

## 公開要素モデルのディバッガーからの動作方法

### 1. 公開要素モデルのコンパイル

CommonMPでは、使用できる要素モデルを ソース付きで公開しています。

要素モデルを開発したときの ソース 及び マイクロソフト社製 Visual Studio のプロジェクトは

¥CommonMP¥Source¥HYMCO¥PublicModelElement¥ に 各DLL作成プロジェクト毎にフォルダーを作成して 格納しています。

これらの プロジェクトを 一括してコンパイルしたいときには

¥CommonMP¥Source¥HYMCO¥PublicModelElement¥ PublicModelElement.sln をダブルクリックし マイクロソフト社製 Visual Studio を立ち上げてください。

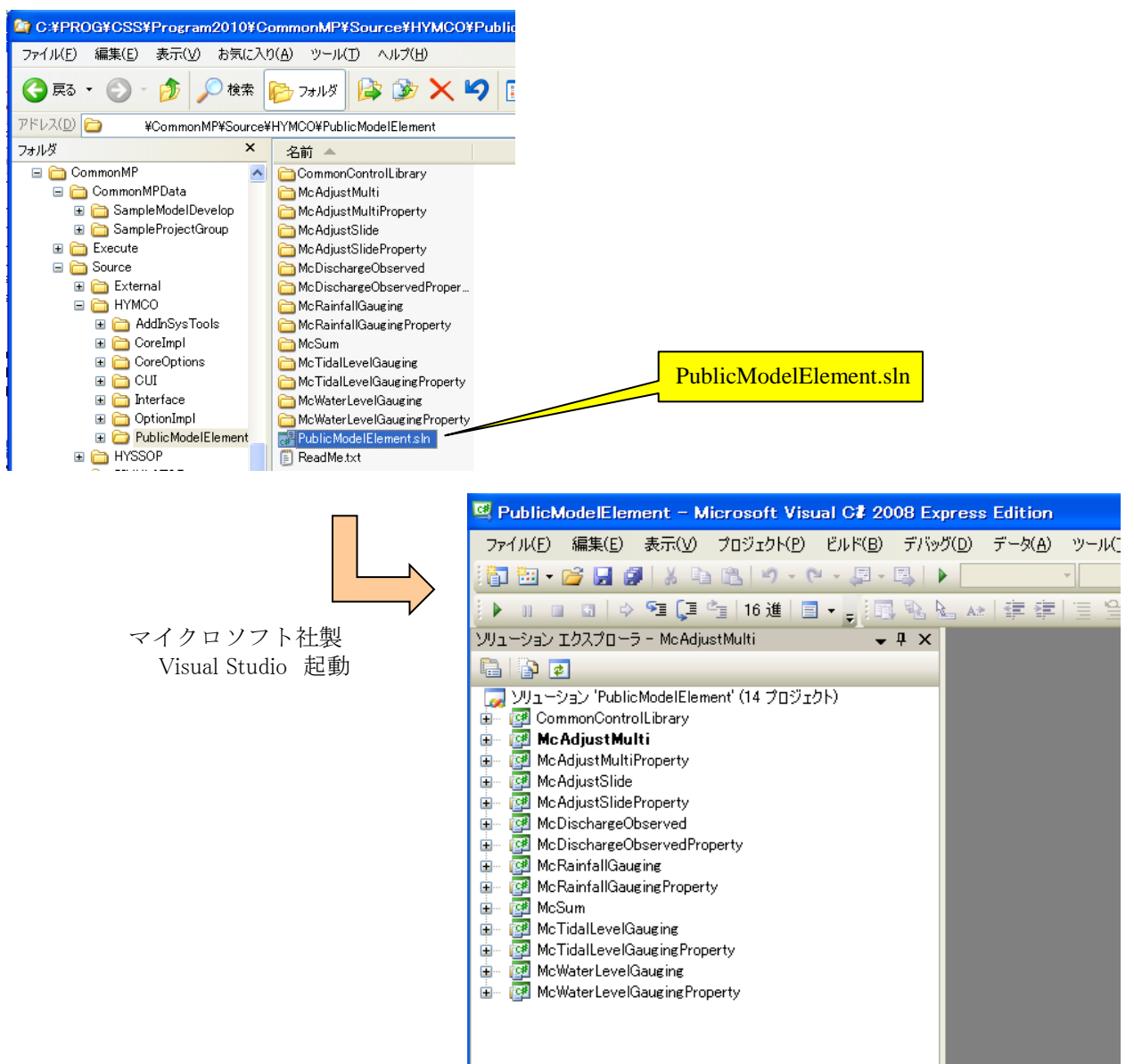


図 1. 1 ビジュアルスタジオ起動

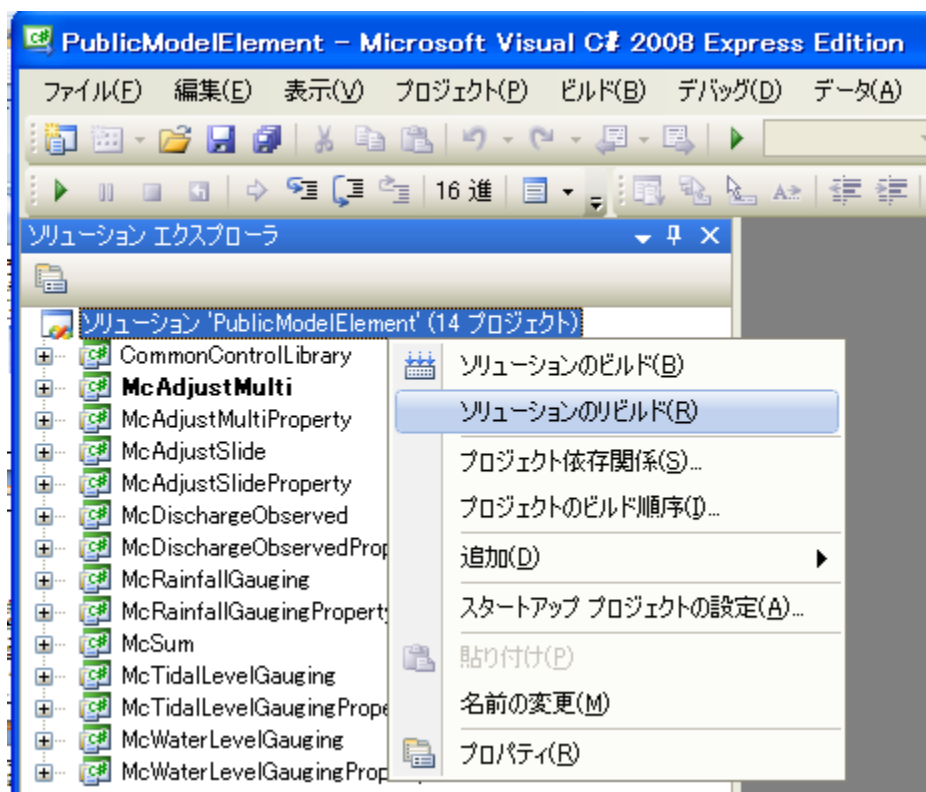


図 1. 2 コンパイル

図1. 2に示すように Visual Studio のポップアップメニューから ソリューションのビルドを選択して下さい。

コンパイルが開始されます。 コンパイルにより生成された各DLLは

¥CommonMP¥Execute¥bin¥ 下に 自動的にコピーされます。 また、 必要なアイコン、解説書も それぞれ、

¥CommonMP¥Execute¥ModelIcon ¥ 及び ¥CommonMP¥Execute¥ModelManual¥ 下にコピーされます。

CommonMPを 動作させると、 公開モデルを使用する事ができます。

## 2. 公開モデルのディバッガー上での動作

公開モデルの動作を理解するためには、ディバッガー上で 動作させて見るのが一番の早道です。

CommonMPでは、 モデル開発の為の デバッグ環境を準備していますので、このディバッガー上で 公開モデルの動作を トレースしてみます。

## 2. 1. モデル開発環境立ち上げ

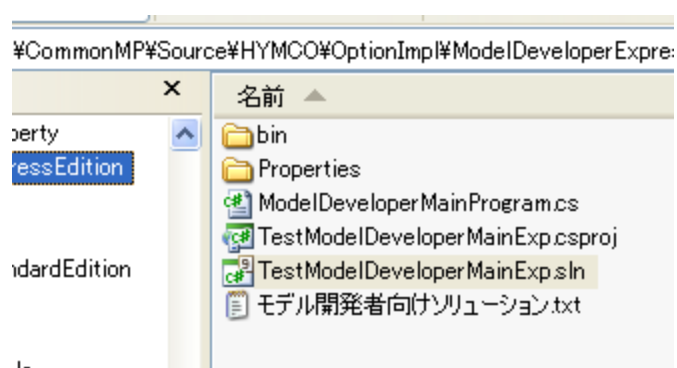
自PC上にインストールされている マイクロソフト社製の VisualStudio の エディションに応じて

¥CommonMP¥Source¥HYMCO¥OptionImpl¥ModelDeveloperExpressEdition  
¥TestModelDeveloperMainExp.sln

または、

¥CommonMP¥Source¥HYMCO¥OptionImpl¥ModelDeveloperStandardEdition  
¥TestModelDeveloperMainStd.sln

を起動させて下さい。(図2. 1参照)



ソリューションを開く

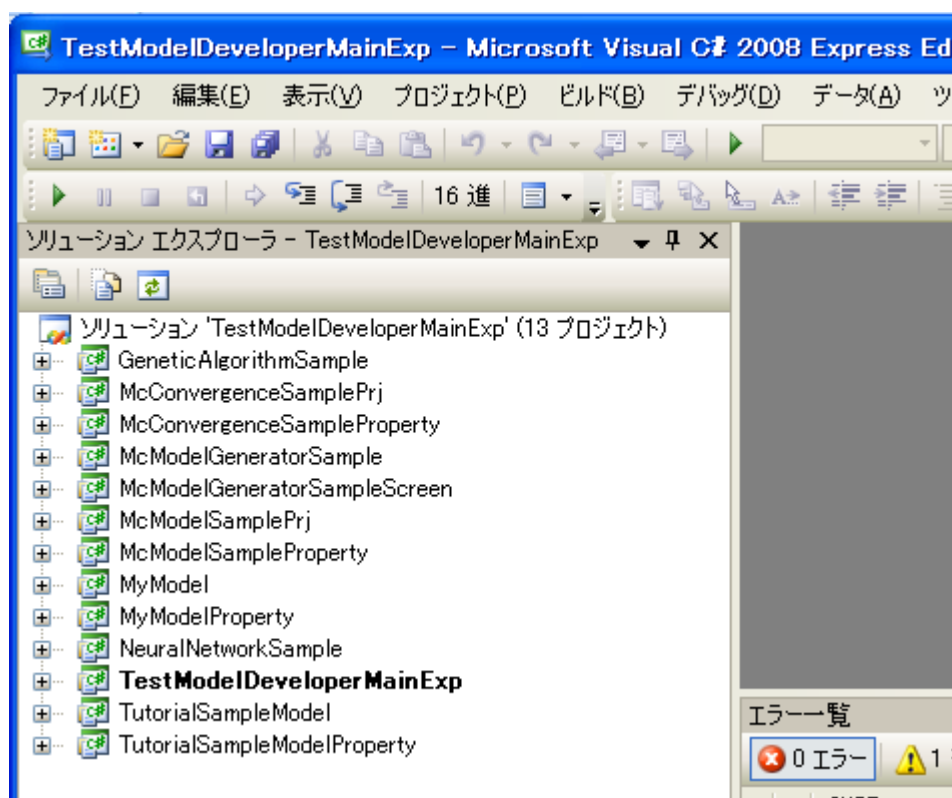


図 2. 1 デバッグ用 ソリューション立ち上げ

## 2. 2. 公開要素モデルをソリューションへ追加

ここで、デバッグ用のソリューションに 公開要素モデルを追加します。図2. 2に示すように ポップアップメニューからVisual Studio のソリューションエクスプローラーの 上段に表示されている『ソリューション' TestModelDeveloperMainExp'』を選択して マウス右クリックによって ポップアップメニューを表示させます。ポップアップメニューの『追加』→『既存のプロジェクト』を選択し、既存プロジェクト追加ダイアログを開きます。

このダイアログ上で ¥CommonMP¥Source¥HYMCO¥PublicModelElement 下に存在する フォルダから デバッグ動作させたいモデルのプロジェクトを選択します。図2. 2の例では、McDischargeObserved¥McDischargeObserved.csproj を選択しました。

すると、McDischargeObserved が ソリューションにされます。

## 2. 3. 参照の追加

前記手順で、ソリューションにプロジェクトを追加しても、デバッグ用のメインプログラム ( TestModelDeveloperMainExp )が、追加プロジェクトを参照しなければ、追加プロジェクトはデバッガー上で 動作しません。そこで、図2. 3に示すように デバッグ用のメインプログラム ( TestModelDeveloperMainExp )の参照設定に プロジェクト『McDischargeObserved.csproj』を追加します。

まず、デバッグ用のメインプログラム ( TestModelDeveloperMainExp ) の参照設定上のポップアップメニューから「既存プロジェクト」を選択します。そこで 表示されたダイアログ上のプロジェクトから 参照追加したいプロジェクトを選択します。例では McDischargeObserved.csprojを選択しました。すると、 参照設定として McDischargeObserved.が設定されました。

この様にして、 通常モデル開発の場合と同様に 公開モデル内に ブレークポイント等を設定し、動作を詳細に解析する事が可能となります。

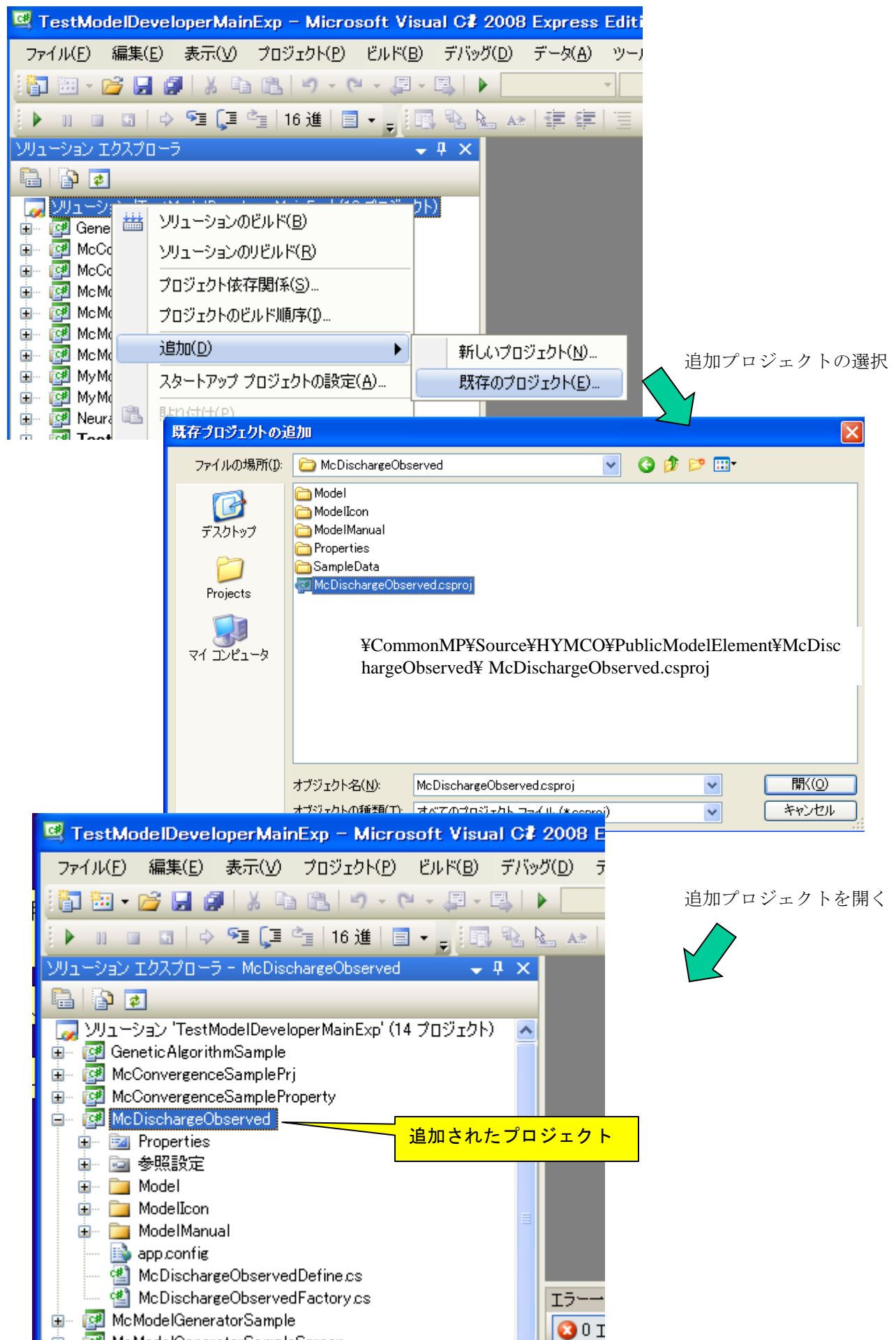


図 2. 2 公開要素モデルプロジェクトのソリューションへの追加

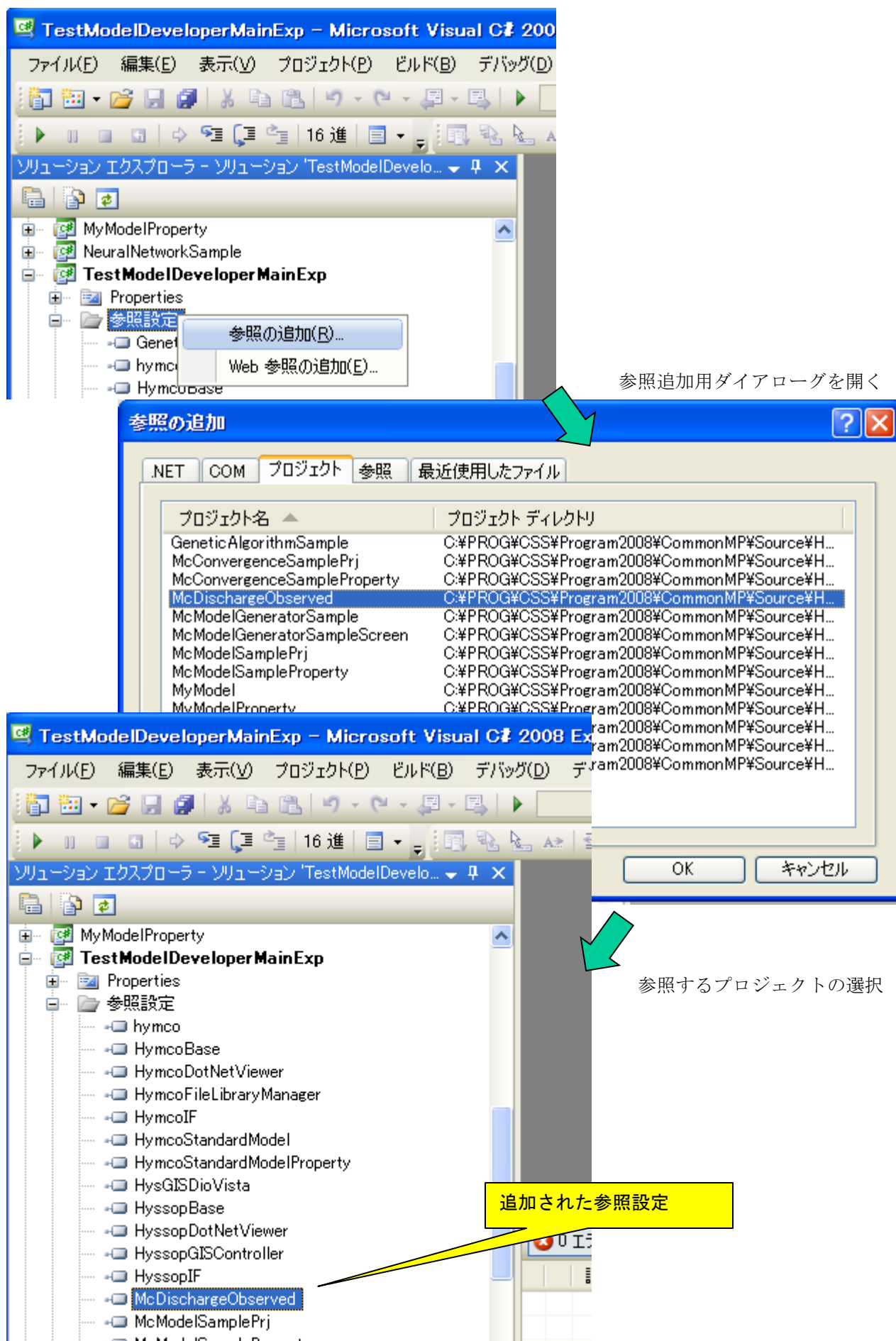


図 2. 3 参照の追加

3. 補足設定

前記2章までの操作により、プログラムは動作するようになりました。しかしながら そのままでは、アイコンの表示や、モデルの解説書の表示にエラーが発生します。(図3. 1参照)

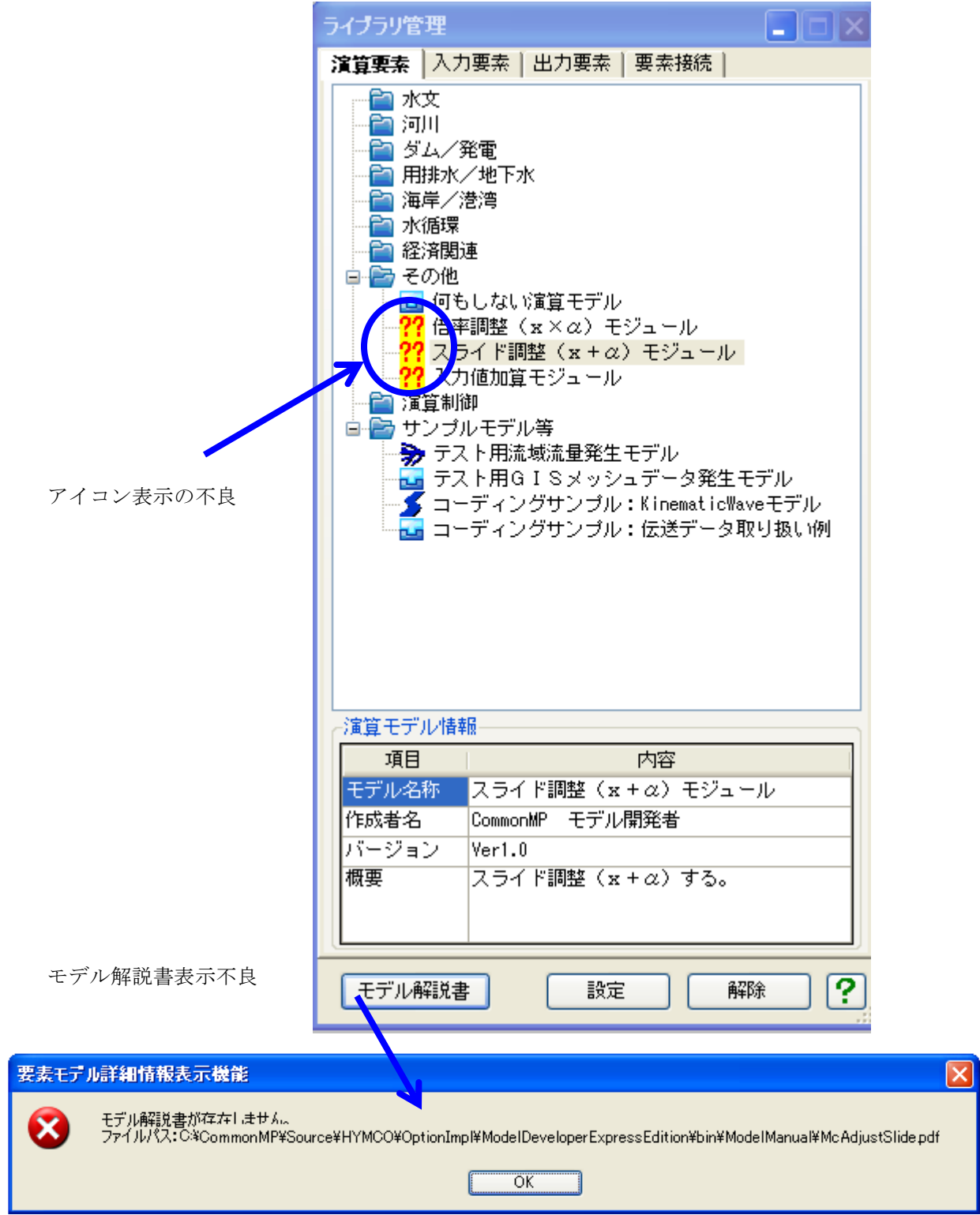


図 3. 1 モデルアイコン、モデル解説書表示不具合

この現象は、デバッグ環境の 所定のアイコン格納位置は、解説書格納位置に それぞれ、モデルのアイコンや、解説書が格納されていないことに起因します。

1章に示した手順でコンパイルを行った場合、 正式なCommonMPが動作する環境 ( ¥CommonMP¥Execute¥ ) 下には アイコン、解説書フォルダーは格納されました。しかしながら、デバッグ中の動作は、別な環境上で動作します。

¥CommonMP¥Source¥HYMCO¥OptionImpl¥ModelDeveloperExpressEdition  
¥ TestModelDeveloperMainExp.sln

から デバッグ環境を立ち上げた場合、 デバッグ中のCommonMPは  
¥CommonMP¥Source¥HYMCO¥OptionImpl¥ModelDeveloperExpressEdition¥bin  
を ホーム 環境として認識します。

従って、追加した 公開モデルの アイコン、解説書も  
¥CommonMP¥Source¥HYMCO¥OptionImpl¥ModelDeveloperExpressEdition¥bin

下の 所定の位置に配置する必要があります。

そこで、必要なファイルを 手動で コピーする必要があります。

公開モデル は 各プロジェクトフォルダー下に ModelIcon、 ModelManual の各フォルダーを持っています。そこで、手動で内部に存在する アイコンファイル (例: McDischargeObserved.ico) や、解説書ファイル (例: McDischargeObserved.pdf ) をデバッグ環境下 (例えば ¥CommonMP¥Source¥HYMCO¥OptionImpl¥ModelDeveloperExpressEdition¥bin ) の ModelIcon、 ModelManual フォルダー下に コピーして下さい。

この様にする事で、デバッグ環境でも 本番CommonMP動作環境と同じように モデルを動作させる事ができます。