



Android Goodies Documentation

[Android Goodies Documentation](#)

[Running the demo scene](#)

[1. Default Dialogs](#)

[Message dialog with positive button](#)

[Message dialog with positive and negative buttons](#)

[Message dialog with positive, negative and neutral buttons](#)

[Dialog with simple items chooser](#)

[Dialog with radio buttons items chooser](#)

[Dialog with check boxes buttons items chooser](#)

[2. Time and Date Pickers](#)

[Showing Date Picker](#)

[3. Native Share](#)

[Native share](#)

[Sending Email](#)

[Sending SMS](#)

[Tweeting](#)

[4. Open Maps](#)

[Open location with zoom level](#)

[Open location with label](#)

[Open location with address](#)

[5. Other useful methods](#)

[Checking if the app with certain package is installed](#)

[Showing Toast](#)

[Immersive Mode](#)

[Other Device info](#)

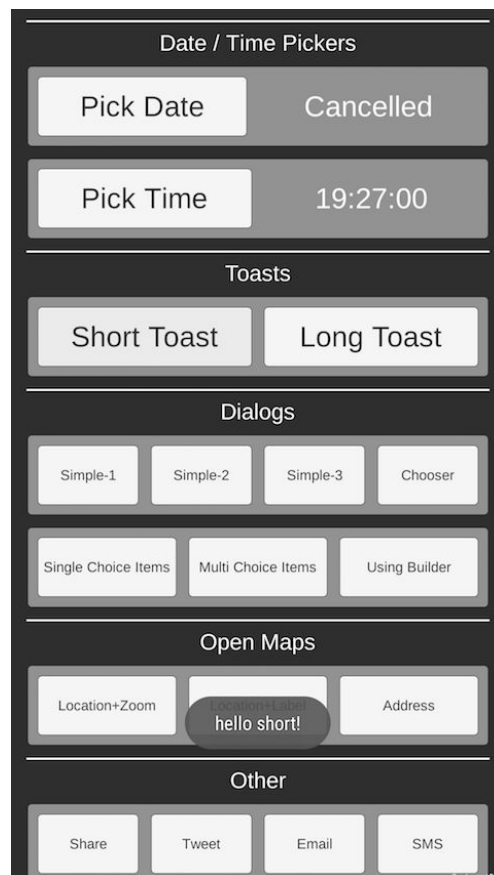
Android Goodies is a collection of Android native methods that allow you to do different things like showing native dialogs, opening a certain location on the map sharing text and many more.

The plugin is in active development and new things will be added. Please contact us if you have any questions or requests)

Running the demo scene

To build and run the demo scene just connect your Android device switch target platform to Android in Unity build settings and run the example scene on device or emulator. No extra setup is required.

You will see the menu like this on device where you can start testing:



1. Default Dialogs

You can show standard Android AlertDialog with:

- Only positive button
- Positive and negative buttons
- Positive, negative and neutral buttons
- Dialog with simple items chooser
- Dialog with radio buttons items chooser
- Dialog with check boxes buttons items chooser

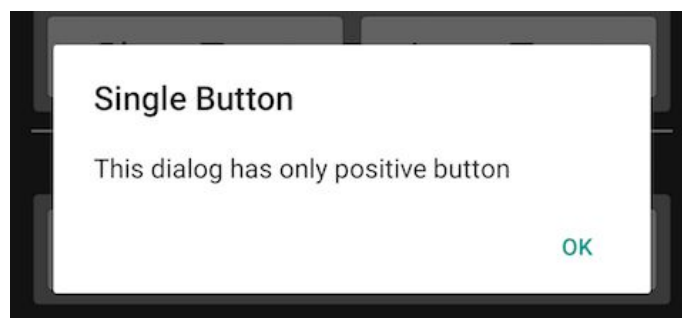
Message dialog with positive button

Opens Android [AlertDialog](#) with only positive button and optional dismiss callback.

Example usage:

```
AlertDialog.showAlertDialog("Single Button", "This dialog has only  
positive button", "Ok",  
() => AndroidGoodiesMisc.showToast("Positive button Click"),  
() => AndroidGoodiesMisc.showToast("Dismissed"));
```

Result:



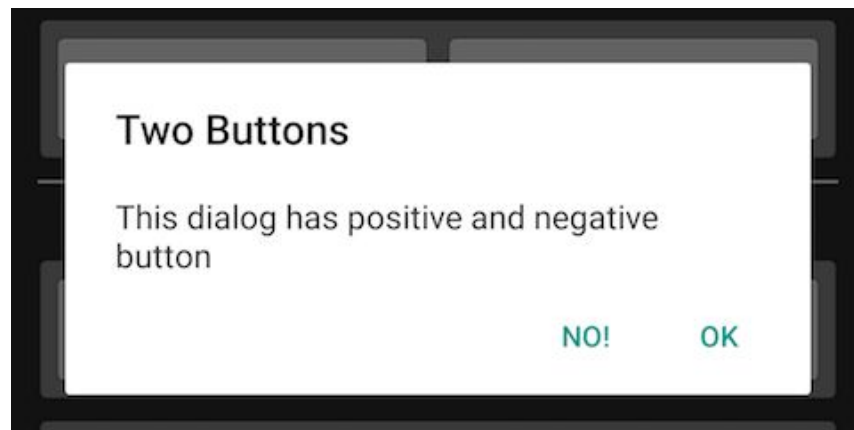
Message dialog with positive and negative buttons

Opens Android [AlertDialog](#) with positive and negative buttons and optional dismiss callback.

Example usage:

```
AlertDialog.ShowMessageDialog("Two Buttons", "This dialog has positive  
and negative button",  
    "Ok", () => AndroidGoodiesMisc.ShowToast("Positive button Click"),  
    "No!", () => AndroidGoodiesMisc.ShowToast("Negative button Click"),  
    () => AndroidGoodiesMisc.ShowToast("Dismissed"));
```

Result:



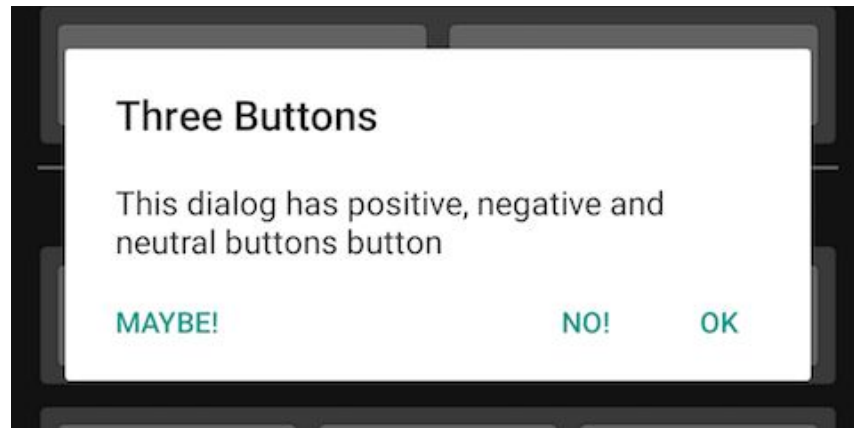
Message dialog with positive, negative and neutral buttons

Opens Android [AlertDialog](#) with positive, negative and neutral buttons and optional dismiss callback.

Example usage:

```
AlertDialog.ShowMessageDialog("Three Buttons",  
    "This dialog has positive, negative and neutral buttons button",  
    "Ok", () => AndroidGoodiesMisc.ShowToast("Positive button Click"),  
    "No!", () => AndroidGoodiesMisc.ShowToast("Negative button Click"),  
    "Maybe!", () => AndroidGoodiesMisc.ShowToast("Neutral button Click"),  
    () => AndroidGoodiesMisc.ShowToast("Dismissed"));
```

Result:



Dialog with simple items chooser

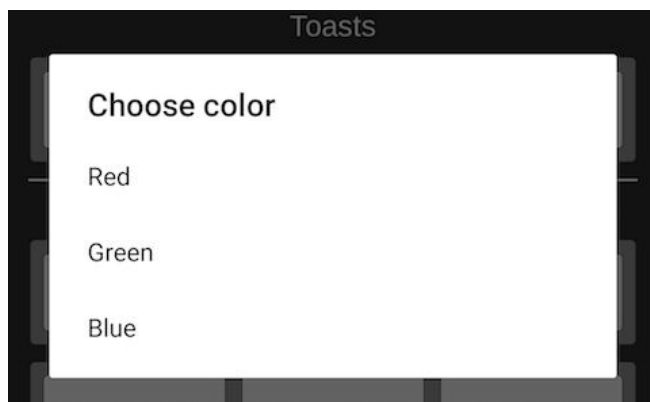
Opens Android [AlertDialog](#) with a list of items to choose one of them. The dialog is dismissed automatically after the user selects one of the options.

Example usage:

```
string[] Colors = { "Red", "Green", "Blue" };
```

```
AlertDialog.ShowChooserDialog("Choose color", Colors,  
    colorIndex => AndroidGoodiesMisc.ShowToast(Colors[colorIndex] + "  
selected"));
```

Result:



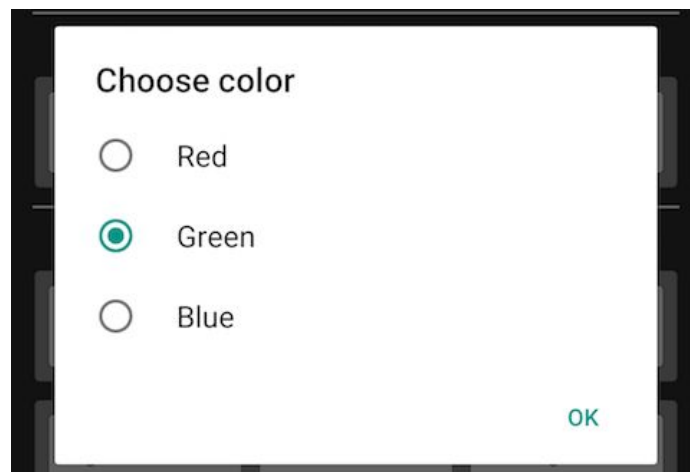
Dialog with radio buttons items chooser

Opens Android [AlertDialog](#) with a list of items to choose one of them with radio buttons. Also has a positive button with callback.

Example usage:

```
string[] Colors = { "Red", "Green", "Blue" };  
int defaultSelectedItemIndex = 1;  
  
AlertDialog.ShowSingleItemChoiceDialog("Choose color", Colors,  
    defaultSelectedItemIndex,  
    colorIndex => AndroidGoodiesMisc.ShowToast(Colors[colorIndex] + "  
selected"),  
    "OK", () => AndroidGoodiesMisc.ShowToast("OK!"));
```

Result:



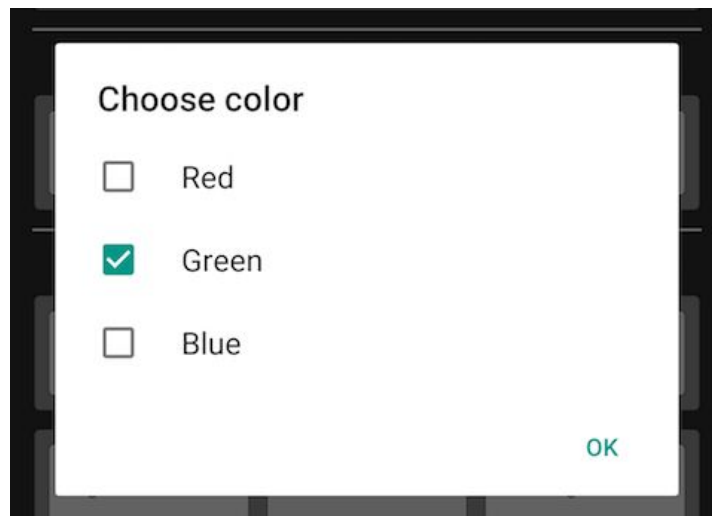
Dialog with check boxes buttons items chooser

Opens Android [AlertDialog](#) with a list of items to choose multiple of them with check boxes. Also has a positive button with callback.

Example usage:

```
string[] Colors = { "Red", "Green", "Blue" };  
bool[] initiallyCheckedItems = { false, true, false }; // second item is  
selected when dialog is shown  
  
AlertDialog.ShowMultiItemChoiceDialog("Choose color", Colors,  
    initiallyCheckedItems,  
    (colorIndex, isChecked) => AndroidGoodiesMisc.ShowToast(Colors[colorIndex]  
+ " selected? " + isChecked), "OK",  
    () => AndroidGoodiesMisc.ShowToast("OK!"));
```

Result:



2. Time and Date Pickers

Android goodies allows you to show default Android [TimePickerDialog](#) and [DatePickerDialog](#).

Showing Time Picker

Show the default Android [TimePickerDialog](#)

Usage Example:

```

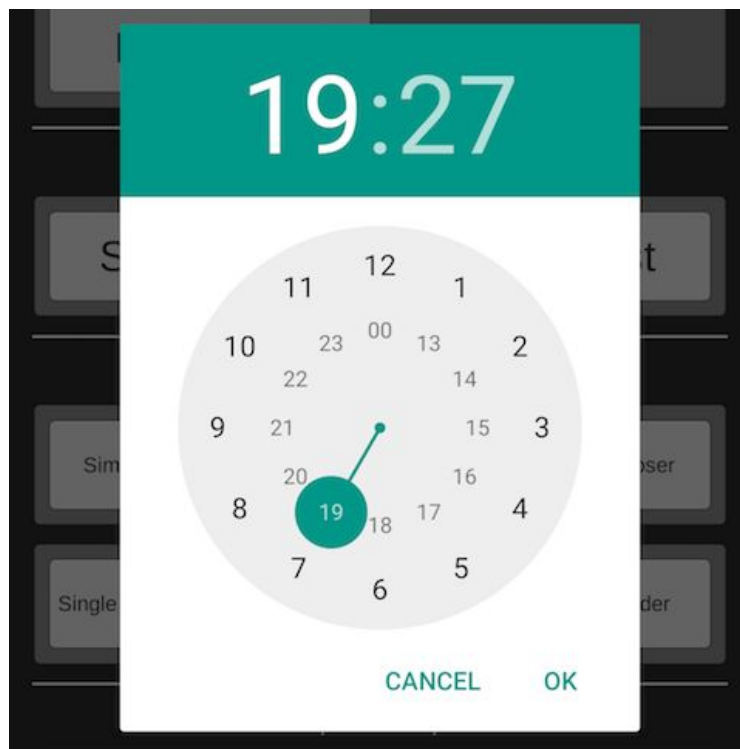
public void OnTimePickClick()
{
    var now = DateTime.Now;
    AndroidDateTimePicker.ShowTimePicker(now.Hour, now.Minute, OnTimePicked,
    OnTimePickCancel);
}

private void OnTimePicked(int hourOfDay, int minute)
{
    var picked = new DateTime(2016, 11, 11, hourOfDay, minute, 00);
    timeText.text = picked.ToString("T");
}

private void OnTimePickCancel()
{
    timeText.text = "Cancelled picking time";
}

```

Result:



Showing Date Picker

Show the default Android [DatePickerDialog](#)

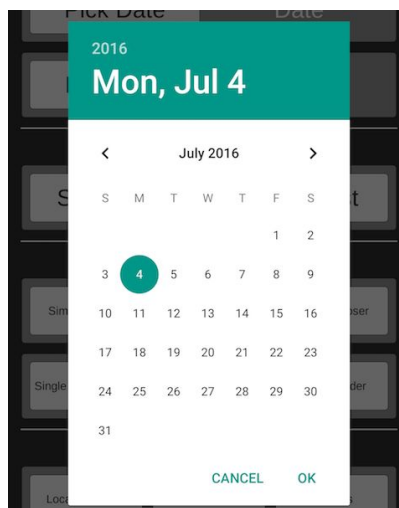
Usage Example:

```
public void OnPickDateClick()
{
    var now = DateTime.Now;
    AndroidDateTimePicker.ShowDatePicker(now.Year, now.Month, now.Day,
    OnDatePicked, OnDatePickCancel);
}

private void OnDatePicked(int year, int month, int day)
{
    var picked = new DateTime(year, month, day);
    dateText.text = picked.ToString("yyyy MMMMM dd");
}

private void OnDatePickCancel()
{
    dateText.text = "Cancelled picking date";
}
```

Result:



3. Native Share

Supported Share intents:

- Native share text
- Sending email
- Sending SMS text
- Tweet

Native share

[Share text using Android intent](#). Shows chooser by default.

Example usage:

```
AndroidShare.ShareText("My subject", "My text to share");
```

Sending Email

Send an email using Android intent. Shows chooser by default.

```
bool withChooser = true;  
AndroidShare.SendEmail(new [] { "x@gmail.com", "hello@gmail.com" }, "subj",  
"body", withChooser);
```

Sending SMS

Send and SMS text message to specified phone number

Example usage:

```
bool withChooser = true;  
AndroidShare.SendSms("123123123", "Hello", withChooser);
```

Tweeting

Android Goodies allows you to tweet directly through official Twitter app. If the app is not installed the call will fallback to the browser one.

Example code:

```
AndroidShare.Tweet("I am tweeting like a boss");
```

4. Open Maps

Android Goodies provides simple and easy interface to [open a map application \(providing coordinates or address\) using intent](#). If there is no application to handle the intent it will log an exception.

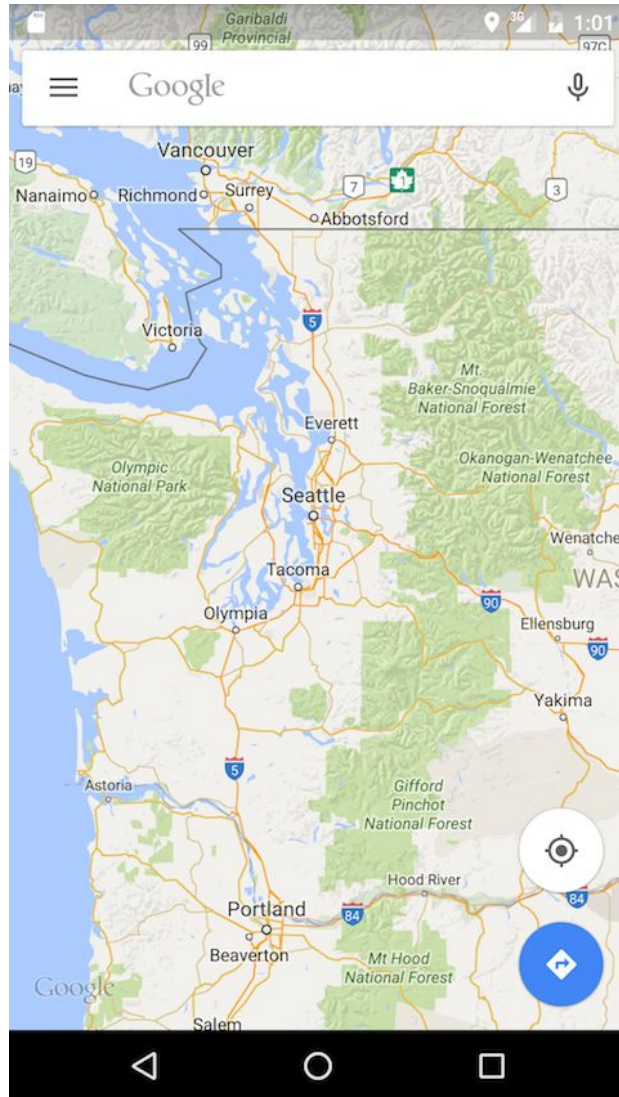
Open location with zoom level

Show the map at the given longitude and latitude at a certain zoom level. A zoom level of 1 shows the whole Earth, centered at the given lat,lng. The highest (closest) zoom level is 23.

Example:

```
AndroidMaps.OpenMapLocation(47.6f, -122.3f, 9);
```

Result:



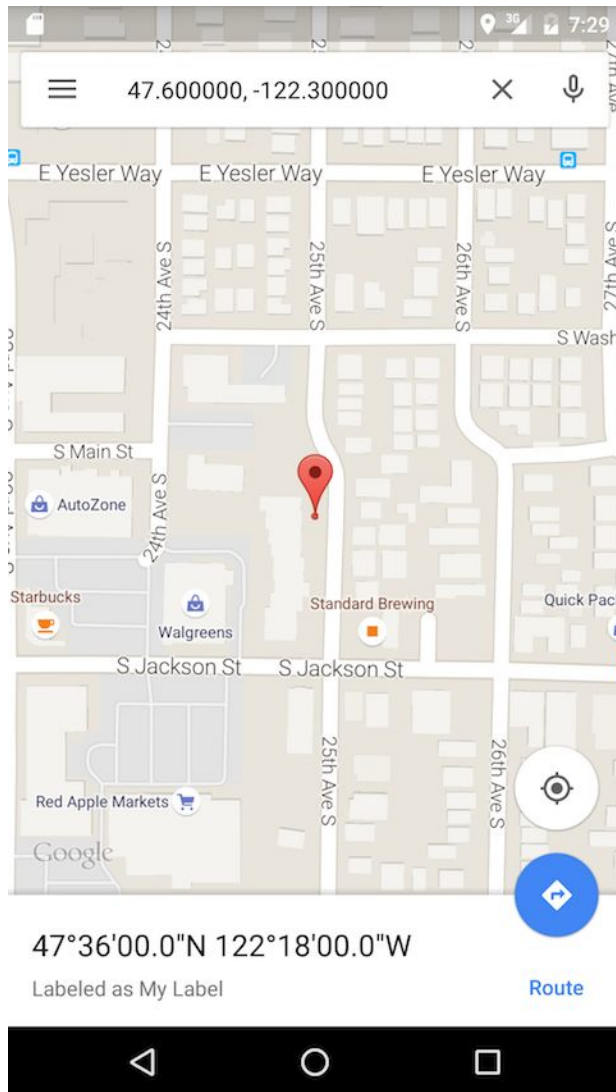
Open location with label

Show the map at the given longitude and latitude with a certain label.

Example:

```
AndroidMaps.OpenMapLocationWithLabel(47.6f, -122.3f, "My Label");
```

Result:



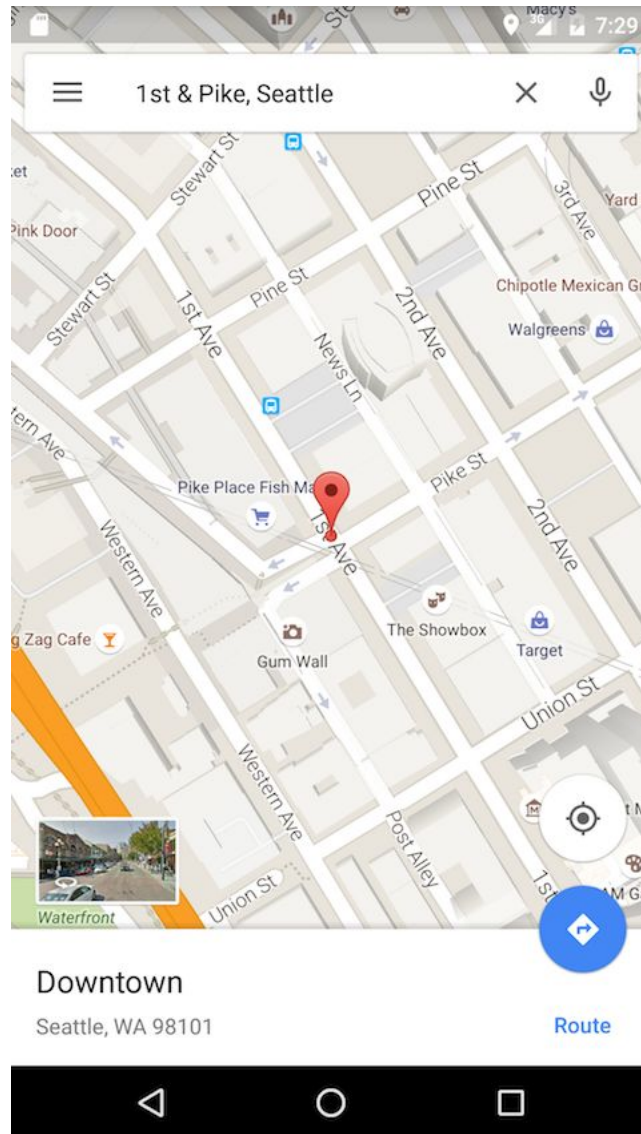
Open location with address

Opens the map location with the provided address.

Example:

```
AndroidMaps.OpenMapLocation("1st & Pike, Seattle");
```

Result:



5. Other useful methods

Android Goodies also contains other methods that might be useful for Android.

Checking if the app with certain package is installed

Knowing the app package you can check whether the app is installed on the device using method `AndroidUtils.IsPackageInstalled(string package)`.

You can find the app package on Google Play like this:



There are also helper methods to check if official Twitter app is installed, if user has any email app client and if user has any SMS app or maps client.

- `AndroidShare.IsTwitterInstalled()`
- `AndroidShare.UserHasEmailApp()`
- `AndroidShare.UserHasSmsApp()`
- `AndroidMaps.UserHasMapsApp()`

Showing Toast

You can show toast (short or long) like this:

```
AndroidGoodiesMisc.ShowToast("hello long!",  
AndroidGoodiesMisc.ToastLength.Long);
```

Immersive Mode

To enable immersive mode invoke:

```
AndroidGoodiesMisc.EnableImmersiveMode();
```

Note that Unity 5 has immersive mode enabled by default, so if you are using Unity 5 or higher, this method is redundant.

Other Device info

From `android.provider.Settings.Secure`:

- `ANDROID_ID` - A 64-bit number (as a hex string) that is randomly generated when the user first sets up the device and should remain constant for the lifetime of the

user's device. The value may change if a factory reset is performed on the device.

You can retrieve the following properties from `android.os.Build` class:

- `DEVICE` - The name of the industrial design.
- `MODEL` - The end-user-visible name for the end product.
- `PRODUCT` - The name of the overall product.
- `MANUFACTURER` - The manufacturer of the product/hardware.

You can retrieve the following properties from `android.os.Build.VERSION` class:

- `BASE_OS` - The base OS build the product is based on.
- `CODENAME` - The current development codename, or the string "REL" if this is a release build.
- `INCREMENTAL` - The internal value used by the underlying source control to represent this build.
- `PREVIEW_SDK_INT` - The developer preview revision of a prerelease SDK.
- `RELEASE` - The user-visible version string.
- `SDK_INT` - The user-visible SDK version of the framework; its possible values are defined in `Build.VERSION_CODES`.
- `SECURITY_PATCH` - The user-visible security patch level.