

Unity5 サウンドクラスライブラリ

# USndViewer アプリ説明書

最終更新日: 2017/10/16

## ■ 目次

- ・ 概要
- ・ Unity インストール時の注意
- ・ AssetBundle の作り方
- ・ Android へデータを転送する準備
- ・ Android へのデータ転送方法
- ・ iOS へのデータ転送方法
- ・ 使い方
- ・ AssetBundle 生成時にエラーが出る時の対処法

## ■ 概要

UnityEditor の USndViewer を使用して端末で動作をチェックすることができます。

Editor 上では Resources にデータを置いていますが、USndViewer アプリでは AssetBundle 化したデータを使用します。

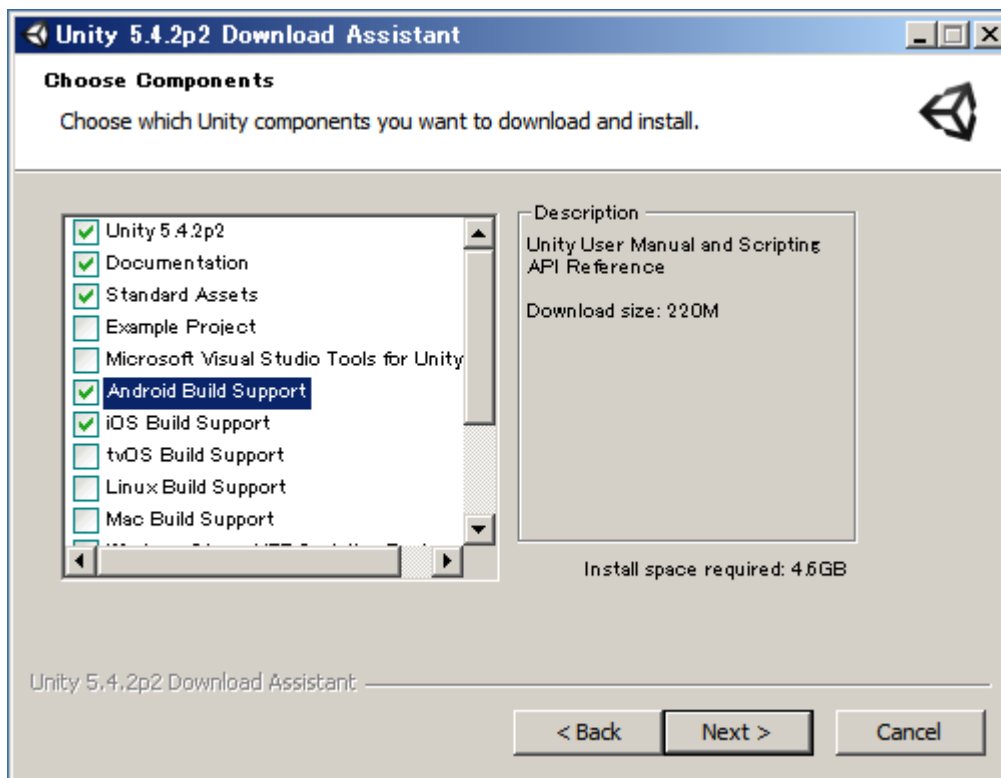
UnityEditor 上で USndViewer を動かす方法は USnd Documentation のチュートリアルをご確認ください。

## ■ Unity インストール時の注意

AssetBundle を作成する必要があるため、UnityEditor の Pro ライセンスが必要です。  
UnityEditor のメニューの Help->About Unity の右下に LicenseType の記載があります。  
そこに Android Pro、iOS Pro の記載があれば AssetBundle の作成が行なえます。

AssetBundle を作成するには各ターゲット向けにターゲットをスイッチできるようになっている必要があります。

インストール時に追加するコンポーネントを選ぶ画面で、Android Build Support, iOS Build Support を選択するようにしてください。後からでもインストールは可能です。



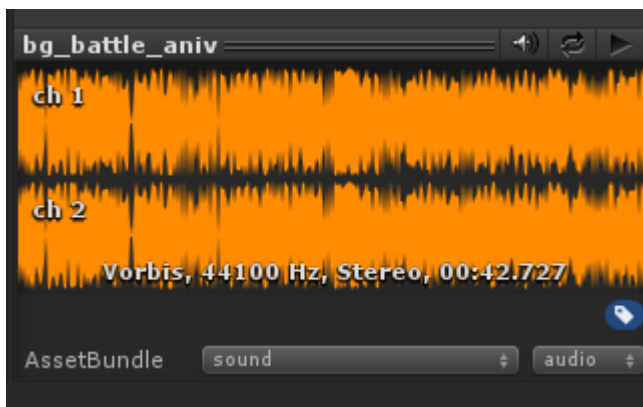
## ■AssetBundle の作り方

まずオーディオファイルを AssetBundle 化します。Resources/AudioClip 以下に置いたものは AssetBundle の拡張子を.audio にします。

UnityEditor の Project->設定を行なうオーディオファイルを選択->Inspector でどの AssetBundle に入れるかを設定します。複数選択でまとめて設定可能です。

Inspector の下部に AssetBundle と書かれてる部分の隣の None と書かれている領域をクリックします。

リストが表示されるので、既に登録済みの名前がある場合は設定したい名前を、新規で名前を設定する場合は New をクリックして文字を入力します。



上図の場合は「sound」という名前を設定しました。オーディオファイルの AssetBundle 名は特に指定はないので自由に設定してください。

次に拡張子の分を設定します。先ほど設定したファイル名の右側の小さな None と書かれている領域が拡張子です。AssetBundle 名と同じ要領で設定してください。オーディオファイルは「audio」という名前にしてください。このルールは USndViewer で読み込むためのものなので、すべてのアプリの規則ではありません。

どのファイルがどの AssetBundle に含まれるかはファイル単位で指定します。フォルダを選択しての指定はできませんが、フォルダ選択中は Inspector にファイルの情報が残っているので間違えて設定しないように注意してください。

オーディオファイル以外のファイルもそれぞれ AssetBundle にします。名前と含める AssetBundle が決まっているので下記の表を参考に設定を行なってください。

フォルダ	AssetBundle ファイル名	AssetBundle 拡張子
AudioClip	任意	audio
MasterBinary	binary	usnd
CategoryBinary	binary	usnd
LabelBinary	binary	usnd
MasterXML	xml	usnd
CategoryXML	xml	usnd
LabelXML	xml	usnd
NameList	namelist	usnd
AudioMixer	mixer	usnd

設定が終わったら、書き出したいターゲットにプロジェクトを変更します。

例えば Android の AssetBundle を生成するときは、File->Build Settings を選択し、Platform のリストから Android を選択してリストの下にある Switch Platform ボタンを押します。Android が有効になっていない場合は Build Support が入っていないので本ドキュメントの Unity インストール時の注意点を参考にインストールを行なってください。

書き出したいターゲットにプロジェクトを変更したら、Project ウィンドウ上で右クリックし、一番下に表示されている「Build AssetBundle」をクリックします。

これで USndViewer 直下 (Assets と同じ階層) にターゲットの AssetBundle が生成されます。

ターゲットごとの AssetBundle が必要なので、iOS の場合はターゲットを iOS に切り替えて「Build AssetBundle」を行なってください。

「Build AssetBundle」が表示されない場合は、Assets¥Editor に CreateAssetBundle.cs が存在するか確認してください。CreateAssetBundle.cs がないとメニューは表示されません。

Build AssetBundle が失敗するときは、項目「AssetBundle 生成時にエラーが出るときの対処法」をご確認ください。

## ■ Android へデータを転送する準備

PC に接続して直接フォルダを置ける場合は、`usndviewer` という名前のフォルダを作成し `AssetBundle` を配置します。

直接データを転送できない場合は、`adb` コマンドを使用します。

`¥¥qute¥group¥sd_dev3¥03_tools¥android_adb`

を取得するか、AndroidStudio をインストールして AndroidSDK をダウンロードすると `adb` が入手できます。Mac の場合は AndroidStudio をインストールしてください。

入手したらパスを設定します。

Windows:

コンピューターを右クリック→プロパティ または コントロールパネル→システムを開き、「コンピューター名、ドメインおよびワークグループの設定」の設定の変更を開きます。詳細設定タブの環境変数を開き、システム環境変数の `PATH` の一番後ろに `platform-tools` フォルダまでのパスを追加します。

Mac:

ターミナルを起動し、下記を実行します。

`pico ~/.bash_profile`

編集・保存できないときは `sudo pico ~/.bash_profile` と打ちます。

`export ANDROID_SDK_PF_TOOLS=/User/{usrXX}/Library/Android/sdk/platform-tools`

`export PATH=$PATH:${ANDROID_SDK_PF_TOOLS}`

入力したら `control+X` を押します。Yes/No を聞かれるので Y と入力して `Enter` を押すとセーブされます。次にターミナルを起動したときに反映されるので、ターミナルを終了してください。

`adb` コマンドにパスが通し終わったら端末が認識されるようにします。

Windows の場合は `adb` ドライバをインストールします。サウンド共有端末は `android_adb` フォルダの `readme.txt` にインストール方法が記載されているので参照してください。

Mac はドライバなしで接続できます。

新しい端末の場合は、端末名と `adb` ドライバで検索すればインストール方法が出てきます。

次に端末の開発者向けオプションを有効にします。借りてきた端末などは **OFF** になっている場合があります。設定に開発者向けオプションという項目がない場合は、設定→端末情報を開き、ビルド番号と書かれているエリアを 7 回連続タップします。開発者向けオプションが表示されたら、**USB デバッグ** を **ON** にします。

以上が終わったら端末を **PC** に接続します。

接続後、コマンドプロンプトまたはターミナルを開き、

**adb devices**

と入力します。**List of devices attached** という文字の下に表示があれば認識されています。**offline** となっていたら **USB デバッグ** を切り替えて接続しなおしてください。

## ■ Android へのデータ転送方法

Android 端末を接続し、コマンドラインまたはターミナルを立ち上げて、転送したいファイルが置いてあるフォルダと同じ階層に移動します。

USndViewer で AssetBundle を作成した場合、USndViewer¥AssetBundles の下にターゲットごとに生成されるので、

```
cd D:¥USndViewer¥AssetBundles
```

というように入力し、**Enter** を押します。

以下で Android フォルダごと端末に転送します。

```
adb push Android/. /sdcard/usndviewer/projectX/
```

特定のファイルだけコピーしたいときは、

```
adb push Android/sound.audio /sdcard/usndviewer/projectX/
```

と入力すると **sound.audio** だけ転送されます。

**usndviewer** 直下のフォルダを選択するようになっているので、プロジェクト名などでフォルダをわけて利用してください。

以上で Android への転送は完了です。

## ■iOS へのデータ転送方法

iOS 端末への転送は iTunes で行ないます。iTunes が入っていない場合はインストールしてください。

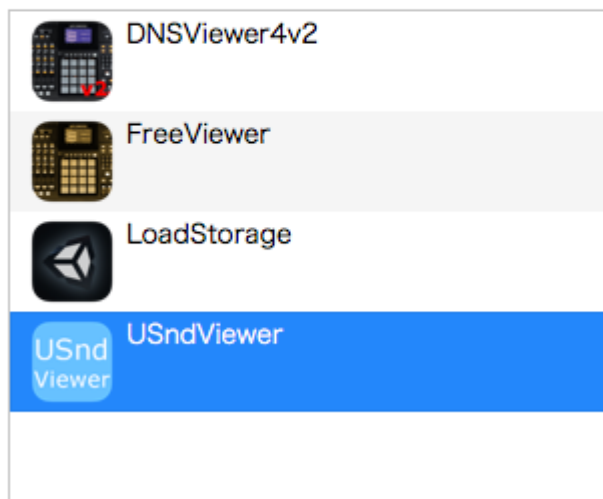
端末を接続し、端末詳細を開いて左側のメニューから **App** を選択します。

右画面に表示されているインストールアプリの管理画面を下にスクロールするとファイル共有が有効になっているアプリケーション一覧が表示されます。

### ファイル共有

以下の App では、iPhone とこのコンピュータとの間

#### App



USnd のアイコンを選択した状態で、一覧の右側のエリアに転送するファイルの入ったフォルダごとドラッグ&ドロップします。

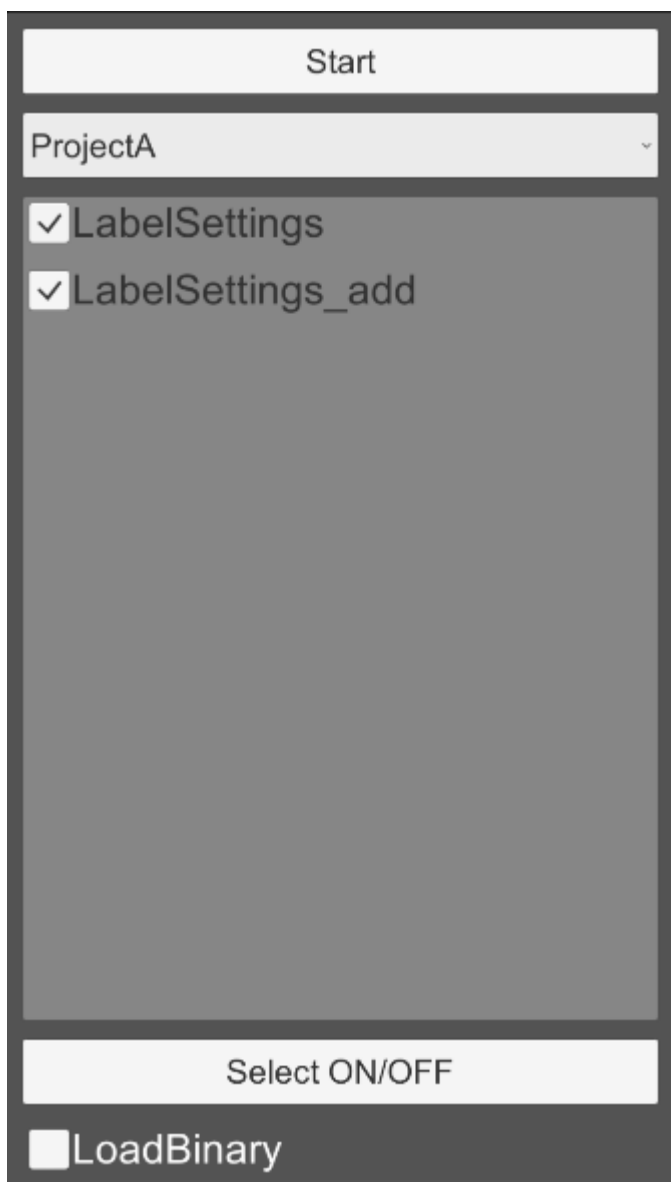
以上で iOS への転送は完了です。

## ■使い方

下図のアイコンのアプリを立ち上げます。インストールされていない場合はご連絡ください。



立ち上げると一番目に検出されたフォルダに入っているラベル設定ファイル名一覧 (NameList/LabelName.txt の内容)が表示されます。



**Start** の下の「**ProjectA**」と書かれた部分でフォルダを選択します。**ProjectA** と書かれてるエリアをタップするとフォルダ一覧が表示されます。

目的のフォルダが表示されていない場合はアプリを再起動してください。



読み込みたいフォルダ名をタップするとラベル設定ファイルの一覧が更新されます。

読み込みたいラベル設定ファイルのチェックのみ残して **Start** ボタンを押すと **XML** ファイルを読み込んで起動します。

バイナリ形式を使用するときは一番下の **LoadBinary** にチェックを入れてから **Start** を押してください。

アセットバンドルを更新したときはアプリを立ち上げなおしてください。

以上

## ■ AssetBundle 生成時にエラーが出るときの対処法

下記のようなエラーが出て失敗する場合、同じ AssetBundle の中に同名のファイル名が存在しています。



Unity は Resources 以下に置かれている場合は、異なるフォルダにある同名のリソースは別リソースとして扱いますが、同名ファイルが同じ AssetBundle に含まれるとエラーになります。その場合、上記のようなエラーが出ますがどのファイルが競合しているかはエラーメッセージに表示されません。

Assets/Editor 以下に CheckFileName.cs を置いておくと、メニューの Assets または右クリックメニューに「Check FileName」が追加されます。

Check FileName を実行すると、Resources 以下に同名ファイルが二つ以上存在するファイルのリストを出力します。AssetBundle の出力先と同じフォルダに CheckFileName.txt が生成されます。

このファイルから同名ファイルが同じ AssetBundle に指定されていないかチェックしてください。