

Visual COBOL チュートリアル

JavaEE アプリケーションから

ネイティブ COBOL サービスの呼出し

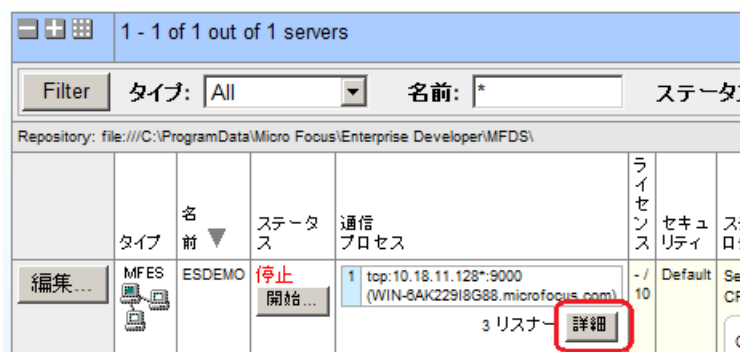
3. Web サービスとしてのアクセス

チュートリアル 2 では、Visual COBOL 製品の IDE である Enterprise Developer が装備する Interface Mapping パースペクティブを使用して Java EE 向け Connector Architecture に基づくパッケージを作成し稼働させました。Interface Mapping パースペクティブを使用すると、このほかに COBOL サービスを Web サービスとして稼働させることもできます。これによって Java EE のみならず .NET までをも含む広範なクライアントから COBOL サービスが利用可能となります。

3.1 サーバーの準備

チュートリアル 3 では、チュートリアル 2 で使用したプロジェクトと COBOL プログラムをそのまま使用し、同じプログラムを Web サービスとしてディプロイします。Web サービスとしてディプロイするために ESDEMO サーバーの設定を一部変更します。

- 1) ESDEMO が停止状態になっていることを確認してください。
- 2) Enterprise Server 管理コンソールにて ESDEMO サーバーのリスナーの [詳細...] ボタンをクリックします。



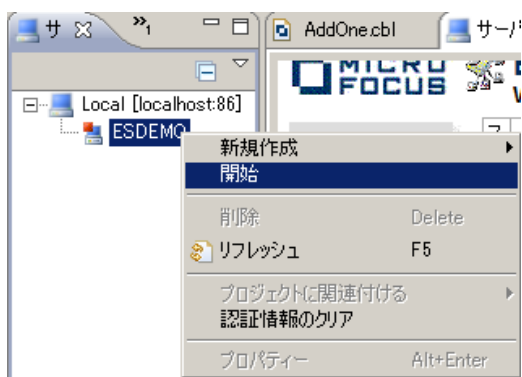
- 3) 以下のようにリスナーの一覧が表示されます。その中から Web リスナーの [編集...] ボタンをクリックします。

通信プロセス 1 ☒ 自動起動					
	リスナー	プロセスID	コントロールチャンネルアドレス	ステータス	前
編集...	3 追加	-	top:10.18.11.128*:9000 (WIN-8AK22918G88.microfocus.com)	停止	10
	名前	アドレス	ステータス	前回のステータス変更	ステータス
編集...	HTTP Echo	top:10.18.11.128*:9002 (WIN-8AK22918G88.microfocus.com)	停止	10/30/12-13:45:46	OK
編集...	Web	top:10.18.11.128*:81554* (WIN-8AK22918G88.microfocus.com)	停止	10/30/12-13:45:47	OK
編集...	Web Services and J2EE	top:10.18.11.128*:9003 (WIN-8AK22918G88.microfocus.com)	停止	10/30/12-13:45:47	OK

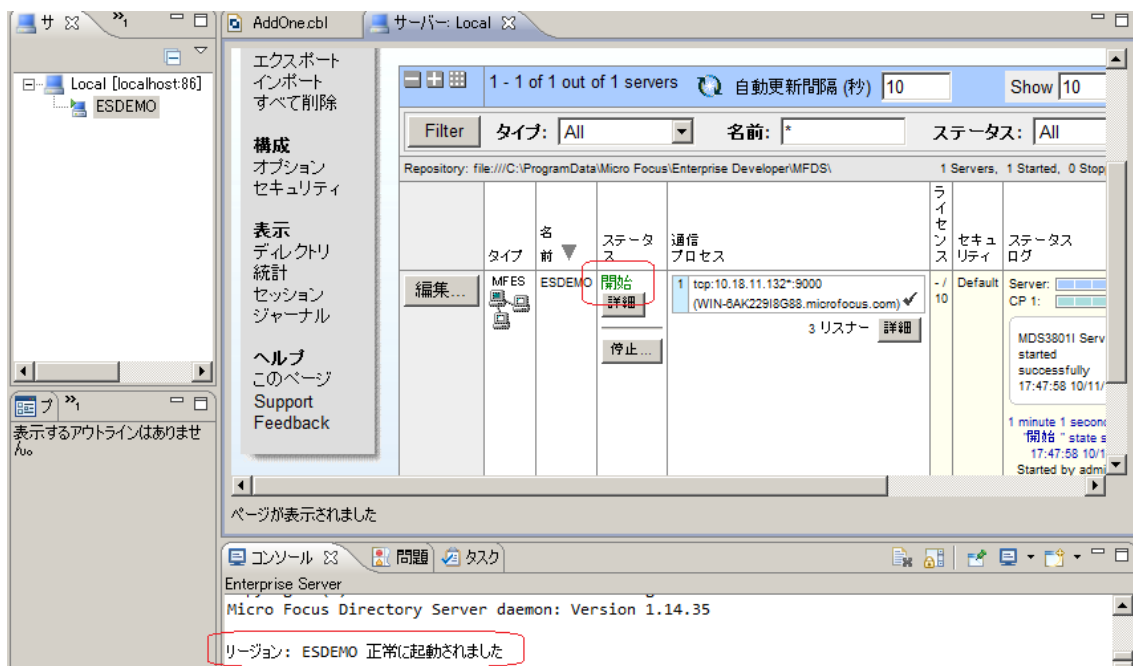
- 4) 以下のように Web リスナーのエンドポイントアドレスを *:9004 に設定し [OK] をクリックします。

サーバー...	リスナー (3)	サービス (7)	ハンドラ (2)	パッケージ (2)
名前 Web				
エンドポイントプロトコル TCP				
エンドポイントアドレス *:9004				

- 5) [サーバーエクスプローラー]内で [ESDEMO]を右クリックして [開始] を選択します。

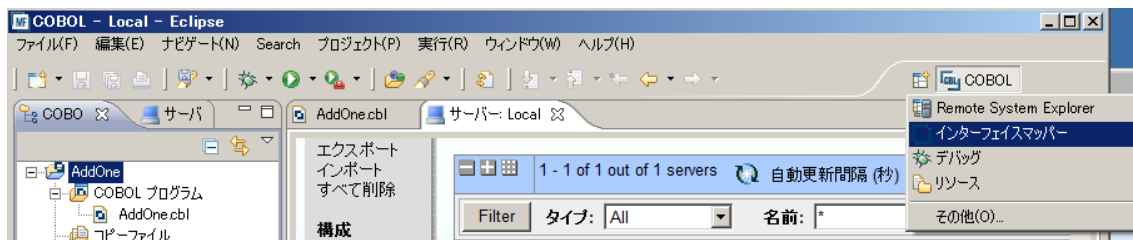


- 6) しばらく待つと以下のように ESDEMO が開始状態となります。

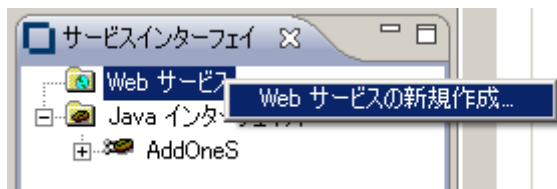


3.2 COBOL サービスのディプロイ

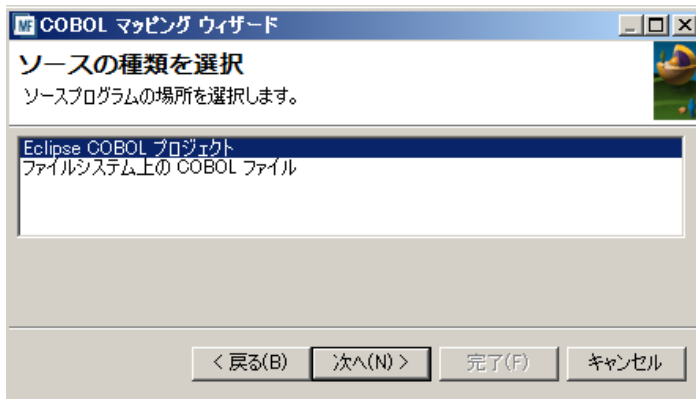
- 1) IDE の右上のパースペクティブ選択にて、[インターフェイスマッパー] パースペクティブを選択します。



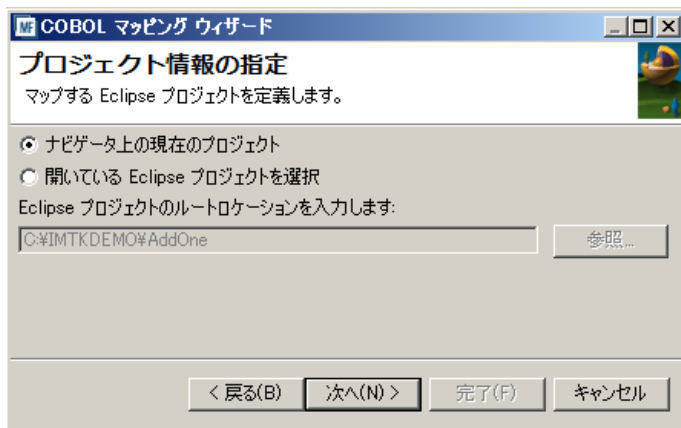
- 2) 以下のように COBOL エクスプローラ内で AddOne プロジェクトを選択した状態で、インターフェイスマッパービュー内で [Web サービス] を右クリックし、[Web サービスの新規作成...] を選択します。



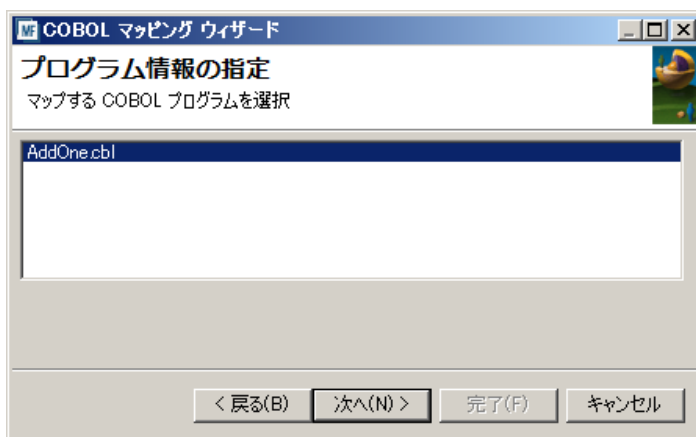
- 3) COBOL マッピングウィザードが開始します。まず、最初に以下のダイアログで [Eclipse COBOL プロジェクト] を選択し [次へ] をクリックします。



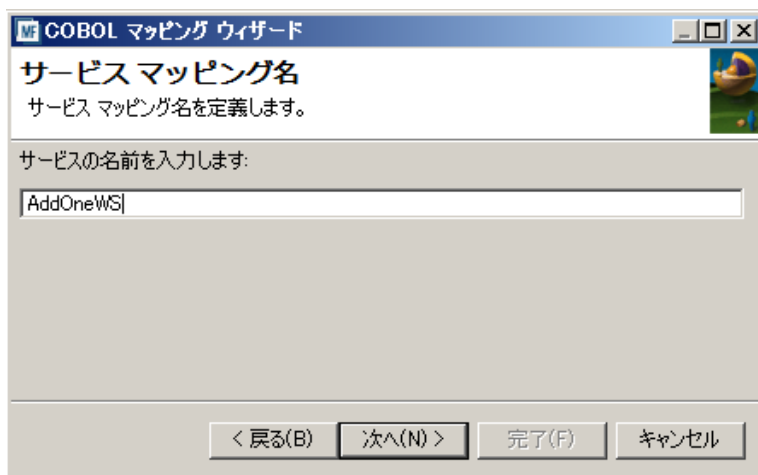
- 4) 以下のダイアログではデフォルトの [ナビゲータ上の現在のプロジェクト] を選択し [次へ] をクリックします。



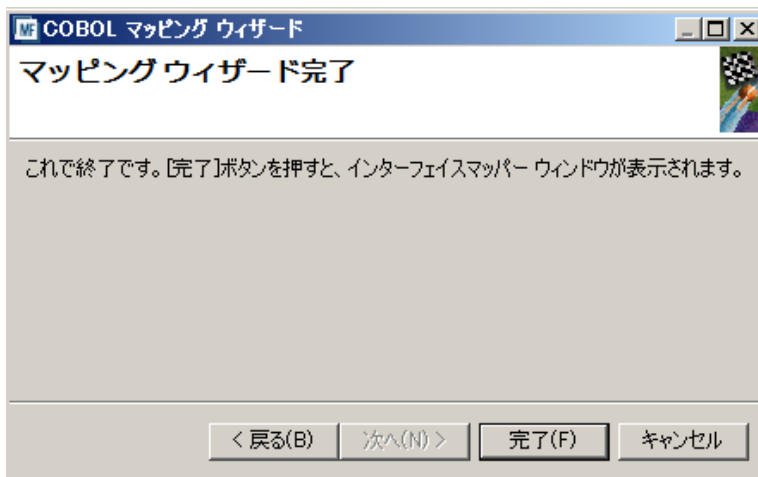
- 5) 以下のダイアログではデフォルトの [AddOne.cbl] を選択し [次へ] をクリックします。



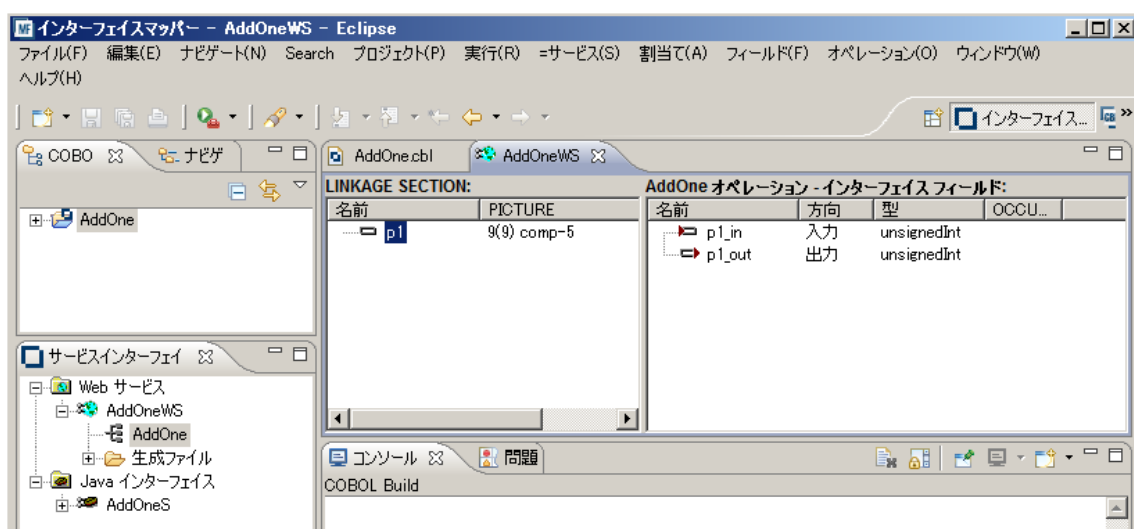
- 6) 以下のダイアログではサービス名としての AddOneWS を入力して [次へ] をクリックします



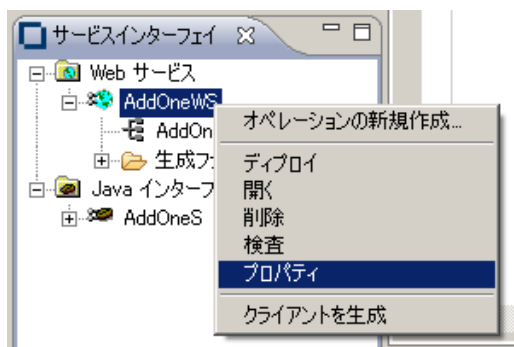
7) 以下のダイアログで [完了] をクリックします。



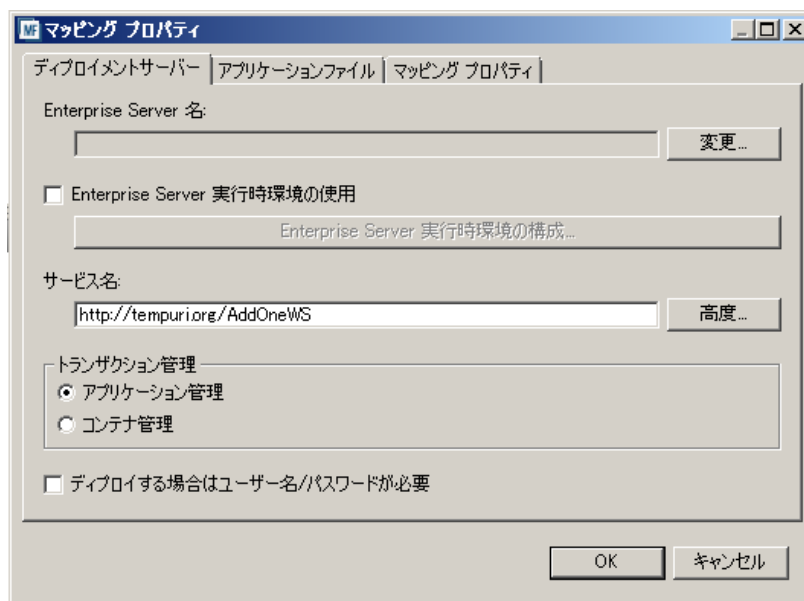
8) 以下のインターフェイスマッパーウィンドウが表示されます。



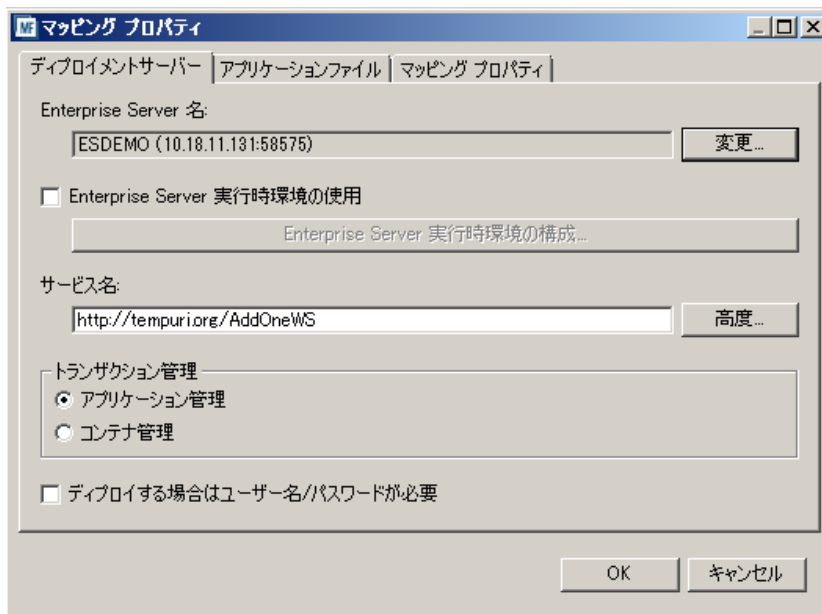
- 9) このチュートリアルではデフォルトで表示されているままのマッピングを使用しますので、このウィンドウはそのまま閉じます。左下の [サービスインターフェイス] ビュー内に作成されている [AddOneWS] を右クリックして [プロパティ] を選択します。



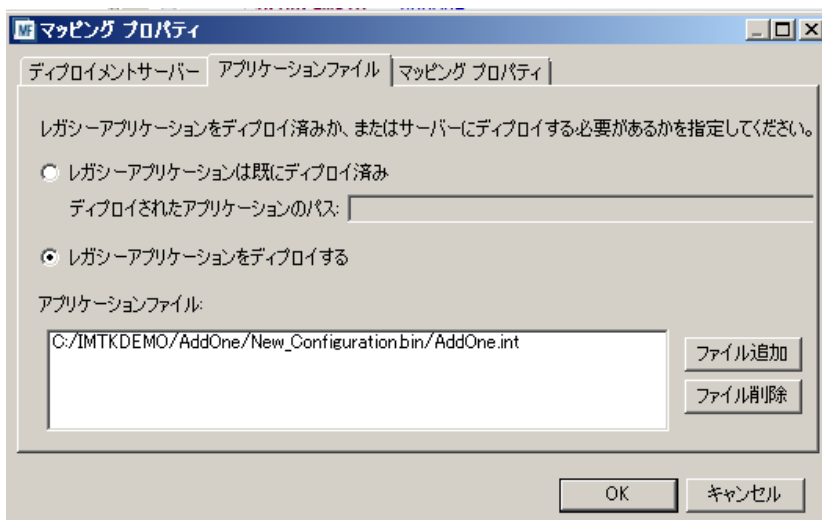
- 10) 以下の [ディプロイメントサーバー] タブで [Enterprise Server 名] の右の [変更...] ボタンをクリックし、開始している ESDEMO を選択します。



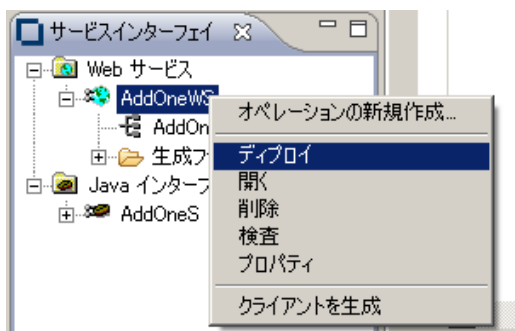
- 11) 以下のように ESDEMO が設定されたことを確認し、[アプリケーションファイル] タブに移動します。



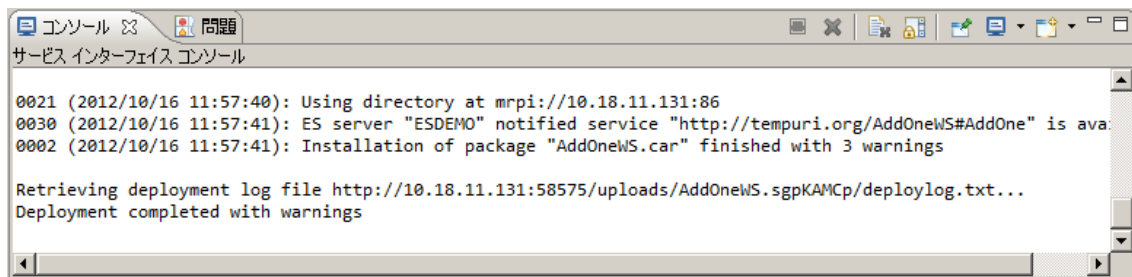
- 12) 以下のようにプロジェクトの bin ディレクトリ下に生成されている AddOne.int を選択し [OK] をクリックします。



- 13) [AddOneWS] を右クリックして [デプロイ] を選択します。

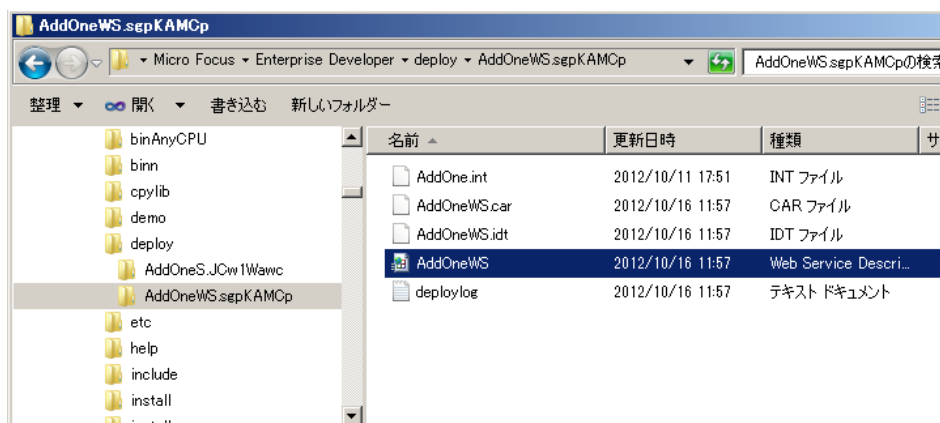


14) コンソールに以下のようにディプロイが完了した旨のメッセージが表示されます。

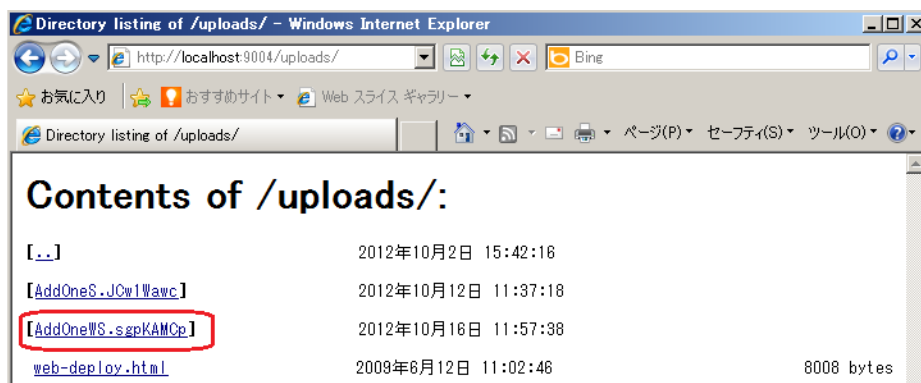


3.3 ディプロイされたサービスの WSDL 確認

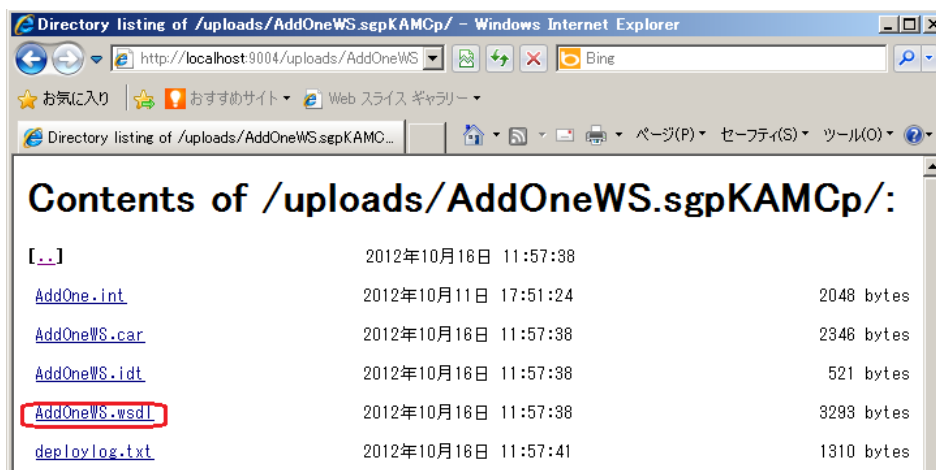
1) Enterprise Server 管理コンソール上に AddOneWS サービスが追加されていることを確認し、以下のようにパッケージパスの下に WSDL が生成していることを確認します。



2) この WSDL は ESDEMO サーバーが装備する Web リスナーによって HTTP でエクスポートされています。本チュートリアル之初にこのリスナーのポートを 9004 に設定しておいたことを思い出してください。Internet Explorer を開き <http://localhost:9004/uploads/> を開きます。以下のようにディプロイ先ディレクトリが参照できます。この中から今ディプロイした AddOneWS パッケージのサブディレクトリを選択してクリックします。ディレクトリ名は毎回変わります。



- 3) 以下の通り、パッケージのディプロイ先ディレクトリには自動生成された WSDL があります。これをクリックします。



- 4) 以下のように WSDL の内容が Web 経由で参照できます。これを使用して世界中のクライアントからインターネットを経由してこの COBOL サービスを呼び出すことができます。

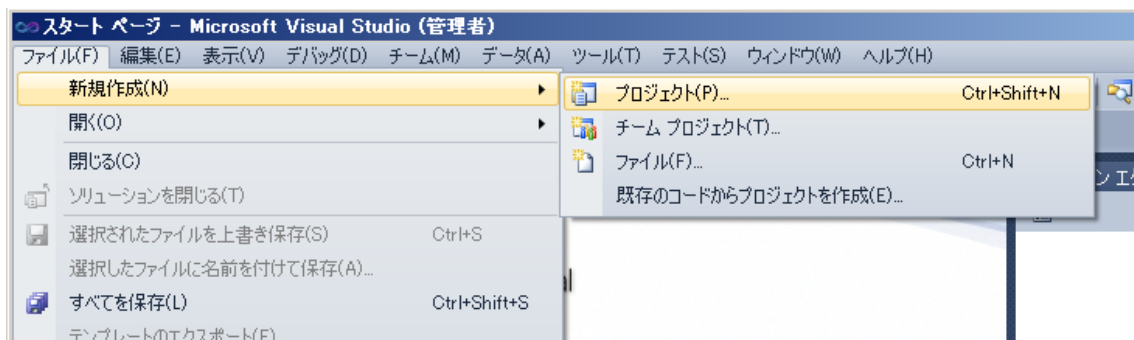


```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
- <definitions xmlns="http://schemas.xmlsoap.org/wsdl/" name="AddOneWS"
  targetNamespace="http://tempuri.org/AddOneWS"
  xmlns:flt="http://xml.microfocus.com/cobol/mapperfault"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://tempuri.org/AddOneWS"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:ws="http://ws-
i.org/schemas/conformanceClaim/" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <!-- Micro Focus Enterprise Developer - This generated WSDL document is provided
  'as is' and without any warranties of any kind, and may be used by licensee solely for
  <!-- Micro Focus shall not be responsible for and hereby disclaims any
  liabilities that may result from licensee's use of or modifications to this generated W
- <types>
- <schema elementFormDefault="qualified"
  targetNamespace="http://tempuri.org/AddOneWS"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://tempuri.org/AddOneWS">
- <element name="AddOne">
- <complexType>
- <sequence>
  <element name="p1_in" type="unsignedInt" />
</sequence>
</complexType>
</element>
</schema>
</types>
</definitions>
```

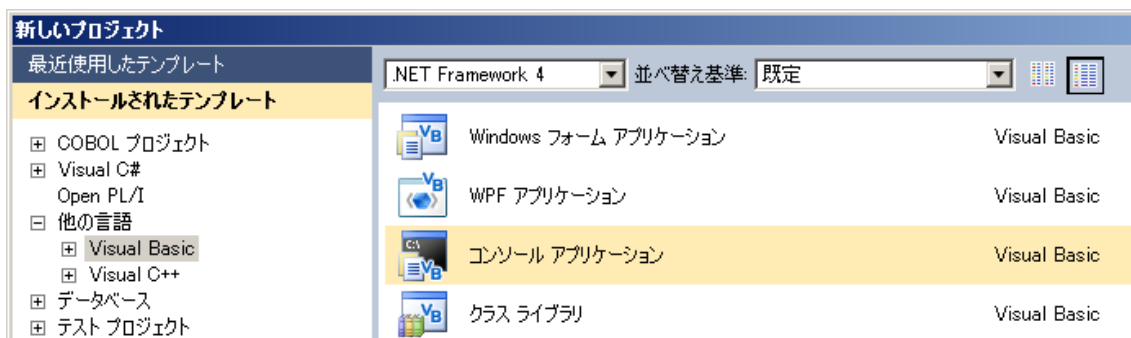
3.4 .NET クライアントからの呼出し

Web サービスとしてデプロイした COBOL サービスは、さまざまなクライアントからアクセスすることができます。これは Visual Basic で簡単なクライアントを作成してみます。

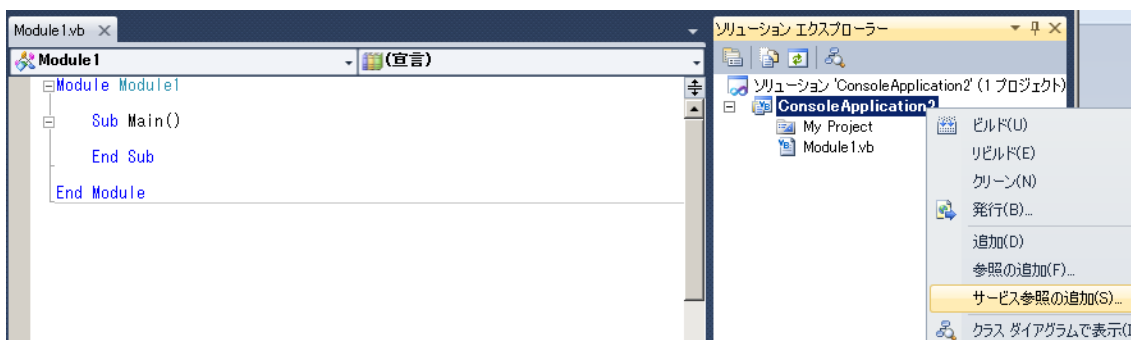
- 1) Windows スタートメニューから Visual Studio 2010 を起動し、[ファイル] > [新規作成] > [プロジェクト] を選択します。



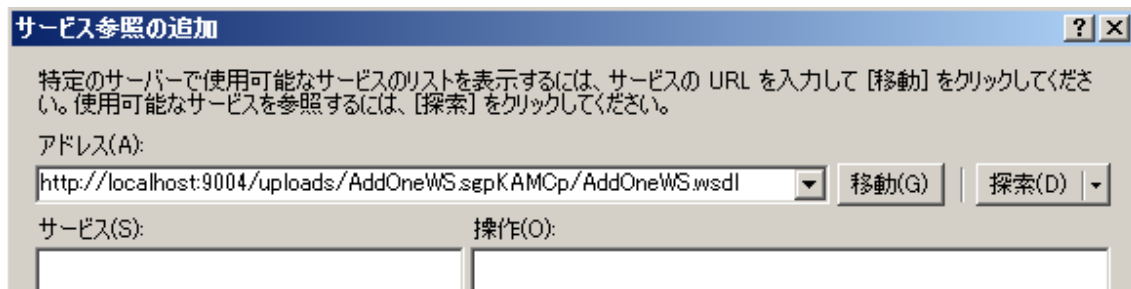
- 2) 以下のように Visual Basic のコンソールアプリケーションのテンプレートを選択し [OK] をクリックします。



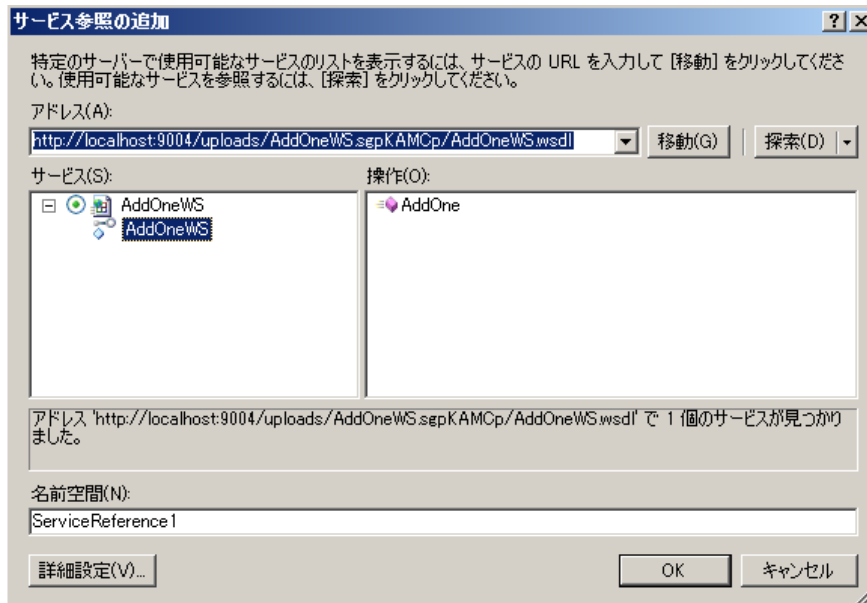
- 3) Visual Basic のプロジェクトテンプレートが生成されます。ソリューションエクスプローラー内でプロジェクトを右クリックし、[サービス参照の追加...] を選択します。



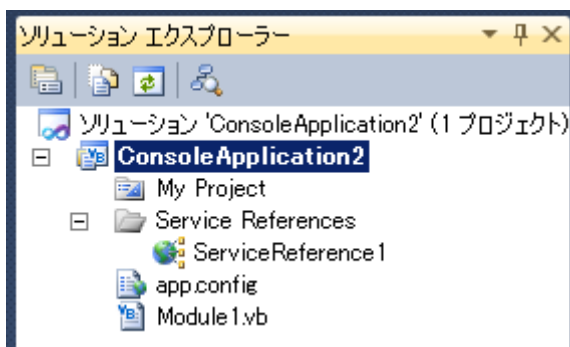
- 4) 以下のダイアログで、[アドレス] に先ほど Internet Explorer で開いた WSDL の URL をコピー・ペーストします。



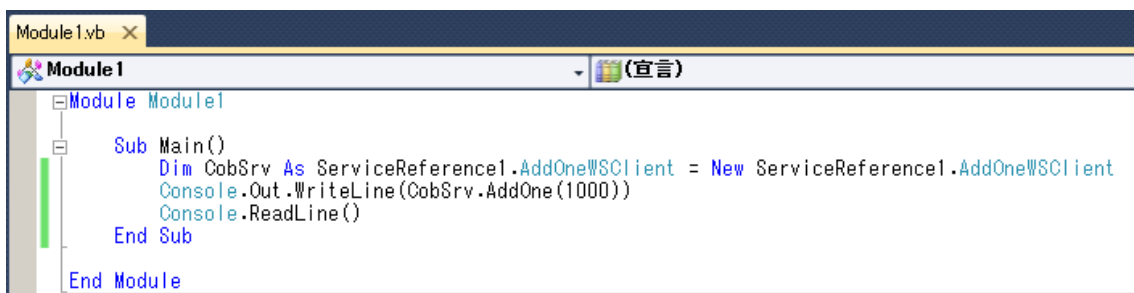
- 5) [移動] ボタンをクリックして WSDL を解析させます。以下のように COBOL サービスの定義内容がオペレーションとして見えています。



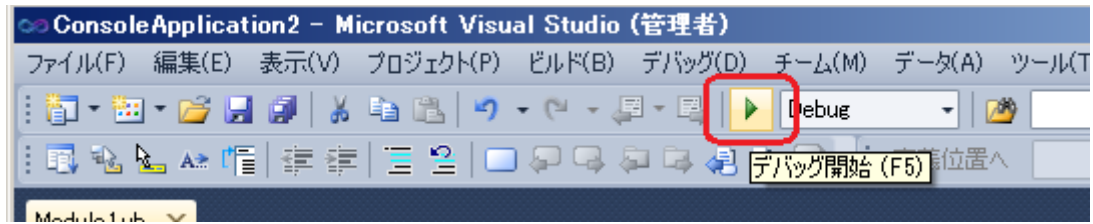
- 6) [OK] をクリックするとソリューションエクスプローラー内にサービス参照が追加されます。これでこのプロジェクトの中から COBOL Web サービスが呼び出し可能になります。



- 7) テンプレート生成されている Module1.vb に以下のようなコードを書いてください。すべてをそのまま打鍵しなくても Visual Studio のエディタが候補を次々と表示してくれますのでその中から選択して行くことで素早くコードを完成できます。



8) 以下の [デバッグ開始] ボタンをクリックし実行します。



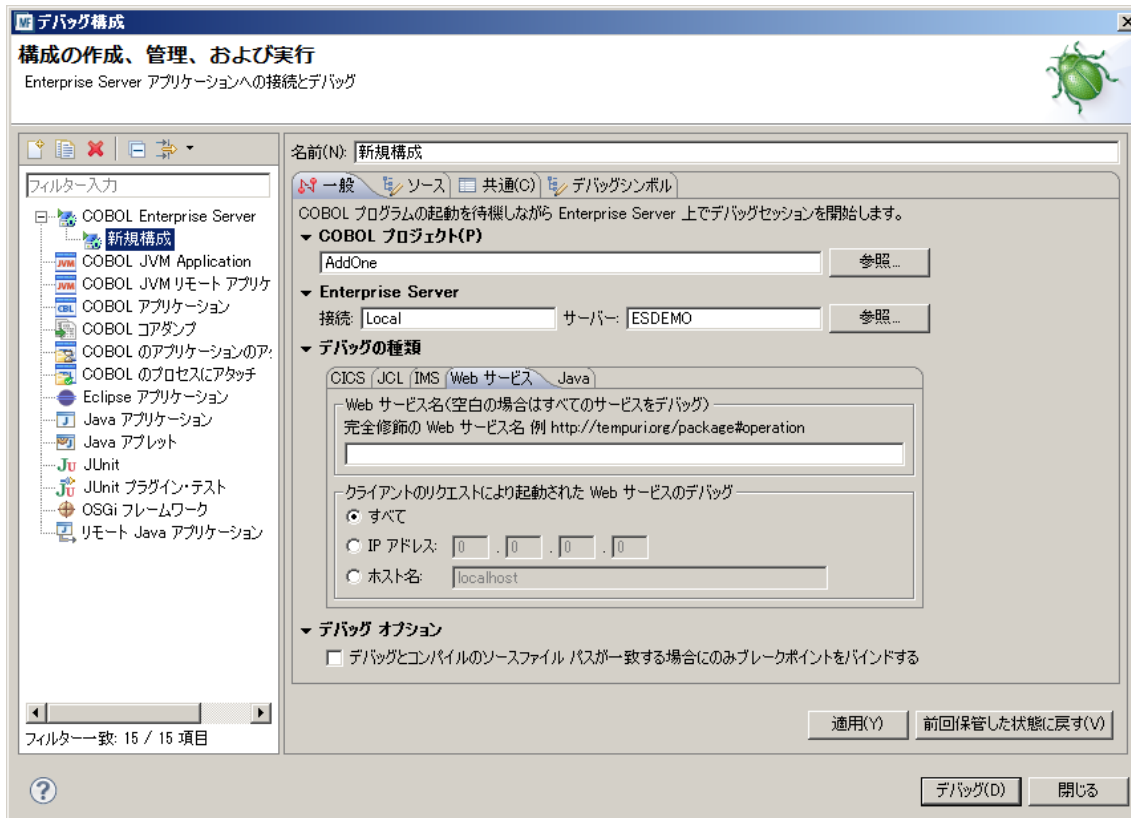
9) コンソールが開き 1000 に COBOL で 1 を加算した 1001 が表示されます。



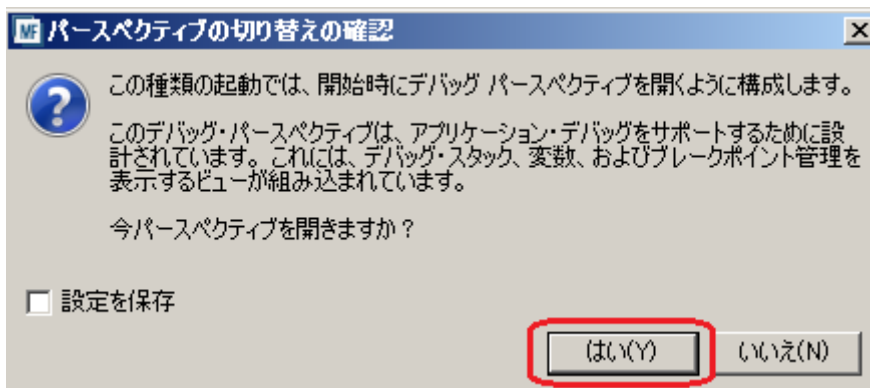
10) Enter キーを打鍵してプログラムを終了します。

3.5 COBOL Web サービスのデバッグ

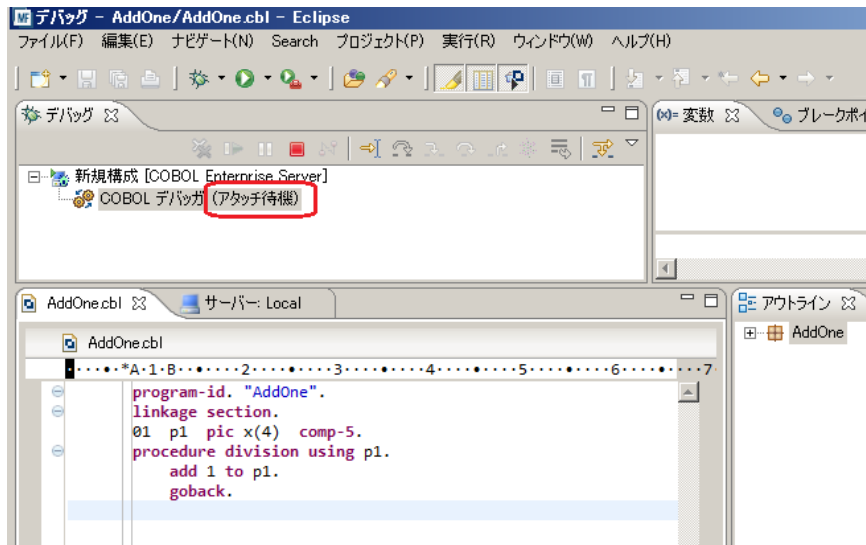
1) COBOL パースペクティブに移ります。AddOne プロジェクトを選択して [実行] > [デバッグの構成...] を選択します。以下のようにチュートリアル 2 で実施した Java インターフェ이스のデバッグ構成が保存されています。ここで [デバッグの種類] として [Web サービス] タブを選択し、[Web サービス名] は空白のままで [デバッグ] をクリックします。



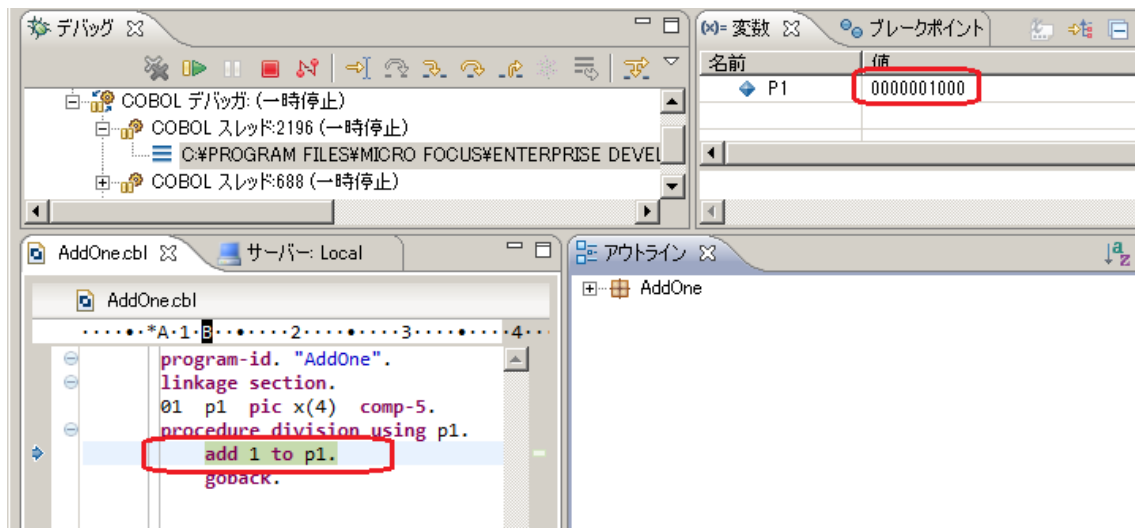
2) 以下のダイアログが現れたら [はい] をクリックします。



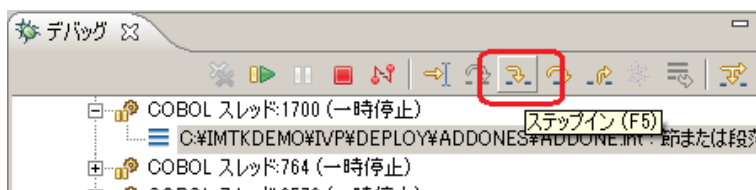
3) 以下の通りデバッグパースペクティブに移動し、「アタッチ待機」の状態になります



- 4) ここで Visual Basic のプロジェクトで再度 [デバッグ開始] ボタンをクリックし実行します。すると以下のように待機中のデバッグパースペクティブに制御が移り、以下のよう
に COBOL サービスプログラムの手続き部の先頭がハイライトされます。[変数ビュー]
内では Visual Basic アプリケーションで指定した数値 1000 がパラメタとして
COBOL に渡されていることがわかります。



- 5) 以下の [ステップイン] ボタンをクリックすると COBOL の実行文がステップ実行
されます。



- 6) ADD 文が実行されて P1 の数値が増加します。さらにもう一度 [ステップイン] ボタンをクリックすると GOBACK 文が実行され COBOL サービスの処理が終了します。このとき Visual Basic アプリケーション側に COBOL サービスの実行結果が返ります。

注) Visual Basic 側の Web サービス呼び出しがタイムアウトすることがありますが、このチュートリアルでは COBOL サービスの呼び出しができていればその目的は達成されていますので無視してください。

- 7) Visual Studio 2010 はそのまま終了します。なにも保存する必要はありません。

以上、チュートリアルを終了します。