報道各位

2017年8月25日
株式会社インプレスR&D
http://nextpublishing.jp/

# 「技術書典シリーズ」第二弾！

## 『Essential Xamarin ネイティブからクロスプラットフォームまで モバイル.NETの世界』発行

## 最先端技術者による全方位の解説書！

インプレスグループで電子出版事業を手がける株式会社インプレスR&Dは、『Essential Xamarin ネイティブからクロスプラットフォームまで モバイル.NET の世界』(著者:榎本 温,平野 翼,中村 充志,奥山 裕紳,末広 尚義,中澤 慧)を発行いたしました。

**『Essential Xamarin ネイティブからクロスプラットフォームまで モバイル.NETの世界』**
http://nextpublishing.jp/isbn/9784844397915


著者:榎本 温,平野 翼,中村 充志,奥山 裕紳,末広 尚義,中澤 慧
小売希望価格:電子書籍版 2300円(税別)／印刷書籍版 2700円(税別)
電子書籍版フォーマット:EPUB3／Kindle Format8
印刷書籍版仕様:B5判／モノクロ／本文240ページ
ISBN:978-4- 8443-9791-5
発行:インプレスR&D

**＜＜発行主旨・内容紹介＞＞**

【技術書典シリーズ第二弾！Xamarinの最先端技術者による解説書！】

Windows やMacOSなど様々な環境で動作し、モバイルアプリの開発をAndroidやiOSなどのクロスプラットフォームで可能とする「Xamarin」の情報を、様々な角度から紹介した解説書です。

関連リポジトリ集やSDK解説まで詳しく掲載した、Xamarinコミュニティの最先端技術者による全方位の解説書となっています。

(本書は、次世代出版メソッド「NextPublishing」を使用し、出版されています。)

## Xamarin とは何かについて、ネイティブ開発と比較しながら紹介

Xamarin.AndroidやXamarin.iOSは、.NETフレームワークのOSS実装である「Mono」の上で動作します。コード共有は、このMonoが複数のプラットフォームに対応していること、またPCL（Portable Class Library）という同一のバイナリを複数プラットフォームが使用できる仕組みの恩恵です。これについては「1.5 クロスプラットフォームアプリ開発とコードの共有」で詳しく説明します。

Xamarin Platformの上位層には「Xamarin.Forms」という、ユーザーインターフェースも共通化できるフレームワークがあります。これについては「1.6 Xamarin.Formsとは」で説明します。

### Xamarin Platformという呼称

Xamarin.Android、Xamarin.iOSといった「プラットフォームのAPIの薄いラッパー」を提供するツール群は、過去に「Xamarin Native」と公式サイトで記載されていました。しかし、「Native」という単語の曖昧さが嫌われてか、現在の公式サイトにはこの表記は見られません。国内外のXamarinの発表資料では現在も「Xamarin Native」という呼称が使われていたり、最近では「Xamarin Traditional」を使っている資料もあるようです。できればどれかに統一して欲しいと思いますが、個人的には「Xamarin Platform」に一票！です。

### Androidアプリ開発者から見たXamarin.Android

Mono上で動作するXamarin.Androidの動作の仕組みについては、本書の代表でもある @atsushieno 氏による「インサイドXamarin[2]」にて詳しく解説されています。Xamarinの中の人による詳解なので、これ以上の説明はまったく不要ですが、筆者の解釈で描いた図をひとつだけ紹介します。

図 1.1: Android ネイティブとXamarin.Androidの構成

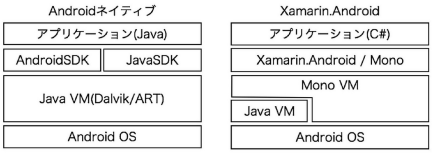



図1.1は、ネイティブとXamarin.Androidでの構成と実行形式の違いを示したものです。Androidアプリは JavaVM（Dalvik や ART）上で動作しますが、Xamarin.Androidでは、JavaVMに加え、C#コード（からコンパイルされた中間言語=MSIL）を解釈、実行するMonoVMが並行して動作します。MonoVMを含むランタイムは、（リリースビルドでは）アプリに同梱され、動作します。

ここでは、Androidアプリ開発者がXamarin.Androidでアプリを開発してみたならば、という視点で、Xamarin.Androidについて説明してみます。



Getting Started

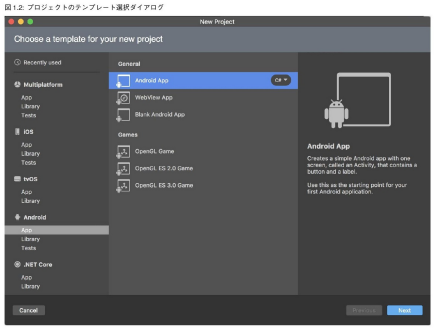
Xamarin.Androidでアプリを開発するには、統合開発ツールである Visual Studio for Mac または Xamarin Studio（macOSの場合）、Visual Studio 2015（Windowsの場合）を使用します。Androidアプリ開発者のみなさんはきっとMac使いなので、ここではVisual Studio for Macを使って説明しましょう。

### Visual Studio for MacとXamarin Studio

2016年11月にPreview版が公開されたVisual Studio for Macは、2017年6月に正式リリースされました[3]。その実体はXamarin Studioに.NET Standard対応やAzure連携の機能を追加して外観を変えたものです。しばらくの間は両者が並行して存在する模様ですが、Visual Studio for Macが正式リリースとなったことで、Xamarin Studioは今後収束していくと予想されます。これからXamarinアプリ開発を始めるならば、Visual Studio for Macを使用しましょう。

Visual Studio for Mac（以下Visual Studioと略）を起動して、「New Project...」を押すと、プロジェクトのテンプレート選択ダイアログが表示されます。左サイドバーの「Multiplatform」には、AndroidとiOSアプリを同時開発するためのプロジェクトテンプレートが並んでいますが、もっともシンプルにXamarinでAndroidアプリを作成する場合は、「Android-App」から「Android App」を選択します。

図 1.2: プロジェクトのテンプレート選択ダイアログ






## 豊富なスクリーンショットで、わかりやすく解説

図 2.4: Configure your Mac app

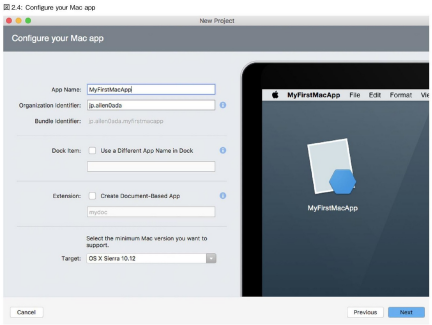
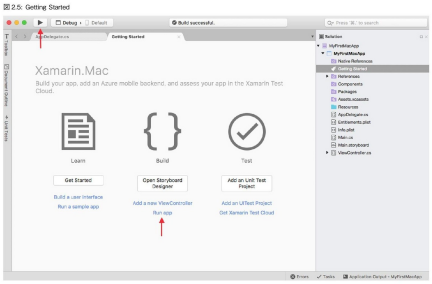



図 2.5: Getting Started




本質ではないため、ここではウィンドウサイズを固定することとし、お好きなサイズと位置で配置してください。Auto Layout についてはiOSのそれとほぼ変わらないため、Web上にある既存の資



図 2.6: Xcode のメインウィンドウで Main.storyboard を開く

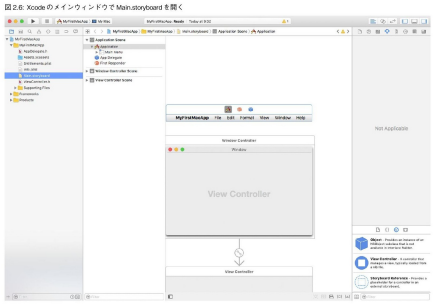
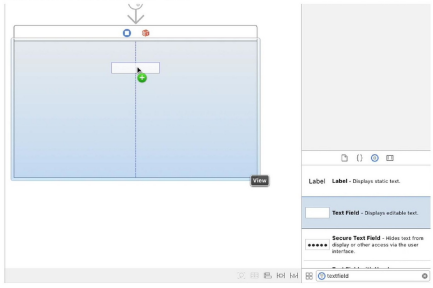


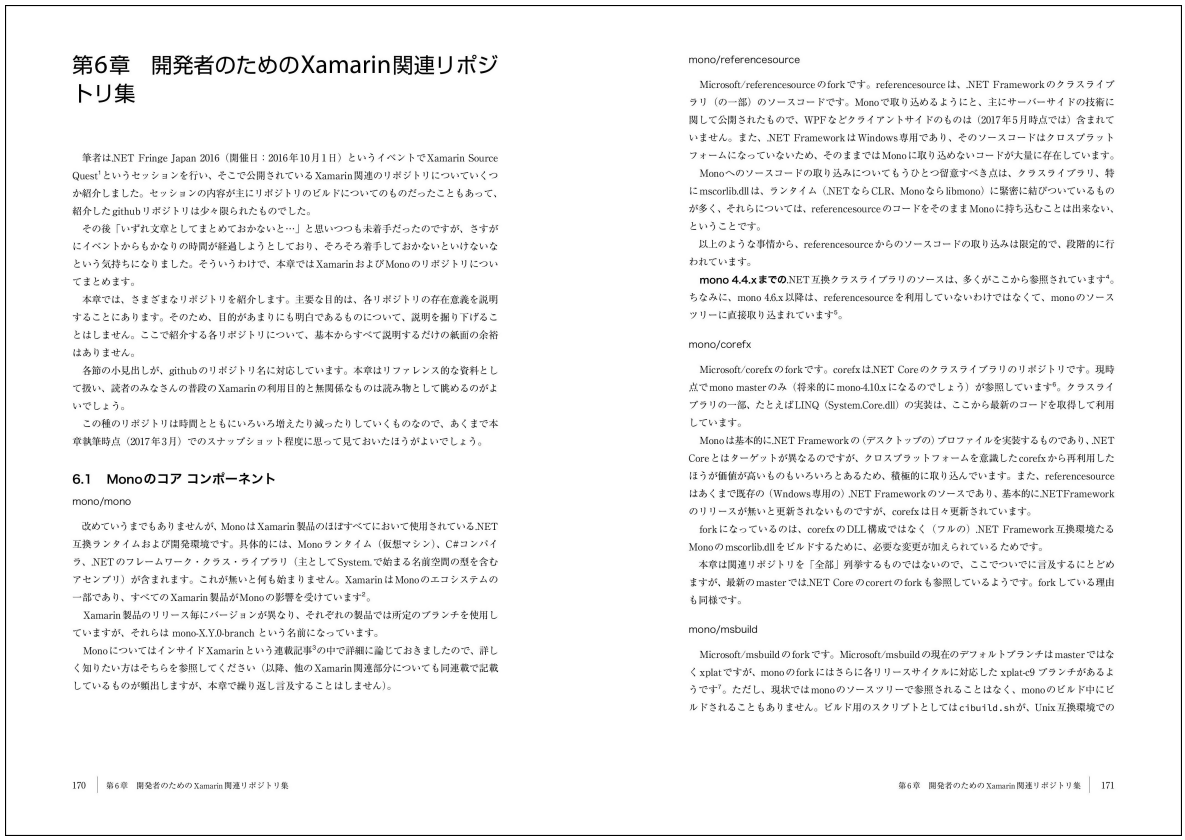

図 2.7: オブジェクトライブラリからドラッグ＆ドロップする




料を活用してください。図2.8に示した配置例では、Push Buttonの Title を変更し、Key Equivalent に Return キーを設定しています。

配置したコントロールに ViewControllerからアクセスするために Action,Outlet を作成していき

## 様々な Xamarin 関連リポジトリを詳細に解説



第6章　開発者のためのXamarin関連リポジトリ集

筆者は.NET Fringe Japan 2016（開催日：2016年10月1日）というイベントでXamarin Source Quest[1]というセッションを行い、そこで公開されているXamarin関連のリポジトリについていくつか紹介しました。セッションの内容が主にリポジトリのビルドについてのものだったこともあって、紹介したgithubリポジトリは少々限られたものでした。

その後「いずれ文章としてまとめておかないと…」と思いつつも未着手だったのですが、さすがにイベントからもかなりの時間が経過しようとしており、そろそろ着手しておかないといけないなという気持ちになりました。そういうわけで、本章ではXamarinおよびMonoのリポジトリについてまとめます。

本章では、さまざまなリポジトリを紹介します。主要な目的は、各リポジトリの存在意義を説明することにあります。そのため、目的があまりにも明白であるものについて、説明を掘り下げることはしません。ここで紹介する各リポジトリについて、基本からすべて説明するだけの紙面の余裕はありません。

各節の小見出しが、githubのリポジトリ名に対応しています。本章はリファレンス的な資料として扱い、読者のみなさんの普段のXamarinの利用目的と無関係なものは読み物として眺めるのがよいでしょう。

この種のリポジトリは時間とともにいろいろ増えたり減ったりしていくものなので、あくまで本章執筆時点（2017年3月）でのスナップショット程度に思って見ておいたほうがよいでしょう。

### 6.1　Monoのコア コンポーネント

mono/mono

改めていうまでもありませんが、MonoはXamarin製品のほぼすべてにおいて使用されている.NET互換ランタイムおよび開発環境です。具体的には、Monoランタイム（仮想マシン）、C#コンパイラ、.NETのフレームワーク・クラス・ライブラリ（主としてSystem.で始まる名前空間の型を含むアセンブリ）が含まれます。これが無いと何も始まりません。XamarinはMonoのエコシステムの一部であり、すべてのXamarin製品がMonoの影響を受けています[2]。

Xamarin製品のリリース毎にバージョンが異なり、それぞれの製品では所定のブランチを使用していますが、それらは mono-X.Y.0-branch という名前になっています。

MonoについてはインサイドXamarinという連載記事[3]の中で詳細に論じておきましたので、詳しく知りたい方はそちらを参照してください（以降、他のXamarin関連部分についても同連載で記載しているものが頻出しますが、本章で繰り返し言及することはしません）。

mono/referencesource

Microsoft/referencesourceのforkです。referencesourceは、.NET Frameworkのクラスライブラリ（の一部）のソースコードです。Monoで取り込めるようにと、主にサーバーサイドの技術に関して公開されたもので、WPFなどクライアントサイドのものは（2017年5月時点では）含まれていません。また、.NET FrameworkはWindows専用であり、そのソースコードはクロスプラットフォームになっていないため、そのままではMonoに取り込めないコードが大量に存在しています。

Monoへのソースコードの取り込みについてもうひとつ留意すべき点は、クラスライブラリ、特にmscorlib.dllは、ランタイム（.NETならCLR、Monoならlibmono）に緊密に結びついているものが多く、それらについては、referencesourceのコードをそのままMonoに持ち込むことは出来ない、ということです。

以上のような事情から、referencesourceからのソースコードの取り込みは限定的で、段階的に行われています。

**mono 4.4.xまでの**.NET互換クラスライブラリのソースは、多くがここから参照されています[4]。ちなみに、mono 4.6.x以降は、referencesourceを利用していないわけではなくて、monoのソースツリーに直接取り込まれています[5]。

mono/corefx

Microsoft/corefxのforkです。corefxは.NET Coreのクラスライブラリのリポジトリです。現時点でmono masterのみ（将来的にmono-4.10.xになるのでしょう）が参照しています[6]。クラスライブラリの一部、たとえばLINQ（System.Core.dll）の実装は、ここから最新のコードを取得して利用しています。

Monoは基本的に.NET Frameworkの（デスクトップの）プロファイルを実装するものであり、.NET Coreとはターゲットが異なるのですが、クロスプラットフォームを意識したcorefxから再利用したほうが価値が高いものもいろいろとあるため、積極的に取り込んでいます。また、referencesourceはあくまで既存の（Windows専用の）.NET Frameworkのソースであり、基本的に.NETFrameworkのリリースが無いと更新されないものですが、corefxは日々更新されています。

forkになっているのは、corefxのDLL構成ではなく（フルの）.NET Framework互換環境たるMonoのmscorlib.dllをビルドするために、必要な変更が加えられているためです。

本章は関連リポジトリを「全部」列挙するものではないので、ここでついでに言及するにとどめますが、最新のmasterではNET Coreのcorertのforkも参照しているようです。forkしている理由も同様です。

mono/msbuild

Microsoft/msbuildのforkです。Microsoft/msbuildの現在のデフォルトブランチはmasterではなくxplatですが、monoのforkにはさらに各リリースサイクルに対応したxplat-c9ブランチがあるようです[7]。ただし、現状ではmonoのソースツリーで参照されることはなく、monoのビルド中にビルドされることもありません。ビルド用のスクリプトとしてはcibuild.shが、Unix互換環境での

170　第6章　開発者のためのXamarin関連リポジトリ集

第6章　開発者のためのXamarin関連リポジトリ集　171

## ＜＜目次＞＞

8.14 ROM/静的確保 RAM の削減余地を探る
8.15 まとめ

＜＜著者紹介＞＞

**榎本 温（えのもと あつし）**

.NET はオープンソースで実現すべきと思って Mono の開発に参加していつの間にか 15 年。
エイプリルフールのジョークだった「マイクロソフトで働くことになりました」の状態に今でも馴染まない不良会社員です。卒業して音楽ソフトで遊んで暮らすのが目下の望み。

**平野 翼（ひらの つばさ）**

1988 年 9 月 16 日デビュー。株式会社ライフベア勤務。
Mac が幅をきかせる映像制作の現場において C#を武器に闘った結果、道を踏み外す。
おいしいものを食べたり飲んだりして，適当に写真を撮ったり映像を作ったりして生きていきたい。
Mac はそれなりに好き。

**中村 充志（なかむら あつし）**

1976 年 1 月 22 日生まれ。金融系エンタープライズ一直線の SIer 育ちのアーキテクト。Java から C#を渡り歩く。
趣味で作った Android アプリを iOS へ移植しようと Xamarin と出会う。でも結局アプリは作っていない。
現在の悩みは、Xamarin 案件がなかなか獲得できないこと。だれか Xamarin の金融案件ください。

**奥山 裕紳（おくやま ひろのぶ）**

ネットやコミュニティでは amay077(あめい) という名前で活動しています。
Windows 系の企業から転職して Android/iOS アプリエンジニアになったと思ったら、Xamarin によっていつの間にか .NET と C# に戻って来ていました。
愛知県在住のフルリモートワーカー。得意分野は位置情報、地理空間情報。

**末広 尚義（すえひろ ひさよし）**

1983 年 10 月 15 日生まれ。Linux で人生の半分近くを過ごしてきましたが、Xamarin を触るために久しぶりに windows マシーンを使い始めました。
環境になれなさすぎて早めに Xamarin Linux 版が出ないと死んでしまいます。
android や PHP な著書もありますが仕事は組み込み linux 方面で最近は仕事で Go とか elixir を書いてすごしています。

**中澤 慧（なかざわ けい）**

ここ数年はゲーム会社でゲームを作ったり作らなかったりする日々を過ごしています。

＜＜販売ストア＞＞

電子書籍:

Amazon Kindle ストア、楽天 kobo イーブックストア、Apple iBookstore、紀伊國屋書店 Kinoppy、Google Play Store、honto 電子書籍ストア、Sony Reader Store、BookLive!、BOOK☆WALKER

印刷書籍:

Amazon.co.jp、三省堂書店オンデマンド、honto ネットストア、楽天ブックス

※ 各ストアでの販売は準備が整いしだい開始されます。
※ 全国の一般書店からもご注文いただけます。

**【株式会社インプレス R&D】** http://nextpublishing.jp/

株式会社インプレス R&D（本社：東京都千代田区、代表取締役社長：井芹昌信）は、デジタルファーストの次世代型電子出版プラットフォーム「NextPublishing」を運営する企業です。また自らも、NextPublishing を使った「インターネット白書」の出版など IT 関連メディア事業を展開しています。

※NextPublishing は、インプレス R&D が開発した電子出版プラットフォーム（またはメソッド）の名称です。電子書籍と印刷書籍の同時制作、プリント・オンデマンド（POD）による品切れ解消などの伝統的出版の課題を解決しています。これにより、伝統的出版では経済的に困難な多品種少部数の出版を可能にし、優秀な個人や組織が持つ多様な知の流通を目指しています。

**【インプレスグループ】** http://www.impressholdings.com/

株式会社インプレスホールディングス(本社：東京都千代田区、代表取締役：唐島夏生、証券コード：東証1部9479)を持株会社とするメディアグループ。「IT」「音楽」「デザイン」「山岳・自然」「モバイルサービス」を主要テーマに専門性の高いコンテンツ＋サービスを提供するメディア事業を展開しています。2017 年 4 月 1 日に創設 25 周年を迎えました。

**【お問い合わせ先】**

株式会社インプレス R&D NextPublishing センター
〒101-0051　東京都千代田区神田神保町 1-105
TEL 03-6837-4820
電子メール: np-info@impress.co.jp