Magic xpi コネクタビルダによる コネクタ作成方法

マジックソフトウェア・ジャパン株式会社

目次：Magic xpi コネクタビルダによるコネクタ作成方法

第1章

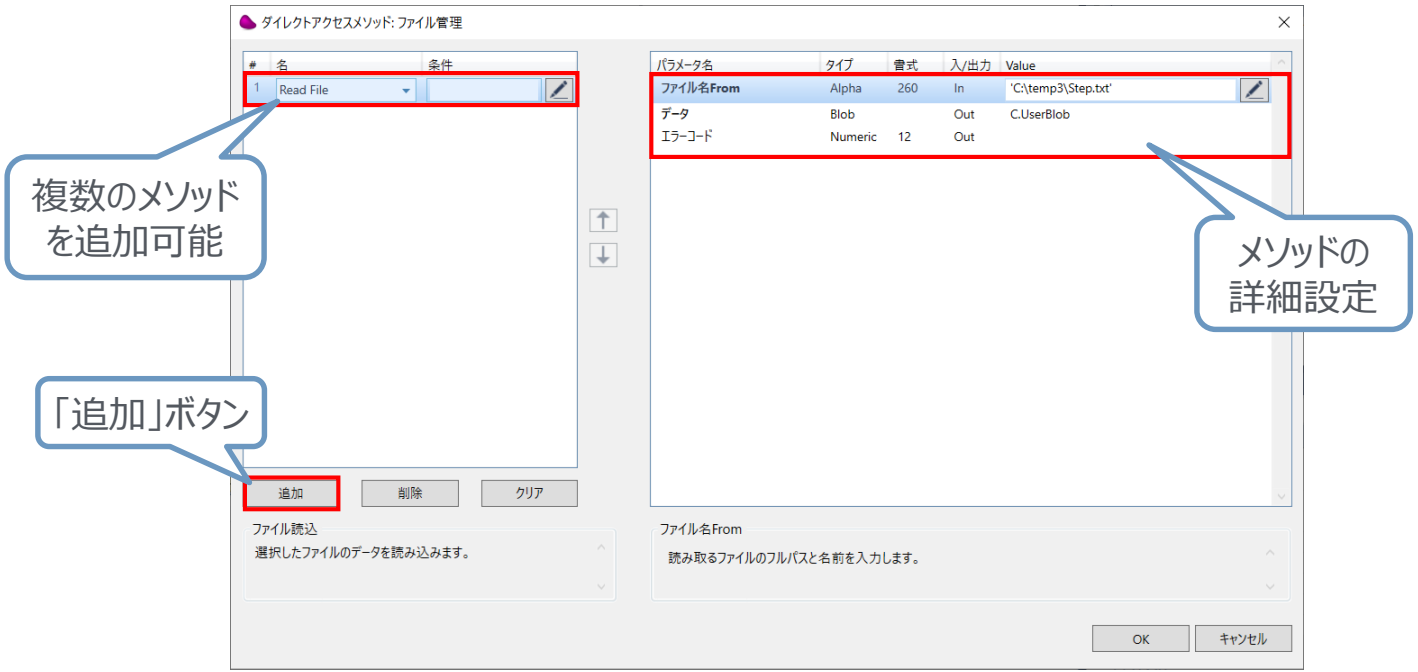
Methodインタフェース

コネクタの作成方法

1.1
Method
インタフェースについて

1.1 Methodインタフェースについて

- xpi標準のファイル管理コンポーネントなどに見られるインタフェース。
- 左ペインの「追加」ボタンでメソッドを追加し、右ペインでメソッドの詳細を設定します。



1.1 Methodインタフェースについて

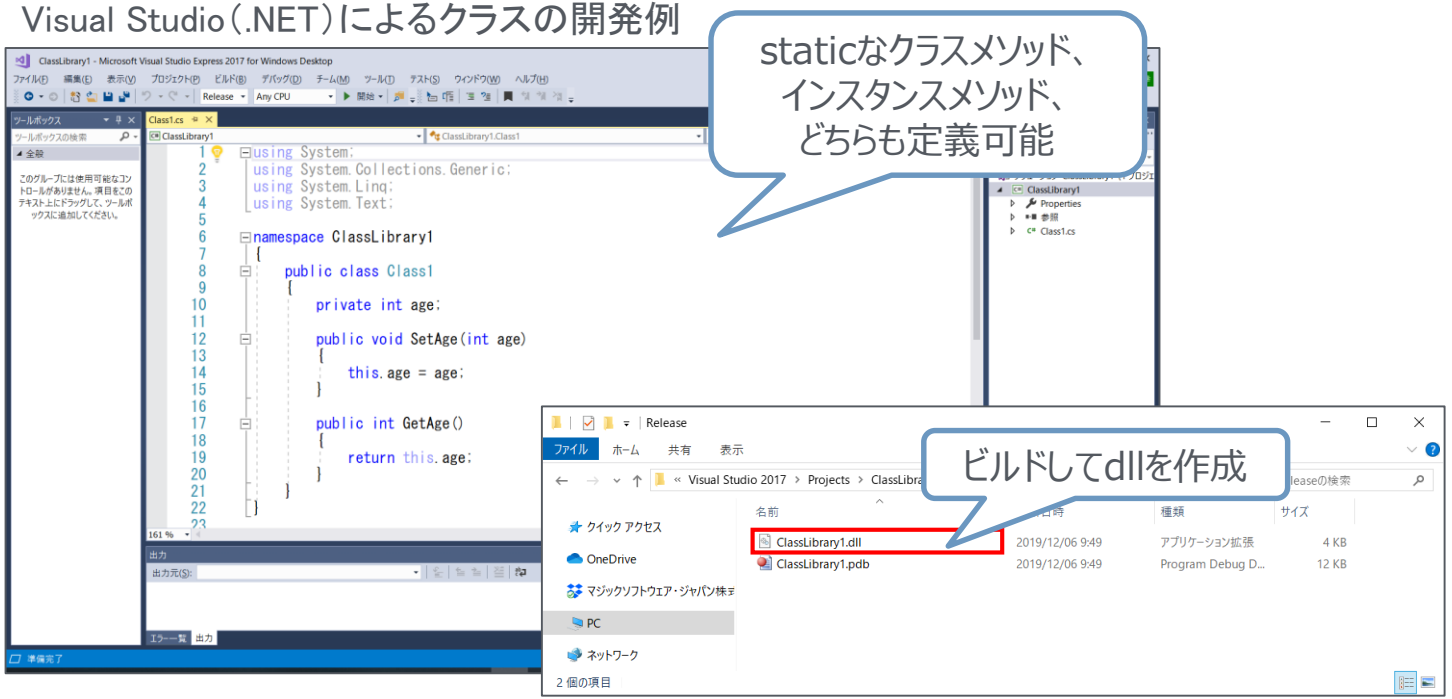
- Methodインタフェースコネクタを作成する場合、UIは前頁の画面固定となり、開発不要です。
- コネクタで使用するメソッドを持つクラスを.NETまたはJavaで開発します。

1.2
クラスの作成

1.2 クラスの作成

- Methodインタフェースコネクタを作成する場合、予めメソッドを持つクラスを作成します。
- 開発言語は .NET もしくは Javaのどちらかを選択することができます。

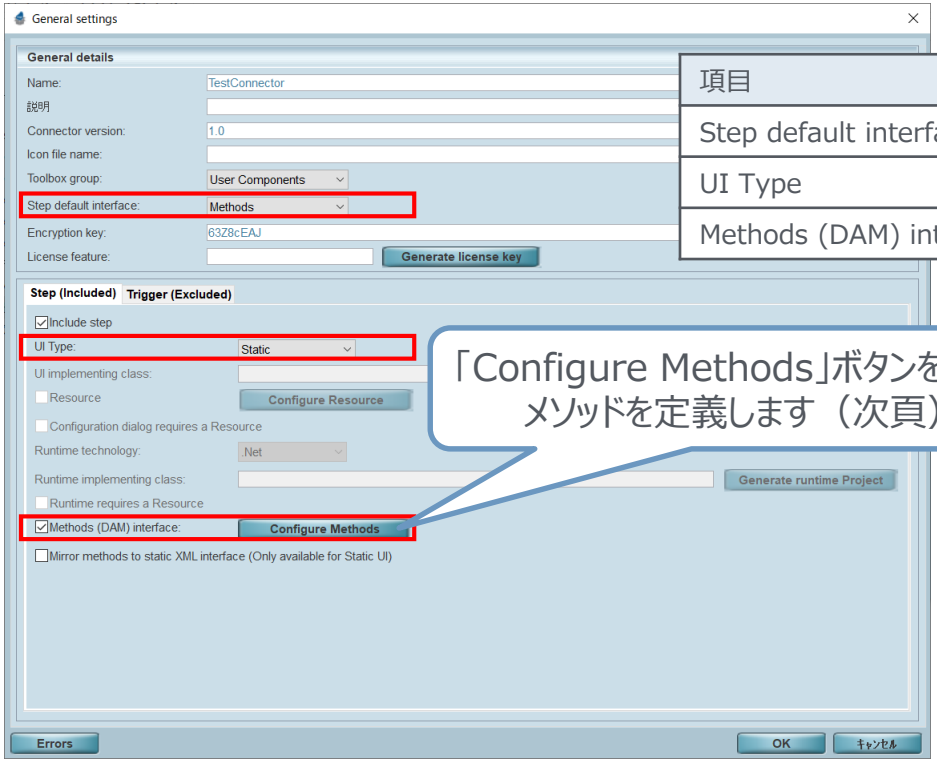
Visual Studio（.NET）によるクラスの開発例



1.3
コネクタビルダによる コネクタの作成

1.3 コネクタビルダによるコネクタの作成

- コネクタビルダでコネクタを作成します。

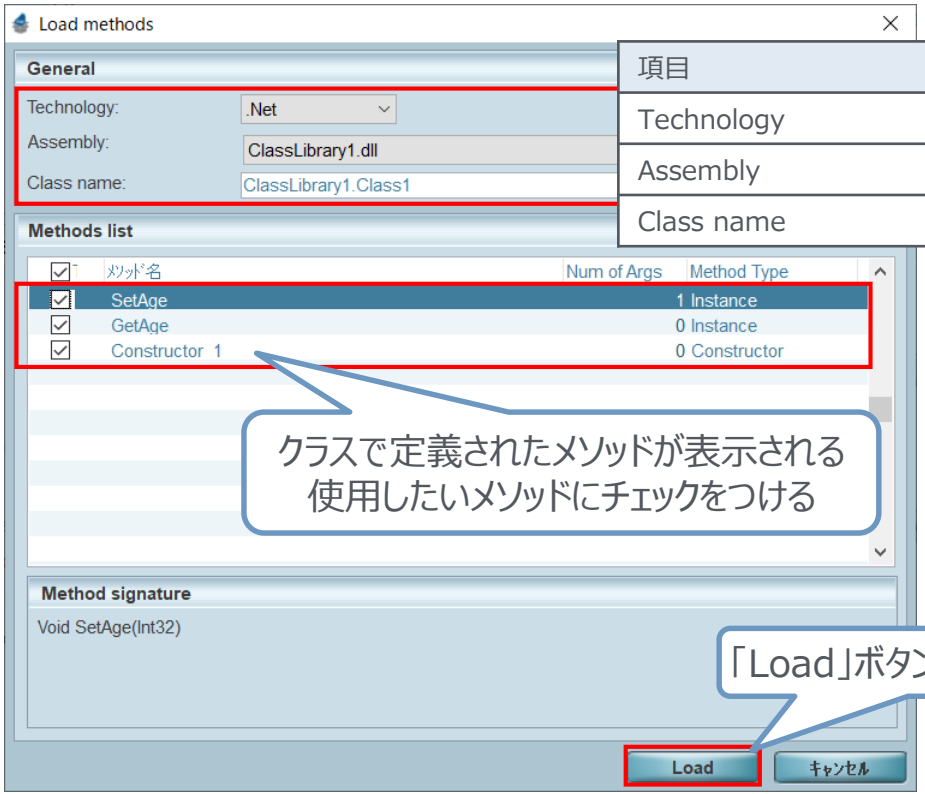


項目	値
Step default interface	Methods
UI Type	Static
Methods (DAM) interface	チェックをつける

「Configure Methods」ボタンを押し、メソッドを定義します（次頁）

1.3 コネクタビルダによるコネクタの作成

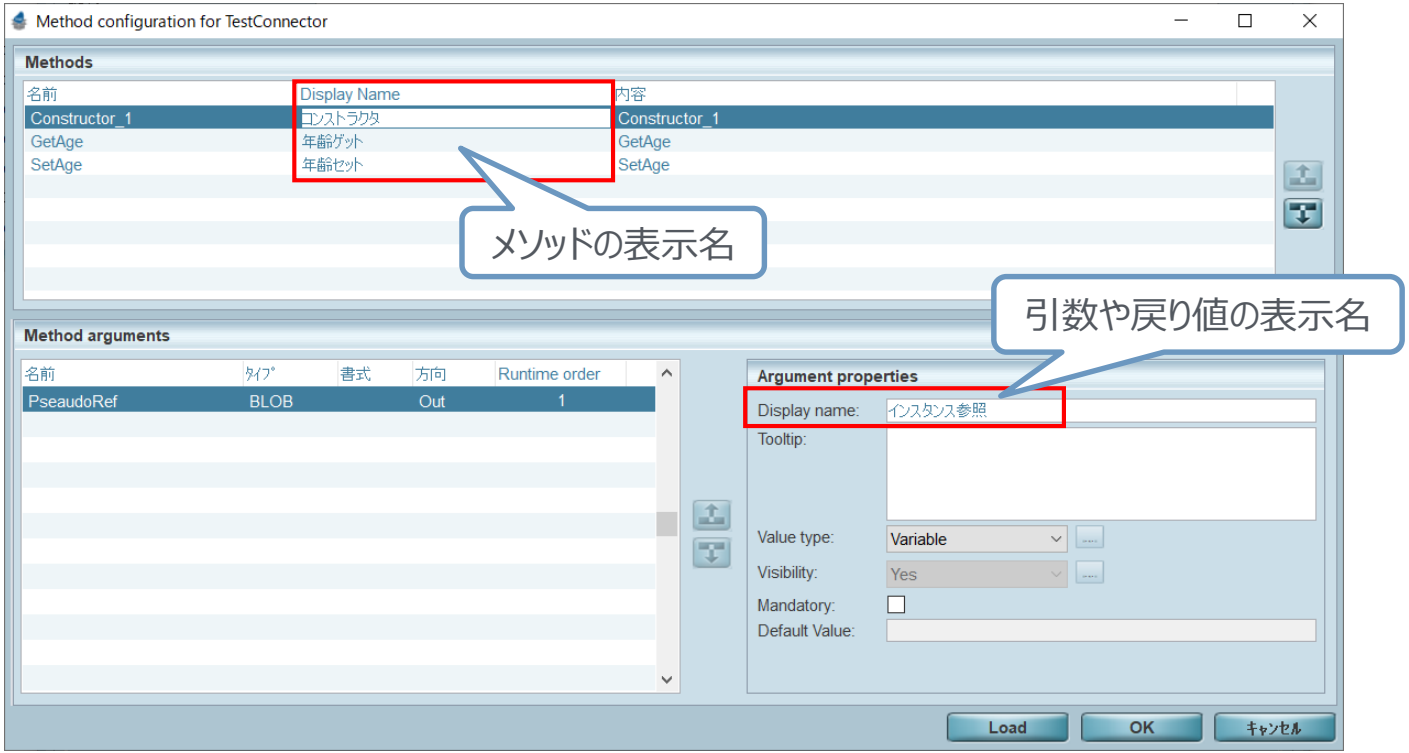
- 1.2で作成したクラスをロードします。



項目	値
Technology	.Net または Java
Assembly	dll または jar
Class name	ネームスペース.クラス名

1.3 コネクタビルダによるコネクタの作成

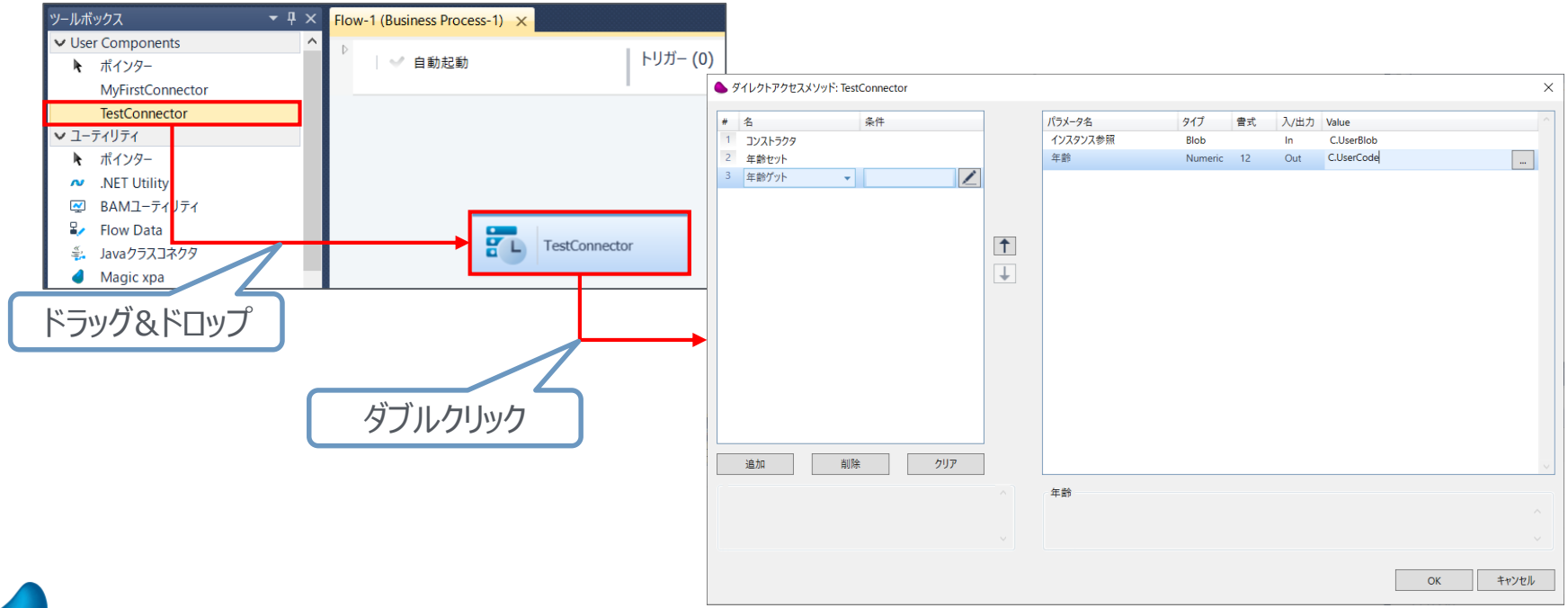
- 各メソッドごとに、メソッド、引数、戻り値の表示名を設定します。



1.4
Magic xpiスタジオにおけるコネクタの使用

1.4 Magic xpiスタジオにおけるコネクタの使用

- xpiスタジオを開き、作成したコネクタをフローエリアにドラッグ&ドロップします。
- コネクタをダブルクリックし、設定画面を開くと、コネクタビルダで定義したメソッドが使用できます。



第2章

Data Mapperインタフェース

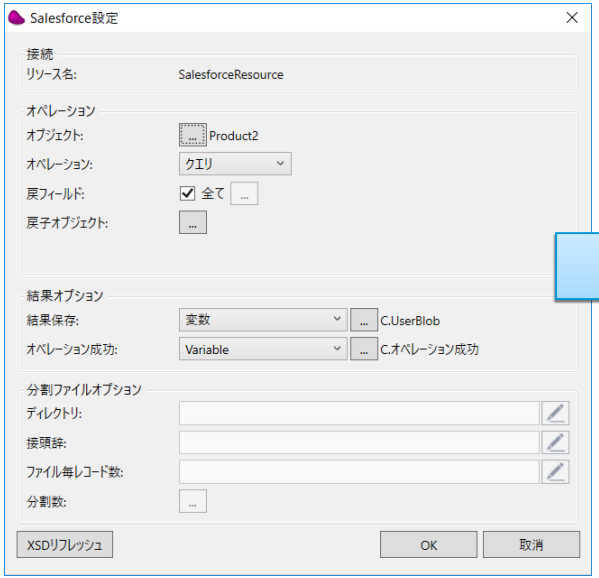
コネクタの作成方法

2.1
Data Mapper
インタフェースについて

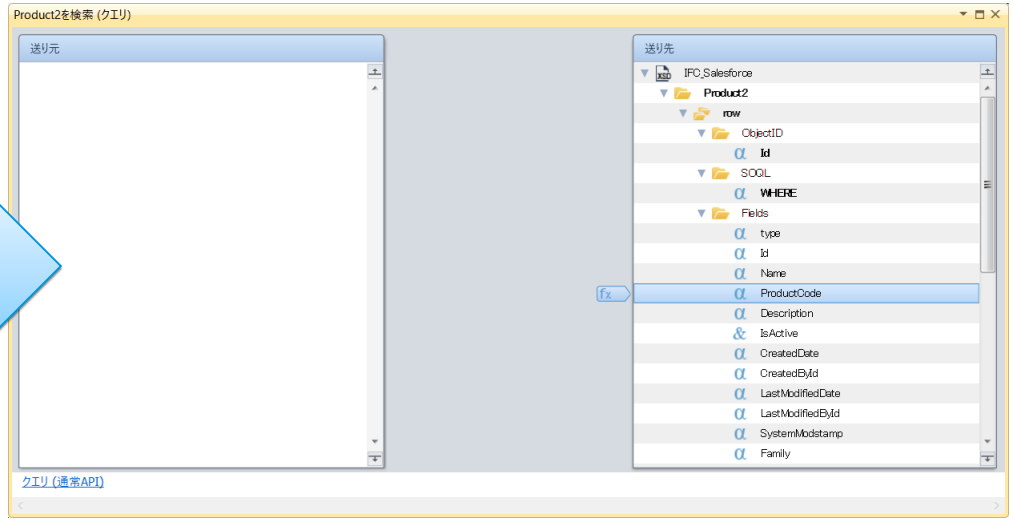
2.1 Data Mapperインタフェースについて

- xpi標準のSalesforceコンポーネントなどに見られるインタフェース。
- 設定画面で設定後、データマッパーを使用して設定を行います。

設定画面



データマッパー



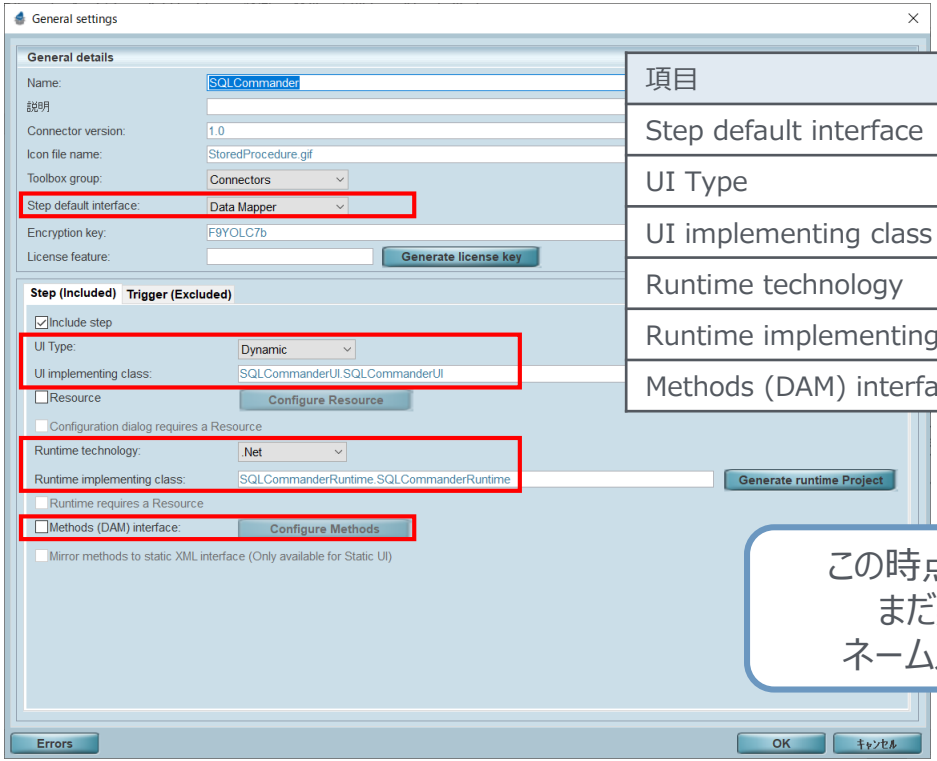
2.1 Data Mapperインタフェースについて

- Data Mapperインタフェースコネクタを作成する場合、独自の設定画面の開発が必要です。
  設定画面（UIクラス）の開発には.NETを使用します。
- データマッパーの送り先には、XML、JSON、FlatFileのいずれかを選択することができ、
  XMLやJSONの場合はスキーマファイルを準備、FlatFileの場合は区切文字や項目の名前、型
  などをプログラムの中で定義する必要があります。
- 実行時の動作を実装したRuntimeクラスを作成する必要があります。
  Runtimeクラスの開発には.NET、Java、Magic xpaのいずれかを使用することができます。

2.2
コネクタビルダによるコネクタの作成

2.2 コネクタビルダによるコネクタの作成

- コネクタビルダでコネクタを作成します。



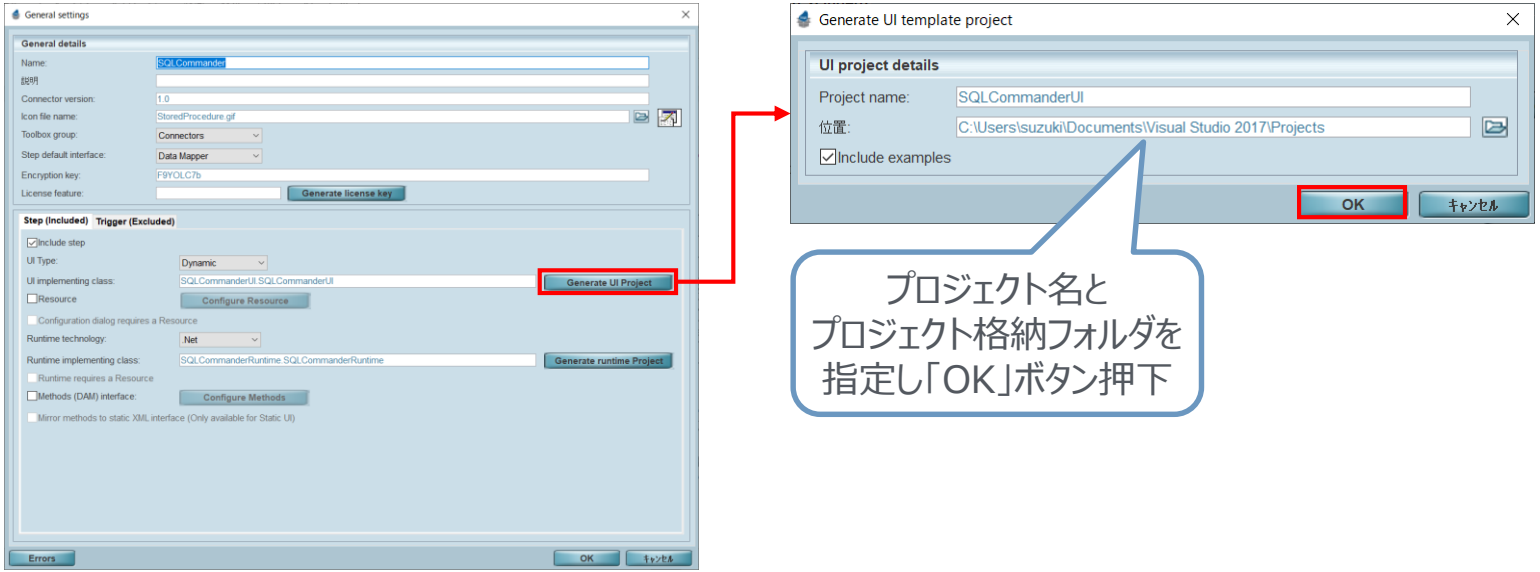
項目	値
Step default interface	Data Mapper
UI Type	Dynamic
UI implementing class	ネームスペース名.UIクラス名
Runtime technology	.Net または Java または Magic xpa
Runtime implementing class	ネームスペース名.Runtimeクラス名
Methods (DAM) interface	チェックをはずす

この時点では、UIクラス、Runtimeクラスは
まだ実装できていなくて大丈夫です。
ネームスペース名とクラス名だけ決めます。

2.3
UIクラスの作成

2.3 UIクラスの作成

- 「Generate UI Project」ボタンを押し、UIクラスのひな型を作成します。



2.3 UIクラスの作成

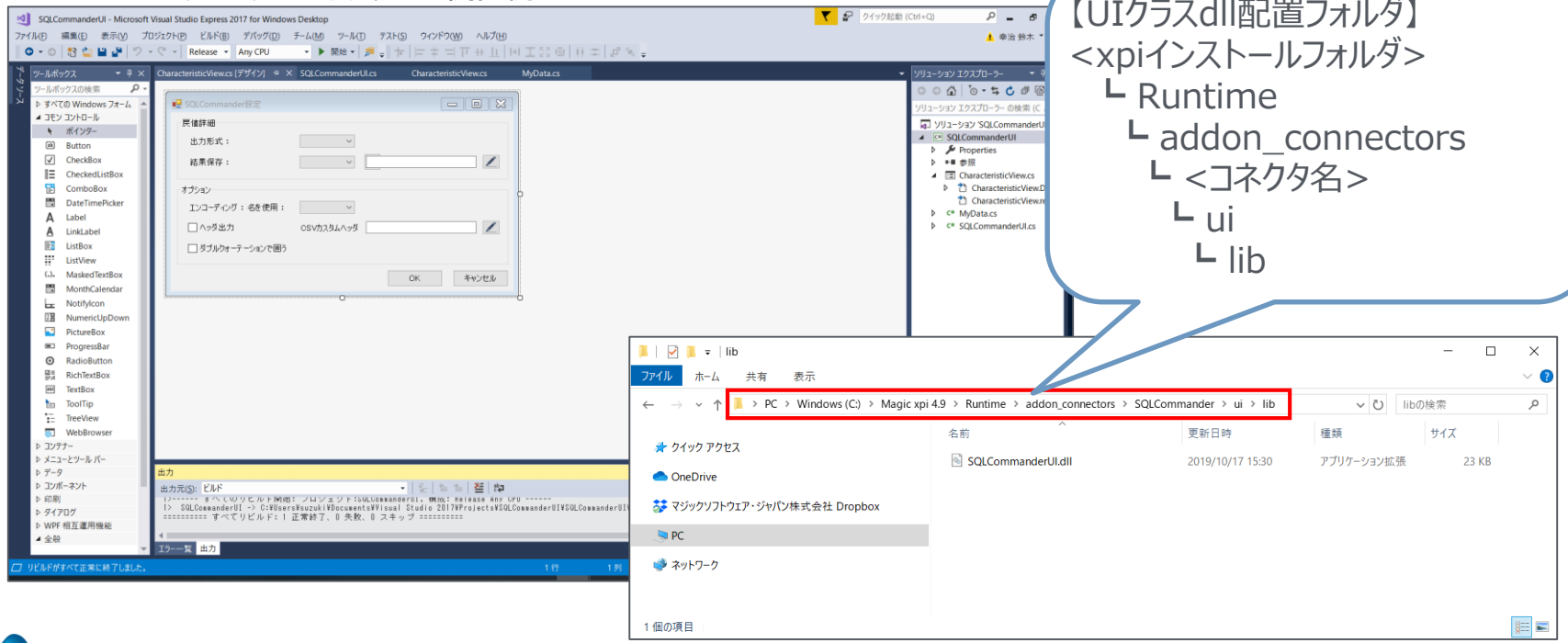
- サンプルコードつきの下記のクラスやメソッドが作成されますので、コードを修正します。
- 設定画面は作成されないので、Windowsフォームで別途作成する必要があります（下記赤字部分）。

クラス	メソッド	役割	MVCモデル
MyData	コンストラクタ	設定画面で設定した内容を保持するためのクラス	Model
Windowsフォーム	コンストラクタ	ひな型が作成されないので、Windowsフォームで設定画面を作成する	View
	ConfigurationChanged	設定画面で設定が変更されたか否かを保持するプロパティ	
	ConfigurationSuccess	設定画面で設定した内容が妥当であるか否かを保持するプロパティ	
コネクタビルダで入力したUIクラス	コンストラクタ	設定画面やマッパー画面を開くときの挙動を定義するクラス	Controller
	Configure	設定画面を開くときの挙動を実装する 別途作成したWindowsフォームをインスタンス化し、ShowDialogメソッドで設定画面を開く	
	GetSchema	マッパー画面を開くときの挙動を実装する スキーマファイルを読み込むなどの方法で、送り先の項目を一覧表示する	

2.3 UIクラスの作成

- UIクラスをビルドし、dllを所定のフォルダに配置します。

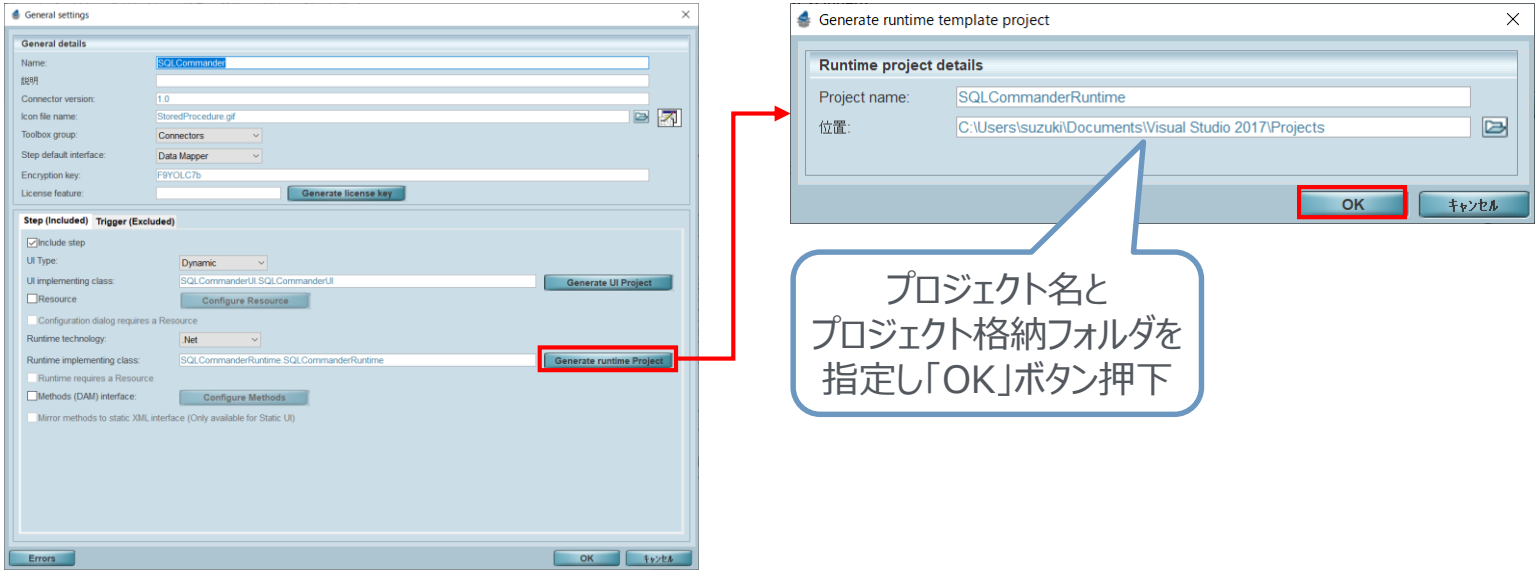
Visual Studio（.NET）によるクラスの開発例



2.4
Runtimeクラスの作成

2.4 Runtimeクラスの作成

- 「Generate runtime Project」ボタンを押し、Runtimeクラスのひな型を作成します。



2.4 Runtimeクラスの作成

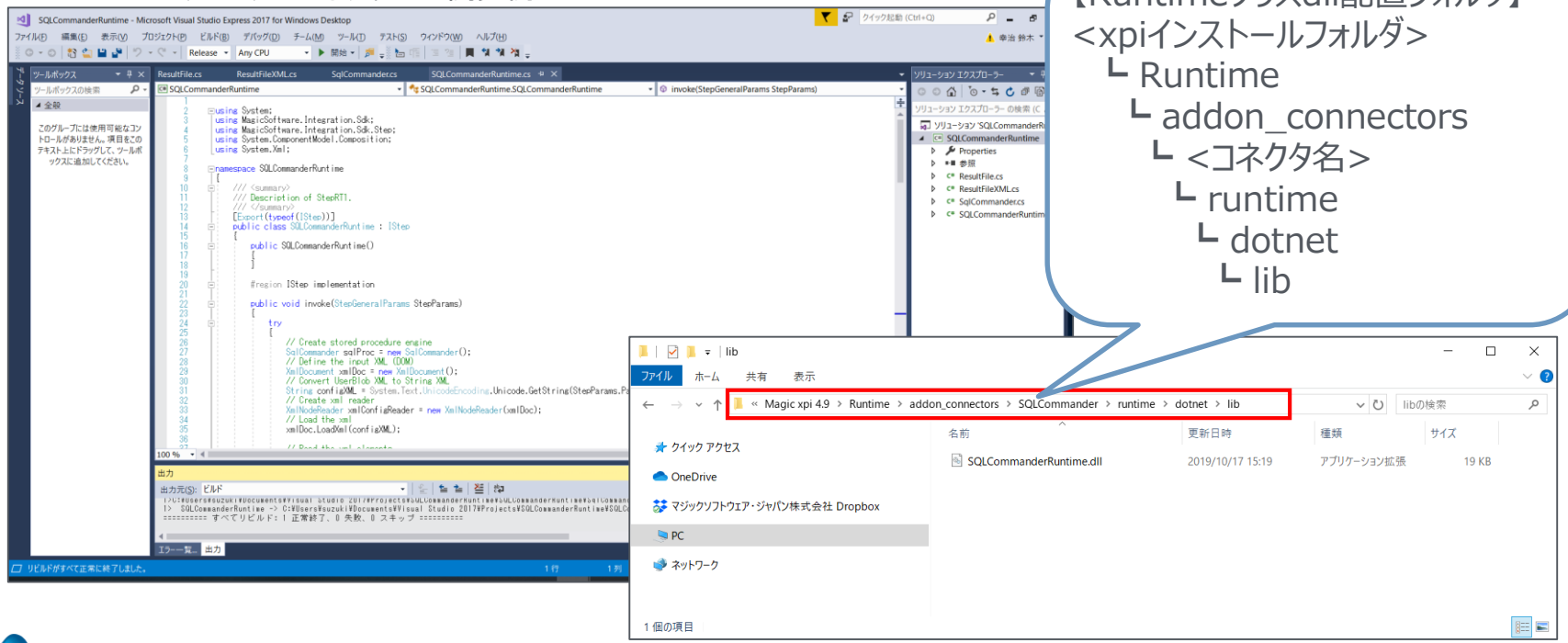
- サンプルコードつきの下記のクラスやメソッドが作成されますので、コードを修正します。
- 必要に応じて他のクラスを作成し、オブジェクト指向でコーディングすることができます。

クラス	メソッド	役割
コネクタビルダで入力したRuntimeクラス	コンストラクタ	コネクタが実行されたときの動作を定義するクラス
	invoke	コネクタが実行されたときの動作を実装する

2.4 Runtimeクラスの作成

- Runtimeクラスをビルドし、dllを所定のフォルダに配置します。

Visual Studio（.NET）によるクラスの開発例



2.4 Runtimeクラスの作成

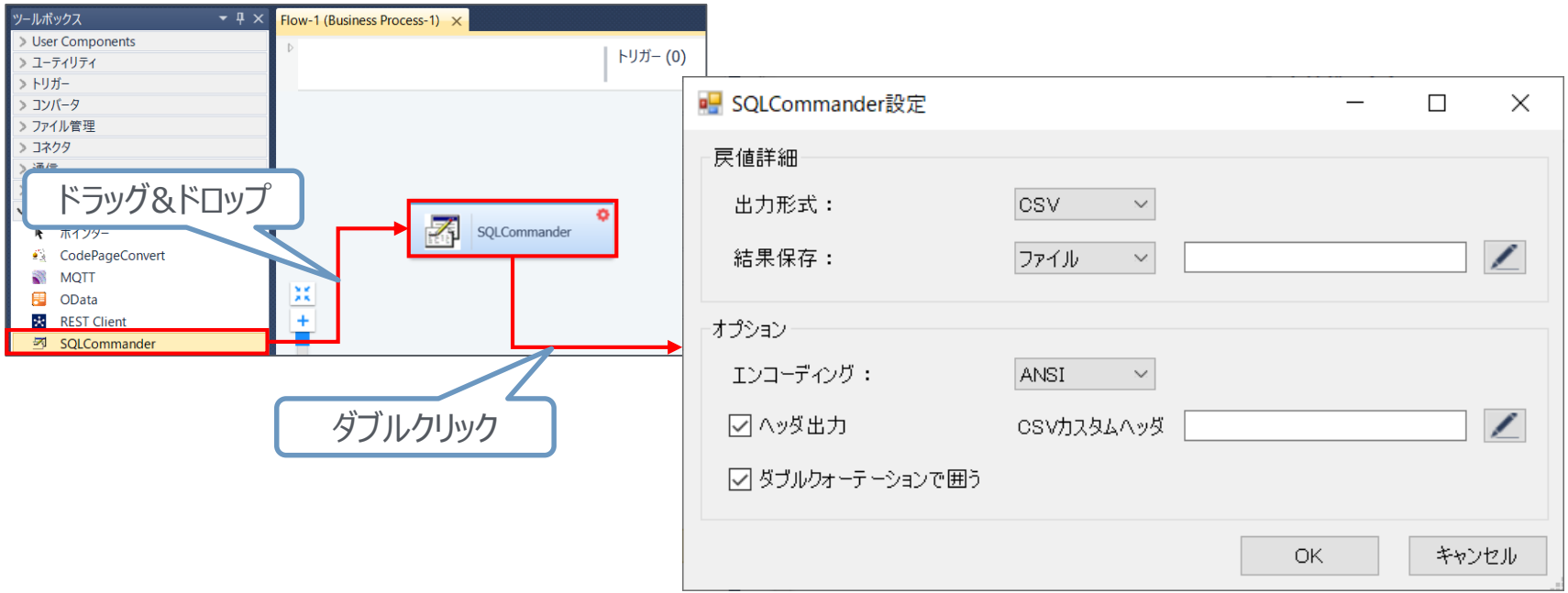
- Runtimeクラスの開発には.NET、Java、Magic xpaのいずれかを使用することができます。
- 使用する言語に応じて、ビルドで作成されるファイルや配置フォルダが異なります。

開発言語	ファイル	配置フォルダ
.NET	dll	<xpiインストールフォルダ>¥Runtime¥addon_connectors¥<コネクタ名>¥runtime¥dotnet¥lib
Java	jar	<xpiインストールフォルダ>¥Runtime¥addon_connectors¥<コネクタ名>¥runtime¥java¥lib
Magic xpa	ecf	<xpiインストールフォルダ>¥Runtime¥addon_connectors¥<コネクタ名>¥runtime¥magic¥lib

2.5
Magic xpiスタジオにおけるコネクタの使用

2.5 Magic xpiスタジオにおけるコネクタの使用

- xpiスタジオを開き、作成したコネクタをフローエリアにドラッグ&ドロップします。
- コネクタをダブルクリックすると、Windowsフォームで作成した設定画面が開きます。



2.5 Magic xpiスタジオにおけるコネクタの使用

- 設定画面で設定後、OKボタンを押すとマッパー画面が開きます。
- マッパー画面では任意の送り元から、または計算値を使用して、送り先の値を設定します。