Read me

For the plugin to work – drag and drop the prefab (Assets/NativePlugin/Plugin/NativePlugin.prefab) into the root scene of your unity project.

The plugin is not guaranteed to work on Android versions lower than 4.0. Immersive mode functions on versions of android higher than 4.4.

This plugin does not redefine “Activity Unity3D”. It can be safely used with plugins from the asset store.

Setting up AndroidManifest.xml:

1. If your app doesn’t use any other Android plugins, these setting are not required.
2. Local notifications require the <application> tag to contain the following line to function properly:

```
<receiver android:name="net.furylion.nativeunityplugin.PluginReceiver" />
```

3. Vibration requires the <manifest> tag to contain the following line:

```
<uses-permission android:name="android.permission.VIBRATE" />
```

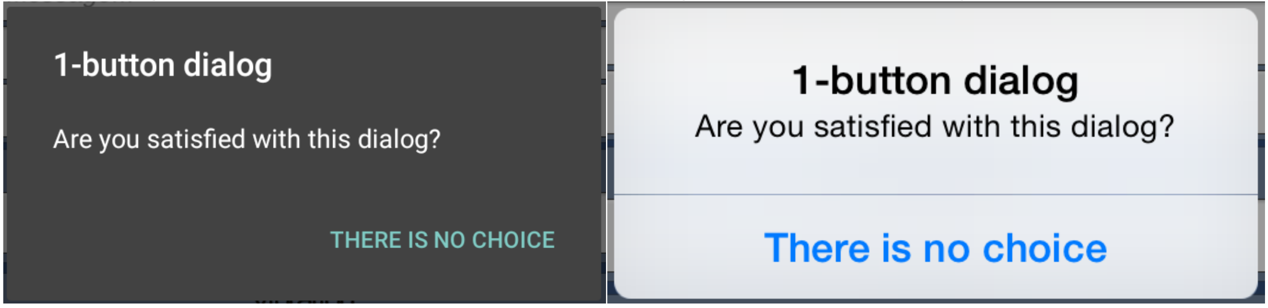
Native alert dialog:

Single button

```
void ShowDialog(string title, string message, string textButton, Action onClickButton)
string title – window title
string message – window message body
string textButton – button text
Action onClickButton – button press handler
```

Example:

```
NativePlugin.instance.ShowDialog("1-button dialog", "Are you satisfied with this dialog?", "There is no choice", () => { NativePlugin.instance.ShowDialog("Accepted!", "You've pressed a button", "Ok", null); textOneButtonDialog.text = (int.Parse(textOneButtonDialog.text) + 1).ToString(); });
```



1-button dialog

Are you satisfied with this dialog?

1-button dialog

Are you satisfied with this dialog?

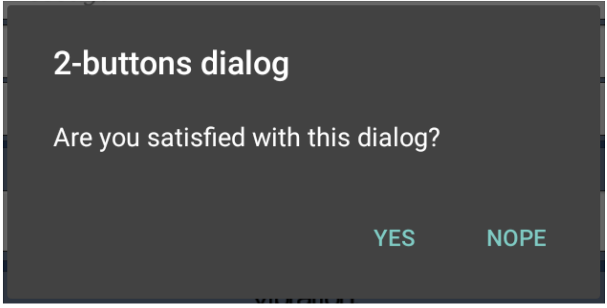
Two buttons

```
void ShowDialog(string title, string message, string textButtonLeft, string textButtonRight,
                                                       Action onClickButtonLeft, Action onClickButtonRight)
```

```
string title – window title
string message – window message body
string textButtonLeft – left button text
string textButtonRight – right button text
Action onClickButtonLeft – left button press handler
Action onClickButtonRight – right button press handler
```

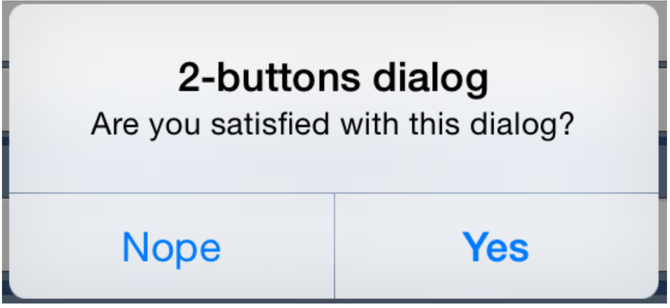
Example:

```
NativePlugin.instance.ShowDialog("2-buttons dialog", "Are you satisfied with this dialog?", "Nope",
"Yes", () => { NativePlugin.instance.ShowDialog("Accepted!", "Why no?", "Ok", null);
textTwoButtonsDialogOne.text = (int.Parse(textTwoButtonsDialogOne.text) + 1).ToString(); }, () => {
NativePlugin.instance.ShowDialog("Accepted!", "Pleased to know it ^^", "Ok", null);
textTwoButtonsDialogTwo.text = (int.Parse(textTwoButtonsDialogTwo.text) + 1).ToString(); });
```



2-buttons dialog

Are you satisfied with this dialog?



2-buttons dialog

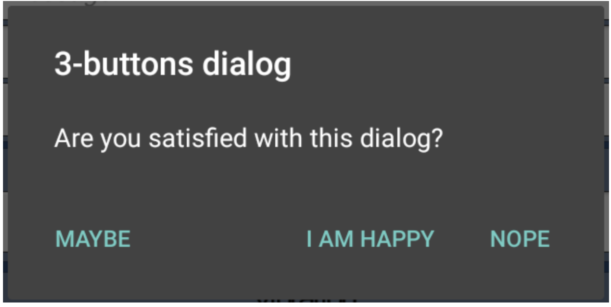
Are you satisfied with this dialog?

Three buttons

```
void ShowDialog(string title, string message, string textButtonLeft, string textButtonCenter,
string textButtonRight, Action onClickButtonLeft, Action onClickButtonCenter, Action onClickButtonRight)
string title – window title
string message – window message body
string textButtonLeft – left button text
string textButtonCenter  – middle button text
string textButtonRight – right button text
Action onClickButtonLeft – left button press handler
Action textButtonCenter – middle button press handler
Action onClickButtonRight – right button press handler
```

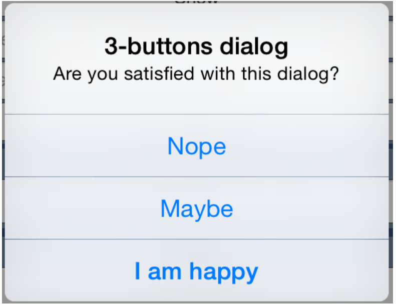
Example:

```
NativePlugin.instance.ShowDialog("3-buttons dialog", "Are you satisfied with this dialog?", "Nope",
"Maybe", "I am happy", () => { NativePlugin.instance.ShowDialog("Accepted!", "Oh no!", "Ok", null);
textThreeeButtonsDialogOne.text = (int.Parse(textThreeeButtonsDialogOne.text) + 1).ToString(); }, () => {
NativePlugin.instance.ShowDialog("Accepted!", "You are still in doubt", "Ok", null);
textThreeeButtonsDialogTwo.text = (int.Parse(textThreeeButtonsDialogTwo.text) + 1).ToString(); }, () => {
NativePlugin.instance.ShowDialog("Accepted!", "You are happy!", "That's right!", null);
textThreeeButtonsDialogThree.text = (int.Parse(textThreeeButtonsDialogThree.text) + 1).ToString(); });
```



3-buttons dialog

Are you satisfied with this dialog?

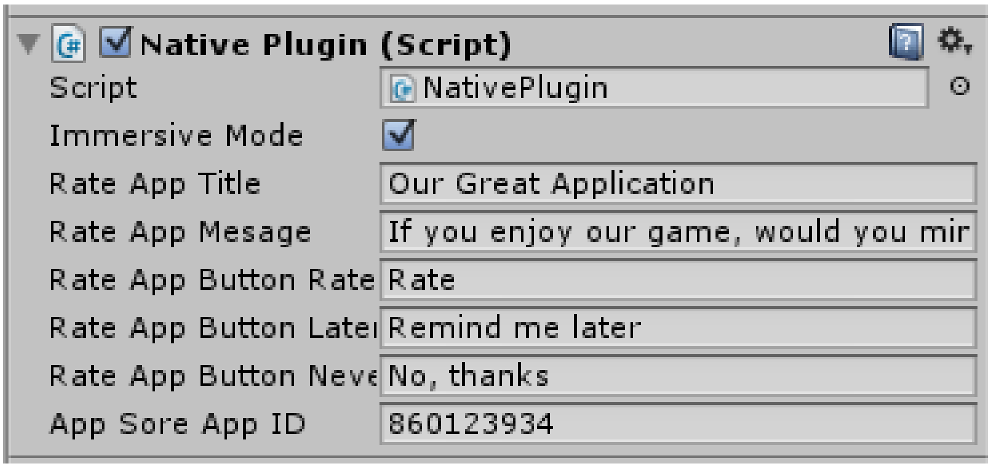


3-buttons dialog

Are you satisfied with this dialog?

Rate application

Displays a dialog box, offering the user to rate the application in Google Play, Apple App Store, Amazon Appstore with 3 options: rate now, ask later, never ask. If the user selects rate now or never ask – the dialog is never displayed again. The updated version of the application dialog will be displayed again.



Native Plugin (Script)

Script: NativePlugin

Immersive Mode: ☑

Rate App Title: Our Great Application

Rate App Mesage: If you enjoy our game, would you mir

Rate App Button Rate: Rate

Rate App Button Later: Remind me later

Rate App Button Neve: No, thanks

App Sore App ID: 860123934

Rate App Title – dialog title

Rate App Message – description message

Rate App Button Rate – rate button text

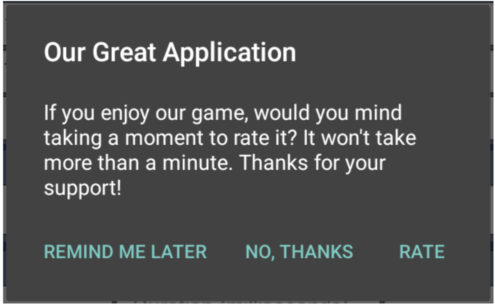
Rate App Button Later – later button text

Rate App Button Never – never button text

App Store App ID – application id in App Store

Example:

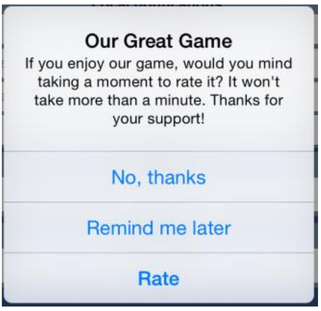
```
NativePlugin.instance.ShowRateMe();
```



Our Great Application

If you enjoy our game, would you mind taking a moment to rate it? It won't take more than a minute. Thanks for your support!

REMIND ME LATER NO, THANKS RATE



Our Great Game

If you enjoy our game, would you mind taking a moment to rate it? It won't take more than a minute. Thanks for your support!

No, thanks

Remind me later

Rate

Local notification

Notifies the user of an event via sound, vibration, LED indicator and message (after a predefined period in seconds). Tapping the notification message launches the app, or resumes, if the app is launched.

Show notification:

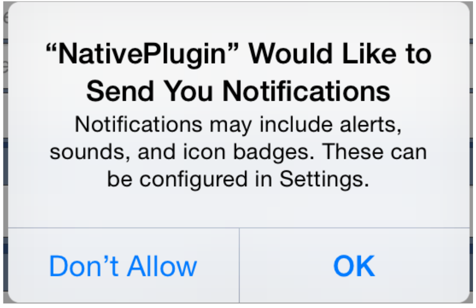
```
void ShowNotification(string title, string message, string ticker, int seconds)
string title – notification title
string message – notification text body
string ticker – short message
int seconds – notification launch interval in seconds
```

Cancel notification:

```
void CancelNotification()
```

Example:

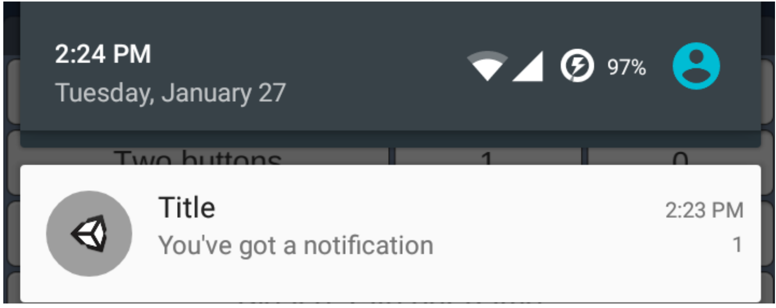
```
// Before sending the 1'st local notification,
// you need to prompt user to allow sending local notifications on iOS 8.
// "Apps that use local or push notifications must explicitly register the types of alerts
// that they display to users by using a UIUserNotificationSettings object.
// This registration process is separate from the process for registering remote notifications,
// and users must grant permission to deliver notifications through the requested options."
// From the section about UIKit Framework
https://developer.apple.com/library/prerelease/ios/releasenotes/General/WhatsNewIniOS/Articles/iOS8.html
NativePlugin.RegisterForNotifications();
```



“NativePlugin” Would Like to Send You Notifications

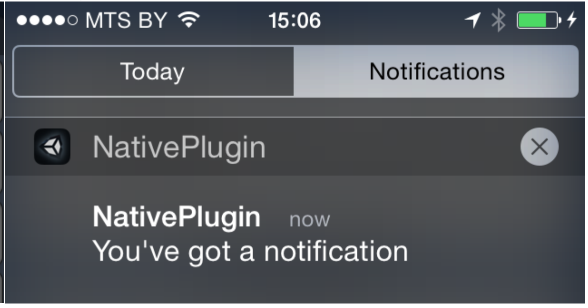
Notifications may include alerts, sounds, and icon badges. These can be configured in Settings.

```
NativePlugin.instance.ShowNotification(inputTextMessageNotification.text, seconds,
inputTextTitleNotification.text, inputTextTickerNotification.text);
```



2:24 PM
Tuesday, January 27

Title 2:23 PM
You've got a notification 1



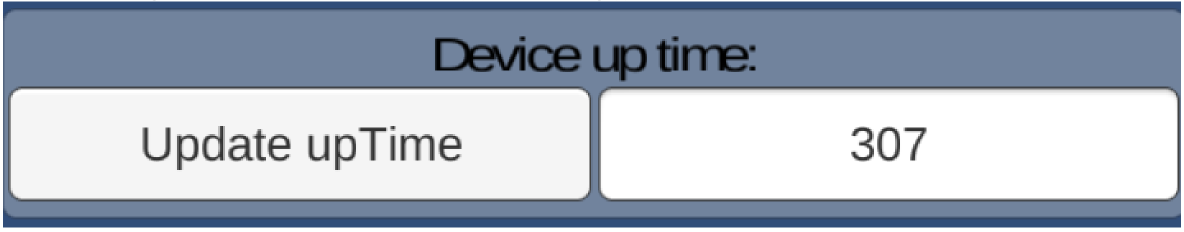
NativePlugin

NativePlugin now
You've got a notification

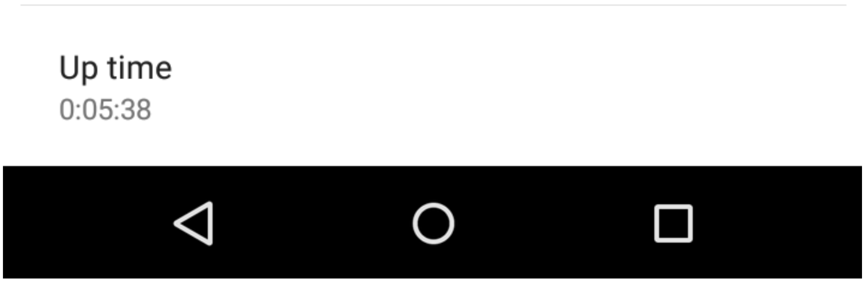
Device up time

Example:

```
text.text = NativePlugin.instance.GetUptime().ToString();
```



307 sec. / 60 ≈ 5 min



Up time
0:05:38

Vibration

Enables vibration if supported by device:

```
void Vibrate(long milliSeconds)
long milliSeconds – vibration time in milliseconds
```

Cancels vibration:

```
void VibrateCancel()
```

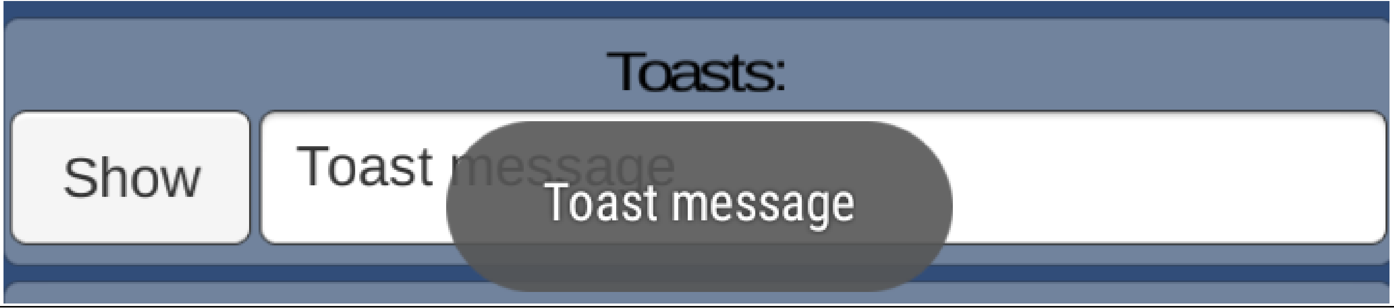
Toast (text popup notification)

Displays a text notification for a set amount of time.

```
void ShowToast(string message, toastLength duration = toastLength.LENGTH_SHORT)
string message – text message
toastLength duration – display time (2 seconds default)
public enum toastLength
{
    LENGTH_LONG = 1,    // 3.5 seconds
    LENGTH_SHORT = 0    // 2 seconds
}
```

Example*:*

```
NativePlugin.instance.ShowToast(inputField.text);
```



Immersive mode (Android 4.4+ only)

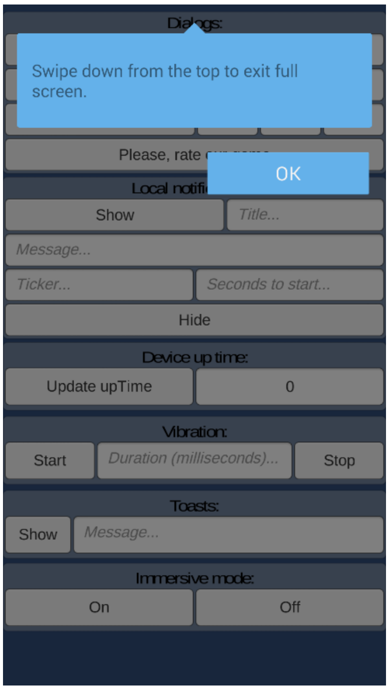
Hides/shows system screen buttons (back, home, etc…).

Hide system buttons

```
void TurnImmersiveModeOn()
```

Show system buttons

```
void TurnImmersiveModeOff()
```



Swipe down from the top to exit full screen.