Welcome to Adobe Flash Player 32 and Adobe AIR 32!

Last Updated: Sep 24, 2019

Welcome to the latest Flash Runtime version beta! We've been hard at work adding new features to Flash Player and AIR and are looking forward to feedback from our development and Flash Player community.

This beta release includes new features as well as enhancements and bug fixes related to security, stability, performance, and device compatibility for Flash Player 32 and AIR 32. 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
AIR Beta Forum
Beta ActionScript reference
Bug Database

NOTES:

- The ActiveX Flash Player in this release is not compatible with Windows 8.x or 10
- Flash Player for Windows 8.x/10 is available as part of the generally available Windows 8.x/10 update

New and updated features

AIR: Audio/Video playback enhancements

AIR audio/video playback has been enhanced to use system capabilities. We have made significant changes and would like to hear your feedback. The changes impact all platforms, in case you encounter issues with your application with this beta kindly report them with details of impacted application.

iOS SDK Upgrade

AIR Runtime is now built with iOS 12.1 SDK, which enables AIR developers to use ANEs built with iOS 12.1 APIs without using the platformSDK switch while packaging with ADT.

Geolocation Enhancement in iOS Applications

Until AIR SDK 31, when an application runs in the background, access to geolocation services is paused after some time. This is as designed behavior implemented by Apple. When the services are paused it is the users responsibility to enable the geolocation services by re-launching the application.

Native iOS application developers can set the flag pausesLocationUpdatesAutomatically as false which helps in keeping navigation services running in the background. Since continuous running can cause battery drain, native iOS flag desiredAccuracy permits the developer to specify the accuracy of the geolocation data required by the app.

Starting AIR SDK 32, two new properties have been defined on the Geolocation object to allow users to replicate the native iOS behavior. The properties are:

pausesLocationUpdatesAutomatically: This would allow application developers to choose if they want to keep the geolocation services active when the application is in the background. The value of this property is set to true by default. Setting it to false would keep the geolocation services running in the background. Setting it to true will replicate default native iOS behavior of pausing the location services after some period of inactivity.

desiredAccuracy: This property allows application developers to specify the accuracy of the geolocation data to prevent drainage of battery with the long continuous usage of geolocation services. It can be set to:
Geolocation.LOCATION_ACCURACY_BEST_FOR_NAVIGATION: for the highest possible accuracy that uses additional sensor data to facilitate navigation apps.

Geolocation.LOCATION_ACCURACY_BEST: for the best level of accuracy available.

Geolocation.LOCATION_ACCURACY_NEAREST_TEN_METERS: for the accuracy of within ten meters of the desired target.

Geolocation.LOCATION_ACCURACY_HUNDRED_METERS: for the accuracy of within one hundred meters of the desired target.

Geolocation.LOCATION_ACCURACY_KILOMETER: for accuracy to the nearest kilometer.

Geolocation.LOCATION_ACCURACY_THREE_KILOMETERS: for accuracy to the nearest three kilometers.

The properties can only be set after geolocation permission is requested. Also, the following key needs to be set in the application XML to allow location access to the application when running in the background.

```xml
<key>UIBackgroundModes</key>
<array>
<string>location</string>
</array>
```

The properties are introduced for iOS, Android remains unaffected by this change.

### Launch images for iPhone XS, iPhone XS Max, and iPhone XR

With AIR SDK version 32, support of launch images for iPhone XS, iPhone XS Max and iPhone XR is added. Refer the table below for launch image names and dimensions:

<table>
<thead>
<tr>
<th>Devices</th>
<th>Resolution (pixels)</th>
<th>Launch image name</th>
<th>Orientation</th>
</tr>
</thead>
<tbody>
<tr>
<td>iPhone X, XS</td>
<td>1125x2436</td>
<td>Default-812h@3x~iphone.png</td>
<td>Portrait</td>
</tr>
<tr>
<td>iPhone X, XS</td>
<td>2436x1125</td>
<td>Default-Landscape-812h@3x~iphone.png</td>
<td>Landscape</td>
</tr>
<tr>
<td>iPhone XR</td>
<td>828x1792</td>
<td>Default-896h@2x~iphone.png</td>
<td>Portrait</td>
</tr>
<tr>
<td>iPhone XR</td>
<td>1792x828</td>
<td>Default-Landscape-896h@2x~iphone.png</td>
<td>Landscape</td>
</tr>
<tr>
<td>iPhone XS Max</td>
<td>1242x2688</td>
<td>Default-414w-896h@3x~iphone.png</td>
<td>Portrait</td>
</tr>
<tr>
<td>iPhone XS Max</td>
<td>2688x1242</td>
<td>Default-Landscape-414w-896h@3x~iphone.png</td>
<td>Landscape</td>
</tr>
</tbody>
</table>

### Addition of property cpuAddressSize in Capabilities class

Starting AIR and Flash Player version 32, a new property `cpuAddressSize` is added to flash.system.Capabilities class. The `cpuAddressSize` specifies whether the Flash Player instance or an AIR application running on the system is 32-bit or 64-bit. This read only property returns value 32 or 64 for 32-bit and 64 for 64-bit applications respectively.

### Runtime Versions

- Flash Player: 32.0.0.269
- AIR Runtime: 32.0.0.144
- AIR SDK & Compiler: 32.0.0.144

### Known Issues

#### May 23, 2019

**AIR**

- Audio has slight distortion on playback of few Audio/Video files

#### Jan 25, 2019

**Flash Player**

- None
AIR

- iOS SDK version warning on submitting air apps to app store (AIR-4198787)
- Unable to install app packaged for simulator target from terminal (FLASH-4191010)
- Swift 4.2 dylibs are not signed correctly when packaged with AIR app (AIR-4198777)

Nov 21, 2018

Flash Player
None

AIR

- contentsScaleFactor can only have one value on multi-window app on Windows 10 (AIR-4198669)
- dx tool failed (com.android.dx.cf.code.SimException) unable to build ane with java8 (AIR-4198743)

Fixed Issues

Flash Player
Sept 24, 2019
- Security fixes

Flash Player
June 17, 2019
- Security fixes

AIR

Mar 15, 2019
- ld error on Windows when using platformSDK option (AIR-4198796)

Flash Player
Feb 20, 2019
- Security fixes

AIR

Feb 13, 2019
- iOS SDK version warning on submitting air apps to app store (AIR-4198787)
- Unable to install app packaged for simulator target from terminal (FLASH-4191010)

Jan 25, 2019

Flash Player
- Security fixes

AIR

- Add launch images for 2018 models of iPad Pro 11" and 12.9" (AIR-4198773)
- SecureSocket can’t connect when hostname only matches Subject Alternative Names (AIR-4198762)
- TouchEvent systems stops sending TOUCH_END events to air app on iPhoneX (AIR-4198768)
- iOS AIR Application produces error with new LD upgrade (FLASH-4190918)

Dec 10, 2018

Flash Player
- Security fixes

Nov 21, 2018

Flash Player
- Stability fixes

AIR
• Client (AIR Android) fails to make rtmps connection with the server if TLS 1.0 is disabled on the server (FLASH-4190642)
• Launch Image name mapping for iPhone XS and iPhone XR (FLASH-4190711)
• Installing on latest iOS devices through ADT fails (XS, XS max) (AIR-4198748)
• Sound: Custom sampling with shared sound source creates crackling artifacts (AIR-4198737)
• Multi-touch in Win10 experiencing performance hit (FLASH-4085538)
• [iOS 12] Keyboard isn’t coming at its default position when packed using platformsdk option (FLASH-4190759)
• AIR application quits unexpectedly when playing video loops for many hours (FLASH-4190569)

**Authoring for Flash Player 32 and AIR 32**

• Update application descriptor namespace to 32
• SWF version should be 43

**System Requirements**

For system requirements of the current release of AIR in production, visit [http://www.adobe.com/products/air/systemreqs/](http://www.adobe.com/products/air/systemreqs/)