Mocean Android SDK

Developer Guide

For Android SDK Version 3.0 Alpha

***NOTE: This 3.0 Alpha release of the Mocean AD SDK is intended to provide developers with early access to the upcoming MRAID 2.0 based SDK. The specification, this code, and the overall feature set is not yet final. Comments and feedback are welcome.***

***YOU MUST NOT DELIVER PRODUCTS BASED ON THIS ALPHA SDK!***

***For more information, please visit:  
 http://developer.moceanmobile.com/SDK\_Pre-release\_Pending\_Changing***

Table of Contents

[Overview 3](#_Toc335998494)

[What’s new in 3.0 Alpha: 3](#_Toc335998495)

[How to upgrade from previous versions: 4](#_Toc335998496)

[System requirements: 4](#_Toc335998497)

[SDK contents: 4](#_Toc335998498)

[Installation instructions: 4](#_Toc335998499)

[Sample usage 9](#_Toc335998500)

## Overview

***NOTE: This 3.0 Alpha release of the Mocean AD SDK is intended to provide developers with early access to the upcoming MRAID 2.0 based SDK. The specification, this code, and the overall feature set is not yet final. Comments and feedback are welcome.***

***YOU MUST NOT DELIVER PRODUCTS BASED ON THIS ALPHA SDK!***

***For more information, please visit:  
 http://developer.moceanmobile.com/SDK\_Pre-release\_Pending\_Changing***

This document provides a description of the following:

* What’s new in 3.0 Alpha
* How to upgrade from previous versions
* System requirements
* Installation Instructions (revised)
* SDK contents
* SDK API reference
* Sample of usage

### What’s new in 3.0 Alpha:

* The SDK has been redesigned and refocused around MRIAD 2.0 (DRAFT) Specification (see <http://www.iab.net/mraid/>). Other parts of the API have been streamlined and cleaned up to support this focus.
* ORMMA support has been dropped.
* Callback interfaces have been expanded and renamed. See the new MASTAdDelegate class documentation.
* Ad request parameter support has been streamlined and simplified. See the new MASTAdRequest class documentation.
* All public methods intended for SDK users now have javadoc documentation. HTML versions of this are included in the Documentation\javadoc folder of the SDK release, replacing the version formerly included in this word document.

### How to upgrade from previous versions:

1. Use the new MASTAdRequest class to set properties sent to the back-end when requesting ads. For example, to change the zone for an existing ad view object named “adView”, the new code looks as follows:

|  |
| --- |
| int newZone = 1234; // Sample, use a zone obtained from your Mocean account rep.  adview.getAdRequest().setProperty(MASTAdRequest.*parameter\_zone*, newZone); |

Only a small set of key parameters exist as named parameters in this version. Any others can be set via the custom\_parameters map object. The full set of available ad request parameters supported by the Mocean back-end are documented online at:  
 <http://developer.moceanmobile.com/Mocean_Ad_Request_API>

1. Use the new MASTAdDelegate class for your callback/listeners. The following table shows the old and new names for each previous interface and the methods within those interfaces:

|  |  |
| --- | --- |
| MASTOnAdClickListener   * click() | [MASTAdDelegate.AdClickEventHandler](file:///C:\Users\mwalker\Documents\Eclipse%20Projects\MASTAdView\Documentation\javadoc\com\MASTAdView\MASTAdDelegate.AdClickEventHandler.html)   * onClickEvent() |
| MASTOnAdDownload   * begin() * end() * error() | [MASTAdDelegate.AdDownloadEventHandler](file:///C:\Users\mwalker\Documents\Eclipse%20Projects\MASTAdView\Documentation\javadoc\com\MASTAdView\MASTAdDelegate.AdDownloadEventHandler.html)   * onDownloadbegin() * onDownloadError() * onDownloadError() |
| MASTOnOrmmaListener   * event() | ORMMA dropped, replaced with: [MASTAdDelegate.MraidEventHandler](file:///C:\Users\mwalker\Documents\Eclipse%20Projects\MASTAdView\Documentation\javadoc\com\MASTAdView\MASTAdDelegate.MraidEventHandler.html)   * onMraidEvent() |
| MASTOnThirdPartyRequest   * event() | [MASTAdDelegate.ThirdPartyEventHandler](file:///C:\Users\mwalker\Documents\Eclipse%20Projects\MASTAdView\Documentation\javadoc\com\MASTAdView\MASTAdDelegate.ThirdPartyEventHandler.html)   * onThirdPartyEvent() |
| MASTOnActivityHandler   * onAttachedToActivity() * onDetachedFromActivity() | [MASTAdDelegate.AdActivityEventHandler](file:///C:\Users\mwalker\Documents\Eclipse%20Projects\MASTAdView\Documentation\javadoc\com\MASTAdView\MASTAdDelegate.AdActivityEventHandler.html)   * onAdAttachedToActivity() * onAdDetachedFromActivity() |

### System requirements:

* Android SDK (API level 8, platform version 2.2 or later)
* Eclipse 3.5 or later
* 10 Mb free disk space

### SDK contents:

* Lib - SDK library files
* Sample - sample usage/test app

### Installation instructions:

*Installing Android SDK*

Download and install Android SDK (http://developer.android.com/sdk/index.html).

Follow installation instructions provided by Android SDK. Once SDK has been installed follow to the next step to install Android mOcean SDK.

*Installing Android mOcean SDK*

The SDK is distributed as a library source code project, but still includes a pre-compiled jar library. To add the SDK to a project, the developer must configure the project properties to indicate the location of SDK files, as well as the names of library dependencies.

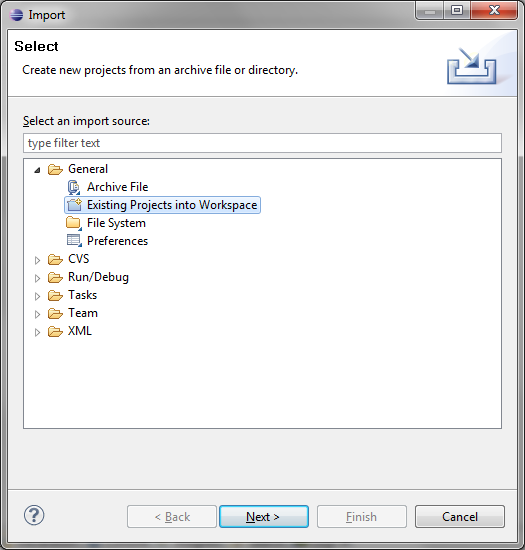
1. Unpack the SDK zip file into a convenient location in your source code working area.
2. Open or create a new Android project in the Eclipse development environment.
3. Import the SDK project into your workspace as an existing Android project.
   1. Choose Import from the File menu, then Existing Projects into Workspace as show in Figure 1 below.  
        
      

Figure 1- Import Existing Project

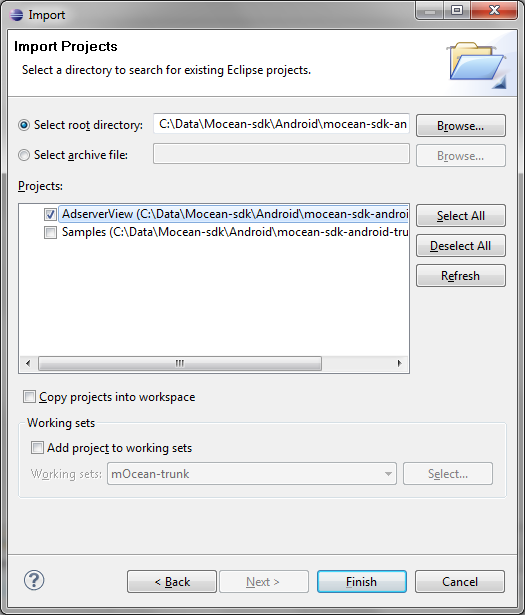
* 1. Browse to the location where you unpacked the SDK file and import the AdserverView project; you can also optionally import the Samples project if you want to work with the SDK sample application. See Figure 2 below for an example.  
       
     

Figure 2 - Import Projects into SDK

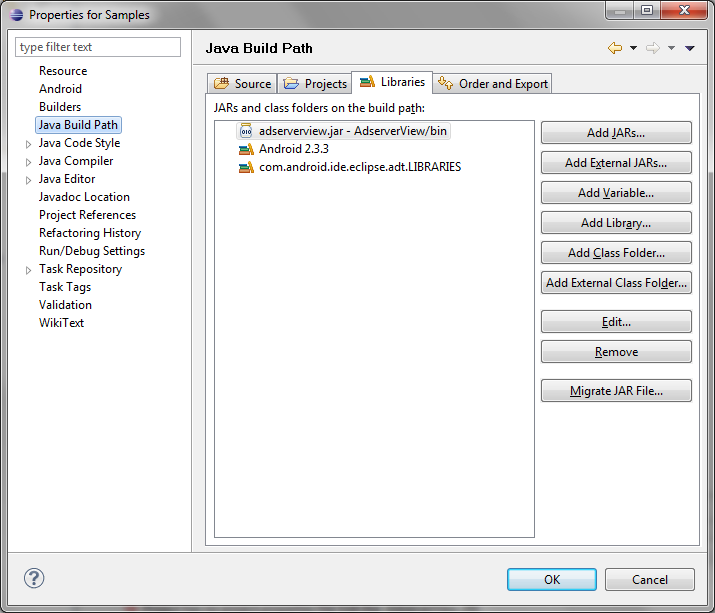
* 1. Choose Properties from the Project menu, and then select the Java Build Path category followed by the Libraries tab, as shown in Figure 3 below.  
       
     

Figure 3 - Add Library Jar

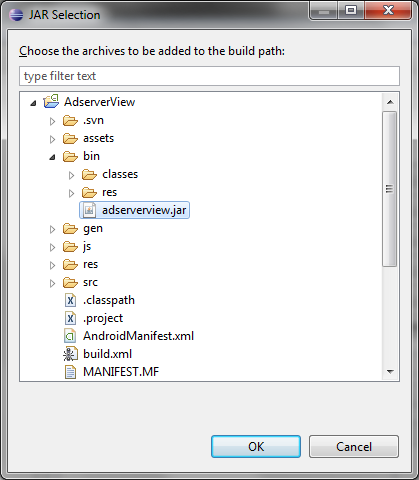
* 1. Choose the Add Jar button, and then navigate into the bin folder of the MASTAdView project and choose the mastadview.jar Jar file as shown in Figure 4 below.  
       
     

Figure 4 - Ad SDK Jar File

* 1. **IMPORTANT**: If using release 18 or later of the Android SDK tools, choose the Order and Export tab, and check the box to export the SDK Jar file as shown in Figure 5 below.   
       
     ***Without this, applications will compile but the resulting apk file will not include the required SDK code and the app will crash at runtime due to missing symbols.***

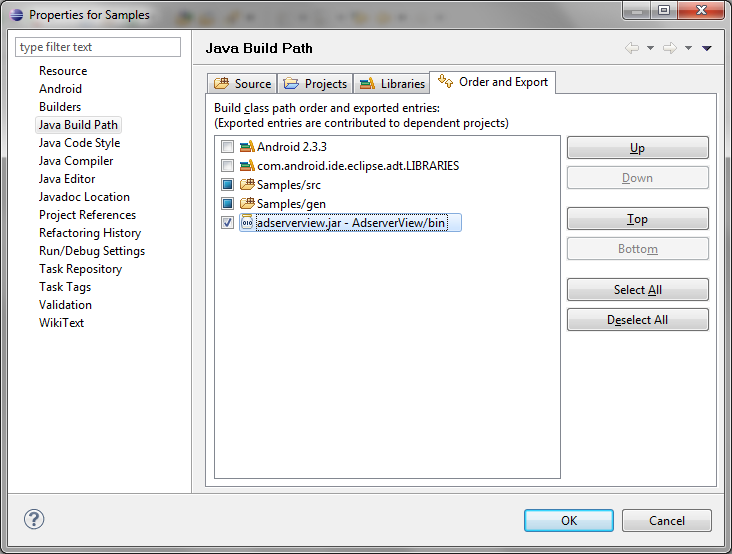


Figure 5 - Export SDK Jar File

*Updating the manifest file (AndroidManifest.xml)*

Add “minSdkVersion” parameter in project manifest file (AndroidManifest.xml)

Example: <uses-sdk android:minSdkVersion="5" />

Set the security permissions in your manifest file (AndroidManifest.xml). At a minimum you ***must*** add these permissions for the ad view to work:

|  |  |
| --- | --- |
| **Permission** | **Description & Manifest XML fragment** |
| *INTERNET* | Access the Internet. Required for ad content download.  <uses-permission android:name=*"android.permission.INTERNET"*></uses-permission> |
| *Network State* | Access the network state. Required for ad request parameter setting, and MRAID support.  <uses-permission android:name=*"android.permission.ACCESS\_NETWORK\_STATE"*></uses-permission> |

Depending on the ad content you display in your app, the following ***may*** also be needed:

|  |  |
| --- | --- |
| **Permission** | **Description & Manifest XML fragment** |
| *Fine Location* | Use GPS to obtain location information. Needed if SDK enables location detection; off by default.  <uses-permission android:name=*"android.permission.ACCESS\_FINE\_LOCATION"*></uses-permission> |
| *Phone State* | Read state of phone data connection. Required for ad request parameter setting.  <uses-permission android:name=*"android.permission.READ\_PHONE\_STATE"*></uses-permission> |
| *Read Calendar* | Read calendar events. Needed if MRAID ad makes use of calendar features.  <uses-permission android:name=*"android.permission.READ\_CALENDAR"*></uses-permission> |
| *Write Calendar* | Write calendar events. Needed if MRAID ad makes use of calendar features.  <uses-permission android:name=*"android.permission.WRITE\_CALENDAR"*></uses-permission> |
| *Call Phone* | Initiate a phone call. Needed if an ad makes use of the MRAID feature to place a phone call.  <uses-permission android:name=*"android.permission.CALL\_PHONE"*></uses-permission> |
| *Send SMS* | Send an SMS (text) message. Needed if an ad makes use of the MRAID feature to send a text message.  <uses-permission android:name=*"android.permission.SEND\_SMS"*></uses-permission> |
| *External Storage* | Access the SD card storage area. Required for debug logs, photo, and file access to support SDK logging and MRAID features.  <uses-permission android:name=*"android.permission.WRITE\_EXTERNAL\_STORAGE"*></uses-permission> |

## Sample usage

See the Sample project source code included with the SDK for a more complete example, and also consult the update Getting Started Guide available for download along with this SDK package (when available for the Alpha 3.0 SDK.)

To create view of advertising and to add it to the form, use one of the following two variations:

1. Dynamic creation: create ad view in code; for example:

|  |
| --- |
| MASTAdView adView = new MASTAdView(this, "5441", "9312");  adView.setLayoutParams(new  ViewGroup.LayoutParams(ViewGroup.LayoutParams.FILL\_PARENT, 100));  adserverView.update(); // fetch initial ad content  linearLayout.addView(adView); // insert ad view into layout |

*NOTE: the site and zone in this code are samples; you must use a valid site and zone obtained from your account representative in an actual product.*

2. Template creation: create ad view in an XML layout; for example:

|  |
| --- |
| <com.adserver.adview.MASTAdView  android:id="@+id/adViewer1"  android:layout\_width="fill\_parent"  android:layout\_height="100px"  site="5441"  zone="6365"  defaultImage="@drawable/test\_banner"  /> |

*NOTE: the site and zone in this code are samples; you must use a valid site and zone obtained from your account representative in an actual product.*

1. To show View of interstitial advertising, create view in code as shown in (1) above, and invoke the show() method (instead of adding the view to the layout manager), after calling update():

|  |
| --- |
| // create or obtain reference to view object…  adserverView.show(); |

-- END OF DOCUMENT --