TRNSYS18

TypeStudioで作るTRNSYSコンポーネント

このドキュメントは以下のライセンスで提供されます。

[クリエイティブ・コモンズ・ライセンス](http://creativecommons.org/licenses/by-nc-sa/4.0/)

quattro corporate design Co., Ltd. 作『TRNSYS.JP Library』は[クリエイティブ・コモンズ 表示 - 非営利 - 継承 4.0 国際 ライセンス](http://creativecommons.org/licenses/by-nc-sa/4.0/)で提供されています。

内容

[1 はじめに 4](#_Toc26188962)

[1.1 TypeStudio登場！ 4](#_Toc26188963)

[2 Proforma（プロフォルマ）の作成 5](#_Toc26188964)

[2.1 そもそもProformaとは？ 5](#_Toc26188965)

[2.2 計算式の準備 6](#_Toc26188966)

[2.3 Proformaの新規作成 7](#_Toc26188967)

[3 計算機能の実装 12](#_Toc26188968)

[3.1 ソースコードのエクスポートと編集 12](#_Toc26188969)

[3.2 TypeStudioの起動とソースコードのインポート 14](#_Toc26188970)

[3.3 ソースコードの確認 15](#_Toc26188971)

[3.4 処理の実装 16](#_Toc26188972)

[3.4.1 変数の追加 16](#_Toc26188973)

[3.4.2 計算処理の追加 16](#_Toc26188974)

[3.4.3 出力処理の追加 17](#_Toc26188975)

[3.4.4 ビルド 18](#_Toc26188976)

[4 コンポーネントの動作検証 20](#_Toc26188977)

[4.1 プロジェクトを用意する 20](#_Toc26188978)

[4.2 Type201Heaterの接続 21](#_Toc26188979)

[4.3 結果を確認する 22](#_Toc26188980)

[4.4 計算結果の検証 22](#_Toc26188981)

[5 Outputの初期化について 24](#_Toc26188982)

[6 参考資料 25](#_Toc26188983)

[6.1 リポジトリ 25](#_Toc26188984)

[6.2 ブログ 25](#_Toc26188985)

このドキュメントではTRNSYS、TypeStudioの基本的な操作方法を説明しています。

以下の環境を前提として記載しています。

TRNSYS 18.01.0001 (64bit)

OS Windows10 Pro (64bit, 1903)

# はじめに

TRNSYSには標準的なコンポーネントが多数用意されています。シミュレーションはそれだけでも十分に行えますが、特殊な機器や研究開発中の装置など、標準には存在しない処理を組み入れたい場合にはオリジナルのコンポーネントが必要になることがあります。

従来からIntel FORTRANを使用した開発手段が提供されていました。Intel FORTRANは本格的なプログラミングが可能な非常に強力な開発ツールです。開発者向けに様々な機能が提供され、詳細な設定が可能です。しかし一方では研究開発で気軽にプログラムを書いて試す用途には少々ハードルが高いツールであるとも言えます。

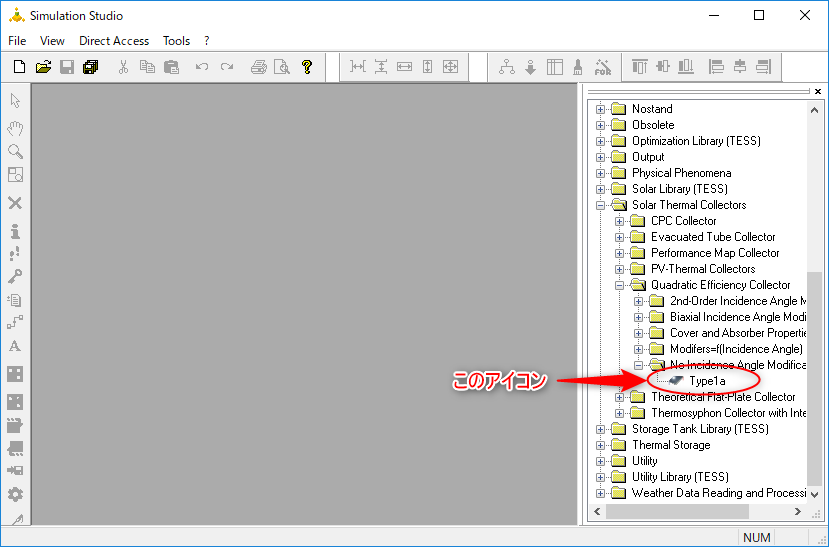
* 1. TypeStudio登場！

TRNSYS18では、コンポーネントの開発用の専用ツール、TypeStudioがリリースされました。こちらはTRNSYS専用のFORTRANコンパイラとエディタで構成されています。TRNSYSコンポーネントの開発に特化しているため、特に面倒な設定も不要です。気軽にコンポーネントの開発を始めることができます。

**[](file:///C:\Users\yasuda\AppData\Local\Temp\OpenLiveWriter1641855762\supfiles10EA2100\image8.png)**

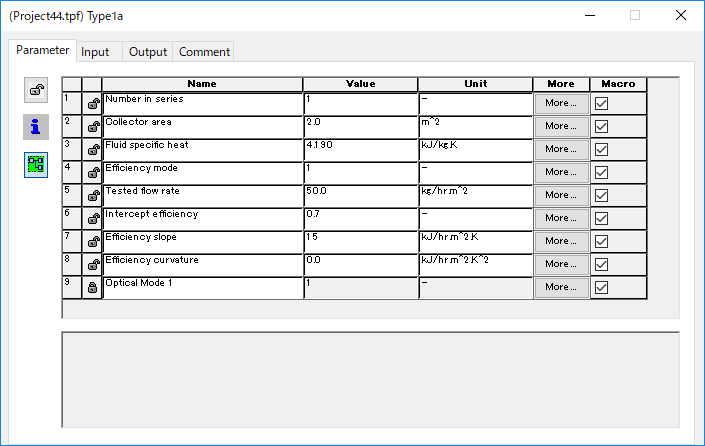
# Proforma（プロフォルマ）の作成

さて、コンポーネントを作成する前に**Proforma**を用意します。普段はあまり意識しないProformaですが、Simulation Studioでコンポーネントとして見えるアイコンの実体が**Proforma**です。



* 1. そもそもProformaとは？

Simulation Studioでコンポーネントの設定を行う際に図のような画面（Variables Window）が表示されます。この画面に表示される内容を定義しているのが**Proforma**ファイル（拡張子.TMF）です。

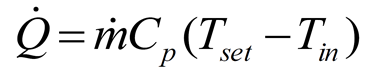


TRNSYSのコンポーネントの実体はFORTRANのサブルーチン（C言語だと関数）に相当します。

ProformaではSimulation Studioとサブルーチンのデータ（Parameters, Inputs, Outputs)の橋渡しルールを定義しています。

* 1. 計算式の準備

まずは、これから作成するコンポーネントの式を用意します。今回はドキュメントに載っているシンプルなヒーターの式を例にProformaを作成します。



Tset 設定温度  
Tin 入口温度  
ｍ 流量  
Cp 比熱  
Q 負荷

**※TRNSYS18, Vol.7 Developer’s / Programmer’s Guide,  p7-17, Eq.7.3.1-1 より**

さて、この式を元にコンポーネントのProformaを作っていくわけですが、はじめにそれぞれの値を**Parameter, Input, Output**に割り当てます。

**Q**は計算結果なのでOutputで問題なさそうです。**ｍ, Tset, Tin**は時系列で変化する値として**Input**で扱います。**Cp**は時系列で変化するなら**Input**で、変化しないのであれば固定値で**Parameter**とします。今回は変化しないものとして**Parameter**で扱うことにします。

値の割り当てを整理すると次の表のようになります。

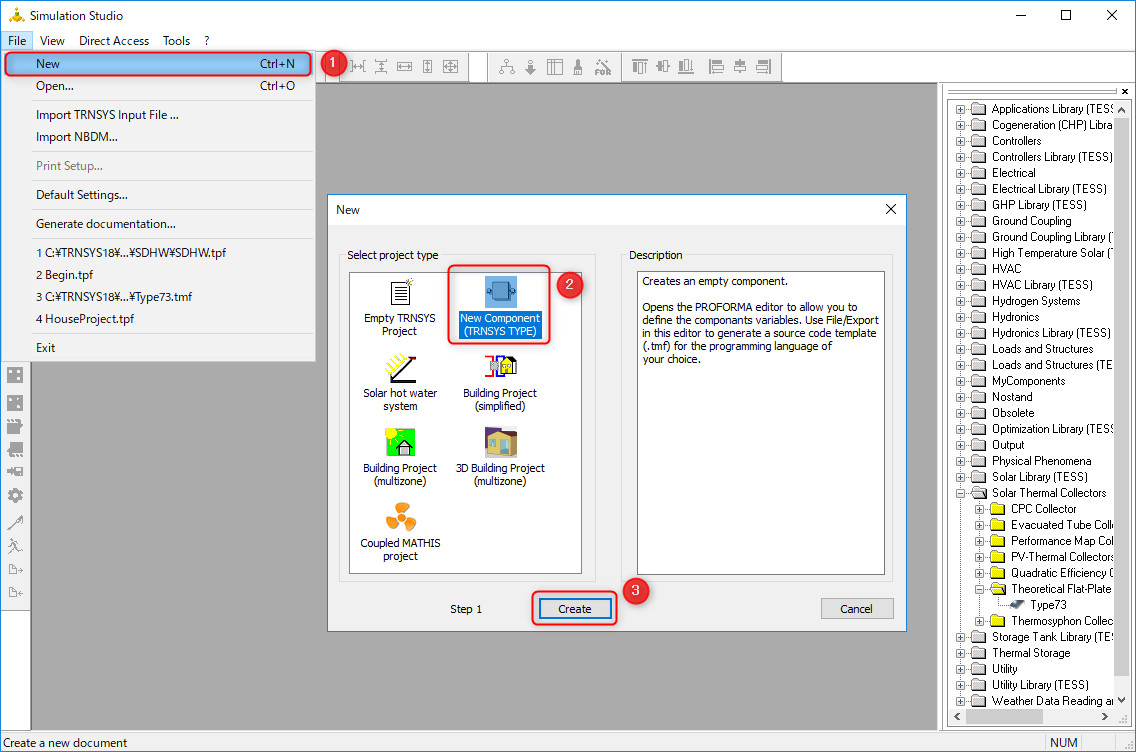
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Role | Dimension | Unit | Def. |
| m | input | Flow Rate | kg/hr | 100.0 |
| Tset | input | Temperature | C | 50.0 |
| Tin | input | Temperature | C | 20.0 |
| Cp | parameter | Specific Heat | kJ/kgK | 4.190 |
| Q | output | Power | kJ/hr | 0.0 |

※デフォルト値（Def.）はInputの初期値として使われるので適切な値に設定します。Cpはここでは水の比熱です。

* 1. Proformaの新規作成

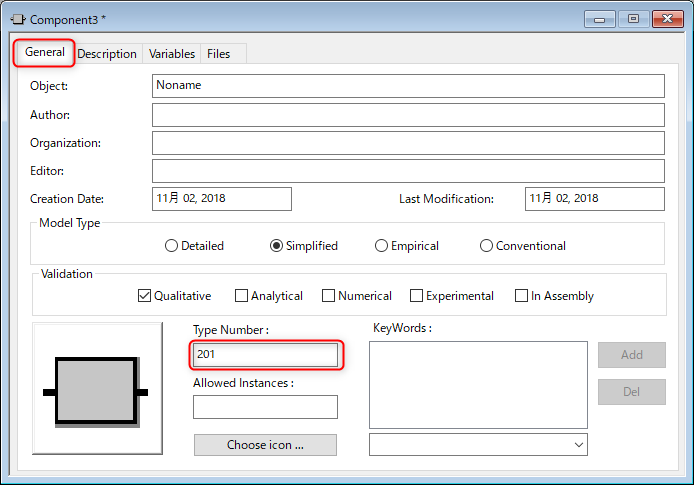
Simulation StudioでProformaを作成します。

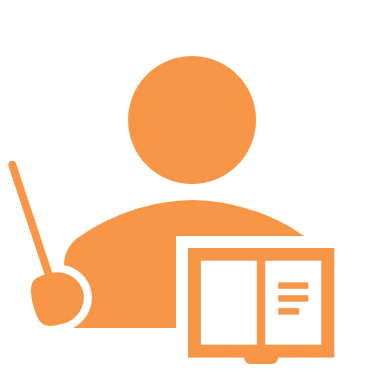
1. メニューから**[File]-[New]**を選択する。
2. 表示されるウィンドウで**「New Component」**を選択して、
3. **Createボタン**をクリックする。



Proformaのウィンドウが表示されるので、式に合せて設定を行います。

はじめに「**General**」タブのType Numberの項目にType番号を指定します。ここでは**201**を指定してください。



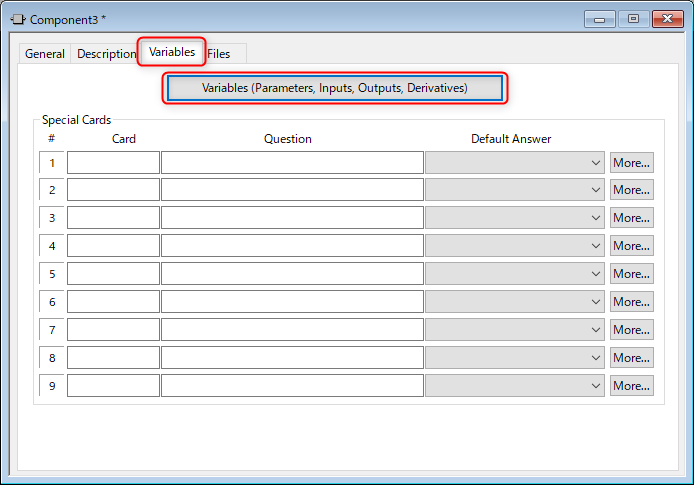
Type番号

コンポーネントは識別用に1～9999までの番号が割り当てられます。この番号はすべてのコンポーネントで重複のない一意の番号が割り当てられます。

既存のコンポーネントと重複しないよう、ユーザー作成のコンポーネントには予め201～300の番号が確保されています。新しいコンポーネントを作成する場合は、この範囲から番号を割り当てます。

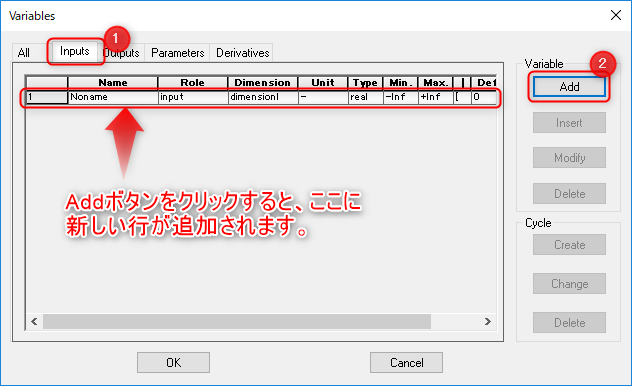
詳しくは「7.3.2. The First Line」を参照。

次に、「**Variable**」タブで、「**Variables(Parameters, Inputs, Outputs, Derivatives)**」ボタンをクリックして、設定画面を表示します。

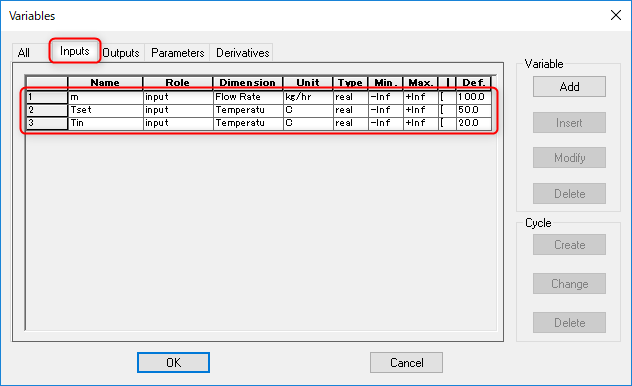


Inputの項目を追加します。

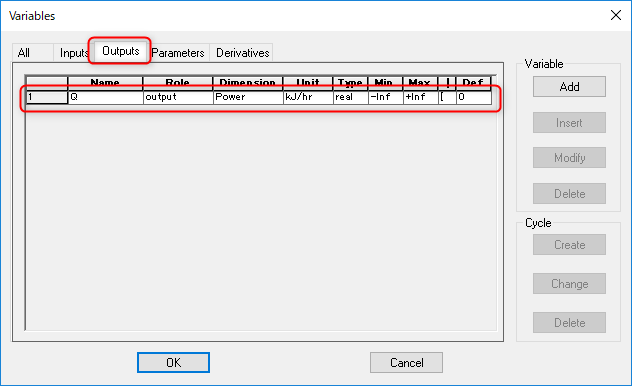
1. 「**Inputs**」タブを選択して
2. 画面右のAddボタンをクリックして新しい項目を追加します。



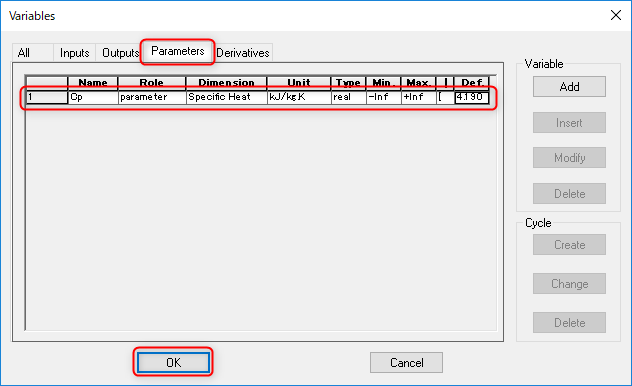
先ほどの表に合せて**Inputs**の項目を3つ追加して、設定を行います。表に従ってName、Dimension、Unit、Def.の各項目を設定します。



つづいて、同じように「**Outputs**」タブで項目を一つ追加、設定します。



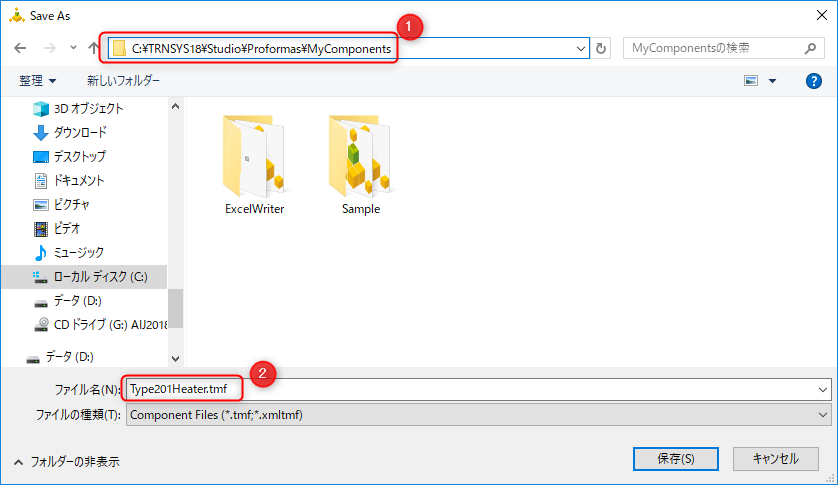
そして最後に「**Parameters**」タブで項目を一つ追加、設定したらOKボタンをクリックしてウィンドウを閉じます。



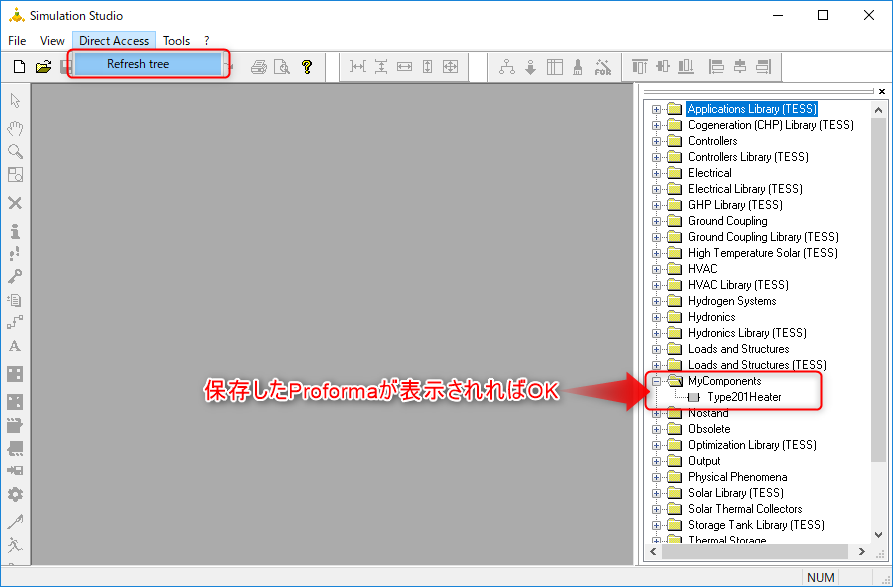
メニューから**[File]-[Save]**を選択してProformaを保存します。

① 保存先として新しいフォルダ”C:\TRNSYS18\Studio\Proformas\**MyComponents** “を作成します。

② Proformaを**Type201Heater.tmf**という名前で保存します。

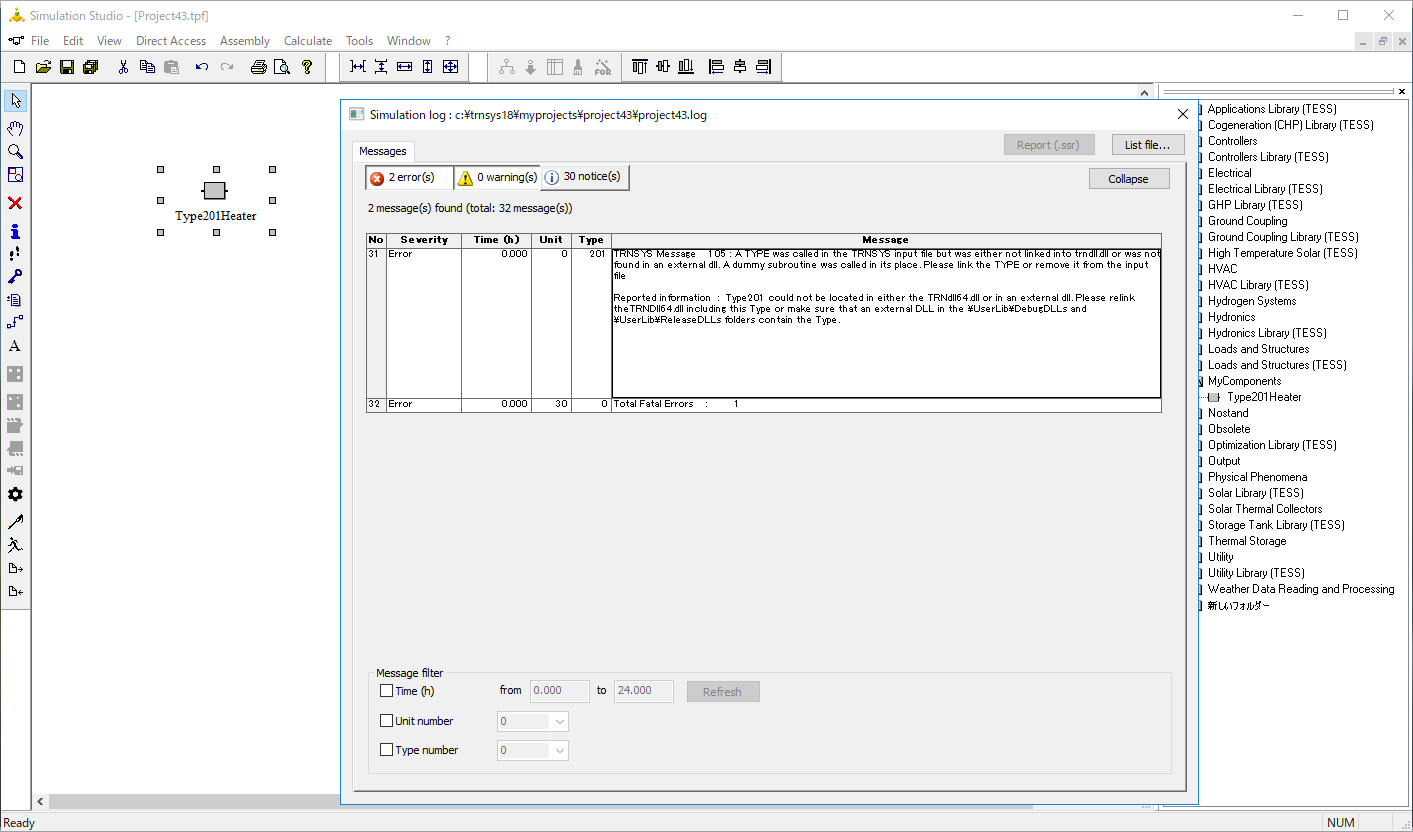


Simulation Studioのメニューから[**Direct Access**]-[**Refresh tree**]を選んで、ツールバーにType201Heaterが表示されればProformaの準備は完了です。



この段階で、すでにコンポーネントとして配置、他のコンポーネントと接続できる状態になっています。

ただし、試しに実行すると図のようなエラーが発生します。これはProformaは出来上がったものの、コンポーネントの本体、つまり計算処理の部分が作成されていないためです。次頁から計算処理の部分を作成します。



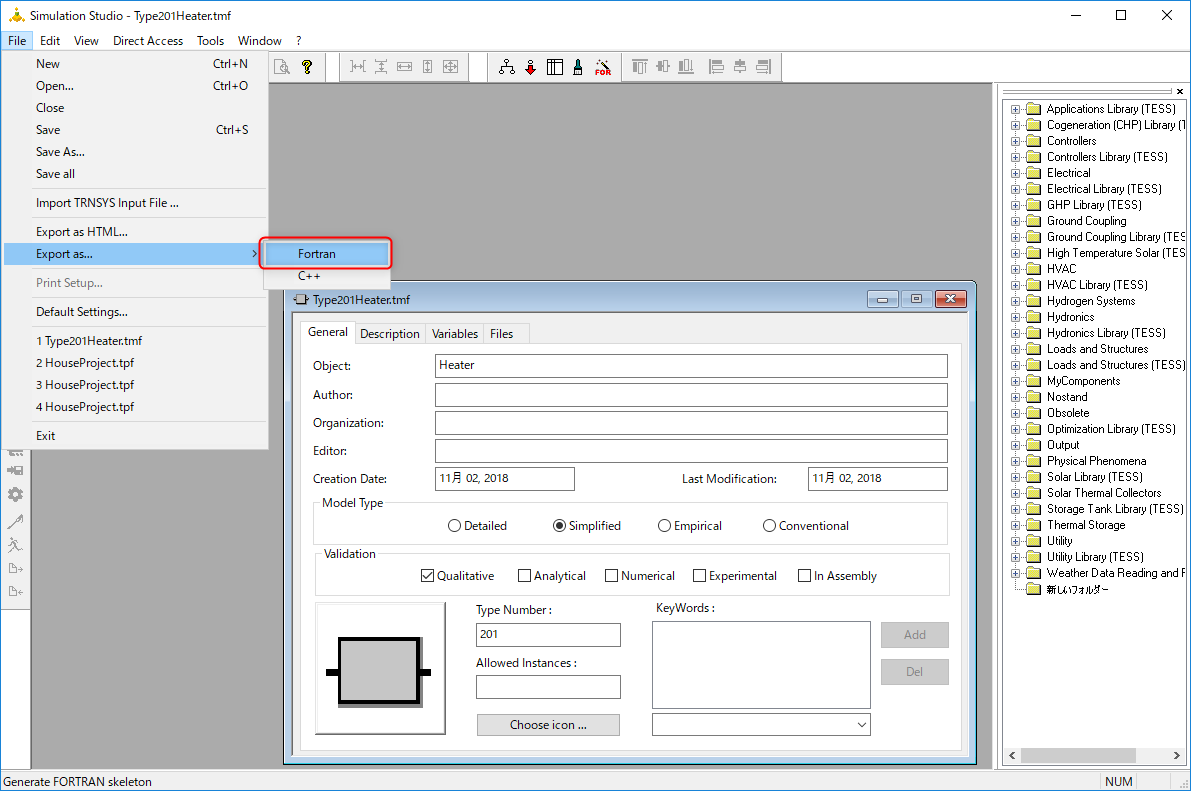
# 計算機能の実装

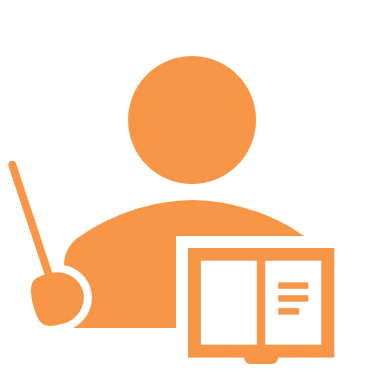
* 1. ソースコードのエクスポートと編集

**前頁で作成したProforma**から**ソースコードの雛形**をエクスポートします。

Simulation StudioでProformaを開いた状態にします。（Proformaを閉じている場合は、"C:\TRNSYS18\Studio\Proformas\MyComponents\Type201Heater.tmf"を再度開いて下さい）

次にメニューから [**File**]-[**Export** **as**]-[**Fortran**] を選んでソースコードをエクスポートします。

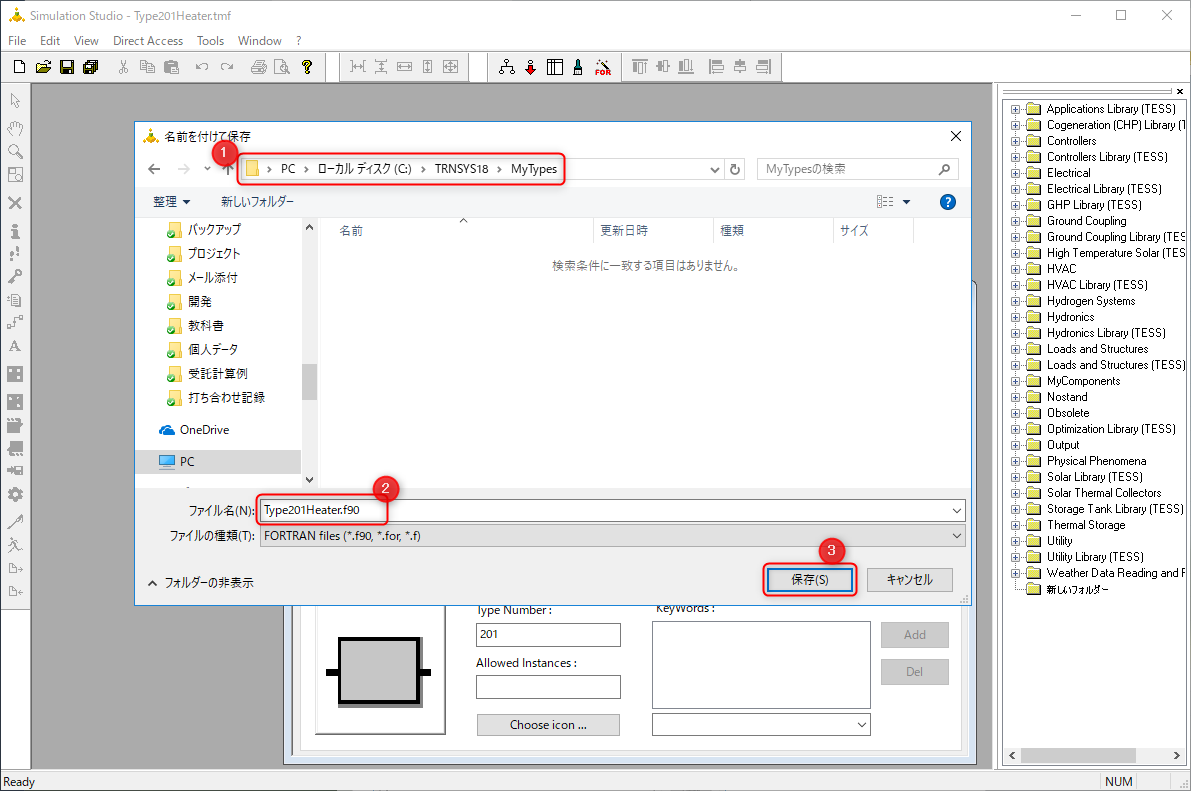


ソースコードのエクスポート

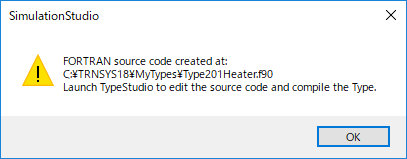
コンポーネントのソースコードはFortranの他、C++でエクスポートできます。

C++のソースコードは、Visual Studioで編集、コンパイルが可能です。

1. エクスポート先のフォルダとして"**C:\TRNSYS18\MyTypes**"を選択。
2. ファイル名、"**Type201Heater.f90**"を指定。
3. 「保存」ボタンをクリックしてエクスポートします。



エクスポートが終わると**TypeStudio**で編集を促す、図のようなメッセージが表示されます。



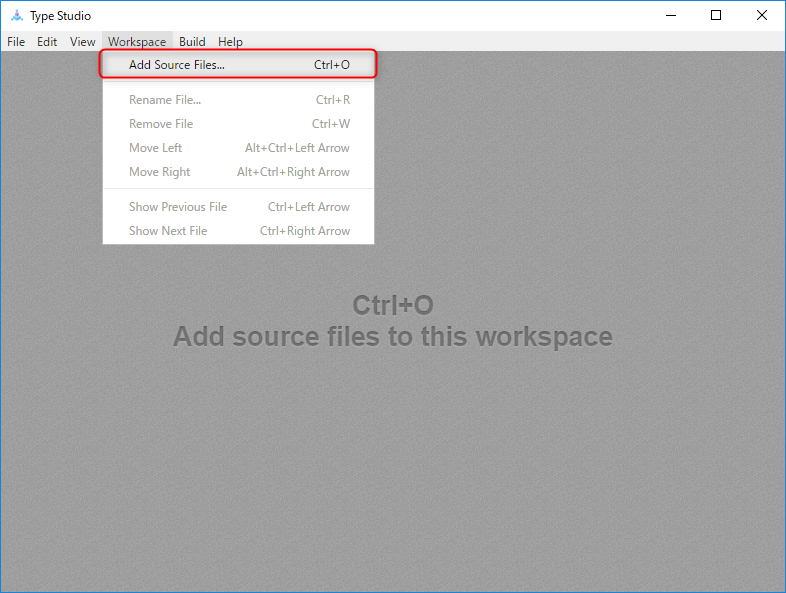
これでソースコードの準備は完了です。

* 1. TypeStudioの起動とソースコードのインポート

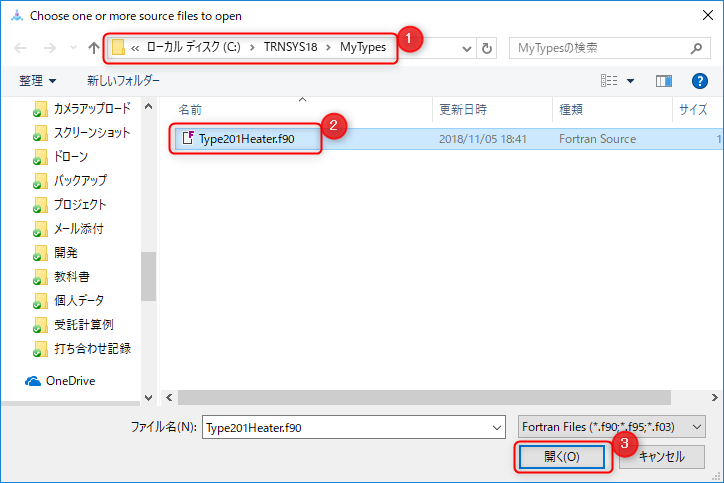
**前頁でエクスポートしたソースコードをTypeStudioで編集します。**

**スタートメニュー**から**TypeStudio**を起動して、エクスポートしたソースコードを読み込みます。

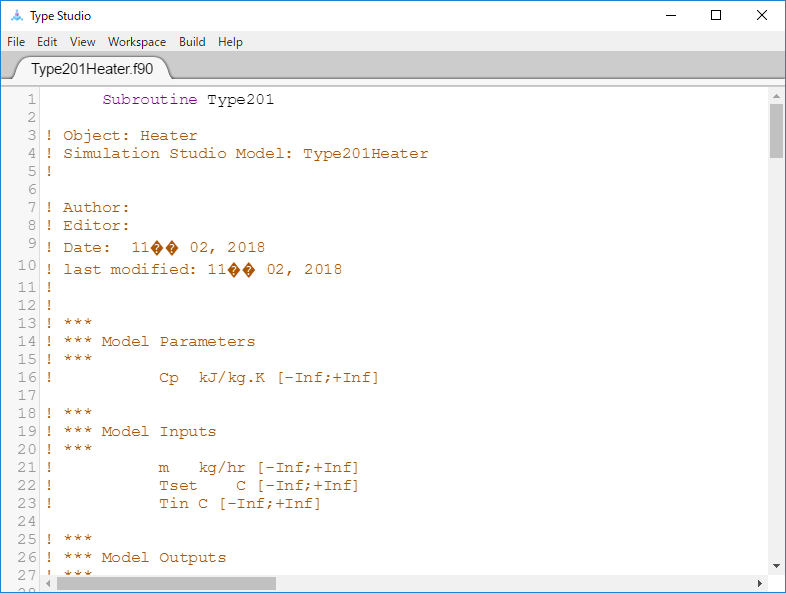
TypeStudioのメニューから[**Workspace**]-[**Add Source Files**]を選択して、



1. 先ほどソースコードをエクスポートしたフォルダ "**C:\TRNSYS18\MyTypes**"で、
2. "**Type201Heater.f90**"を選択して、
3. 「開く」ボタンをクリックしてソースコードを読み込みます。

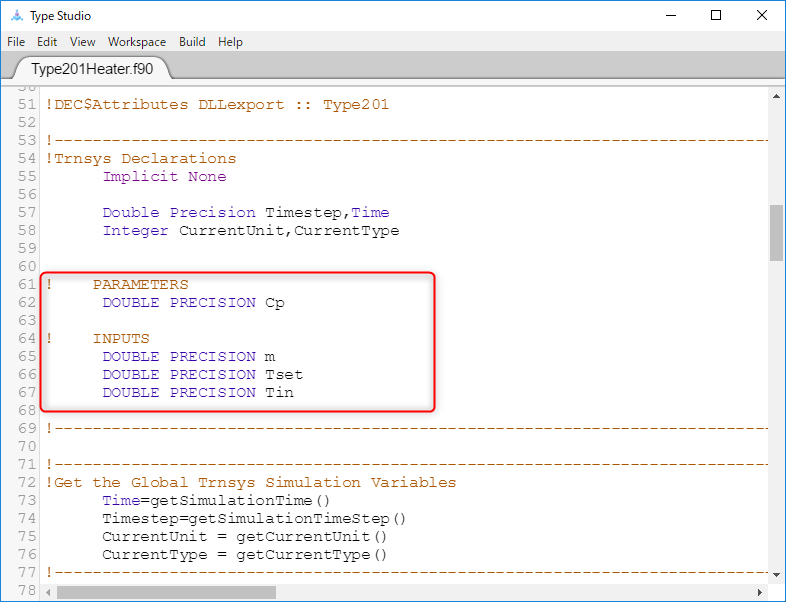


TypeStudioの画面が図のように表示されていれば準備は完了です。



* 1. ソースコードの確認

ソースコードの中身を確認すると、61行目から67行目にProformaの**Parameters**, **Inputs**の値に相当する変数が宣言されている事が確認できます。このようにProformaで定義した情報がソースコードに反映されていることが分かります。

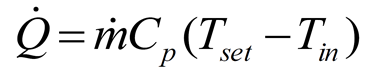


* 1. 処理の実装

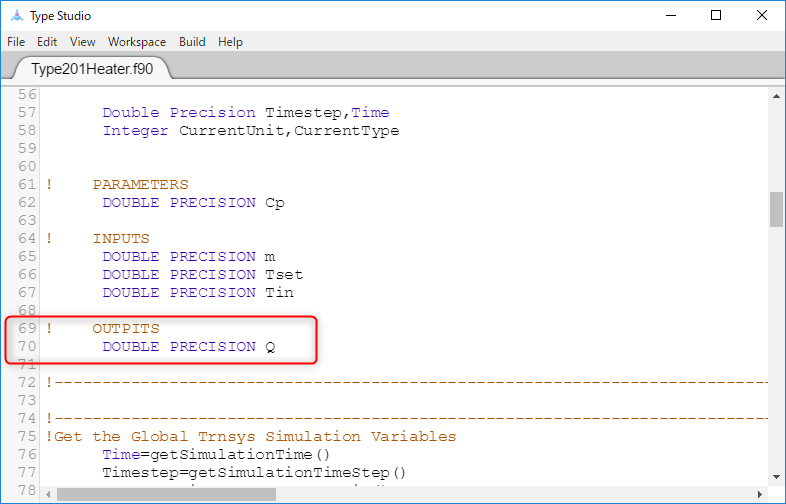
雛形のソースコードへ計算処理や出力を実装します。

* + 1. 変数の追加

はじめに出力用に"**Q**"に対応する変数を追加します。

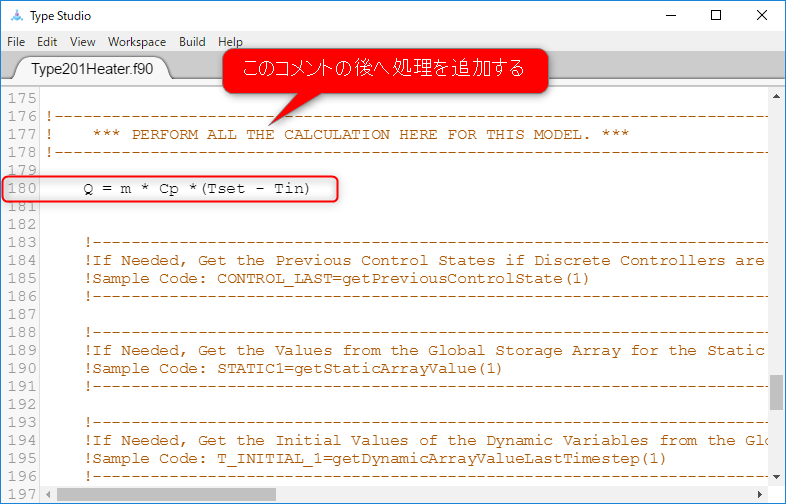


図の赤枠のようにParameters, Inputsの変数宣言の後へ変数"**Q**"を追加します。



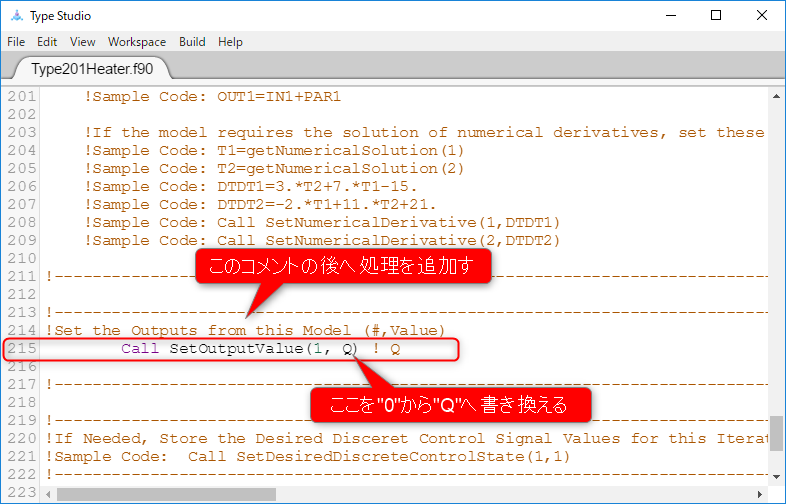
* + 1. 計算処理の追加

コンポーネントのソースコードでは処理を追加する位置が決められています。ソースコードのコメントで"**PERFORM ALL THE CALCULATION HERE FOR THIS MODEL.** "と書かれた位置に処理を追加します。この例では**180行付近**にコメントがあるので、計算式に沿って、図のように処理を追加してください。



* + 1. 出力処理の追加

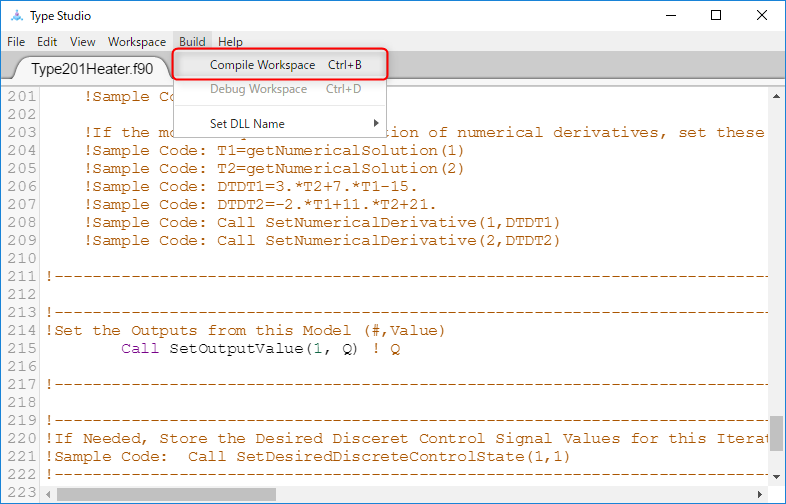
最後に、計算した値を**Outputs**へ出力します。これも追加する位置が決まっています。ソースコードのコメント"**Set the Outputs from this Model (#,Value)**"を探して、その後に出力処理を記述します。  
この例では**214行目付近**にコメントが見つかります。すでに出力処理の雛形が用意されているので、出力する値の部分のみを図のように書き換えます。



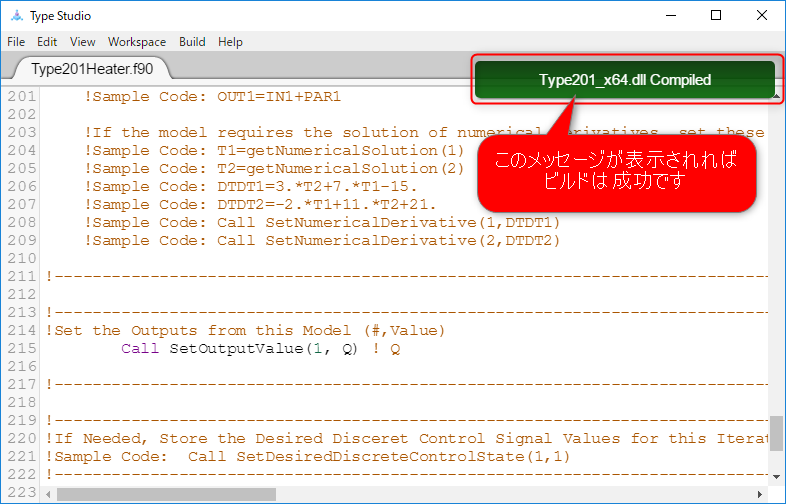
* + 1. ビルド

ソースコードの修正が終わったら、いよいよビルドです。（ソースコードからコンポーネントの実体を生成します）

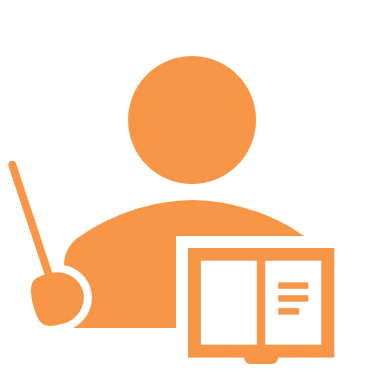
メニューから[Build]-[Compile Workspace]を選択します。



図のように画面右上に緑色のメッセージが表示されればビルドは成功です。もし違うメッセージが表示されていたら、どこかに間違いがあります。ソースコードを見直して修正、再度ビルドを行ってください。

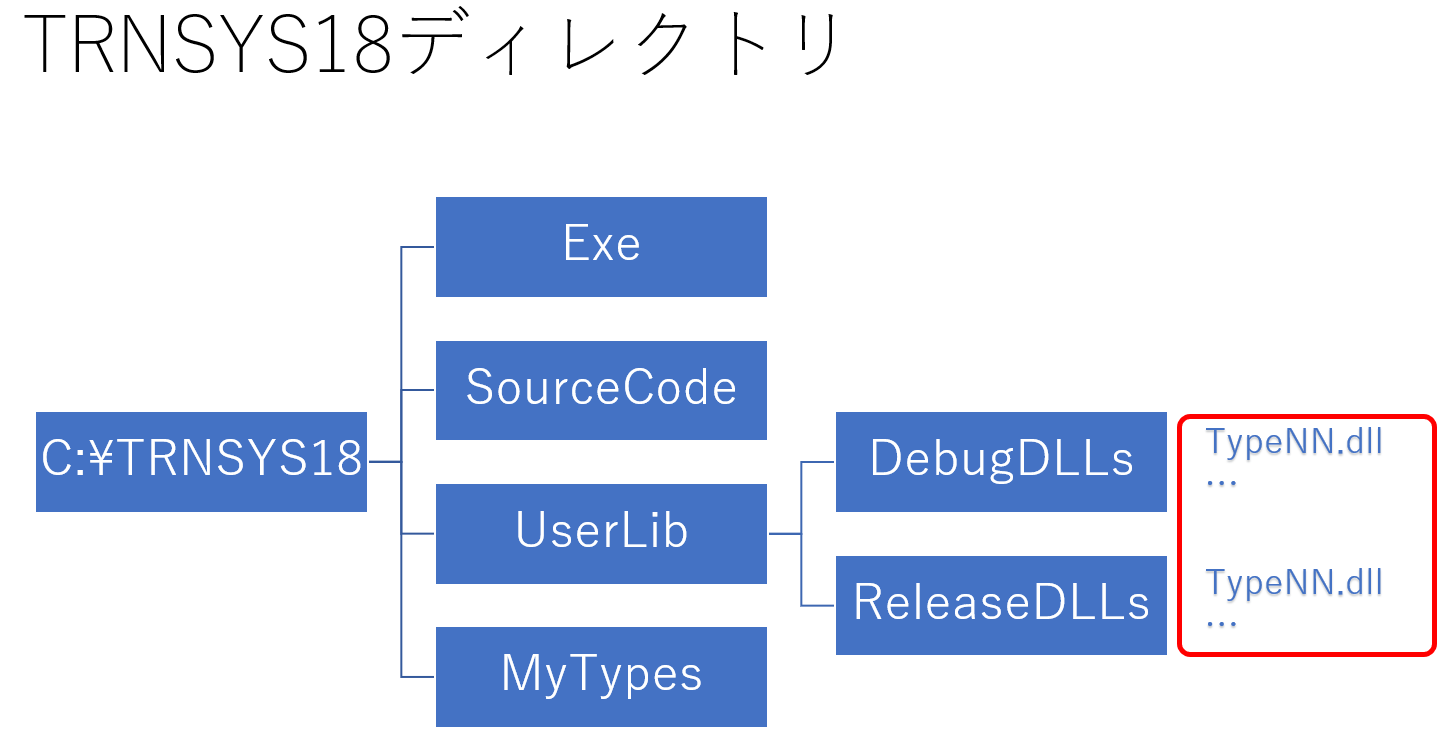


ビルドが成功したらコンポーネントが利用できる状態です。（関連ファイルがTRNSYSのフォルダに配置されて使用できる状態になります）

ビルドと配置

TRNSYSのコンポーネントはDLL（Dynamic Link Library）形式のファイルとして作成されます。

TypeStudioでビルドが成功すると、C:\TRNSYS18\UserLib\DebugDLLsフォルダ、ReleaseDLLsフォルダにDLLが自動的に配置されます。

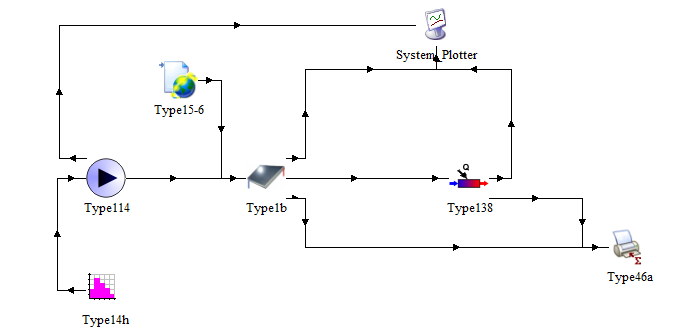


# コンポーネントの動作検証

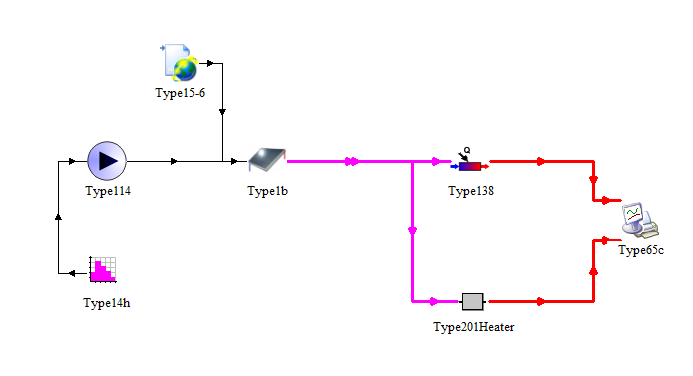
ビルドしたコンポーネントの動作を検証します。

* 1. プロジェクトを用意する

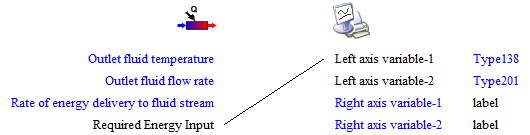
テスト用にプロジェクトを用意します。今回はExamplesフォルダのBegin.tpf（"C:\TRNSYS18\Examples\Begin\Begin.tpf"）を利用してテスト用のプロジェクトを作成します。



コンポーネントを整理して、前回作成したコンポーネント（Type201Heater)を図のように配置、接続します。

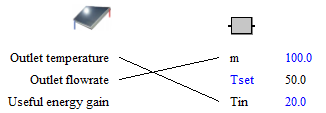


**このプロジェクトで使用されているType138**では作成したヒーターと**同じ計算**を行っています。計算結果を比較するため**Type138**から**Required Energy Input**の値をType65へ出力します。

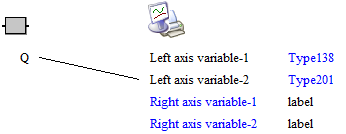


* 1. Type201Heaterの接続

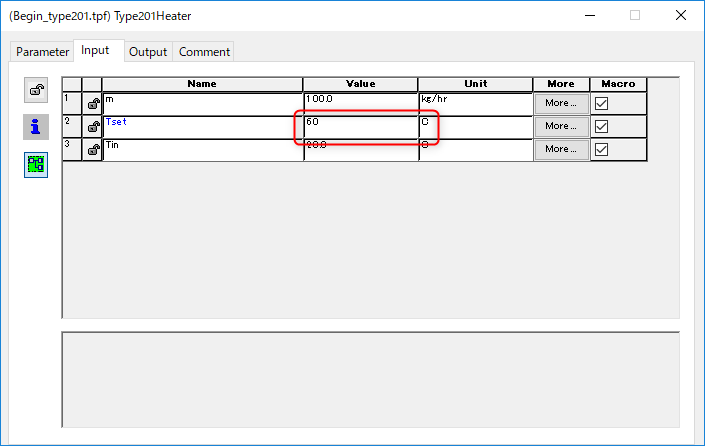
**Input**は集熱パネルから出口温度（Outlet temperature）、流量(Outlet flowrate)の値をそれぞれ対応する**Tin, m**へ接続します。



そして**Output**はType65へ負荷（**Q**)を接続しておきます。

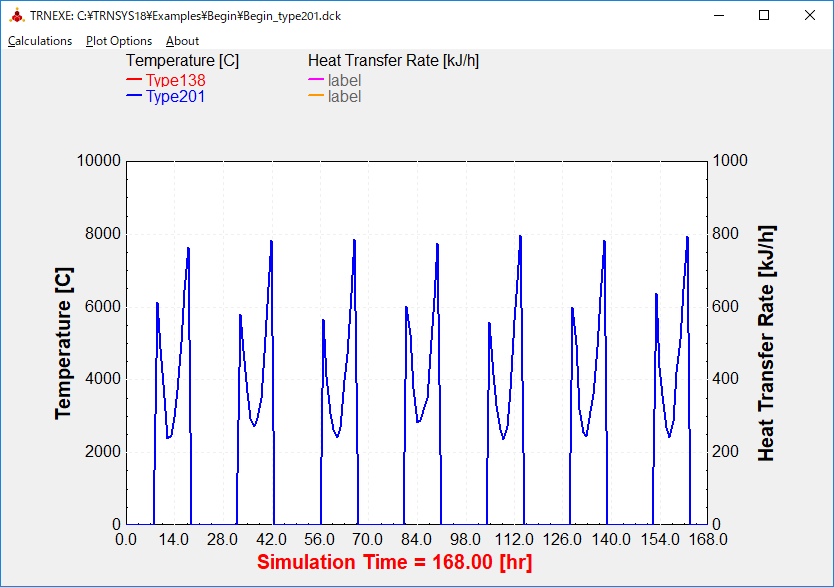


最後に、配置されているType138の設定温度（Setpoint temperature）に合せて、Type201Heaterの設定温度（**Tset**）を**60℃**へ変更します。



* 1. 結果を確認する

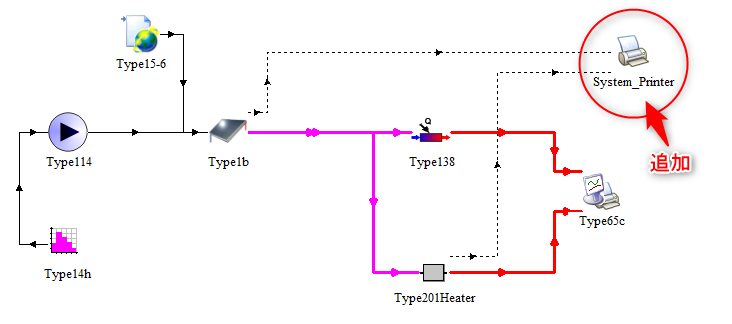
計算を実行して計算結果の確認を行います。



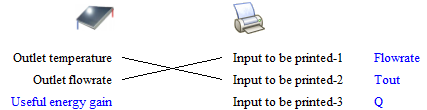
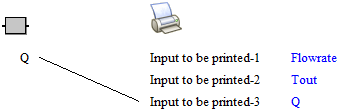
図のようにType138とType201の**結果がまったく同じ**でグラフが重なって描かれています。少々分かりにくい画面になってしまいましたが、コンポーネントは正しく動作しているようです。  
（注：画面はType65/Parameters/Left axis maximumを**10000**へ変更しています）

* 1. 計算結果の検証

この例では同じ計算を行うコンポーネントを使って比較検証しました。同じ計算を行うコンポーネントが通常は存在しないので、なんらかの方法で結果を検証する必要があります。  
ここでは関連するデータを書き出してExcelを使って検証する例をご紹介します。まずは検証用のデータを書き出すためType25を配置して、図のように接続します。

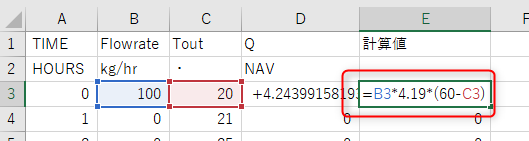


コンポーネントがInputsで受け取っているパネルの出口温度、流量、それとコンポーネントの計算結果をType25で出力します。



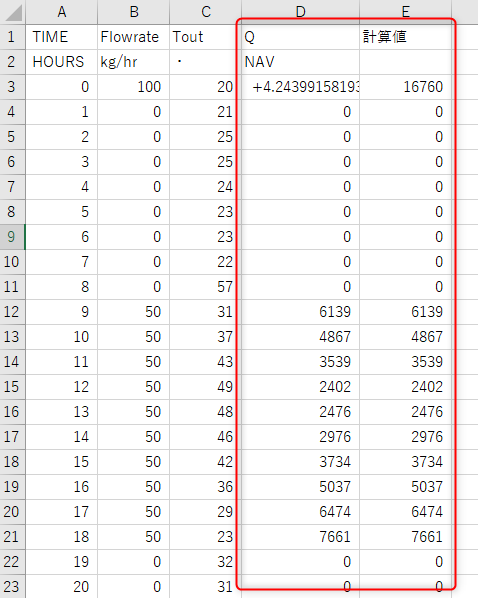
Type25de書き出された値を使ってExcelで計算して同じ結果になるか比較することで処理内容を検証します。

ファイルをExcelで開いて、コンポーネントが内部で行っているのと同じ計算式をセルに入力します。



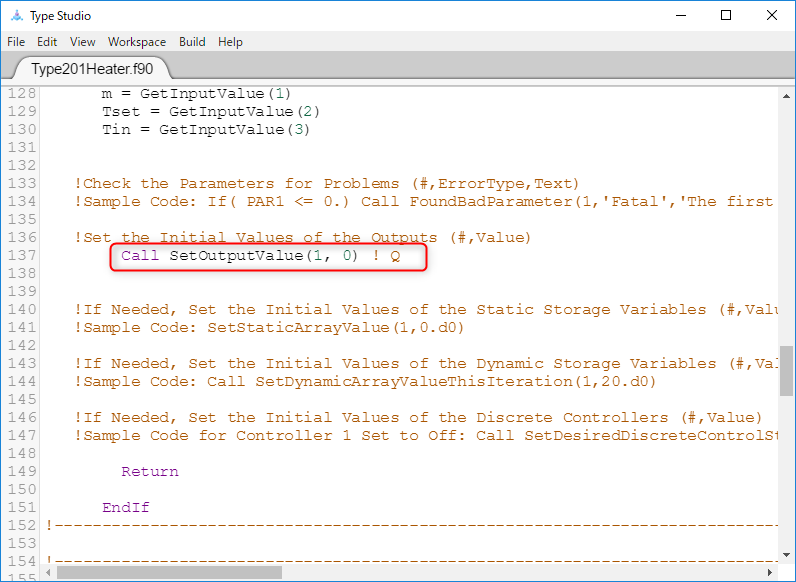
注：上の図では、0hourのQの値が変な値になっていますが、これは出力値の初期化が原因です。理由と対策は後述します。

**Qの値**と**Excelで計算した値**を比べると、同じ値になっている事が確認できます。



# Outputの初期化について

さて、0hourでおかしな値が出力されていた件ですが、これはOutputの初期化が原因です。コンポーネントのソースコードを再確認すると Output の初期化を行っている箇所があります。（下図）



ここで、Outputの1番目、つまり**Q**の値を”**0**”に初期化しています。一見”0”で問題ないように見えますが、この書き方はFORTRANでは整数値として扱われます。やや専門的な話になりますが、整数値から実数値へのキャストによる誤差が発生しています。  
さきほどの図では値の部分が見切れていていましたが、"+4.2439915819305446-314"（指数表記）で書き出されています。値としては限りなく"0"に近いのですが、計算上はこういうケースが発生します。

対策はごくシンプルで初期化の値を"**0**"から"**0.0**"へ書き換えて**実数**としてFORTRANが認識できるようにすれば解決です。



# 参考資料

* 1. リポジトリ
* プロフォルマ

<https://github.com/TRNSYSJP/TRNSYS.JP/tree/master/TRNSYS18/Studio/Proformas/MyComponents/Type201Heater>

* プロジェクト

<https://github.com/TRNSYSJP/TRNSYS.JP/tree/master/TRNSYS18/MyProjects/Type201Heater>

* ソースコード

<https://github.com/TRNSYSJP/TRNSYS.JP/tree/master/TRNSYS18/MyTypes/Type201Heater>

* 1. ブログ

TRNSYS TypeStudioのテキストを公開

<https://www.kankyoukei.com/2018/11/trnsystypestudio.html>

TypeStudioで作るTRNSYSコンポーネント（1）

<https://www.kankyoukei.com/2018/11/typestudiotrnsys-2.html>

TRNSYS18

TypeStudioで作るTRNSYSコンポーネント

2019年12月2日　更新

発行　株式会社クアトロ