



Simple Android Notifications

By Hippo Games

TABLE OF CONTENTS

1	About.....	2
2	Contact us.....	2
3	Features.....	2
4	Paid version features.....	2
5	Setup & test guide.....	3
6	Use NotificationManager to send notifications.....	3
7	List of build-in small icons.....	3
8	Receiving callback on notification click.....	4
9	Multiline support.....	4
10	Grouped notifications.....	5
11	Notification channels.....	5
12	Requirements.....	5
13	FAQ & Troubleshooting.....	6
13.1	Adding custom notification icons.....	6
13.2	Adding custom sounds.....	8
13.3	Notification priority and importance.....	8
13.4	Optimization.....	9

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, Unity 2017 and Unity 2018. You can play demo scene to check notifications on your device.

2 CONTACT US

We have a public project on GitHub, so please submit your requests, questions and bugs here:

<https://github.com/hippogamesunity/SimpleAndroidNotificationsPublic/issues>

Find the latest manual and FAQ in our wiki:

<https://github.com/hippogamesunity/SimpleAndroidNotificationsPublic/wiki>

You can also contact us by email:

hippogamesunity@gmail.com

3 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

4 PAID VERSION FEATURES

- Device reboot support
- Cancellation of already displayed notifications
- Repeated notifications
- Multiline support (API 16+)
- Grouped (stacked) notifications (API 23+)
- 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

5 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!

6 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)`

7 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

8 RECEIVING CALLBACK ON NOTIFICATION CLICK

You can specify some string data when scheduling notifications to identify what notification was clicked.

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 `OnApplicationPause`. It will return notification `NotificationCallback`, or null if app was launched NOT from notification
4. Check `NotificationManager.GetNotificationCallback().Data` to get callback data

9 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);
```

10 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);
```

11 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);
```

12 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>
```

13 FAQ & TROUBLESHOOTING

Please visit project wiki on GitHub:

<https://github.com/hippogamesunity/SimpleAndroidNotificationsPublic/wiki>

13.1 ADDING CUSTOM NOTIFICATION ICONS

Note: you will 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 `.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

13.2 ADDING CUSTOM SOUNDS

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

Follow those steps if you want to add custom sounds:

1. find `.aar` archive at `Assets\SimpleAndroidNotifications\Plugins\Android\simple-android-notifications.aar`
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/raw/ding.mp3`.
7. change extension from `.zip` back to `.aar`
8. Set `CustomSound` parameter when creating a notification, use file name only (without extension):

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

Notes: in Android API 26 (Oreo) notification sounds are moved to channels. After a new channel is created, it will be displayed in the app's notification settings. All channel settings (including custom sound) should be defined while creating a new channel. There is no way to change channel's settings (including custom sound), except reinstalling the app.

13.3 NOTIFICATION PRIORITY AND IMPORTANCE

Starting from Android API 16 and up to 25 (including) all notifications have priority. Starting from API 26 (Oreo) notification priority becomes obsolete, and channel importance is introduced. You can use `enum NotificationImportance` to specify channel importance. It will be replaced by notification priority for old Android versions.

```
var notificationParams = new NotificationParams
{
    ...
    Importance = NotificationImportance.Max,
    ...
};
```


13.4 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.