

# **embarcadero**<sup>®</sup> Developer Camp

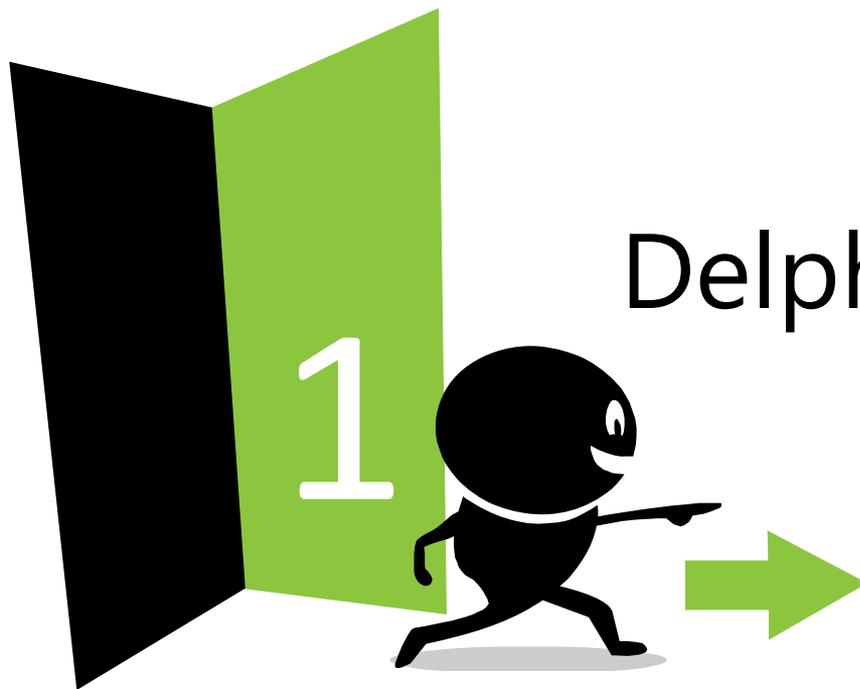
【A1】 Delphiチュートリアルセッション

## 「Windows開発者のための FireMonkeyモバイル開発入門」

エンバカデロ・テクノロジーズ  
エヴァンジェリスト 高橋智宏

# アジェンダ

- Delphi + FireMonkeyの仕組み
  - 各種コンポーネント, スタイル, プロジェクトマネージャ
- iOSアプリの開発
  - 準備するもの
  - 実機に転送
- Androidアプリの開発
  - 準備するもの
  - 実機に転送
- iOS/Android共通の話題
  - iOS or Android ?
  - 画面サイズと画面密度
  - タブレット or スマホ ?
  - 画面の回転(Portrait or Landscape ?)
  - ログイン
  - OnPaintとTCanvas



# Delphi + FireMonkey の仕組み

# 従来のVCLの場合

Delphi 統合開発環境(IDE)  
(エディタ、デバッガ、フォームデザイナー etc.)

Intel/AMD x86用  
Delphi コンパイラ

Intel/AMD x64用  
Delphi コンパイラ

Delphi RTL

VCL

各種コンポーネント

Win32/Win64 API

XP, Vista, 7, 8, 2008, 2012 etc

# iOSの場合

Delphi 統合開発環境(IDE)  
(エディタ、デバッガ、フォームデザイナー etc.)

iOSシミュレータ(Intel x86)用  
Delphi コンパイラ

iOSデバイス(ARMv7)用  
Delphi コンパイラ

Delphi RTL

FireMonkey

各種コンポーネント

iOS API / POSIX

OpenGL ES

iOSシミュレータ

iOSデバイス

# Androidの場合

Delphi 統合開発環境(IDE)  
(エディタ、デバッガ、フォームデザイナー etc.)

Android(ARMv7 + NEON)用 Delphi コンパイラ

Delphi RTL

FireMonkey

各種コンポーネント

Android SDK

Android NDK

OpenGL ES

Androidエミュレータ

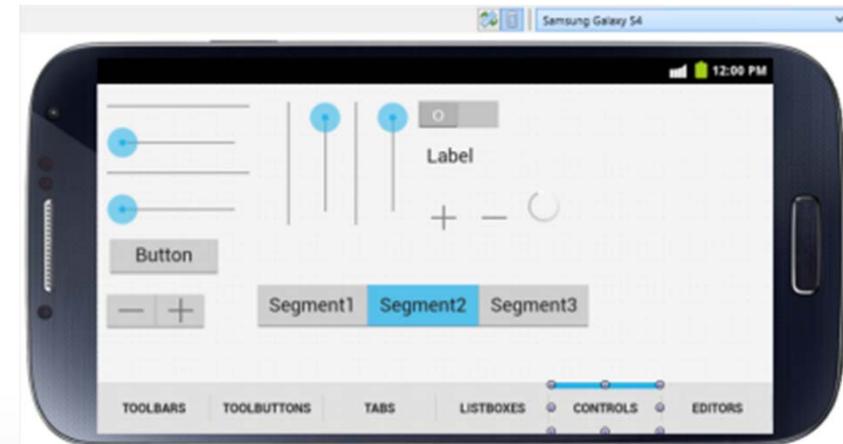
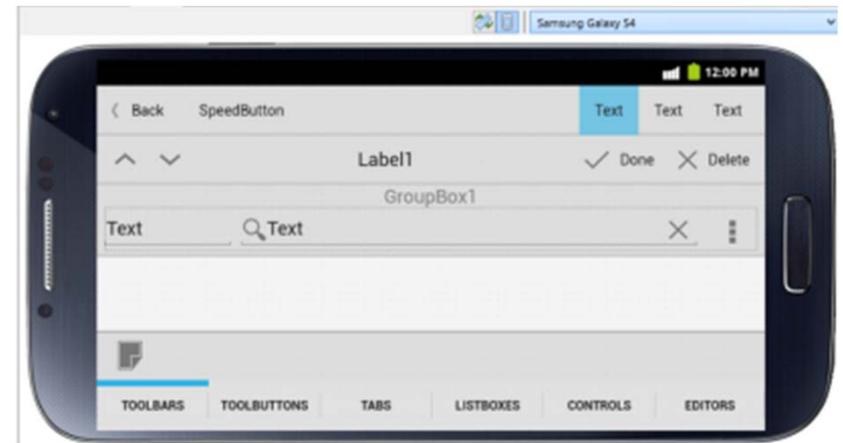
Androidデバイス

# すぐに試せるデモプロジェクト

iPhone(iPod touch)向け



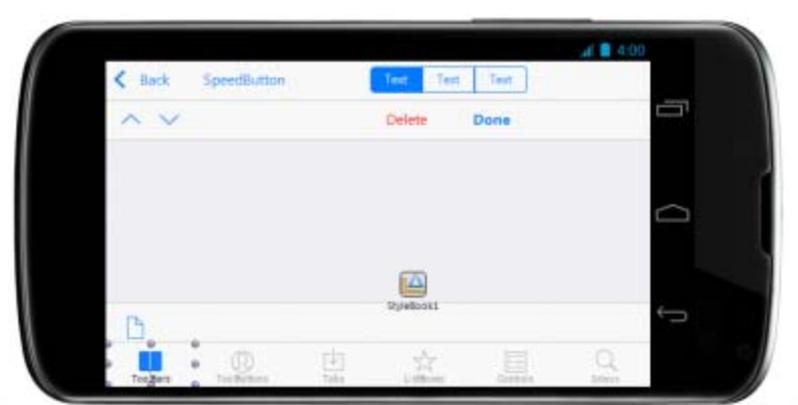
Android



本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

# iOS 7用スタイルも準備済み

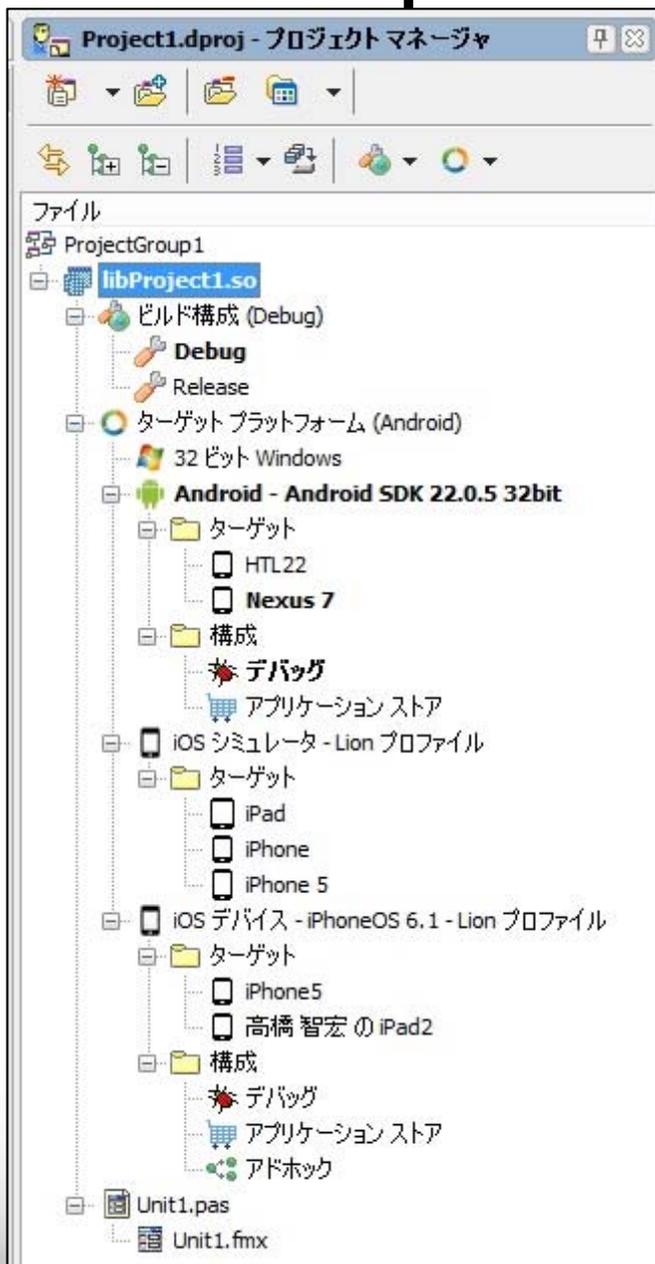
iOS 7の正式リリース後に提供します



本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

# Delphi XE5

～ 新しいプロジェクトマネージャ ～

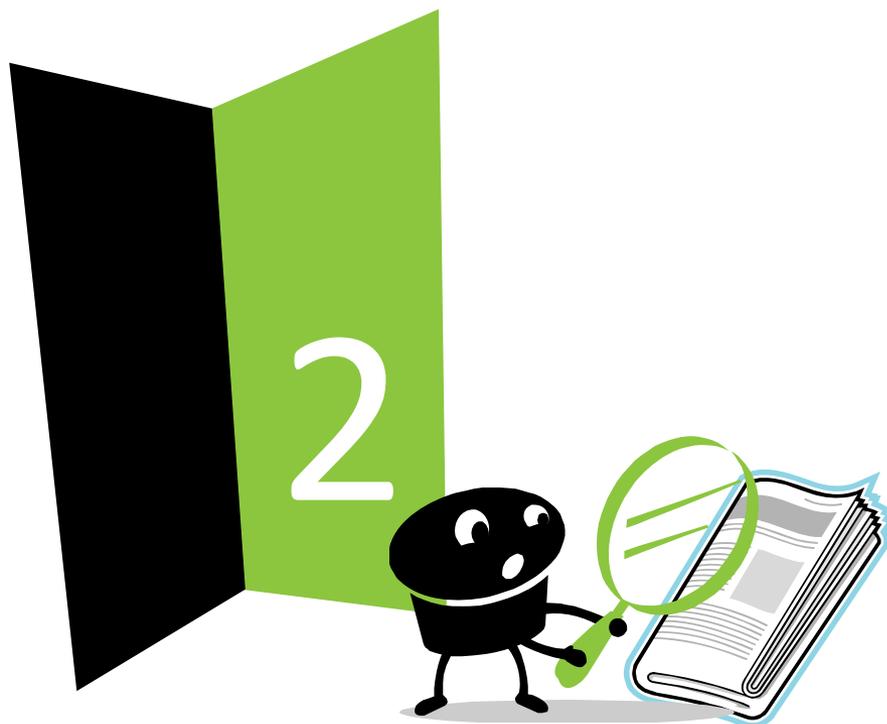


アプリの転送先を選択可能。デバイスを自動認識

- Android
  - ターゲット: 全デバイスおよび全エミュレータ
- iOSシミュレータ
  - ターゲット: iPad, iPhone, iPhone5
- iOSデバイス
  - ターゲット: 全デバイス



の転載を禁止します。本文書の著作権は、著作者に帰属します。



# iOSアプリの開発

# 準備するもの

- Mac OS X Lion(10.7)以降のMacマシン
- Xcode 4.6.3 または Xcode 5(Beta)
- iOS 6.0以降が動作している iPad または iPhone
  - iOSシミュレータ または iOS実機
- Apple iOS Developer Programへの加入
  - 証明書作成, iOS実機の登録, プロファイルのインストール etc...
- XE5のRAD PAServerをMacマシンにインストール&起動

それと...

- Windows 7(32bit/64bit)などのPCまたは仮想マシン
- Delphi XE5(モバイル開発機能)

# Macマシンや実機がなくても

- MacinCloud.comが使えます
  - 無料トライアルまたは有料のMac OS X環境の時間貸し
    - Remote Desktop接続なのでレスポンスは良くない
    - さすがに、iOSシミュレータのみ

<http://www.macincloud.com/>

Remote Desktop Client



です。

# Androidアプリの開発

3



# 準備するもの

- Windows 7(32bit/64bit)などのPCまたは仮想マシン
    - Mac OS Xは非サポート
  - Delphi XE5(モバイル開発機能)
  - Android 2.3.3以降が動作しているAndroid実機
    - ARMv7互換CPU
      - Intel x86およびMIPSは除く
    - NEON命令をサポートしないNVIDIA Tegra2は除く
      - なので、Android 3.x(Honeycomb)は除く
  - USBケーブルとWindows用USBドライバ
    - [開発者向けオプション]で[USBデバッグ]を[ON]にしておく
- あと...
- 自前でWindowsにインストールしてもOKのもの
    - JDK7(32bit/64bit)
    - Android SDK(ADT Bundle, 32bit/64bit)
    - Android NDK(32bit/64bit)

# Android実機がなくても

- エミュレータは遅すぎるので実用的ではありません!!
  - Intel x86エミュレータも使えません...
- SciroccoCloud(シロッコ・クラウド)等のサービスが使えます
  - 無料トライアルまたは有料のAndroid端末時間貸し
    - Webブラウザ経由で日本製各種Android実機を操作
    - エンバカデロでも製品テストに使ってます!!

<http://www.scirocco-cloud.com/ja/>





# iOS/Android 共通の話題

# iOS or Android ?

- コンパイル時に決める

```
{$IFDEF ANDROID}  
  ...  
{$ENDIF}  
{$IFDEF IOS}  
  ...  
{$ENDIF}
```

- 実行時に判定する

```
var  
  os: TOSVersion;  
begin  
  if os.Platform = pfIOS then  
    Button1.Text := 'iOS'  
  else if os.Platform = pfAndroid then  
    Button1.Text := 'Android'  
  else  
    Button1.Text := 'Windows 8 Tablet?';  
end;
```

# 画面サイズと画面密度

- 論理的(デバイス非依存)なサイズ(dp)なのでデザインが楽
  - ただし、画素の細かさには注意が必要(例: UIImageのMultiResBitmap)

```
var
  ws: IFMXWindowService;
begin
  Label1.Text := '幅:' + IntToStr(Screen.Size.Width) + ', 高:' + IntToStr(Screen.Size.Height);
  ws := TPlatformServices.Current.GetPlatformService(IFMXWindowService) as IFMXWindowService;
  Button1.Text := 'スケール:' + FloatToStr(ws.GetWindowScale(Self));
end;
```

Portrait  
の状態



機種	幅(dp)	高さ(dp)	スケール	規格
iPhone 4, iPhone 4S	320	480	2.0	Retina
iPhone 5	320	568	2.0	Retina
iPad 2, iPad mini	768	1024	1.0	Non-retina
iPad 3, iPad 4	768	1024	2.0	Retina
au HTC EVO WiMAX	320	533	1.5	hdpi
au HTC EVO 3D	360	640	1.5	hdpi
LG Nexus 4	384	592	2.0	xhdpi
au HTC J One	360	592	3.0	xxhdpi
Dropad A8	480	800	1.0	mdpi
ASUS Nexus 7 (2012)	600	905	1.331	tvdpi
Kindle Fire HD	533	853	1.5	hdpi
SAMSUNG Nexus 10	752	1280	2.0	xhdpi

iOS

Android

# タブレット or スマホ？

- iOSは判別が簡単!!

```
var
  device: IFMXDeviceService;
begin
  device := TPlatformServices.Current.GetPlatformService(IFMXDeviceService) as IFMXDeviceService;
  if Pos('iPad', device.GetModel) > 0 then
    Application.CreateForm(TFormForiPad, Form2) // iPad
  else
    Application.CreateForm(TFormForiPhone, Form1); // iPhone, iPod touch
```

- Androidには明確な基準が無い??
  - Tablet > Phablet > Phone
  - 幅または高さが480dp以上ならタブレットだと想定する



# 画面の回転(Portrait or Landscape ?)

- フォームのOnResizeイベントをフックして回転を検知

```
uses FMX.Platform;  
  
procedure TForm1.FormResize(Sender: TObject);  
var  
    ss: IFMXScreenService;  
    ori: TScreenOrientation;  
begin  
    ss := TPlatformServices.Current.GetPlatformService(IFMXScreenService) as IFMXScreenService;  
    ori := ss.GetScreenOrientation;  
    if (ori = TScreenOrientation.soPortrait) OR (ori = TScreenOrientation.soInvertedPortrait) then  
        Button1.Text := 'Portrait'  
    else  
        Button1.Text := 'Landscape';  
end;
```

- フォームのレイアウトは?
  - 同一フォーム内で各GUI部品のレイアウトを調整  
または
  - 縦横別々のフォームを Form.Show; して、丸ごと切り替え

# ロギング

- iOS
  - DelphiのUnicodeStringが最終的にはNSString型として出力
  - iOSシミュレータ: コンソール.appで確認
  - iOSデバイス: Xcode付属のOrganizerで確認
- Android
  - DelphiのUnicodeStringが最終的にはPAnsiCharとして出力
  - Android SDKのmonitorユーティリティで確認

```
var
  i: Integer;
  f: Single;
  e: Extended; // Double
  log: IFMXLoggingService;
begin
  i := Random(100);
  f := 123.456;
  e := Random;
  log := TPlatformServices.Current.GetPlatformService(IFMXLoggingService) as IFMXLoggingService;
  log.Log(' %s:%d:%f:%f', ['これはログメッセージ!', i, f, e]);
  //または FMX.Types.Log.d(' %s:%d:%f:%f', ['これはログメッセージ!', i, f, e]);
end;
```

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  {$IFDEF ANDROID}
    System.DefaultSystemCodePage := 65001; // UTF-8: PAnsiChar (Android)
  {$ENDIF}
end;
```

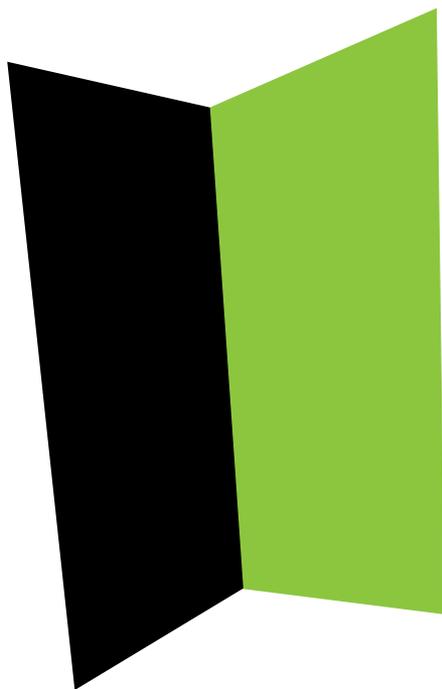
# OnPaintイベントとTCanvas

- 画面への描画はWindowsOS向けVCL(GDI,HDC)とは異なる
  - 必ずOnPaintイベントとそのCanvasパラメータを使用すること!!
  - Repaintメソッドの呼び出しも必要

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  paint_color := TAlphaColorRec.Black;
end;

procedure TForm1.Button1Click(Sender: TObject);
begin
  if paint_color = TAlphaColorRec.Black then
    paint_color := TAlphaColorRec.Red
  else
    paint_color := TAlphaColorRec.Black;
  PaintBox1.Repaint;
end;

procedure TForm1.PaintBox1Paint(Sender: TObject; Canvas: TCanvas);
begin
  Canvas.BeginScene();
  Canvas.Fill.Color := paint_color;
  Canvas.FillRect(RectF(0, 0, PaintBox1.Width, PaintBox1.Height), 0, 0, [], 1);
  Canvas.EndScene;
end;
```



# Q&A