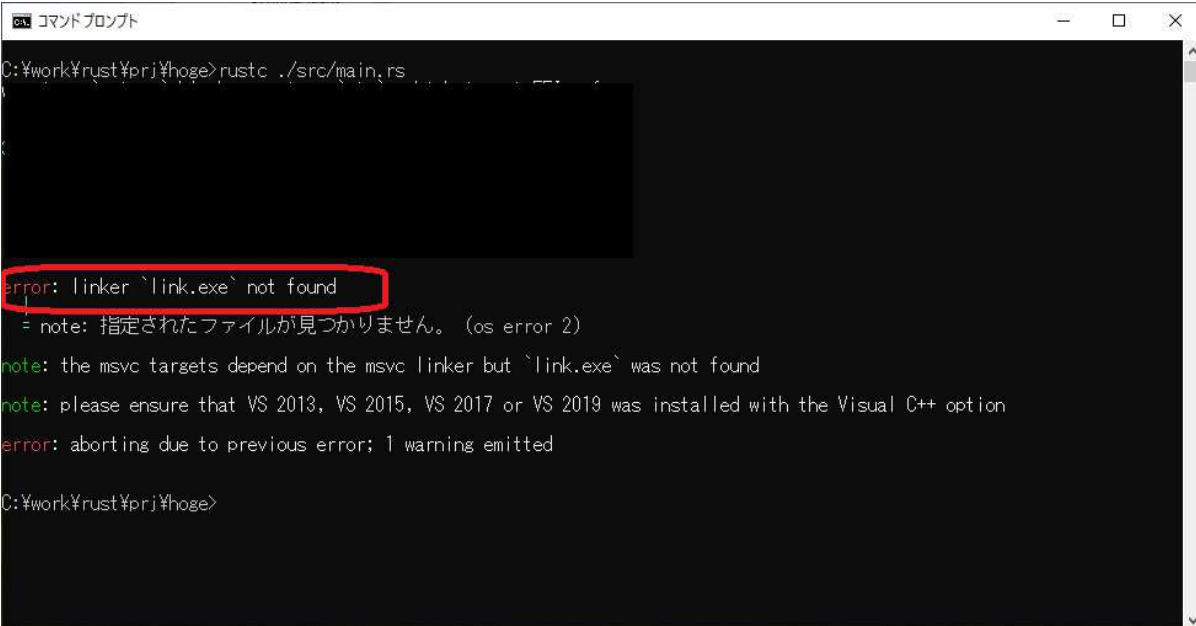


Qzlogs C Build Tools インストールマニュアル

2021/02/10 V1.0 QuiZ Lab.

Rust言語で Qzlogsを利用するには マイクロソフト社の Visual Studio Build Tools 2019が必要です

これが必要な理由 Qzlogsが提供する .dll, .libファイルとのリンク処理が必須です
インストールしないでQzlogsのサンプルなどを実行しようとコンパイルすると以下のようなエラーになります



```
コマンドプロンプト
C:\work\rust\prj\hoge>rustc ./src/main.rs
error: linker `link.exe` not found
  = note: 指定されたファイルが見つかりません。 (os error 2)
note: the msvc targets depend on the msvc linker but `link.exe` was not found
note: please ensure that VS 2013, VS 2015, VS 2017 or VS 2019 was installed with the Visual C++ option
error: aborting due to previous error; 1 warning emitted

C:\work\rust\prj\hoge>
```

1. 入手とインストール方法

<https://visualstudio.microsoft.com/ja/downloads/>

リンク先に移動し

2. スクロールし、Visual Studio 2019のツールを展開

Visual Studio 2019 for Windows

すべてのダウンロードを検索する

すべて展開 すべて折りたたむ

> Visual Studio 2019

Visual Studio 2019 のツール

Remote Tools for Visual Studio 2019

Remote Tools for Visual Studio 2019 を使用すると、Visual Studio がインストールされていないコンピューターでアプリケーションのデプロイ、リモートデバッグ、リモートテスト、パフォーマンスのプロファイル作成、単体テストを実行できます。

日本語 (Japanese) x64 ARM64 x86

ダウンロード ↓

Performance Tools for Visual Studio 2019

Visual Studio を使用せずにパフォーマンス プロファイルを実行できるようにするスタンドアロンのコマンドライン ツールです。詳細については、「[コマンドラインからのプロファイリング ツールの使用](#)」を参照してください。

日本語 (Japanese) x64 x86

ダウンロード ↓

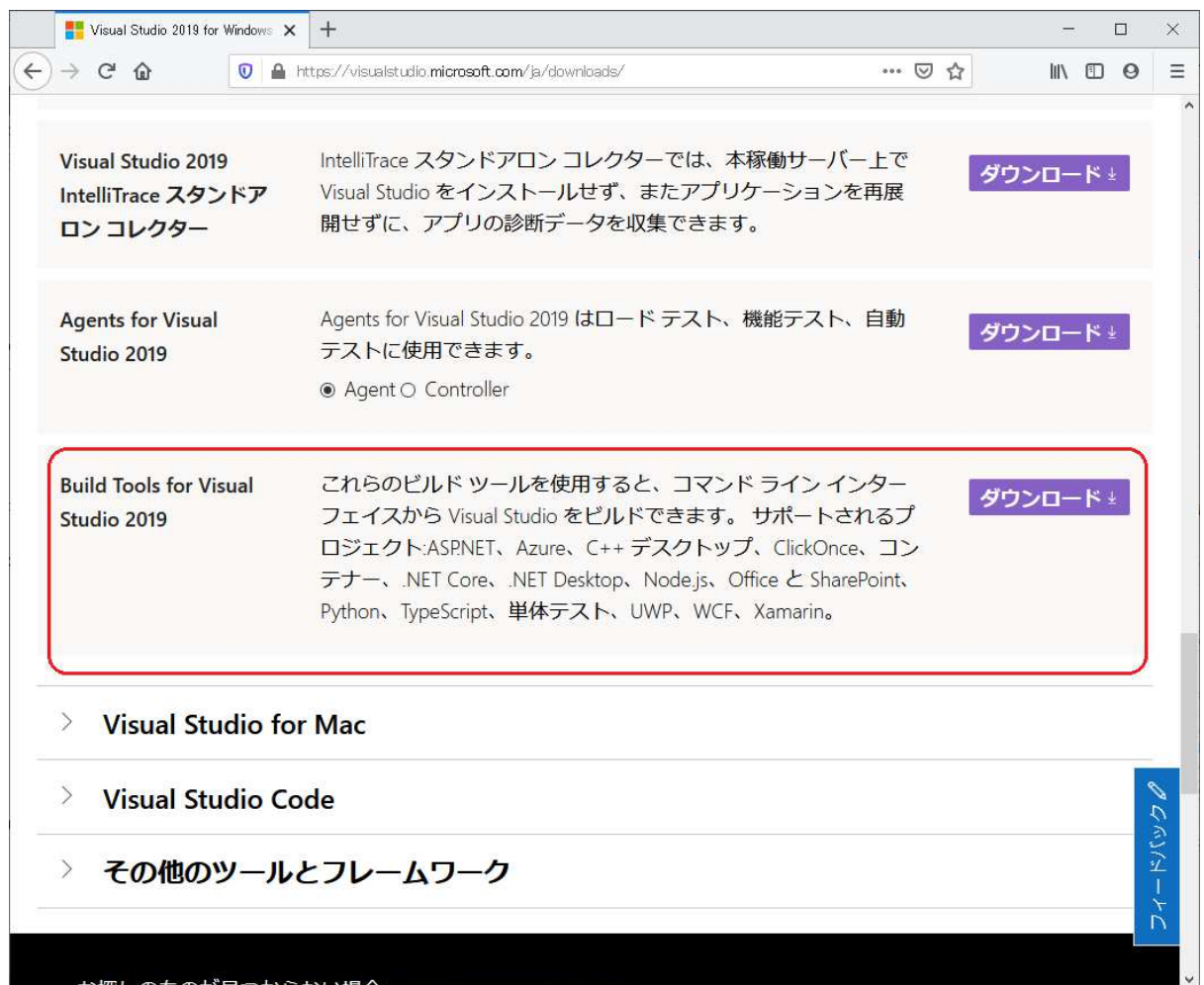
Visual Studio 2019

IntelliTrace スタンドアロン コレクターでは、本稼働サーバー上で

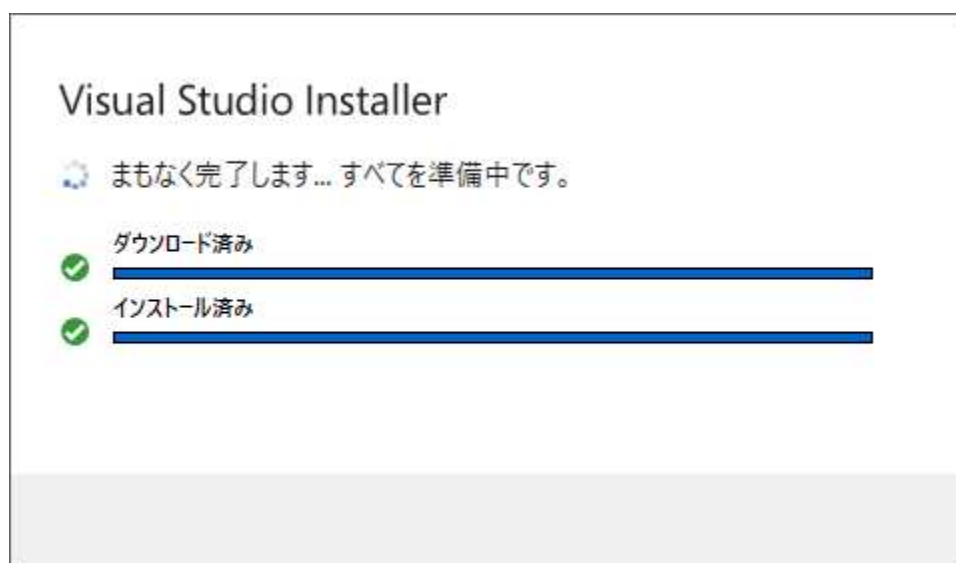
ダウンロード ↓

フィードバック

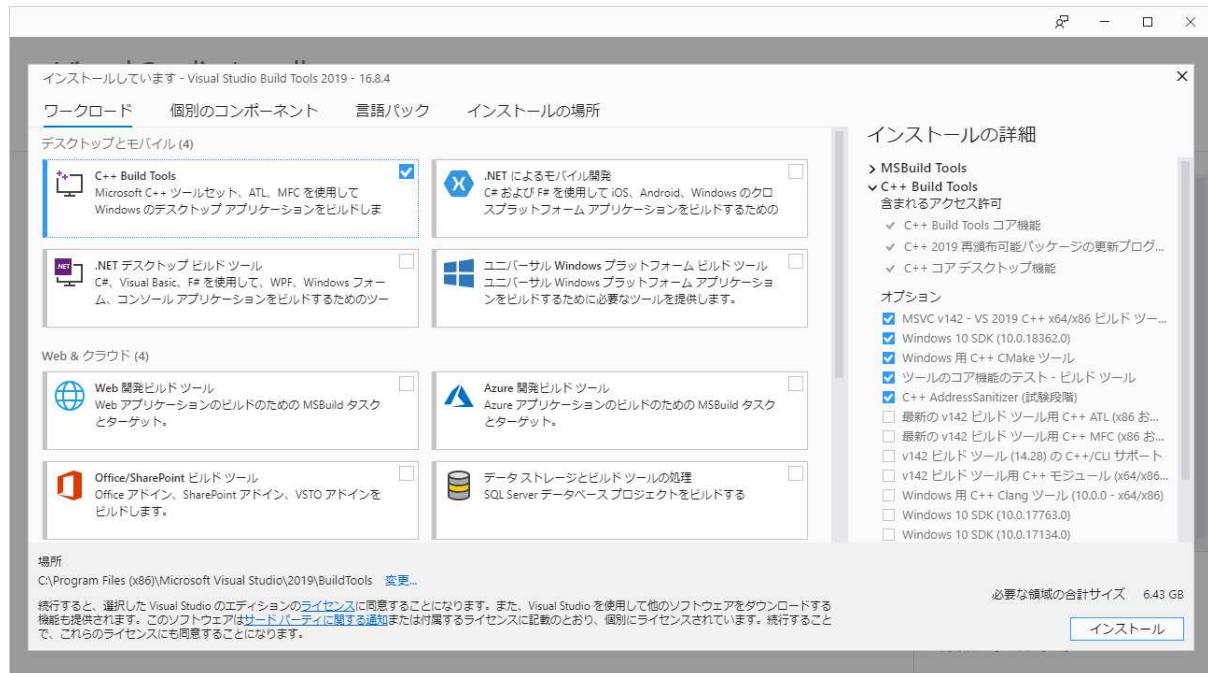
3. 展開をスクロールし、Build Tools for Visual Studio 2019のダウンロードボタンを押す



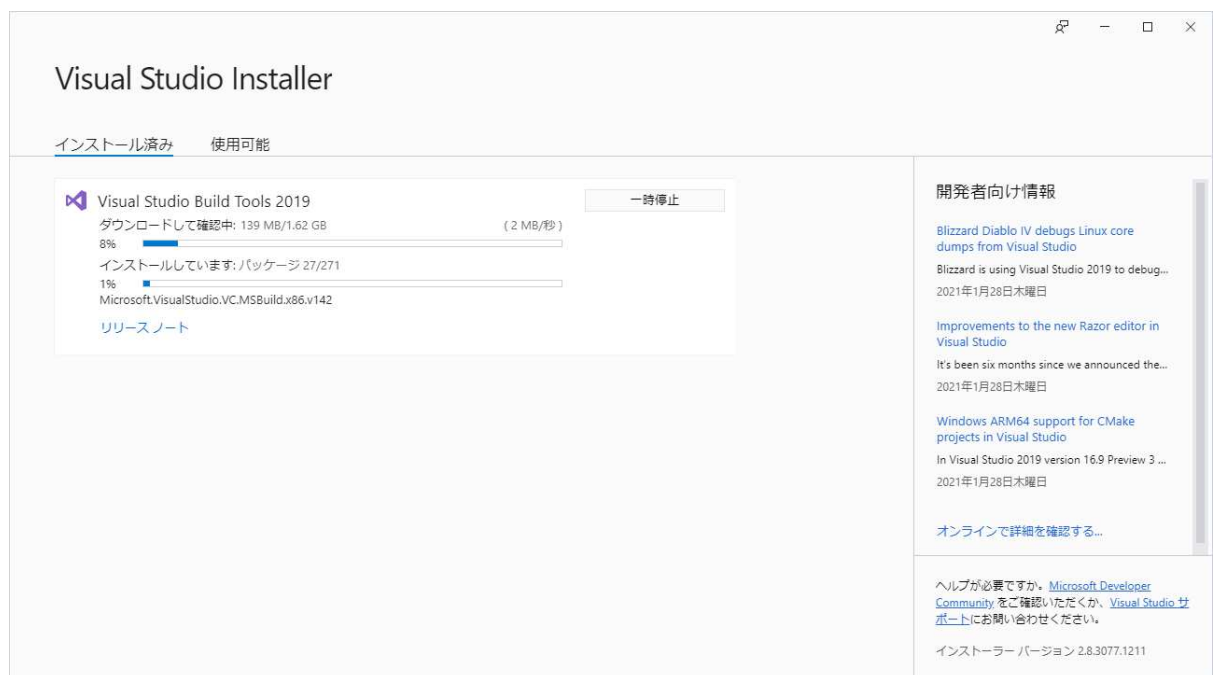
4. インストーラがダウンロードされるまで待ち、ダウンロードされたsetupプログラムをクリックする。このプログラムのインストールを許可する



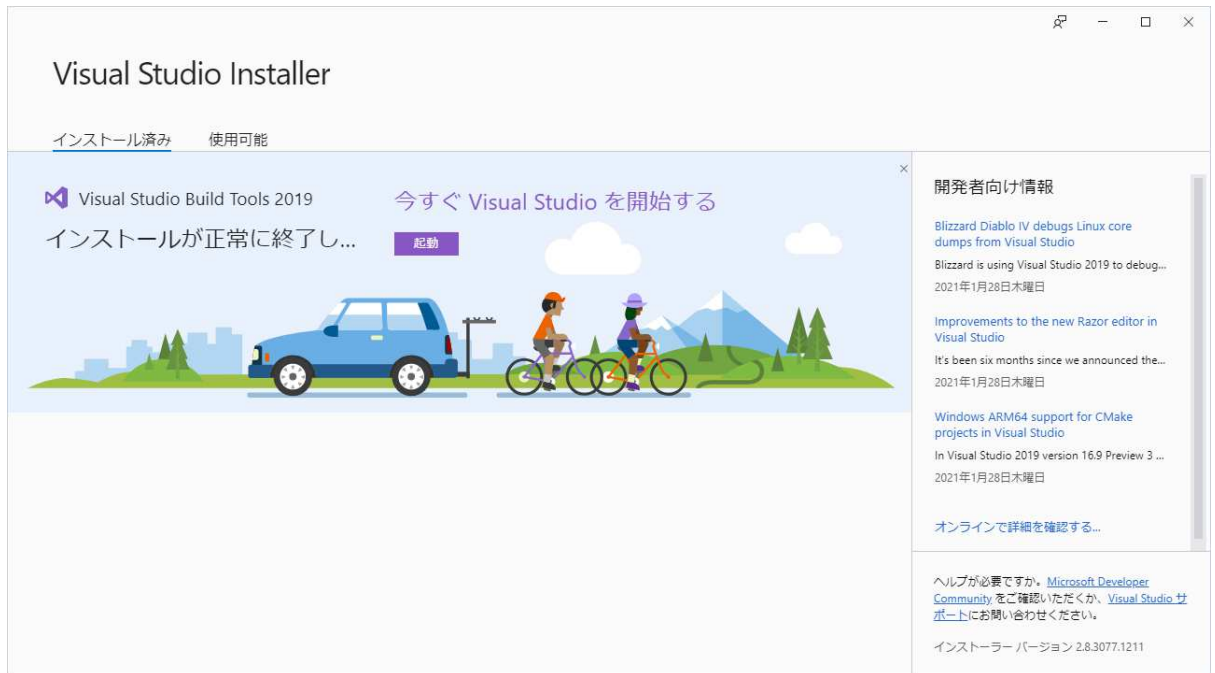
5. C++ Build Toolsにチェックをし、右下の「インストール」を押す



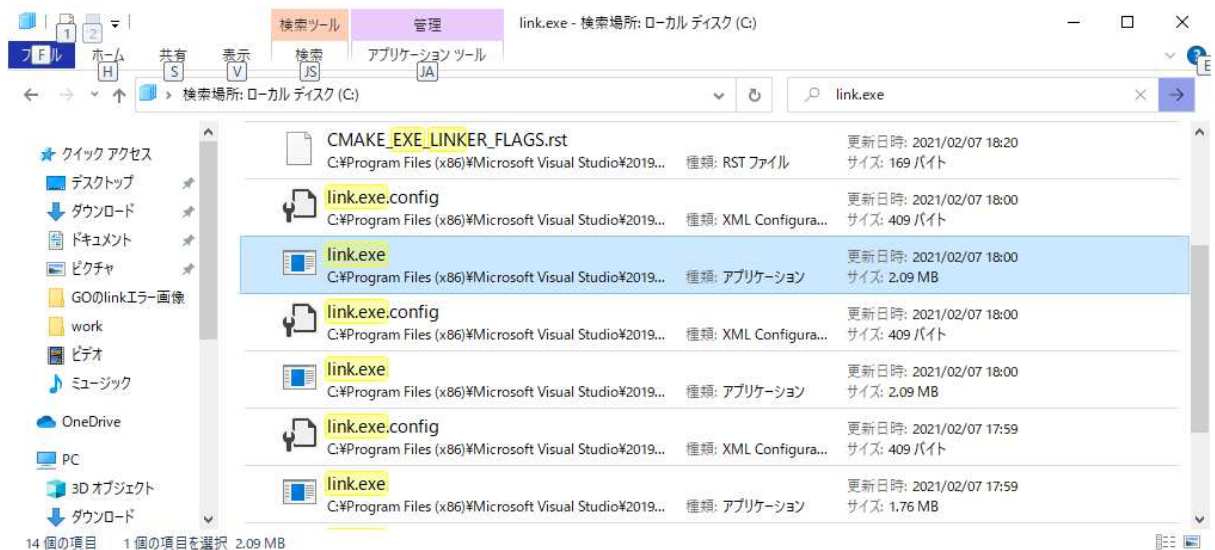
6. インストールが終了するまで待つ



完了画面

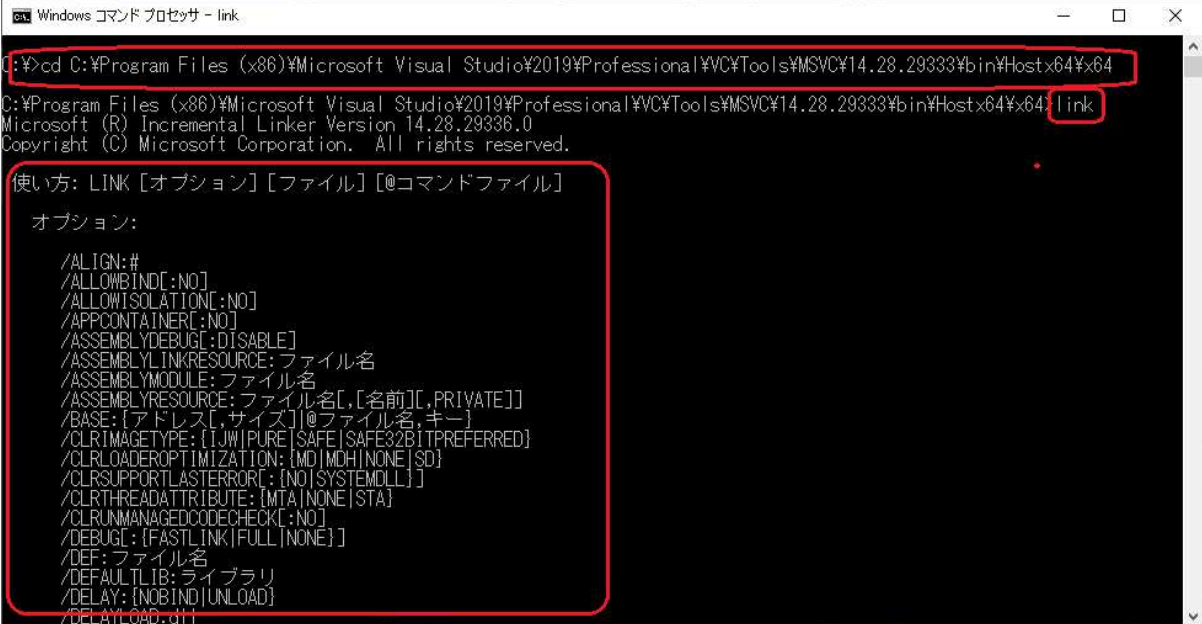


どこにインストールされたか気になる方は C ドライブを link.exe で検索するといいです。複数の link.exe がありますが、x64, x86 用などの専用品が複数個インストールされます。Rust 用の Qzlogs は x64 の link.exe が対象となります。



7. コマンド・プロンプトを起動する

link.exeが存在すれば、QzlogsのRust版のコンパイルが通るようになります
気になる方は、見つかった場所にcmd.exeで移動して、パスをlink.exeのある場所で
linkとコマンドを打ち込めば、使い方が出ます。これが出ればOK!



```
Windows コマンド プロセッサ - link
C:\Program Files (x86)\Microsoft Visual Studio\2019\Professional\VC\Tools\MSVC\14.28.29333\bin\Hostx64\x64
C:\Program Files (x86)\Microsoft Visual Studio\2019\Professional\VC\Tools\MSVC\14.28.29333\bin\Hostx64\x64>link
Microsoft (R) Incremental Linker Version 14.28.29336.0
Copyright (C) Microsoft Corporation. All rights reserved.

使い方: LINK [オプション] [ファイル] [@コマンドファイル]

オプション:
/ALIGN:#
/ALLOWBIND[:NO]
/ALLOWISOLATION[:NO]
/APPCONTAINER[:NO]
/ASSEMBLYDEBUG[:DISABLE]
/ASSEMBLYLINKRESOURCE:ファイル名
/ASSEMBLYMODULE:ファイル名
/ASSEMBLYRESOURCE:ファイル名[,名前[,PRIVATE]]
/BASE:[アドレス[,サイズ]]@ファイル名,キー
/CLRIMAGETYPE:[I]W[PURE][SAFE][SAFE32BITPREFERRED]
/CLRLOADEROPTIMIZATION:[MD][MDH][NONE][SD]
/CLRSUPPORTLATERERROR[:[NO][SYSTEMDLL]]
/CLRTHREADATTRIBUTE:[MTA][NONE][STA]
/CLRUNMANAGEDCODECHECK[:NO]
/DEBUG[:[FASTLINK][FULL][NONE]]
/DEF:ファイル名
/DEFAULTLIB:ライブラリ
/DELAY:[NOBIND][UNLOAD]
/DELAYLOAD:dll
```

8. cmd.exeを起動した直後でlink.exeを起動しようとしても、パスが通ってないので起動ません。
9. Rustは環境パスを設定しなくとも rustc コマンドがリンカのパスを通してくれるようです
10. あとは、Qzlogsのマニュアルを参照し、サンプルプログラムの環境作成と、ビルド、実行をお試しください
11. Visual Sudio 2017, 2019をインストールされている方でも、C/C++のツールをインストールしていない場合は、C/C++のビルドツールを追加してください

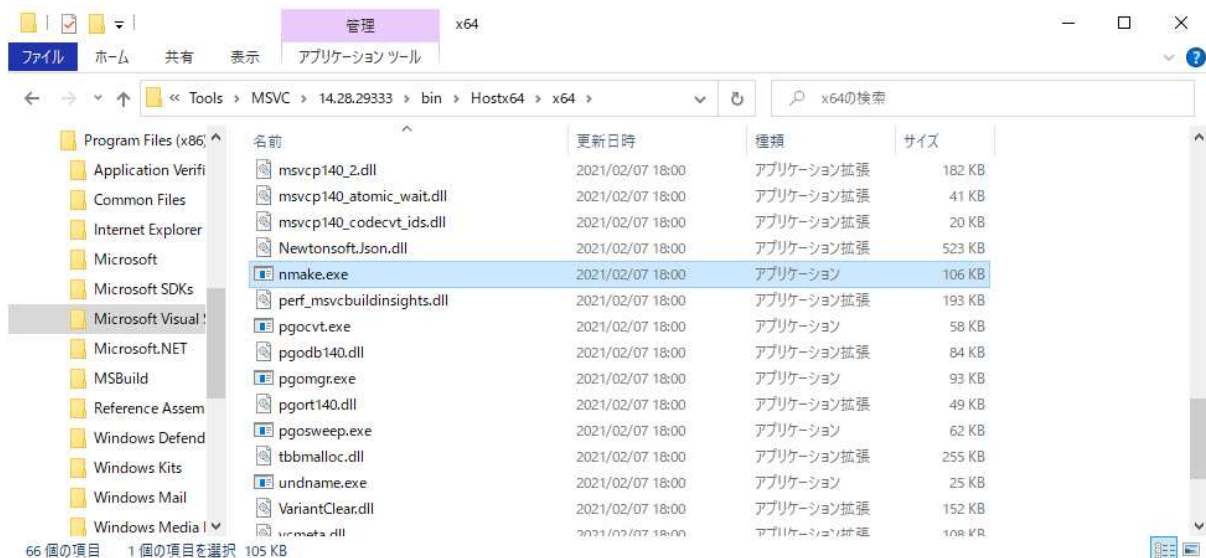
12. ビルドツール nmakeのインストール

Build Tools for Visual Studio 2019 をインストールすると、cl.exe のCコンパイラ、GNU makeとはすこし仕様は異なりますが、マイクロソフト版makeユーティリティの **nmake.exe**が利用できるようになります。ただ、DOS窓 cmd.exeを起動しただけでは、nmake.exeに対するパスが通っていませんので、DOS窓から nmake [Enter]としてもうんともすんともいけません。

nmake.exeを利用できるようにすると、サンプル・プログラムを楽に作成することが可能になります。

C:ドライブを選択して、以下の位置までエクスプローラで掘り下げます。

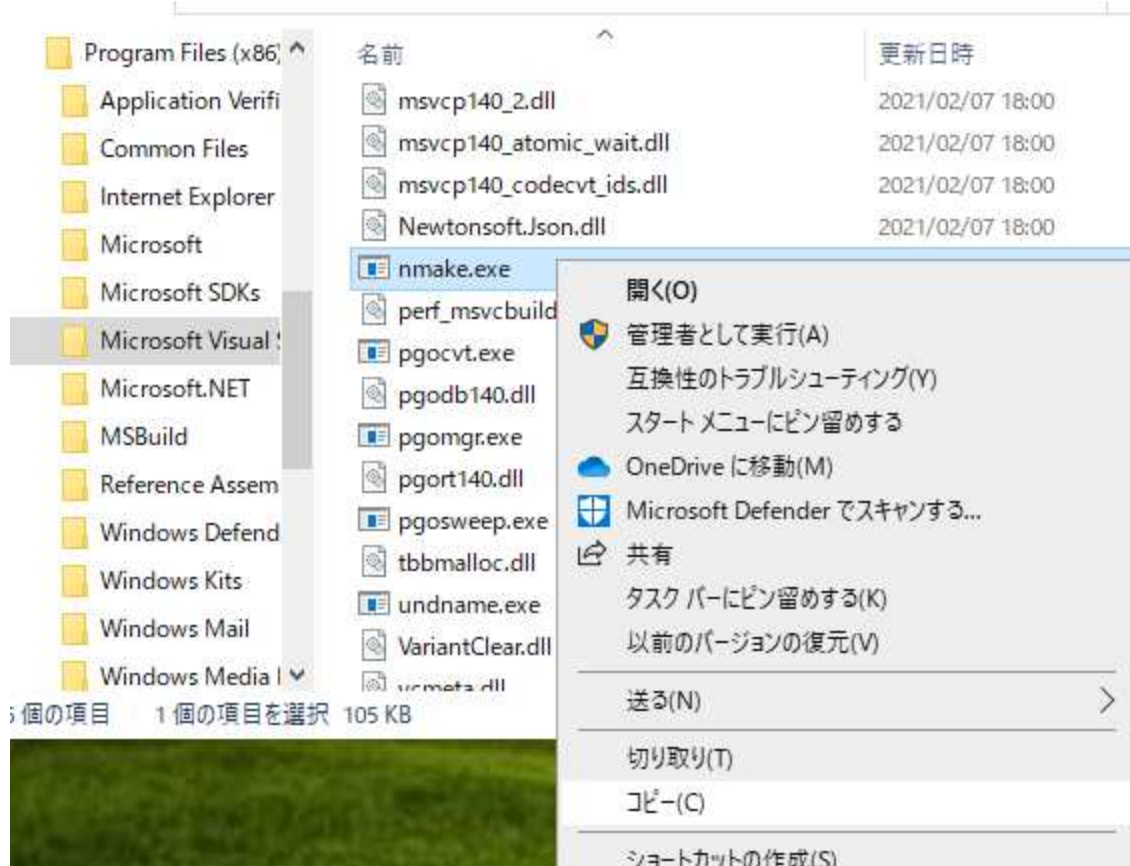
C:\Program Files (x86)\Microsoft Visual
Studio\2019\BuildTools\VC\Tools\MSVC\14.28.29333\bin\Hostx64\x64



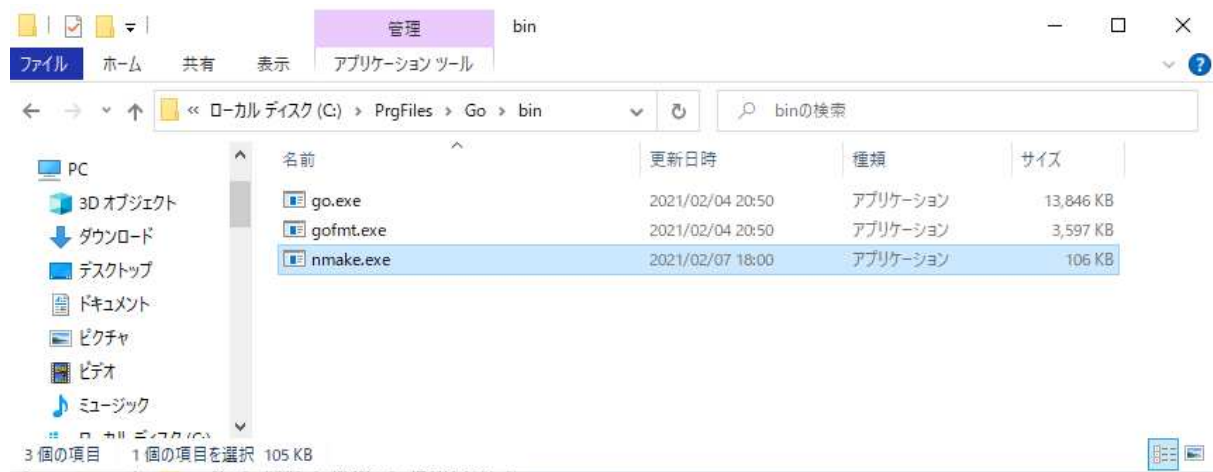
nmake.exeを発見しました。上記のパスをシステムの詳細設定-環境変数にセットすれば動作します。長いパスなので、めんどうだなと思う方は、Go, Rust のインストール先の bin フォルダに nmake.exe をコピーしてしまっても動作します。このほうが楽ですが、オリジナルの bin フォルダを汚したく無い方はパスを通してください。

13. Go にnmake.exeを設置する方法

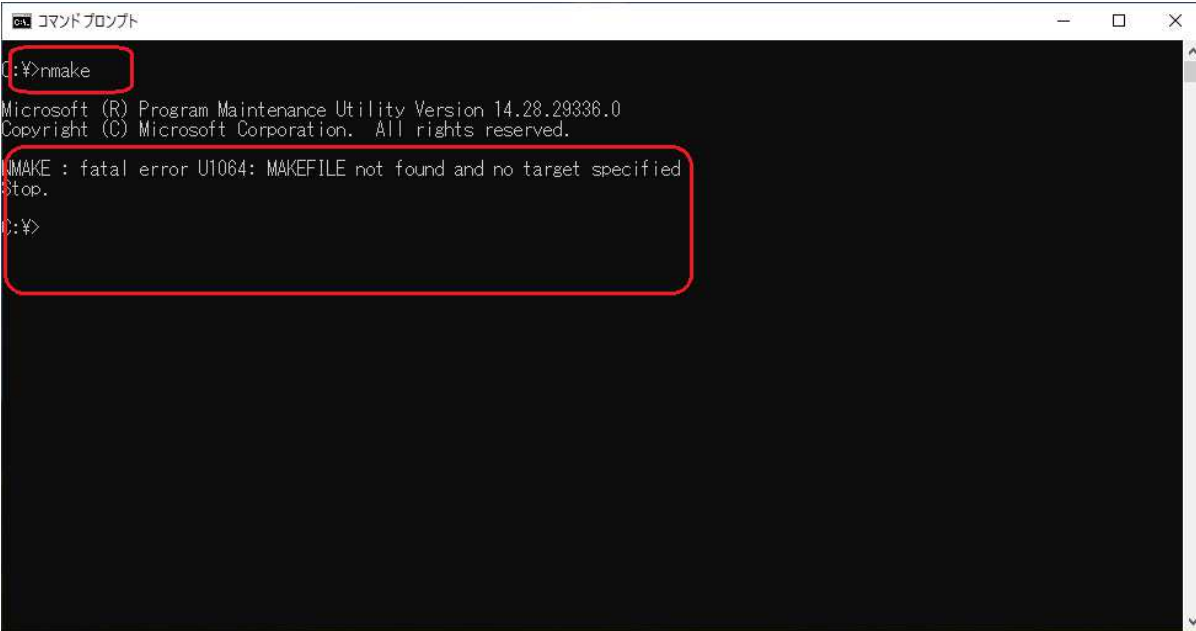
発見した、x64対応の nmake.exeをコピーします。



Goインストール先のbinに貼り付けます



コマンド・プロンプトを起動し、どこぞの場所でも構いませんので、nmake [Enter]します



```
コマンド プロンプト
C:\>nmake
Microsoft (R) Program Maintenance Utility Version 14.28.29336.0
Copyright (C) Microsoft Corporation. All rights reserved.

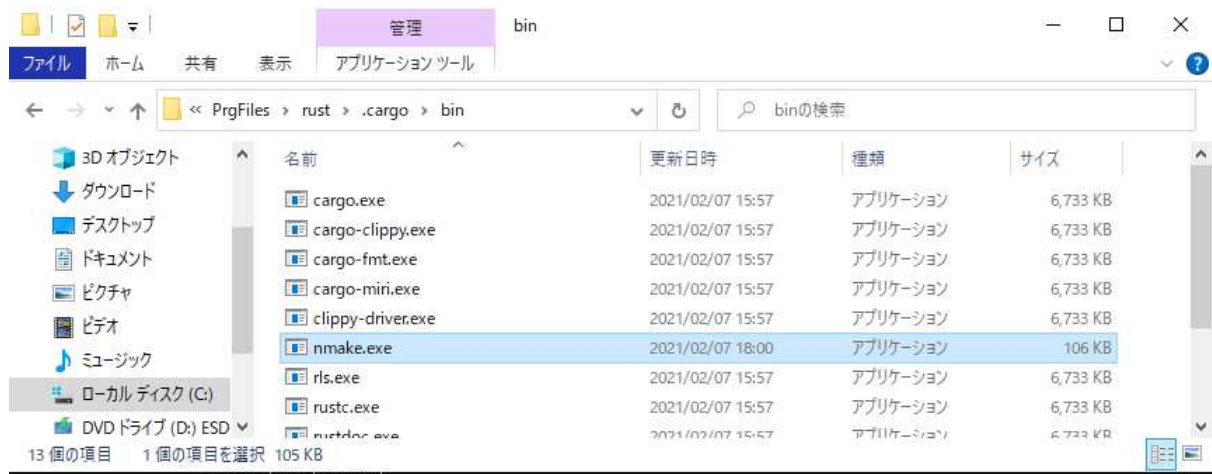
nmake : fatal error U1064: MAKEFILE not found and no target specified
Stop.
C:\>
```

入力ファイル(makefile)がないよと、nmake に文句を言われれば動作テスト完了です。

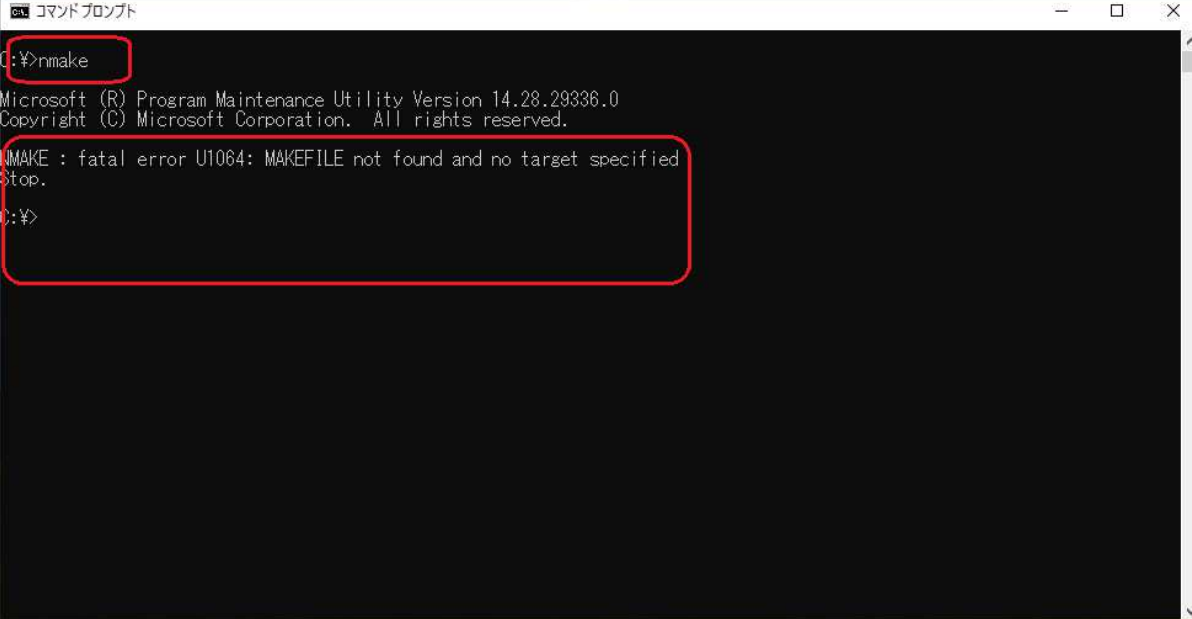
14. Rust にnmake.exeを配置する場合

13. Goのインストールのときと同じく、まずコピーします。

Rustのインストール先、.cargo\bin に貼り付けます。



Goと同様に、どこかの位置でコマンド・プロンプトを起動し、nmake [Enter]として以下の結果になればOKです。



```
コマンドプロンプト
C:\>nmake
Microsoft (R) Program Maintenance Utility Version 14.28.29336.0
Copyright (C) Microsoft Corporation. All rights reserved.

nmake : fatal error U1064: MAKEFILE not found and no target specified
Stop.
C:\>
```