# Bit Toys Plugin:  Getting Started



Version 1.30 - 6/2/2017
support@bit-toys.com

## Step 1:  Getting Started

Thank you for choosing Bit Toys to link your physical merchandize and digital content.  In this document, you will learn how to integrate the Bit Toys Plugin into your existing Unity3D projects. Please review **`Bit Toys Plugin Licensing Agreement.PDF`** and refer to **`Bit Toys Plugin Integration Guide.pdf`** for technical details related to each feature and API.  To start, be sure you have the followings:

- Unity version 5 or higher
- Developer ID assigned by Bit Toys
- Application ID assigned by Bit Toys
- Test tags provided by Bit Toys:  NFC, QR, Audio Tag, etc.
- (Optional) Bit Toys Bluetooth NFC Reader

## Step 2:  Import Files

1. Import **`BitToysPlugin1.30.unitypackage`**
2. If applicable, overwrite previous version of the plugin

## Step 3:  Set Up Security Profiles

### Android:

The Bit Toys Platform is secured by using the application's MD5 Signature from Android apps.  This signature must be added to the Bit Toys Platform for API access.

**Getting the Signature:**
Obtain the MD5 Signature by reading the application's Keystore file.  If you already have a Keystore, please proceed to step 2:

1. **Creating a Keystore and signing your app:**
    - In the Unity editor open File ➜ Build Settings...

- Select Android platform and click Player Settings...
- In the Inspector, expand the Publishing Setting tab
- Check the Create New Keystore box, then click on Browse Keystore
- Name and store the created file in a secured location
- Enter and confirm a Keystore Password
- In the Alias drop down, Create a New Key
- Fill out the form with Alias and Alias Password. This Alias will be used in Step 2, and the Alias Password here doesn't have to be the same as the Keystore Password

2. **Retrieving the Signature:**
   - Search your computer for `keytool.exe` located in the Java bin folder, for example: `C:\Program Files\Java\jdk1.7.0_79\bin`
   - Open a terminal window in keytool's folder *(Shift + Right Click ➜ Open command window here)*
   - Enter the following command with your Alias and Path to Keystore:
     ```
     keytool -exportcert -alias Your_Alias -keystore "C:\path\to\keystore\YourKeystoreName.keystore" -list -v
     ```
   - Enter the Keystore Password, it would print keystore info to the console
   - Save the MD5 line and submit it to Bit Toys

## iOS:

Similar to Android, the iOS is secured using the app's Bundle ID. It must be added to the Bit Toys Platform for API access.

To set or retrieve the Bundle ID in Unity:

1. Navigate to File ➜ Build Settings... ➜ iOS ➜ Player Settings...
2. In the Inspector, expand the Other Settings tab
3. Under Identification, set your Bundle ID under Bundle Identifier, for example: `com.ExampleCompany.ExampleApp`
4. Send this Bundle ID to Bit Toys

## Submit Info to Bit Toys:

Once Android MD5 and iOS Bundle ID are configured, submit them to Bit Toys via https://onetimesecret.com/ with a Passphrase. Please send the link generated by https://onetimesecret.com/ and the Passphrase *separately and through different means* (e.g. email, Skype, Line, etc.) as a security precaution.

In return, Bit Toys will generate and issue an Authentication Key. Please keep this Authentication Key in a secured location. For security reasons, Bit Toys does not store

this key. If lost or compromised, Bit Toys can only regenerate a new key, and your apps would have to be updated in order to maintain access to Bit Toys features.

## Step 4: Scene Setup

1. If you don't have a working scene inside Unity, create a new scene
2. Under Hierarchy, select Create ➜ Create Empty
3. Rename this newly created GameObject to Bit Toys Object
4. Add the BitToys script to the new Bit Toys Object, it can be found under `Assets/Plugins/Bit Toys/BitToysUnityDLL/BitToys`
5. **NOTE**: This Bit Toys Object needs to persist throughout the life of your application, it automatically calls `DontDestroyOnLoad()` on awake

## Step 5: Developer ID, Application ID & Authentication Key

Bit Toys should provide you with a Developer ID, an Application ID, and an Authentication Key. Find the Bit Toys Object in your scene and enter appropriate values.

## Step 6: Activate Needed Features

The Bit Toys Object also has a number of checkboxes, one for each of the supported features. By default, they're all unchecked. Check the ones that the application will support. This will tell the plugin that the application will use those features.

## Step 7: Android Manifest

For Android applications, **`BitToysPlugin1.30.unitypackage`** comes with an Android Manifest file. This manifest must be present for the Bit Toys Plugin to function. The manifest can be found in the `Assets/Plugins/Android/` folder. By default, the imported manifest will contain permissions for each of the available Bit Toys Plugin features. If some features aren't used, their permissions can be removed. Permissions are located at the bottom of the Android Manifest file:

```
<!-- Native NFC -->
  <uses-feature android:name="android.hardware.nfc" android:required="true" />
  <uses-permission android:name="android.permission.NFC" />
```

```
<!-- QR -->
  <uses-feature android:name="android.hardware.camera.AUTOFOCUS" />
  <uses-permission android:name="android.permission.CAMERA" />

<!-- Audio Tag -->
  <uses-permission android:name="android.permission.RECORD_AUDIO" />
  <uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />

<!-- Bluetooth NFC Reader -->
  <uses-permission android:name="android.permission.BLUETOOTH" />
  <uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
```

## Step 8: Define Toy Style IDs

A Style ID is a unique identifier for a specific type of toy. Example: If you have 5 toys, 2 Green Dinosaurs, 2 Red Dinosaurs and 1 Blue Dinosaur. The 2 Green Dinosaurs will share the same Style ID, the 2 Red Dinosaurs will have another Style ID, and the Blue Dinosaur will have a 3rd Style ID.

Your sample tags come with pre-defined Style ID. For your own product lines, please work with Bit Toys on defining your list of Style IDs.

## Step 9: You are Ready!

You are now setup and ready to start using the Bit Toys Platform. Refer to the Integration Guide for API details.

## Optional:  Stripping Assemblies

If you set your project code stripping level to: Strip Assemblies, it may strip the assembly System.Security that the Bit Toys Plugin relies on and cause issues.  To prevent these issues, create a `link.xml` file in the root `Assets` folder of your Unity Project.  The `link.xml` file should contain the following lines:

```
<linker>
  <assembly fullname="mscorlib">
   <type fullname="System.Security.Cryptography.*" preserve="all"/>
  </assembly>
</linker>
```

This stops the assembly stripper from stripping what the Bit Toys Plugin needs.

### Checklist:

| | |
|---|---|
| | Add Developer ID to the Bit Toys Object |
| | Add Application ID to the Bit Toys Object |
| | **Android**: Submit MD5 Signature to Bit Toys |
| | **iOS:** Submit Bundle ID to Bit Toys |
| | Add Authentication Key to Bit Toys Object |
| | Select Desired Features |
| | Make edits to the Android Manifest, if applicable |
| | Define Style IDs of your toys with Bit Toys |