Welcome to Adobe® Flash® Player 12 and Adobe AIR 4!

Last Updated: January 21, 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 12 and AIR 4. This document may be updated periodically as more information becomes available.

With this release, we are introducing a new numbering scheme for our product versions. Adopting the pattern set by Google with Chrome and Mozilla with Firefox, we will simply update the major version number with each subsequent release. In other words, beginning with the release of "Jones", Flash Player will become Flash Player 12. With each new major release, roughly every 3 months, that number will increase by one.

This change will also apply to AIR and the AIR SDK, albeit not right away. Our "Jones" release will be numbered AIR 4 and AIR SDK 4; however, with our "King" (Q2 2014) release, the version number will be synchronized with the Flash Player version at 13.

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.

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

- **iOS - Improved Packaging Engine** – We're very excited about this new feature. The new packaging engine that we're working on can improve iOS packaging time up to 10 times over the current packager! However this feature is still early in development and we need your feedback. 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":
  ```
  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
  ```

  Please see the Known Bugs section below for issues impacting this feature

- **Support for Android 4.4 (KitKat)** - We've completed our support testing with AIR against the latest Android 4.4 OS release. Please let
us know if you encounter any problems.

- **Mac .pkg Installation Support** – Deploying Flash Player and keeping it up to date is a critical task for system administrators worldwide. In Flash Player 11.9 we added .pkg support for ease of deployment. In Flash Player 12 we’ve incorporated public beta feedback from system administrators to improve the workflow and UI.

- **Internet Explorer 11 on Windows 7 Support** - Microsoft has released IE 11 on Windows 7 and we plan to officially support this in Flash Player 12.

- **Safe Mode in Safari 6.1 and higher** - Safari 6.1 includes an updated version of Safari with Safe Mode, which implements new security restrictions on the operation of browser plug-ins. To the extent possible, we’ve adapted Flash Player to transparently handle these changes. A small number of behaviors will change in the context of Safe Mode and are explained in the Known Issues section below.

- **64-Bit PPAPI Flash Player for Chrome** - We'd like to encourage those interested in 64-bit Flash Player for Chrome to please download the latest build available in the Chrome Canary Channel.

- **Graphics: Buffer Usage flag for Stage3D** - We’ve added a new enum Class called Context3DBufferUsage which defines two constants, STATIC_DRAW (default) and DYNAMIC_DRAW. These constants specify how buffers will be used in Stage3D. We’ve also added a new parameter, called “bufferUsage”, to the constructors of VertexBuffer and IndexBuffer. This “bufferUsage” parameter uses one of the constants form Context3DBufferUsage, which will allow you to select the appropriate buffer type according to your needs.

- **Android Workers** - Introduced as a beta feature in AIR 3.9, we’re continuing to improve this feature based on your feedback in preparation for an official release in AIR.

- **Android - Support for native resources access by R* mechanism in native extension** - Currently, to use the native Android resources in the Android Native Extension one has to use getResourceID() API while typically to access the resource IDs developers use the R.* mechanism. AIR 4.0 onwards, apps developers will be able to access the resources by R.* mechanism. All the dependencies need to be specified in platform.xml as following and all the dependencies and resources to be packaged in the ANE.

```xml
<platform xmlns="http://ns.adobe.com/air/extension/4.0">
  <packagedDependencies>
    <packagedDependency>sample-lib-method.jar</packagedDependency>
    <packagedDependency>sample-lib-services.jar</packagedDependency>
  </packagedDependencies>
  <packagedResources>
    <packagedResource>
      <packageName>com.myane.sampleasextension</packageName>
      <folderName>ane-res</folderName>
    </packagedResource>
    <packagedResource>
      <packageName>com.example.lib</packageName>
      <folderName>lib-res</folderName>
    </packagedResource>
  </packagedResources>
</platform>
```

Sample ANE packaging command, the folder Android-ARM contains all the jars and the resources of ane and the dependent jars.

```bash
bin/adt -package -target ane sample.ane extension.xml -swc sampleane.swc -platform Android-ARM -platformoptions platform.xml -C Android-ARM
```

Note:

- The platform.xml, extention.xml and application descriptor should have namespace 4.0 or above.
• The ANE project should be a library project for R.* resource access mechanism to work

• **Supplementary Characters Enhancement Support for TextField – EXTENDED BETA** – This is a desktop enhancement for supporting surrogate pairs in the TextField control. Now, characters out of the Basic Multilingual Plane (BMP) with Unicode code points between U+10000 and U+10FFFF will work correctly in the TextField control. It greatly enlarges the code point range we support and includes characters like emotion symbols (emoticons) and complex CCJK characters. This feature is being introduced in the Flash Player 12 beta but will go live in a subsequent release. Due to the sensitive nature of text display, we would like an extended test period to ensure no bugs are introduced.

• **Stage3D Creation of Context3D with Profile Array** - We've added a new interface to

  Stage3D.requestContext3DMatchingProfiles(profiles:Vector.<String> ) which will create a Context3D with highest level suitable profile that is in profile array, based on the current hardware. A developer can check the newly added property 'profile' to obtain the current profile that was chosen by the Flash Runtime.

**Known Bugs:**

- 3679802 - Unable to compile flex App with debug swf using -useLegacyAOT switch on windows
- 3671999 - Wifi/USB debugging session gets started using fdb but unable to set breakpoint as app gets launched immediately after session gets started.
- 3672804 - Compilation of PI app is getting failed.Adt throws the "Duplicate symbol for architecture armv7" error.

**Build Changes**

**Runtime Versions**

Flash Player Desktop only: 12.0.0.54
AIR Runtime Desktop: update coming soon!
AIR Runtime Android: update coming soon!
AIR SDK & Compiler: update coming soon!

**Authoring**

**Authoring for Flash Player 12.0**

To use the new Flash Player, you will need to target SWF version 23 by passing in an extra compiler argument to the ASC 2.0 compiler:

```
-swf-version=23.
```

Directions are below:

- Download the new playerglobal.swc for Flash Player 12.0
- Backup the existing AIR SDK if you need to restore it later then replace the bundled AIR SDK with the AIR 4.0 SDK. To do this, unzip the AIR 4.0 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\n- 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=23’. This ensures the outputted SWF targets SWF version 23. If you compile on the command-line and not in Flash Builder, you need to add the same compiler argument.

**Authoring for AIR 4.0 Update to the AIR 4.0 namespace**

You must update your application descriptor file to the 4.0 namespace in order to access the new AIR 4.0 API's and behavior. If your application does not require the new AIR 4.0 API's and behavior, you are not required to update the namespace. However, we recommend all users start using the AIR 4.0 namespace even if you are not yet taking advantage of the new 4.0 capabilities. To update the namespace, change the xmlns attribute in your application descriptor to: `<application xmlns="http://ns.adobe.com/air/application/4.0">`

**System Requirements**

Flash Player 12.0 has the following minimum system requirements:

<table>
<thead>
<tr>
<th></th>
<th>Windows</th>
<th>Macintosh</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Processor</strong></td>
<td>2.33 Ghz or faster x86-compatible processor, or Intel® Atom™ 1.6GHz or faster processor for netbook class devices{-}</td>
<td>Intel® Core™ Duo 1.83GHz or faster processor</td>
</tr>
<tr>
<td><strong>Operating System</strong></td>
<td>Microsoft® Windows® XP (32-bit), Windows Server 2008 (32-bit), Windows Vista® (32-bit), Windows 7 (32-bit and 64-bit)</td>
<td>Mac OS® X 10.6, Mac OS X 10.7, Mac OS X 10.8</td>
</tr>
<tr>
<td><strong>Browser</strong></td>
<td>Internet Explorer 8.0 and above, Mozilla Firefox 17.0 and above, Google Chrome, Opera 11</td>
<td>Safari 5.0 and above, Mozilla Firefox 17.0 and above, Google Chrome, Opera 11</td>
</tr>
<tr>
<td><strong>Memory</strong></td>
<td>512MB of RAM (1GB RAM recommended for netbook class devices), 512MB of graphics memory</td>
<td>256MB of RAM, 512MB of graphics memory</td>
</tr>
</tbody>
</table>

AIR 4.0 has the following minimum system requirements:

<table>
<thead>
<tr>
<th></th>
<th>Windows</th>
<th>Macintosh</th>
<th>Android</th>
<th>iOS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Processor / Device Hardware</strong></td>
<td>2.33GHz or faster x86-compatible processor or Intel Atom™ 1.6GHz or faster processor for netbook class devices</td>
<td>Intel® Core™ Duo 1.83GHz or faster processor</td>
<td>ARMv7 processor with Vector FPU, Minimum 550MHz, ES2.0, H.264 &amp; AAC H/W Decoders</td>
<td>iPod touch 4, iPod touch 5, iPhone 3GS, iPhone 4/4S, iPad, iPad 2, iPhone 5, The New iPad, iPad mini</td>
</tr>
<tr>
<td><strong>Operating System</strong></td>
<td>Microsoft® Windows® XP, Windows Server® 2003, Windows Server® 2008, Windows Vista® Home Premium, Business, Ultimate, or Enterprise (including 64-bit editions) with Service Pack 2, or Windows 7{-}</td>
<td>Mac OS® X 10.6 and 10.7</td>
<td>Android 2.3 and above</td>
<td>iOS 5.1 and above</td>
</tr>
<tr>
<td><strong>RAM</strong></td>
<td>512MB of RAM (1GB recommended)</td>
<td>512MB of RAM (1GB recommended)</td>
<td>256MB RAM</td>
<td></td>
</tr>
</tbody>
</table>