

【C2】Delphi/iOSテクニカルセッション

モバイル開発始めるなら今でしょ！ Delphi iOSアプリ開発講座

株式会社シリアルゲームズ
取締役 細川 淳

Delphi iOSアプリ開発講座
はじめに

はじめに - 自己紹介

◎ 株式会社シリアルゲームズ

- クライアント・サーバー型アプリケーションをワンストップで提供しています。
 - サーバサイドアプリケーション
 - エンターテインメント系アプリケーション
 - ちなみに、iOS / Android の技術者募集中です。

◎ そこで主にクライアント OS のアプリを開発しています。

- クライアント OS とは
 - Windows / Android / iOS です。
 - 最近、MacOS X も少し。

はじめに - 本日の内容

- ◎ FM³ とは
- ◎ FM³ による iOS 開発
 - カメラデバイス
 - GPS
 - 配置マネージャ
 - StartUp Copy

注意

- ◎ このセッション資料は RAD Studio XE4 Beta 版で作成されています。
- ◎ 製品版とは異なる場合があります。
- ◎ 製品版と異なる場合、口頭で説明します。

Delphi iOSアプリ開発講座

FM³ - FireMonkey 3

FM³ とは

◎ FireMonkey

- XE2 で搭載された新しいコンポーネントフレームワーク

◎ FireMonkey2

- XE3 で搭載された FireMonkey のバージョン2

◎ FireMonkey3

- XE4 で iOS に対応
- 秋頃発売？ の XE5 ？ では Android に対応
- One Source で両対応アプリが作れる！

FM³ がとにかくスゴい

他の環境では
こうはいかない！

◎ ネイティブ！だから速い！

- Intel / ARM のネイティブコードが出力される！
- 著名なマルチプラットフォーム環境は、全部 VM で動く
 - Unity / Adobe Air など
- 側(がわ)ネイティブアプリと比べたら段違い！
 - コンテンツが Web で提供されているもの



ネイティブ部分に WebView が置いてあるだけ！

FM³ がとにかくスゴい

他の環境では
こうはいかない！

◎ 2D / 3D の両方に対応している！

- iOS / Android では OpenGL で描かれる
- 3D と 2D を混在できる
- GUI パーツが豊富！

➡ つまり GUI 構築も超楽！

- Unity は 2D GUI が苦手！
- ゲームも FM3 で制作できそう
 - Box2D などの物理演算エンジンも組み込める
 - 個人的に研究中

FM³ がとにかくスゴい



他の環境では
こうはいかない！

◎ 単一の環境！

- 単一の IDE / 言語 / Framework(FM3) で全プラットフォームのアプリが作れる！
 - MS / Apple / Google が提供している開発環境は、IDE も 言語も Framework も全部別！

◎ 各 OS のネイティブ API も呼べる！


- いざとなれば OS のネイティブ API を呼べる
 - {\$IFDEF MSWINDOWS} や {\$IFDEF IOS} で分けられる
 - 他のマルチプラットフォーム環境では、呼べないか、特殊な方法で呼ぶ必要がある。

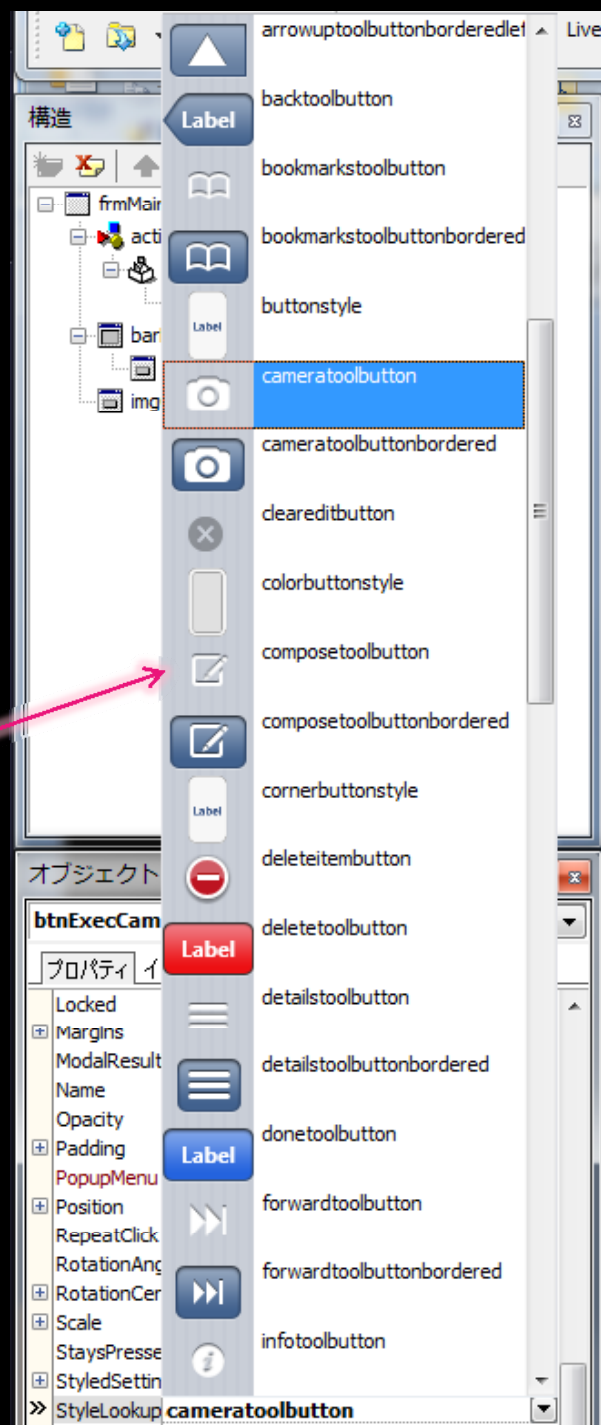
FM³ がとにかくスゴい

- ◎ OS の GUI デザインからの解放
 -かと思いきや
- ◎ OS の GUI デザインも表示できる！

StyleLookup で OS GUI を表示できる！
見た目はネイティブ GUI と同じ！

FM3 が OpenGL を使って描いたコントロールは
ネイティブの GUI と寸分違わぬため、これを

 **ピクセル・パーフェクト**
と呼んでいる



FM³ がとにかくスゴい

他の環境では
こうはいかない！

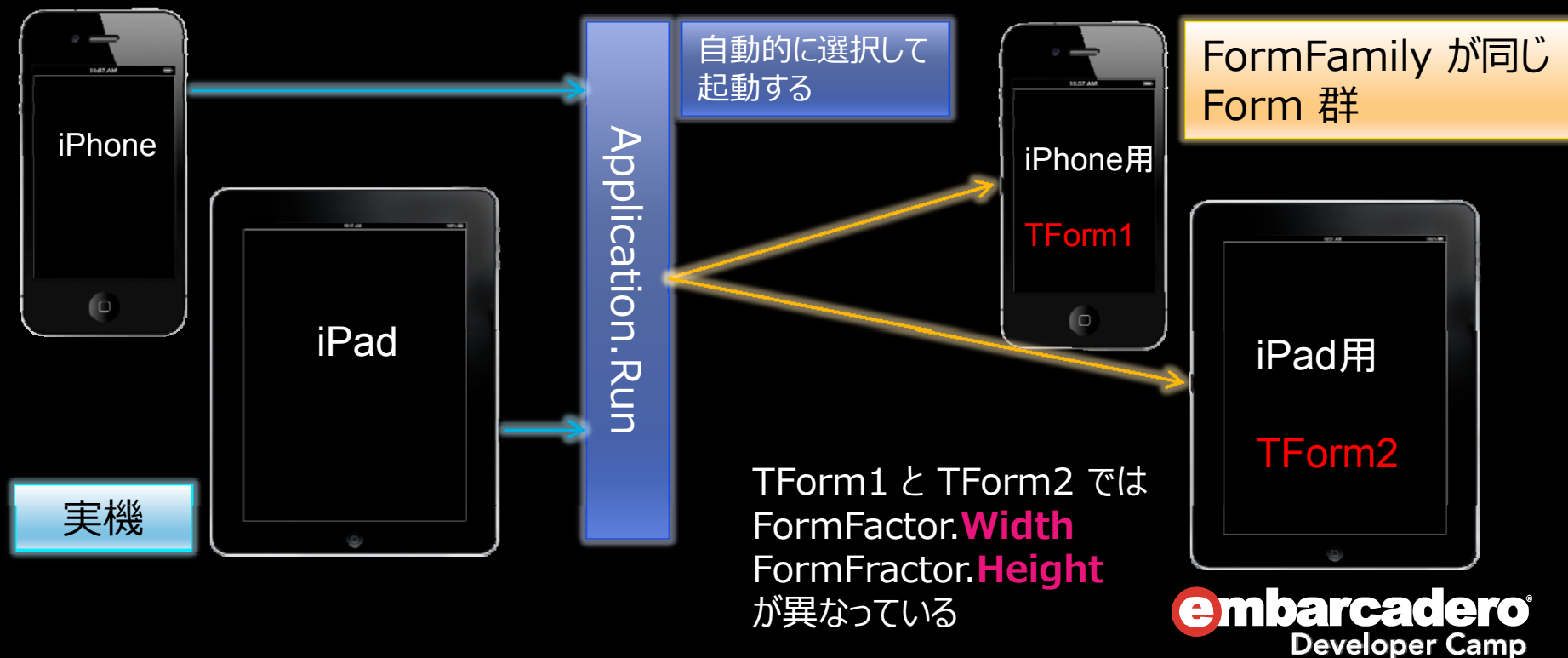
- ◎ コーディング不要で Retina に対応！
 - つまり、**コントロールの配置に悩まない**！
 - 解像度がめちゃくちゃたくさんある **Android** でも安心！
 - ただし、Retina 対応の Style が必要
 - 提供されている Style は Retina に対応している
- ◎ 豊富な**メディアライブラリ・センサーライブラリ**！
- ◎ 豊富な**アニメーション・トランジション・エフェクト**コンポーネント！

FM³ がとにかくスゴい

他の環境では
こうはいかない！

◎ FormFamily & FormFactor で解像度の問題も解決！

- 実機の解像度によって、起動する Form を変更するシステム！



FM³ がそうでもないところ

- ◎ FormFamily を使って TForm1 と TForm2 が別れてもコードは共通化できない
 - CommonUnit を作って、その関数を呼ぶようにする
 - CreateNew / LoadRes を使って自前でやればできる気がするけど.....
- ◎ サブフォームをアニメーション付きで呼び出せない

```
procedure TForm1.Button1Click(Sender: TObject);  
begin  
    Form2.Show; // 瞬時に切り替わる  
end;
```

- ただし、Unity など他の環境でも同じ
 - また、Android では、それが普通
- Formデザイナー上に Panel を複数置いて切り替える方法で代替可能だけど.....

➡ Style / StyleLookup / FormFamily デモ

Delphi iOSアプリ開発講座

カメラを使う - デフォルトのカメラ

簡単なカメラを動かしてみる

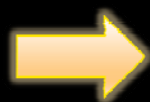
◎ 要件

- デフォルトのカメラを動かす
- ボタンを押してカメラを起動
- 撮影したらイメージを取得
- 取得したイメージを表示する

簡単なカメラを動かしてみる

◎ 手順

- TToolBar を配置
 - StyleLookup = bottomtoolbar
- TButton を配置
 - align = alCenter
 - StyleLookup = cameratoolbuttonbordered
- TActionList を配置
 - TakePhotoFromCameraAction を生成
 - TButton.Action と結びつける
- TImageControl を配置
 - align = alClient
 - TakePhotoFromCameraAction.DidFinishTaking でイメージを表示



この手順に従って、実際に作ってみます。

簡単なカメラを動かしてみる

```
procedure TForm1.TakePhotoFromCameraAction1DidFinishTaking(Image: TBitmap);  
begin  
    ImageControl1.Bitmap.Assign(Image);  
end;
```

Delphi iOSアプリ開発講座

カメラを使う - リアルタイム・カメラ

カメラを操る

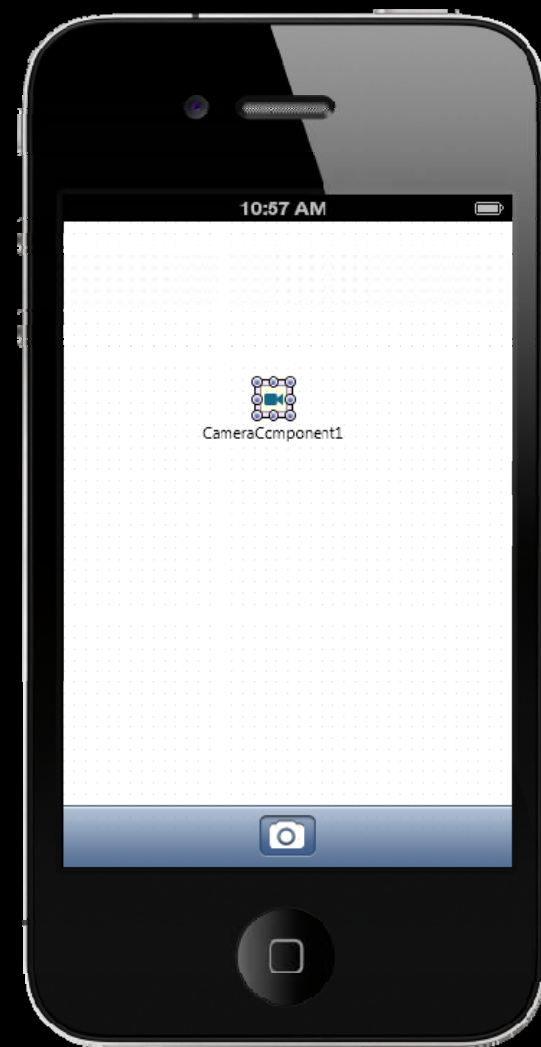
◎ 要件

- カメラデバイスからイメージを得る
- 得たイメージを表示する

カメラを操る

◎ 手順

- TToolBar を配置
 - StyleLookup = bottomtoolbar
- TButton を配置
 - align = alCenter
 - StyleLookup = cameratoolbuttonbordered
- TImageControl を配置
 - align = alClient
- TCameraControl を配置
 - OnSampleBufferReady を実装



➡ この手順に従って、実際に作ってみます。

カメラを操る

```
procedure TForm1.Button1Click(Sender: TObject);
begin
    CameraComponent1.Active := not CameraComponent1.Active;
end;

procedure TForm1.CameraComponent1SampleBufferReady(
    Sender: TObject;
    const ATime: Int64);
begin
    TThread.Synchronize(
        TThread.CurrentThread,
        procedure
        begin
            CameraComponent1.SampleBufferToBitmap(ImageControl1.Bitmap, True);
        end
    );
end;
```

Delphi iOSアプリ開発講座

LocationSensor を使う

LocationSensor を操る

◎ 要件

- LocationSensor.OnLocationChanged で座標を得る
- 得た座標を表示する

LocationSensor を操る

◎ 手順

- TToolBar を配置
 - StyleLookup = bottomtoolbar
- TButton を配置
 - align = alCenter
 - StyleLookup = cameratoolbuttonbordered
- TMemo を配置
 - align = alClient
- TLocationSensor を配置
 - OnLocationChanged を実装



➡ この手順に従って、実際に作ってみます。

LocationSensor を操る

```
procedure TForm1.Button1Click(Sender: TObject);
begin
    LocationSensor1.Active := not LocationSensor1.Active;

    if (LocationSensor1.Active) then begin
        Memo1.Lines.Add('Start');
        Button1.Text := 'Stop';
    end
    else begin
        Memo1.Lines.Add('Stop');
        Button1.Text := 'Start';
    end;
end;

procedure TForm1.LocationSensor1LocationChanged(
    Sender: TObject;
    const OldLocation, NewLocation: TLocationCoord2D);
var
    Pos: String;
begin
    Pos := Format('%.8f, %.8f', [NewLocation.Latitude, NewLocation.Longitude]);
    Form1.Memo1.Lines.Add(Pos);
end;
```

Delphi iOSアプリ開発講座

MotionSensor

MotionSensor を操る

◎ 要件

- Timer を使って一定時間毎に座標を取得する
 - MotionSensor には OnLocationChanged のようなイベントは存在しない
 - 値の揺らぎが大きいため

MotionSensor を操る

◎ 手順

- TToolBar を配置
 - StyleLookup = bottomtoolbar
- TButton を配置
 - align = alCenter
 - StyleLookup = cameratoolbuttonbordered
- TMemo を配置
 - align = alClient
- TMotionSensor を配置
- TTimer を配置
 - OnInterval イベントを実装



➡ この手順に従って、実際に作ってみます。

MotionSensor を操る

```
procedure TForm1.Button1Click(Sender: TObject);
begin
    MotionSensor1.Active := not MotionSensor1.Active;

    if (MotionSensor1.Active) then begin
        Button1.Text := 'Stop';
        Timer1.Enabled := True;
    end
    else begin
        Button1.Text := 'Start';
        Timer1.Enabled := False;
    end;
end;
```

MotionSensor を操る

```
procedure TForm1.Timer1Timer(Sender: TObject);
```

```
var
```

```
    Data: String;
```

```
begin
```

```
    // 加速度が使える、表示する
```

```
    if
```

```
    (
```

```
    [
```

```
        TCustomMotionSensor.TProperty.AccelerationX,
```

```
        TCustomMotionSensor.TProperty.AccelerationY,
```

```
        TCustomMotionSensor.TProperty.AccelerationZ
```

```
    ]
```

```
    * MotionSensor1.Sensor.AvailableProperties
```

```
    <> []
```

```
)
```

```
then
```

```
    Data := Data +
```

```
        Format('Accelerate X: %.8f', [MotionSensor1.Sensor.AccelerationX]) + sLineBreak +
```

```
        Format('Accelerate Y: %.8f', [MotionSensor1.Sensor.AccelerationY]) + sLineBreak +
```

```
        Format('Accelerate Z: %.8f', [MotionSensor1.Sensor.AccelerationZ]) + sLineBreak +
```

```
        sLineBreak;
```

使える Property の集合

sLineBreak は
プラットフォーム毎の
改行文字列

MotionSensor を操る

```
// 角加速度が使えれば、表示する
if
(
    [
        TCustomMotionSensor.TProperty.AngleAccel X,
        TCustomMotionSensor.TProperty.AngleAccel Y,
        TCustomMotionSensor.TProperty.AngleAccel Z
    ]
    * MotionSensor1.Sensor.AvailableProperties
<> []
)
then
    Data := Data +
        Format(' Angle X: %.8f', [MotionSensor1.Sensor.AngleAccel X]) + sLineBreak +
        Format(' Angle Y: %.8f', [MotionSensor1.Sensor.AngleAccel Y]) + sLineBreak +
        Format(' Angle Z: %.8f', [MotionSensor1.Sensor.AngleAccel Z]) + sLineBreak +
        sLineBreak;
```

MotionSensor を操る

```
// Motion があれば、表示する
if
(
    [TCustomMotionSensor.TProperty.Motion]
    * MotionSensor1.Sensor.AvailableProperties
    <> []
)
then
    Data := Data +
        Format('Motion %.8f', [MotionSensor1.Sensor.Motion]) + sLineBreak;

// 速度があれば、表示する
if
(
    [TCustomMotionSensor.TProperty.Speed]
    * MotionSensor1.Sensor.AvailableProperties
    <> []
)
then
    Data := Data +
        Format('Speed %.8f', [MotionSensor1.Sensor.Speed]);

Memo1.Lines.Text := Data;
end;
```

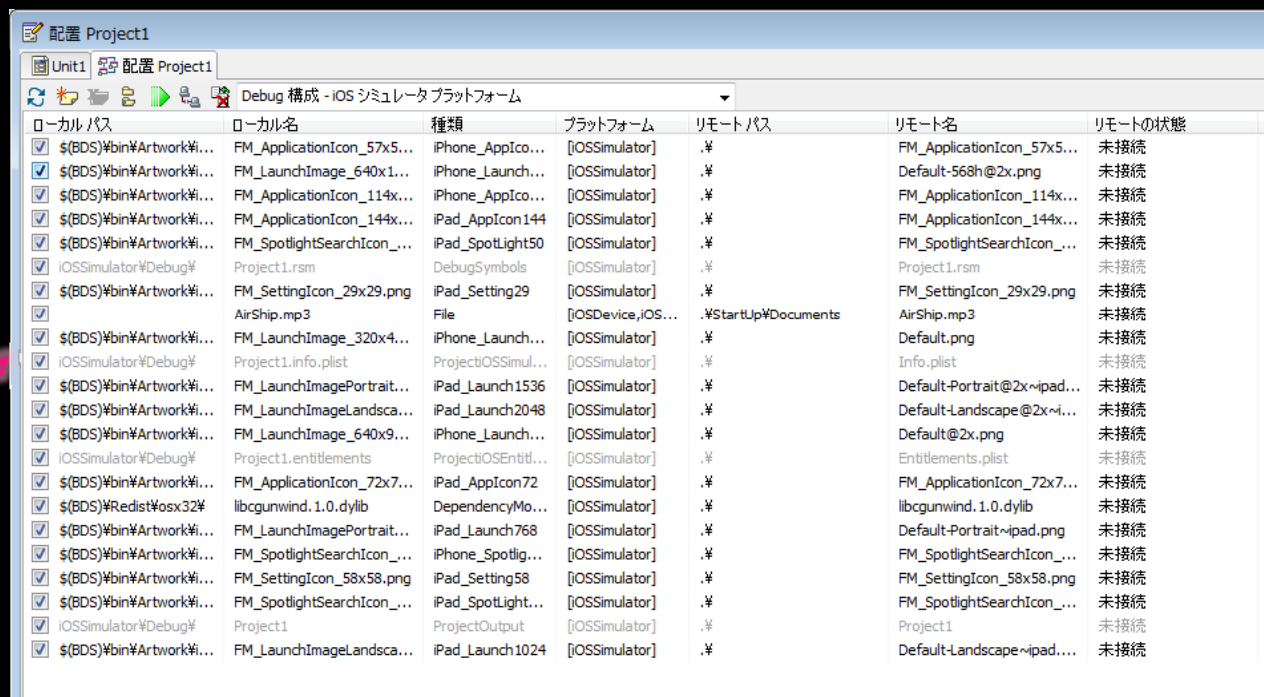
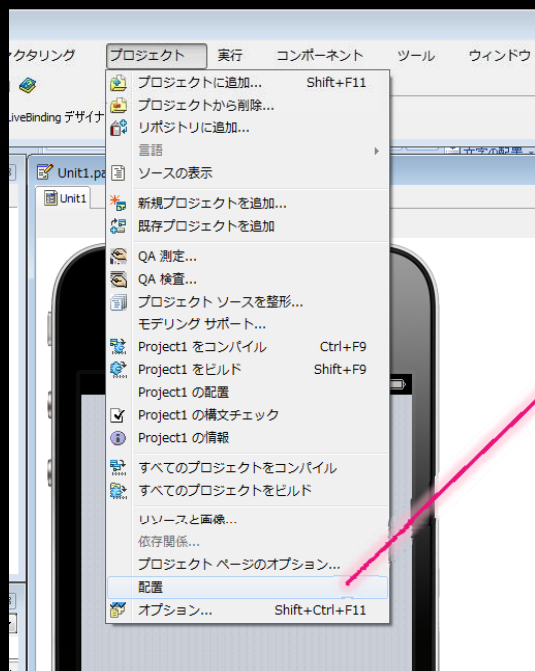
Delphi iOSアプリ開発講座

配置マネージャ - MP3 の再生

配置マネージャ

◎ 配置マネージャ

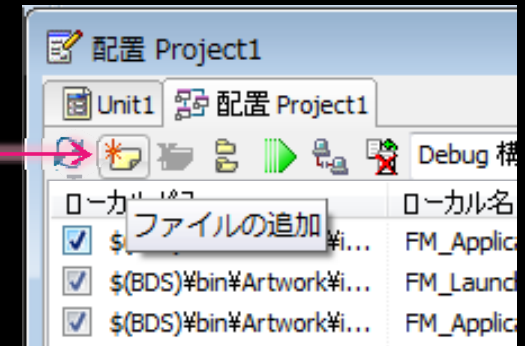
- IPA にパッケージングするファイルを管理している
- 配置マネージャにファイルを追加すると一緒にパックしてくれる



配置マネージャ

◎ 実際に追加してみる

- 左上のファイルの追加ボタンを押す
- ファイルを追加する
 - ここでは AirShip.mp3 を追加する
- リモートパスを ".¥Startup¥Documents" にする
 - 先頭の "." ピリオドを忘れずに！



<input checked="" type="checkbox"/>	\$(BDS)¥bin¥Artwork¥...	FM_LaunchImageLandsca...	iPad_Launch2048	[iOSSimulator]	.¥	Default-Landscape@2x¥i...	未接続
<input checked="" type="checkbox"/>		AirShip.mp3	File	[iOSDevice,iOS...]	.¥	AirShip.mp3	未接続
<input checked="" type="checkbox"/>	\$(BDS)¥bin¥Artwork¥...	FM_LaunchImage_640x9...	iPhone_Launch...	[iOSSimulator]	.¥	Default@2x.png	未接続

<input checked="" type="checkbox"/>	\$(BDS)¥bin¥Artwork¥...	FM_LaunchImageLandsca...	iPad_Launch2048	[iOSSimulator]	.¥	Default-Landscape@2x¥i...	未接続
<input checked="" type="checkbox"/>		AirShip.mp3	File	[iOSDevice,iOS...]	.¥Startup¥Documents	AirShip.mp3	未接続
<input checked="" type="checkbox"/>	\$(BDS)¥bin¥Artwork¥...	FM_LaunchImage_640x9...	iPhone_Launch...	[iOSSimulator]	.¥	Default@2x.png	未接続

配置マネージャ

◎ プログラムからユーザーリソースにアクセスする方法

```
procedure TForm1.FormCreate(Sender: TObject);  
begin  
    MediaPlayer1.FileName :=  
        GetHomePath + PathDelim + 'Documents' + PathDelim + 'AirShip.mp3';  
end;
```

GetHomePath は
アプリのワーキングフォルダを指す


PathDelim はパスの区切り文字
iOS は "/"

配置マネージャ - System.StartUpCopy

◎ .¥StartUp¥Documents とは何か？

- StartUp

- System.StartUpCopy が起動時にコピーするマーク



```
program Project1;

uses
  System.StartUpCopy,
  FMX.Forms,
  Unit1 in 'Unit1.pas' {Form1};

{$R *.res}

begin
  Application.Initialize;
  Application.CreateForm(TForm1, Form1);
  Application.Run;
end.
```

- Documents

- コピー先
- アプリが自由に使えるフォルダ

TMediaPlayer

- ◎ TMediaPlayer を使うと簡単に音を鳴らせる

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  if (MediaPlayer1.State = TMediaState.Playing) then begin
    MediaPlayer1.Stop;
    Button1.StyleLookup := 'arrowrighttoolbuttonbordered';
  end
  else begin
    MediaPlayer1.Play;
    Button1.StyleLookup := 'pausetoolbuttonbordered';
  end;
end;
```



MediaPlayer.Play / Stop で再生・停止ができる
簡単！

➡ 実際に鳴らしてみます。

TMediaPlayer

◎ 注意点

- OnNotify が廃止されている
 - そのため、Repeat 処理が非常に難しい
 - Timer などでは MediaPlayer1.State が Playing かどうかを見て、Playing じゃなくなったときに、再度 Play を呼ぶなどで代替できるが、どうしても一瞬の空隙がうまれる

Thank you!