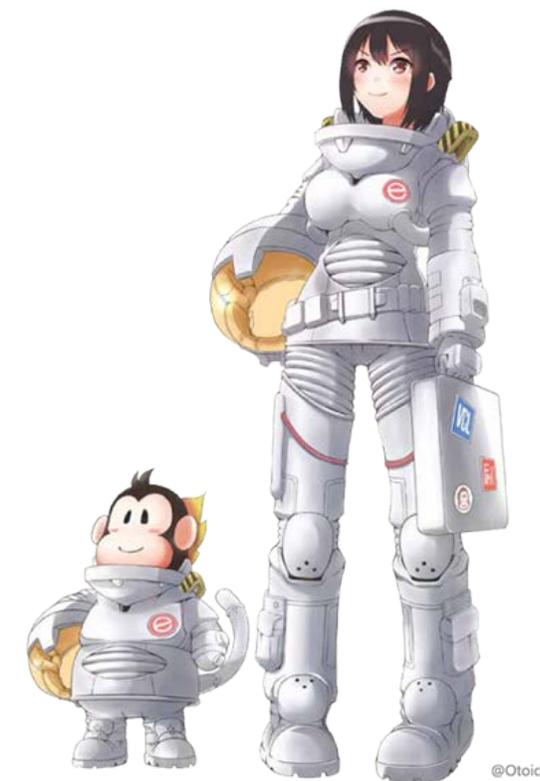


【A5】 Delphiテクニカルセッション
クロスプラットフォーム開発で役立つ
Delphi 新機能活用ポイント

第34回 エンバカデロ・デベロッパーキャンプ

株式会社シリアルゲームズ
App Div 3 マネージャー
取締役 細川 淳



embarcadero®
DEVELOPER CAMP

■はじめに



embarcadero®
DEVELOPER CAMP

アジェンダ

- IDE の新機能
- コマンドラインツール
- Delphi / Object Pascal の新機能

**※機能が多すぎたためすべては紹介できません！
特に RTL / VCL / FMX については全てを省略しました。**

■ IDE の新機能



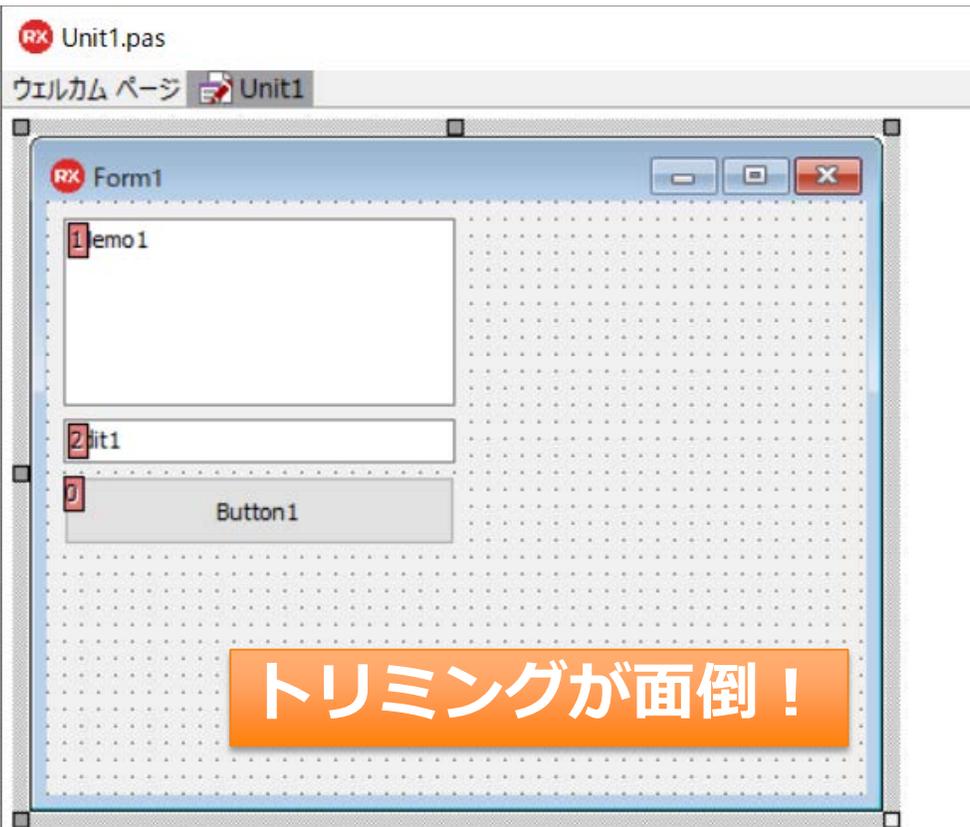
embarcadero®
DEVELOPER CAMP

フォームのスナップショット: XE~ / VCL + FMX△

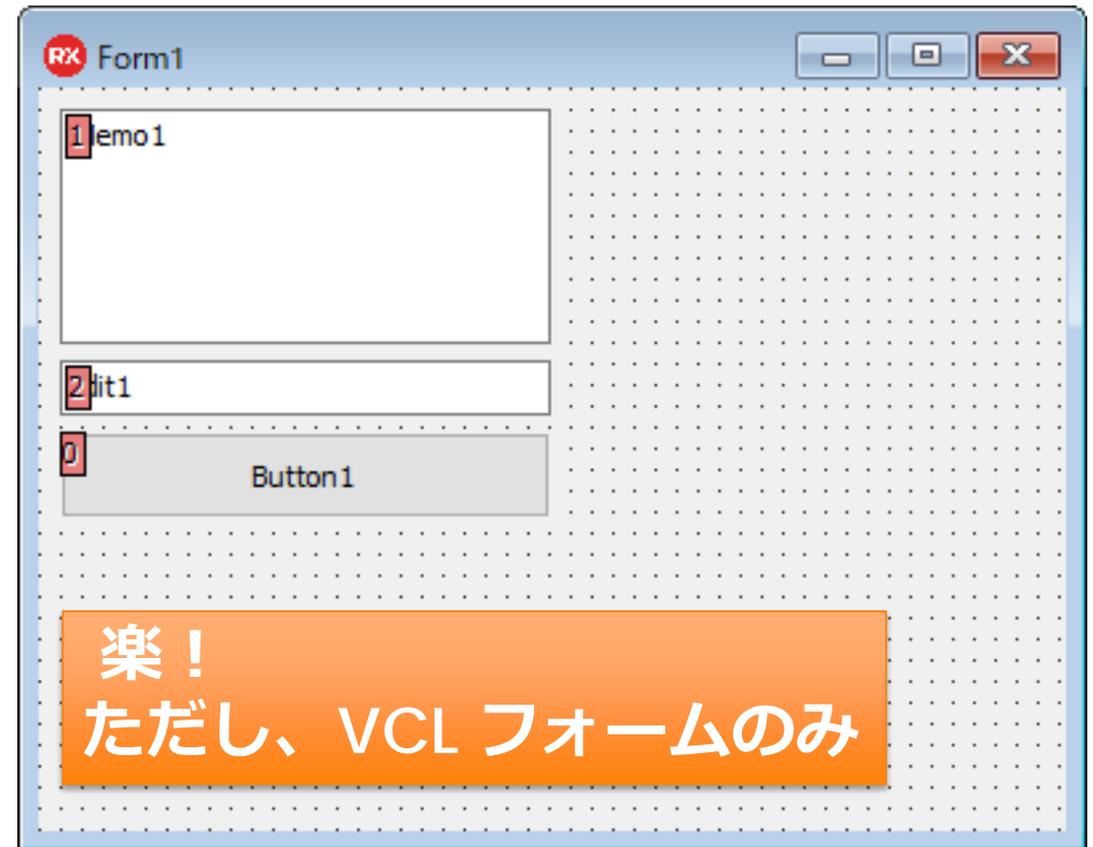
- IDE のフォームデザイナーにフォーカスがあるときに **Ctrl + C**
- VCL
 - スナップショット画像
 - コンポーネントストリーム形式のテキスト
- FMX
 - コンポーネントストリーム形式のテキスト

フォームのスナップショット: XE~ / VCL + FMX△

普通に Alt + PrintScreen して
Photoshop でトリミングしたもの



Ctrl + C でスナップショットを取ったもの



※赤い四角の囲み文字は IDE の機能拡張 (CnPack の機能)

フォームのスナップショット: XE~ / VCL + FMX△

- テキスト形式
 - 選択されているコンポーネントのコンポーネントストリーム形式のテキストを取得できる
 - Form が選択されていればフォーム全体の、コントロールが選択されていればコントロールだけのコンポーネントストリーム形式テキストを取得

フォームのスナップショット: XE~ / VCL + FMX△

VCL フォーム

```
object Form1: TForm1
  Left = 0
  Top = 0
  Caption = 'Form1'
  ClientHeight = 281
  ClientWidth = 387
  Color = clBtnFace
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -11
  Font.Name = 'Tahoma'
  Font.Style = []
  OldCreateOrder = False
  PixelsPerInch = 96
  TextHeight = 13
  object Button1: TButton
    Left = 8
    Top = 130
    Width = 185
    Height = 33
    Caption = 'Button1'
    TabOrder = 0
  end
  object Memo1: TMemo
    Left = 8
    Top = 8
    Width = 185
    Height = 89
    Lines.Strings = (
      'Memo1')
    TabOrder = 1
  end
  object Edit1: TEdit
    Left = 8
    Top = 103
    Width = 185
    Height = 21
    TabOrder = 2
    Text = 'Edit1'
  end
end
```

VCL コントロールを選択

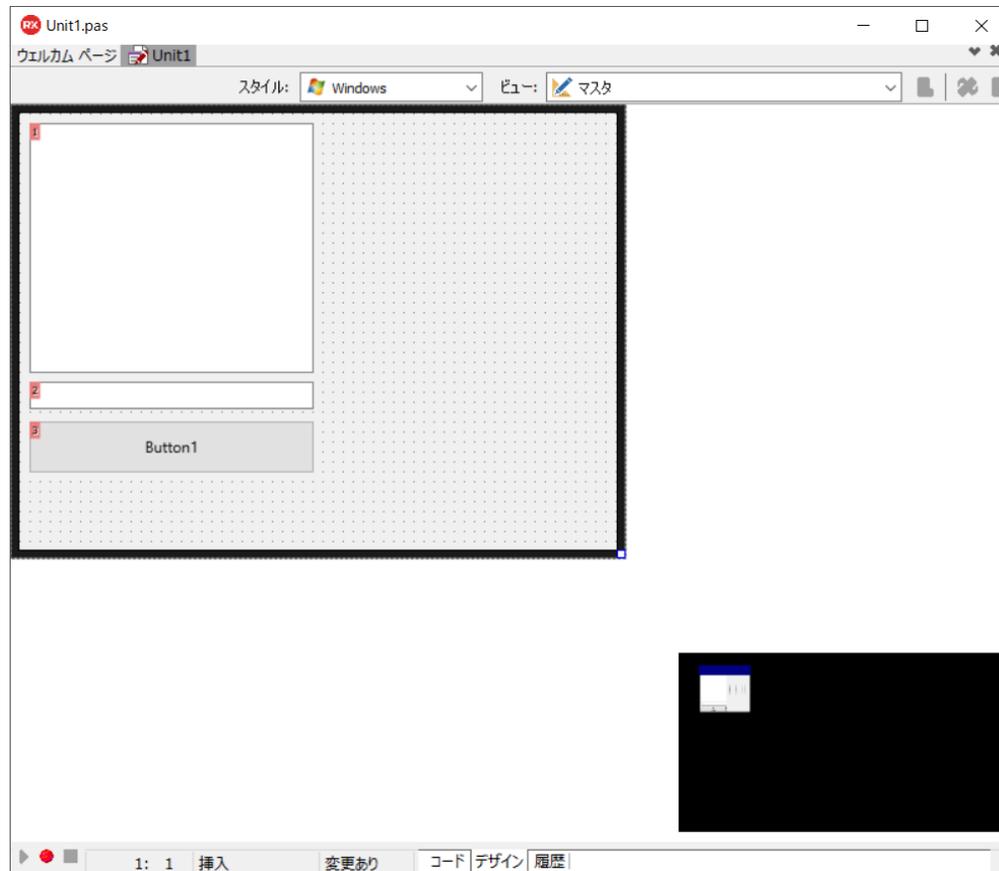
```
object Button1: TButton
  Left = 8
  Top = 130
  Width = 185
  Height = 33
  Caption = 'Button1'
  TabOrder = 0
end
```

FMX フォーム

```
object Form1: TForm1
  Left = 0
  Top = 0
  Caption = 'Form1'
  ClientHeight = 351
  ClientWidth = 473
  FormFactor.Width = 320
  FormFactor.Height = 480
  FormFactor.Devices = [Desktop]
  DesignerMasterStyle = 0
  object Memo1: TMemo
    Touch.InteractiveGestures = [Pan, LongTap,
      DoubleTap]
    DataDetectorTypes = []
    Position.X = 8.000000000000000000
    Position.Y = 8.000000000000000000
    Size.Width = 225.000000000000000000
    Size.Height = 201.000000000000000000
    Size.PlatformDefault = False
    TabOrder = 1
    Viewport.Width = 221.000000000000000000
    Viewport.Height = 197.000000000000000000
  end
  object Edit1: TEdit
    Touch.InteractiveGestures = [LongTap, DoubleTap]
    TabOrder = 2
    Position.X = 8.000000000000000000
    Position.Y = 216.000000000000000000
    Size.Width = 225.000000000000000000
    Size.Height = 22.000000000000000000
    Size.PlatformDefault = False
  end
  object Button1: TButton
    Position.X = 8.000000000000000000
    Position.Y = 248.000000000000000000
    Size.Width = 225.000000000000000000
    Size.Height = 41.000000000000000000
    Size.PlatformDefault = False
    TabOrder = 3
    Text = 'Button1'
  end
end
```

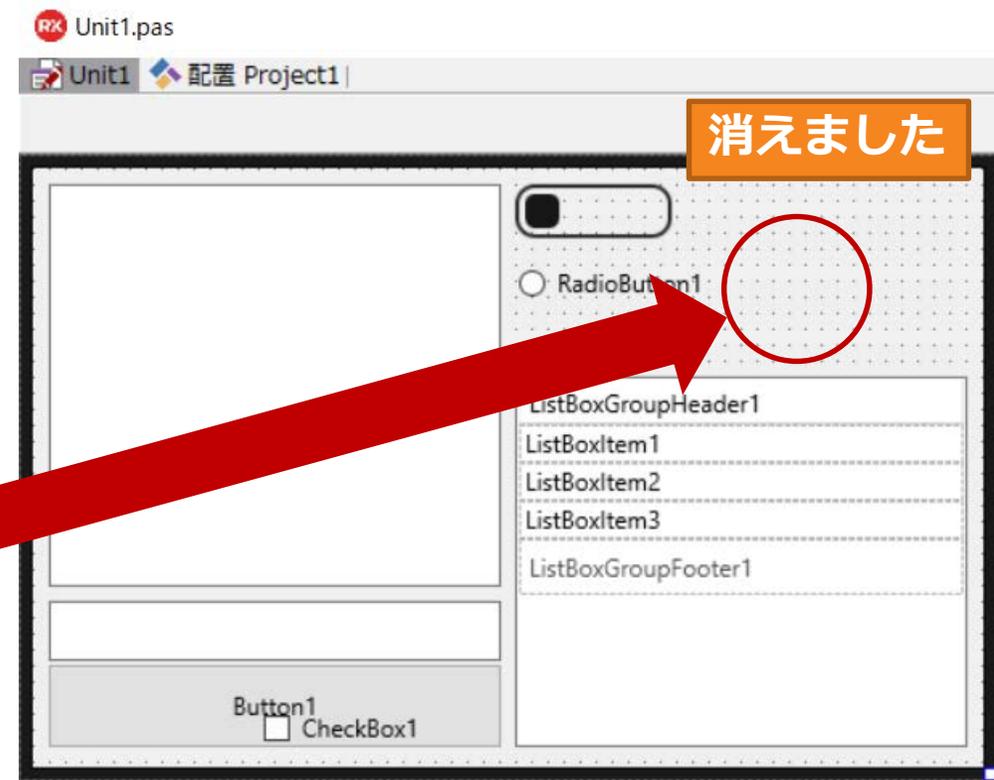
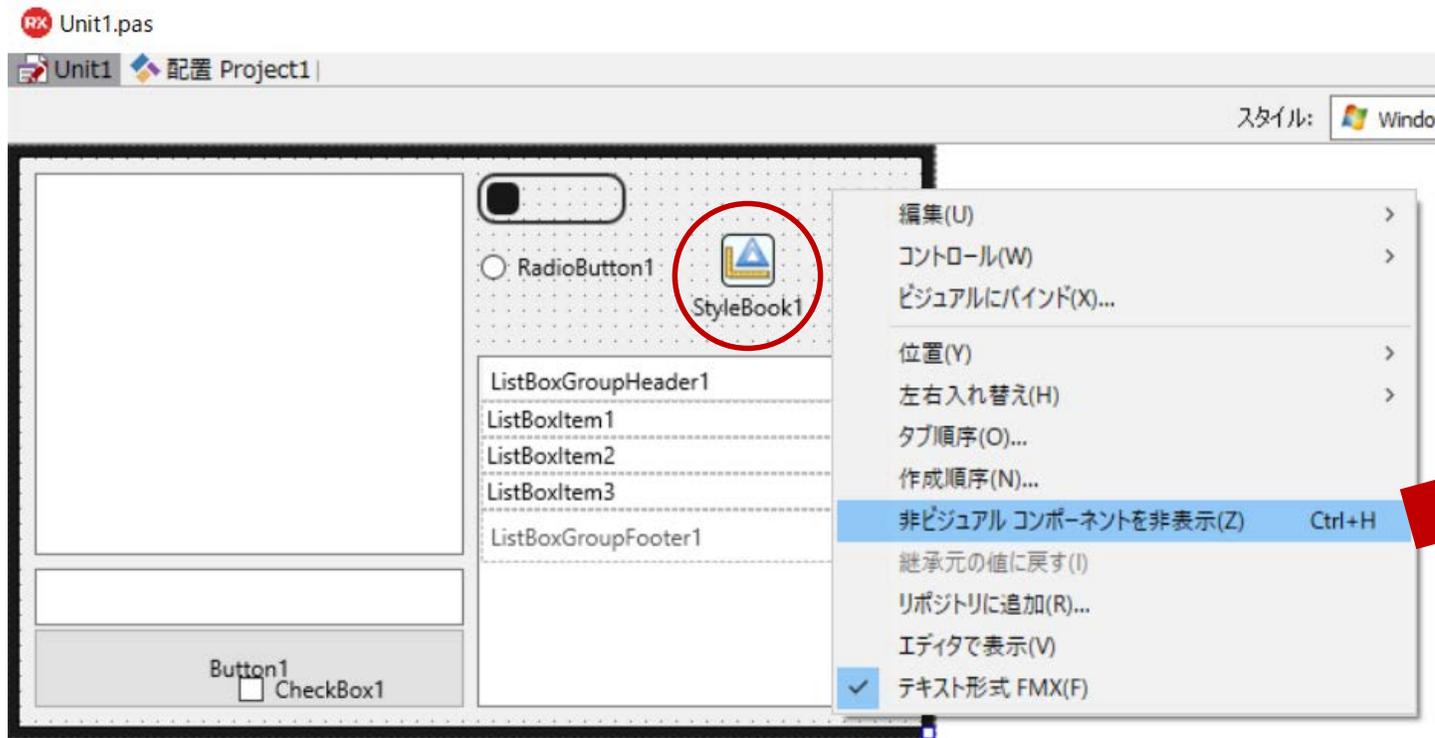
フォームスクリーン位置プレビュー: XE~ / VCL + FMX

- 右下に画面内での位置が表示されます
 - つかんで移動すると Left, Top が変わります



非ビジュアル コンポーネントを非表示にする: 10 Seattle~

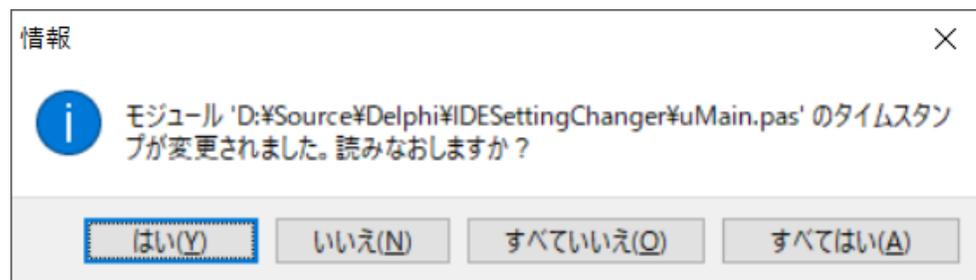
- 非ビジュアルコンポーネントが邪魔になるときは、この機能で非表示にできます。



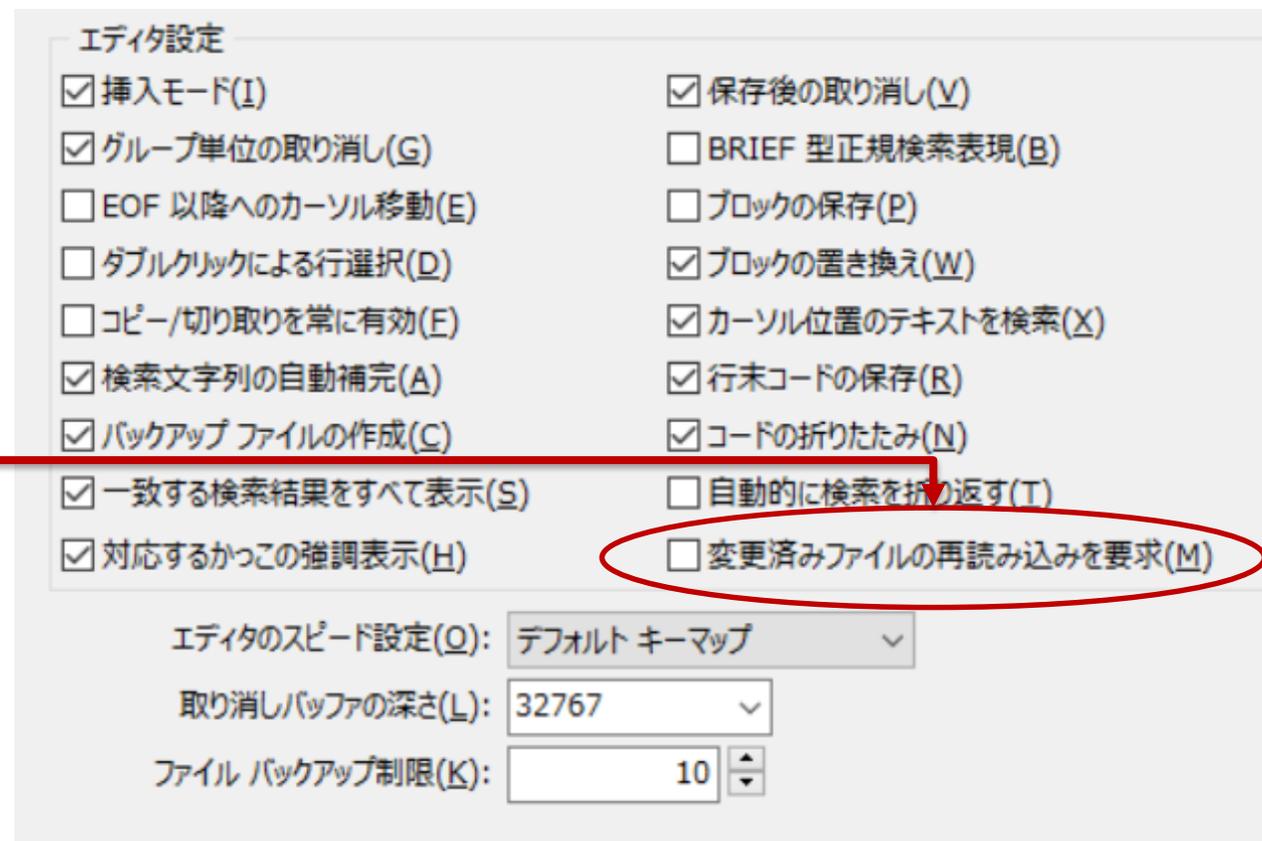
変更済みファイルの再読み込みを要求: XE~

- 外部（Gitなど）で変更されたファイルの再読込を制御

通常は変更されるとダイアログが表示されます



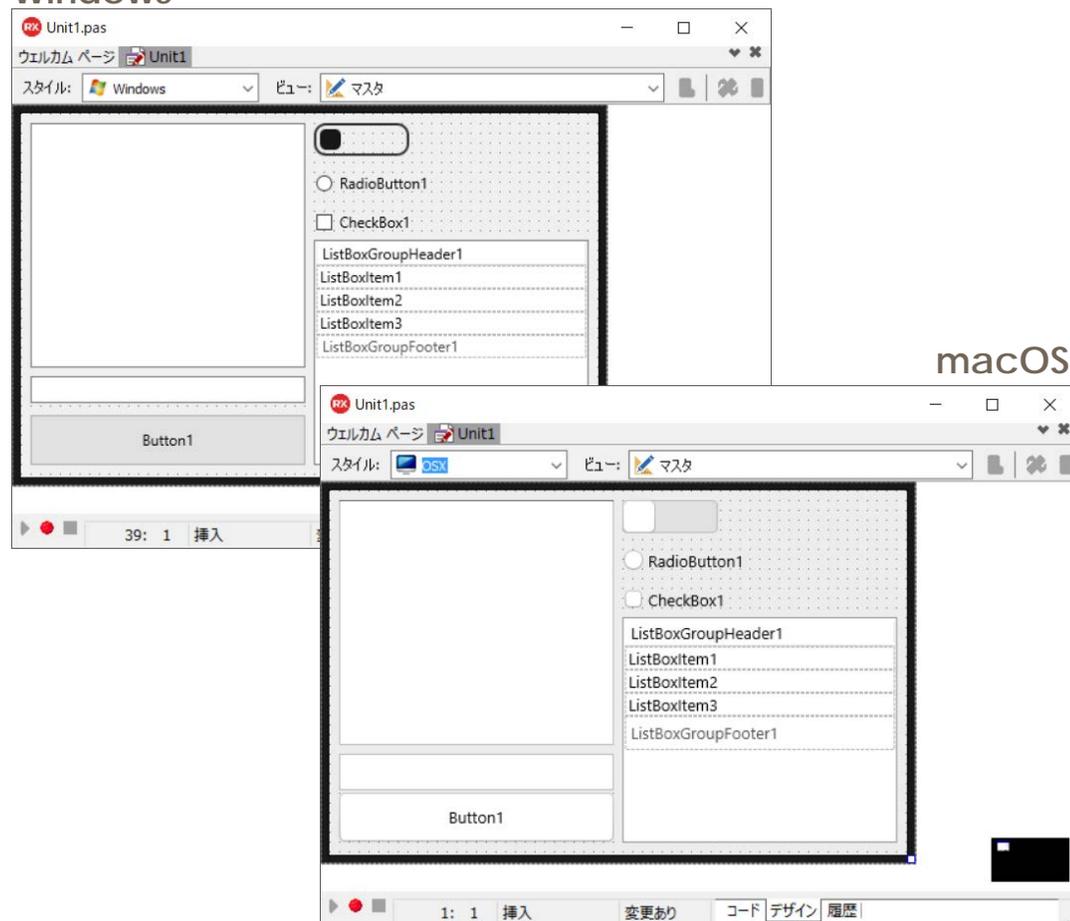
このチェックを外すとダイアログは表示されなくなり、自動的に再読込されます



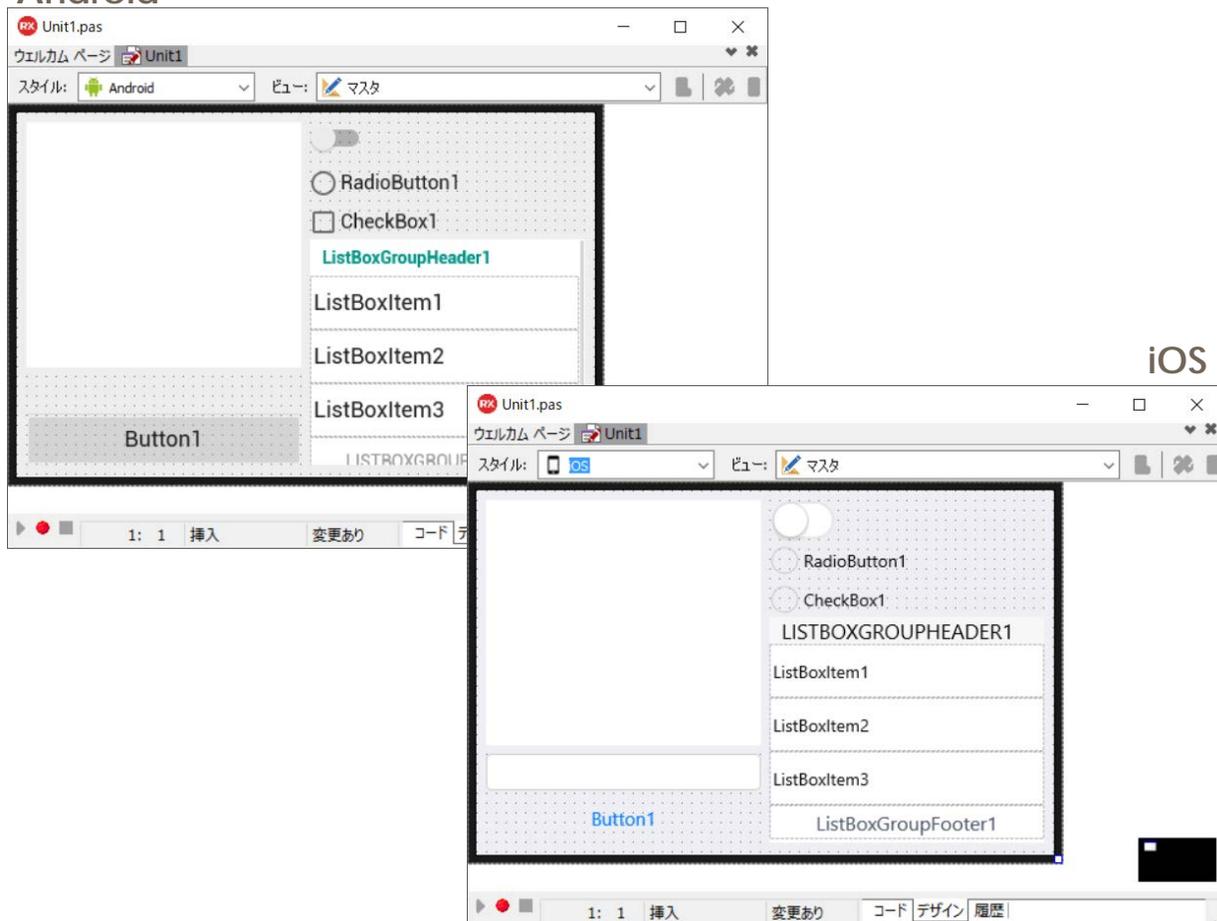
デザインデバイス: XE5~ / FMX

- 設計時に各 OS の見た目でコントロールが表現されます

Windows



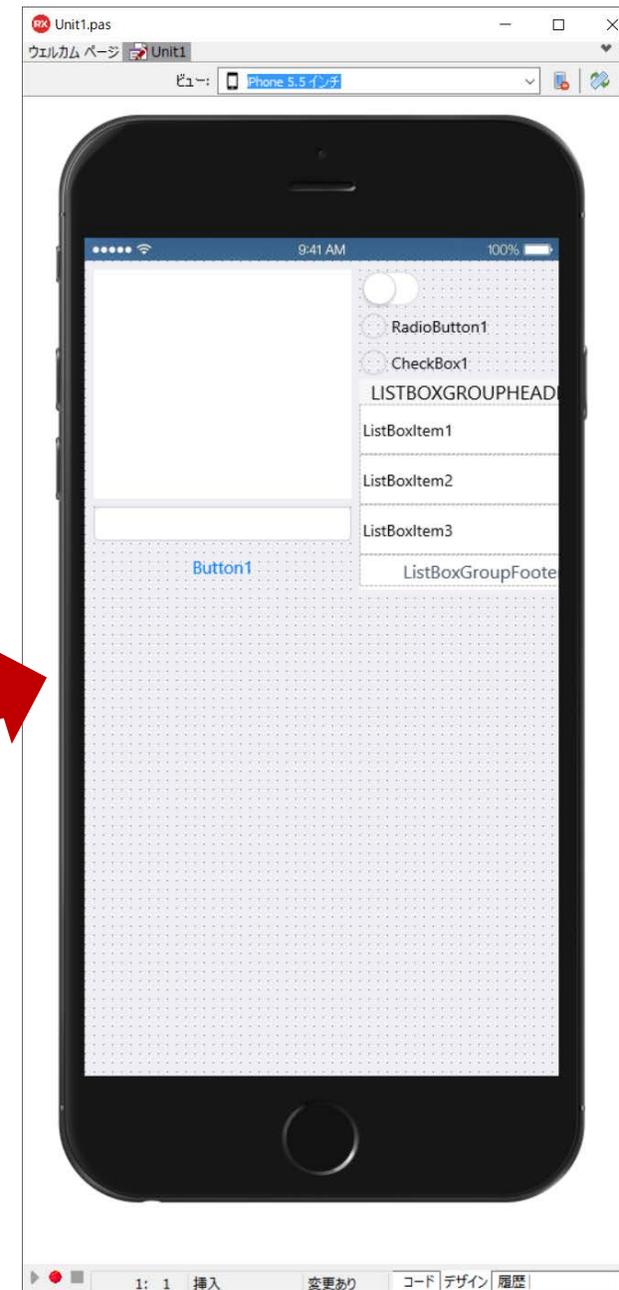
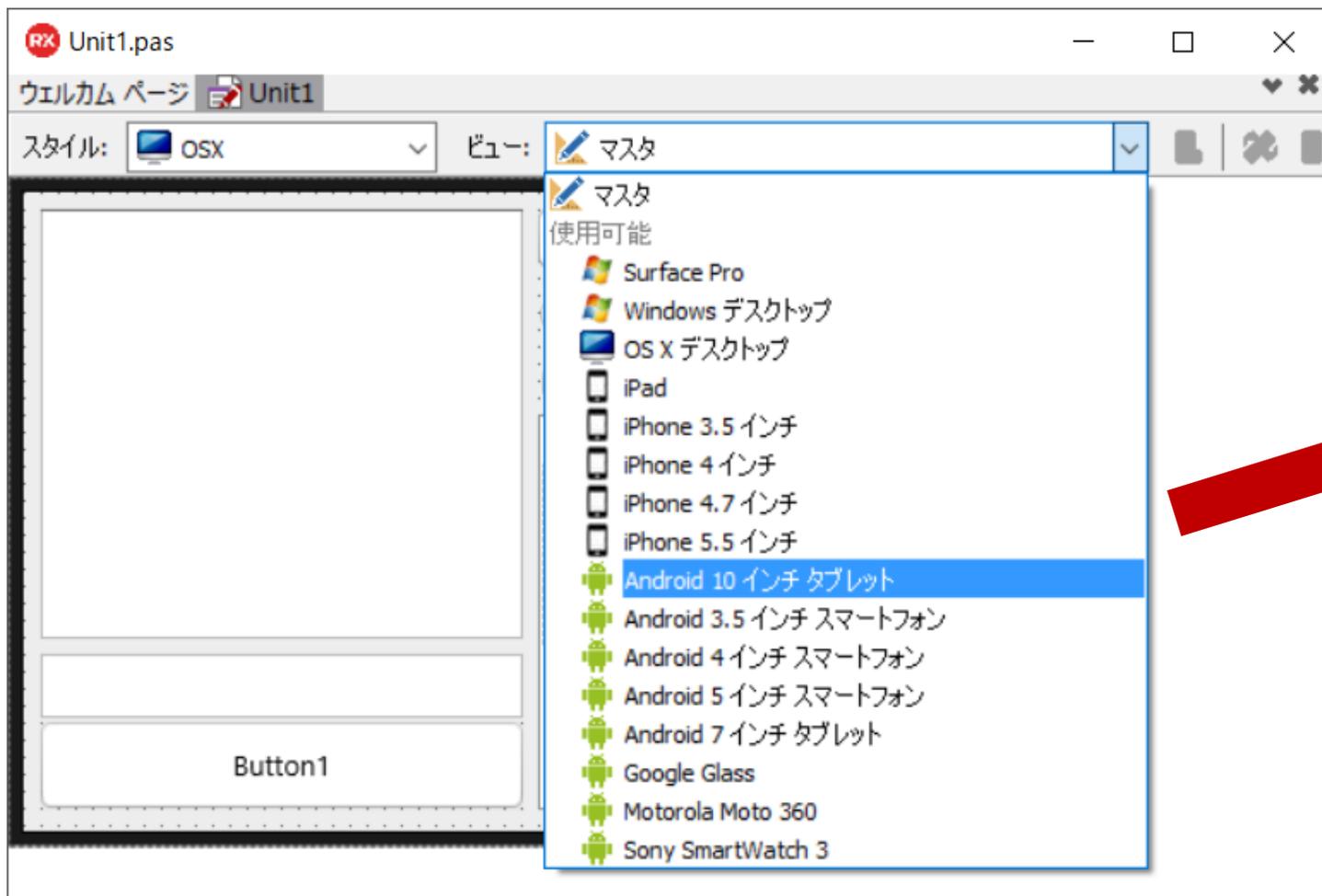
Android



FireUI: XE7～ / FMX

- フォームを継承する仕組み
 - コントロールはすべてマスタに置く必要があります
 - コントロールのプロパティは継承先で自由に変更できます
 - 例えば Visible を False にすればコントロールが表示されなくなります
 - また、Form.StyleBook に別のスタイルを設定すれば、OS や対象機種によって別の見た目を表現できます。

FireUI: XE7~ / FMX



FireUI: XE7~ / FMX

- ソースには \$R コンパイラ指令でリソースを読み込むように記述が追加されます。

```
var
    Form1: TForm1;

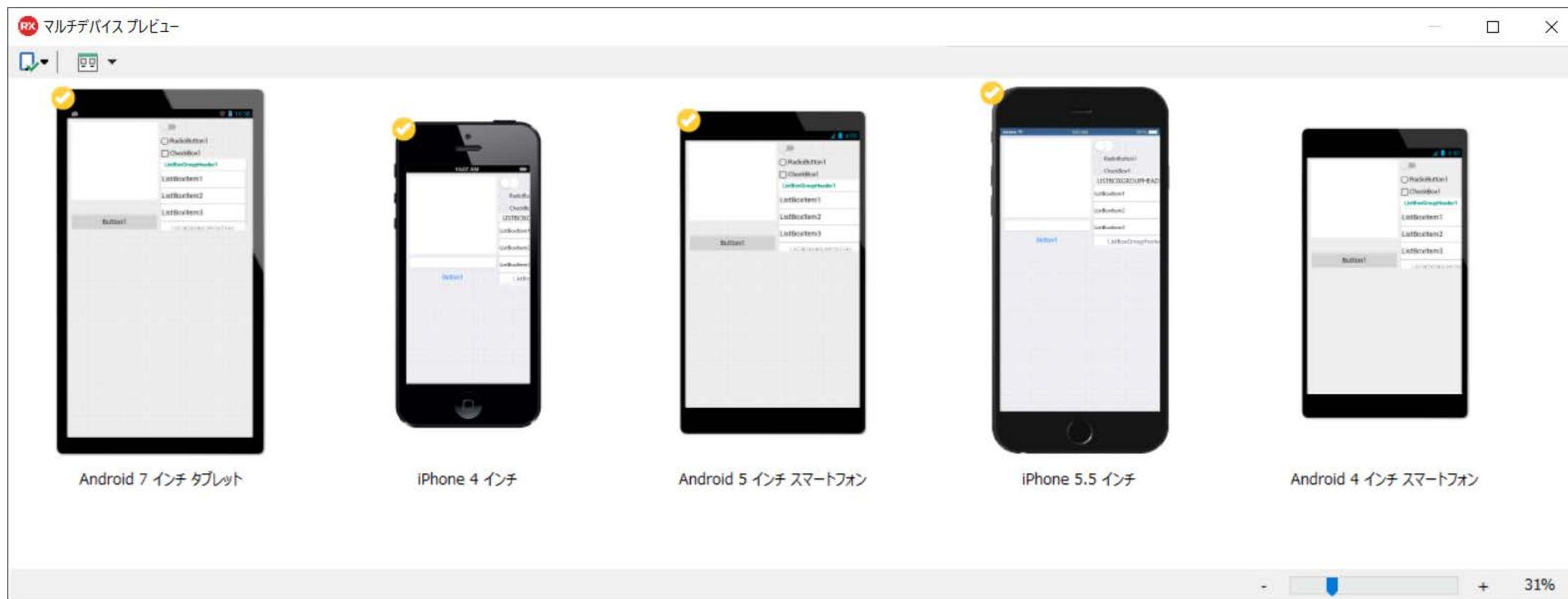
implementation

{$R *.fmx}
{$R *.LgXhdpiTb.fmx ANDROID}
{$R *.iPhone4in.fmx IOS}
{$R *.LgXhdpiPh.fmx ANDROID}
{$R *.iPhone55in.fmx IOS}

end.
```

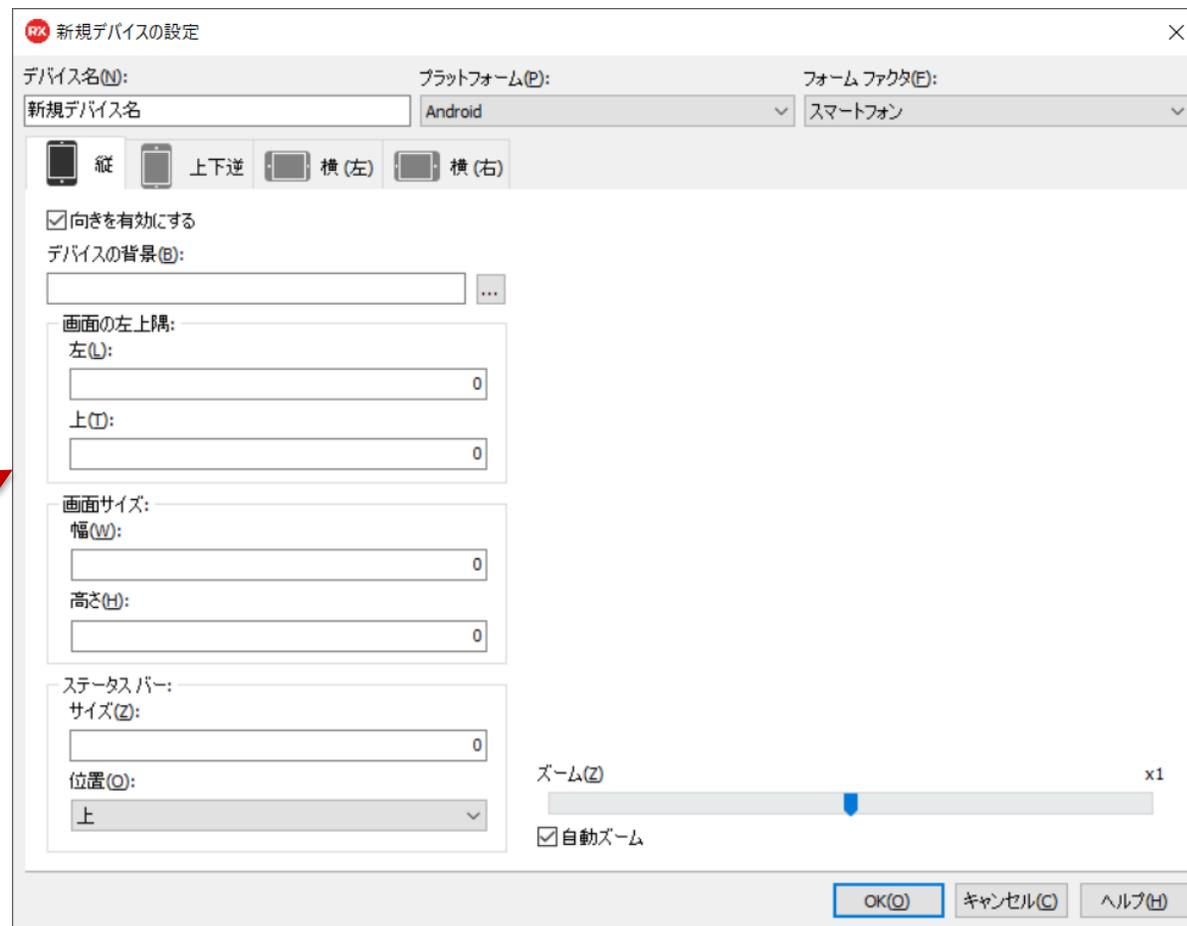
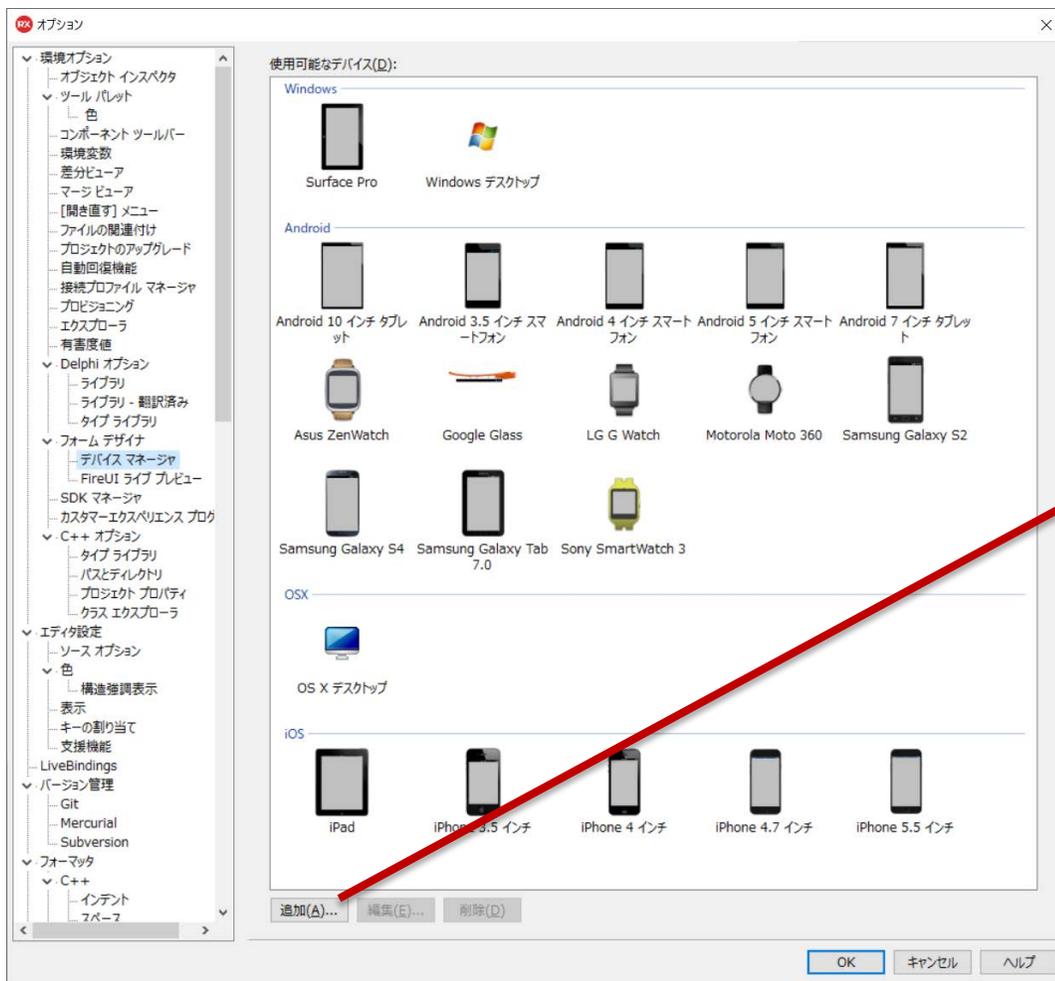
マルチデバイスプレビュー: XE8~ / FMX

- FireUI で追加されたデバイスでの見た目を一気にプレビューできます



デバイスマネージャ: XE8~ / FMX

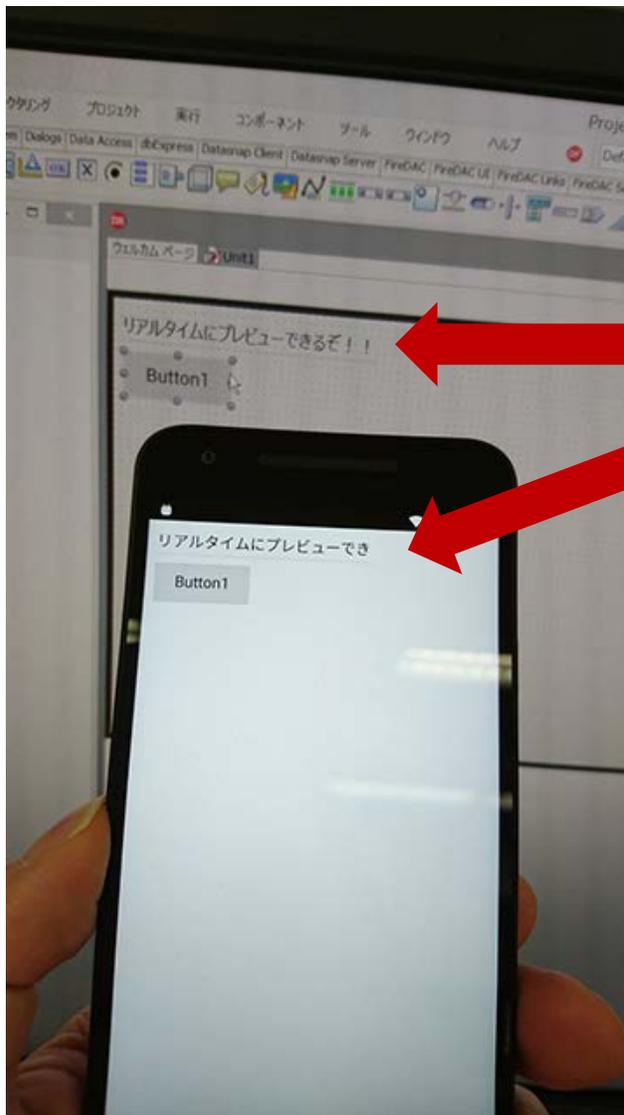
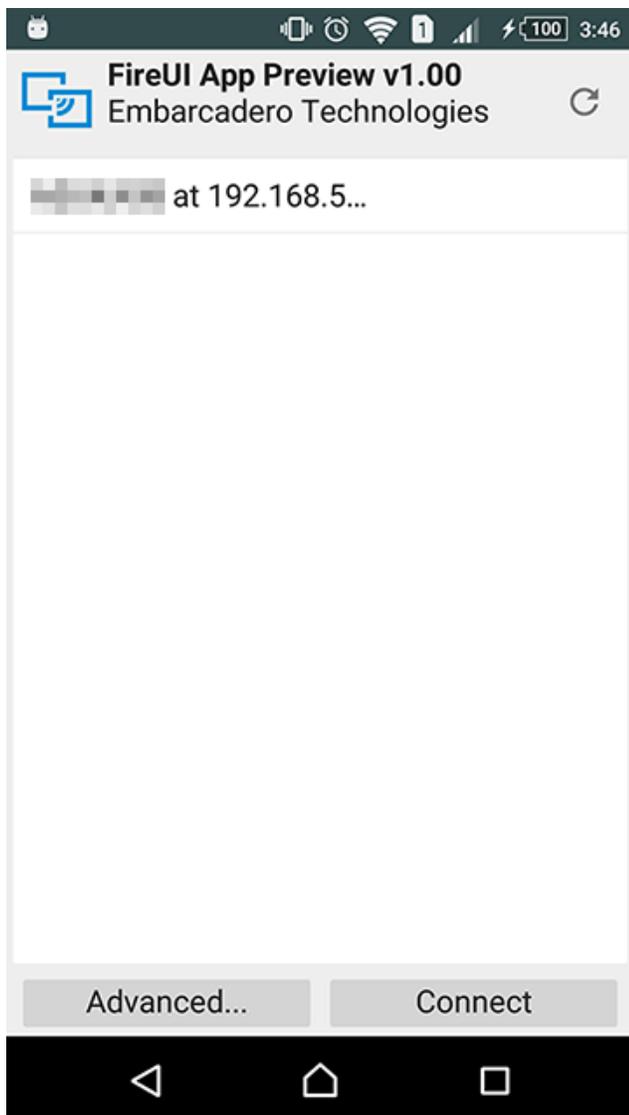
- FireUI に表示するデバイスを追加・編集・削除できる機能です



FireUI Live Preview: 10.1 Berlin～ / FMX

- フォームデザイナーの画面を直接実機に表示する機能です。
 - 実機で見るとフォントの違いからデザイン画面と見え方が異なることがあります、大変便利です。
 - 実際に使用する場合は、AppStore / Play Store で配布されている “FireUI App Preview” アプリのインストールが必要です。
- 拙作 Qiita の記事
[Delphi 10.1 Berlin で実装された FireUI App Preview](http://qiita.com/pik/items/e8f75f5df7238e5c0c73)
<http://qiita.com/pik/items/e8f75f5df7238e5c0c73>

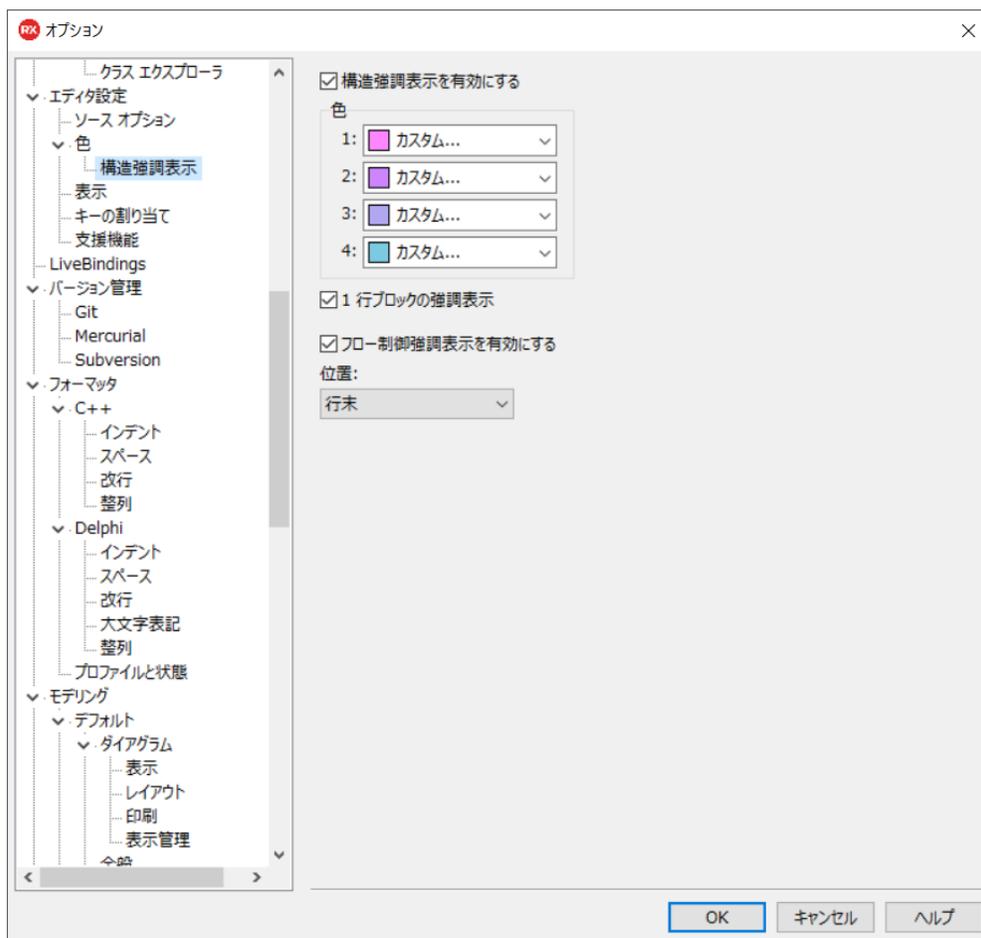
FireUI Live Preview: 10.1 Berlin~ / FMX



フォントの違いで
フォームデザイナーと表示が
異なっています。

構造強調表示(Castalia): XE8~

- コードエディタで構造を表示してくれる機能です



```
240 // 変更内容読み込み
    with TStringList.Create do
    try
        LoadFromFile(Target);
        Contents := Text;
    finally
        DisposeOf;
    end;

250 // テンポラリ削除
    try
        TFile.Delete(Target);
    except on E: Exception do
        begin
            ShowMsg(E.ToString);
            Exit;
        end;
    end;

260 if (Res.IsEmpty) then
    begin
        ShowMsg(ERR_MSG_EXECUTE);
        Exit;
    end;

264 // リプレース
    for i := 0 to Editor.EditViewCount - 1 do
    begin
        EditView := Editor.EditViews[i];
        if EditView = nil then
            Continue;
```

メソッドの有害度値: 10.1 Berlin~

- メソッドの有害度を表示します。
 - 例えば、長すぎるメソッドだったり、引数が多い、また、どれだけ複雑な実行経路があるかを示したりできます。
 - 保守のためにメソッドを分割するなどのリファクタリングに役に立ちます。

メソッドの有害度値...

メソッド名	長さ	パラメータ	If 文の深さ	サイクロマティック複雑度	有害度 ▼
TFormatter.FormatterClick	71	1	3	15	2.013
TFormatter.ExecCommand	27	1	1	3	1.087
TFormatter.ExecCommand.R...	11	0	2	4	0.404
Initialize	9	0	2	3	0.338
Finalize	4	0	1	2	0.183
TFormatter.FormatterClick.S...	1	1	0	1	0.096

ブックマークスタック: 10.1 Berlin~

ブックマークを逆順にたどりながらブックマークを削除する

```
220 // 変更格納先テンポラリファイル作成
    Target := TPath.GetTempFileName;
    try
        TFile.Copy(Editor.FileName, Target, True);
    except on E: Exception do
        begin
            ShowMsg(E.ToString);
            Exit;
        end;
    end;
230
    if not TFile.Exists(Target) then
    begin
        ShowMsg(ERR_MSG_CREATE_TEMPFILE);
        Exit;
    end;
    // 実行
238 Res := ExecCommand(Format(COMMAND_LINE, [Formatter, Target]));
240 // 変更内容読み込み
    with TStringList.Create do
    try
        LoadFromFile(Target);
        Contents := Text;
    finally
        DisposeOf;
    end;
```

CTRL + Q + Ctrl + G

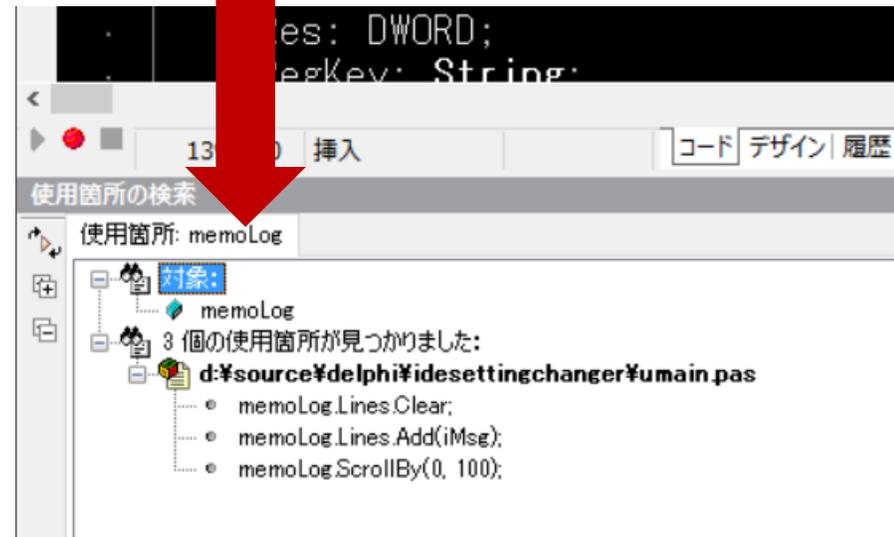
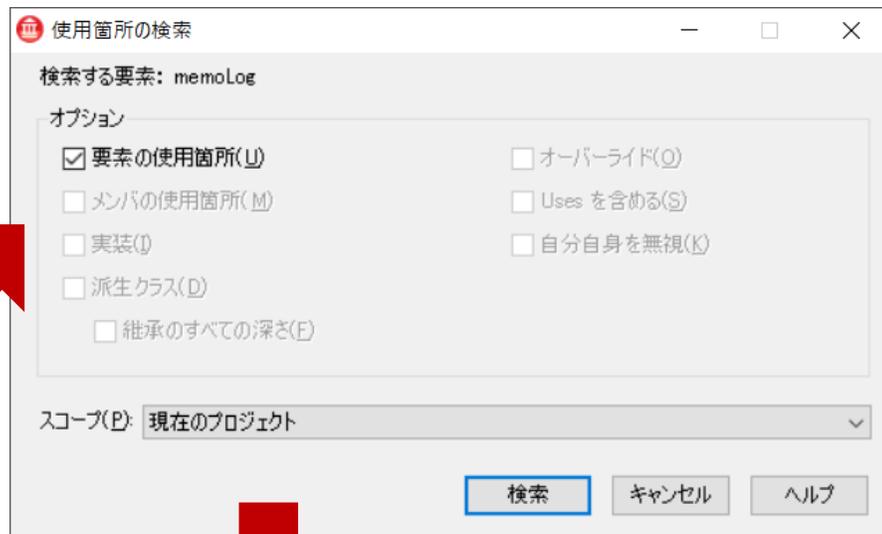
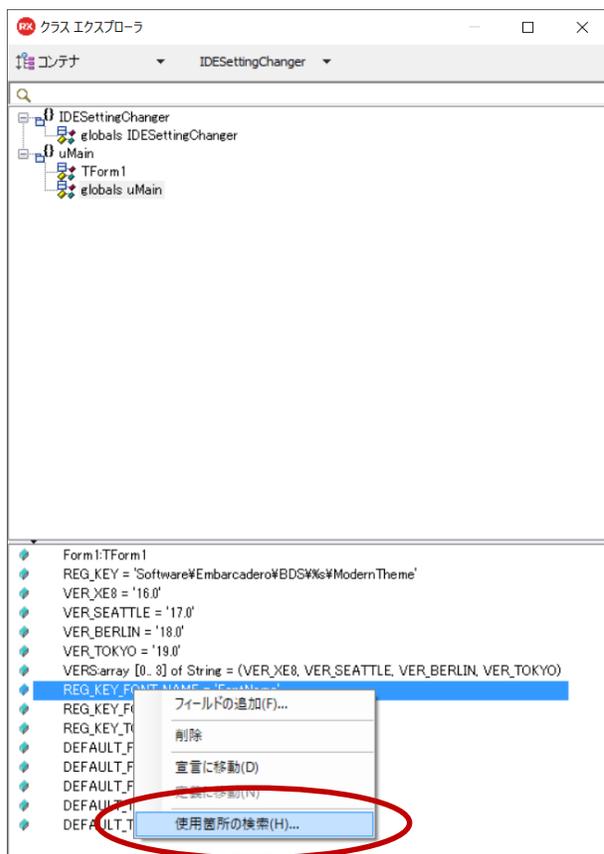
ブックマークのところに自動的にカーソルが移動します。
(移動後、そのブックマークは消えます)

```
220 // 変更格納先テンポラリファイル作成
221 Target := TPath.GetTempFileName;
    try
        TFile.Copy(Editor.FileName, Target, True);
    except on E: Exception do
        begin
            ShowMsg(E.ToString);
        end;
    end;
230
    if not TFile.Exists(Target) then
    begin
        ShowMsg(ERR_MSG_CREATE_TEMPFILE);
        Exit;
    end;
    // 実行
    Res := ExecCommand(Format(COMMAND_LINE, [Formatter, Target]));
240 // 変更内容読み込み
    with TStringList.Create do
    try
```

CTRL + Q + Ctrl + G

[使用箇所の検索] ダイアログ ボックス: XE~ / VCL + FMX

- コードエディタで右クリック
- クラスエクスプローラのメンバを右クリック



スレッド名の設定: XE (2010?) ~

- 実行中のスレッドに名前を設定できます。
 - これによってスレッドの識別がしやすくなりデバッグに役立ちます。

スレッドの状態

スレッド ID	状況	状態	位置
IDESettingChange...			
8912	実行中		\$752029BC
18132			\$7789396C
10064			\$7789396C
11444			\$7789396C

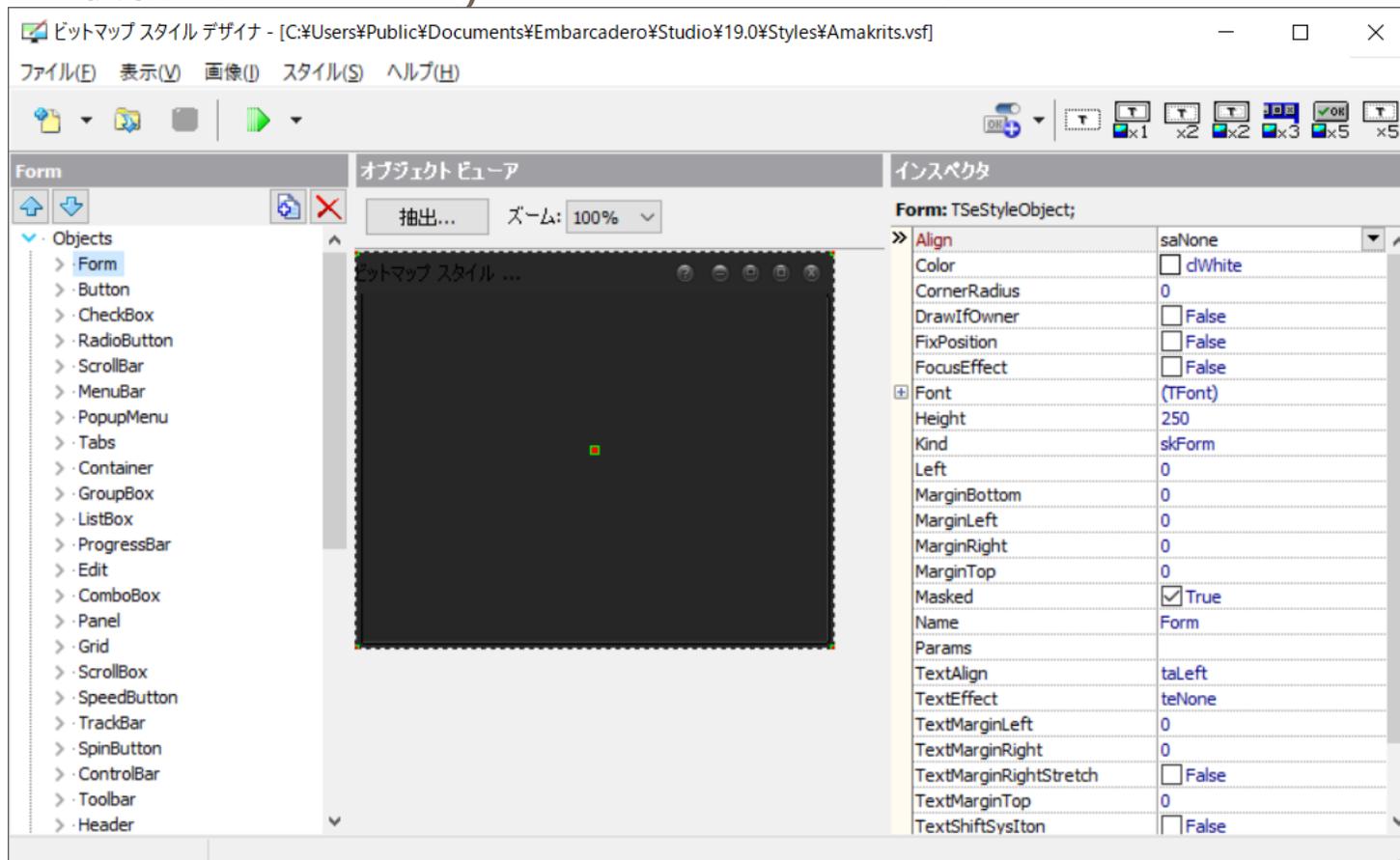
スレッド名 (I):
MainThread

スレッドの状態

スレッド ID	状況	状態	位置
IDESettingChanger.ex...			
MainThread (8912)	実行中		\$752029BC
14096	実行中		\$7789396C
16020	実行中		\$778922CC
7140	実行中		\$7789396C

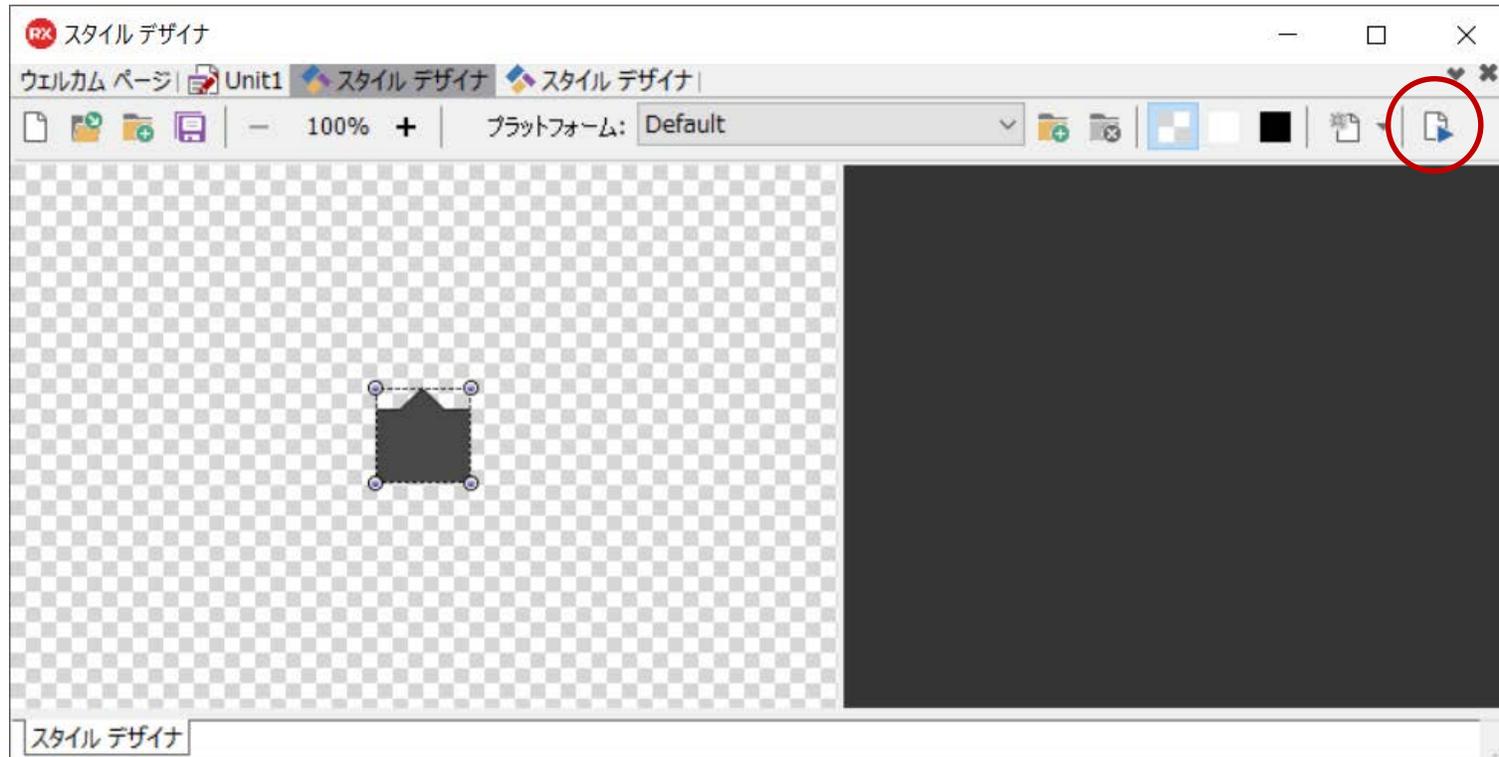
ビットマップスタイルデザイナー: XE3~ / VCL, XE5 ~ / FMX

- ビットマップ形式のスタイルを作成するエディタです
 - 最初は VCL Style のみをサポートしていました



スタイルデザイナー: XE4~ / FMX

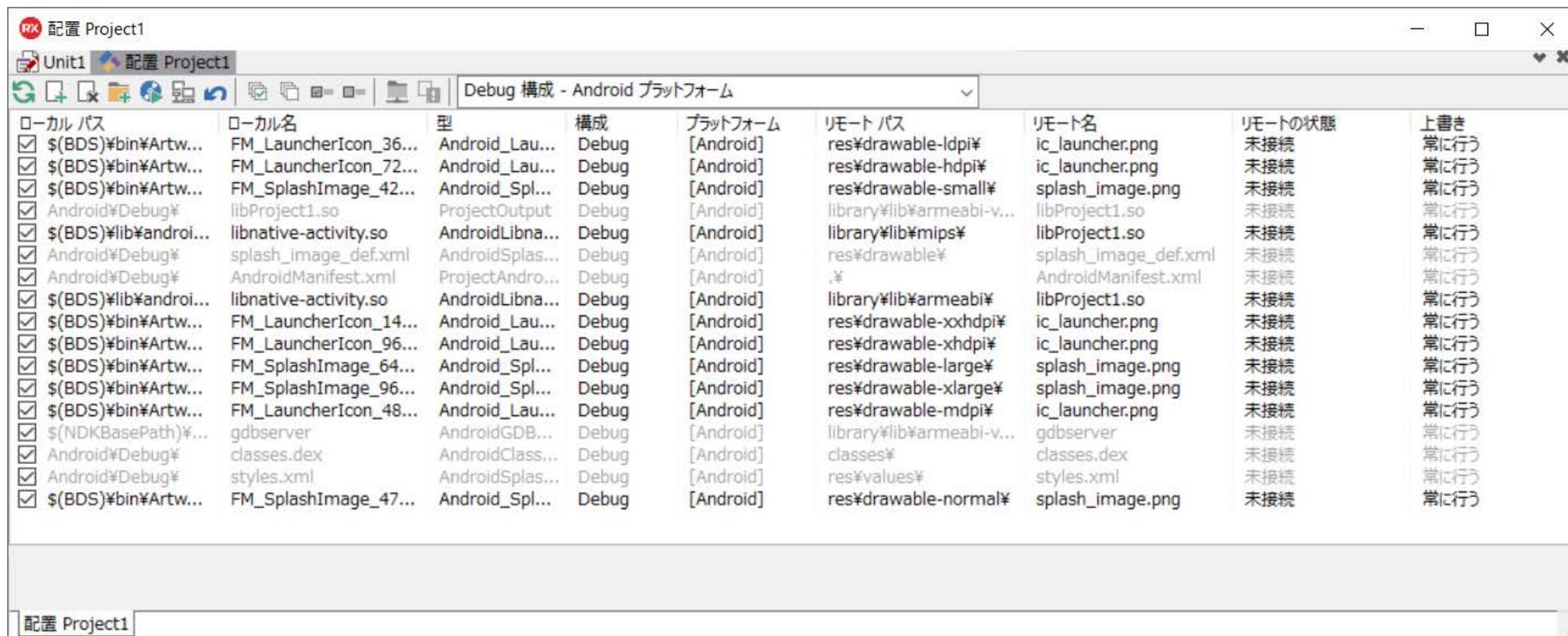
- TStyleBook のエディタです



一時期無くなっていた
適用ボタンが復活しました

配置マネージャ: XE5~

- アプリケーションの動作に必要なファイルをアプリのパッケージに含めて、初回起動時に指定の場所に配置してくれる機能です。

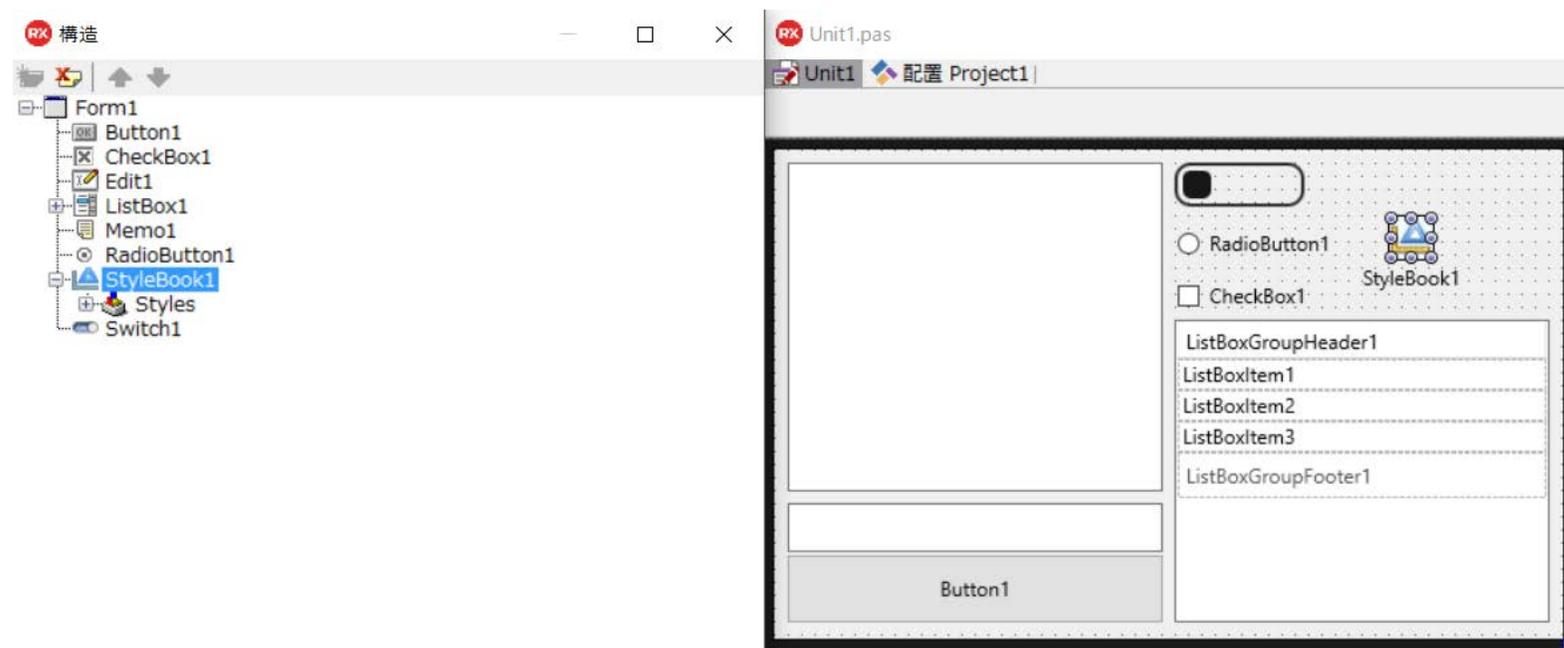


ローカルパス	ローカル名	型	構成	プラットフォーム	リモートパス	リモート名	リモートの状態	上書き
<input checked="" type="checkbox"/> \$(BDS)¥bin¥Artw...	FM_LauncherIcon_36...	Android_Lau...	Debug	[Android]	res¥drawable-ldpi¥	ic_launcher.png	未接続	常に行う
<input checked="" type="checkbox"/> \$(BDS)¥bin¥Artw...	FM_LauncherIcon_72...	Android_Lau...	Debug	[Android]	res¥drawable-hdpi¥	ic_launcher.png	未接続	常に行う
<input checked="" type="checkbox"/> \$(BDS)¥bin¥Artw...	FM_SplashImage_42...	Android_Spl...	Debug	[Android]	res¥drawable-small¥	splash_image.png	未接続	常に行う
<input checked="" type="checkbox"/> Android¥Debug¥	libProject1.so	ProjectOutput	Debug	[Android]	library¥lib¥armeabi-v...	libProject1.so	未接続	常に行う
<input checked="" type="checkbox"/> \$(BDS)¥lib¥androi...	libnative-activity.so	AndroidLibna...	Debug	[Android]	library¥lib¥mips¥	libProject1.so	未接続	常に行う
<input checked="" type="checkbox"/> Android¥Debug¥	splash_image_def.xml	AndroidSplas...	Debug	[Android]	res¥drawable¥	splash_image_def.xml	未接続	常に行う
<input checked="" type="checkbox"/> Android¥Debug¥	AndroidManifest.xml	ProjectAndro...	Debug	[Android]	¥	AndroidManifest.xml	未接続	常に行う
<input checked="" type="checkbox"/> \$(BDS)¥lib¥androi...	libnative-activity.so	AndroidLibna...	Debug	[Android]	library¥lib¥armeabi¥	libProject1.so	未接続	常に行う
<input checked="" type="checkbox"/> \$(BDS)¥bin¥Artw...	FM_LauncherIcon_14...	Android_Lau...	Debug	[Android]	res¥drawable-xxhdpi¥	ic_launcher.png	未接続	常に行う
<input checked="" type="checkbox"/> \$(BDS)¥bin¥Artw...	FM_LauncherIcon_96...	Android_Lau...	Debug	[Android]	res¥drawable-xhdpi¥	ic_launcher.png	未接続	常に行う
<input checked="" type="checkbox"/> \$(BDS)¥bin¥Artw...	FM_SplashImage_64...	Android_Spl...	Debug	[Android]	res¥drawable-large¥	splash_image.png	未接続	常に行う
<input checked="" type="checkbox"/> \$(BDS)¥bin¥Artw...	FM_SplashImage_96...	Android_Spl...	Debug	[Android]	res¥drawable-xlarge¥	splash_image.png	未接続	常に行う
<input checked="" type="checkbox"/> \$(BDS)¥bin¥Artw...	FM_LauncherIcon_48...	Android_Lau...	Debug	[Android]	res¥drawable-mdpi¥	ic_launcher.png	未接続	常に行う
<input checked="" type="checkbox"/> \$(NDKBasePath)¥...	gdbserver	AndroidGDB...	Debug	[Android]	library¥lib¥armeabi-v...	gdbserver	未接続	常に行う
<input checked="" type="checkbox"/> Android¥Debug¥	classes.dex	AndroidClass...	Debug	[Android]	classes¥	classes.dex	未接続	常に行う
<input checked="" type="checkbox"/> Android¥Debug¥	styles.xml	AndroidSplas...	Debug	[Android]	res¥values¥	styles.xml	未接続	常に行う
<input checked="" type="checkbox"/> \$(BDS)¥bin¥Artw...	FM_SplashImage_47...	Android_Spl...	Debug	[Android]	res¥drawable-normal¥	splash_image.png	未接続	常に行う

構造ビュー: XE4~

- フォームやコードの構造を表示するビューです
 - フォームデザイナーか、コードエディタか、で内容が変わります。
 - フォームデザイナーでは、親子関係や順序をドラッグ&ドロップで変更できます。

フォームデザイナー時



構造ビュー: XE4~

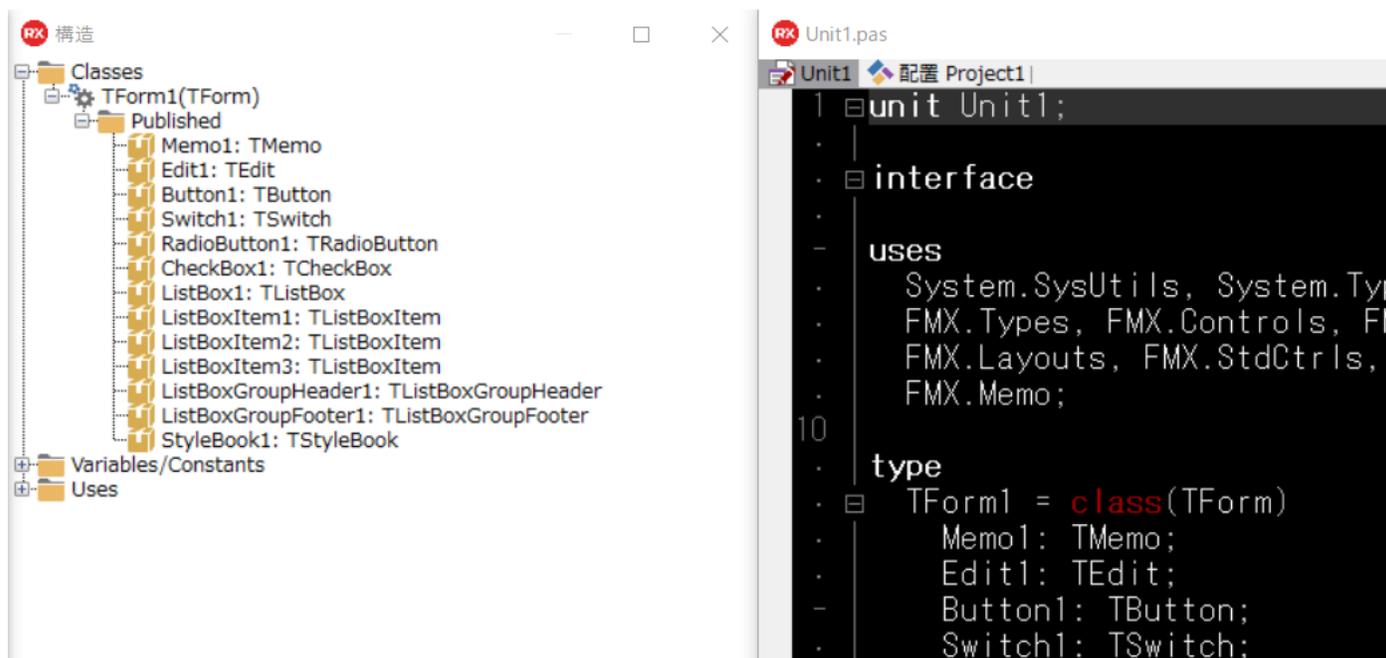
- フォームやコードの構造を表示するビューです



構造ビュー: XE4~

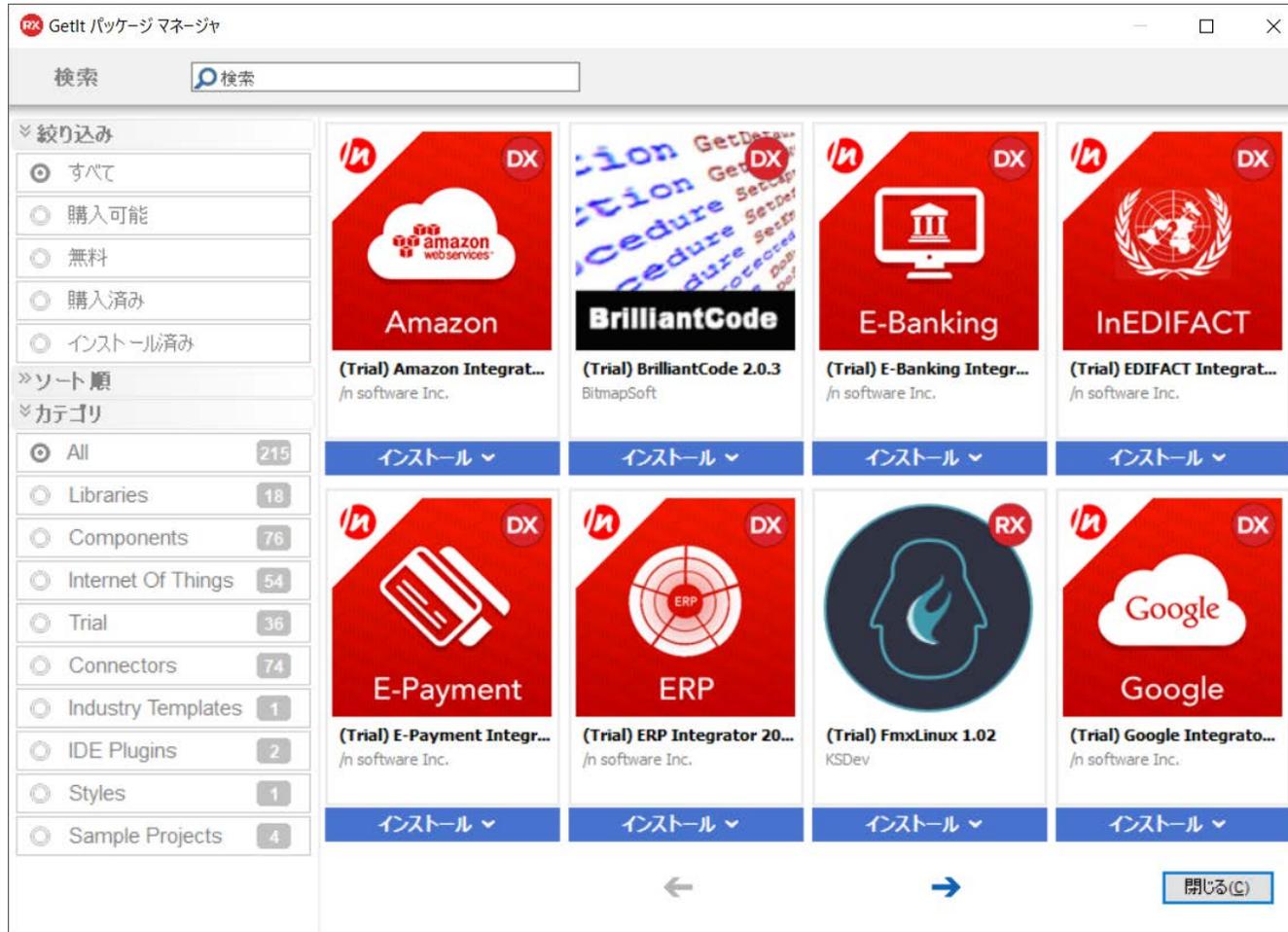
- フォームやコードの構造を表示するビューです
 - コードエディタではクラスエクスプローラのような表示になります。

コードエディタ時



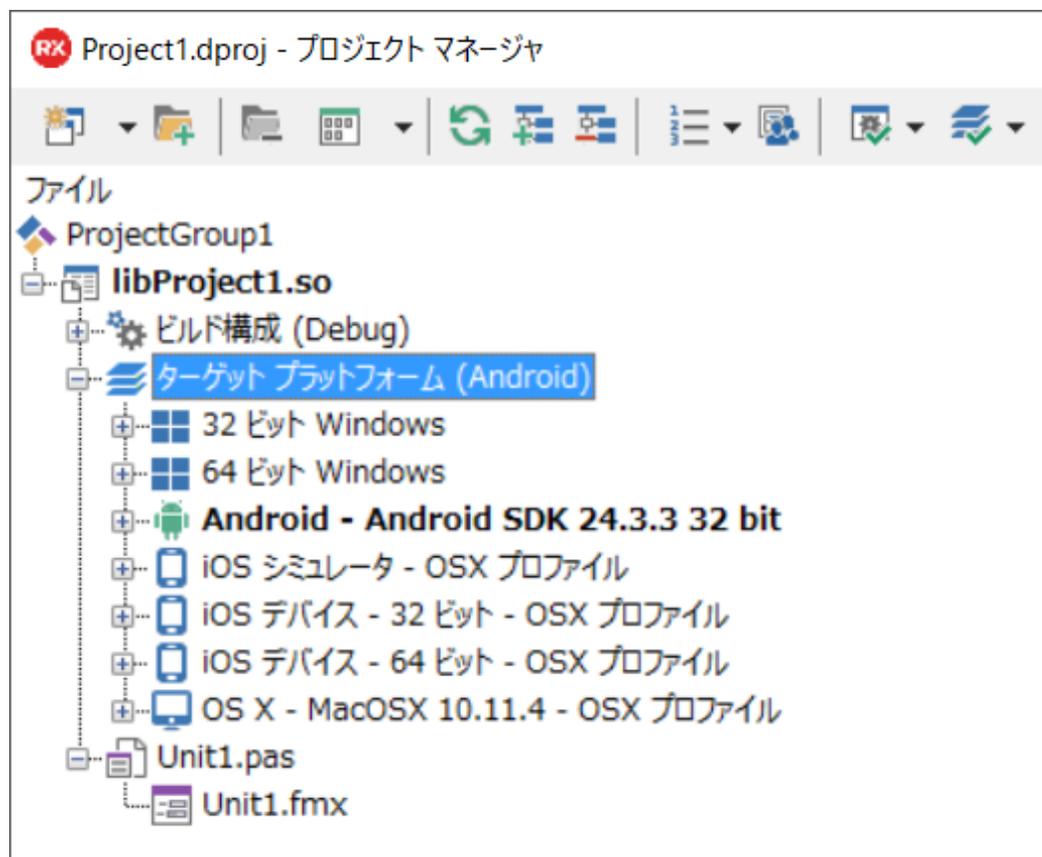
GetIT: 10 Seattle

- パッケージマネージャです



プロジェクトマネージャの新機能: XE以降まとめて

- ターゲットプラットフォーム
 - マルチプラットフォームになったことでビルドするターゲットOSを指定できるようになりました



プロジェクトマネージャの新機能: XE以降まとめて

- アプリの動作に必要なファイルの配置
 - 配置マネージャではなくプロジェクトマネージャに追加して配置することも可能です。

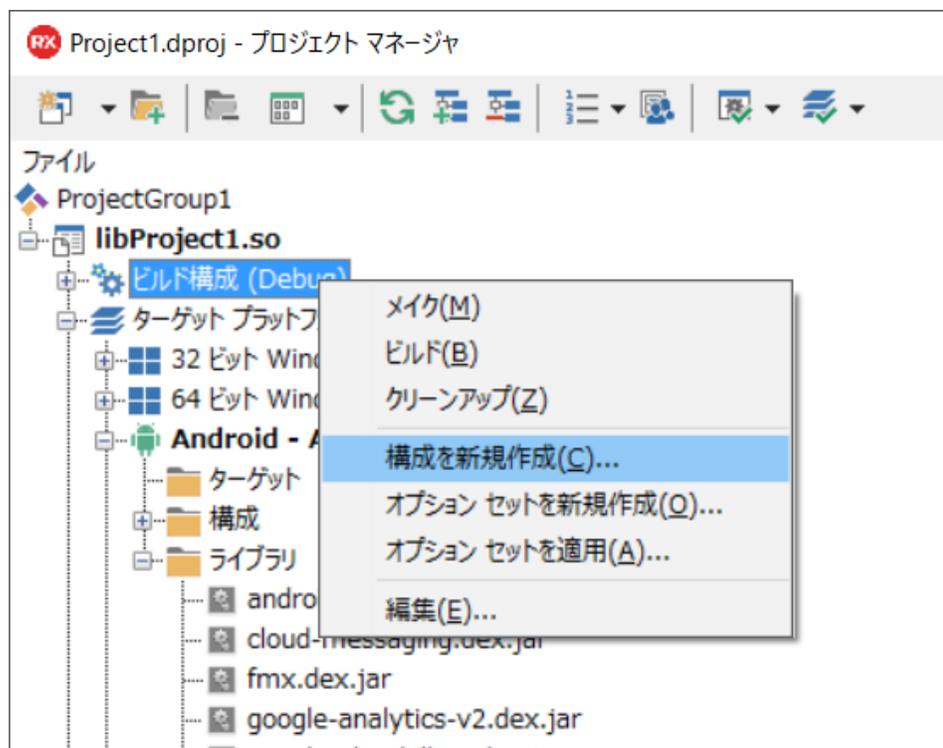
ローカルパス	ローカル名	型	構成	プラットフォーム	リモートパス	リモート名	リモートの状態	上書き
\$(BDS)¥bin¥Artw...	FM_LauncherIcon_36...	Android_Lau...	Debug	[Android]	res¥drawable-ldpi¥	ic_launcher.png	未接続	常に行う
✓ D:¥Temp¥	ubu.png	ProjectFile	Debug	[Android]	¥assets¥internal¥	ubu.png	未接続	常に行う
✓ \$(BDS)¥bin¥Artw...	FM_Laun...	Android_Lau...	Debug	[Android]	res¥drawable-hdpi¥	ic_launcher.png	未接続	常に行う
✓ \$(BDS)¥bin¥Artw...	FM_Spl...	Android_Spl...	Debug	[Android]	res¥drawable-small¥	splash_image.png	未接続	常に行う
✓ Android¥Debug¥	libProject...	ProjectOutput	Debug	[Android]	library¥lib¥armeabi-v...	libProject1.so	未接続	常に行う
✓ \$(BDS)¥lib¥androi...	libnative...	AndroidLibna...	Debug	[Android]	library¥lib¥mips¥	libProject1.so	未接続	常に行う
✓ Android¥Debug¥	splash_image_...	AndroidSplas...	Debug	[Android]	res¥drawable¥	splash_image_def.xml	未接続	常に行う
✓ Android¥Debug¥	AndroidManifest...	ProjectAndro...	Debug	[Android]	¥	AndroidManifest.xml	未接続	常に行う
✓ \$(BDS)¥lib¥androi...	libnative-activity.so	AndroidLibna...	Debug	[Android]	library¥lib¥armeabi¥	libProject1.so	未接続	常に行う
✓ \$(BDS)¥bin¥Artw...	FM_LauncherIcon_14...	Android_Lau...	Debug	[Android]	res¥drawable-xxhdpi¥	ic_launcher.png	未接続	常に行う
✓ \$(BDS)¥bin¥Artw...	FM_LauncherIcon_96...	Android_Lau...	Debug	[Android]	res¥drawable-xhdpi¥	ic_launcher.png	未接続	常に行う
✓ \$(BDS)¥bin¥Artw...	FM_SplashImage_64...	Android_Spl...	Debug	[Android]	res¥drawable-large¥	splash_image.png	未接続	常に行う
✓ \$(BDS)¥bin¥Artw...	FM_SplashImage_96...	Android_Spl...	Debug	[Android]	res¥drawable-xlarge¥	splash_image.png	未接続	常に行う
✓ \$(BDS)¥bin¥Artw...	FM_LauncherIcon_48...	Android_Lau...	Debug	[Android]	res¥drawable-mdpi¥	ic_launcher.png	未接続	常に行う
✓ \$(NDKBasePath)¥...	gdbserver	AndroidGDB...	Debug	[Android]	library¥lib¥armeabi-v...	gdbserver	未接続	常に行う
✓ Android¥Debug¥	classes.dex	AndroidClass...	Debug	[Android]	classes¥	classes.dex	未接続	常に行う
✓ Android¥Debug¥	styles.xml	AndroidSplas...	Debug	[Android]	res¥values¥	styles.xml	未接続	常に行う
✓ \$(BDS)¥bin¥Artw...	FM_SplashImage_47...	Android_Spl...	Debug	[Android]	res¥drawable-normal¥	splash_image.png	未接続	常に行う

ドロップすると自動的に配置マネージャに追加されます

プロジェクトマネージャの新機能: XE以降まとめて

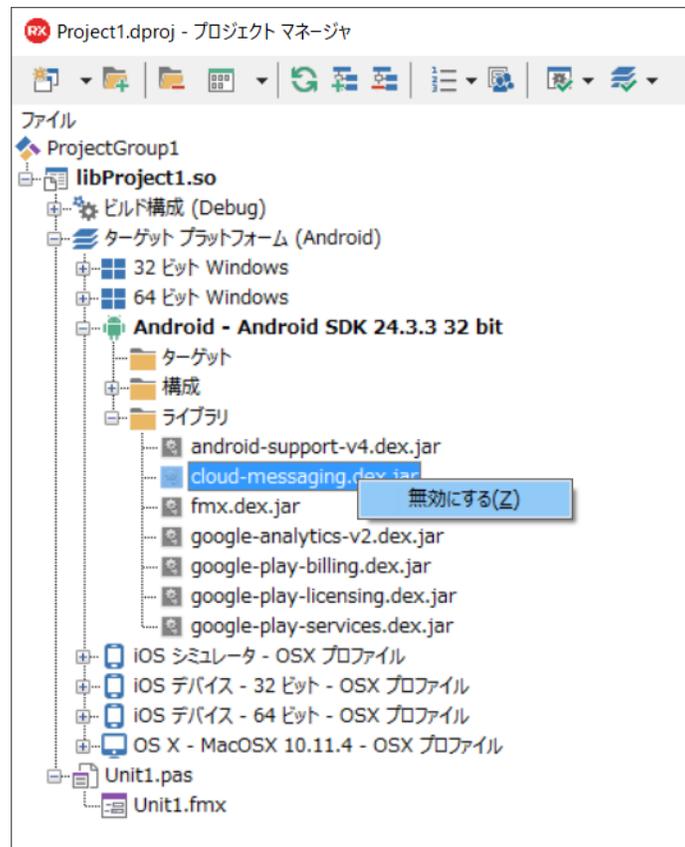
■ ビルド構成

- Release / Debug はもちろんカスタムのビルド構成も作れます。



プロジェクトマネージャの新機能: XE以降まとめて

- Android 向け組み込み RAD Studio Java ライブラリの無効化
 - 作成しているアプリに必要な無いライブラリを無効化して容量を削減できるようになりました



その他の機能

- 選択範囲の拡張: 10.1 Berlin～
 - 構造的なブロック単位で拡張できます
- エディタのフォントサイズ変更: 10.0 Seattle～
 - Ctrl + Num + / Ctrl + Num - で変更できます
- Version Insight: XE～
 - Git / Subversion / Mercurial と連携できます。
- Visual LiveBindings: XE3～
 - LiveBindings をビジュアルに編集できます。

その他の機能

- 設定移行ツール: XE8～
 - Delphi のバージョンを上げる際に、旧バージョンからレジストリの内容などを持ってきて、新バージョンのレジストリに書き込み移行を手助けします。
- ListView 項目デザイナー: 10.1 Berlin～ / FMX
 - 設計モードを使って設計時にビジュアルな外観をカスタマイズできるようになりました

■ コマンドラインツール



embarcadero®
DEVELOPER CAMP

PASever: XE2～

- 言わずと知れた Platform Assistant Server です。
 - Win32 以外の OS でデバッグ実行する時などに使います。
 - 主に macOS / iOS のデバッグで macOS にインストールされ使われます
 - 起動後 v + Enter を入力すると詳細表示モードになり、PASever が受け取り実行した内容が表示されるようになります。
 - これによって iOS の署名失敗時などの理由が解るようになりました。

```
MyMac:~ myname$ /Applications/PASever-16.0.app/Contents/MacOS/paserver ; exit;
Platform Assistant Server Version 7.0.1.40
Copyright (c) 2009-2015 Embarcadero Technologies, Inc.

Connection Profile password <press Enter for no password>:

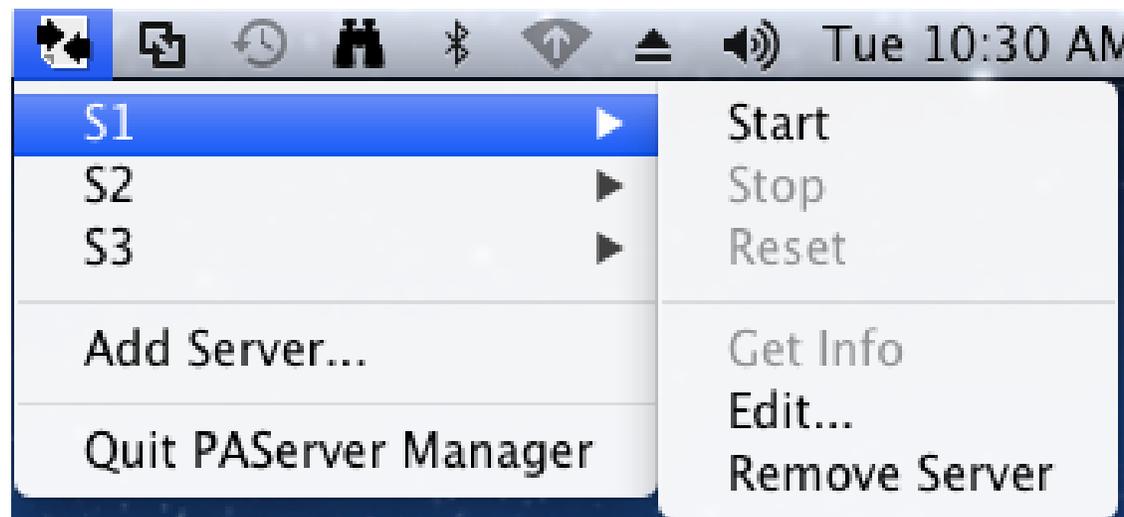
Acquiring permission to support debugging...succeeded

Starting Platform Assistant Server on port 64211

Type ? for available commands
>
```

PA Server Manager: XE7~

- 複数のバージョンの PA Server を切り替えるアプリケーションです
 - macOS で動作します。



Java2OP: XE5~

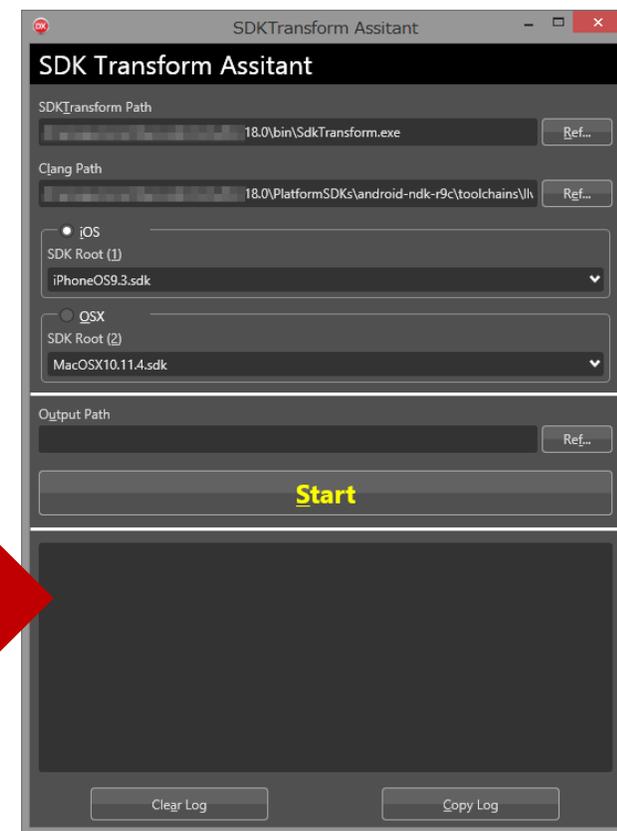
- Android API や class / jar ファイルからブリッジファイルを生成します

```
.  8 unit Androidapi.JNI.SGWebClient;
.
.
9  interface
10
.  uses
.    Androidapi.JNIBridge,
.    Androidapi.JNI.JavaTypes,
.    Androidapi.JNI.Webkit,
-   Androidapi.JNI.GraphicsContentViewText,
.    Androidapi.JNI.Os,
.    Androidapi.JNI.Net;
.
.  type
20  JSGWebClient = interface;//jp.serialgames.WebClient
.
.  JSGWebClientClass = interface(JWebViewClientClass)
.    [{EF3AD3BB-A754-48BF-AAF7-08F96F695E51}]
.    {class} function init: JSGWebClient; cdecl; overload;
-   end;
.
.  [JavaSignature('jp/serialgames/SGWebClient')]
.  JSGWebClient = interface(JWebViewClient)
.    [{2CA265FA-191F-4BCC-8E77-9FEF883428E5}]
30  function getWebView: JWebView; cdecl;
.  function getHash: Integer; cdecl;
.  function getMessage1: JMessage; cdecl;
.  function getMessage2: JMessage; cdecl;
.  function getBitmap: JBitmap; cdecl;
-   function getHttpAuthHandler: JHttpAuthHandler; cdecl;
.  function getSslErrorHandler: JSslErrorHandler; cdecl;
.  function getSslError: JSslError; cdecl;
.  function getKeyEvent: JKeyEvent; cdecl;
```

SDKTransform: 10.1 Berlin~

- iOS / macOS の framework にアクセスするためのブリッジファイルを生成するツール群です
- 詳しい説明がどこにもない孤高のツールです。
 - 詳しくは拙作の Qiita で...
[SDKTransform について](http://qiita.com/pik/items/52fbbbfa0f391ad95c92)
<http://qiita.com/pik/items/52fbbbfa0f391ad95c92>

この記事中で SDKTransform を GUI で操作する
ツールを作成しています。



■ Object Pascal



Unit Scope: XE2~

- Unit 名の前に接頭辞 “Unit Scope” を指定できるようになりました。
 - 例えば System.SysUtils や Vcl.Controls, FMX.Objects などです
 - Unit Scope によって簡単なグループ分けができます。

動的配列に対する操作: XE7~

- 動的配列に対して文字列のような操作ができるようになりました

```
var
  A: array of integer;
Begin
  A := [ 1, 2, 3];
  A := A + [4, 5]; // + 演算子で追加

  // System.Insert, Delete, Concat が利用可能
  Insert(5, A, 2);
  Delete(A, 1, 2);
  A := Concat([1, 2, 3], [4, 5, 6]);
end;
```

unsafe, volatile, weak: 10.1 Berlin

- すべてのコンパイラで属性 `unsafe` / `volatile` / `weak` が使えるようになった
 - Berlin 以前ではこれらの属性はモバイルコンパイラ向けでデスクトップコンパイラでは無視されていました

Class Helper の Scope: 10.1 Berlin~

- Class Helper のアクセス範囲が変更されました
 - class helper を使った private へのアクセスができなくなりました。

```
type
  TMyClass = class
    private
      FPrivateInt : Integer;
    end;

  TMyHelper = class helper for TMyClass
    procedure Foo;
  end;

procedure TMyHelper.Foo;
begin
  Writeln(FPrivateInt); // Seattle までは OK, Berlin 以降はコンパイルエラー
end;
```

■付録

- Object Pascal の Delphi 7 以降の新機能抜粋



Delphi 7 以降の Object Pascal の機能

- 演算子 overload
- Class Helper / Record Helper
- strict private / strict protected
- 高度な record 型
- abstract / sealed 修飾子

Delphi 7 以降の Object Pascal の機能

- クラス定数
- クラス内のユーザー定義型
- ネストしたクラス
- クラスフィールド
- クラスプロパティ
- final / sealed メソッド
- 静的クラスメソッド(static)
- for-in ループ

THANKS!

www.embarcadero.com/jp

第34回 エンバカデロ・デベロッパーキャンプ