Reviewer Guide

embarcadero

Appmethod で
カメラアプリ作成体験

エンバカデロ・テクノロジーズ

2014 年 7 月

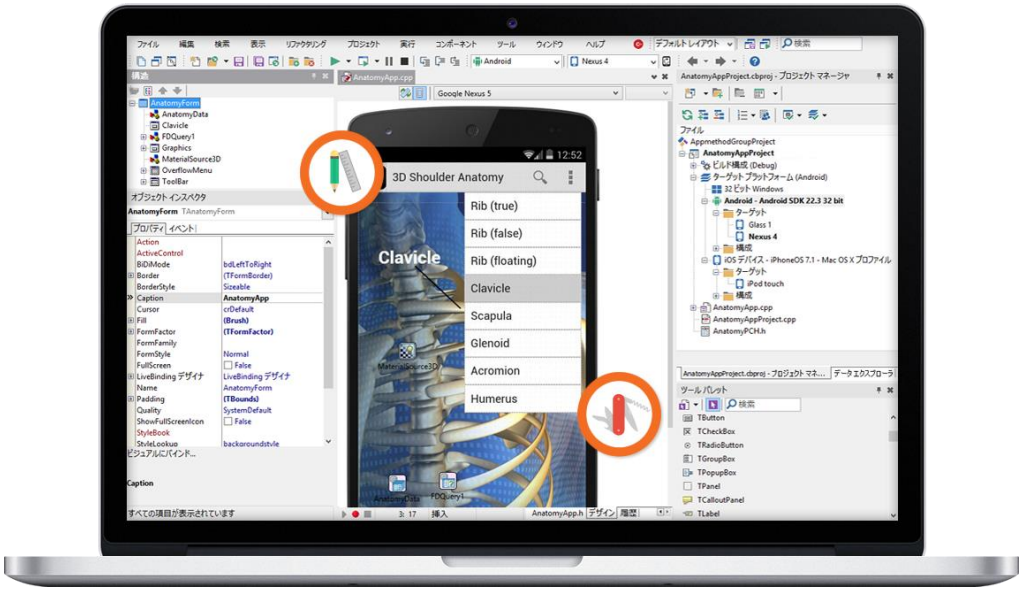
エンバカデロ・テクノロジーズ

目次

はじめに

Appmethod は、モバイルファースト開発を支援するエンバカデロの新しい開発プラットフォームです。開発者は、マルチデバイスに対応した CPU/GPU ネイティブのコンポーネントフレームワークを用いて、主要なスマートフォン、タブレット、デスクトップ向けアプリを構築できます。

Appmethod では、スマートフォンやタブレットの画面が表示されたデザイナ上にコンポーネントと呼ばれる部品を配置するビジュアルスタイルで開発を進めます。デバイス機能もコンポーネント化されており、わずかなコードでスマートフォンの機能を活用したアプリを開発できます。



今回、Appmethod の魅力を短時間で理解していただくために、簡単なカメラアプリの作成に挑戦します。本書に示された手順に従えば、デバイスのカメラ機能を用いたアプリを 10 分程度で作成することができます。

事前セットアップ

Appmethod のインストール

Appmethod をまだインストールしていない方は、http://www.appmethod.com/jp からダウンロードしてください。30 日間はマルチデバイス開発のフル機能を、それ以降も期限なしで C++による Android ベースのスマートフォン向けの開発機能を利用できます。

Appmethod をインストールするシステムの要件については、インストールノートをご覧ください。

- Appmethod インストール ノート
  http://docwiki.appmethod.com/appmethod/1.14/topics/ja/Appmethod_1.14_%E3%81%AE%E3%82%A4%E3%83%B3%E3%82%B9%E3%83%88%E3%83%BC%E3%83%AB_%E3%83%8E%E3%83%BC%E3%83%88

Android デバイスの準備

このカメラアプリ作成体験を実施するためには、Appmethod が導入済みのパソコンと USB ケーブル、そして Android 端末が必要です。この Android 端末は、あらかじめ「開発者向けオプション」を表示したうえで「USB デバッグ」オプションを有効にしておく必要があります。

詳しくは、Appmethod の docwiki を参照してください。

- Android デバイスで USB デバッグを有効にする
  http://docwiki.appmethod.com/appmethod/1.14/topics/ja/Android_%E3%83%87%E3%83%90%E3%82%A4%E3%82%B9%E3%81%A7_USB_%E3%83%87%E3%83%90%E3%83%83%E3%82%B0%E3%82%92%E6%9C%89%E5%8A%B9%E3%81%AB%E3%81%99%E3%82%8B

IDE の起動とデバイスの接続

Appmethod をインストールしたら、アイコンをクリックするかメニューから Appmethod を起動します。Appmethod の IDE（統合開発環境）が起動したら、USB ケーブルで Android デバイスを接続します。

開発環境の概要

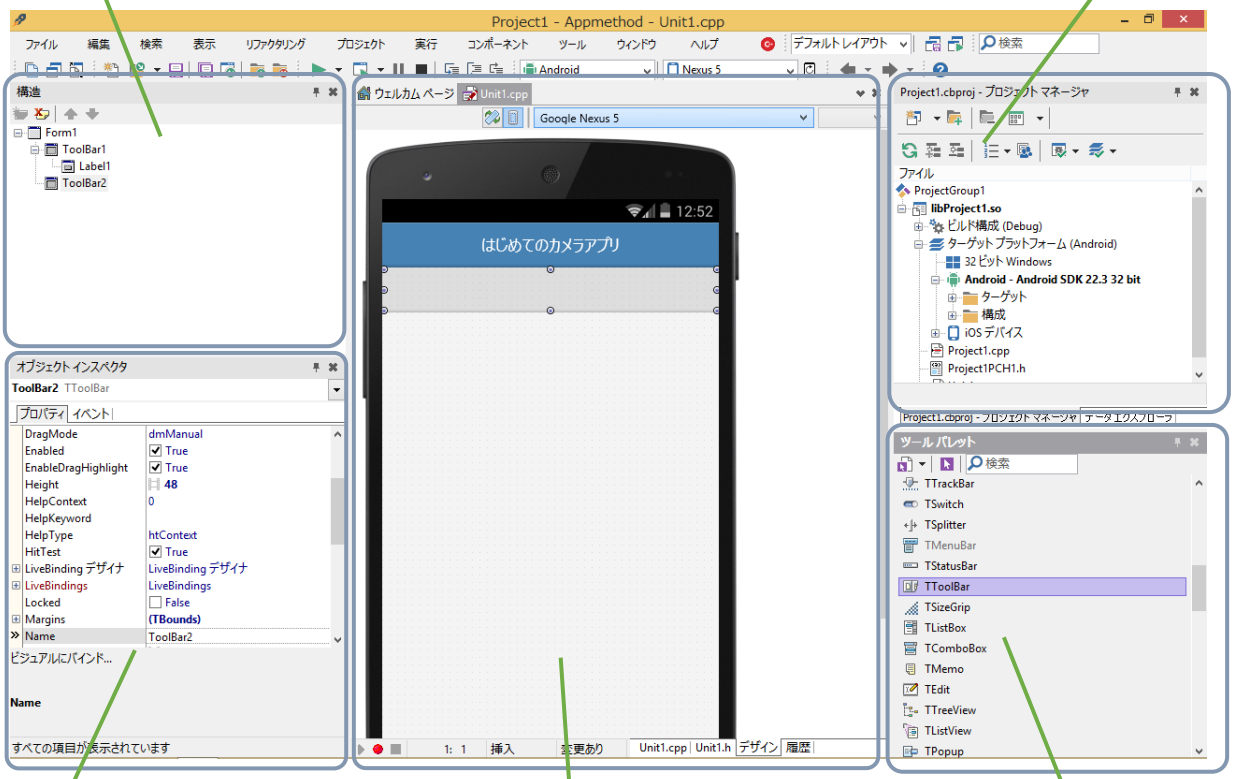
以下は、Appmethod の統合開発環境です。

構造ビュー

配置したコンポーネントの配置関係を
ツリー構造で示します。

プロジェクトマネージャ

プロジェクトで使用するファイル
やターゲットを管理します。



オブジェクトインスペクタ

配置したコンポーネントの
外観や動作を変更します。

フォームデザイナ（設計画面）

ここにコンポーネントをドラッグします。

ツールパレット

ここからコンポーネントを
選びます。

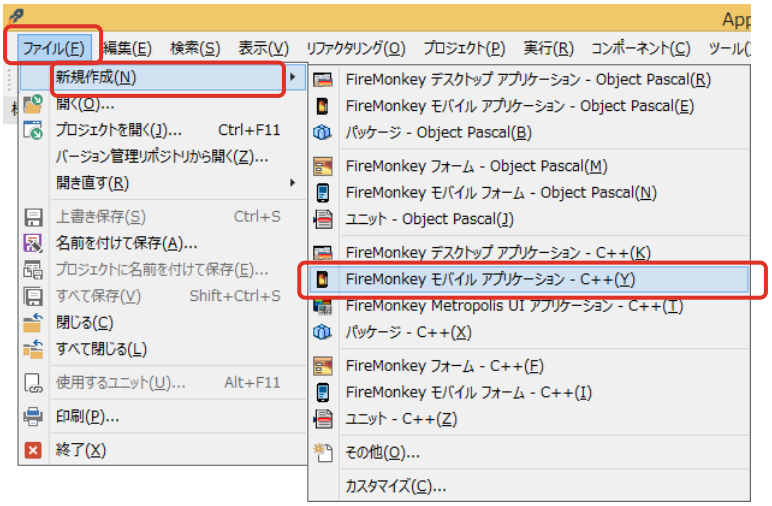
※ フォームデザイナで選択したコンポーネントの
情報が表示されます。

フェーズ 1: 画面の設計

それではカメラアプリの作成を始めましょう。最初の手順では、画面の設計を行います。

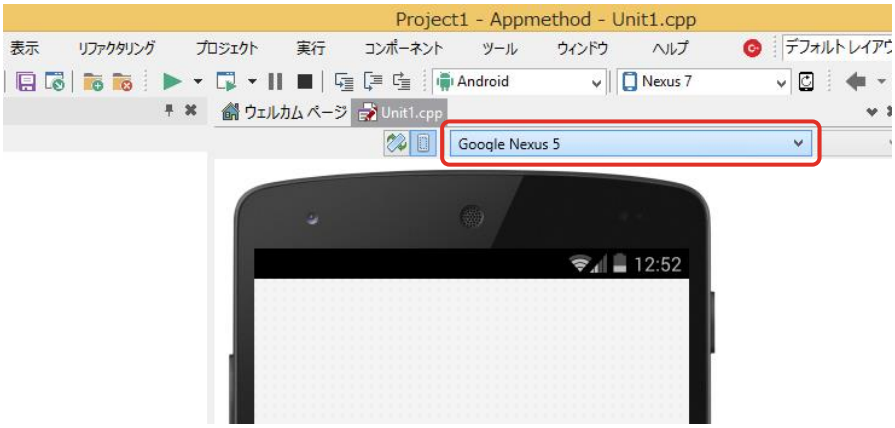
【手順 1】 「プロジェクト」の作成

手順 1-1. メニューから 「ファイル」→「新規作成」→「FireMonkey モバイルアプリケーション – C++」をクリックします。



手順 1-2.「空のアプリケーション」をクリックして、［OK］ボタンをクリックします。

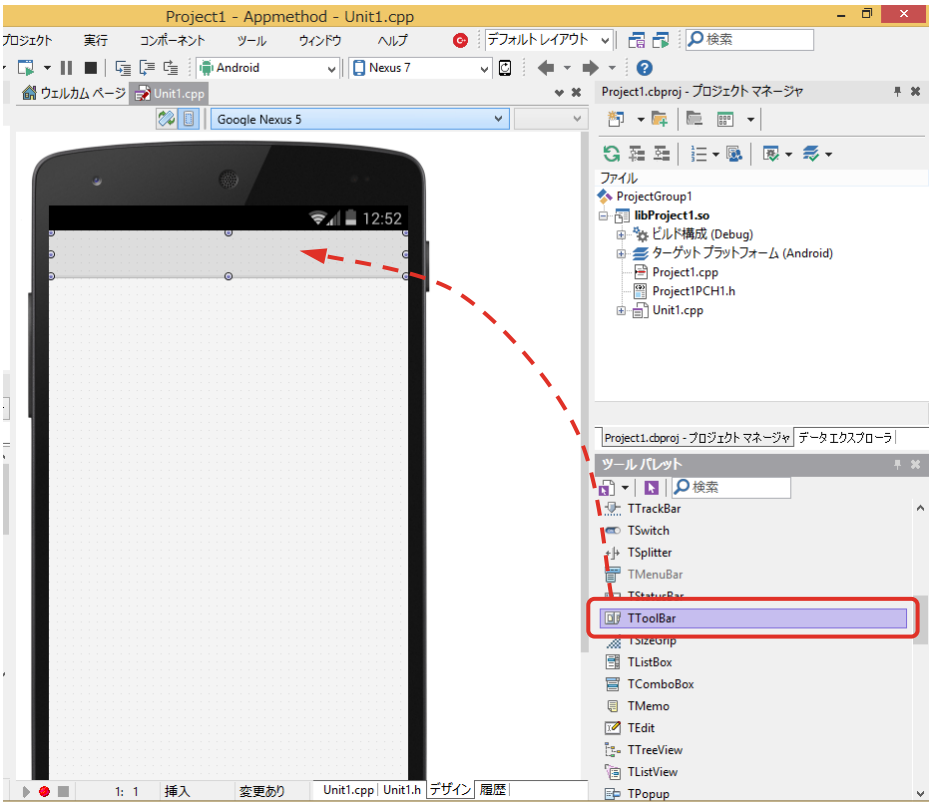
手順 1-3. フォームデザイナが表示されたら、画面中央上部のドロップダウンリスト（図参照）から「Google Nexus 5」を選択します。



この例では、Nexus 系の Android デバイスをターゲットに開発します。Appmethod では、単一のコードベースから、他の Android デバイスや iPhone / iPad もターゲットとすることができます。

【手順 2】 「ツールバー」の配置など（1）

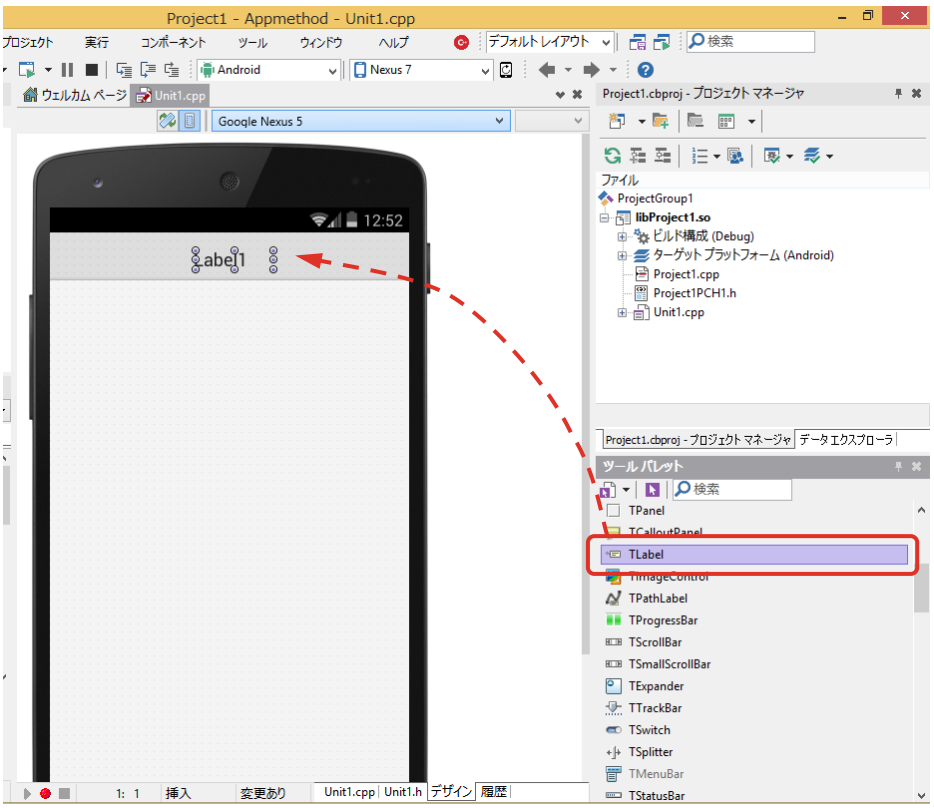
手順 2-1. 以下の手順を実施します。



最初のツールバーには、アプリのタイトルを表示します。

① 「ツールパレット」の「Standard」の［+］をクリックして展開状態にします。
② 「ツールパレット」から、Standard の中にある 「TToolBar」をドラッグしてデザインの上部にドロップします。

手順 2-2. Standard の中にある「TLabel」をドラッグしてツールバーの上にドロップします。

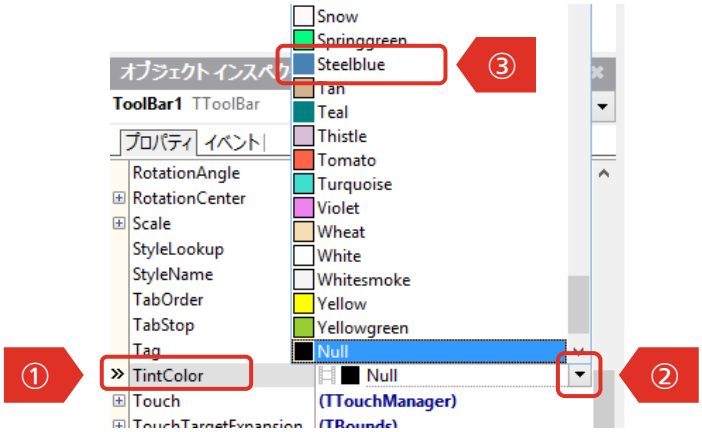


【手順 3】 「ツールバー」の設定

手順 3-1. 次にオブジェクトインスペクタを使って、配置したコンポーネントのプロパティを変更していきましょう。まず、フォームデザイナで配置したツールバーToolBar1 をクリックして選択状態にします。

手順 3-2. オブジェクトインスペクタに、ToolBar1 のプロパティが表示されます。ここで、以下の手順を実施します。

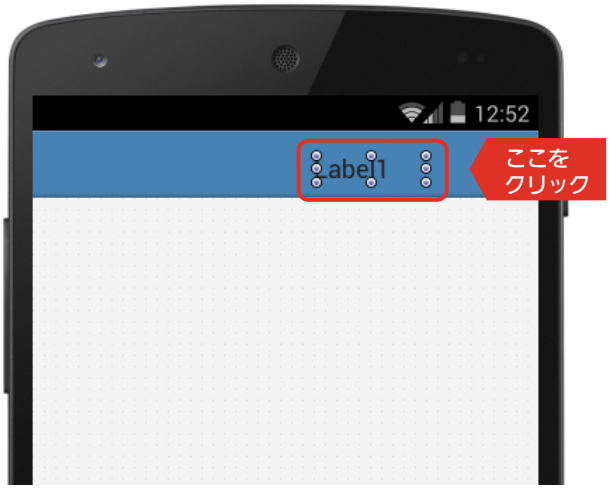
ツールバーの背景色を変更します。



① 「オブジェクトインスペクタ」の「プロパティ」の「TintColor」をクリックします。
② 「値」を示す箇所の右側の▼ボタンをクリックします。
③ 「Steelblue」をクリックして選択します。

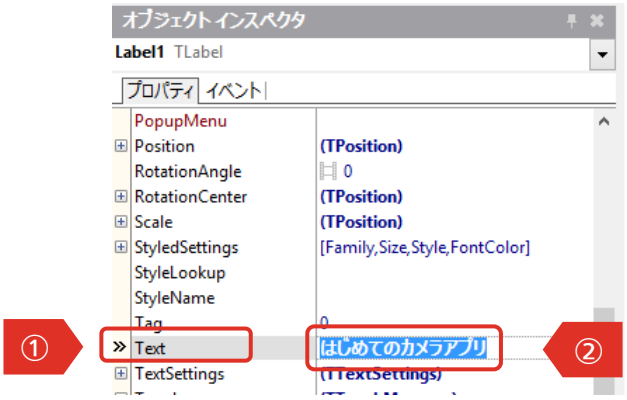
【手順 4】 「ラベル」の設定

手順 4-1. 次に、配置したラベルのテキストを設定します。フォームデザイナで、「Label1」をクリックして選択状態にします。



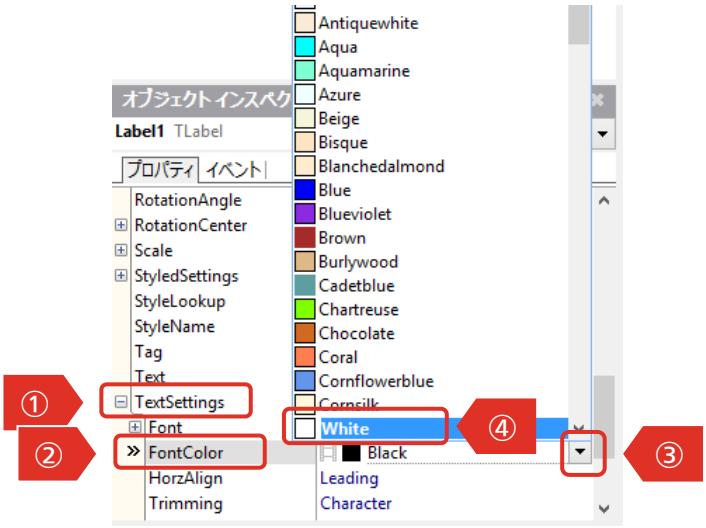
手順 4-2. オブジェクトインスペクタに、Label1 のプロパティが表示されるので、以下の手順を実施します。

アプリのタイトル文字列を指定します。



① 「オブジェクトインスペクタ」の「Text」をクリックします。
② 「値」に「はじめてのカメラアプリ」と入力して「ENTER」を押します。

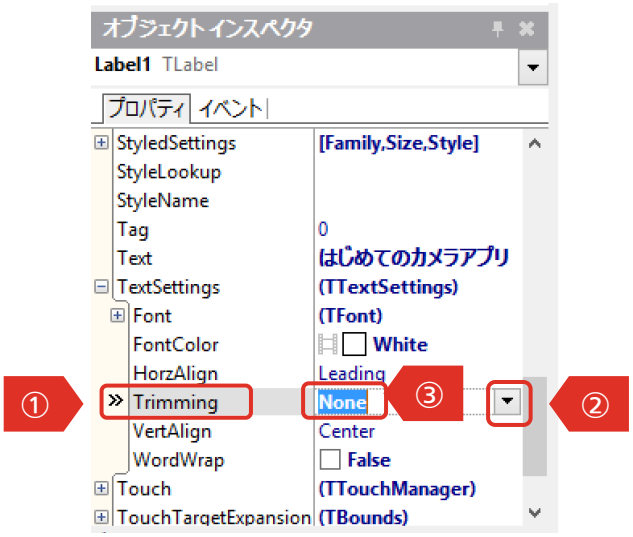
手順 4-3. 以下の手順を実施します。



① 「オブジェクトインスペクタ」の「TextSettings」をクリックします。左の［+］ボタンをクリックして展開します。
② 下にある「FontColor」をクリックします。
③ 「値」の右側の▼をクリックします。
④ 「White」をクリックして選択します。

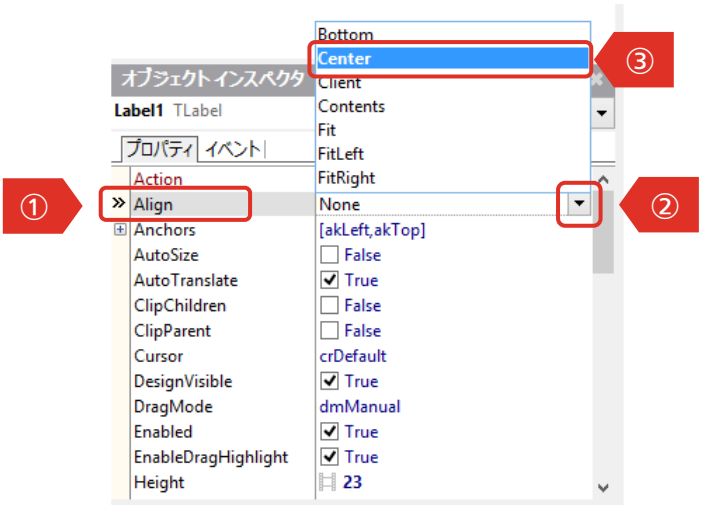
手順 4-4. 以下の手順を実施します。

以下の手順では、文字列がきれいに表示されるように設定します。



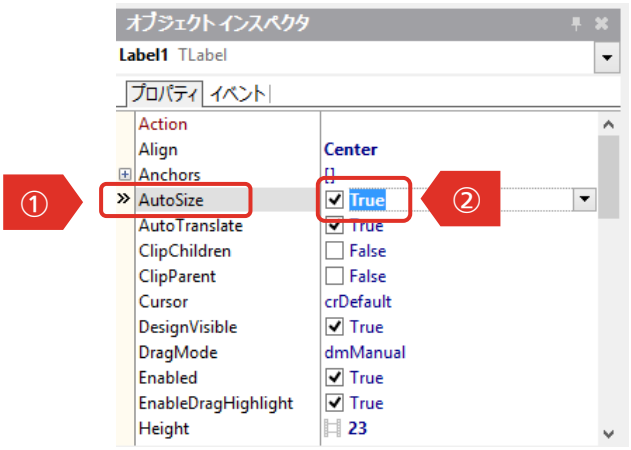
① 「オブジェクトインスペクタ」の「TextSettings」の中の「Trimming」をクリックします。
② 「値」の▼をクリックします。
③ 「None」をクリックして選択します。

手順 4-5. 以下の手順を実施します。



① 「オブジェクトインスペクタ」の「Align」をクリックします。
② 「値」の▼をクリックします。
③ 「Center」をクリックして選択します。

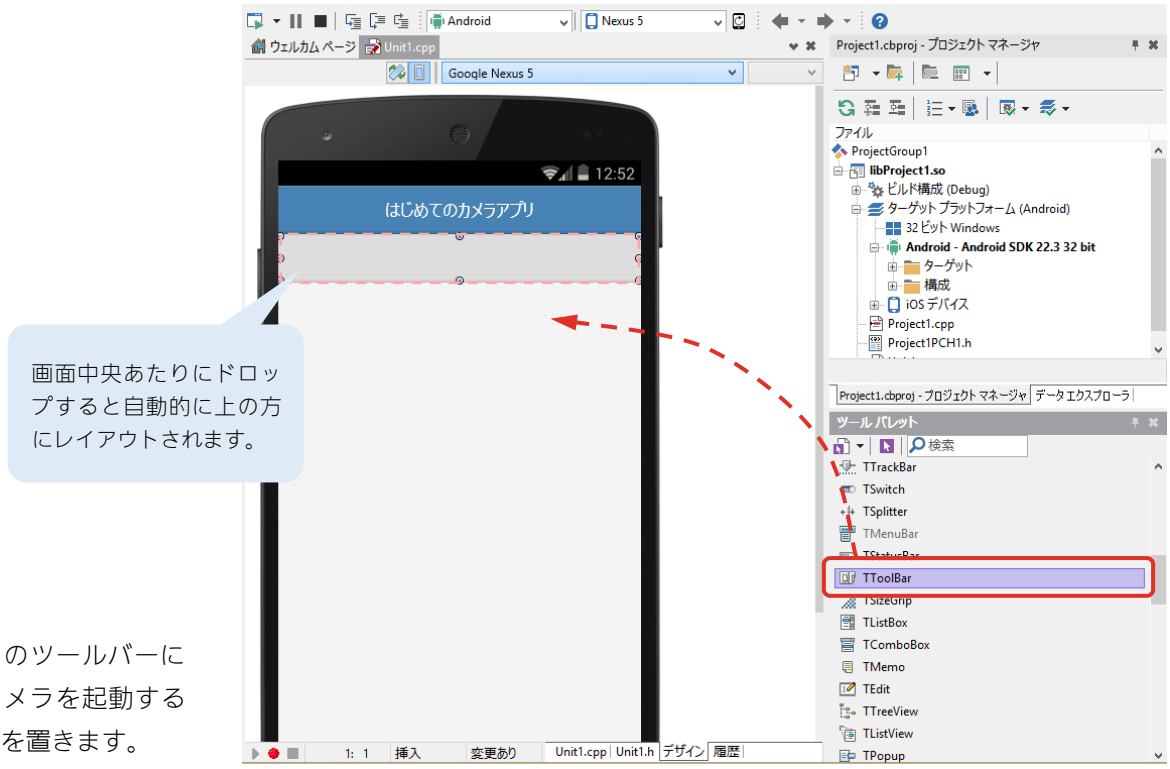
手順 4-6. 以下の手順を実施します。



① 「オブジェクトインスペクタ」の「AutoSize」をクリックします。
② 「値」の □ をクリックしてチェック状態にします。

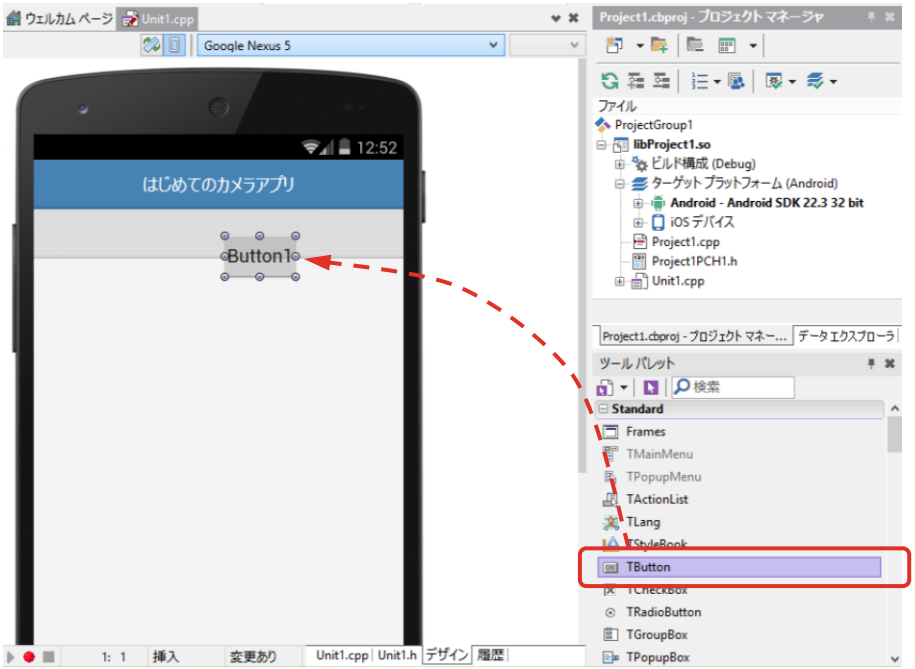
【手順 5】 「ツールバー」の配置など（2）

手順 5-1.「ツールパレット」から、Standard の中にある「TToolBar」をドラッグしてデザインフォームの中ほどにドロップします。



2 番目のツールバーには、カメラを起動するボタンを置きます。

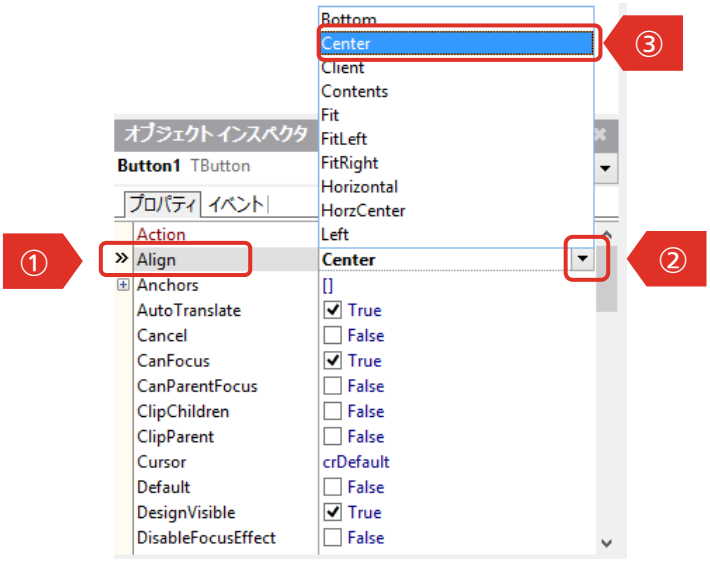
手順 5-2. Standard の中にある「TButton」をドラッグして 2 つ目のツールバーの上にドロップします。



【手順 6】「ボタン」の設定

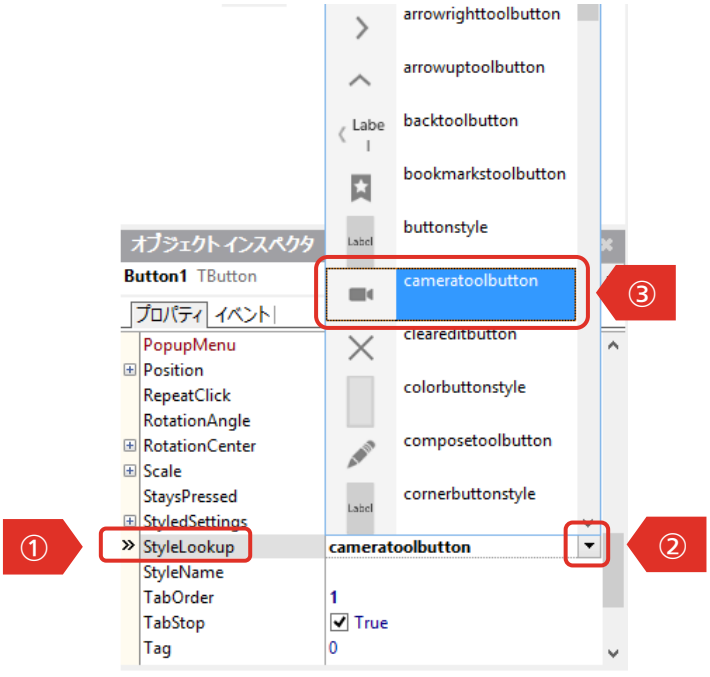
手順 6-1.「Button1」をクリックして選択状態にします。

手順 6-2. 以下の手順を実施します。



① 「オブジェクトインスペクタ」の「Align」をクリックします。
② 「値」を示す箇所の右側の▼ボタンをクリックします。
③ 「Center」をクリックして選択します。

手順 6-3. 以下の手順を実施します。

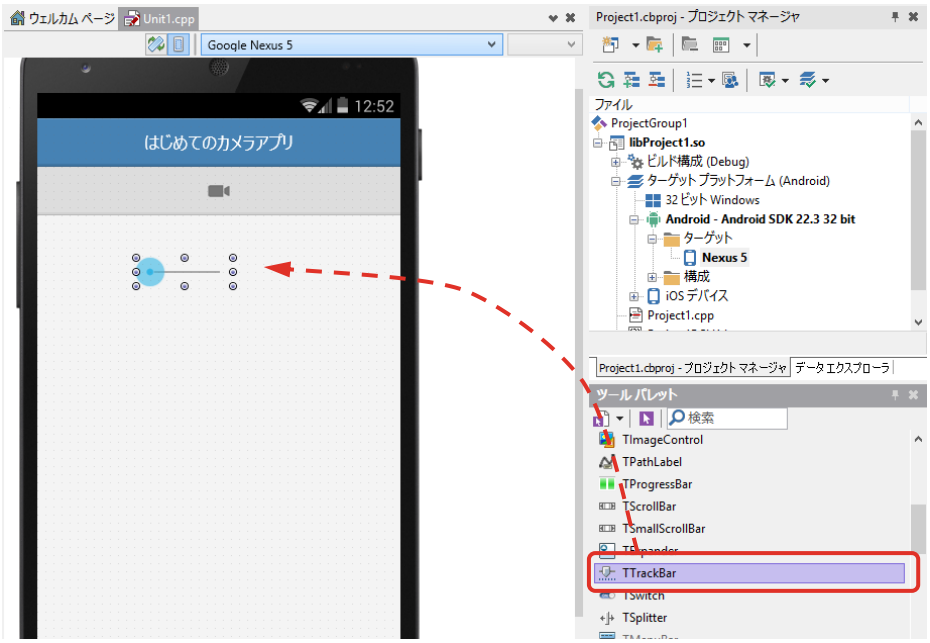


① 「オブジェクトインスペクタ」の「StyleLookup」をクリックします。
② 「値」を示す箇所の右側の▼ボタンをクリックします。
③ 「cameratoolbutton」をクリックして選択します。

【手順 7】 「トラックバー」の配置など

手順 7-1.「ツールパレット」から、Standard の中にある 「TTrackBar」をドラッグしてデザインの少し上部にドロップします。

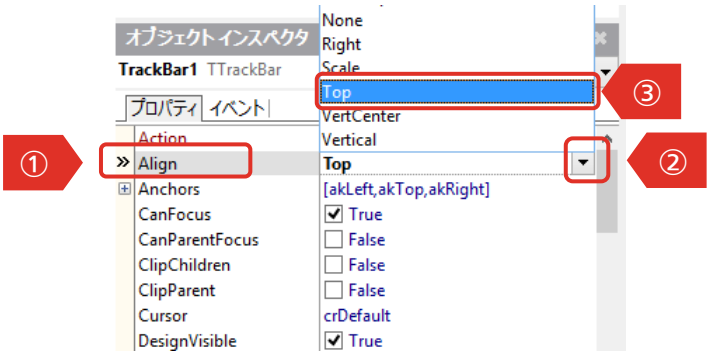
このトラックバーは、写真をセピア色に加工するときに使います。



ヒント

「ツールパレット」の「検索」ボックスにコンポーネントの名称の一部を入力することにより、ツールパレットを検索して絞り込み表示することができます。TTrackBar を検索するには、例えば ttrac と入力します。

手順 7-2. 以下の手順を実施します。



① 「オブジェクトインスペクタ」の「Align」をクリックします。
② 「値」を示す箇所の右側の▼ボタンをクリックします。
③ 「Top」をクリックして選択します。

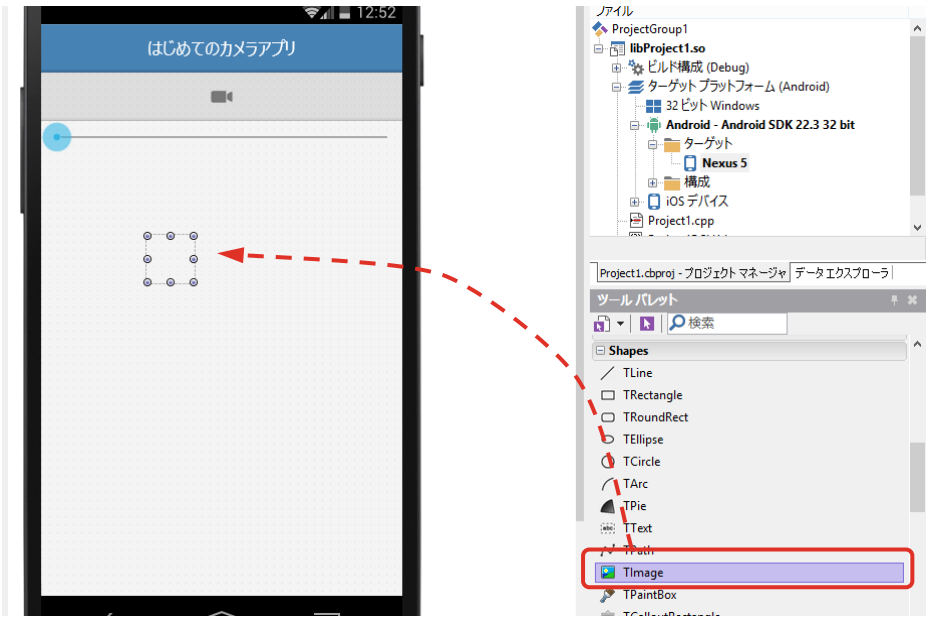
手順 7-3. 以下の手順を実施します。

① 「オブジェクトインスペクタ」の「Max」をクリックします。
② 「値」を示す箇所に「1」を入力して「ENTER」を押します。

> **注意**
>
> もし、日本語入力が開始されている場合は、ここで日本語入力を終了してください。

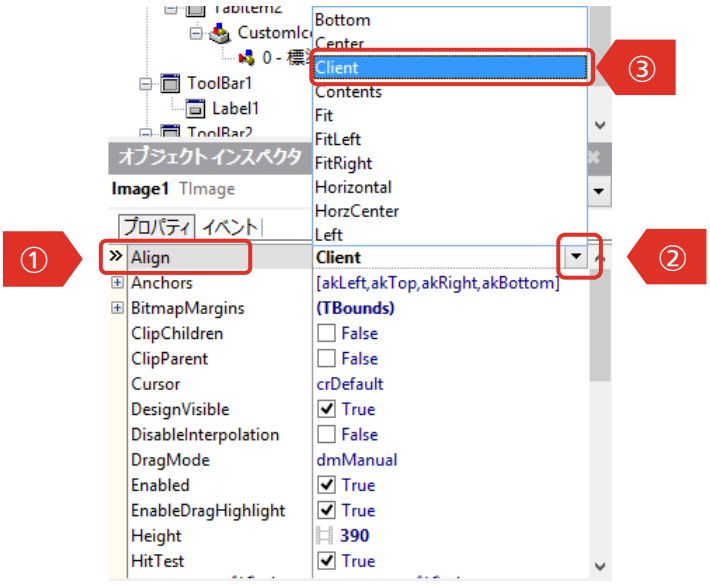
【手順 8】 「イメージ」の配置など

手順 8-1.「ツールパレット」から、Shapes の中にある 「TImage」をドラッグしてデザインフォームの中ほどにドロップします。
※「Standard」を閉じると操作しやすくなります。



アプリの中央に写真を表示するためのコンポーネントです。

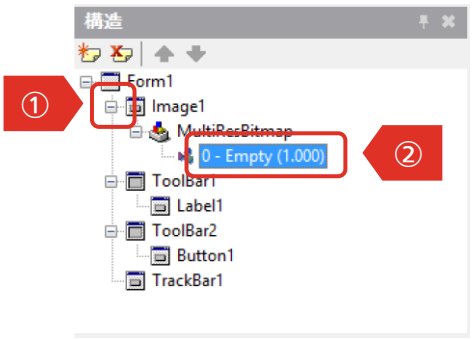
手順 8-2. 以下の手順を実施します。



① 「オブジェクトインスペクタ」の「Align」をクリックします。
② 「値」を示す箇所の右側の▼ボタンをクリックします。
③ 「Client」をクリックして選択します。

これで Image1 は、画面の残りの領域いっぱいに表示されるようになりました。

手順 8-3. 次に、Image1 に画像を読み込みます。以下の手順を実施します。



① 「構造」の「Image1」の「MultiResBitmap」の左側の[+]をクリックして展開状態にします。
② 「0 – Empty (1.000)」をクリックして選択します。

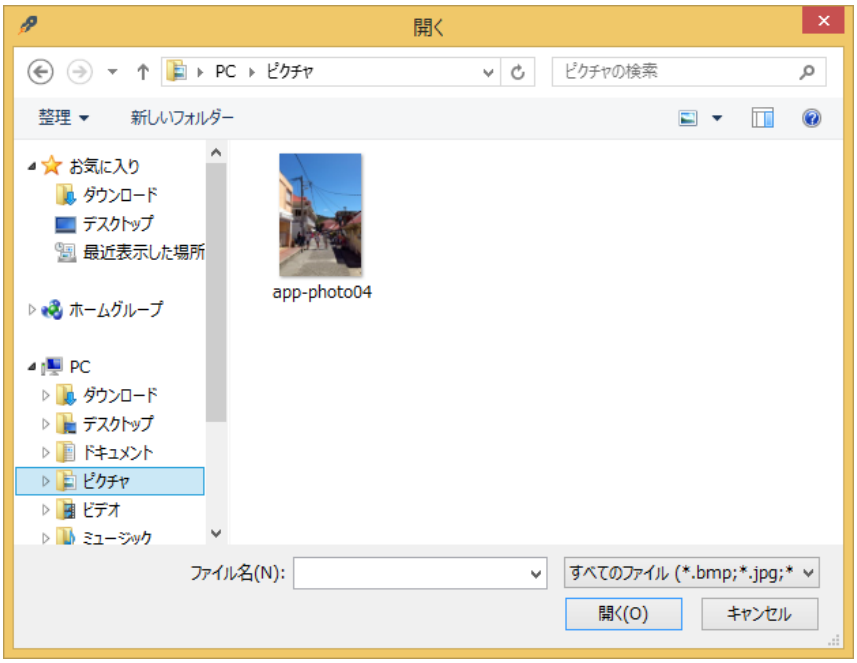
手順 8-4. 以下の手順を実施します。

① 「オブジェクトインスペクタ」の「Bitmap」をクリックします。
② 「値」を示す箇所の右側の「…」ボタンをクリックして「ビットマップ エディタ」を開きます。

手順 8-5. 表示されたビットマップ エディタで、［読み込み］ボタンをクリックします。

ここでは、アプリ起動時に表示される壁紙を読み込んでおきましょう。

手順 8-6. 以下の手順を実施します。

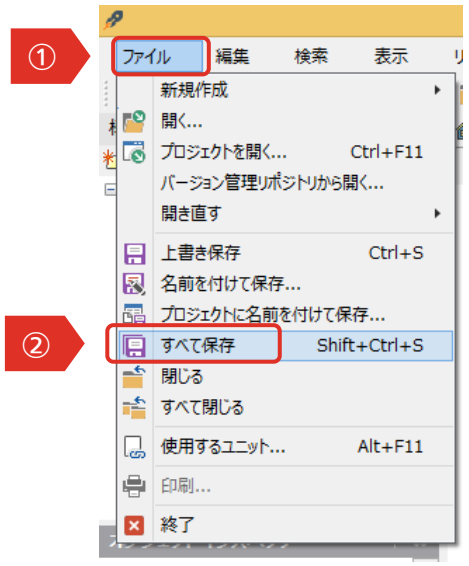


① 画像ファイルが保存されたフォルダを選択します。
② 画像ファイルを選択します。
③ 「開く」をクリックします。

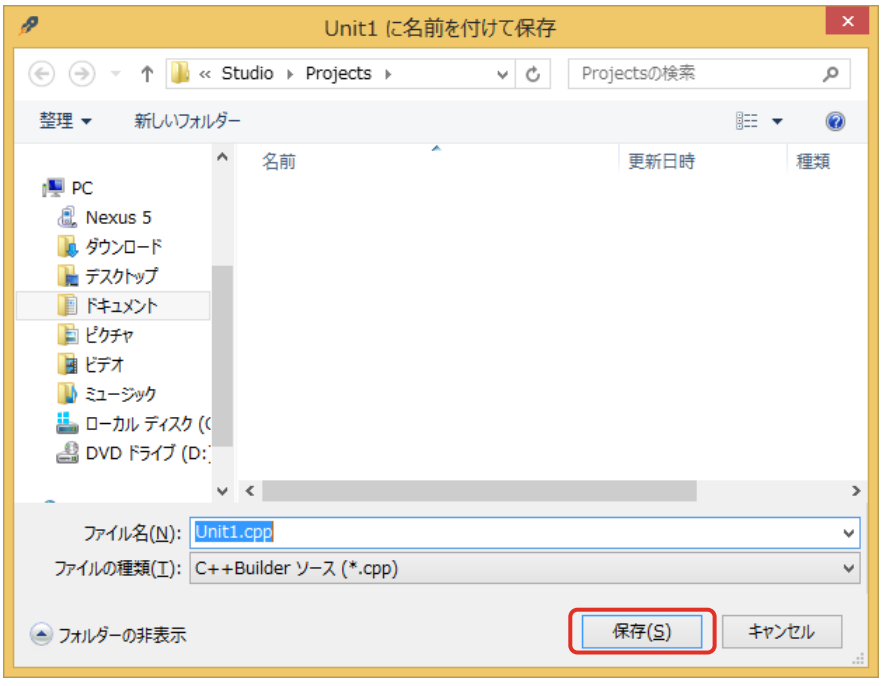
手順 8-7. 「OK」をクリックします。

以上で画面のレイアウトは終了です。一旦すべてのファイルを保存しておきましょう。

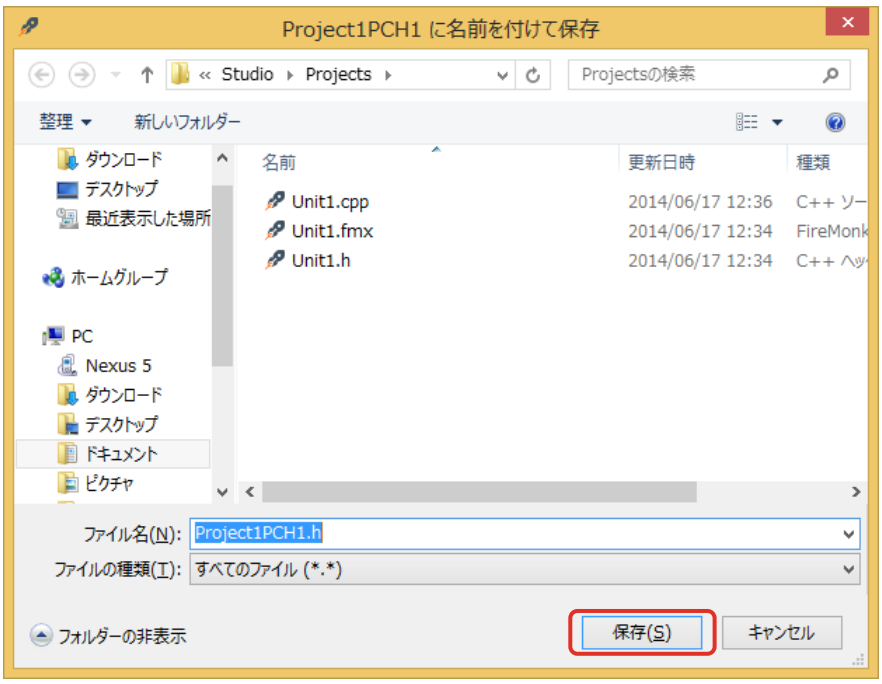
手順 8-8. メニューから「ファイル」→「すべて保存」をクリックします。



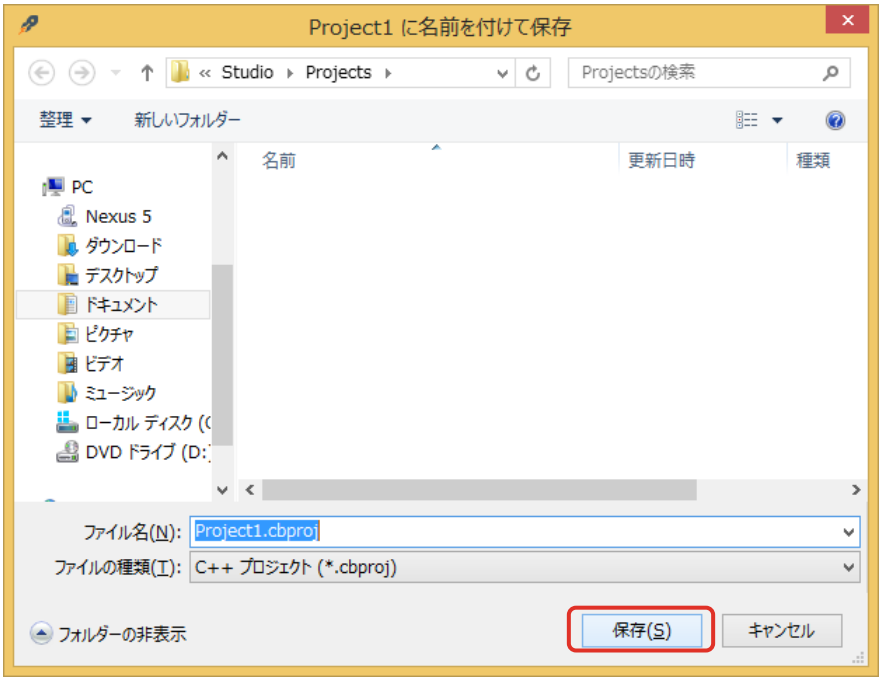
手順 8-9. フォームのソースファイル Unit1.cpp を保存します。［保存］をクリックします。



手順 8-10. ［保存］をクリックして、プロジェクトのヘッダファイル Project1PCH1.h を保存します。



手順 8-11. ［保存］をクリックして、プロジェクトを保存します。

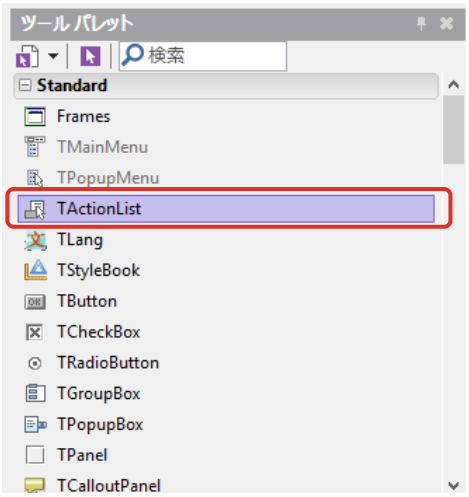


フェーズ 2: カメラ撮影のコードの記述

【手順 9】 「カメラアクション」の配置など

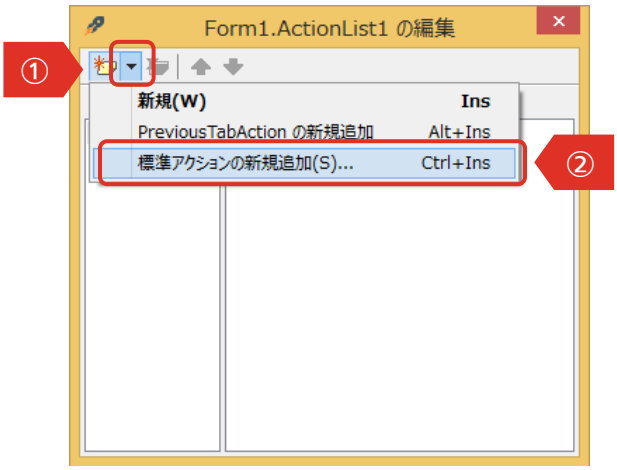
手順 9-1.「ツールパレット」から、Standard の中にある 「TActionList」をダブルクリックします。

カメラを起動するための機能は、アクションとしてあらかじめ定義されています。



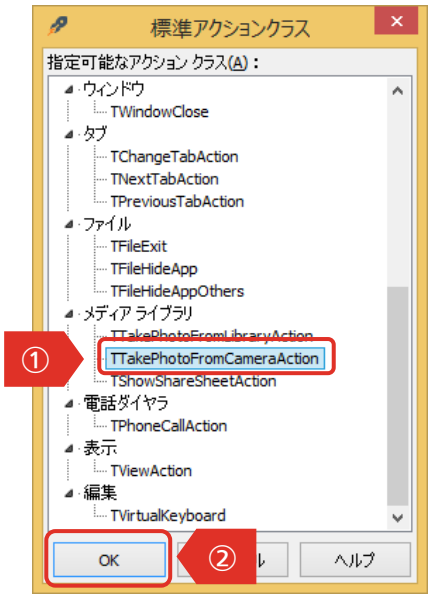
手順 9-2. デザイン上の「ActionList1」アイコンをダブルクリックして「編集」ダイアログを開きます。

手順 9-3. アクションを追加します。以下の手順を実施します。



① 「▼」をクリックしてメニューを開きます。
② 「標準アクションの新規追加」をクリックして「標準アクションクラス」ダイアログを開きます。

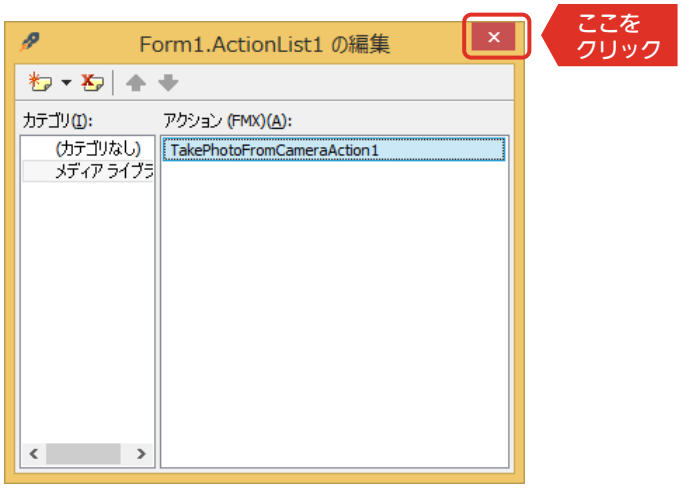
手順 9-4. カメラで撮影した写真を取得するアクションは、標準アクションとして定義されています。以下の手順を実施します。



TTakePhotoFromCameraAction は、カメラで撮影した写真を取得するアクションです。

① 「TTakePhotoFromCameraAction」をクリックします。
② 「OK」をクリックして「標準アクションクラス」ダイアログを閉じます。

手順 9-5. 「X」をクリックして「編集」ダイアログを閉じます。

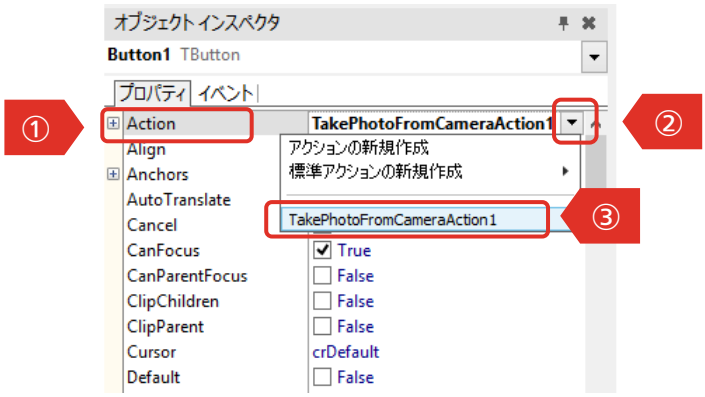


【手順 10】 イベントの関連付け

手順 10-1. カメラボタンをクリックして選択状態にします。

手順 10-2. 以下の手順を実施します。

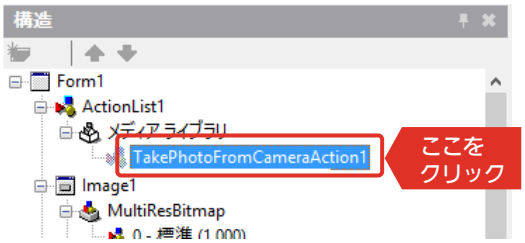
カメラで撮影した写真を取得したら、それを画面上に表示するコードを書きます。



① 「オブジェクトインスペクタ」の「Action」をクリックします。
② 「値」を示す箇所の右側の▼ボタンをクリックしてメニューを開きます。
③ 「TakePhotoFromCameraAction1」をクリックして選択します。

【手順 11】 コーディング

手順 11-1. 「構造」から「TakePhotoFromCameraAction1」をクリックして選択状態にします。

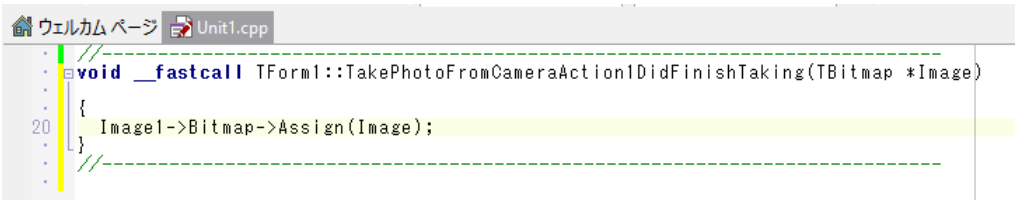


手順 11-2. 以下の手順を実施します。

① 「オブジェクトインスペクタ」の「イベント」タブをクリックします。
② 「OnDidFinishTaking」をクリックします。
③ 「値」を示す箇所の空白部分をダブルクリックしてエディタを開きます。

手順 11-3. 以下のコードを入力します。

ここで初めて C++ のコードを書きます。記述するコードは僅か一行です。



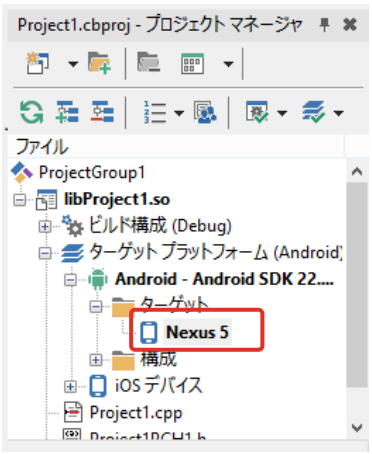

```
Image1->Bitmap->Assign(Image);
```

手順 11-4. メニューから「ファイル」→「すべて保存」をクリックします。

手順 11-5. Appmethod が動作している PC に Android 端末が USB ケーブルで接続されていることを確認してください。

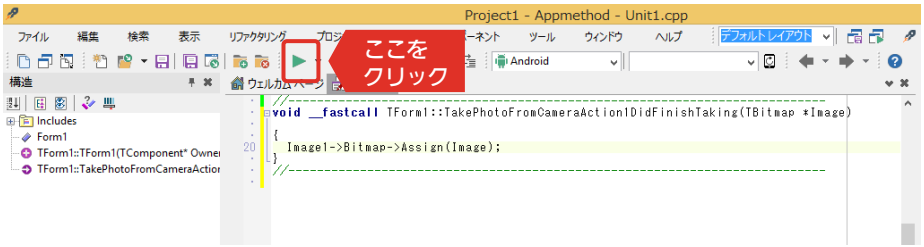
手順 11-6. 画面右のプロジェクトマネージャで、接続した Android 端末がターゲットに表示され、アクティブになっている（太字になっているとアクティブです）ことを確認してください。もし、アクティブになっていない場合には、ダブルクリックして、アクティブに変更してください。

この例では、Nexus 5 を接続しています。



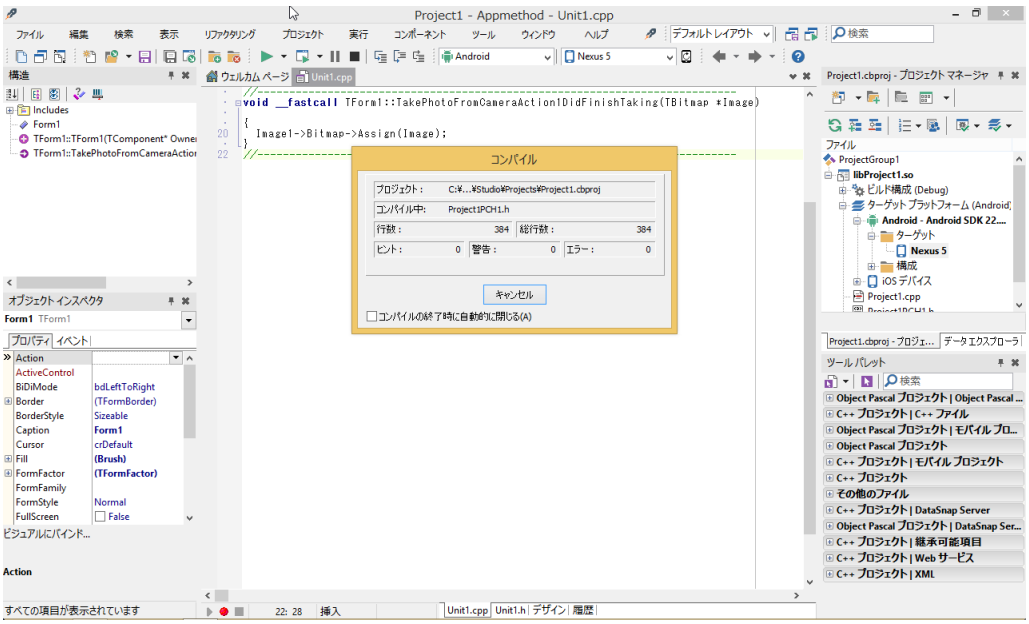
手順 11-7. 緑色の実行ボタンをクリックします。

実行ボタンをクリックするとコンパイルが実行され、Android 端末に作成したアプリが転送されます。



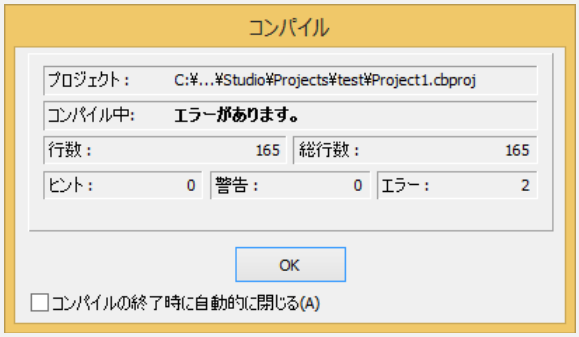
すると、次のような画面が表示され、コンパイルが実行されます。

ダイアログのタイトルは、コンパイル→配置→実行 の順に変わります。

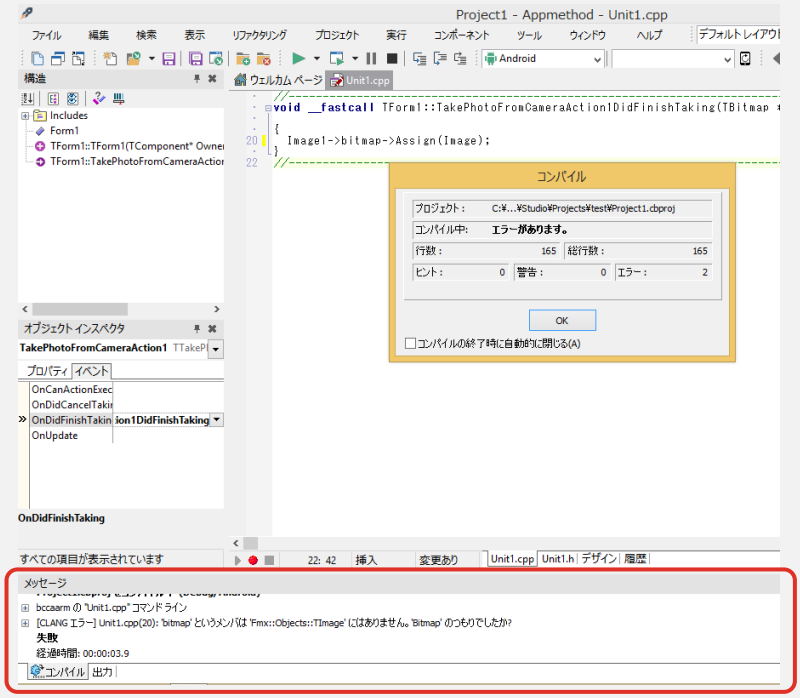


ヒント - コンパイルでエラーが発生したら

もし、入力したコーディングに問題がある場合は、以下のような画面が表示されます。この場合は、手順 11-3 からやり直してください。



「メッセージ」にエラー解決のためのヒントが表示される場合があります。



手順 11-8. 接続している Android デバイスにアプリが転送されます。アプリが起動したら、カメラ起動ボタンを押します。

手順 11-9. Android の以下のアイコンを押すと撮影します。

手順 11-10. 「レ」（チェックマーク）のアイコンを押すと撮影を終了します。

手順 11-11. 撮影した画面がカメラアプリに表示されます。

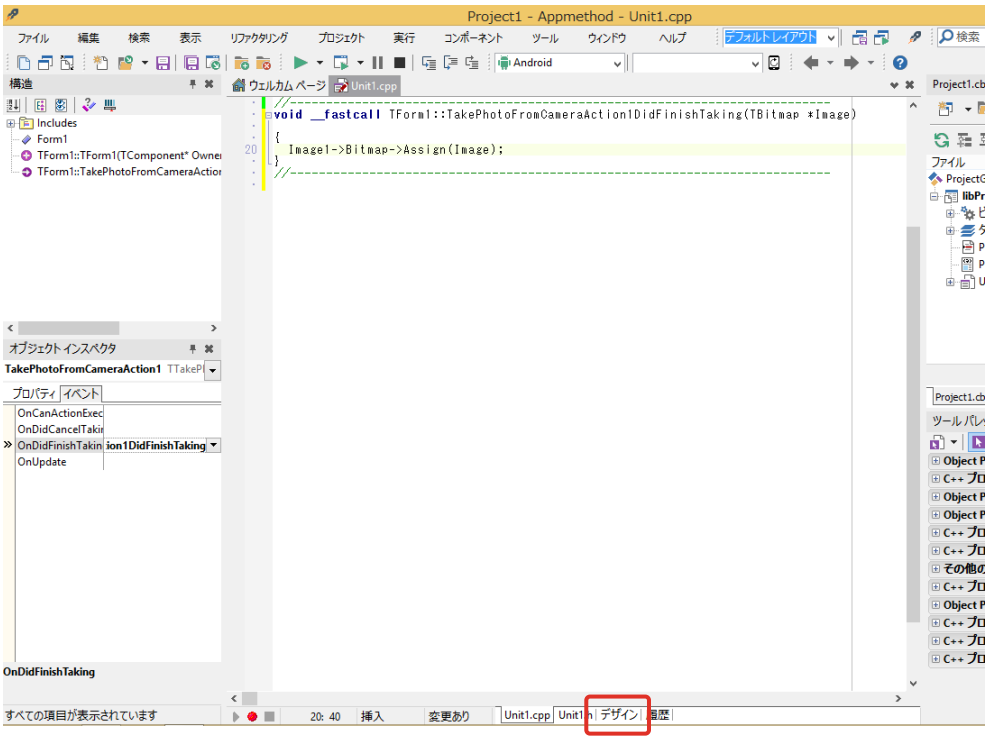
セピア加工のためのトラックバーをタップしてもまだ何も起こりません。この機能は、このあと作成します。

フェーズ 3: 効果を追加する

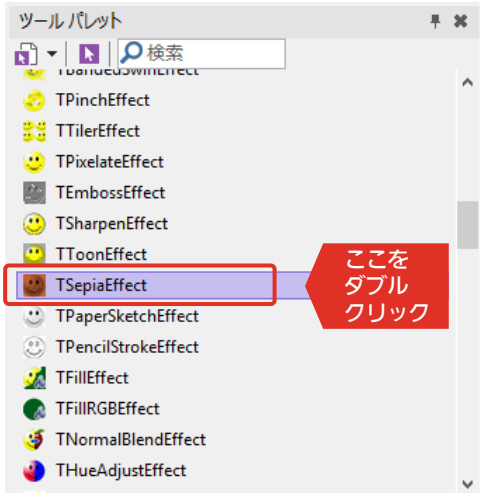
最後の手順では、トラックバーにセピア効果を調整する機能を追加します。

【手順 12】 「セピア加工」の配置など

手順 12-1. エディタの下部にある「デザイン」タブをクリックします。

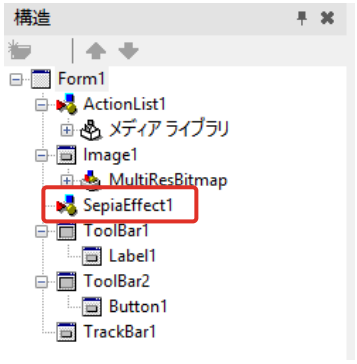


手順 12-2.「ツールパレット」から、Effects の中にある 「TSepiaEffect」をダブルクリックします。



セピア加工を行う機能もコンポーネントとして提供されています。

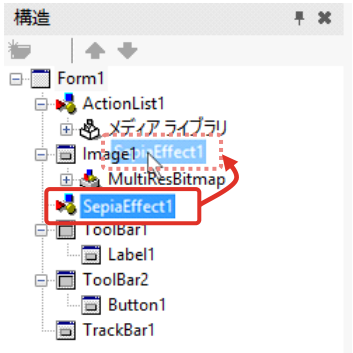
手順 12-3.「手順 12-2」の結果、以下のように「構造」に「SepiaEffect1」が追加されます。



TSepiaEffect のようなグラフィック効果の機能を提供するコンポーネントは、フォームデザイナには表示されません。構造ペインで、効果を加えたいコンポーネントの上にドロップすることで、効果が有効になります。

手順 12-4.「構造」の「SepiaEffect1」をドラッグして「Image1」にドロップします。

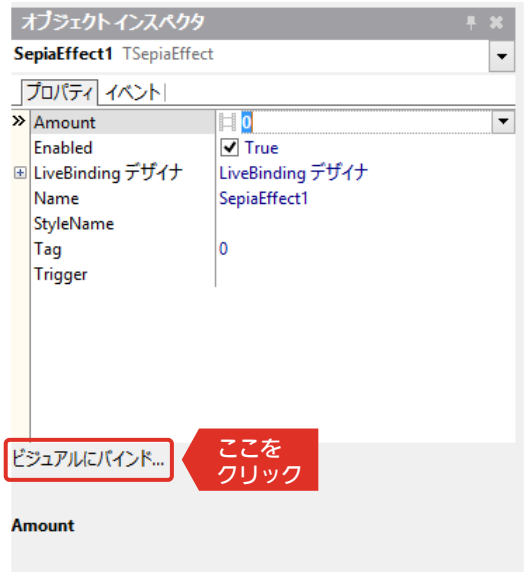
この操作で Image1 に対してセピア加工が行われます。



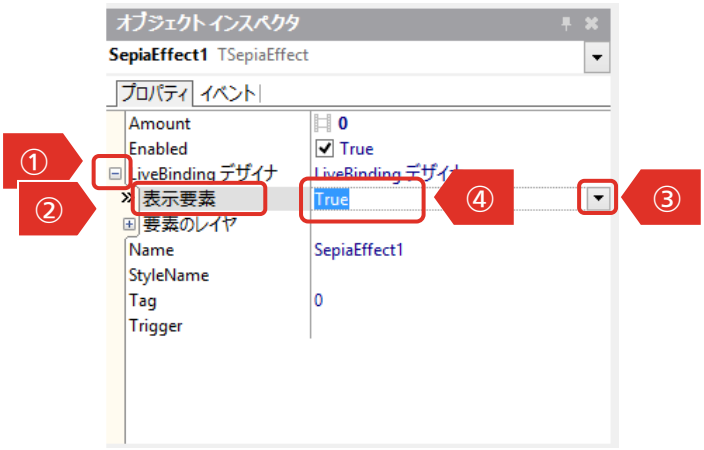
手順 12-5. 以下の手順を実施します。

① 「オブジェクトインスペクタ」の「プロパティ」をクリックします。
② 「オブジェクトインスペクタ」の「Amount」をクリックします。
③ 「値」に「0」を入力して「ENTER」を押します。

手順 12-6.「オブジェクトインスペクタ」の「ビジュアルにバインド...」をクリックします。



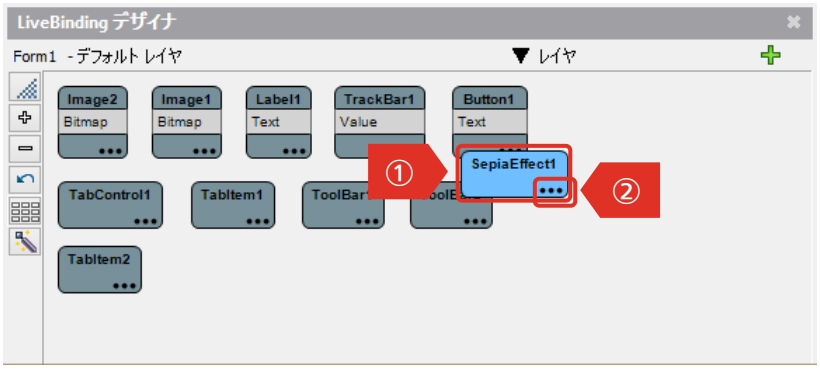
手順 12-7. 以下の手順を実施します。



① 「オブジェクトインスペクタ」の「LiveBinding デザイナ」をクリックします。
左側の[+]をクリックして展開状態にします。
② 「表示要素」をクリックします。

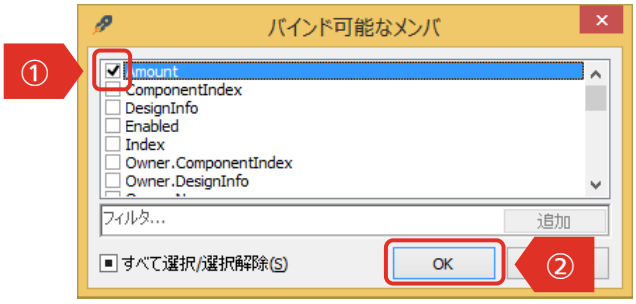
【ここからは、値が True ではない場合にのみ操作してください】
③ 「値」を示す箇所の右側の▼ボタンをクリックします。
④ 「True」をクリックして選択します。

手順 12-8. LiveBinding デザイナには、フォーム上に配置したコンポーネントが並んでいます。LiveBinding デザイナでは、ビジュアル操作でこれらを関連付けして、動作を定義することができま++す。以下の手順を実施します。



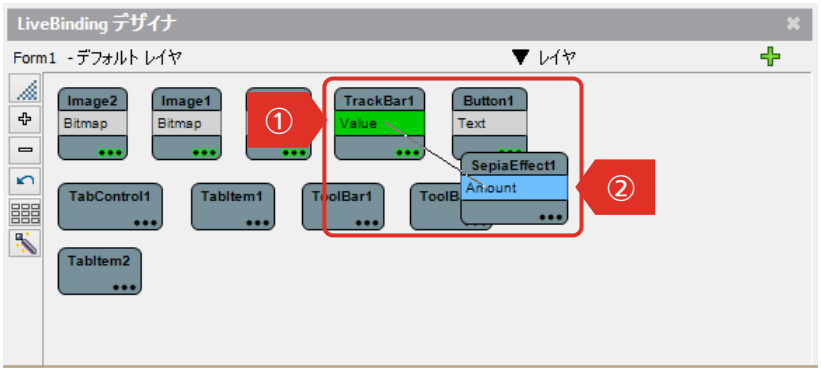
① 「LiveBinding デザイナ」の「SepiaEffect1」をクリックします。
② 「SepiaEffect1」の「…」をクリックして「バインド可能なメンバ」ダイアログを開きます。

手順 12-9. 以下の手順を実施します。



① 「バインド可能なメンバ」の「Amount」の ☐ をクリックしてチェックします。
② 「OK」をクリックして「バインド可能なメンバ」ダイアログを閉じます。

手順 12-10. 以下の手順を実施します。

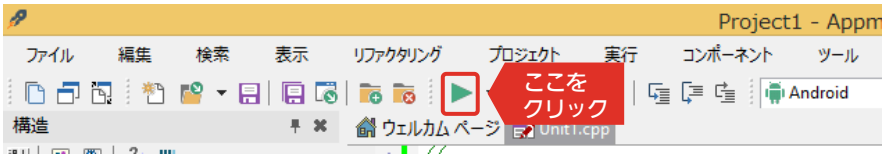


① 「TrackBar1」の「Value」をドラッグします。
② 「SepiaEffect1」の「Amount」にドロップします。

以上の操作で、TrackBar1 の Value プロパティと SepiaEffect1 の Amount プロパティが線で結ばれます。これにより TrackBar1 の値が、セピア効果を行う SepiaEffect1 の Amount（セピアにする割合）に連動するようになります。

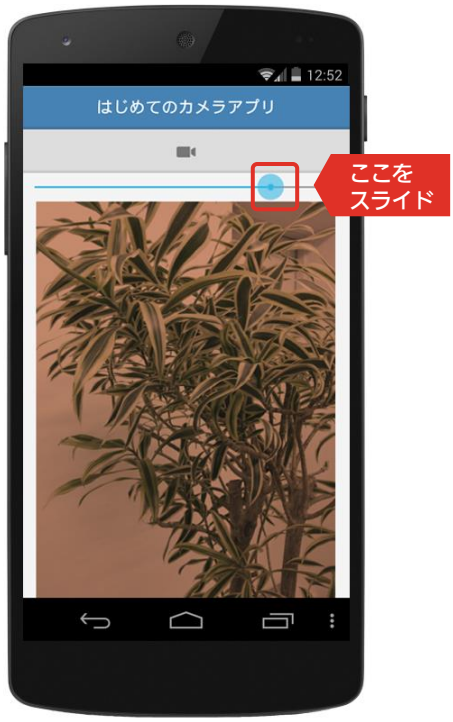
手順 12-11. メニューから「ファイル」→「すべて保存」をクリックします。

手順 12-12. 緑色の実行ボタンをクリックします。



機能を追加したアプリがコンパイルされ、Android 端末に転送されます。

手順 12-13. Android アプリでカメラ撮影し、セピア加工トラックバーをスライドします。



以上でカメラアプリは完成です。このほかにも、さまざまな機能を追加してみましょう。

エンバカデロ・テクノロジーズは、1993 年にデータベースツールベンダーとして設立され、2008 年にボーランドの開発ツール部門「CodeGear」との合併によって、アプリケーション開発者とデータベース技術者が多様な環境でソフトウェアアプリケーションを設計、構築、実行するためのツールを提供する最大規模の独立系ツールベンダーとなりました。米国企業の総収入ランキング「フォーチュン 100」のうち 90 以上の企業と、世界で 300 万以上のコミュニティが、エンバカデロの Delphi®、C++Builder®といった CodeGear™製品や ER/Studio®、DBArtisan®、RapidSQL®をはじめとする DatabaseGear™製品を採用し、生産性の向上と革新的なソフトウェア開発を実現しています。エンバカデロ・テクノロジーズは、サンフランシスコに本社を置き、世界各国に支社を展開しています。詳細は、www.embarcadero.com/jp をご覧ください。