derivative instructions
    - Fragment depth output
    - Conditional forward jump

- **Intel x86 Android Support**

As announced in our Flash Runtime blog, we're adding support for Intel x86 Android to AIR.

An ADT command line option (`-arch`) has been added to allow packaging apps with Android x86 support. **Please note that both shared and captive packaging is now allowed for x86 architecture.** More information can be found in this blog post: http://blogs.adobe.com/airodynamics/2014/06/26/versioning-of-multiple-apks-for-android/

Sample APK packaging command for x86 devices:

```
adt -package -target ( apk | apk-captive-runtime ) -arch x86 -storetype pkcs12 -keystore abc.p12
HelloWorld.apk HelloWorld-app.xml HelloWorld.swf
```

```
adt -package -target apk-debug <debug options> -arch x86 -storetype pkcs12 -keystore abc.p12
HelloWorld.apk HelloWorld-app.xml HelloWorld.swf
```

Note that -arch is optional. If not specified, armv7 is assumed.

Packaging for x86 architecture in Flash Builder:
Open the debug/run configurations of the project in Flash Builder and click on "Customize launch.." button. Add new parameter "-arch" with value "x86" and place it before "-storetype". Click "OK" to apply changes.

Except RTMPE and DRM, all other features and capabilities are completely functional and supported. Native extensions written for x86 platforms could also be packaged and used by an app for x86 devices. To support that, a new ANE platform 'Android-x86' is now available. Following example highlights the usage of the same -

```
<extension xmlns="http://ns.adobe.com/air/extension/14.0">
<id>com.adobe.sample.ane</id>
<versionNumber>1.0</versionNumber>
<platforms>
<platform name="Android-ARM">
<applicationDeployment>
<nativeLibrary>sample.jar</nativeLibrary>
<initializer>com.example.ane.Extension</initializer>
<finalizer>com.example.ane.Extension</finalizer>
</applicationDeployment>
</platform>
<platform name="Android-x86">
<applicationDeployment>
<nativeLibrary>sample.jar</nativeLibrary>
<initializer>com.example.ane.Extension</initializer>
<finalizer>com.example.ane.Extension</finalizer>
```

```
</applicationDeployment>
</platform>
</extension>
```

New packaging command for ANE:

```
adt -package -target ane Sample.ane extension.xml -swc Sample.swc -platform Android-ARM  -C Android-ARM/ . -platform Android-x86 -C Android-x86/ .
```

- **Improved Packaging Engine - iOS**

  Based on the feedback recieved from the developer community, tons of improvements and bug fixes have been made in the new packaging engine for iOS. We still encourage developers to report issues to http://bugbase.adobe.com, to ensure that we are able to deliver a solid packager with the final release.

  To enable this feature, please use "-useLegacyAOT no" in the ADT command, before the signing options. As of now, this feature is not available with FlashPro but it can be used with Flash builder by adding a parameter -useLegacyAOT under "Customize launch" option.

  ADT command for compiling an applications using “-useLegacyAOT no":

  ```
  adt -package -target ( ipa-test | ipa-debug | ipa-app-store | ipa-ad-hoc) -useLegacyAOT no
  -provisioning-profile <Path to profile>
  -keystore <path to .p12 file> -storetype pkcs12
  -storepass xxxx HelloWorld.ipa Helloworld-app.xml HelloWorld.swf
  ```

- **AIR Gamepad**

AIR Gamepad feature enables the app developers to provide a second screen on Android mobile devices for the Flash based browser games. AIR Gamepad API enables Flash based browser apps to connect to a paired Android device running the AIR Runtime app and therefore allowing the Android devices to be used as game controllers or second-screen interfaces.

Key Functionalities of this feature:

- Gesture events
- Touch events
- Accelerometer events
- Vibration
- Applying skin onto AIR gamepad screen

To learn more about the AIR Gamepad APIs, please refer to the documentation found here.

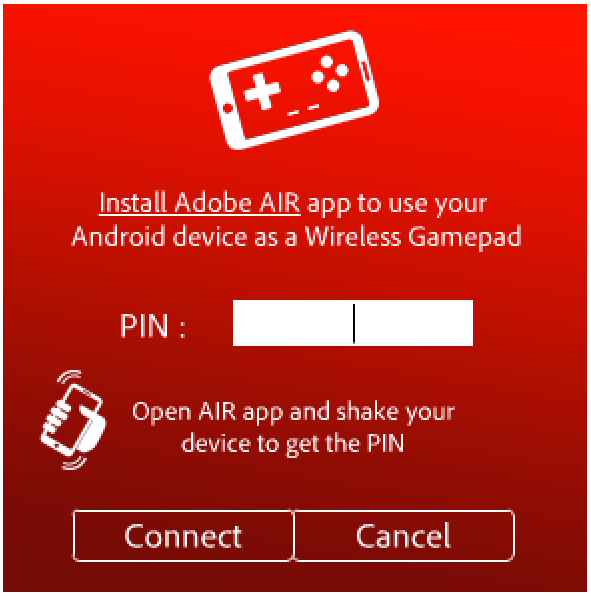
To try out the Wand.swc which can be downloaded from here.

To try out live samples of AIR Gamepad open the following links:

- ModelViewer
- HungryHero

How to use this feature(workflow):

After calling the AIR Wand connect() function in the Flash application, it shows the following message pop-up at Runtime asking for PIN to make connection with Android device running the AIR Runtime app:



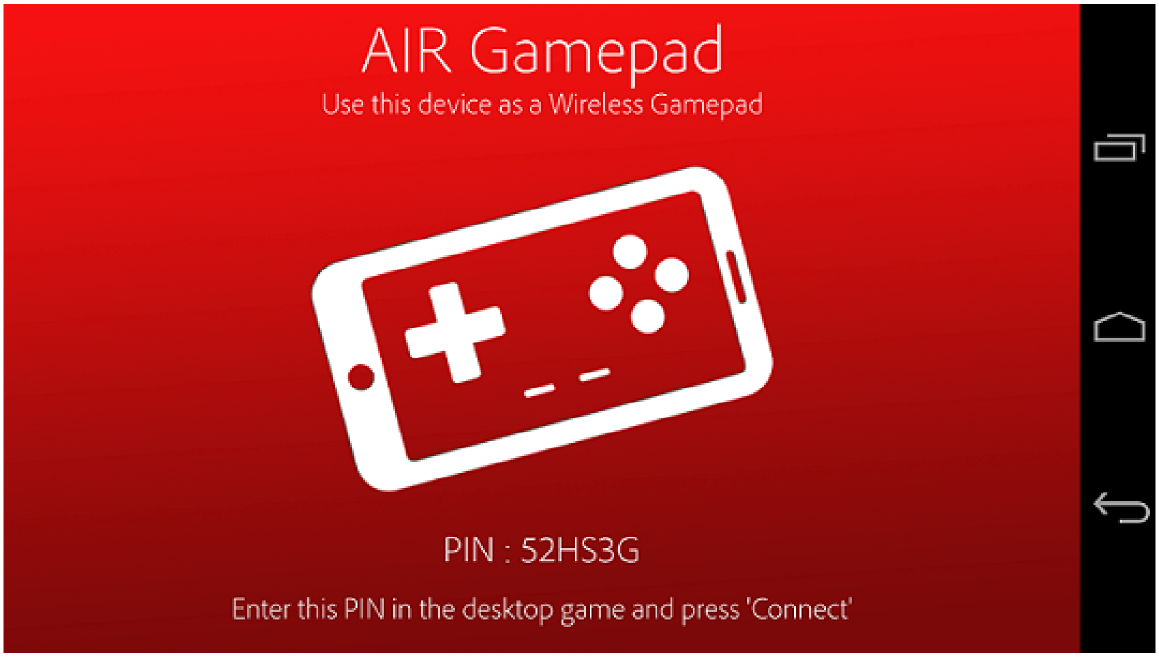
Install Adobe AIR app to use your Android device as a Wireless Gamepad

PIN :

Open AIR app and shake your device to get the PIN

Fig1. Message pop-up shown by Flash game on calling connect() when launched on browser.

Now install the Adobe AIR Runtime app on the Android device or if it installed then launch the Adobe AIR Runtime app and shake the device. AIR Gamepad screen shows up in AIR Runtime app with a PIN, enter this PIN in the pop-up shown by browser and connect. After connection is established skin is displayed on Android device and user can control the Flash based browser games from the Android device.



AIR Gamepad

Use this device as a Wireless Gamepad

PIN : 52HS3G

Enter this PIN in the desktop game and press 'Connect'

Fig2. The AIR Gamepad screen shown by AIR Runtime App on Android device.

Note:

- AIR Gamepad APIs allow only single device pairing.
- AIR Runtime.apk has to be installed on the Android device.
- Both the Android device and the Desktop machine need to be on the same internet network.
- Only Landscape RotatedLeft orientation is supported on the Android Device.

- The skin which will be applied through applySkin() has to be in JPEG format.

Fixed Issues

Flash Player

- Multiple stability and security issues
- [GPU] Resolved an issue injected in Flash Player 14 where the specified Depth Buffer was ignored on a subset of WinXP-era GPUs whic h lacked support for Multiple Render Targets. (3776790)
- [Chrome] Improved camera capture performance in PPAPI Flash Player for Windows (3730085)

AIR

- Multiple stability fixes

Known Issues

- PPAPI Debugger - Chrome incorrectly reports the installed location of the plugin in chrome:plugins
- In Chrome, Flash Player can sometimes crash if playing DRM content encrypted with a rotating IV (Initialation Vector)

Build Changes

Runtime Versions

Flash Player Desktop: **14.0.0.160**
AIR Runtime Desktop: **14.0.0.166**
AIR Runtime Android: **14.0.0.166**
AIR SDK & Compiler: **14.0.0.166**

Authoring

Authoring for Flash Player 14

To use the new Flash Player, you will need to target SWF version 25 by passing in an extra compiler argument to the ASC 2.0 compiler: -swf-version=25. Directions are below:

- Download the new playerglobal.swc for Flash Player 14
- Download and install Flash Builder 4.7 from Creative Cloud: -https://www.adobe.com/products/gaming/tools.html
- Backup the existing AIR SDK if you need to restore it later then replace the bundled AIR SDK with the AIR 13 SDK. To do this, unzip the AIR 13 SDK to this location:\
  - MacOS: /Applications/Adobe Flash Builder 4.7/eclipse/plugins/com.adobe.flash.compiler_4.7.0.348297/AIRSDK
  - Windows: C:\Program Files\Adobe\Adobe Flash Builder 4.7 (64 Bit)\eclipse\plugins\com.adobe.flash.compiler_4.7.0.349722\AIRSDK\
- In Flash Builder, create a new project: File -> New -> project.\
- Open the project Properties panel (right-click and chose 'Properties'). Select\ ActionScriptCompiler from the list on the left.
- Add to the 'Additional compiler arguments' input: -swf-version=25. This ensures the outputted SWF targets SWF version 25. If you compile on the command-line and not in Flash Builder, you need to add the same compiler argument.

Authoring for AIR 14 Update to the AIR 14 namespace

You must update your application descriptor file to the 14 namespace in order to access the new AIR 14 API's and behavior. If your application does not require the new AIR 14 API's and behavior, you are not required to update the namespace. However, we recommend all users start using the AIR 14 namespace even if you are not yet taking advantage of the new 14 capabilities. To update the namespace, change the xmlns attribute in your application descriptor to: <application xmlns="http://ns.adobe.com/air/application/ 14.0">

System Requirements

For current Flash Player system requirements visit - http://www.adobe.com/products/flashplayer/systemreqs/
For current AIR system requirements, visit - http://www.adobe.com/products/air/systemreqs/

Flash Player 14 has the following minimum system requirements:

-	Windows	Macintosh
Processor	2.33 Ghz or faster x86-compatible processor, or Intel® Atom™ 1.6GHz or faster processor for netbook class devices{-}	Intel® Core™ Duo 1.83GHz or faster processor
Operating System	Microsoft® Windows® XP (32-bit), Windows Server 2008 (32-bit), Windows Vista® (32-bit), Windows 7 (32-bit and 64-bit), Windows 8 (32 bit and 64 bit), or Windows Server 2012 (64 bit)	Mac OS® X v10.6, v10.7, v10.8, or v10.9
Browser	Internet Explorer 8.0 and above, Mozilla Firefox 17 and above, Google Chrome, Opera 11	Safari 5.0 and above, Mozilla Firefox 17.0 and above, Google Chrome, Opera 11
Memory	512MB of RAM (1GB RAM recommended for netbook class devices), 512MB of graphics memory	256MB of RAM, 512MB of graphics memory

AIR 14 has the following minimum system requirements:

-	Windows	Macintosh	Android	iOS
Processor / Device Hardware	2.33GHz or faster x86-compatible processor or Intel Atom™ 1.6GHz or faster processor for netbook class devices	Intel® Core™ Duo 1.83GHz or faster processor	ARMv7 processor with Vector FPU, Minimum 550MHz, OpenGL ES 2.0, H.264 & AAC H/W Decoders	iPod touch 4, iPhone 4, iPhone 5, iPad 2, iPad mini or iPad with Retina display
Operating System	Microsoft® Windows® XP, Windows Server® 2008, Windows Vista® Home Premium, Business, Ultimate, or Enterprise (including 64-bit editions) with Service Pack 2, Windows 7, or Windows 8 Classic	Mac OS® X v10.6, v10.7, v10.8, or v10.9	Android 2.3 and above	iOS 6.1 and above
RAM	512MB of RAM (1GB recommended)	512MB of RAM (1GB recommended)	256MB RAM	