
iOSNotification Documentation

Release 1.0

Area730

January 16, 2017

1	Contents	1
1.1	Installation	1
1.2	Local Notification	2
1.3	Push Notification With OneSignal	7
1.4	Modifying a plugin	7
1.5	Other	8
1.6	Installation	8
1.7	Create IOSNotification With Code	9
1.8	Create IOSNotification With Visual Tool	13
1.9	Push Notification with OneSignal integration	14
1.10	Modifying plugin	14
1.11	Other	14

1.1 Installation

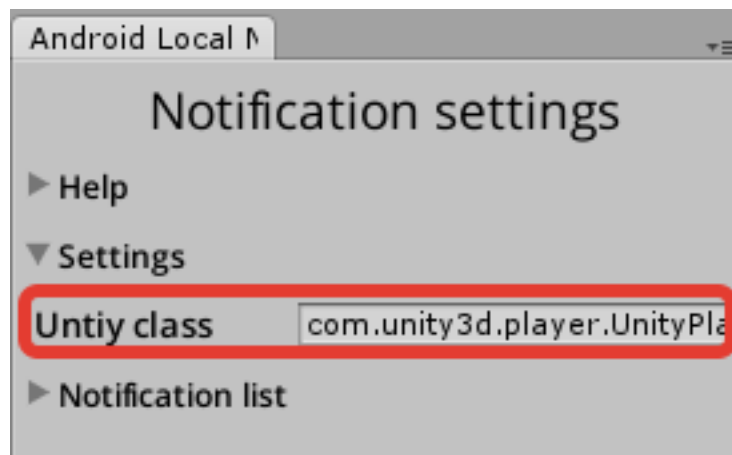
1. Import [this plugin](#) into your Unity project.
2. Check if you have **AndroidManifest.xml** in Assets/Plugins/Android folder.

If you don't - add this [manifest](#) to *Assets/Plugins/Android* folder.

If you do - check if it contains `UnityPlayerNativeActivity` or the one that extends it.

If you have `UnityPlayerNativeActivity` - you are good to go.

If you have activity that extends **UnityPlayerNativeActivity**- set its full name (e.g. `com.unity3d.player.UnityPlayerNativeActivity`) in *Window->Ultimate Local Notifications -> Settings*



If you don't have any - add this activity to your manifest:

```
<activity android:name="com.unity3d.player.UnityPlayerNativeActivity" android:label="@string/app_name"
  <intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.LAUNCHER" />
    <category android:name="android.intent.category.LEANBACK_LAUNCHER" />
  </intent-filter>
  <meta-data android:name="unityplayer.UnityActivity" android:value="true" />
  <meta-data android:name="unityplayer.ForwardNativeEventsToDalvik" android:value="false" />
</activity>
```

1.2 Local Notification

1.2.1 Schedule simple notifications

The package contains code samples in `Assets/Area730/Notifications/Examples/Scripts` folder.

The notifications are created using `NotificationBuilder` class. Its constructor takes 3 arguments - id of the notification, title and notification text. Next example shows how to schedule the notification that will be shown immediately:

```
int id          = 1;
string title    = "Notification title";
string body     = "Notification body";

NotificationBuilder builder = new NotificationBuilder(id, title, body);
AndroidNotifications.scheduleNotification(builder.build());
```

1.2.2 Schedule delayed notifications

If you want to set delay - call `builder.setDelay(int miliseconds)` or `builder.setDelay(System.TimeSpan delayTime)`. The next example shows how to create a notification that will be shown in one hour:

```
int id          = 1;
string title    = "Notification title";
string body     = "Notification body";

// Show notification in one hour
TimeSpan delay = new TimeSpan(1, 0, 0);

NotificationBuilder builder = new NotificationBuilder(id, title, body);
builder.setDelay(delay);

AndroidNotifications.scheduleNotification(builder.build());
```

1.2.3 Customization

`NotificationBuilder` allows you to set different parameters of your notification such as color, small icon, large icon, auto cancel, alert once, ticker, notification number, sound, vibration pattern, group, sort key and if the notification repeats every interval of time.

All methods with description you can find in `NotificationBuilder.cs` file.

Next example shows scheduling of the notification that will be shown in 15 minutes with ticker, default audio and vibration, with autocancel (if you click it - it will be removed from the list), and with red background color (background color is not supported on some Android versions, please refer to Android docs for more info)

```
int id          = 1;
string title    = "New notification";
string body     = "You have some unfinished business!";

// Show notification in 15 minutes
TimeSpan delay = new TimeSpan(0, 15, 0);

NotificationBuilder builder = new NotificationBuilder(id, title, body);
builder
```

```
.setTicker("New notification from your app!")
.setDefaults(NotificationBuilder.DEFAULT_ALL)
.setDelay(delay)
.setAutoCancel(true)
.setColor("#B30000");
```

```
AndroidNotifications.scheduleNotification(builder.build())
```

1.2.4 Repeating notifications

To set repeating notification you should set notification as repeating **and set the time interval**. Next example shows scheduling of the notification that will be shown in 5 minutes and then shown every 10 minutes:

```
int id          = 1;
string title    = "New repeating notification";
string body     = "You have some unfinished business!";

// Show notification in 5 minutes
TimeSpan delay  = new TimeSpan(0, 5, 0);

// Show notification with 10 minute interval
TimeSpan interval = new TimeSpan(0, 10, 0);

NotificationBuilder builder = new NotificationBuilder(id, title, body);
builder
    .setDelay(delay)
    .setRepeating(true)
    .setInterval(interval);

AndroidNotifications.scheduleNotification(builder.build());
```

1.2.5 Settings custom icons

You can set custom icons for your notification. There are 2 types of icon - small and large. Small icon is mask. Both icons should be located in *Assets/Plugins/Android/Notifications/res/drawable* folder or one of the drawable folders (e.g. *drawable-mdpi* etc.).

You can use these icon generators:

1. **Small icon generator** - generate and download archive with your icons. Then just copy all drawable folders from the archive into *Assets/Plugins/Android/Notifications/res* folder and set **the name of the icon without extension** as your small icon - **builder.setSmallIcon("myIcon")**
2. **Large icon generator** - generate and download archive with your icons. The archive will contain mipmap folders (*mipmap-mdpi*, *mipmap-hdpi* etc.). Copy the icons into corresponding **drawable** folders in *Assets/Plugins/Android/Notifications/res* folder (icon from *mipmap-hdpi* into *drawable-hdpi*, *mipmap-mdpi* into *drawable-mdpi* etc.). Next, set **the name of the icon without extension** as your large icon - **builder.setLargeIcon("myLargeIcon")**

```
int id          = 1;
string title    = "Custom icon";
string body     = "You have some unfinished business!";

// Show notification in 5 minutes
TimeSpan delay = new TimeSpan(0, 5, 0);
```

```
// WARNING: icons should be in Assets/Plugins/Android/Notification/res/drawable(-mdpi etc.) folders
NotificationBuilder builder = new NotificationBuilder(id, title, body);
builder
    .setDelay(delay)
    .setSmallIcon("mySmallIconFilename")
    .setLargeIcon("myLargeIconFilename");

AndroidNotifications.scheduleNotification(builder.build());
```

1.2.6 Settings custom sound

You can set custom sound for your notification. The sound should be located in *Assets/Plugins/Android/Notifications/res/raw* folder. To set custom sound use `builder.setSound("mySound")` method. **Name of the sound file should be without extension.**

```
int    id      = 1;
string title   = "Custom sound";
string body    = "You have some unfinished business!";

// Show notification in 5 minutes
TimeSpan delay = new TimeSpan(0, 5, 0);

// WARNING: the sound should be in Assets/Plugins/Android/Notification/res/raw folder
NotificationBuilder builder = new NotificationBuilder(id, title, body);
builder
    .setDelay(delay)
    .setSound("mySoundFileName");

AndroidNotifications.scheduleNotification(builder.build());
```

1.2.7 Cancel notification by id (both repeating and one-time)

To cancel the notification by id, simply call `AndroidNotifications.cancelNotification(int id)`.

```
//cancel notification with id 7
AndroidNotifications.cancelNotification(7);
```

1.2.8 Cancel all notifications

```
AndroidNotifications.cancelAll();
```

1.2.9 Clear shown notifications

To clear certain notification use `AndroidNotifications.clear(int id)`.

```
// clear shown notification with id 7
AndroidNotifications.clear(7);
```

To clear all shown notifications use `AndroidNotifications.clearAll()`.

```
// clear all shown notifications
AndroidNotifications.clearAll();
```


1.2.10 Updating notifications

To update one-time or repeating notification, schedule a notification with updated data and with ID of the notification you want to update.

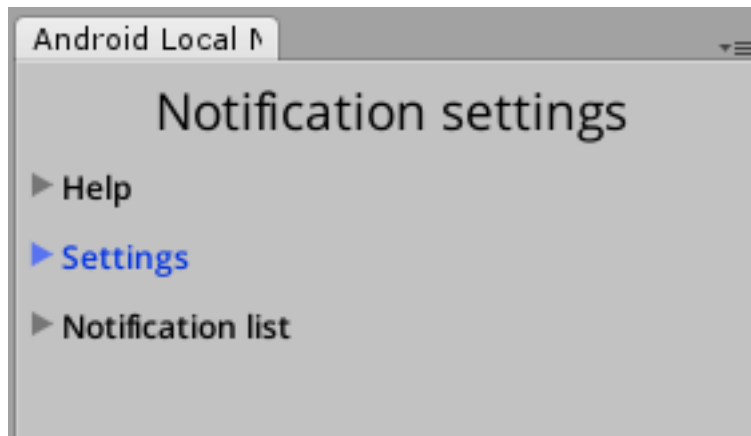
1.2.11 Show android toast notification

To show a toast notification use `AndroidNotifications.showToast(string text)`.

```
AndroidNotifications.showToast("Download completed");
```

1.2.12 Notification editor

Plugin comes with editor extension that allows you to create notifications without the line of code. To open the notification editor window go to *Window -> Android Local Notifications*.



In **Help** section you will find some useful links. In **Settings** section you can set custom Unity class if your activity extends *UnityPlayerNativeActivity*. In **Notification List** section you can add and modify notifications.



When you set custom notification sound or icons in editor window - they will be automatically copied to Notifications/res/drawable and Notifications/res/raw folders. **Though you will still need to add resized versions to drawable-hdpi and other folders using icon generators mentioned above.**

For detailed information on notification options please refer to [official Android docs](#)

1.2.13 Schedule notification created in editor

You can get notification you created by its name you set in editor



Next example shows scheduling of the notification created in editor with name **notificationOne**

```
string notificationName = "notificationOne";

// Method returns builder so you can config your notification afterwards if you want
NotificationBuilder builder = AndroidNotifications.GetNotificationBuilderByName(notificationName);

// If notification with specified name doesn't exist builder will be null
if (builder != null)
{
    Notification notif = builder.build();
    AndroidNotifications.scheduleNotification(notif);
}
```

1.3 Push Notification With OneSignal

To configure push notification for android platform follow next steps:

1. Create [GMS](#) application by following [this](#) tutorial instruction.
2. Do [step 3](#) to config your *AndroidManifest.xml*
3. Go to *Assets/Area730/Notifications/PushNotification* drag and drop *PushController.prefab* or just add *CrossPlatformPushNotificationController.cs* script to your gameobject.
4. Fill the values in *CrossPlatformPushNotificationController.cs*.

Now you are ready to send notifications. After these steps you will be able to send push notification using [One Signal](#) service.

1.4 Modifying a plugin

Source code of the plugin is included in the package. You can easily extend it if you want. Java library is built with **AndroidStudio**. There are 2 tasks in *build.gradle* file you should modify - **deleteOldJar** and **exportJar**.

```
//task to delete the old jar
task deleteOldJar(type: Delete) {
    // Set path to the jar file you exported
    delete 'D:\\Projects\\Unity 5x\\Plugins\\Notifications\\Assets\\Plugins\\Android\\Notifications\\libs\\Notifications.jar'
}

//task to export contents as jar
task exportJar(type: Copy) {
    from('build/intermediates/bundles/release/')

    // Set path to the folder where to export the .jar library
    // Set it to YourProject/Assets/Plugins/Android/Notifications/libs/ if you want your library to be exported straight to unity
    into('D:\\Projects\\Unity 5x\\Plugins\\Notifications\\Assets\\Plugins\\Android\\Notifications\\libs\\')
    include('classes.jar')
    ///Rename the jar
    rename('classes.jar', 'Notifications.jar')
}
```

In **deleteOldJar** task set path to the jar file you will export so every time you run a new build the old version will be deleted. In **exportJar** set the path where you want to export your jar.

To export jar from AndroidStudio go to *Gradle Projects/Tasks/Other* and run **exportJar** task.

In Unity plugin is in *Assets/Plugins/Android/Notifications* folder. It is stored as android library project.

To debug this plugin in AndroidStudio add **Area730Log** log tag to you logcat filter.

1.5 Other

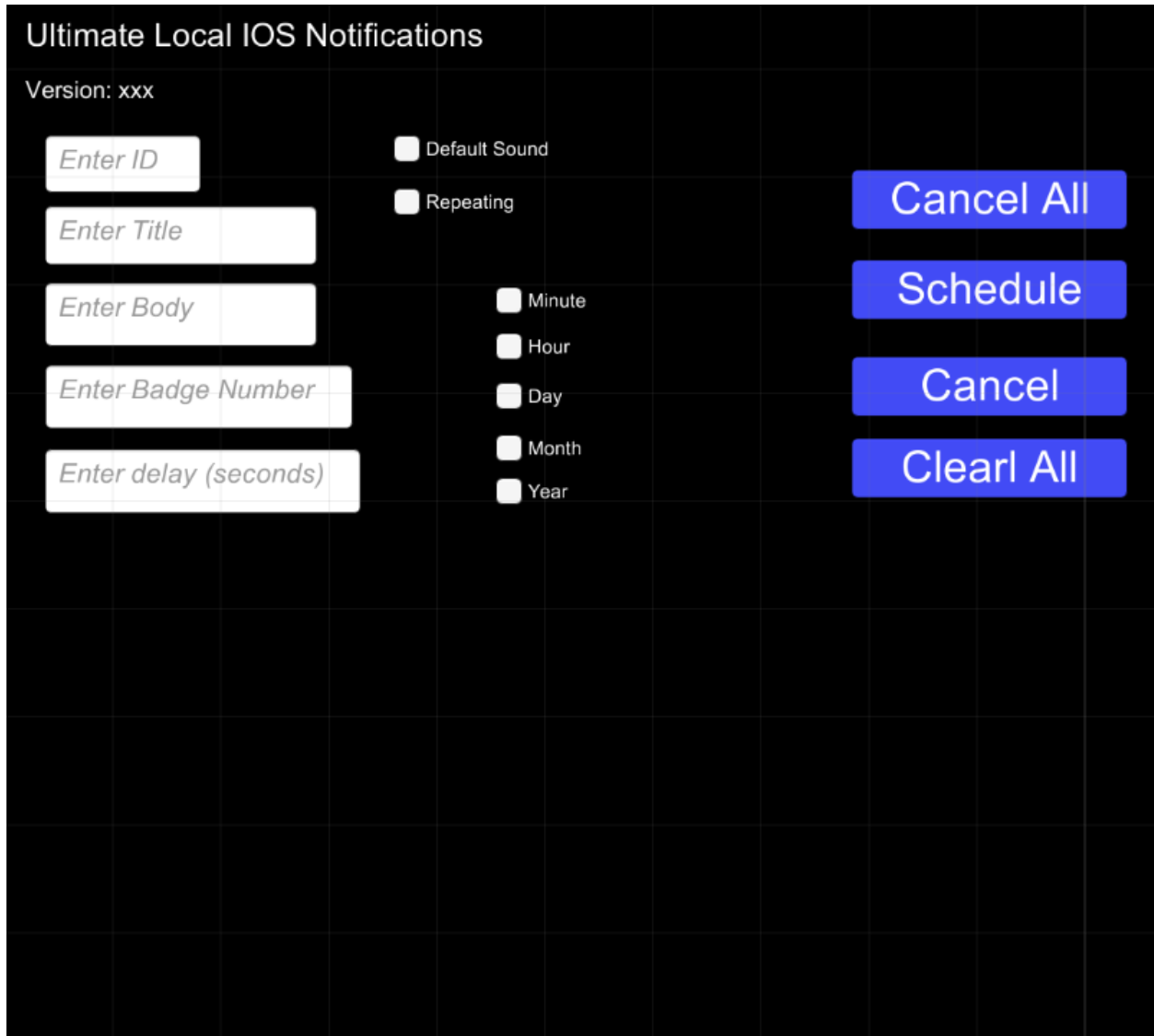
All classes are located in **Area730.Notifications** namespace

Example scene with sample code is included in the package (*Assets/Area730/Notifications/Examples*)

1.6 Installation

Import [this plugin](#) to your Unity project

Now you could build and run application to test. But please read all documentation!



1.7 Create iOSNotification With Code

1.7.1 Schedule simple notification

The package contains code samples in `Assets/Area730/Notifications/Examples/Scripts` folder. Also you can build and run example scene `Assets/Area730/Notifications/IOS/Examples` to test notification.

The notifications are created using `IOSNotificationBuilder` class. Its constructor takes 3 arguments - `id` of the notification, `title` and `notification text`.

Next example shows how to schedule the notification that will be shown immediately:

```
int id          = 1;
string title    = "Notification titile";
string body     = "Notification body";
```

```
IOSNotificationBuilder builder = new IOSNotificationBuilder (id, title, body);
IOSNotifications.scheduleNotification(builder.build());
```

1.7.2 Schedule delayed notifications

If you want to set delay - call `builder.setDelay(int milliseconds)` or `builder.setDelay(System.TimeSpan delayTime)`. The next example shows how to create a notification that will be shown in one hour:

```
int id          = 1;
string title    = "Notification titile";
string body     = "Notification body";

// Show notification in one hour
TimeSpan delay = new TimeSpan(1, 0, 0);

IOSNotificationBuilder builder = new IOSNotificationBuilder (id, title, body);
builder.setDelay(delay);

IOSNotifications.scheduleNotification(builder.build());
```

1.7.3 Repeating notifications

To set repeating notification you should set notification as repeating and set the time interval. According to [Apple documntaion](#) it is allowed to repeat notification every:

1. Minute
2. Hour
3. Day
4. Month
5. Year

```
int id          = 1;
string title    = "Notification titile";
string body     = "Notification body";

// Show notification in one hour
IOSNotificationBuilder builder = new IOSNotificationBuilder (id, title, body);
builder.setInterval(IntervalUnits.HOUR);

IOSNotifications.scheduleNotification(builder.build());
```

1.7.4 Set Up Badge Number

```
int id          = 1;
string title    = "Notification titile";
string body     = "Notification body";

// Show notification in one hour
IOSNotificationBuilder builder = new IOSNotificationBuilder (id, title, body);
builder.setNumber(3);
```

```
IOSNotifications.scheduleNotification(builder.build());
```

1.7.5 Settings custom sound

Now its supported only wav format sound notification. Next section show how to use custom sound for notification

```
IOSNotificationBuilder builder = new IOSNotificationBuilder (id, title, body);  
builder.setSound("notification_sound");//without wav extention  
  
IOSNotifications.scheduleNotification(builder.build());
```

***Important** When you set up sound via script please add source file to the xCode project into **Data/Raw* folder manually.* If you change audioclips via Editor please check *Assets/StreamingAssets* and *Assets/Plugins/iOS/Notifications* folders to delete old clips.

1.7.6 Cancel notification by id (both repeating and one-time)

```
//cancel notification with id 7  
IOSNotifications.cancelNotification(7);
```

1.7.7 Cancel all notification

```
//cancel all notification  
IOSNotifications.cancelAll();
```

1.7.8 Clear shown notifications

```
IOSNotifications.clearAll();
```

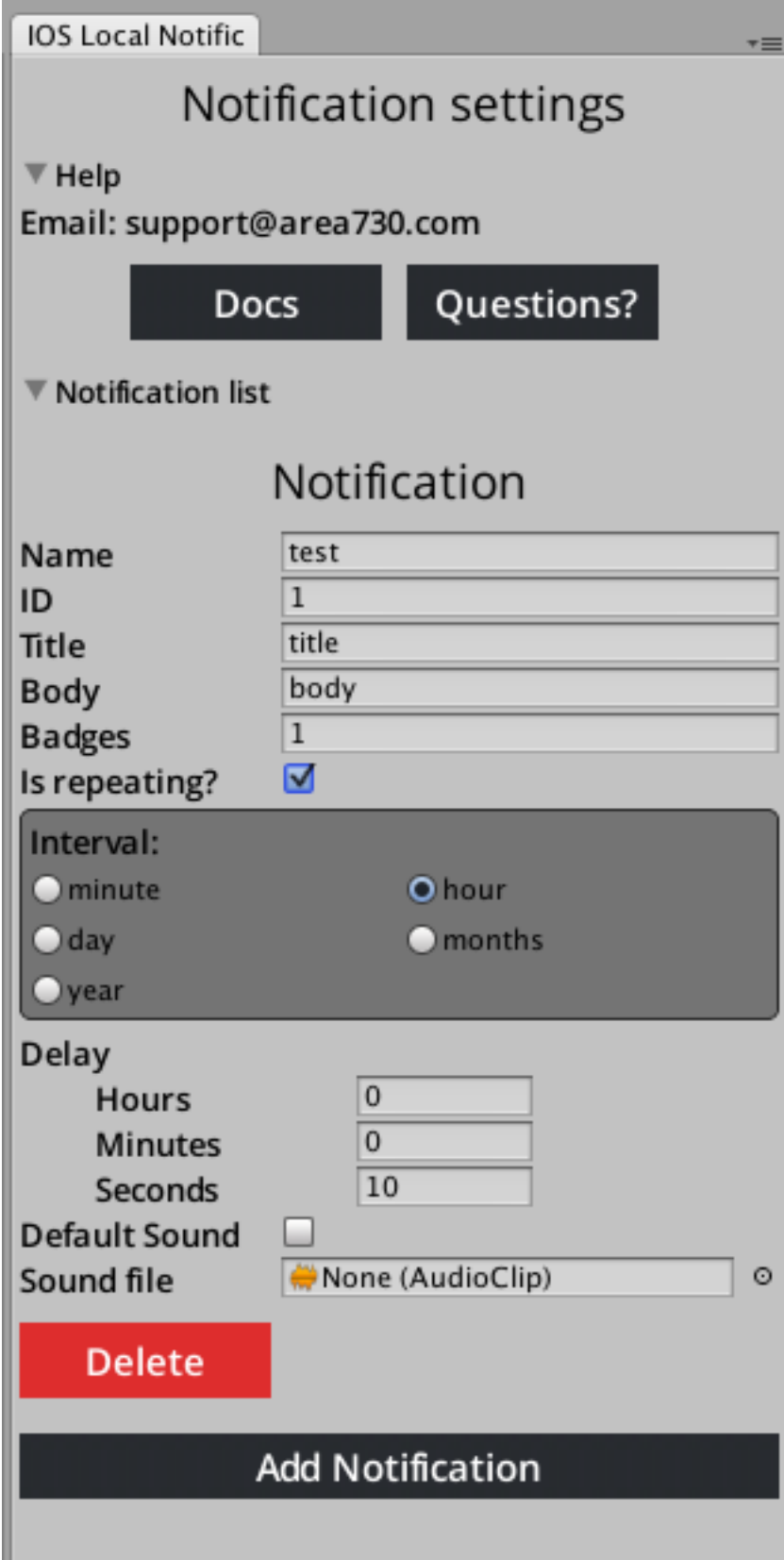
1.7.9 Updating notifications

To update one-time or repeating notification, schedule a notification with updated data and with ID of the notification you want to update.

1.7.10 Show IOS toast notification

```
IOSNotifications.showToast("Download completed");
```


1.8 Create iOSNotification With Visual Tool



The screenshot shows a window titled "iOS Local Notific" with a hamburger menu icon in the top right corner. The main heading is "Notification settings". Below this, there is a "Help" section with a dropdown arrow, an email address "support@area730.com", and two buttons: "Docs" and "Questions?". A "Notification list" section with a dropdown arrow follows. The main configuration area is titled "Notification" and contains several fields: "Name" (test), "ID" (1), "Title" (title), "Body" (body), and "Badges" (1). The "Is repeating?" checkbox is checked. Below this is an "Interval:" section with radio buttons for "minute", "hour" (selected), "day", "months", and "year". The "Delay" section has three input fields: "Hours" (0), "Minutes" (0), and "Seconds" (10). The "Default Sound" checkbox is unchecked, and the "Sound file" dropdown is set to "None (AudioClip)". At the bottom, there is a red "Delete" button and a large dark grey "Add Notification" button.

Notification settings

▼ Help
Email: support@area730.com

Docs Questions?

▼ Notification list

Notification

Name test

ID 1

Title title

Body body

Badges 1

Is repeating?

Interval:

minute hour

day months

year


Delay

Hours 0

Minutes 0

Seconds 10

Default Sound

Sound file  None (AudioClip) 

Delete

Add Notification

To open visual tool to create notification go to *Window->IOS Local Notification*

Next example shows scheduling of the notification created in editor with name **notificationOne**

```
string notificationName = "notificationOne";

// Method returns builder so you can config your notification afterwards if you want
IOSNotificationBuilder builder = IOSNotifications.GetNotificationBuilderByName(notificationName);

// If notification with specified name doesn't exist builder will be null
if (builder != null)
{
    IOSNotification notif = builder.build();
    IOSNotifications.scheduleNotification(notif);
}
```

1.9 Push Notification with OneSignal integration

Add `CrossPlatformPushNotificationController.cs` to some object in your scene and paste id from created application in onesignal. For more information go [here](#)

1.10 Modifying plugin

All native source code is holding in *Assets/Plugins/IOS/Notifications*

1.11 Other

All classes are located in **Area730.Notifications.IOS** namespace

Example scene with sample code is included in the package (*Assets/Area730/Notifications/Examples*)