Intel Parallel Studio XE2018

TRNSYS18 コンポーネントサンプル

このファイルはIntel Parallel Studio XE2018を使用したTRNSYS18コンポーネントのサンプルです。

# 関連フォルダ構成

* 1. Githubのパス

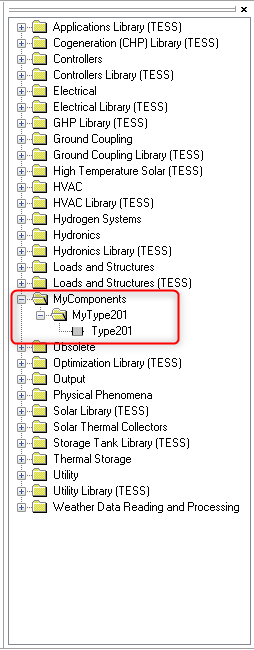
このサンプルでは以下のフォルダに関連するファイルを配置しています。

* TRNSYS.JP/TRNSYS18/Compilers/MyType201

ソースコードとビルド用のプロジェクト一式

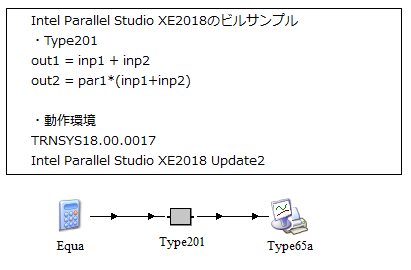
* TRNSYS.JP/TRNSYS18/Studio/Proformas/MyComponents/MyType201

プロフォルマ(\*.tmf)



* TRNSYS.JP/TRNSYS18/ MyProjects/MyType201Project

テスト用プロジェクト



* 1. TRNSYS18のパス

TRNSYS18で試す場合は「TRNSYS.JP/TRNSYS18」を「C:\TRNSYS18」へ置き換えて、下の表のように配置してください。

|  |
| --- |
| フォルダ |
| C:\TRNSYS18\Compilers\MyType201 |
| C:\TRNSYS18\Studio\Proformas\MyComponents\MyType201 |
| C:\TRNSYS18\Examples\MyType201Project |

# 動作環境

以下の環境で動作を確認しています。

Windows10 Pro(64bit, 1803)

Intel Parallel Studio XE2018 Update2

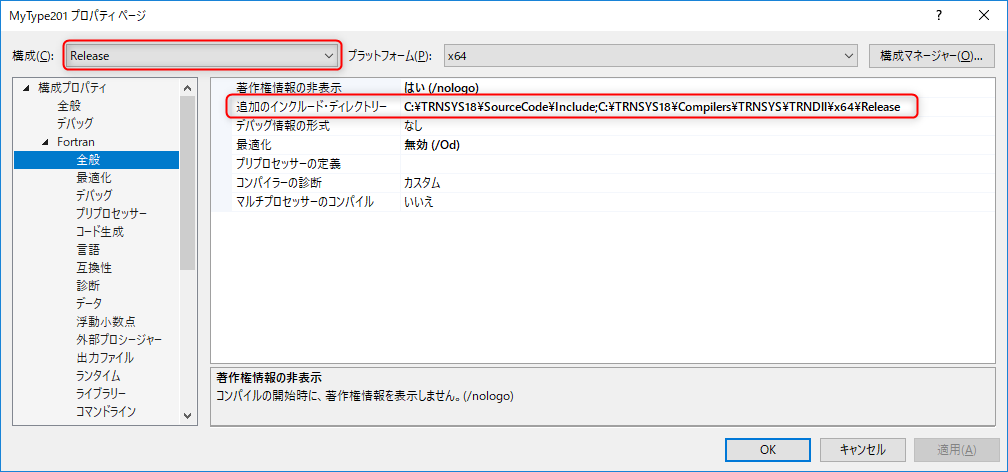
Visual Studio 2017(ver15.7.1)

TRNSYS18.00.0017(64bit)

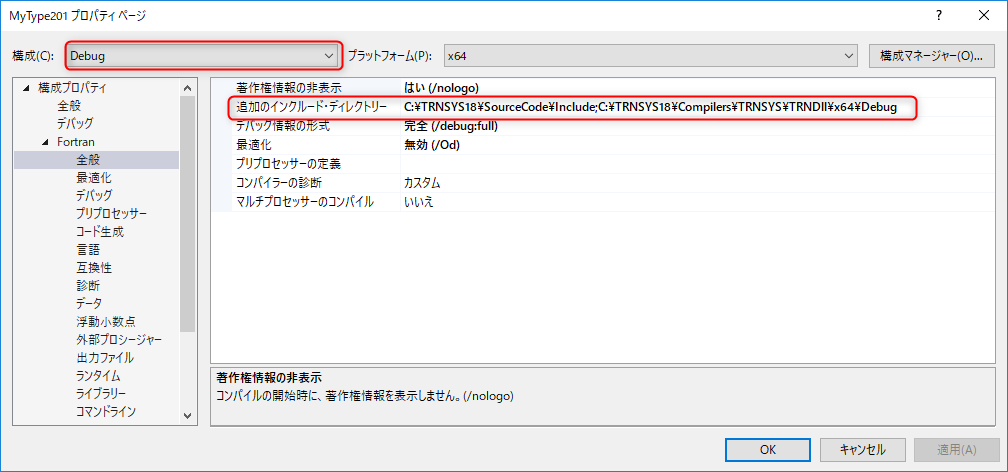
# プロジェクト設定

Fortran,リンカーの主な設定項目を以下に示します。以下、Release、Debugで設定が異なる項目については併記しています。

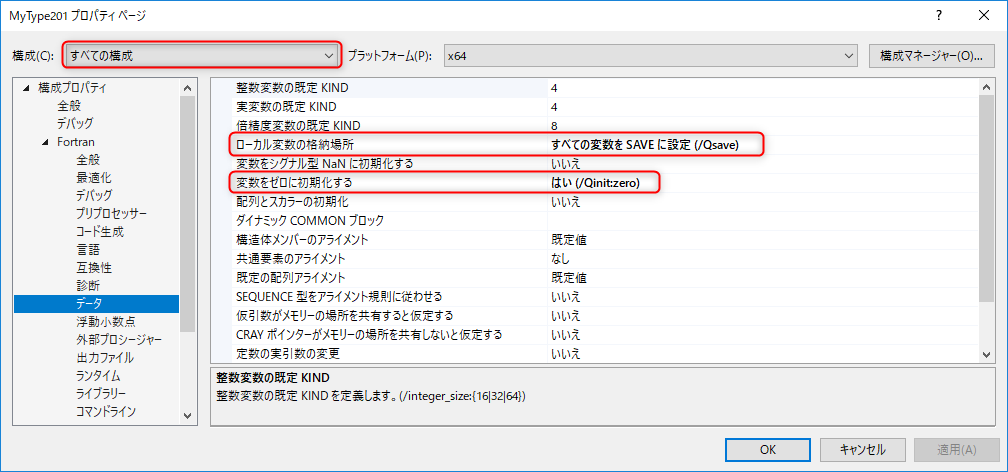
* 1. FORTRAN
     1. 追加のインクルード・ディレクトリー
* Release



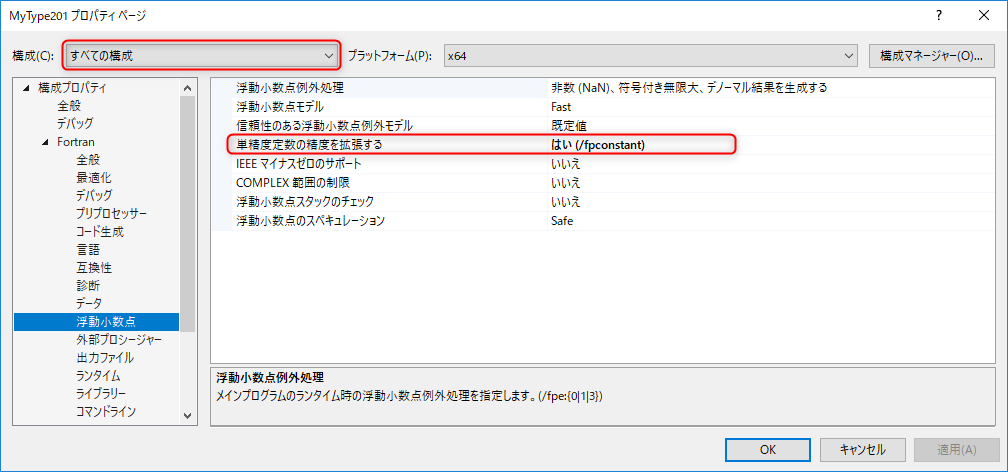
* Debug



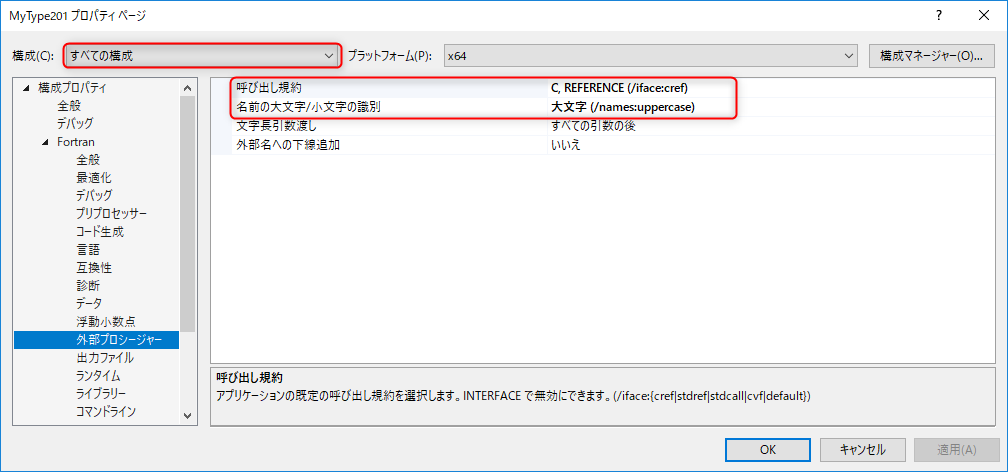
* + 1. データ
* すべての構成



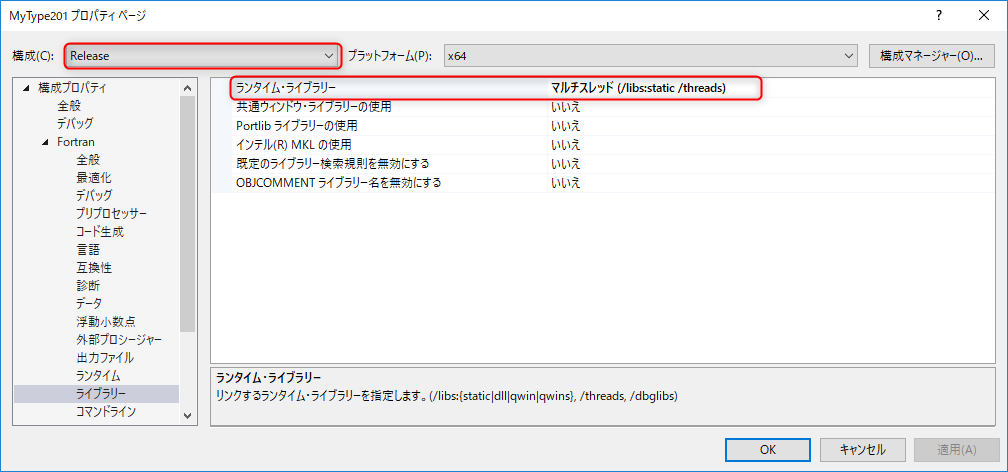
* + 1. 浮動小数点
* すべての構成



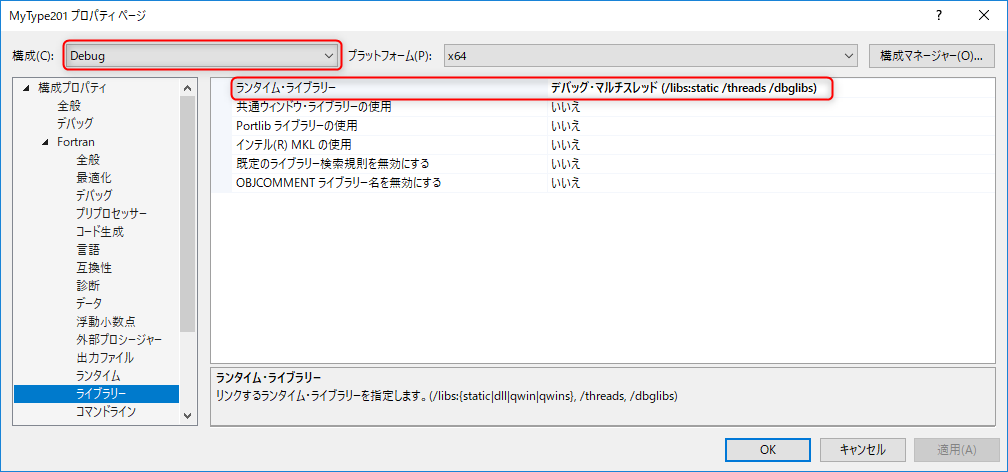
* + 1. 外部プロシージャ
* すべての構成



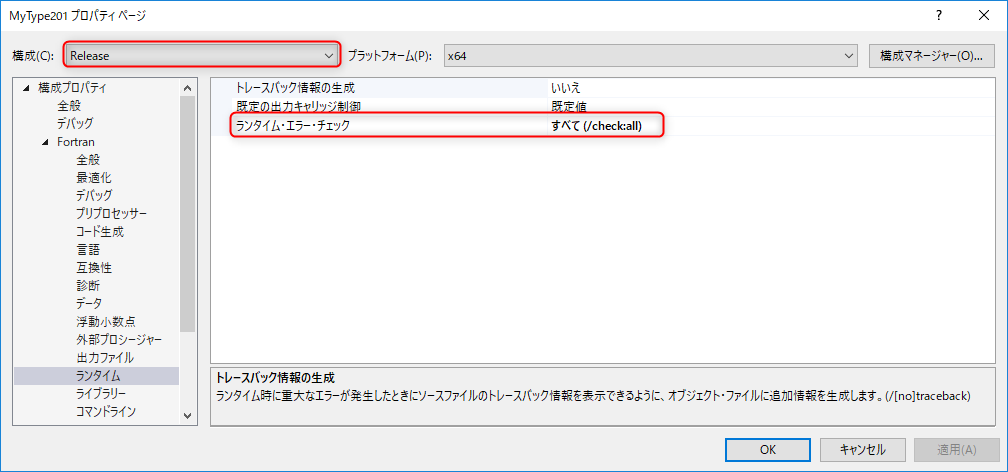
* + 1. ライブラリー
* Release



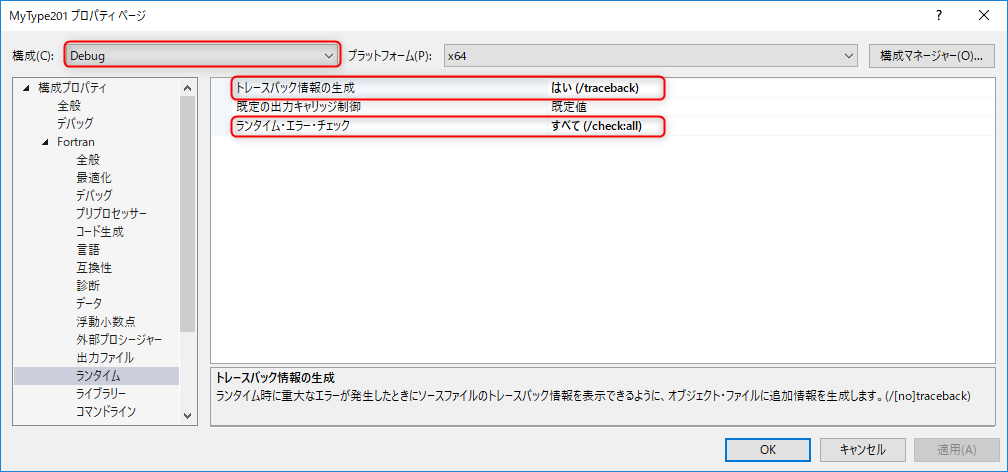
* Debug



* + 1. ランタイム
* Release



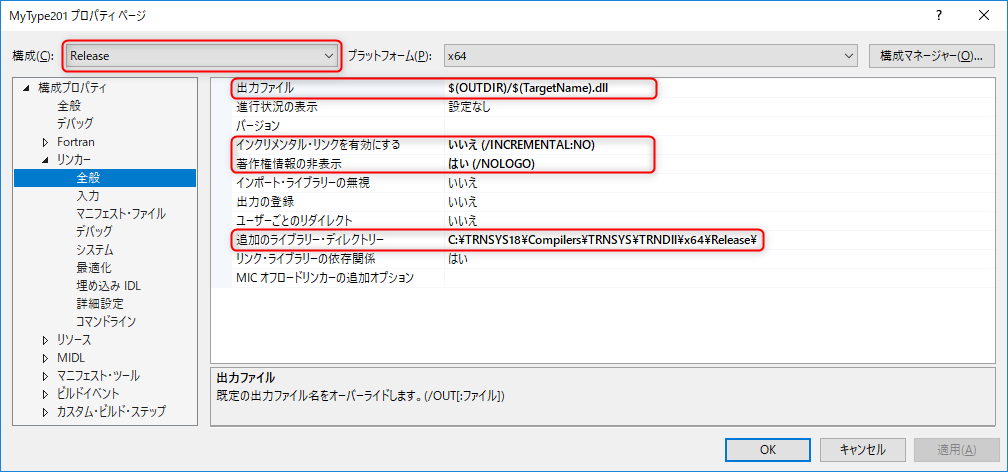
* Debug



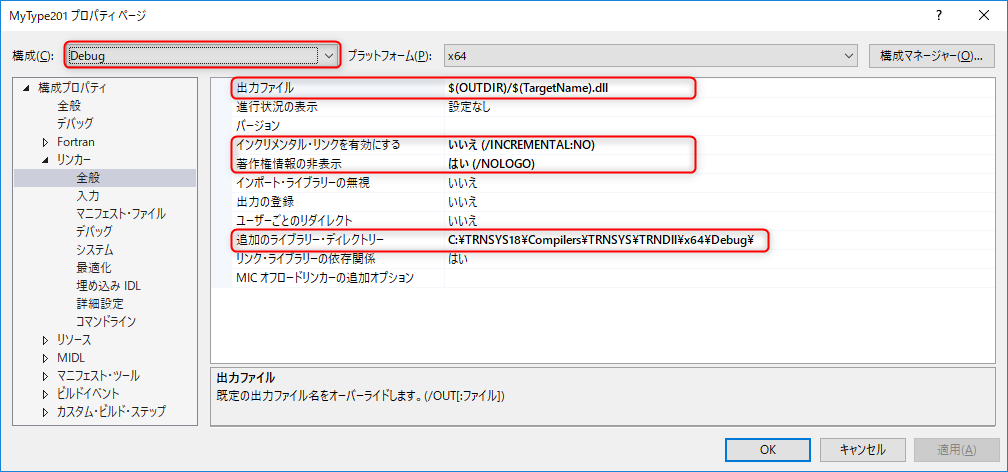
* 1. リンカー
     1. 全般

注）「追加のライブラリー・ディレクトリー」のみ設定が異なる。

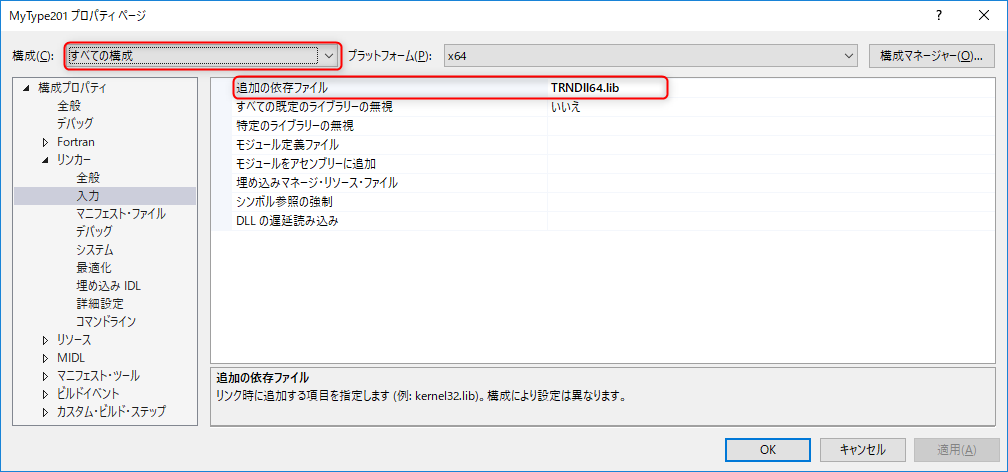
* Release



* Debug



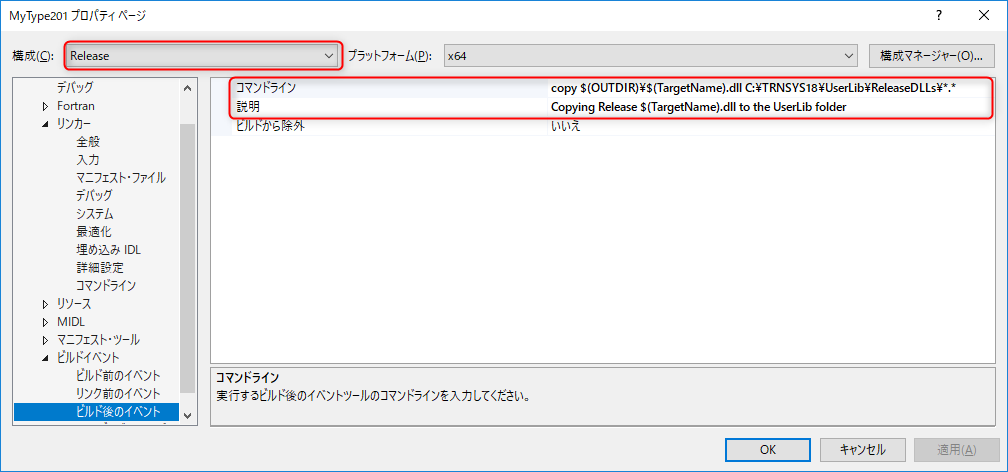
* + 1. 入力
* すべての構成



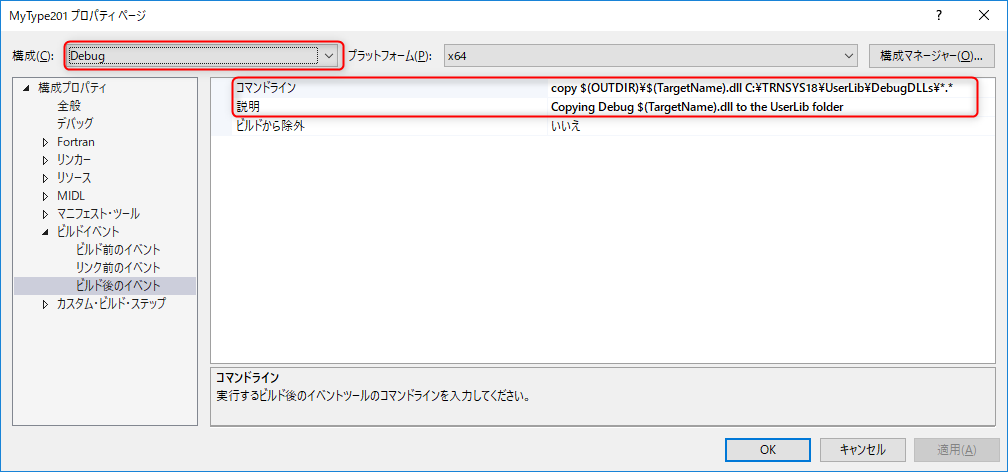
※「全般」「追加のライブラリー・ディレクトリー」でRelease,Debugで参照先が変わるように設定している。このためTRNDll64.libはビルドのモードに応じて適切なライブラリが参照されます。

注）TRNDLL64.libはあらかじめ"C:\TRNSYS18\Compilers\TRNSYS\TRNSYS.sln"を開いてRelease,Debugの両方のモードでビルドしておいてください。

* 1. ビルドイベント
     1. ビルド後のイベント
* Release



* Debug



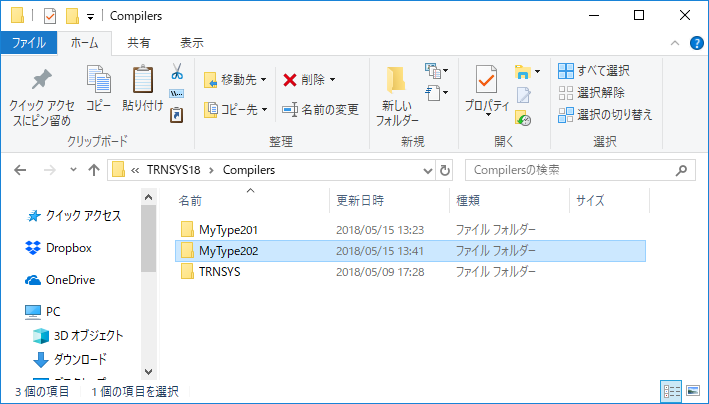
# 新しいTypeの作成

サンプルのプロジェクトを利用して新しいTypeの作成する場合は以下の手順で行います。この例ではサンプルプロジェクトを元に**Type202**を作成しています。

* 1. 作業フォルダの準備

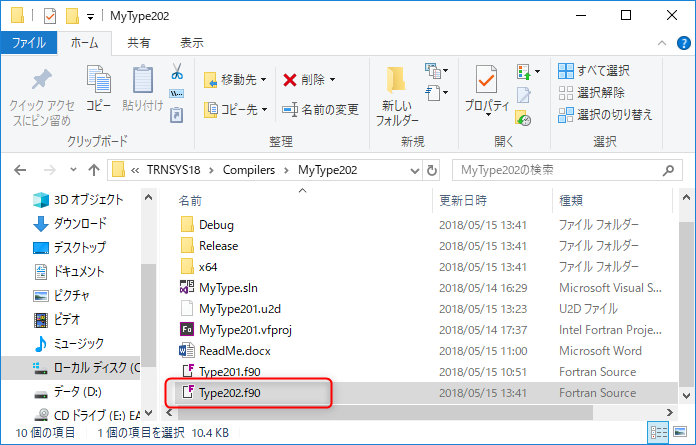
はじめにサンプルのプロジェクトをコピーして作業用のフォルダを用意します。

フォルダ（"C:\TRNSYS18\Compilers\MyType201"）をコピーして名前を**MyType202**へ変更します。



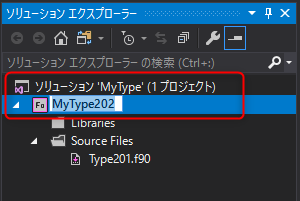
* 1. ソースコードの準備

Simulation Studioで新しいType(MyType202)のプロフォルマを作成、FORTRANのソースコード（Type202.f90）を同じフォルダへエクスポートして置きます。



* 1. プロジェクト名の変更

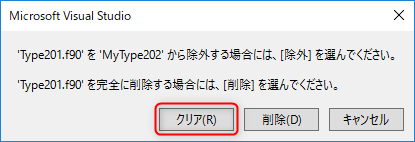
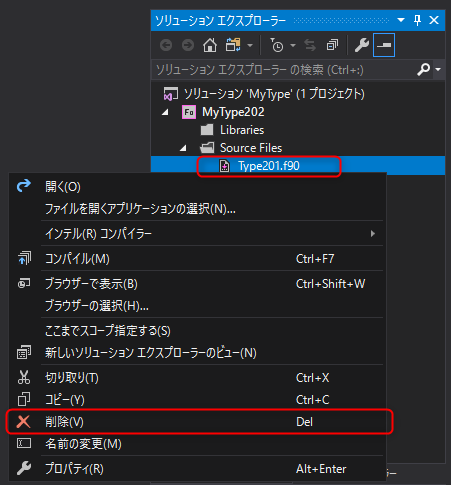
Visual Studio 2017を起動して、ソリューション（MyType.sln）を開き、プロジェクト名をMyType201から**MyType202**へ変更します。



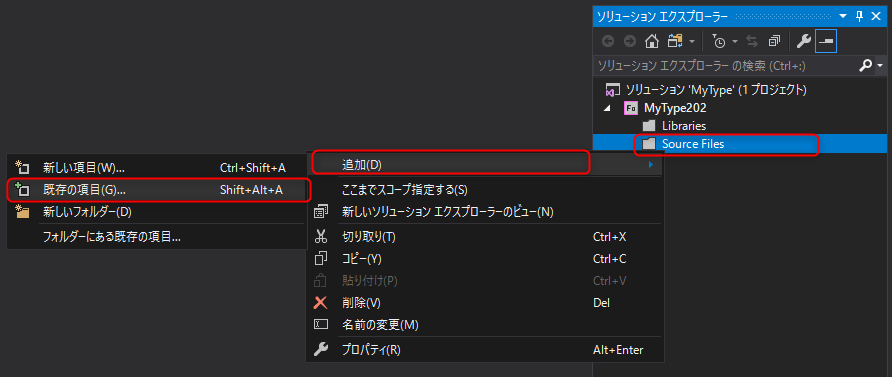
※このプロジェクト名の変更で、出力されるDLLの名前が**MyType202.dll**へ変わります。

* 1. ソースコードの差し替え

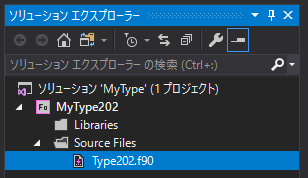
ソリューションエクスプローラーで既存の**ソースコード(Type201.f90)**をプロジェクトから**削除（クリア）**する。



**Source Files**フォルダを選んで、右クリックから**[追加]-[既存の項目]**で、追加した新しい**ソースコード(Type202.f90)**を追加する。



ソリューションエクスプローラーの表示が以下のようになっている事を確認する。



以上でプロジェクトの準備は終了です。Type202.f90へ処理内容を記述して、無事ビルドできれば完了です。