Antenna House PDF Tool API V7.0 サンプルコードのビルドと実行手順（.NET6）

アンテナハウス株式会社

目次

1. はじめに

本書では、まず、PDF Tool API の C#サンプルコードを Windows 環境の Microsoft Visual Studio 2022 でビルドあるいはアプリケーション発行する方法と、発行したアプリケーションファイルを Windows と Linux それぞれで実行する方法を説明します。
次に、Visual Studio Code を利用して、プロジェクト作成、ビルドと実行方法を説明します。

2. PDF Tool API の開発環境を整える

ここでは、Windows において、Microsoft VIsual Studio 2022 を利用して PDF Tool API のサンプルコードをビルドするための環境整備について説明します。

2.1. ライブラリファイル、ヘッダーファイルを配置する

2.1.1. インストーラを利用する場合

(1) 「Setup-Windows¥setup-lib.exe」をダブルクリックするなどして起動します。
(2) ダイアログの指示にしたがってインストールを行います。
(3) インストール途中に、Microsoft Visual C++ 2019 ランタイムライブラリのインストールを促すダイアログが表示された場合は、指示にしたがってインストールを行ってください。
(4) デフォルトのインストール先は以下のフォルダパスです。
{システムドライブ}:¥Program Files¥Antenna House¥PDFToolAPI_V7_lib

2.1.2. インストーラを利用しない場合

「Lib-Windows」フォルダにライブラリファイルやヘッダファイルなどがあります。開発環境の任意の場所に配置します。

2.2.　ライセンスファイルを配置する

2.2.1.　インストーラを利用した場合

インストーラにより、ライセンスファイル「ptalic.dat」は「{インストールフォルダ}¥License」フォルダに配置され、環境変数「PTL70_LIC_PATH」にフォルダパスが設定されます。
インストールされるライセンスファイルは、インストール後 30 日間有効の評価用です。出力される PDF ファイルに透かし文字列が挿入されます。

2.2.2.　インストーラを利用しなかった場合

◆　配置方法 1

(1)　　弊社より発行するライセンスファイルを開発環境の任意の場所に配置します。

(2)　　環境変数「PTL70_LIC_PATH」を作成し、配置したフォルダパスを設定します。

◆　配置方法 2

ライセンスファイルを、PDF Tool API のモジュールファイル「PdfTk70.dll」と同じ場所に配置します。
この場合、環境変数「PTL70_LIC_PATH」の作成は必要ありません。

2.3. フォントの準備

テキスト透かしを挿入したりページ上に文字を描画するには、フォント情報が必要です。
Windows 版では、PDF Tool API はシステムのフォントフォルダに存在するフォントを参照します。

2.3.1. フォントの場所

Windows では、オペレーティングシステムの仕様によりフォントファイルは下記のフォントフォルダに存在しています。

{システムドライブ}:¥WINDOWS¥Fonts
{システムドライブ}:¥Users¥{ユーザー名}¥AppData¥Local¥Microsoft¥Windows¥Fonts

上記のフォントフォルダとは異なる場所にあるフォントを参照する場合には、「フォント構築ファイル」を設定します。

2.3.2. フォント構築ファイルの作成

(1) フォント構築ファイルは、下記の場所にあります。

{インストールフォルダ}¥fontconfig

または、

Lib_Windows¥fontconfig

(2) fontconfig フォルダ内には以下の 2 つのファイルがあります。

font-config.xml ：フォント構築ファイルのひな型
font-config.dtd ：font-config.xml の定義ファイル

(3) font-config.xml の「font-folder path」タグに、フォントファイルが存在するフォルダパスを記述します。

(例)

```
<font-config>
  <font-folder path="C:¥TestFont"></font-folder>
</font-config>
```

(4) font-config.xml と font-config.dtd を任意の場所に配置します。

(5) 環境変数「PTL70_FONT_CONFIGFILE」を作成し、font-config.xml のフルパスを設定します。

2.4. Windows 開発環境における環境変数のまとめ

環境変数名	設定値	設定が必要な場合
PTL70_LIC_PATH	ライセンスファイル「ptalic.dat」が存在するフォルダパス	「PdfTk70.dll」とは異なる場所にライセンスファイルを配置する場合
PTL70_FONT_CONFIGFILE	フォント構築ファイル「font-config.xml」のフルパス	システムのフォントフォルダとは異なる場所にあるフォントを参照する場合

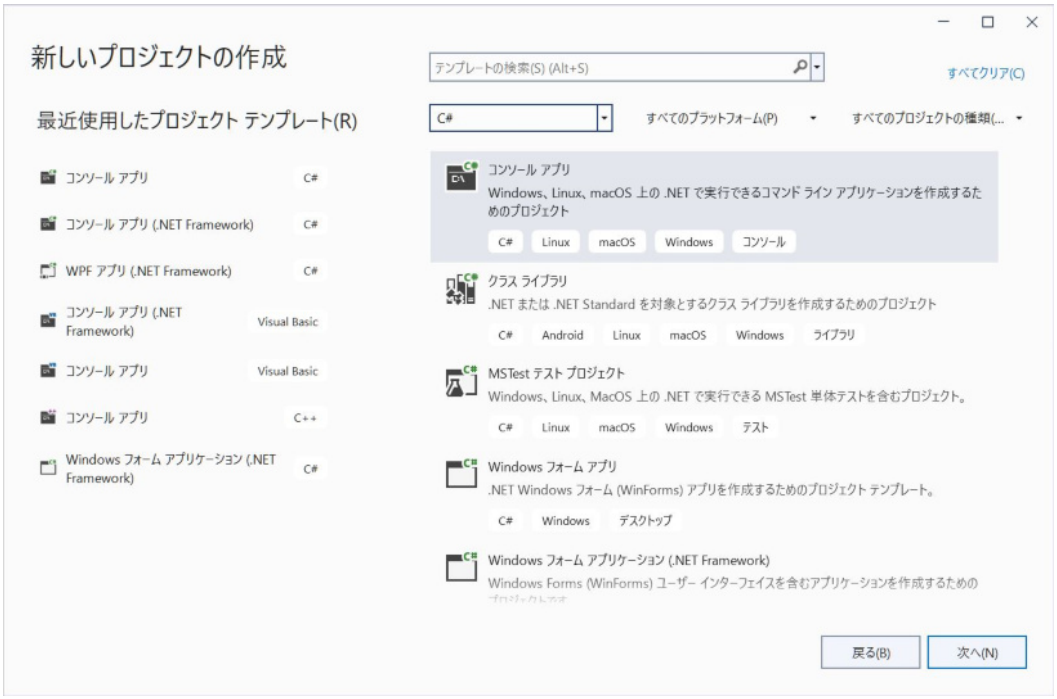
3. サンプルコードのプロジェクト作成とビルド方法

ここでは、サンプルコードをビルドするための Microsoft Visual Studio（Visual Studio）プロジェクトの作成とビルド手順について説明します。
.NET の C#のサンプルコードのソースファイルは、製品版／評価版パッケージの「SampleCode/dotnet」ディレクトリにあります。

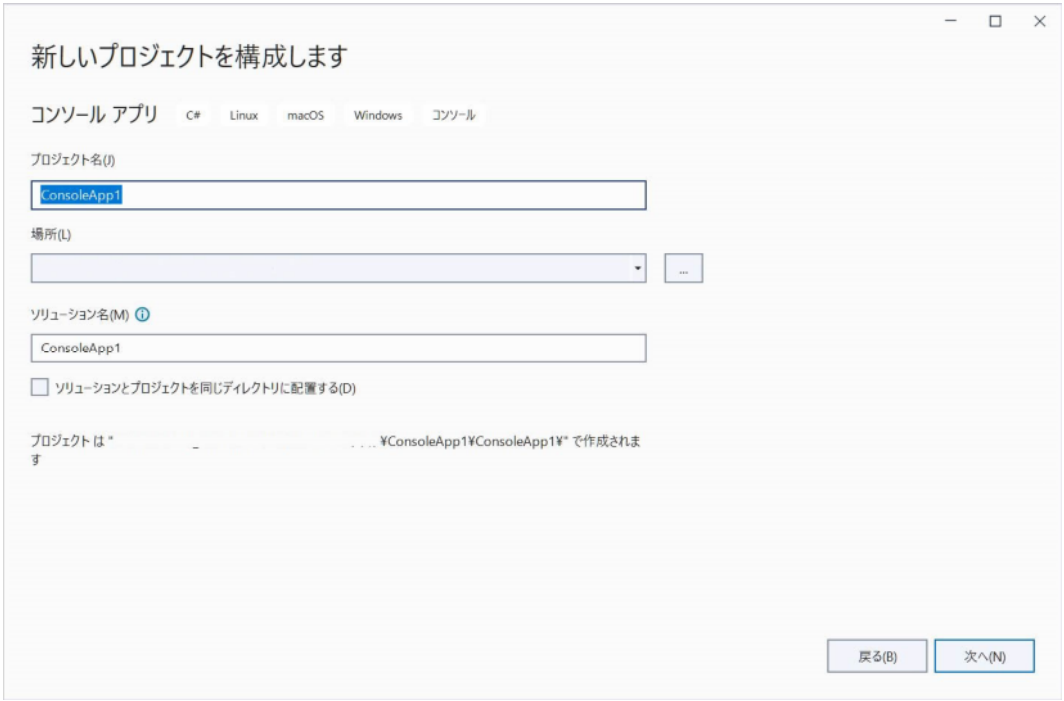
3.1. プロジェクトの新規作成と PDF Tool API モジュールファイルの参照追加

(1) Visual Studio 2022 を起動します。

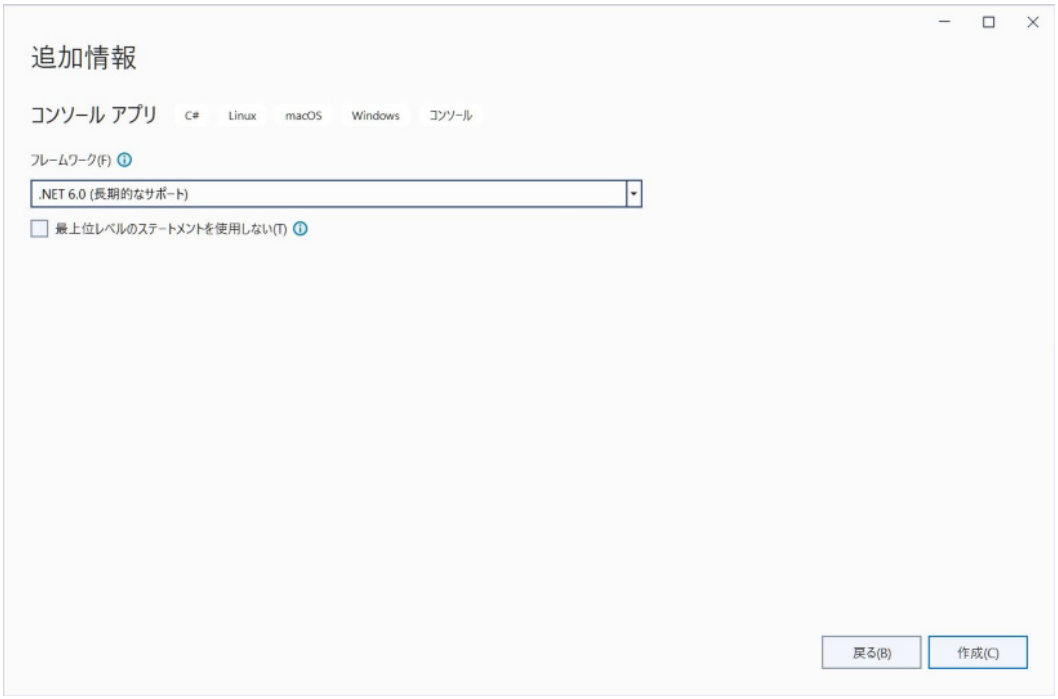
(2) 「新しいプロジェクトを作成」において、C#の「コンソールアプリ」を選択し「次へ」ボタンをクリックします。



(3) 「プロジェクト名」、「場所」を設定し「次へ」ボタンをクリックします。



(4) 「フレームワーク」に「.NET 6.0 (長期的なサポート)」が選択されているのを確認し「作成」ボタンをクリックするとプロジェクトが開きます。

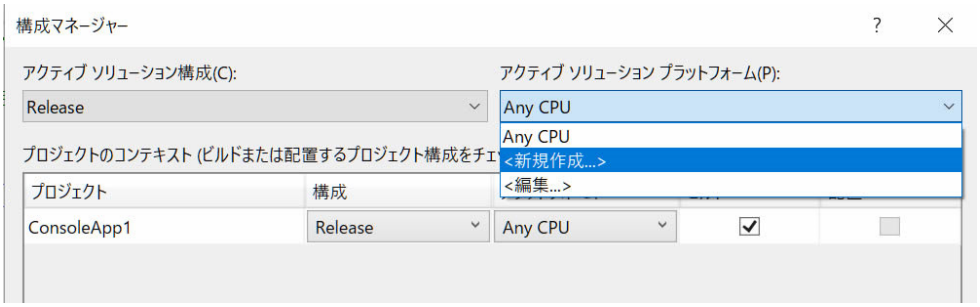


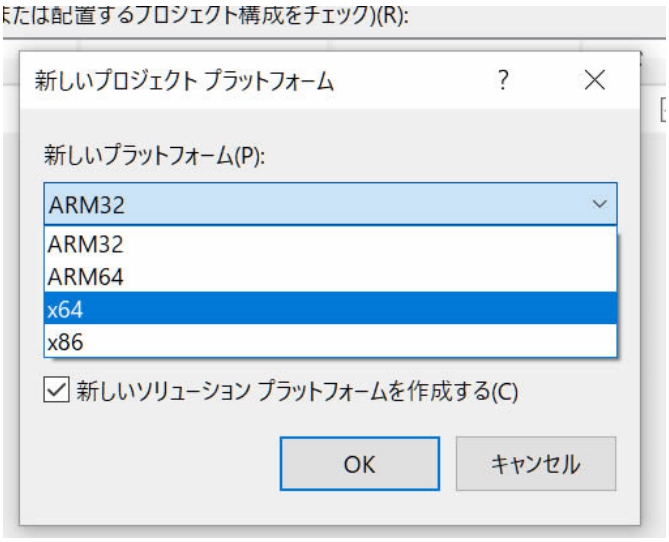
(5) 「プロジェクト」メニューの「既存の項目の追加...」を選択するとファイル選択ダイアログが表示されます。

(6) サンプルの cs ファイルをひとつ選択します。プロジェクトのソースファイルとして追加されます。プロジェクト生成時に作成される cs ファイルはプロジェクトから削除してください。

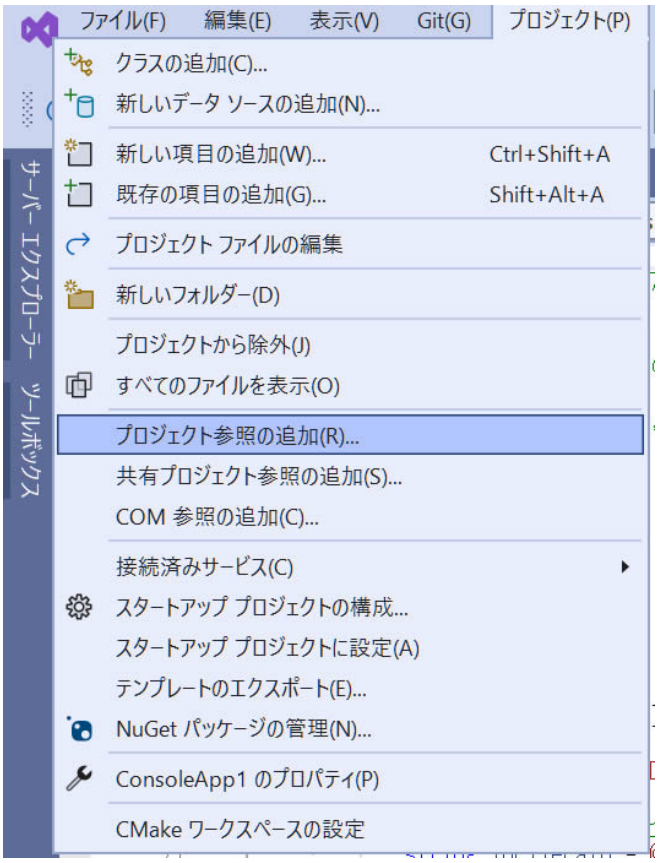
(7) 「ビルド」メニューの「構成マネージャー...」を選択します。

(8) 「アクティブソリューション構成」と「アクティブソリューションプラットフォーム」を設定します。プラットフォームは、「x86」または「x64」のどちらかを新規作成します。

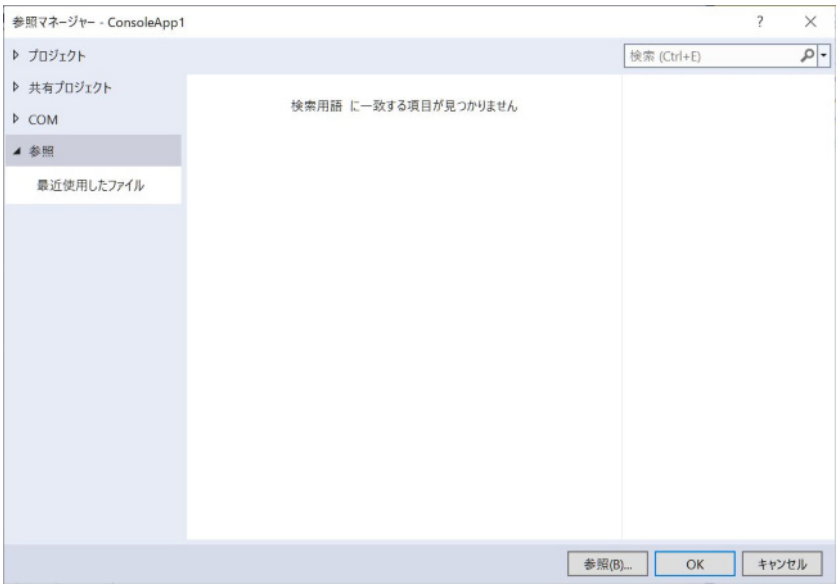




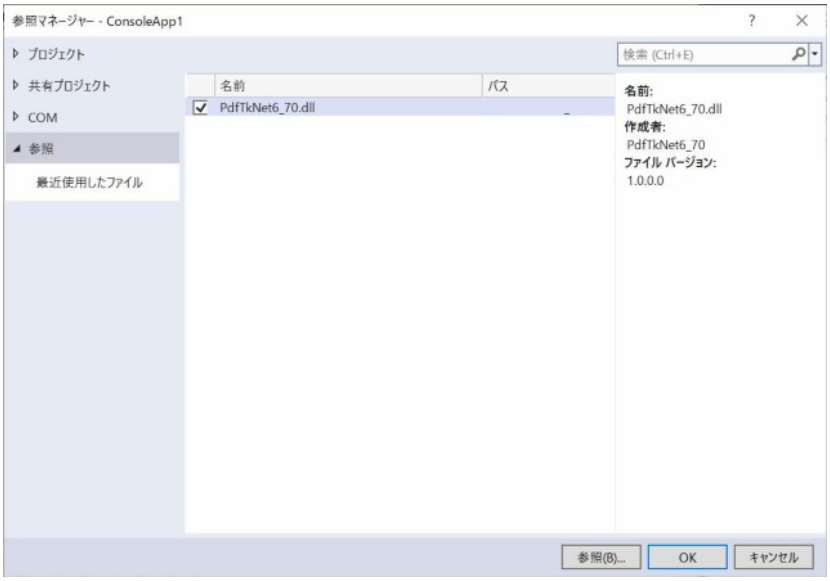
(9) 「プロジェクト」メニューの「プロジェクト参照の追加...」を選択すると、「参照マネージャー」ダイアログが開きます。



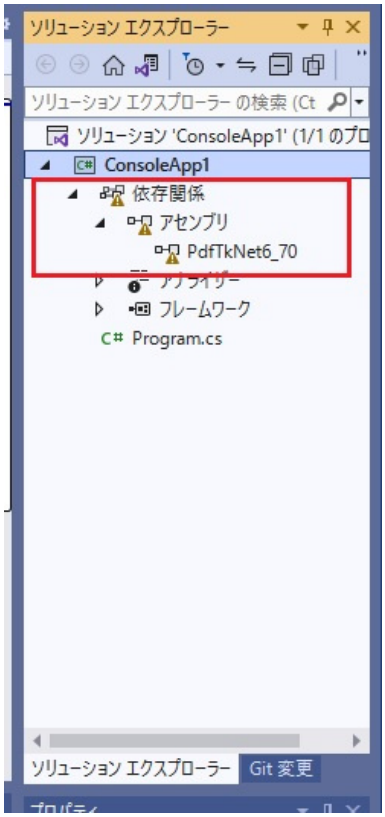
(10) 「参照」タブを開き、ダイアログ右下の「参照...」ボタンをクリックします。



(11) ファイル選択ダイアログが開きます。インストールフォルダまたは任意の場所に配置した PDF Tool API モジュールファイルの「PdfTkNet6_70.dll」を選択します。ソリューションプラットフォームが「x64」の場合は「bin64」フォルダにある 64bit 用、「x86」の場合は「bin32」フォルダにある 32bit 用の「PdfTkNet6_70.dll」を選択してください。



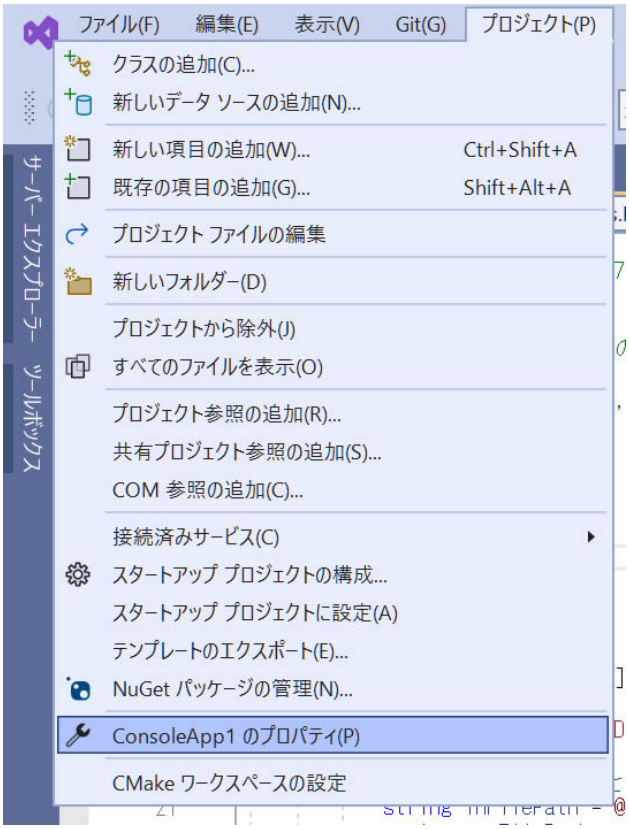
(12) 「OK」ボタンをクリックして「参照マネージャー」ダイアログを閉じます。参照した DLL は、ソリューションエクスプローラーの「依存関係」 - 「アセンプリ」で確認できます。



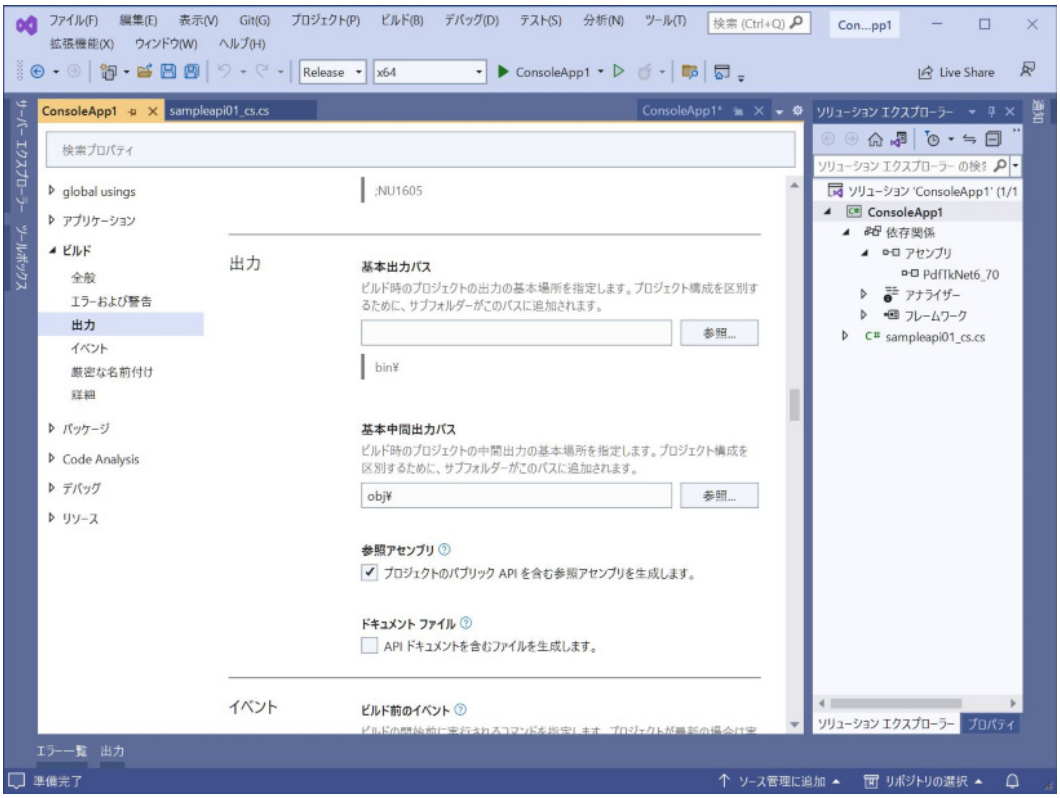
3.2. ビルドの設定とビルド

Windows 用のビルドを行います。

(1) 「プロジェクト」メニューの「(プロジェクト名) のプロパティ」を選択します。
(図のプロジェクト名は ConsoleApp1 です。)



(2) 左側の「ビルド」タブを開き「出力」を選択します。



(3) 「基本出力パス」の欄に PdfTkNet6_70.dll、PdfTkNet6.dll を含む PDF Tool API の dll が存在するフォルダを指定します。

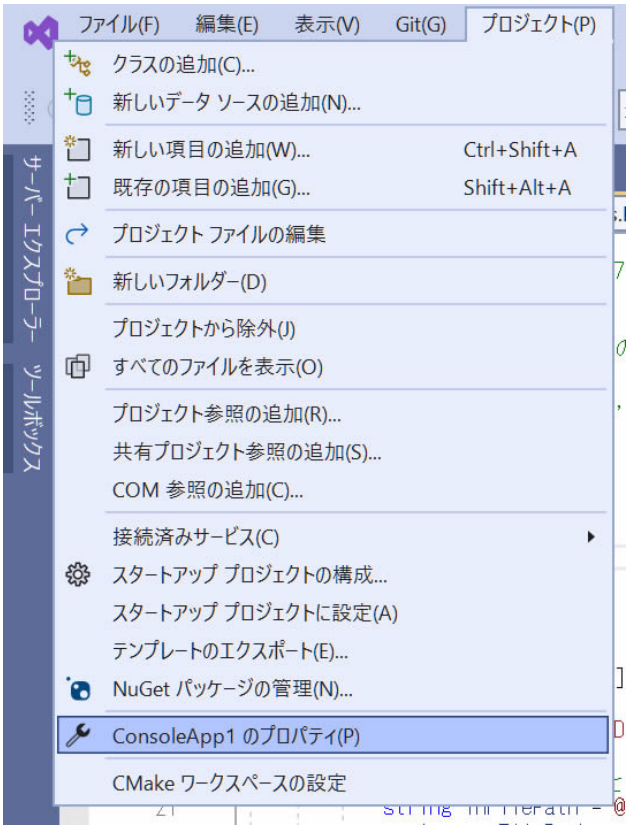
(4) 「ビルド」メニューの「ソリューションのビルド」をクリックすると、ビルドが開始されます。

(5) ビルドされた exe を実行するには、exe とともに作成される、exe と同名の「.dll」と「.runtimeconfig.json」が必要です。

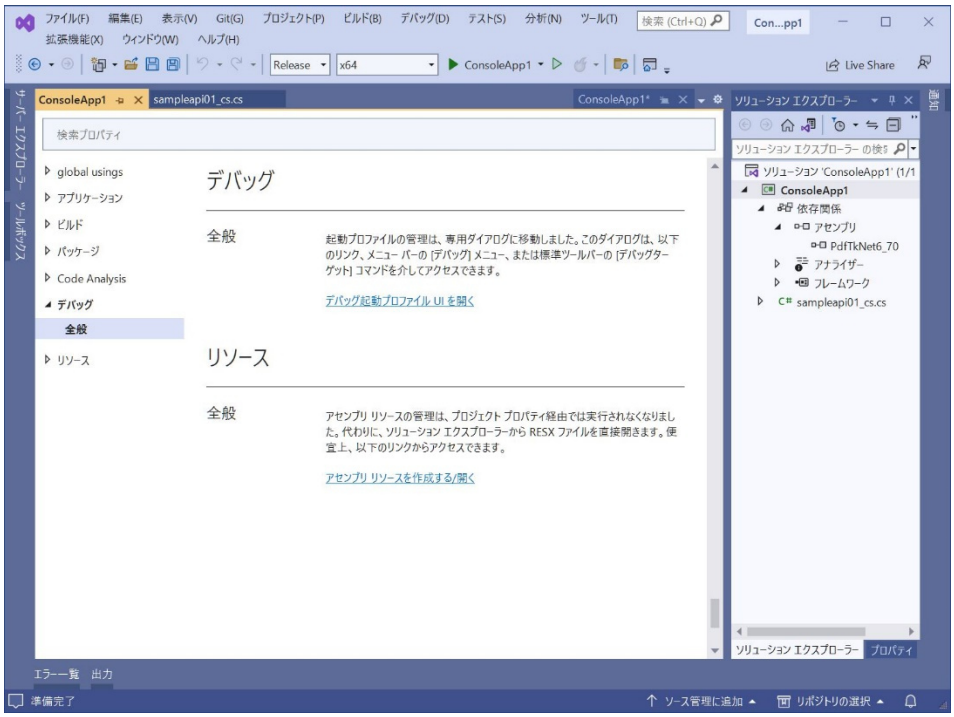
3.3. デバッグビルドの設定とデバッグ実行

Windows 用のデバッグビルドを行います。

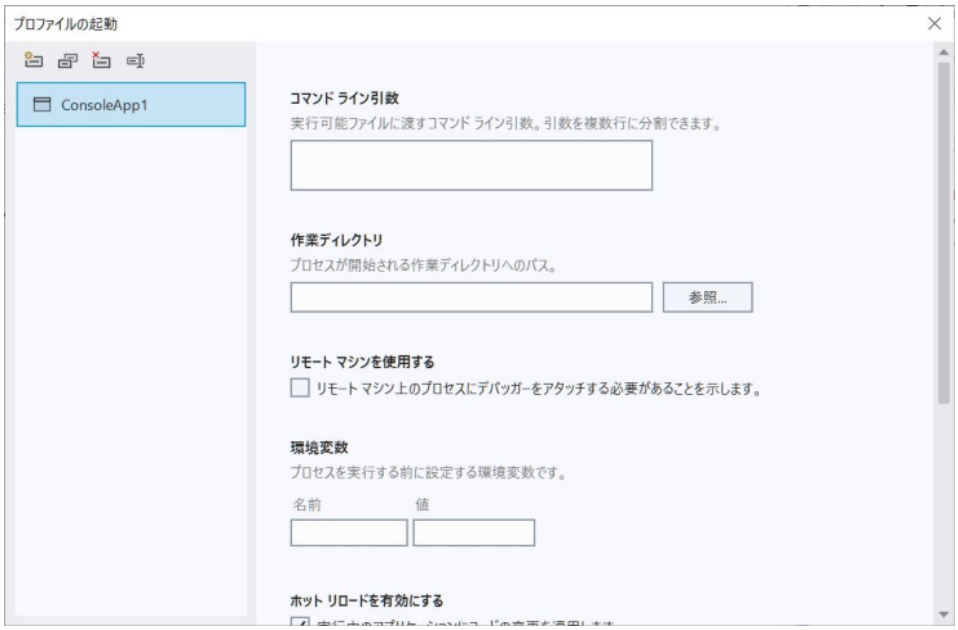
(1) 「プロジェクト」メニューの「(プロジェクト名) のプロパティ」を選択します。
(図のプロジェクト名は ConsoleApp1 です。)



(2) 左側の「デバッグ」タグを開き「デバッグ起動プロファイル UI を開く」をクリックして「プロファイルの起動」ダイアログを開きます。



(3) 「プロファイルの起動」ダイアログの「作業ディレクトリ」の欄に PdfTkNet6_70.dll、PdfTkNet6.dll を含む PDF Tool API の dll のあるフォルダを指定します。必要に応じて、「コマンドライン引数」を指定します。

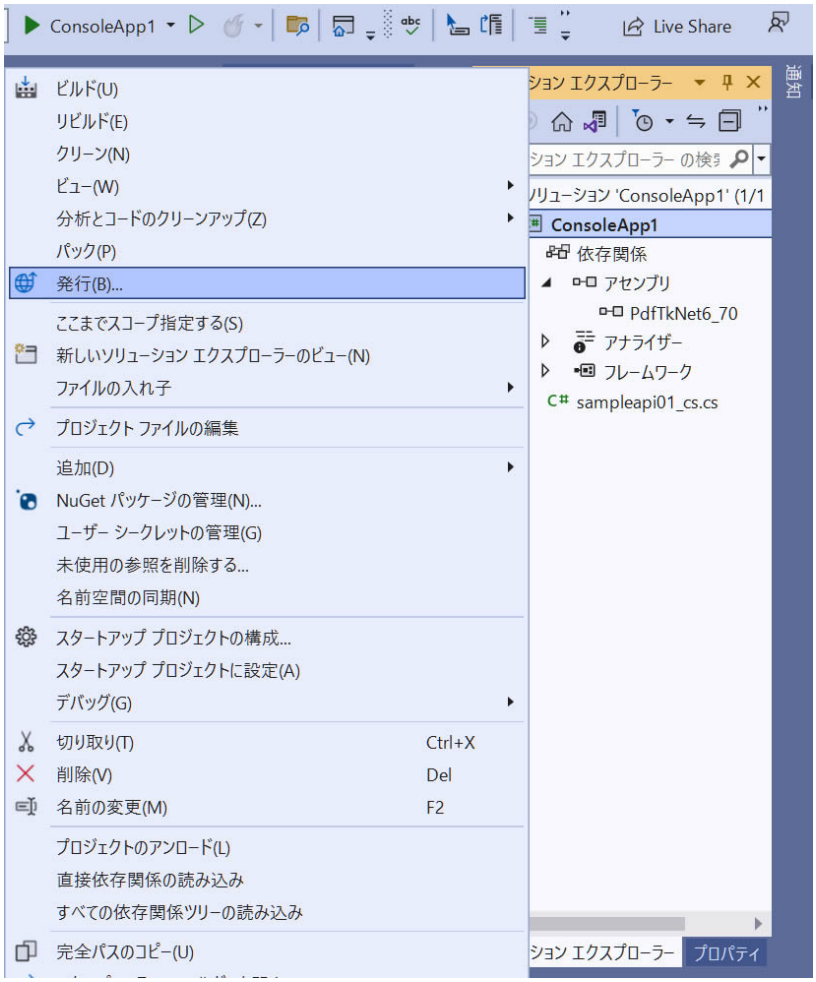


(4) 「デバッグ」メニューの「デバッグの開始」をクリックすると、デバッグ実行できます。

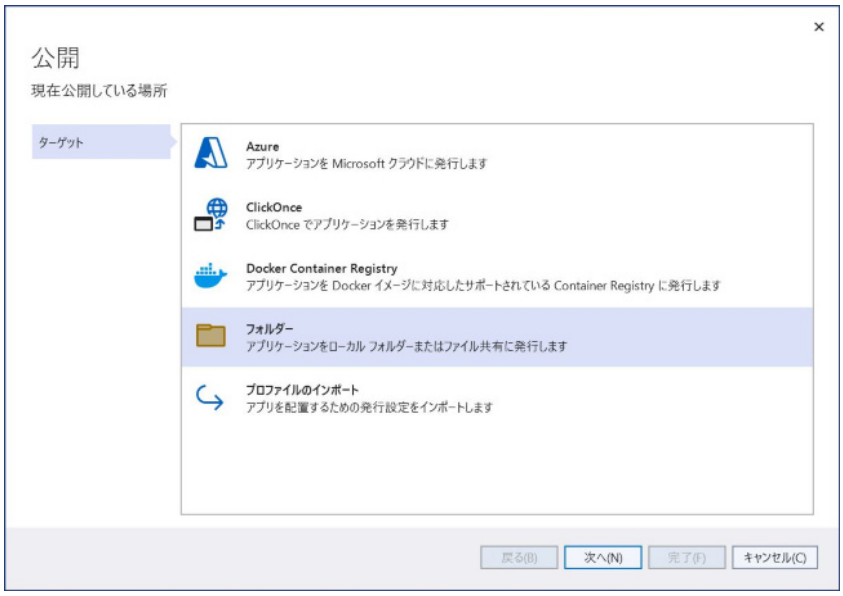
3.4.　アプリケーションの発行

Windows 用、Linux 用それぞれのアプリケーションファイルを発行します。

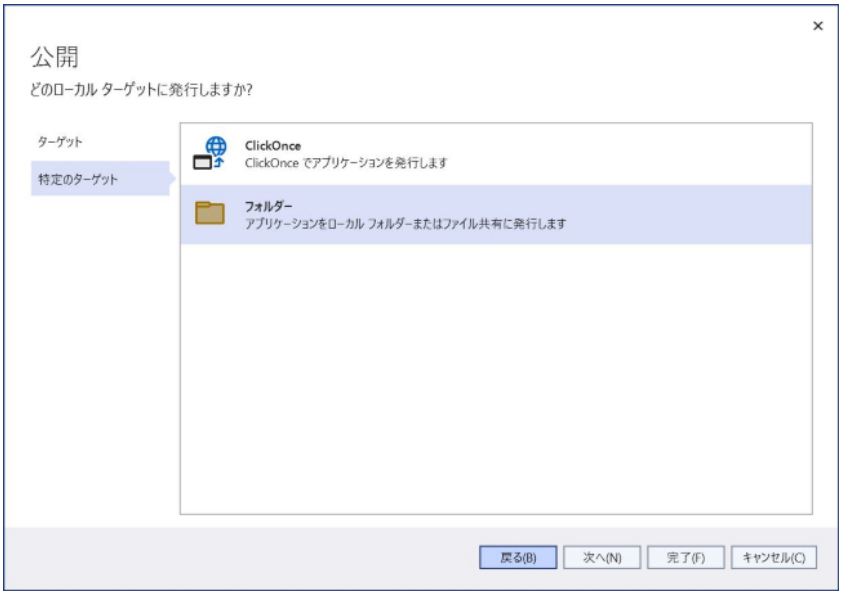
(1)　プロジェクトで右クリックし開いたメニューで「発行. . .」を選択します。



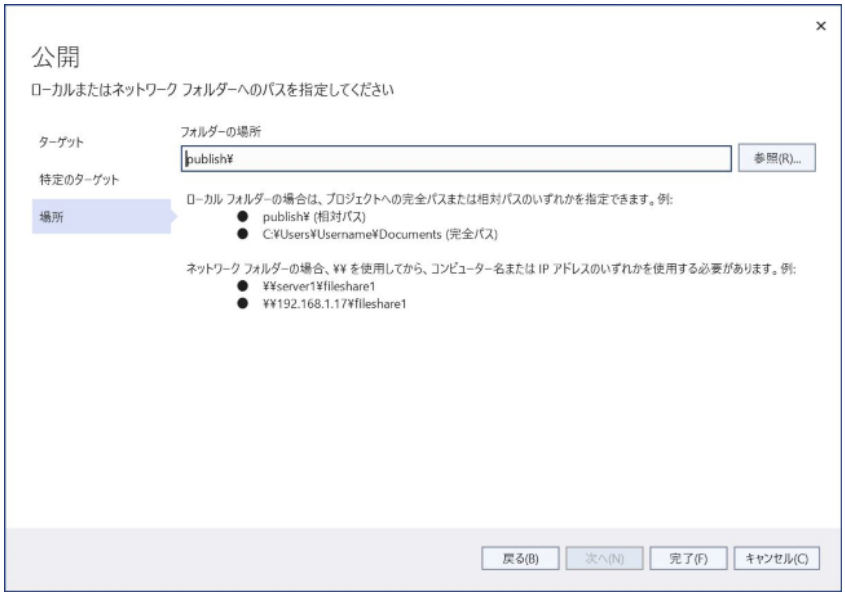
(2) 「公開」ダイアログが開くので公開先を指定します。ここでは、ローカルフォルダーにアプリケーションを作成しますので、「フォルダー」を選択し、右下の「次へ」ボタンをクリックします。



(3) 次の画面でも「フォルダー」を選択し右下の「次へ」ボタンをクリックします。

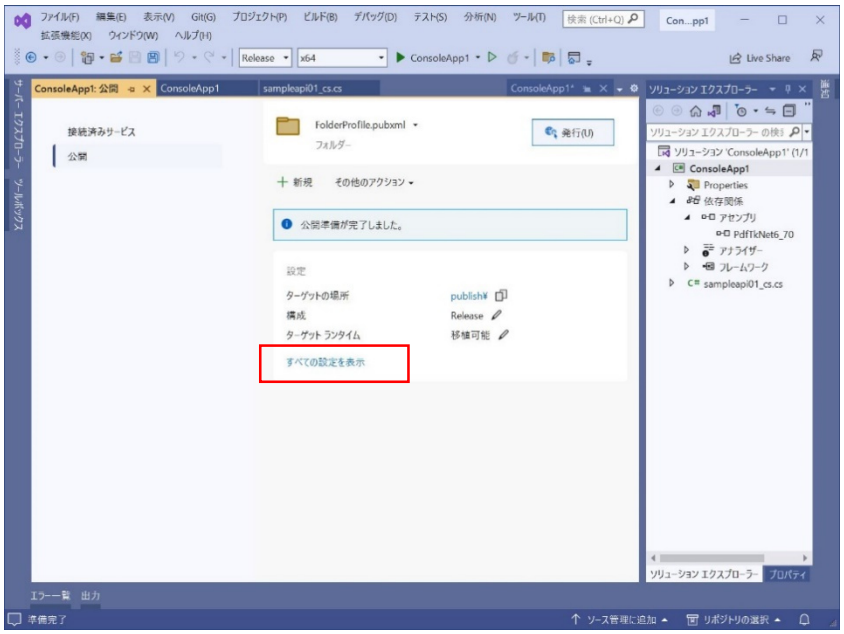


(4) 「フォルダーの場所」を指定します。



(5) 「実行プロファイル作成の進行状況」が表示された場合は、「閉じる」ボタンをクリックします。

(6) 中央下にある「すべての設定を表示」をクリックします。



(7) 「プロファイル設定」ダイアログが開くので各項目を設定します。

ターゲットフレームワーク：net6.0

配置モード：「自己完結」

発行されるアプリケーションファイルには、ターゲットフレームワークのランタイムライブラリーが含まれます。このため、アプリケーションファイルを実行する環境に、ターゲットフレームワークの.NET ランタイムライブラリーをインストールする必要はありません。ただし、ファイルサイズは大きくなります。

配置モード：「フレームワーク依存」

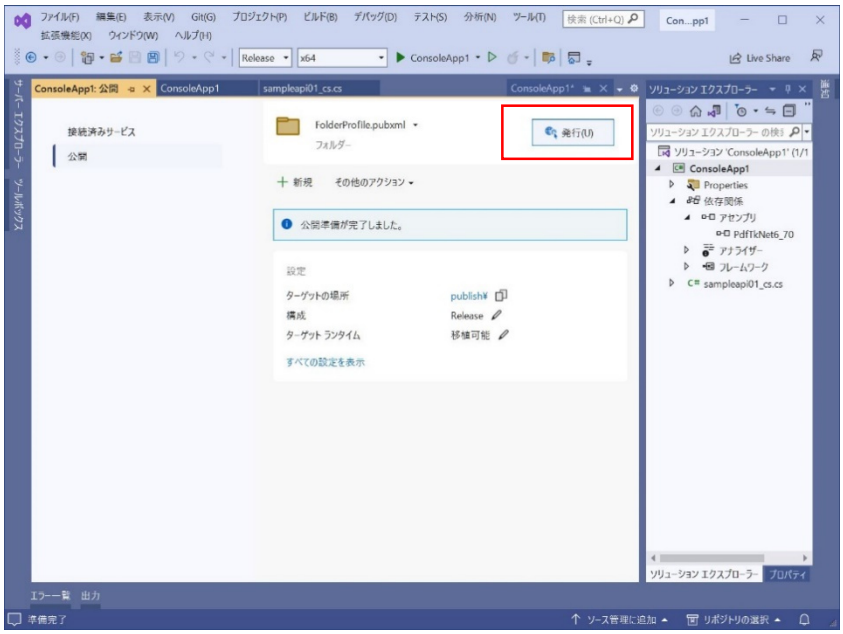
発行されるアプリケーションファイルには、ターゲットフレームワークのランタイムライブラリーは含まれません。アプリケーションファイルを実行する環境には、ターゲットフレームワークの.NET ランタイムライブラリーをインストールしてください。

ターゲットランタイム

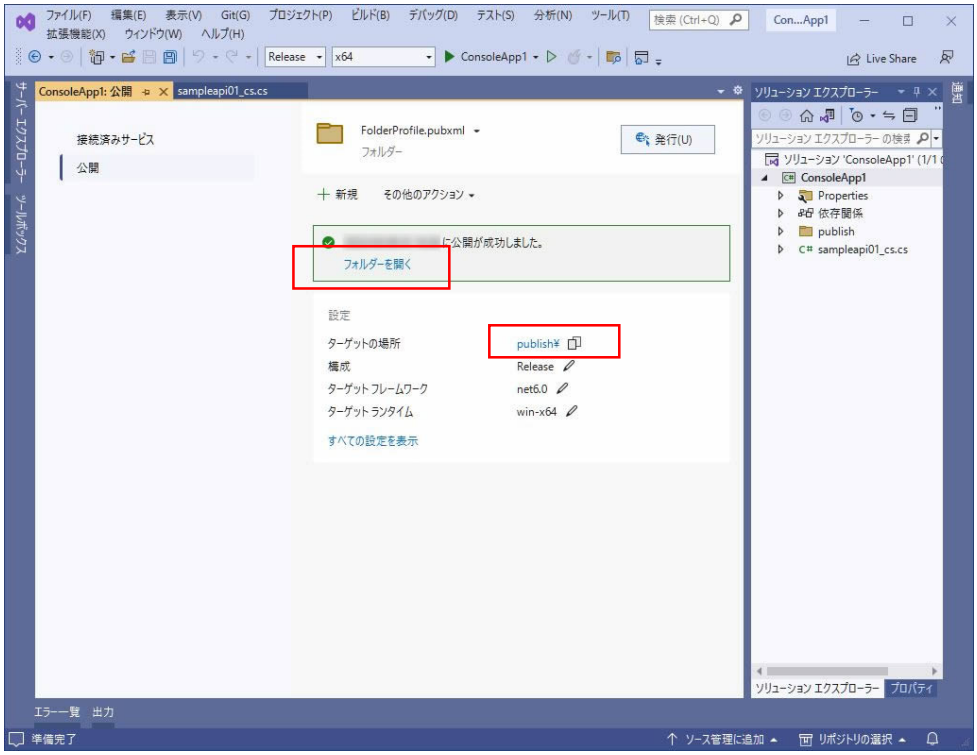
「win-x64　(Windows 64bit)」、「win-x86　(Windows 32bit)」、「linux-x64 (Linux 64bit)」のいずれかを選択します。PDF Tool API は、arm、osx（Mac OS X）には対応していません。

(8)　「ファイルの公開オプション」を開きます。「単一ファイルの作成」にチェックを入れます。

(9)　「発行」ボタンをクリックすると、「(日時) に公開が成功しました。」と表示され発行が完了します。



(10) 「フォルダーを開く」または「ターゲットの場所」をクリックすると「フォルダーの場所」に指定したフォルダが開き、発行されたファイルを確認できます。



(11) 「ターゲットランタイム」が「win-x86」、「win-x64」の場合、発行された拡張子「.exe」ファイルが実行ファイルです。.exe ファイルと PDF Tool API のモジュールファイルを同じフォルダに配置することで実行可能です。

「ターゲットランタイム」が「linux-x64」の場合、発行された拡張子のないファイルが実行ファイルです。

4.　発行されたファイルの実行方法について

ここでは、『アプリケーションの発行』で発行されたアプリケーションファイルを、開発環境とは異なるコンピュータ上で実行する方法を説明します。

4.1.　Windows での実行について

(1)　「配置モード」が「フレームワーク依存」であるアプリケーションファイルの場合、あらかじめ、ターゲットフレームワークの.NET ランタイムライブラリーをインストールしてください。※1
(2)　PDF Tool API のモジュールファイルをセットアップします。※2
(3)　発行されたアプリケーションファイルを、PDF Tool API のモジュールファイルと同じ場所に配置します。あるいは、環境変数「PATH」に、PDF Tool API のモジュールファイルが存在するフォルダパスを設定します。
(4)　PDF Tool API のライセンスファイルを配置します。「「PdfTk70.dll」」と同じ場所に配置する、または、ライセンスファイルを配置したフォルダパスを環境変数「PTL70_LIC_PATH」に設定してください。※3
(5)　必要に応じ、フォント構築ファイルの設定を行ってください。※4
(6)　コマンドプロンプトを起動しアプリケーションファイルが存在するフォルダパスをカレントディレクトリにして実行します。必要に応じて、コマンド入力時に引数を指定してください。

※1 .NET6 ランタイム　インストーラの入手先
https://dotnet.microsoft.com/ja-jp/download/dotnet/6.0
「.NET Runtime 6.0.xx」の項にある「Windows」の「x64」または「x86」インストーラーをダウンロードしてセットアップしてくだい。

※2 実行環境にモジュールファイルをセットアップする場合、PDF Tool API 付属のインストーラは使用しないでください。

※3 『Windows 開発環境における環境変数のまとめ』をご参照ください。

※4 『フォントの準備』をご参照ください。

4.2. Linux での実行について

(1) 「配置モード」が「フレームワーク依存」であるアプリケーションファイルの場合、あらかじめ、ターゲットフレームワークの.NET ランタイムライブラリーをインストールしてください。※1
(2) PDF Tool API のモジュールファイルをセットアップします。※2
(3) アプリケーションファイルを任意の場所に配置します。
(4) PDF Tool API のライセンスファイルを配置します。
(5) テキスト透かしやテキスト追加など文字を扱う処理を行う場合、「font-config.xml」（フォント構築ファイル）にてフォントディレクトリの設定を行います。Linux では、フォント構築ファイルは必須です。
(6) モジュールファイル、フォント構築ファイル、ライセンスファイルを利用するための環境変数を設定します。※3
(7) 「端末」を起動し、アプリケーションファイルがあるディレクトリをカレントにして実行します。必要に応じて、コマンド入力時に引数を指定してください。※4a、※4b

※1 『付録 1.,NET6 ランタイムのインストール方法』参照

※2 実行環境にモジュールファイルをセットアップする場合、PDF Tool API 付属のインストーラは使用しないでください。

※3 環境変数と設定値
LD_LIBRARY_PATH={PDF Tool API モジュールファイルのディレクトリ}:${LD_LIBRARY_PATH}
PTL70_LIC_PATH={ライセンスファイルのディレクトリ}
PTL70_FONT_CONFIGFILE={ font-config.xml のパス}

※4a 「配置モード」が「自己完結」設定のアプリケーションファイルを実行する場合
アプリケーションファイルに.NET6 ランタイムライブラリーが含まれているため、「dotnet」コマンドは不要です。
コマンド：./{アプリケーションファイル名}{オプション}

※4b 「配置モード」が「フレームワーク」設定のアプリケーションファイルを実行する場合
「dotnet」コマンド付けて実行ます。
コマンド：dotnet {アプリケーションファイル名}{オプション

付録 1..NET6 ランタイムのインストール方法

(1) 下記のサイトから NET6 ランタイムのインストーラを入手します。
https://dotnet.microsoft.com/ja-jp/download/dotnet/6.0
「.NET Runtime 6.0.xx」の項にある「dotnet-install scripts」をクリックします。

(2) 「Bash」の項のアドレスからスクリプトをダウンロードします。

(3) インストールを実行します。
コマンド：sh ./dotnet-install.sh --version latest --runtime dotnet
（.NET6 ランタイムの最新バージョンをインストールする場合）

(4) インストール先を指定しなかった場合、「ホーム($HOME)/.dotnet」ディレクトリにインストールされます。「.dotnet」ディレクトリが表示されない場合、「隠しファイルを表示する」にチェックを入れてください。

(5) インストールした.NET6 ランタイムの「dotnet」コマンドを使用するには、次の環境変数の設定が必要です。
「ホーム」の.bashrc （または.bash_profile）に以下の環境変数を追記し、システムを再起動します。
export DOTNET_ROOT=$HOME/.dotnet
export PATH=$PATH:$DOTNET_ROOT:$DOTNET_ROOT/tools

(6) インストールしたランタイムのバージョンは以下のコマンドとオプションで確認できます。

コマンド：dotnet --list-runtimes

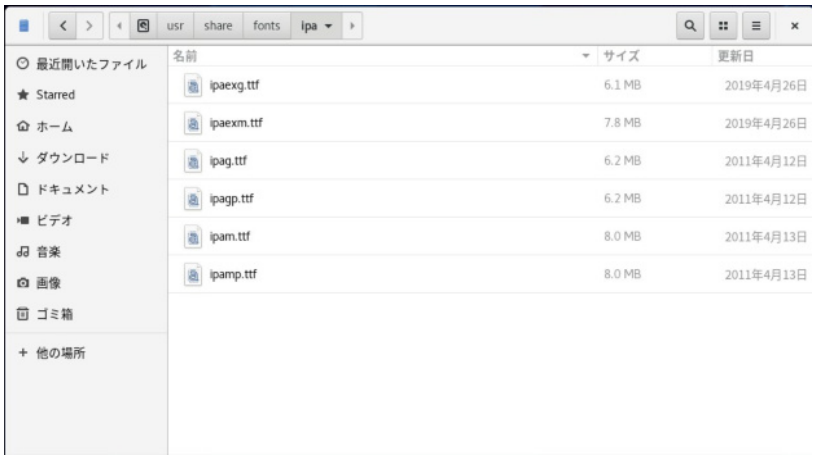
「Microsoft.NETCore.App 6.0.16 [/home/test/.dotnet/shared/Microsoft.NETCore.App]」
のように表示されます。

付録 2.IPA フォントのインストール方法

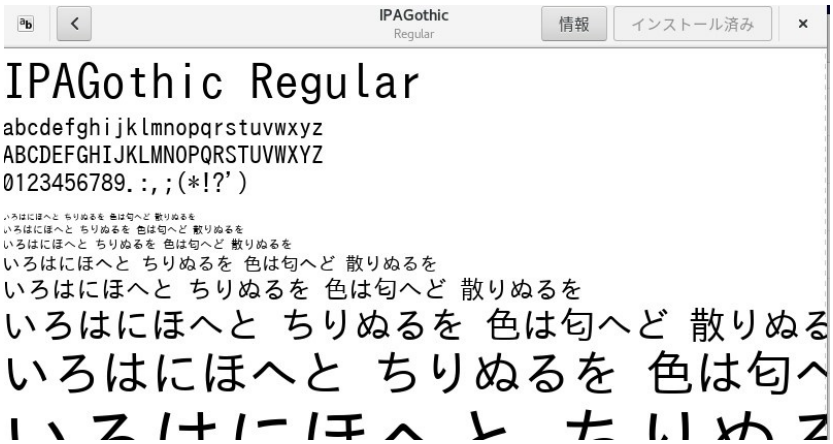
Linux では、動作環境によって日本語用のフォントが存在しない場合があります。ここでは、「IPA フォント」のインストール方法を説明します。

(1) 下記のサイトからフォントファイルをダウンロードします。
文字情報技術促進協議会：https://moji.or.jp/ipafont/

(2) ダウンロードした ttf ファイルを任意のディレクトリに配置します。ここでは、「/usr/share/fonts/ ipa」に配置します。



(3) ttf ファイルを開き、右上の「インストール」ボタンをクリックしてインストールします。



5. Visual Studio Code でのプログラム作成と実行方法

ここでは、Visual Studio Code(VSCode)を使用して、.NET6 のプログラム作成と実行方法を説明します。

5.1. Linux 環境

5.1.1. NET SDK のインストール

『付録 1..NET6 ランタイムのインストール方法』の(3)でのインストールコマンドを
./dotnet-install.sh --version latest
に変更して実行します。このコマンドにより、.NET6 をランタイムとしてではなく、SDK としてインストールします。

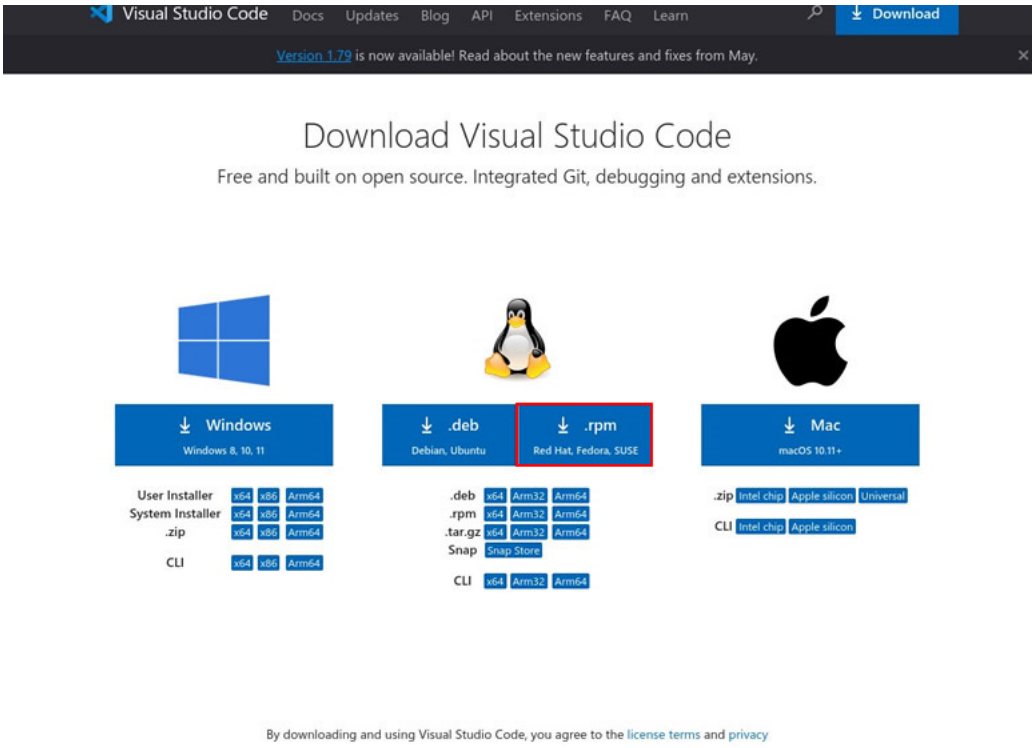
インストールしたランタイムのバージョンは以下のコマンドとオプションで確認できます。
コマンド：dotnet --list-sdks
「6.0.408 [/home/test/.dotnet/sdk]」
のように表示されます。

5.1.2. Visual Studio Code のインストール

(1) Download Visual Studio Code

https://code.visualstudio.com/download

を開いて「.rpm」をダウンロードします。

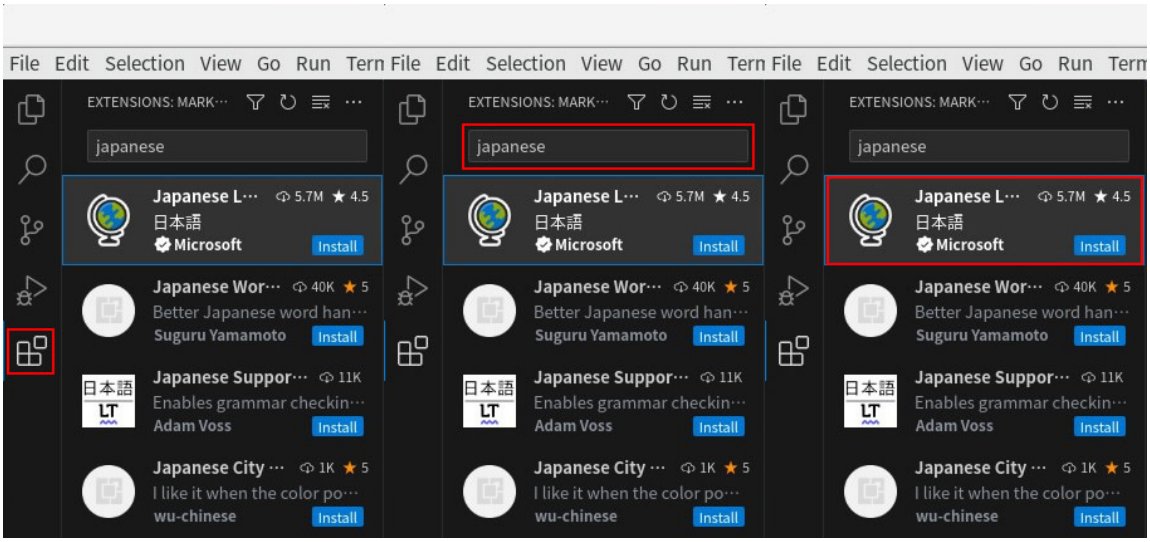


(2) ダウンロードした rpm ファイルをクリックして開きます。
左上の「インストール」ボタンを押すとインストールが始まります。

(3) VSCode 拡張機能「Japanese Language Pack for VS Code」のインストール

左側アクティビティバーにある拡張機能をクリックします。

開いたサイドバー上部にある検索欄に「Japanese」と記入し結果に表れる「Japanese Language Pack for VS Code」を選択します。



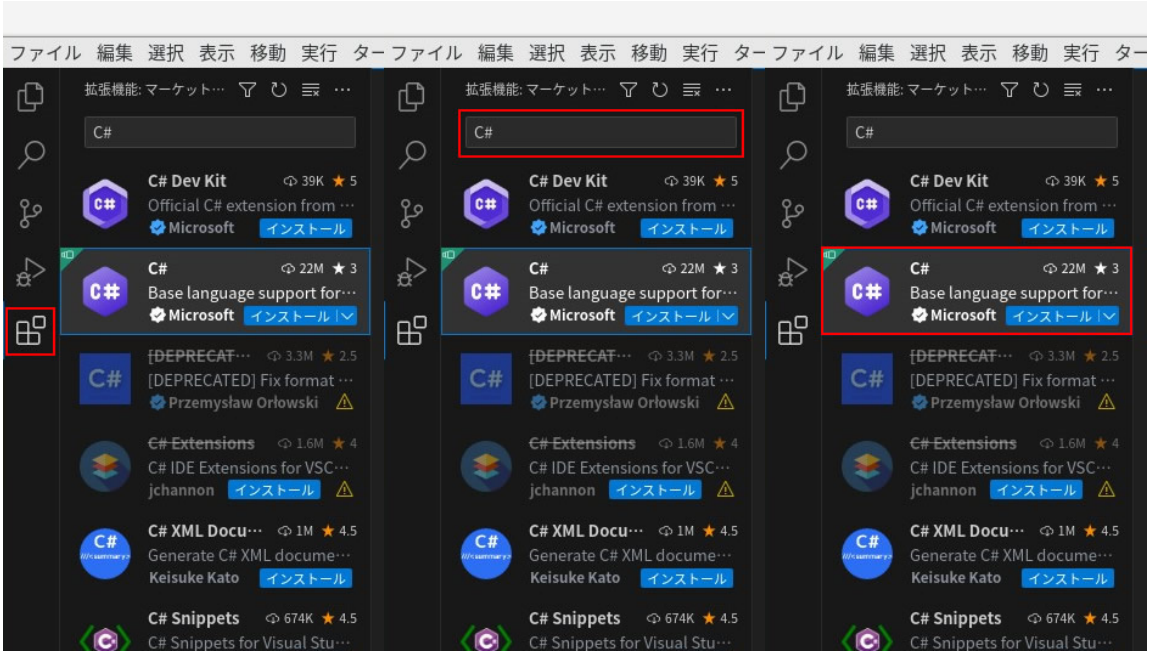
青色の「install」ボタンを押すとインストールが始まります。

インストール終了後に再起動をすることによって日本語が適応されます。

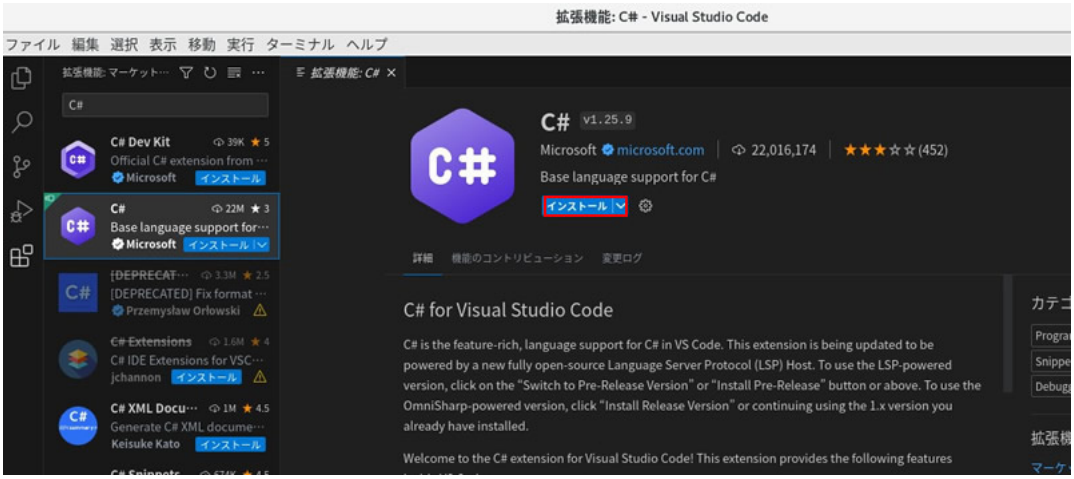
(4) VSCode 拡張機能「C#」のインストール

左側アクティビティバーにある拡張機能をクリックします。

開いたサイドバー上部にある検索欄に「C#」と記入し結果に表れる「C#」を選択します。



青色の「install」ボタンを押すとインストールが始まります。



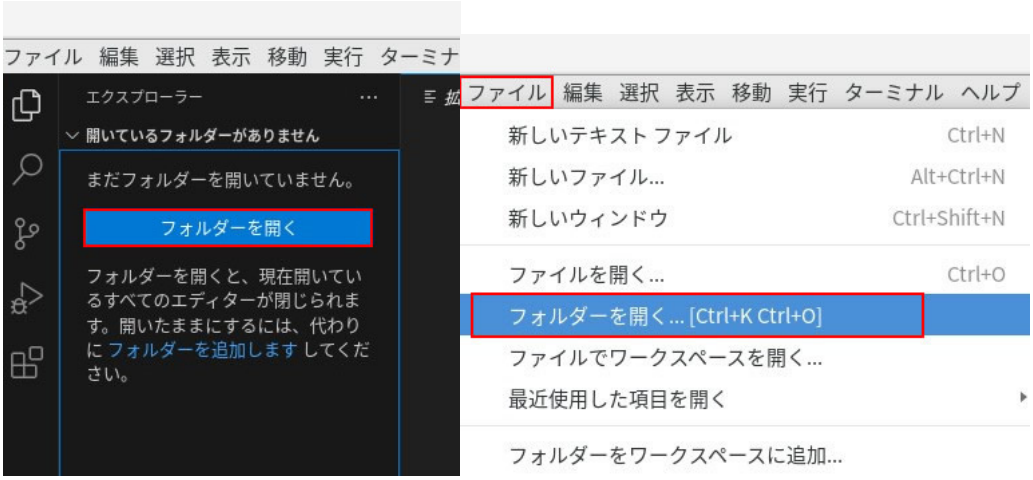
5.1.3. プロジェクトの作成

「フォルダを開く」と「ターミナルを開く」の２つの方法があります。

5.1.3.1. フォルダを開く

(1) 左側アクティビティバーにあるエクスプローラーを選択します。

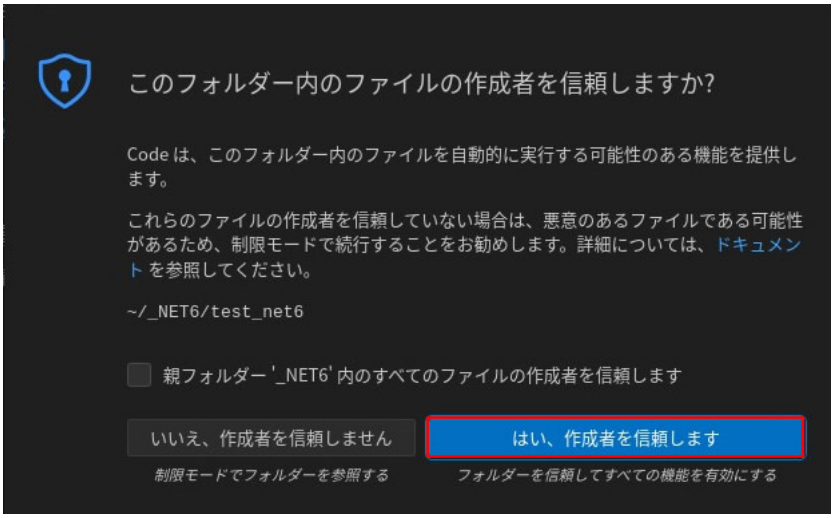
(2) サイドバーに表示された「フォルダーを開く」をクリックします。
または、メニューバー左上の「ファイル」を選択し、開いたメニューの「フォルダーを開く」をクリックします。



(3) プロジェクトのファイルを生成するフォルダを指定して右上の「開く」をクリックします。

(4) 左側のサイドバーのエクスプローラーに指定したファイル名が表示されます。
（画像では TEST_NET6 というフォルダを指定しています）

(5) 下記のダイアログが表示されるので「はい、作成者を信頼します」を選択します。



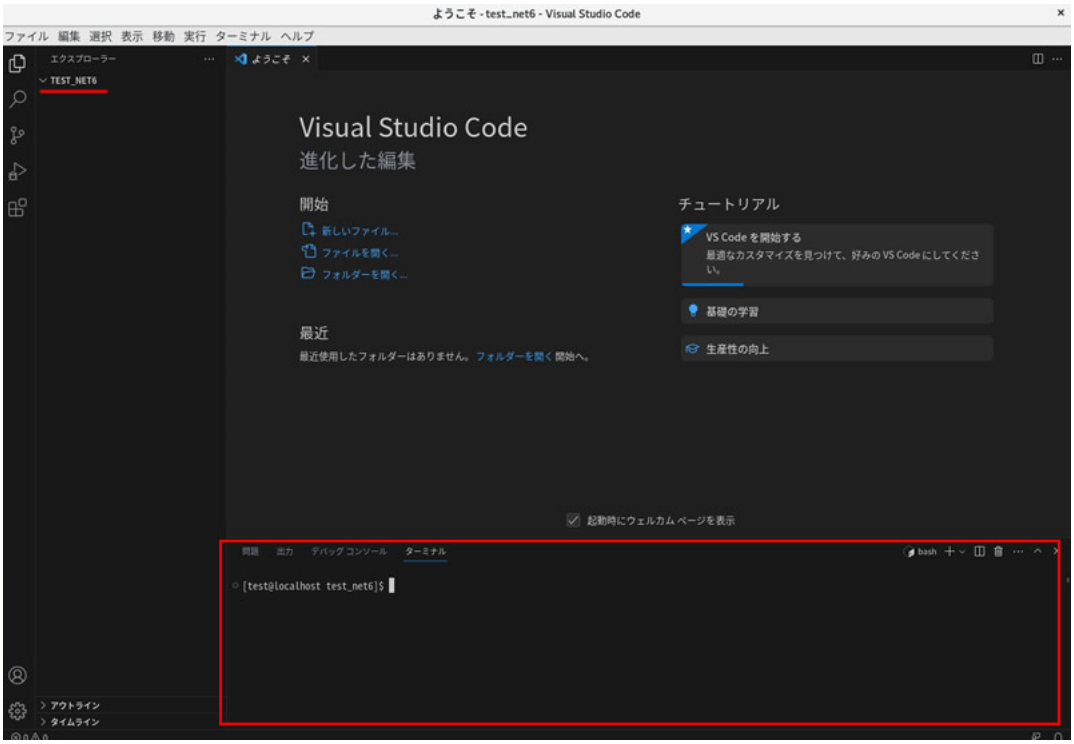
5.1.3.2.　ターミナルを開く

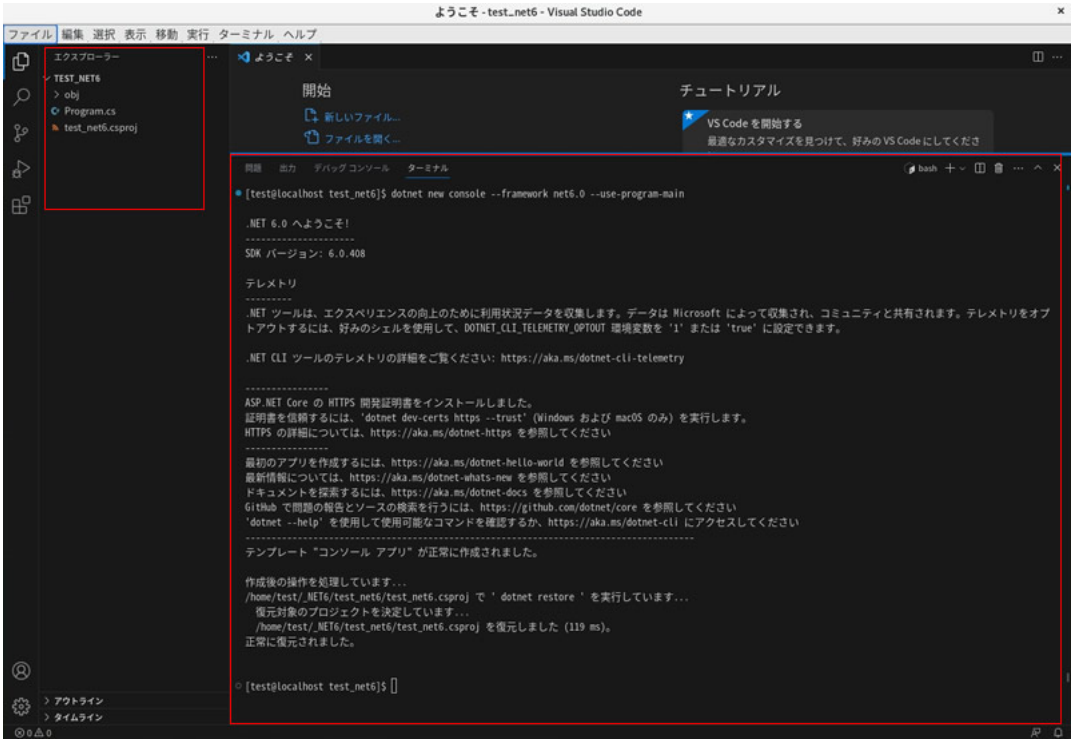
(1)　「ターミナル」メニューの「新しいターミナル」を選択します。

(2) 開かれたターミナル画面に dotnet new のコマンドを入力して新しいコンソールアプリケーションのプロジェクトを作成します。

コマンド：dotnet new console --framework net6.0 --use-program-main

（ dotnet new <TEMPLATE> - .NET CLI | Microsoft Learn ）





5.1.4. PDF Tool API のサンプルコードを使用したプログラムの作成

(1) PDF Tool API のインストール
「2.PDF Tool API の開発環境を整える」を参照してください。

(2) VSCode を起動してプロジェクトを作成します。

(3) 「(フォルダ名) .csproj」 を開いて赤字の箇所を追記します。
<HintPath> には PdfTkNet6_70.dll のパスを記入します。

```
<Project Sdk="Microsoft.NET.Sdk">

  <PropertyGroup>
    <OutputType>Exe</OutputType>
    <TargetFramework>net6.0</TargetFramework>
    <ImplicitUsings>enable</ImplicitUsings>
    <Platform>x64</Platform>
    <Nullable>enable</Nullable>
  </PropertyGroup>

  <ItemGroup>
    <Reference Include="PdfTkNet6_70">
      <HintPath>/usr/AHPDFToolV7-lib/bin/PdfTkNet6_70.dll</HintPath>
    </Reference>
  </ItemGroup>

</Project>
```

(4) 「ファイル」メニューの「ファイルを開く」を選択して使用したいサンプルコードの cs ファイルを開きます。

(5) 「ファイル」メニューの「名前を付けて保存」を選択してプロジェクト生成時に作成された「Program.cs」に上書き保存します。または、「ファイル」メニューの「名前を付けて保存」を選択して別名保存を行い、その後「Program.cs」を選択し右クリックメニューで「削除」します。

(6) ターミナルに下記のコマンドを入力しビルドを行います。

```
dotnet build -c Release
```

(7) ビルドが完了したら、下記のコマンドで実行します。

```
dotnet run
```

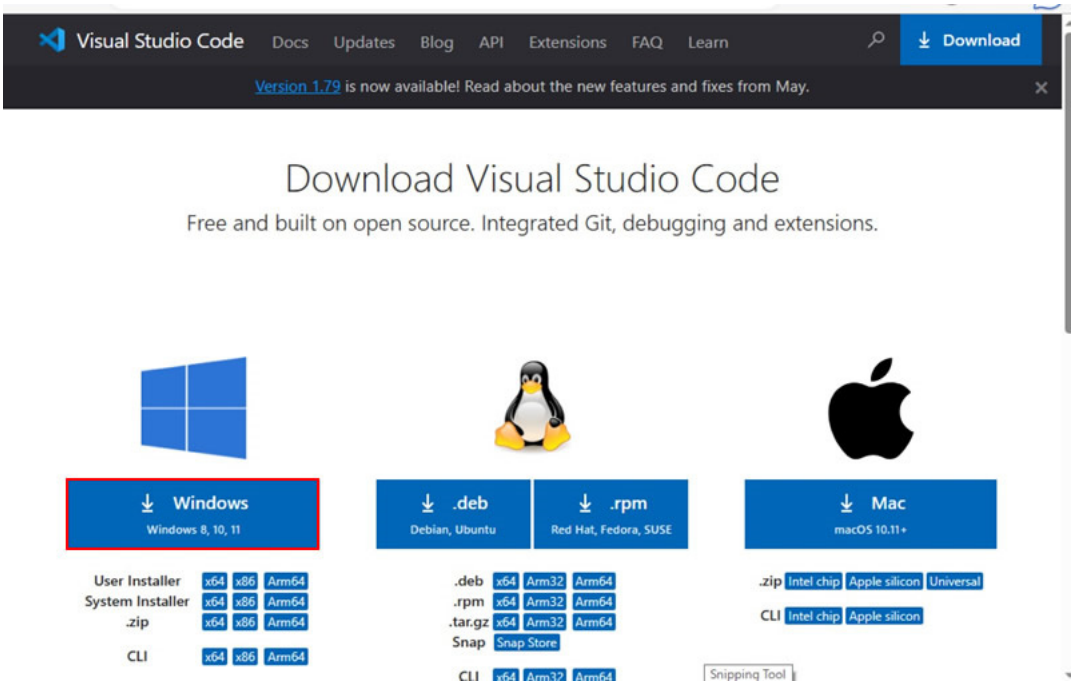
5.2. Windows 環境

5.2.1. NET SDK のインストール

.NET6.0 のウンロード
https://dotnet.microsoft.com/ja-jp/download/dotnet/6.0
「SDK 6.0.xx」の項にある「Windows」の「x64」または「x86」インストーラーをダウンロードしてセットアップします。

5.2.2. Visual Studio Code のインストール

(1) Download Visual Studio Code
https://code.visualstudio.com/download
を開いて「windows」をダウンロードします。



(2) ダウンロードした exe ファイルをクリックして開くとインストールが始まります。

(3) VSCode の拡張機能「Japanese Language Pack for VS Code」と「C#」をインストールします。「5.1.2. Visual Studio Code のインストール」の(3)、(4)をご参照ください。

5.2.3. プロジェクトの作成

「5.1.3.プロジェクトの作成」をご参照ください。

5.2.4. PDF Tool API のサンプルコードを使用したプログラムの作成

「5.1.4.PDF Tool API のサンプルコードを使用したプログラムの作成」をご参照ください。

履歴

日付	更新内容
2023.5.29	・初版
2023.7.4	・「5.Visual Studio Code でのプログラム作成と実行方法」を追加しました。
2024.10.31	・「3.サンプルコードのプロジェクト作成とビルド方法」の説明文を修正しました。

Antenna House PDF Tool API V7.0

サンプルコードのビルドと実行手順(.NET6)

2024.10.31