Simple Android Notifications

by Hippo Games

TABLE OF CONTENTS

1 About

Simple Android Notifications for Unity provides you a simple way to create local notifications for your games and apps. It uses native Android Alarm Manager and Notification Manager.

The plugin supports Android 4.0.3 and later and is compatible with Unity 5 and Unity 2017. You can play demo scene to check notifications on your device.

2 Features

- Create local notifications
- Select color, icon
- Enable/disable sound, vibration, lights
- Demo scene
- Supports Android 4.0.3+ (API level 15+)
- Compatible with Unity 5
- 6 general preloaded icons
- Nothing extra
- Device reboot is not supported

3 Paid version features

- Device reboot support
- Cancellation of already displayed notifications
- Repeated notifications
- Multiline support (API 16+)
- Grouped notifications (API 20+)
- Android Oreo support (API 26, API 27) **[NEW]**
- Notification channels (API 26+) **[NEW]**
- 20 general preloaded icons
- Notification execute mode
- Advanced notification id management
- Custom icons
- Custom sound
- Custom vibration
- Custom lights

4 Setup & test guide

1. Create a new empty project
2. Import the package (Assets/Import Package/Custom Package)
3. Set Minimum API Level 15 (Player Settings/Android/Other Settings)

4. Open and add SimpleAndroidNotifications/DemoScene.scene to Scenes in Build (Build Settings)
5. Build project to android device (Build & Run)
6. Test notifications and enjoy!

5 USE NOTIFICATIONMANAGER TO SEND NOTIFICATIONS

- `NotificationManager.Send(TimeSpan delay, string title, string message, Color smallIconColor, NotificationIcon smallIcon)`
- `NotificationManager.SendWithAppIcon(TimeSpan delay, string title, string message, Color smallIconColor, NotificationIcon smallIcon)`
- `NotificationManager.SendCustom(NotificationParams notificationParams)`

6 LIST OF BUILD-IN SMALL ICONS

Use `enum NotificationIcon` to select a small icon from preloaded collection:

- Bell
- Biohazard
- Clock
- Cloud
- Coin
- Crown
- Diamond
- Dollar
- Event
- Gear
- Heart
- Message
- Power
- Return
- Save
- Search
- Skull
- Star
- Sync
- Wrench

7 RECEIVING CALLBACK ON NOTIFICATION CLICK

First of all, please note, that callback feature works with the following restriction:

> Receiving callback will not work if your app was sleeping. It will only work if app was opened (not resumed) by clicking notification!

If you need to receive data from notification that was clicked, do the following:

1. refer to example scene, where notification with callback is creating
2. Set `NotificationParams.CallbackData` when creating notification. It has string type, so you can store JSON here
3. Call `NotificationManager.GetNotificationCallback()` in Awake or anywhere else. It will return notification `NotificationCallback`, or null if app was launched NOT from notification

8 MULTILINE SUPPORT

Multiline notifications require API 16+ (Android 4.1 JELLY_BEAN). Otherwise multiline message will be displayed as a single line (and will not cause any errors). Displaying of multiline notifications can be different on custom android shells (for example, MUI, ZUI). Long messages will be displayed as multiline automatically. You can also use `\n` to specify new lines manually. To enable multiline support, just set `Multiline = true`.

```
var notificationParams = new NotificationParams
{
    Id = NotificationIdHandler.GetNotificationId(),
    Delay = TimeSpan.FromSeconds(5),
    Title = "Multiline notification",
    Message = "Line#1\nLine#2\nLine#3\nLine#4",
    Ticker = "This is multiline message ticker!",
    Multiline = true
};

NotificationManager.SendCustom(notificationParams);
```

9 GROUPED NOTIFICATIONS

This feature was introduced in API 20 (Android 5.0). Otherwise your notifications will not be grouped (and this will not cause any errors). Displaying of grouped notifications can be different on custom android shells (for example, MUI, ZUI). When you tap on group notification, all child notifications will be closed. Use GroupName to specify group notification title and GroupSummary to specify its summary. If GroupSummary will contain {0} marker, then it will be replaced by notification counter.

```
var id = NotificationIdHandler.GetNotificationId();
var notificationParams = new NotificationParams
{
    Id = id,
    GroupName = "Group",
    GroupSummary = "{0} new messages",
    Delay = TimeSpan.FromSeconds(5),
    Title = "Grouped notification",
    Message = "Message " + id,
    Ticker = "Please rate the asset on the Asset Store!"
};

NotificationManager.SendCustom(notificationParams);
```

10 NOTIFICATION CHANNELS

This feature was introduced in API 26 (Android Oreo)! Also note, that channels are not the same as notifications groups (stacks).
Notification channels provide us with the ability to group the notifications that our application sends into manageable groups. Once our notifications are in these channels, we no longer have input into their functionality - so it is up to the user to manage these channels. When it comes to altering the settings for our application notifications, the user will be presented with these options. So now you can simply specify ChannelId and ChannelName. You can create a channel with different localized names, so ChannelId can be not equal to ChannelName.

```
var notificationParams = new NotificationParams
{
    Id = NotificationIdHandler.GetNotificationId(),
    Delay = TimeSpan.FromSeconds(5),
    Title = "Notification with news channel",
    Message = "Check the channel in your app settings!",
    Ticker = "Notification with news channel",
    ChannelId = "com.company.app.news",
    ChannelName = "News"
};

NotificationManager.SendCustom(notificationParams);
```

11 REQUIREMENTS

These requirements are already done in clean setup.

If you define *Plugins\Android\AndroidManifest.xml*, it should contain UnityPlayerActivity activity. This activity is used by plugin and its name is “hard coded” in plugin C# code.

simple-android-notifications.aar/AndroidManifest.xml should contain the following permissions and receivers:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
	xmlns:android="http://schemas.android.com/apk/res/android"
	package="com.unity3d.player"
	android:versionCode="1"
	android:versionName="1.0">
	<uses-permission android:name="android.permission.VIBRATE" />
	<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
	<uses-sdk android:minSdkVersion="15" />
	<application
		android:icon="@drawable/app_icon"
		android:label="@string/app_name">
		<receiver android:name="com.hippogames.simpleandroidnotifications.Controller" />
		<receiver android:name="com.hippogames.simpleandroidnotifications.RebootManager" >
			<intent-filter>
				<action android:name="android.intent.action.BOOT_COMPLETED" />
			</intent-filter>
		</receiver>
	</application>
</manifest>
```

12 FAQ & TROUBLESHOOTING

Please visit project wiki on GitHub:
https://github.com/hippogamesunity/SimpleAndroidNotificationsPublic/wiki

12.1 ADDING CUSTOM NOTIFICATION ICONS

Note: you’ll need to modify plugin .aar archive. Use Windows instead of Mac OS, because Mac OS can add hidden files and can create incorrect new archive in result. I would recommend you to use Total Commander for this task.

Follow those steps if you want to add custom icons:

1. find plugin .aar file
2. change extension from .aar to .zip (you can also open .aar directly if your file manager supports this feature)
3. open .zip in any archive program
4. find /res folder
5. provide all icon sizes and put files to each /drawable folder
6. change extension from .zip to to .aar
7. extend `enum NotificationIcon`
8. update `string GetSmallIconName(NotificationIcon icon)` if needed

Recommended size for small icons:

- 24x24 for drawable-mdpi
- 36x36 for drawable-hdpi
- 48x48 for drawable-xhdpi
- 72x72 for drawable-xxhdpi
- 96x96 for drawable-xxxhdpi
- 96x96 for drawable

Recommended size for large icons (notification background):

- 48x48 for drawable-mdpi
- 72x72 for drawable-hdpi
- 96x96 for drawable-xhdpi
- 144x144 for drawable-xxhdpi
- 192x192 for drawable-xxxhdpi
- 192x192 for drawable

12.2 Adding custom sounds

Note: you'll need to modify plugin .aar archive. Use Windows instead of Mac OS, because Mac OS can add hidden files and can create incorrect new archive in result. I would recommend you to use Total Commander for this task.

Follow those steps if you want to add custom sounds:

1. find plugin .aar file
2. change extension from .aar to .zip (you can also open .aar directly if your file manager supports this feature)
3. open .zip in any archive program
4. find /res folder
5. create /raw folder inside /res folder
6. place your sound files here. Tested extensions: mp3, wav. Sound file name must contain only [a-z0-9_.]. Don't use uppercase! You must get something like *simple-android-notifications.aar/res/aar/ding.mp3.*
7. change extension from .zip to to .aar
8. Set *CustomSound* parameter when creating notification, use file name only (without extension):

```
var notificationParams = new NotificationParams
{
    …
    Sound = true,
    CustomSound = "ding",
    …
};
```

12.3 Optimization

Note, that preloaded icons take about 300 KB of disk space. Open .aar plugin using any zip program and remove all unused icons from /res folder to reduce your apk final size.