

# Visual Studio 2015 製品版 1st Step Guide



# 2015 年 7 月 17 日 (v. 1.0)

Microsoft

このドキュメントは現状版として提供されます。このドキュメントに記載されている情報や見解 (URL 等のインターネット Web サイトに関する情報を含む) は、将来予告なしに変更されることがあります。このドキュメントに記載された例は、説明のみを目的とした架空のものです。実在する事物とは一切関連ありません。内部的な参照目的に限り、このドキュメントを複製して使用することができます。

このドキュメントは、情報提供の目的にのみ提供されるものであり、マイクロソフトはこのドキュメントおよびその記載事項について、明示と黙示とを問わずなんら保証を行いません。

#### © 2015 Microsoft. All rights reserved.

Microsoft、System Center、Visual Studio、および Windows Live は Microsoft グループ各社の商標です。 その他、記載されている会社名および製品名は、各社の商標です。

最終更新日:2015 年 7 月 17 日



# 目次

Visual	Studio 2015 製品版 1st Step Guide	1
1. Vi	sual Studio 2015 のインストール	4
1.1.	インストーラーのダウンロードとインストール	4
1.2.	Visual Studio 2015 の初回設定	9
2. AS	SP.NET 5 アプリケーション開発	15
2.1.	クロスプラットフォーム対応の核、DNX	15
2.2.	ASP.NET 5 のプロジェクトを作成	
2.3.	プロジェクト構造の変化	
2.4.	ローカルでの実行と Azure Web App へのデプロイ	19
3. Aj	pache Cordova アプリケーション開発	25
3.1.	Apache Cordova プロジェクトの作成	25
3.2.	Apache Ripple を利用したデバッグ	
3.3.	Visual Studio Emulator for Android の追加	
4. 強	化された C++機能と Android / iOS への対応	
4.1.	C++11 への準拠強化と C++14/17 への対応	
4.2.	クロスプラットフォームへの対応	
5. Xa	amarin を利用した Android / iOS アプリケーション開発	35
6. Vi	sual Studio 2015 Tools for Unity の利用	
6.1.	 Unity へパッケージの追加	
6.2.	Visual Studio 2015 を利用したデバッグ	41
7. Gi	itHub との連携の強化	43
7.1.	<b>GitHub</b> との接続を行う	43
7.2.	GitHub 上のリポジトリをクローン	
7.3.	新しいリポジトリを GitHub 上へ作成	47



### 1. Visual Studio 2015 のインストール

7/20 に RTM がリリースされた Visual Studio 2015 の評価版をダウンロードして、インストールと初期設定 を行っていきましょう。

#### 1.1. インストーラーのダウンロードとインストール

Visual Studio 2015 評価版のダウンロードは以下のリンク先から無料で行えます。複数マシンへの展開に 便利な ISO も公開されていますが、ひとまずは単体のインストーラーをダウンロードしてインストールを 行います。

https://www.visualstudio.com/downloads/download-visual-studio-vs

ダウンロードしたインストーラーを起動するとセットアップが始まります。 インストールする場所はデフォルトのままで問題ありません。インストールの種類は [カスタム] を選択 します。



🔀 Visual Studio	- ×
Enterprise 2015	
インストールの場所を選択してください	
C:\Program Files (x86)\Microsoft Visual Studio 14.0	0
セットアップには、すべてのドライブで最大 9 GB が必要です	
インストールの種類を選択してください <b>Typical</b> Includes C#/VB Web and Desktop features <b>カスタム(U)</b> Allows you to customize features for your inst	allation
機能はセットアップ後にコントロール パネルの [プログラムと根 加および削除できます。	機能]を介していっでも追
[インストール] ボタンをクリックすることにより、 <u>ライセンス条項</u> <u>する声明</u> に同意したものとみなされます。	および <u>プライバシーに関</u>
キャンセル(C)	😌 インストール(N)

途中でインストールする機能を選択する場面がありますので、必要な機能だけをインストールすることも 出来ますが、今回は「すべてを選択」にチェックを入れて Visual Studio 2015 で使える機能を全てインスト ールすることにします。

Visual Studio 2015 の特徴となるクロスプラットフォームなアプリケーション開発と GitHub との統合を利用するために、最低でも「クロスプラットフォーム モバイル開発」と「Git for Windows」「Visual Studio 向け GitHub 拡張」にはチェックを入れておきましょう。



© 2015 Microsoft Corporation. All rights reserved.

K Visual Studio	_ X	
Enterprise 2015 機能の選択		
<ul> <li>プログラミング言語</li> <li>Visual C++</li> <li>Visual F#</li> <li>Python Tools for Visual Studio</li> <li>Windows 即発とWab 即発</li> <li>クロスプラットフォーム モバイル開発</li> <li>子ョョツール</li> <li>Git for Windows [3rd Party]</li> <li>Visual Studio 向け GitHub 拡張 [3rd Party]</li> <li>Visual Studio 1021(1±2)=1/2</li> </ul>		
■ すべて選択(S) セットアップには、すべてのドライブで最大 23 GB が必要です。	既定値にリセット(R)	
<b>戻る(B)</b>	次へ(N)	

選択が終われば「次へ」ボタンをクリックし、インストールを行う機能の確認を行います。この時に実際 にインストールが行われるソフトウェア名を確認できるようになっています。

例を挙げればクロスプラットフォームなモバイルアプリケーションの開発に必要な Android SDK や JDK が インストールされることが確認できます。



🔀 Visual Studio	- ×
Enterprise 2015	
選択されている機能	
MICROSOFT SOFTWARE Microsoft Visual Studio Emulator for Android Visual C++ Android 開発 Visual C++ iOS 開発 Microsoft Web Developer Tools ツールと Windows SDK Windows 8.1 SDK と Universal CRT SDK HTML/JavaScript (Apache Cordova) サードパーティー契のソフトウェア PowerShell Tools for Visual Studio Java SE 開発キット (7.0.550.13) Android SDK	うイセンス条項 ライセンス条項 ライセンス条項 ライセンス条項 ライセンス条項 ライセンス条項 ライセンス条項 ライセンス条項 ライセンス条項
セットアップには、すべてのドライブで最大 23 GB が必 [インストール]をクリックすると、インストール対象とし ウェアおよびコンポーネントのライセンス条項に同意し れらのライセンス条項をお読みになり、承諾する責任 ユーザーがインストールすることを選択したいかなるり も権利を付与しません。これは、該当するライセンス ティによって提供されます。	マロン マクマロ な要です。 て選択したアプリケーション ソフト たことになります。ユーザーはこ 丘があります。Microsoft は、 サード パーティ製のソフトウェアに 、条項に示されているサード パー

右下にある「インストール」ボタンをクリックすると、選択した機能のインストールが始まります。まず 必要なパッケージのダウンロードを行ってからインストールを行うため、環境によっては時間がかかる場 合があります。



Visual Studio	-	×
Enterprise 2015		
取得しています:		
Microsoft .NET Framework 4.6		
適用しています:		
Microsoft Visual Studio Enterprise 2015		
キャンセル(C	)	

Visual Studio 2015 のインストールが完了すると、それを伝える表示に切り替わります。インストールの完 了には再起動が必要となるので、ダイアログの下部に表示されている「今すぐ再起動」ボタンをクリック します。





Windows の再起動後に、Visual Studio 2015 が利用可能な状態になるため、このタイミングで再起動を行っておきます。

#### 1.2. Visual Studio 2015 の初回設定

Visual Studio 2015 のインストールが完了しましたので、Visual Studio を起動して初回の設定を行っていきます。

Visual Studio を初めて起動すると、Visual Studio へのサインインを要求されるので、Microsoft アカウント を持っている場合には「サインイン」ボタンをクリックします。サインイン自体は後からでも行えるため、

© 2015 Microsoft Corporation. All rights reserved.



今は不要という場合にはボタンの下にある「後で行う。」リンクをクリックします。



Visual Studio にサインインすると、現在 Visual Studio 2013 を利用している場合には各種設定が自動的に 反映されるようになっています。Microsoft Azure のサブスクリプションをお持ちの場合には、それも反映 されます。

サインインを選んだ場合には、サインインを行うメールアドレスを入力する画面が表示されます。登録済 み Microsoft アカウントのメールアドレスを入力し、「続行」ボタンをクリックします。



© 2015 Microsoft Corporation. All rights reserved.

	x
Visual Studio	
サインインに使用するアカウントの電子メール アドレスを 力してください。	入
someone@example.com	
続行	
職場または学校アカウントは、このシンボルが表示されているすべて ページで使用できます。 © Microsoft 2015 利用規約 プライバシー	τσ

「続行」ボタンをクリックすると、Microsoft アカウントのパスワードを入力する画面になるため、正しい パスワードを入力し「サインイン」ボタンをクリックします。



	x
Microsoft アカウント	
サインイン	
Microsoft アカウント	
パスワード	
サインイン	
Microsoft アカウントをお持ちでない場合 新規登録	

サインインに成功すると、Visual Studio を初めて使うための設定が行われるので、少し時間がかかること があります。この間、画面にはログインしたユーザーアカウントの情報が表示されます。





初めて使うための設定が完了すると、Visual Studio のスタートページが表示されます。





これで Visual Studio 2015 を使う準備がすべて整いました。それでは実際に Visual Studio 2015 で追加された新しい機能について、順番に見ていきたいと思います。



#### 2. ASP.NET 5 アプリケーション開発

Visual Studio 2015 で一番大きく変化したのが ASP.NET 5 アプリケーション開発です。

ASP.NET 5 はこれまでの ASP.NET 4.5 から大きく開発スタイルを変え、肥大化したフレームワークの整理 や.NET Compiler Platform (Roslyn)を使った実行時コンパイルと、NuGet を使った依存関係の解決、さらに はクラウド環境への最適化を図るために.NET Core Runtime (CoreCLR)と呼ばれる、実行環境を丸ごと sideby-side で配置が行えるポータブルなランタイムが追加されました。

大きく開発スタイルが変わったため、これまでの ASP.NET MVC 5 プロジェクトからの互換性が無くなって いますが、ASP.NET MVC としての考え方が変わったわけではないので、これまでの知識が全て無駄になる というわけでは決してありません。

#### 2.1. クロスプラットフォーム対応の核、DNX

ASP.NET 5 は新しく開発された.NET Execution Environment(DNX)と呼ばれる、クロスプラットフォーム上 での実行に対応したランタイムで動作します。DNX には.NET Framework 4.6 上で動作する DNX 4.6 と.NET Core Runtime(CoreCLR)上で動作する DNX Core 5 が存在しており、ランタイム自体が Roslyn や NuGet を 統合しているため、追加ソフトウェアの必要なく C#のコンパイルや依存関係の解決が行えるようになり ました。それらはアプリケーションのポータビリティ改善に役立っています。

クロスプラットフォームでの実行には CoreCLR を用いる DNX Core での実行が必須となります。現時点では x64 環境の Linux と OS X 上での実行がサポートされています。

#### 2.2. ASP.NET 5 のプロジェクトを作成

ASP.NET 5 ではランタイムやフレームワークなどの足回りが大きく変化し、実行前に手動でコンパイルす る必要すらなくなりました。極端な話でテキストエディタとコマンドプロンプトだけあれば開発は行えま すが、Visual Studio 2015 を使うことでこれまでと同じように開発が行えるように環境が整備されていま す。

これまでと同じように Visual Studio 2015 の「ファイル」メニューから「新しいプロジェクトの作成」を選 択するだけで、ASP.NET 5 のプロジェクトを作成できます。



			新しいプロジェクト		? ×
▷ 最近使用したファイル		.NET F	ramework 4.6 👻 並べ替え: 既定		インストール済み テンプレート の検索 (Ctrl・ 🔎 🗸
▲ インストール済み		<b>C</b> <sup>*</sup> 1	ASPNET Web アプルケーション	Visual C#	種類: Visual C#
⊿ テンプレート					ASP.NET アプリケーションを作成するためのプロ
✓ Visual C#			クラス ライブラリ (パッケージ)	Visual C#	ジェクト テンノレートです。ASP.NET Web フォーム、MVC、または Web API のアプリケー
Web		<b>C</b> :\	コンソール アプリケーション (パッケージ)	Visual C#	ションを作成し、ASP.NET のさまざまな機能を 追加できます。
Cloud					Physical Application Insights
Extensibili	ty				<ul> <li>Application Insights をプロジェクトに追 加(A)</li> </ul>
▷ iOS LightSwite	ch				アプリケーションの動作を理解して最適化するのに役立ちます。
Mobile Ap	ps				詳細を表示
Silverlight	repoint				
WCF					(Mic… *
Workflow					資格情報を再入力してください。
テスト Android APK	projects (*.apk)				
▶ 他の言語	projecto ( reprij				
▶ その他のプロジェ	クトの種類				
モデリング プロジ. サンプル	፤ሳኮ				
シンジル					
* 1/25/12			オンラインでテンプレートを検索するには、ここをクリックします	<u>.</u>	
名前(N):	WebApplication	1			
場所(L):	c:¥users¥shibay	yan¥docı	uments¥visual studio 2015¥Projects	•	参照(B)
ソリューション名(M): WebApplication:		1			✓ ソリューションのディレクトリを作成(D)
					□ ソース管理に追加(U)
					OK キャンセル

新しいプロジェクトダイアログでは ASP.NET Web アプリケーションを選択すれば、その後に ASP.NET プロジェクトの種類を選択できます。ダイアログでは ASP.NET 4.6 と ASP.NET 5 の 2 つが選択できるようになっています。



新し	ASP.NET	プロジェクト - Web	Application1 ? ×
テンプレートを選択する(S):			
ASP.NET 4.6 テンプレート			プレビュー - ASP.NET 5 アプリケーションを作成するためのプロ ジェクト テンプレートです。このテンプレートは、ASP.NET MVC
Empty Web Forms MVC	() Web API	Single Page	を使用し、Web アブリテーションと RES ITUI HTTP サービスの 構築に使用できます。 詳細
Azure API App Azure Mobile		Application	
ASP.NET 5 プレビュー テンプレート	1		
Empty Web API Web Application			認証の変更(A) 認証・個別のコーザー アカウント
フォルダーおよびコア参照を追加する:			Microsoft Azure
Web Forms MVC Web API			<ul> <li>         ・ クラウドにホストする(H)         Web App ×     </li> </ul>
ユニットテストを追加する(U)			
テスト プロジェクト名(T): WebApplication1.Te	sts		
			ОК <b>+</b> 7721

ここで ASP.NET 5 の Web Application を選択して「OK」ボタンをクリックすると、プロジェクトの作成が 始まります。そしてプロジェクトの作成が完了すると、以下のような画面が表示されます。

WebApplication1 - Microsoft Vi ァイル(F) 編集(E) 表示(V) プロジェクト(P) マ ○   認 - 😩 🎴 🖉   ⑦ - ♡ -	sual Studio (管理者) ) ビルド(B) デバッグ(D) チーム(M) Debug ・ Any CPU ・ ト	ツール(T) アーキテクチャ(C) テスト(! IIS Express •   声 <sub>マ</sub>	5) 分析(N) ウィンドウ(W) ヘルプ(H)		👂 🗕 🗖 🗙 shibayan - 🎇
Welcome to ASP.NET 5 + × URL(U): C:¥Users¥shibayan¥Documen	ts¥Visual Studio 2015¥Projects¥Web/	Application1¥src¥WebApplication1¥Pro	ject_Readme.html	<ul> <li>ソリューション エクスプローラー</li> <li>○ ○ △ ○ ○ ○ □ □ □ □ □</li> </ul>	+ + × ■ 記
We've made some big updates i ASP.NET 5 has been rearchitecte Your new project automatically (for client-side build and autom We hope you enjoy the new cap The ASP.NET Team	to ASP.NET in this release, so it's <b>important</b> this ed to make it <b>lean</b> and <b>composable</b> takes advantage of modern client-se ation tasks). abilities in ASP.NET 5 and Visual St ASP. NET 5 project <b>Learn</b>	<ul> <li>&gt; yu1-3a ≥ u5/21a-5a → 0k8% (Ctrl+ [] yu1-3a &gt; WebApplication1 (1 ; ] global.json         [] global.json         [] global.json         [] global.json         [] global.json         [] global.json         [] group for the set of the</li></ul>	:) ~ ~ 70529F)		
				プロパティ	u - ∓ # ×
This application	NEW CONCEPTS	CUSTOMIZE APP	DEPLOY	WebApplication1 全般	-
consists of:	The 'wwwroot' explained	Add Controllers and Views	Run your app locally	Trイル名 WebApp	lication1.xproi
	d Bower for g client side Gulp	Add Data using EntityFramework	Run your app on ASP.NET Core 5	フォルダー パス c:¥users ルート名前空間 WebApp	c:¥users¥shibayan¥document: WebApplication1
Sample pages using ASP.NET MVC 6		Add Authentication using Identity	Run commands in your project.json		
Gulp and Bower for managing client-side		Add real time support using SignalR	Publish to Microsoft Azure Web Sites		
resources	Develop on different platforms	Add Class library	Publish to the file system		
Theming using Bootstrap	· · · · · · · · · · · · · · · · · · ·	Add Web APIs with MVC 6			
		Add client packages using Bower		ルート名前空間	
ケージの復元が完了しました。	dh = ala				





これまでの ASP.NET 4.5 のプロジェクト作成ダイアログに統合される形で ASP.NET 5 のプロジェクトを作 成できることからも、開発環境としてはこれまでと同等に扱うという意思が感じられます。事実、ASP.NET 5 アプリケーションの開発は Visual Studio を使うと、これまでと同じく全て GUI で行えるようになってい ます。

ちなみに ASP.NET 5 からは MVC / Web API / Web Pages が ASP.NET MVC 6 として統合されたため、ASP.NET 5 は実質的に ASP.NET MVC 6 を指すことになります。

#### 2.3. プロジェクト構造の変化

ASP.NET 5 アプリケーションは、これまでの ASP.NET MVC 5 などとは異なりプロジェクト構造が大きく変 化しています。分かりやすい点としては src ディレクトリの中に実際にソースコードが格納されており、 www.root というこれまで ASP.NET アプリケーションでは意識しなかったディレクトリが追加されていま す。

そして CTP の頃からの変更として、ASP.NET 5 プロジェクトでは標準で bower と Gulp がインストールさ れるようになり、プロジェクトツリーにこれまでの「リファレンス」とは別に「依存関係」として Bower と NPM が管理しているパッケージが表示されるようになりました。





Bower は bowerjson、NPM は packagejson に定義されたパッケージが表示されます。インストールに関 しても、それぞれのファイルにパッケージ名を追加するだけで、自動的にインストールが行われるように なっています。

NuGet ももちろん健在で、パッケージマネージャーコンソールから使う以外にも、project.json ファイルに パッケージ ID 追加するだけでインストールが自動的に行われるようになりました。

#### 2.4. ローカルでの実行と Azure Web App へのデプロイ

最近のトレンドを取り入れてモダンな Web アプリケーションのフレームワークとして生まれ変わった ASP.NET 5 ですが、Visual Studio 2015 を使っている限りでは、これまでと同じように Intellisense が十分に 効いた環境で開発を行えますし、デバッグなどの機能に大きな変化はありません。開発者に選択肢とそれ

© 2015 Microsoft Corporation. All rights reserved.



を選ぶ自由がこれまで以上に与えられているという点で、大きな進化を果たしたバージョンと言えるでしょう。

OWIN ライクな仕組みを取り入れることで、これまでの HTTP パイプラインとは大きく変化していますが、 ローカル環境では IIS Express を使った実行がこれまで通り、簡単に行えるようになっています。



「IIS Express」と表示されているボタンをクリックすると、その場でアプリケーションが起動して、ブラウ ザでアプリケーションのページが表示されます。



ローカル環境で ASP.NET 5 アプリケーションの実行まで行いましたが、Visual Studio 2015 を使うことで、 これまでの ASP.NET 4.5 と開発手順が変わっていないことをわかっていただけたと思います。

ASP.NET 5 アプリケーションの実行環境としては、既に Azure Web Apps で対応が行われているため、Visual Studio の発行機能を使うことでシームレスに開発から公開まで行えるようになっています。クラウドに最適化されたバージョンである ASP.NET 5 はサーバーの依存関係に影響を受けることなく実行を行えますし、

© 2015 Microsoft Corporation. All rights reserved.



既に Linux 上の Docker を使って Visual Studio からのデプロイやリモートデバッグなどが行えることが発表されています。

実際に Azure Web Apps へのデプロイは非常に簡単で、これまで通りメニューから「公開」を選択し、まずは新しい Web App を作成します。

Micro	soft Azure で Web App を作成	×
Microsoft Azure で Microsoft アカ	Web App を作成 フントーー	
Web App 名(N):	aspnet5-demo-app	9
App Service プラン(A):	Default2 (Japan East) ×	
リソース グループ(G):	Default-Web-JapanEast ~	
リージョン(R):	Japan East 🗸	
データベース サーバー(D):	データベースなし 、	
支払制限を解除したか、従量 ると料金に影響する可能性がる 詳細	制を使用している場合、追加のリソースをプロビジョニングす あります。 法律条項	
	作成キャンセル	

ここでは"aspnet5-demo-app"としていますが、同名の Web Apps は作成できないため、任意の名前を入力 してください。リージョンなどを選択後、右下の「作成」ボタンをクリックすると、Web App の作成が開 始されます。

少し時間がかかりますが、作成が完了すると以下のような画面に切り替わります。



	Web を発行	? ×
💮 Web を発行		
ว้อวราง	aspnet5-demo-app	
接続設定	サーバー(E): aspnet5-demo-app.scm.azurewebsites.net:443	
プレビュー	サイト名(T): aspnet5-demo-app ユーザー名(N): \$aspnet5-demo-app	
	パスワード(W): ●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●	••••
	宛先 URL(L): http://aspnet5-demo-app.azurewebsites.net	
	接続の検証(V)	
	< 戻る(R) 次へ(X) > 発行(P) 閉し	;a(0)

この画面では作成した Web App の接続情報が表示されますが、基本的にはデフォルトの設定のままで問 題ありません。右下にある「次へ」ボタンをクリックし、さらに次の設定を行っていきましょう。

次は構成と使用する DNX のバージョンを選択する画面が表示されます。これも基本的にはデフォルトの 設定のままで問題ありませんが、CoreCLR を利用したい場合にはターゲット DNX バージョンを変更する と、そのバージョンがデプロイされるようになります。



	Web を発行	? ×
Web を発行		
プロファイル	aspnet5-demo-app	
接続 設定 プレビュー	構成(C): Release ターゲット DNX パージョン(X): 既定値 (dnx-clr-win-x86.1.0.0-beta5) ◆ ファイル発行オプション	* *
	<b>発行スクリプト</b> ✓ PowerShell スクリプトを使用して発行(S)	
	< 戻る(R) 次へ(X) > 発行(P)	閉じる(0)

右下の「発行」ボタンをクリックすると、作成した Web Apps ヘアプリケーションが自動的にデプロイさ れます。デプロイが完了するとブラウザが起動して Web App が表示されるので、簡単に動作確認が行え るようになっています。





ローカル環境で実行したものと全く同じページを表示することが出来ました。

デプロイされた ASP.NET 5 アプリケーションは.NET Compiler Platform (Roslyn)を利用したランタイムで初回起動時にコンパイルが行われますが、Visual Studio からのデプロイ時に予め NuGet パッケージとしてコンパイルしておくことで、起動時のパフォーマンスを改善することも可能です。



# 3. Apache Cordova アプリケーション開発

Visual Studio 2013 向けには CTP として公開されていた Apache Cordova 開発を行うための Multi-Device Hybrid Apps for Visual Studio が 2015 Preview で統合されたため、追加でパッケージをインストールすることなく JavaScript と TypeScript の両方で Cordova アプリケーションの開発が行えるようになりました。

#### 3.1. Apache Cordova プロジェクトの作成

Visual Studio の新規プロジェクト作成ダイアログにある JavaScript と TypeScript のカテゴリ内に、Apache Cordova の空のアプリケーションテンプレートが追加されました。

			新しいプロジェク	١٢	? ×
▶ 最近使用したファイル		.NET F	Framework 4.6 👻 並べ替え: 既定	- # E	インストール済み テンプレート の検索 (Ctrl. 🔎 🗸
▲ インストール済み		<b>I</b> MI	空のアプリ (Apache Cordova)	TypeScript	種類: TypeScript
<ul> <li>▲ テンプレート</li> <li>▶ Visual C#</li> <li>Android APK p</li> <li>▲ 他の言語</li> <li>Python</li> <li>Visual F#</li> <li>ゲーム</li> <li>▶ Visual Basi</li> <li>▶ Visual Basi</li> <li>▶ Visual Basi</li> <li>▶ Visual C+</li> <li>SOL Servet</li> <li>▲ JavaScript</li> <li>▲ JavaScript</li> <li>▲ PowerShel</li> <li>▲ TypeScript</li> <li>▲ Typ</li></ul>	ic + Cordova Apps Cordova Apps Cordova Apps - 	÷.		, i broat ibr	Apache Cordova™ を使用する空のプロジェ クトです。複数のセルイル / ラットフォーム (Android, iOS, Windows, Windows Phone 8) を対象とするアプリをビルドするのに役 立ちます。
			オンラインでテンプレートを検索するには、	<u>ここをクリックします。</u>	
名前(N):	BlankCordovaAp	op1			
場所(L):	c:¥users¥shibay	an¥doc	uments¥visual studio 2015¥Projects	•	参照(B)
ソリューション名(M):	BlankCordovaAp	op1			<ul> <li>✓ ソリューションのディレクトリを作成(D)</li> <li>○ ソース管理に追加(U)</li> <li>○ K</li> </ul>

それぞれのプロジェクトを作成すると、Windows / Windows Phone 8 / Android / iOS 向けに開発するため に最低限必要なリソースが準備されたテンプレートが生成されます。

スタートページとして、どのような手順でコードを書いていけばいいのかといった説明が表示されるので、 まずはこの手順に従って開発を進めていくことをお勧めします。



Weil Studie - rdv ad 機構理解的 a * * * * * * * * * * * * * * * * * *	■ BlankCordovaApp1 - Microsoft Visual Studio (管理者) アイル(F) 編集(E) 表示(V) プロジェクト(P) ビルF(B) デル(P)(D) チーム(M) ソー 0 - ○ お・雪 当 評 ラ - で - Debug - Android - ▶ Ripple	-ル(T) アーキテクチャ(C) テスト(S) 分析(N) ウインドウ(W e - Nexus (Galaxy) - ) 声 <sub>ま</sub> 〇 〇 〇 〇 〇 〇 〇 〇 〇 〇 〇 〇	) へルプ(H)	▼ じ クイック起動 (Ctrl+Q)	P 🗕 🗖 🗙 shibayan - 🅅
デバイスの機能を追加する (プラグイン Apache Cordova ではプラグインを使用して、単純な Web アプリでは使用できないネイティブデバイス機能 ヘのアクセスを提供します。これには ファイル システ ム、カメラ、加速度計、デバイス コンタクトなどの機 能が含まれます。 デバイスの機能を追加するには、プロシェクトのルー トで config.xml を聞きます。これにより、起動するア プリ機成マネージャーに、Microsoft によって使活済み のコア ブラグインを提供する (Plugini l ダブがありま	Visual Studio rdova の使用見構成です。× URL(U): C:¥Users¥shibayan¥Documents¥Visual Studio 2015¥Projects¥BlankCord VisualStudio.com 資料 ビデオ サンブル FAQ 作業の開始 Apache Cordova を使用した Android、iOS、Windows および Wind Tデバイスの 機能を追加 する 2 コードを使 う	ovaApp1¥BlankCordovaApp1¥Project_Readme.html 既知の問題 サポート Nows Phone 用アブリのビルドの準備が整いました ろ アプリのビ ルドと実行	。 デバッグ ■診断	yy」->>>> オカズガローラー	- キ× )
エラー一覧 出力 Azure App Service のアクライとティ	デバイスの機能を追加する (プラグイ ン) Apache Cordova ではブラグインを使用して、単純な Web アブリでは使用できないネイティブデバイス機能 へのアクセスを提供します。これには ファイル システ ム カメラ、加速度計、デバイス コンタクトなどの機 能が含まれます。 デバイスの機能を追加するには、プロジェクトのルー 下で config.xmlを構きます。これにより、起動するア ブリ構成マネージヤーに、Microsoft によって検証済み のコア プラグインを提供する IPluainl タブがありま	Core         Custom         Installed           Select enotined for retrieve the plugin         Image: Selection for the plugin you want to install           Inter a Gir repository location for the plugin you want to install         Inter/Ugithub.com/MSOpenTech/condexv plugin tet         Image: Selection for the plugin you want to install	Test plugin Add Version 0.0. Plugin ID: con Description This Cor Pluform: All ¥ ¥	イリリニーション エクスプローラー プロパライ           BlankCordovaApp1 全般           副 20 / チー 出力パス           製力パス	70-5-

TypeScript の場合は自動的にコンパイルを行う設定も行われたテンプレートとなるので、開発者は複雑な ことを考えることなく、ビルドを行うだけでアプリケーションを作成できます。

#### 3.2. Apache Ripple を利用したデバッグ

Visual Studio のインストール時に「クロスプラットフォーム モバイル開発」を選択しておくと、Apache Ripple が同時にインストールされるため、後で Google Chrome をインストールすれば Visual Studio から Ripple を使ったデバッグ実行を標準で行えるようになっています。





そして、新しく DOM Explorer が Visual Studio に追加されたことで、Chrome のデベロッパーツールを使うことなく DOM ツリーや CSS の状態を一目で確認出来るようになりました。JavaScript コンソールも持っているので、ログや JavaScript コードの実行も簡単に行えます。



デバッグに必要な情報を全て Visual Studio だけで収集することが出来るので、ブラウザ内蔵のツールを利



用するよりも効率的な開発が行えるようになりました。

#### 3.3. Visual Studio Emulator for Android の追加

Ripple を使うことで手軽に Android 上での動作を想定した開発とデバッグが行えるようになりましたが、 OS のネイティブとして提供されている機能を使う場合などで、実際に Android OS 上での確認も必要とな る場合があります。

そして今回のリリースでは Hyper-V を利用した Visual Studio Emulator for Android が追加されたため、 Hyper-V が利用可能な環境では Windows Phone 8.1 アプリ開発のように、より実機に近いデバッグが行え るようになりました。



Ripple や各種エミュレーターを含め、以下のような実行環境が Apache Cordova アプリケーション用に予め用意されています。

- 実機デバイス
- Google Android エミュレーター
- Ripple Nexus 7 (タブレット)
- Ripple Nexus (Galaxy)
- Ripple Nexus S
- VS Emulator 5" KitKat (4.4) XXHDPI Phone
- VS Emulator 7" KitKat (4.4) XHDPI Tablet

Android 以外のプラットフォームでデバッグを行う際には、デバッガー起動前にツールバーからターゲットプラットフォームを切り替えます。

Visual Studio Emulator for Android はデフォルトで5インチのスマートフォン、7インチのタブレットイメ

© 2015 Microsoft Corporation. All rights reserved.



ージがインストールされていますが、「ツール」メニューから「Visual Studio Emulator for Android」を選ぶ と、イメージの管理画面が表示されます。

Visual Stu	udio Emulator for Android	<mark>:</mark>	-	x
Device P	rofiles	Sig	gn in	•
All API Leve	els 💌			
	7" KitKat (4.4) XHDPI Tablet API Level 19 (KitKat, 4.4) 456 MB	Similar to Asus Google Nexus 7	لاله 🔻	Î
	5" KitKat (4.4) XXHDPI Phone API Level 19 (KitKat, 4.4)	Similar to Samsung Galazy S5, Sony Xperia Z	▲ લુ⊳ >	l
	10.1" Lollipop (5.1.1) XHDPI Tablet API Level 22 (Lollipop, 5.1.1)	Similar to Samsung Google Nexus 10	< I <del>C</del>	I
	7" Lollipop (5.1.1) XHDPI Tablet API Level 22 (Lollipop, 5.1.1)	Similar to Asus Google Nexus 7	< I€-	I
	5.5" Lollipop (5.1.1) XXHDPI Phone API Level 22 (Lollipop, 5.1.1)	Similar to LG G4	< I€	I
	5.1" Lollipop (5.1.1) XXHDPI Phone API Level 22 (Lollipop, 5.1.1)	Similar to Similar to Samsung Galaxy S6, Samsung Galaxy S6 Edge	<b>∢</b> I≮	
	5" Lollipop (5.1.1) XXHDPI Phone	Similar to Sony Xperia Z, LG Nexus 5	¥	-
			Close	

API Level ごとに表示を切り替えて、標準でインストールされているもの以外のイメージもインストール可能です。これによって、Windows上での Android 開発が一段と容易に行えるようになったといえるでしょう。



# 4. 強化された C++機能と Android / iOS への対応

Visual Studio 2015 での C++開発環境のアップデートは過去最大と言っても過言ではありません。特に言 語面での機能拡充はもちろんのこと、標準ライブラリやデバッガー、コンパイル速度の改善や IDE の品質 向上など、各方面に渡り強化が行われています。

#### 4.1. C++11 への準拠強化と C++14/17 への対応

これまでの Visual Studio 2013 などでは C++11 への対応は他のコンパイラーに比べて遅れ気味でしたが、 Visual Studio 2015 では C++11 への対応をほぼ終え、更に C++14 や C++17 への部分的な対応も進められ ています。

以下に Visual Studio 2015 で新しく対応した C++の機能一覧をまとめます。

- C++11
  - Rvalue references
  - ➢ ref-qualifiers
  - > Attributes
  - > constexpr
  - > Alignment
  - Inheriting constructors
  - char16\_t and char32\_t
  - Unicode string literals
  - Universal character names in literals
  - User-defined literals
  - Defaulted and deleted functions
  - Extended sizeof
  - Inline namespaces
  - Unrestricted unions
  - > noexcept
  - Data-dependency ordering: attributes
  - quick\_exit and at\_quick\_exit
  - Thread-local storage
  - Magic statics
- · C++14
  - Binary literals
  - > auto and decltype(auto) return types
  - ➢ init-captures
  - Generic lambdas
  - [[deprecated]] attributes
  - Sized deallocation
  - Digit separators
- C++17
  - New rules for auto with braced-init-lists
  - > typename in template template-parameters

© 2015 Microsoft Corporation. All rights reserved.

- > Attributes for namespaces and enumerators
- ➢ u8 character literals

既に CTP や Preview としてリリースされていたバージョンと比較しても、着実に C++仕様への対応が進め られています。

例を挙げると CTP では未対応または部分的な対応だった char16\_t / char32\_t と Unicode 文字列リテラル(u / U / u8)への対応が追加されています。そして Preview の時点では部分的に対応にとどまっていた constexpr と Generic lambda も、RTM では全ての対応が完了しています。

Visual Studio 2015 では C++を使った Windows / Android / iOS アプリケーション開発に対応しているため、C++コンパイラーの互換性向上は非常に重要となります。

### 4.2. クロスプラットフォームへの対応

CTP の時のようなコンパイラーのアップデートだけではなく、Visual Studio 側のサポートも拡充されています。

「新しいプロジェクト」ダイアログでは、クロスプラットフォーム用のプロジェクトが追加されたことで、 Android と iOS をターゲットとした共有ライブラリを、Pimpl イディオムを使って比較的簡単に作れるようになりました。

			新しいプロジェクト	<b>、</b>	? ×
▶ 最近使用したファイル		.NET F	ramework 4.6 👻 並べ替え: 既定	- # E	インストール済み テンプレート の検索 (Ctrl: 🔎 🗸
▲ インストール済み		E**	Native-Activity アプリケーション (Android)	Visual C++	種類: Visual C++
▲ テンプレート ▷ Visual C# Android APK r	projects (* ank)	bad  ++    ₀□	OpenGLES アプリケーション (Android, iOS)	Visual C++	Native-Activity アプリケーションを作成するプ ロジェクトです。
▲ 他の言語 Python	nojecto ( htph)		Shared Library (Android, iOS)	Visual C++	
、 Visual F# ゲーム			スタティック ライブラリ (Android)	Visual C++	
▷ Visual Bas ✓ Visual C+-	ic +		スタティック ライブラリ (iOS)	Visual C++	
▷ Windov MFC	vs	, , ,	ダイナミック共有ライブラリ (Android)	Visual C++	
クロス プ Extend	ラットフォーム しまた。		基本アプリケーション (Android)	Visual C++	
SQL Serve ▷ JavaScript	er :		空のパッケージ プロジェクト (iOS)	Visual C++	
PowerShel ▶ TypeScript	ll t		メイクファイル プロジェクト (Android)	Visual C++	
▶ その他のプロジェ/ モデリング プロジ: サンプル	クトの種類 ェクト				
▶ オンライン			オンラインでテンプレートを検索するには、ここ	<u>こをクリックします。</u>	
名前(N):	Android1				
場所(L):	c:¥users¥shiba	yan¥doc	uments¥visual studio 2015¥Projects	•	参照(B)
ソリューション名(M):	Android1				☑ ソリューションのディレクトリを作成(D) □ ソース管理に追加(U)
					OK         キャンセル



また、クロスプラットフォーム用のプロジェクトでは Windows Universal App 向けに追加された Shared Project が使われているため、プラットフォーム固有のコードとそれ以外の共有が可能なコードといった形 で開発することが出来ます。



ライブラリ以外では Android NDK を使ったネイティブアプリケーションのプロジェクトが追加されたため、C++を利用した Android アプリケーションの開発は Visual Studio のみで行えるようになりました。





© 2015 Microsoft Corporation. All rights reserved.

これらの新しく追加された Android 向けの C++テンプレートでは、コンパイラーとして Clang 3.4 と GCC4.8 が使えるようになっているため、C++11/14 などの機能を簡単に使えるようになっています。

(C): アクティブ(Debug)	✓ プラットフォーム(P): アクティブ(	x86) マ 構成マネージャー(C
構成プロパティ	▲ プロジェクトの既定値	
全般	構成の種類	ダイナミック ライブラリ (.so)
デバッグ	ターゲット API レベル	KitKat 4.4 - 4.4.4、 (android-19)
VC++ ディレクトリ	STL の使用	GNU STL スタティック ライブラリ (gnustl_static)
▲ C/C++	Thumb モード	無効
全般	▲ 全般	
最適化	出力ディレクトリ	<pre>\$(SolutionDir)\$(Platform)¥\$(Configuration)¥</pre>
フリプロセッサ	中間ディレクトリ	\$(Platform)¥\$(Configuration)¥
コート生成	ターゲット名	lib\$(RootNamespace)
言語	ターゲットの拡張子	.SO
ノリコンハイルバロのハッター	クリーン時に削除する拡張子	*.d;*.tlog;\$(ExtensionsToDeleteOnClean)
計加設と	ビルド ログ ファイル	\$(IntDir)\$(MSBuildProjectName).log
コマンド ライン	プラットフォーム ツールセット	Clang 3.4
N 11271-		Clang 3.4
		GCC 4 8

ターゲットとする言語バージョンに関しても、プロジェクトのプロパティから指定可能となり、最新の言 語機能を使った開発がすぐに行えるようになっています。

Android1.NativeActivityプロパティページ ? ×							
構成(C): アクティブ(Debug)	✓ プラットフォーム(P): アクティブ(	x86)	✓構成マネージャー(0)				
<ul> <li>▲ 構成プロパティ 全般 デパッグ VC++ディレクトリ</li> <li>▲ C/C++ 全般 最適化 プリプロセッサ コード生成 言語 プリコンパイル済みへッダー まやのつった</li> </ul>	<ul> <li>ランタイム型情報を有効にする</li> <li>C 言語標準</li> <li>C++ 言語標準</li> </ul>	いいえ (-fno-rtti) 既定 既定 C++03 (-std=c++98) C++11 (-std=c++11) C++14 (-std=c++1y) C++03 (GNU 言語) (-std=gnu++98) C++11 (GNU 言語) (-std=gnu++11) C++14 (GNU 言語) (-std=gnu++1y)	✓				

開発だけではなく、Visual Studio のインストールと同時に Android エミュレーターもインストールされる ため、デバッグ実行も簡単に行えます。加えて、実機でのデバッグにも対応しています。

また、プラットフォームとして x86 を選択することで、VS Emulator を選択可能になり、ARM を選択する と実機でのデバッグが行えるようになっています。



理君	<b>š</b> )								
B)	デバッグ(D)	₹-L	(M)	XML(X)	ツール(T)	アーキテクチャ(C)	テスト(S)	分析(N)	ウィンドウ(V
•	x86	•	• •	/S Emulator	r 7" KitKat (	(4.4) XHDPI Table	et (x86 - オフ	ライン) 🔸	<b>-</b>
				・ VS Emulator 7" KitKat (4.4) XHDPI Tablet (x86 - オフライン)					
<pre>?&gt; ould also be reflect mas.android.com/apk/</pre>		√	VS Emula	tor 7" KitKa	at (4.4) XHDPI Ta	blet (x86 - :	オフライン)		
			VS Emulator 5" KitKat (4.4) XXHDPI Phone (x86 - オフライン) <sup>ag</sup> pd						

Android アプリケーションの開発であっても Windows Phone と同様に機能が豊富な IDE を使って開発が 行えるのは、大きなメリットだと言えるでしょう。



# 5. Xamarin を利用した Android / iOS アプリケーション開発

Visual Studio 2015 では C#の新規プロジェクトを作成する際に、新しく Android と iOS、そして Mobile App という項目が追加されました。これらは、Xamarin を利用した Android / iOS アプリケーション開発 のテンプレートになります。

これまでのクロスプラットフォームなアプリケーション開発では JavaScript か C++でしたが、Xamarin を 使うことで C#を使って Android / iOS アプリケーションの開発が行えます。標準でインストールされるた め、すぐに開発を始めることが出来ます。

新しいプロジェクト ? ×						
▶ 最近使用したファイル		.NET F	ramework 4.6 👻 並べ替え: 既定	- # <b>E</b>	インストール済み テンプレート の検索 (Ctrl・ 🔎 🗸	
▲ インストール済み		E.	Blank App (Android)	Visual C#	種類: Visual C#	
⊿ テンプレート	<b>^</b>			visual c./	A project for creating a	
▲ Visual C#			Unit Test App (Android)	Visual C#	Xamarin.Android application.	
Windows Web		${\bf Q}^{\rm c*}$	Wear App (Android)	Visual C#		
Cloud Extensibilit	ty	۳. ۲	WebView App (Android)	Visual C#		
₄ iOS Apple V	Vatch	e,	OpenGL App (Android)	Visual C#		
<ul> <li>Classic</li> <li>Extensi</li> </ul>	API ons		Bindings Library (Android)	Visual C#		
iPad iPhone			Class Library (Android)	Visual C#		
Univers	sal h					
Mobile App	os –					
Office/Sha	repoint					
Silverlight						
Workflow	-					
<b>▲</b>	•		オンニノンホニンプレートを始帯すったは	ママをわいったします		
▶ オンライン			<u>オンラインでテンフレードを快発するには</u>	<u> </u>		
名前(N):	App1					
場所(L):	c:¥users¥shiba	yan¥doc	uments¥visual studio 2015¥Projects	•	参照(B)	
ソリューション名(M):	App1				✓ ソリューションのディレクトリを作成(D) リース管理に追加(U)	
					ОК <b>+</b> РУ2И	

プロジェクトテンプレートも Android / iOS ごとに種類が豊富なのも特徴です。

Android では Android Wear アプリの開発まで行えるようになっています。Xamarin では重要な機能といえる、バインディングライブラリもテンプレートから追加可能です。

そして iOS アプリケーションの開発も同様に Visual Studio で行えます。最新の Apple Watch や Extensions の開発まで幅広く行えます。



			新しいプロジェク	۱۲	? ×
▶ 最近使用したファイル		.NET F	ramework 4.6 🝷 並べ替え: 既定	- # <b>E</b>	インストール済み テンプレート の検索 (Ctrl: 🔎 -
▲ インストール済み			Watch App (Apple Watch)	Visual C#	種類: Visual C#
⊿ テンプレート	<b>^</b>			fiodal of	Watch App and Extension projects for
✓ Visual C# Windows		•3	Bindings Library (iOS)	Visual C#	Apple watch.
Web		€ C# ♦2	Class Library (iOS)	Visual C#	
Cloud Extensibili	ty	C#	Unit Test App (iOS)	Visual C#	
⊿ iOS					
Apple \ b Classic	API				
Extensi	ions				
iPad					
iPhone					
Univers	sal				
LightSwite Mobile Apr	n –				
Office/Sha	renoint				
Silverlight	- cponte				
WCF					
Workflow	-				
•	•				
▶ オンライン			オンラインでエンノレートを検系するには、	<u> </u>	
名前(N):	WatchApp1				
場所(L):	c:¥users¥shiba	yan¥doc	uments¥visual studio 2015¥Projects	•	参照(B)
ソリューション名(M):	WatchApp1				✓ ソリューションのディレクトリを作成(D)
					□ ソース管理に追加(U)
					OK         キャンセル

残念ながら iOS アプリケーションに関しては、OS X が動作する実機が必要になります。Windows だけで アプリケーションのビルドまでが行えるわけではありませんので、注意が必要となります。

そして Xamarin の特徴的な機能の一つである Xamarin.Forms を使って開発を行うテンプレートも用意されています。

Xamarin.Forms を使うことで、Android / iOS のアプリケーションを多くのコードを共有したまま開発する ことが可能です。ユーザーインターフェースは WPF で主に使われている XAML を使って、宣言的に記述 することが可能です。



			新しいプロジェクト		? ×
▶ 最近使用したファイル		.NET	Framework 4.6 🔹 並べ替え: 既定	- # E	インストール済み テンプレート の検索 (Ctrl- 🔎 -
▲ インストール済み		C#	Blank App (Native Portable)	Visual C#	種類: Visual C#
▲ テンプレート ▲ Visual C#	1		Blank App (Native Shared)	Visual C#	A multiproject template for building cross-platform native apps, sharing code using a portable class library. On
▷ Windows Web Android		.¢# ⊒≣	Blank App (Xamarin.Forms Portable)	Visual C#	Visual Studio 2010, the Portable Class Library Tools are required.
Cloud Extensibilit	ty		Blank App (Xamarin.Forms Shared)	Visual C#	
₄ iOS Apple V	Vatch		Class Library (Xamarin.Forms Portable)	Visual C#	
<ul> <li>Classic</li> <li>Extensi</li> </ul>	API		UITest	Visual C#	
iPad iPhone					
LightSwite Mobile Apr	h				
Office/Sha Silverlight	repoint				
WCF Workflow	-				
<ul> <li>↓ オンライン</li> </ul>	•		オンラインでテンプレートを検索するには、ここをクリック	します。	
名前(N):	App1				
場所(L):	c:¥users¥shiba	yan¥doo	uments¥visual studio 2015¥Projects	-	参照(B)
ソリューション名(M):	App1				<ul> <li>✓ ソリューションのディレクトリを作成(D)</li> <li>□ ソース管理に追加(U)</li> </ul>
					OK         キャンセル

Mobile Apps カテゴリにもいくつかプロジェクトの種類が用意されています。

中でもプロジェクトの種類で Xamarin.Forms Portable を選択すると、XAML を使ってユーザーインターフェースを定義できるようになります。







プロジェクトを生成すると Windows Phone 8.0 / Android / iOS 向けの 3 プロジェクトと、共有される PCL プロジェクトが生成されます。

これによりコードの大部分を共有しつつ、それぞれのプラットフォームのアプリケーションを同時に開発 していくことが可能です。



# 6. Visual Studio 2015 Tools for Unityの利用

Visual Studio の拡張機能として公開されている Visual Studio Tools for Unity に関しても、2015 Preview の 公開と同時に対応したバージョンが公開されたため、これまでと同様に利用した開発が行えるようになっ ています。

https://visualstudiogallery.msdn.microsoft.com/8d26236e-4a64-4d64-8486-7df95156aba9

Visual S	Free your Unity 3D games inside Micros	ew Tools oft Visual Studio.	for Unity	Report abuse to Microsoft		
CREATED BY	Microsoft	UPDATED	11/12/2014	EXECUTION     EVEN     EV		
REVIEWS	★★★★★(1) Review	VERSION	1.9.8.0			
SUPPORTS	Visual Studio 14	SHARE	🖂 눝 📲 🐏 📑	The second		
DOWNLOADS	Download (379)	FAVORITES	Add to favorites			
TAGS	TAGS unity 3D script c# debug breakpoint					
DESCRIPTION	REVIEWS (1) Q AND A			+ View Large		

注意点として、Visual Studio Tools for Unity をインストールする前に Unity をインストールしておく必要が あります。後からのインストールでは Visual Studio 用のパッケージが Unity に認識されないことがありま す。

#### 6.1. Unity ヘパッケージの追加

Unity で Visual Studio Tools を読み込むには、メニューの「Assets」から「Import Package」を開いて、中 にある「Visual Studio 2015 Tools」を選択します。





これで Visual Studio との連携準備が出来ましたので、後は Unity からスクリプトを追加して生成された cs ファイルをダブルクリックすると Visual Studio のプロジェクトとして開くことが出来るようになります。



▶ UnityVS.New Unity Project 3 - Microsoft Visual Studio (管理者) ファイル(F) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デバッグ(D) チーム(M) ツール(T) テスト(S) アーキテクチャ(C) ロードラスト(L) ○ • ○   谷 • ⓒ   谷 • ⓒ   ♀ (♡ •   Debug • Any CPU • ) ▶ Attach to Unity •   厚 <sub>=</sub> 脸 喧 [ 国 今	マ 🙂 ) 分析(N	クイック起動 (Ctrl ) ウィンドウ(W)	I+Q) へルプ(H) ▼ 2	shibayan	× - 🏹
YewBehaviourScript s       ♥ Start()         YewBehaviourScript +       ♥ Start()         YewBehaviourScript +       ♥ Start()         YewBehaviourScript +       ♥ Start()         YewBehaviourScript +       ₩ NonoBehaviour {         YewBehaviourScript +<	▼ U2-5 0 0 0 1	<ul> <li>メエクスプローラー</li> <li>(1) (2) (2) (2) (2) (2) (2) (2) (2) (2) (2</li></ul>	・ の検索 (Ctrl+:) S.New Unity Project 3.CS aviourScript.cs	<b>ル</b> et 3' (1 プロジェク harp	ب ج ج م (£)
	ソリュー: プロパテ	ション エクスプローラー イ	ƒ−Ь エクスプロ	-∋- クラスビュ ▼	 Ф Х
		\$			
出力   出力元の表示(S): Deployment -   全   転 査   隆   詞					
	~				
準備完了 6行		35 列	32 文字	挿入	<b>#</b>

Unity は.NET Framework 3.5 相当のランタイムとなっているため、Visual Studio でプロジェクトを開く場合 に警告が表示されることがあります。その場合は.NET Framework 3.5 を追加インストールすることで回避 が可能です。

#### 6.2. Visual Studio 2015 を利用したデバッグ

プロジェクトとして開くことが出来れば、後はこれまでと同じように Visual Studio の機能を使って C#コ ードを書くことが出来ます。デバッグも Visual Studio 側で「Attach to Unity and Play」を選択すると Unity 側で自動的にゲームが起動されるため、コーディングに関しては Visual Studio で集中して行えます。





当然ながら、Visual Studio のデバッガーを使っているので、ローカル変数や呼び出し履歴の確認も簡単に 行えるようになっています。



### 7. GitHub との連携の強化

Visual Studio 2015 をインストールする際にオプションとして Git for Windows や GitHub Extension for Visual Studio を同時にインストールできるようになっています。この拡張機能は GitHub から公開されて いるので、GitHub 独自の機能が使うことが出来ます。

#### 7.1. GitHub との接続を行う

GitHub Extension for Visual Studio がインストールされていると、チームエクスプローラに GitHub が表示 されます。



日本語化はされていませんが、Connect を選択すると GitHub アカウントとの連携を行うダイアログが表示されます。

ユーザー名とパスワードを入力し、画面下部にある「Log in」ボタンをクリックするだけで連携が完了します。



	Connect To GitHub	
<b>GitHub</b> Log in		
	GitHub GitHub Enterprise	
	Username or email Password	
	Log in Don't have an account? Sign up	

多要素認証を有効にしている場合にはコードジェネレータで生成された6桁の数字を入力する画面が表示されるので、数字を生成して入力します。

連携が完了すると GitHub の項目がチームエクスプローラに表示されるようになります。

<ul> <li>G ○ ▲ ♥ ▲</li> <li>佐業項目の検索 (Ctrl+^)</li> <li>接続   オフライン</li> </ul>	- م •	
接続   オフライン	+	
接続の管理 ▼		
▲ GitHub Clone   Create   Sign out		

アカウントを切り替える場合には Sign out を実行し、再ログインを行います。

#### 7.2. GitHub 上のリポジトリをクローン

チームエクスプローラ上の GitHub から Clone を選択すると、GitHub でホスティングしているリポジト リー覧が表示されるので、特定のリポジトリをローカル環境へクローンすることが出来ます。

Clone a GitHub Repository	
Search repositories	
📷 of Keyner	
Contractor estimation and the second seco	
aspnet-mvc5-template	
Y	
📮 estat traise service	
Encoder and the set of the set	
Path C:\Users\shibayan\Documents\GitHubVisualStudio Browse	
Clone	

クローンを行いたいリポジトリを選択すると、実際にクローンが行われてチームエクスプローラの「ローカル Git リポジトリ」に一覧表示されるようになります。





一度クローンしてしまえば、後はこれまで通り Git リポジトリとして利用可能です。

GitHub からクローンしたリポジトリには、チームエクスプローラの表示に Pull Request や Issues など、 独自の機能へのリンクが表示されるようになります。



チーム エクスプローラー - ホーム 🔹 平 🗙			
G 🛇 🟠 🛱 🖒 作業項	目の検索 (Ctrl+^) 🌼 🔎 -		
<b>ホーム</b>   aspnet-mvc5-template ▼			
✓ GitHub shibayan/aspnet-mvc5-template https://github.com/shibayan/aspnet-mvc5-temp			
<i>▲</i> プロジェクト	_		
● 変更	🌱 分岐		
Pull Requests	↑↓ 同期		
A Pulse	Graphs		
Issues	🔅 設定		
▲ ソリューション 新規作成   開く MvcApplication.sln   src			

Visual Studio から Pull Request を投げることはまだ出来ませんが、専用の機能へ素早くアクセスできるようになっています。

#### 7.3. 新しいリポジトリを GitHub 上へ作成

GitHub へ新しくリポジトリを作成することもチームエクスプローラから可能になっています。Clone で はなく Create を選択すると、リポジトリを作成する画面が表示されます。



Create a GitHub Repository		
	GitHub	
Name Description Local path Git ignore License	C:\Users\shibayan\Documer Browse VisualStudio • None •	
	<ul> <li>Private Repository</li> <li>Create</li> </ul>	

リポジトリの情報を入力して「Create」ボタンをクリックすると、GitHub 上に新しいリポジトリが作成 されます。

このように Visual Studio 2015 と GitHub Extensions for Visual Studio を使うと、TFS だけではなく GitHub ともシームレスに連携することが出来るようになっています。

