RateBox Adobe AIR Native Extension for iOS and Android

http://www.milkmangames.com

For support, contact support@milkmangames.com

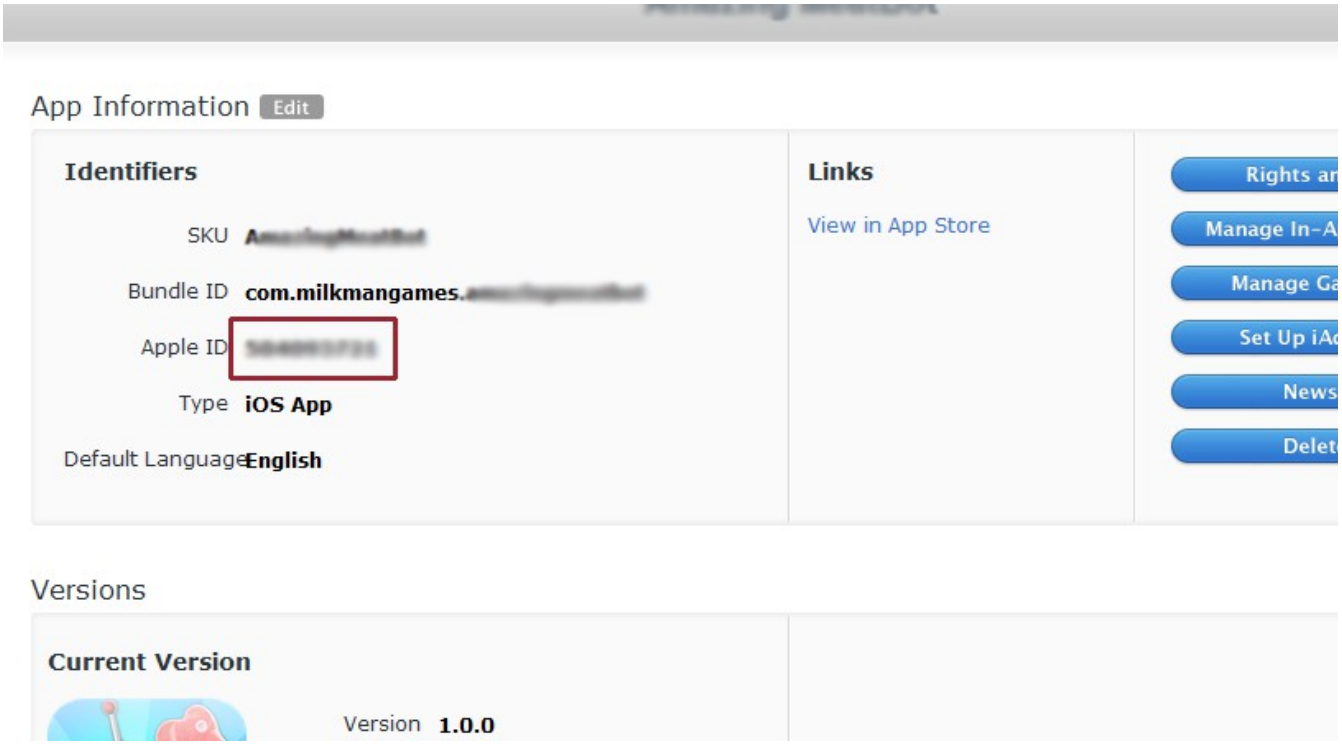
To View full AS3 documentation, see 'docs/as3docs/index.html'.
Review 'example/RateBoxExample.as' for a sample application.

Note that RateBoxExample.as is a Document Class. If you're a Flash Professional user and don't know how to use a document class, see the troubleshooting on "Using the Example Class in Flash CS6" at the end of this guide.

1. Get your iTunes App Apple ID from developer.apple.com

Before you can use RateBox for iOS apps, you'll need to retrieve your iTunes Apple App ID. (This a unique 9 digit string of numbers assigned to your app by Apple, and *not* the same as your application bundle ID from the AIR descriptor file.)

1. Go to http://www.developer.apple.com and log in with your developer account credentials.
2. Select 'iOS Dev Center', then 'iTunes Connect.'
3. Select 'Manage Applications'.
4. Locate and select your app. (If you have not yet registered your app in iTunes Connect, you can do so now.)
5. Retrieve your Apple ID from the App Information section. (You'll need this id to initialize the extension in Actionscript.) The illustration below shows the location of the iTunes App ID:



2. Include the RateBox Library

Add the RateBox library to your project.

In Flash Professional CS6 or Higher:

1. Create a new Android or iOS project
2. Choose File>Publish Settings...
3. Select the wrench icon next to 'Script' for 'ActionScript Settings'
4. Select the Library Path tab.
5. Click 'Browse for Native Extension (ANE) File' and select the com.milkmangames.extensions.RateBox.ane file. Press OK.
6. Select the wrench icon next to 'Target' for Player Settings
7. For Android targets, Select the 'Permissoins' tab, and enable 'INTERNET' and 'ACCESS_NETWORK_STATE'
8. Check the 'Manually manage permissions and manifest additions for this app' box.
9. Press OK

In Flash Builder 4.6 or Higher:

1. Go to *Project Properties*
2. Select *Native Extensions* under *Actionscript Build Path*
3. Choose *Add ANE...* and navigate to the com.milkmangames.extensions.RateBox.ane file
4. Select *Actionscript Build Packaging > Google Android* (or iOS for that target)
5. Select the *Native Extensions* tab, and click the '*Package*' checkbox next to the extension

3. Actionscript API Set-Up

The RateBox extension can be up and running in a few simple calls. See '**example/RateBoxExample.as**' for a full example.

1. Import the API Classes:

```
import com.milkmangames.nativeextensions.*;
import com.milkmangames.nativeextensions.events.*;
```

2. Initialize the RateBox API as early on in your application as possible (in the main constructor function of your document class, or at the beginning of the first frame in Flash Professional.) To set up the API, call the *RateBox.create()* method, with your iOS Apple ID (found in section 1, 'Get Your iTunes App Apple ID'), the title of the rating prompt, and the message for the rating prompt as parameters.

   Note: If you are supporting Android but not iOS, you can set the first parameter to an empty string ("").

   You can check RateBox.isSupported() first, to ensure that the current platform is supported by RateBox (either iOS or Android.):

```
if (RateBox.isSupported())
{
	RateBox.create("123456789","Rate My App","If you like this app, please rate it 5 stars!");
}
```

3. For testing purposes, you can enable Test Mode with a call to RateBox.rateBox.useTestMode(). While in test

mode, pressing the 'Rate Now' button will send the user to rate an existing sample app on the store. This can be useful for seeing the entire flow of events when your own app has not yet been published to The App Store or Google Play. ***Remember to remove the useTestMode() call before publishing your application!***

```
// causes rating prompts to redirect to an existing example app on the store,
// because your own app has not yet been published.  Remember to remove this
// line when you're doing testing and ready to publish!
RateBox.rateBox.useTestMode();
```

4. Call the RateBox.rateBox.onLaunch() method to signal the extension each time your app has launched:

```
// listen for 'invoke' events
flash.desktop.NativeApplication.nativeApplication.addEventListener(InvokeEvent.INVOKE,onInvoke);

function onInvoke(e:InvokeEvent):void
{
	RateBox.rateBox.onLaunch();
}
```

4. Advanced Options and Custom Events

By default, RateBox will display a prompt to the user after the app has been launched 3 times. RateBox automatically handles the user's preferences (if they rated the app already, choose to be reminded to rate later, or never to be asked again), and automatically detects when the app has been updated to a new version so it can reset its timer and request a new rating.

It can also track how many days the app has been installed for, and track custom in-app events to prompt a rating. (For instance, a custom event could be 'finishing a level' in a game, and you could prompt the user to rate after they've finished a level 10 times.)

The text displayed in the options buttons (For 'Rate Now', 'Remind me Later', and 'Don't Ask Again') can also be customized.

Setting Custom Options

The complete parameter list for RateBox.create() is as follows:

```
RateBox.create(iosAppId, title, message, rateNowLabel, declineLabel, neverAgainlabel, minLaunchesTilPrompt,
minEventsTilPrompt, minDaysTilPrompt, coolDownDays);
```

The table below explains each parameter:

Parameter	Description	Default Value
iosAppId (String)	The iTunes Apple App ID from iTunes Connect (refer to Section 1 for details)	(Required)
title (String)	The title that shows at the top of the rating prompt dialog box.	(Required)
message (String)	The message to display to the user in the rating prompt dialog box.	(Required)
rateNowLabel (String)	The label for the 'Rate Now' button that will appear in	"Rate Now"

Parameter	Description	Default Value
	the rating prompt dialog box.	
declineLabel (String)	The label for the 'Not Now' button that will appear in the rating prompt dialog box.	"Not Now"
neverAgainLabel (String)	The label for the 'Don't ask Again' button that will appear in the rating prompt dialog box.	"Don't ask again"
minLaunchesTilPrompt (int)	The number of times the app must be launched with RateBox.rateBox.onLaunch() before the user will be prompted for a rating.	3
minEventsTilPrompt (int)	The number of times a custom invent must be triggered with RateBox.rateBox.incrementEventCount() before the user will be prompted for a rating.	0
minDaysTilPrompt (int)	The number of days the app must be installed for before the the user will be prompted for a rating.	0
coolDownDays (int)	The number of days the extension will wait before asking the user to rate again, if they choose "Not Now" in the dialog box.	1

Custom Option Examples

The following examples show different ways to customize the rating prompt display rules when initializing RateBox:

```
// The following example will display the RateBox after a custom event has been triggered 3 times.
RateBox.create("123456","Rate This App","If you like this app, please rate it!","Rate Now","Ask Me
Later","Don't ask again",0,3,0);


// this would show the rating prompt after the app has been installed for 3 days:
RateBox.create("123456","Rate This App","If you like this app, please rate it!","Rate Now","Ask Me
Later","Don't ask again",0,0,3);

// this would show the rate prompt after the app has been launched 3 times:
RateBox.create("123456","Rate This App","If you like this app, please rate it!","Rate Now","Ask Me
Later","Don't ask again",3,0,0);


// you can mix and match these values if you want.  this would show the rate box if the app has been
launched at least 3 times, and been installed for 5 days.
RateBox.create("123456","Rate This App","If you like this app, please rate it!","Rate Now","Ask Me
Later","Don't ask again",3,0,5);
```

Setting Custom App Event Triggers

If you set the 'minEventsTilPrompt' parameter as described above, RateBox can track custom in-app events to prompt a rating. (For instance, a custom event could be 'finishing a level' in a game, and you could prompt the user to rate after they've finished a level 10 times.)

The following example increments RateBox's internal counter of custom events. If the counter has been reached, the rating prompt will be displayed:

```
function onFinishedLevel():void
{
	RateBox.rateBox.incrementEventCount();
```

```
}
```

Targeting the Amazon App Store

When your app is deployed on a Kindle Fire, the extension will automatically use the Amazon App Store to display the ratings page. When deployed ona Google Android device, the extension will use the Google Play store by default.

However, if you are making a build specifically for the Amazon App Store, but including regular (non-Kindle) Android devices, you should use the useAmazonAppStore() method to force the app to go to the Amazon App Store for ratings instead:

```
// call this ONLY if you're uploading to the Amazon App Store (not Google Play.  Ignored on iOS.)
RateBox.rateBox.useAmazonAppStore();
```

5. Listening for Event Callbacks

RateBox dispatches Actionscript events RateBoxEvent.PROMPT_DISPLAYED when the rating dialog prompt has been displayed to the user, and RateBoxEvent.RATE_SELECTED, RateBoxEvent.LATER_SELECTED, and RateBoxEvent.NEVER_SELECTED when the user selects one of the button actions inside the prompt dialog.

If the device does not have an active internet connection, RateBox will not display the prompt dialog, as there will be no way to access the app store and apply the rating. In this case, a RateBoxEvent.NETWORK_UNAVAILABLE event will be dispatched.

The following example demonstrates setting listener functions for each of the RateBoxEvents:

```
// optional event listeners- dispatched when the user has clicked a button in the rating
dialogRateBox.rateBox.addEventListener(RateBoxEvent.RATE_SELECTED,onDidRate);
RateBox.rateBox.addEventListener(RateBoxEvent.LATER_SELECTED,onDeclinedToRate);
RateBox.rateBox.addEventListener(RateBoxEvent.NEVER_SELECTED,onWillNeverRate);

// optional event listeners- dispatched when the rating prompt is shown,
// or can't be shown due to a network issue
RateBox.rateBox.addEventListener(RateBoxEvent.NETWORK_UNAVAILABLE,onRateNotDisplayed);
RateBox.rateBox.addEventListener(RateBoxEvent.PROMPT_DISPLAYED,onRateDisplayed);
```

6. Handling Rating Prompts Manually

The RateBox extension automatically handles tracking of installation time, launches, and custom events for ratings. However, you can also implement your own custom rating rules if you wish.

By default, RateBox will display prompts automatically when the event conditions are met. You can disable this functionality by calling the setAutoPrompt() function:

```
// prevent RateBox from showing the rating prompt dialog automatically
RateBox.rateBox.setAutoPrompt(false);
```

When autoPrompt is set to false, you can still check if RateBox's internal rating conditions have been met (number of days installed, number of launches, etc.) by calling areRatingConditionsMet():

```
if(RateBox.rateBox.areRatingConditionsMet())
{
	RateBox.rateBox.showRatingPrompt("Rate My App","Please Rate this App!!");
}
```

You may determine if the current version has been rated yet using the didRateCurrentVersion() function. In conjunction with the showRatingPrompt() function, you may implement your own rules about when the rating prompt is displayed:

```
if (!RateBox.rateBox.didRateCurrentVersion())
{
	RateBox.rateBox.showRatingPrompt("Rate My App", "Please rate this if you like it!");
}
```

Finally, the internal rating tracker can be reset by calling resetConditions(). This will also cancel any user preferences, such as having pressed 'Don't ask again':

```
RateBox.rateBox.resetConditions();
```

Further, if you want to completely forego using the extension's native rating prompt, you can design your own rating dialog box in Flash instead. If you do this, the following methods should be called, as appropriate, after the user has pressed one of the buttons in your custom dialog box:

```
// call this to show the rating page immediately - after the user pressed 'OK' in your custom dialog
RateBox.rateBox.gotoRatingsNow();

// call this if the user presses 'never ask again' in your custom dialog
RateBox.rateBox.recordCustomNeverButtonPress();

// call this if the user pressed 'Remind me later' in your custom dialog
RateBox.rateBox.recordCustomLaterButtonPress();
```

7. Update Your Application Descriptor

You'll need to be using the AIR 3.1 SDK or higher, include the extension in your Application Descriptor XML, and update the Android Manifest Additions with some RateBox specific settings. For a working example, see '*example/app.xml*'.

1. Set your AIR SDK to 3.1 or higher in the app descriptor file:

```
<application xmlns="http://ns.adobe.com/air/application/3.1">
```

2. Include a link to the extension in the descriptor:

```
<extensions>
<extensionID>com.milkmangames.extensions.RateBox</extensionID>
</extensions>
```

3. Update your Android Manifest Additions (the INTERNET and ACCESS_NETWORK_STATE permissions are required):

```
<android>
<manifestAdditions><![CDATA[
<manifest android:installLocation="auto">
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
</manifest>
]]></manifestAdditions>
</android>
```

8. Build the Application

When building with the extension, you'll need to specify the directory containing the com.milkmangames.extensions.RateBox.ane file.

Flash Builder 4.6:

Flash Builder 4.6 has full support for native extensions.

1. Make sure you've included the ANE file as described in Section 2, '*Include the RateBox Library.*'
2. Make sure that the extension is enabled for your target configuration.
3. Build your application and deploy as usual.

Flash CS6:

Flash CS6 has full support for native extensions.

1. Make sure you've included the ANE file as described in Section 2, *'Include the RateBox Library'.*
2. Build your mobile application and deploy as usual.

9. Troubleshooting and FAQ

"How do I use the RateBoxExample.as file in Flash Professional CS6?"

1. First, create the application, add the extension, and update Google Play by following this guide, Sections 1-2.
2. Copy and paste RateBoxExample.as into the same folder as your .fla. Do not copy and paste its contents on to the timeline. That will not work.
3. In Flash properties, under 'Document Class', type 'RateBoxExample' (no quotes) and press OK.
4. Build and install the application.

"Why doesn't the extension work when I run my swf on my Mac / iPhone / Windows Computer?"

- The extension depends on functions of the iOS and Android operating systems. The extension will only work when you run it on an Android or iOS phone or tablet.

"I still need help! How can I get technical support?"

- Your purchase comes with free technical support by email – just send us a message at support@milkmangames.com . We're here to help! Please remember that...:
- We're open during United States business hours, excluding U.S. Holidays, Monday-Friday, Pacific Standard Time. We strive to answer all support email within 24 hours (not including weekends and holidays) but usually do so much faster. Remember that we may not be in the same time zone.
- Please remember to mention: which extension you're having a problem with, what IDE you're using (such as 'Flash CS6' or 'Flash Builder 4.6'), and what device you're targeting. If you're experiencing an error message, please specify what that message is.
- We don't provide tech support through blog comments, Facebook, or Twitter. Please email us and we'll be happy to help you out!