Micro Focus Visual COBOL チュートリアル

RESTful Web サービスによる COBOL 資産の再利用

Visual Studio 編

1. 目的

Micro Focus Visual COBOL に付属する COBOL 専用のアプリケーションサーバー「Enterprise Server」は、ネイティブにコンパイルした COBOL のビジネスロジックを REST API を利用し Web サービスとして呼び出す機能を提供しています。RESTful の Web サービスとして呼び出しを行う場合、JSON 形式でやり取りが可能であれば呼び出し側のプログラムに依存することなく連携できるようになります。

このドキュメントでは COBOL のソースコードに一切手を加えることなくビジネスロジックとして Enterprise Server にディプロイし、それを Visual COBOL のクライアント生成機能を使って動作確認用のクライアントを作成し連携する方法を説明します。

2. 前提条件

本チュートリアルは、下記の環境を前提に作成されています。

- 開発クライアント ソフトウェア

OS	Windows Server 2016 Standard Edition（64bit）
COBOL 開発環境製品	Micro Focus Visual COBOL 5.0J for Visual Studio (PU1 適用版)

- チュートリアル用サンプルプログラム

  下記のリンクから事前にチュートリアル用のサンプルファイルをダウンロードして、任意のフォルダに解凍しておいてください。

  サンプルプログラムのダウンロード

内容

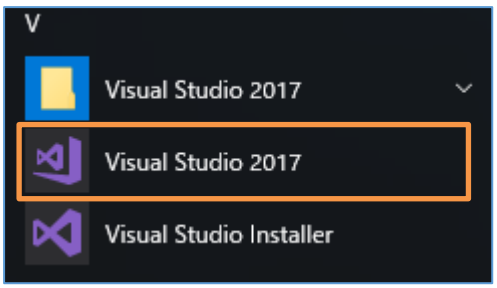
1. 目的
2. 前提条件
3. チュートリアル手順の概要
   - 3.1. Windows クライアントでの開発準備作業
   - 3.2. Enterprise Server の設定変更
   - 3.3. RESTful Web サービスの開発作業
   - 3.4. コンパイルした COBOL アプリケーションを Enterprise Server へディプロイ
   - 3.5. RESTful Web サービスのテスト
   - 3.6. インスタンスの停止

3. チュートリアル手順の概要

3.1. Windows クライアントでの開発準備作業

1) Visual COBOL for Visual Studio を起動

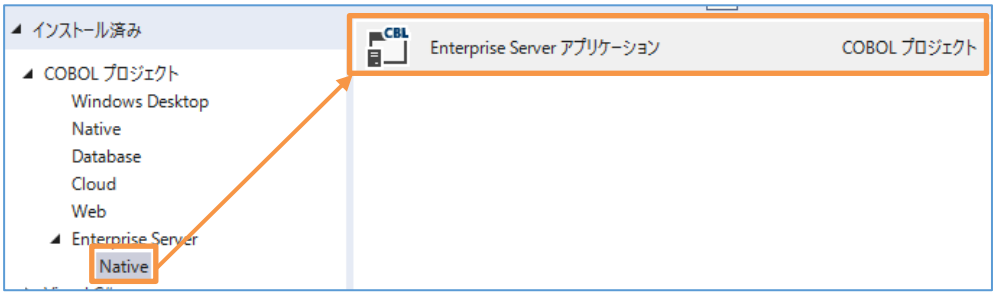
① [スタート] メニュー > [Visual Studio XXXX] を選択します。（XXXX はバージョン番号）



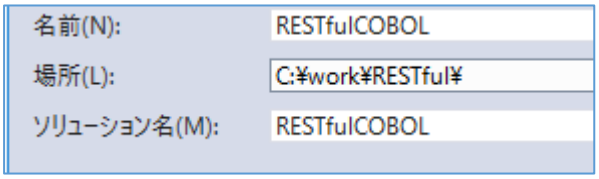
2) [Enterprise Server アプリケーション] プロジェクトの作成

① [ファイル(F)]メニュー > [新規作成(N)] > [プロジェクト(P)] を選択します。

② [COBOL プロジェクト] > [Enterprise Server] > [Native] > [Enterprise Server アプリケーション] を選択します。

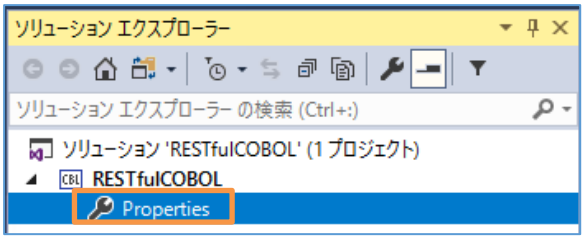


③ [名前(N)] フィールドに “RESTfulCOBOL” を入力して、[OK] ボタンをクリックします。[場所(L)] は任意のフォルダを指定します。

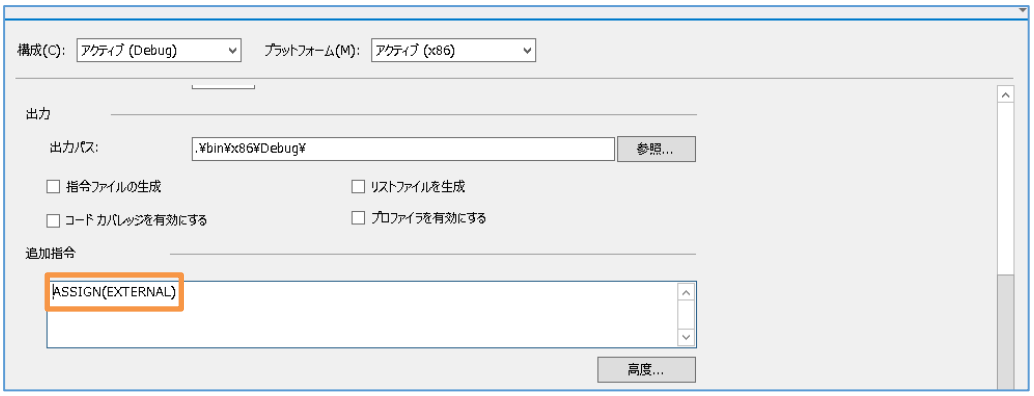


3) コンパイラオプションの指定とソースコードのインポート

① 作成されたプロジェクトの Properties をダブルクリックします。

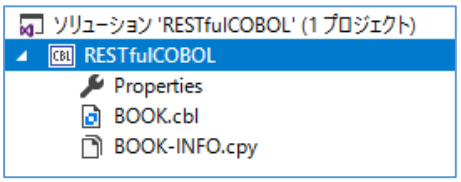


② [COBOL] をクリックし、画面を下にスクロールして、[追加指令] に “ASSIGN(EXTERNAL)” を指定し、画面を閉じます。



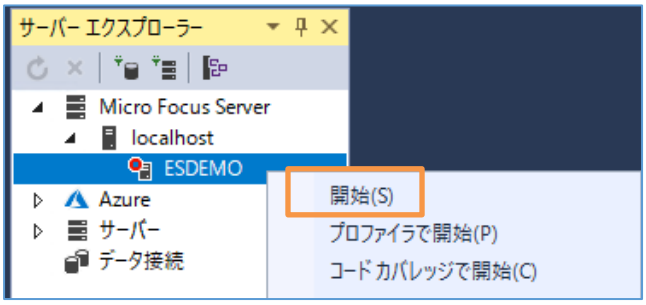
③ エクスプローラを起動し、サンプルのソースコードを解凍したフォルダから ”BOOK-INFO.cpy” と ”BOOK.cbl” をプロジェクトフォルダにドラッグアンドドロップします。

④ ソリューションエクスプローラーから、２つのファイルが正常にロードされていることを確認します。



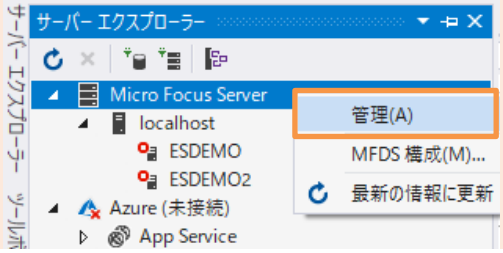
4) Enterprise Server の起動

① [サーバーエクスプローラー] ペインを選択、[Micro Focus Server] > [localhost] > [ESDEMO] と展開します。[ESDEMO] の上で右クリックし、コンテクストメニューから[開始(S)]を選択します。

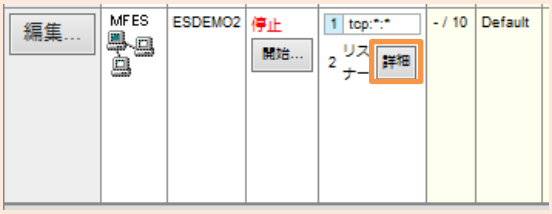


注意）
予めセットアップされた「ESDEMO」ではなく、自分で作成した Enterprise Server を使用する場合は、以下の手順にて、サーバー起動前に Web リスナーの「新ステータス」を “停止” に変更してください。この手順を実施していない場合、後の手順で行う RESTful Web サービスのディプロイに失敗します。
なお、ここでは、自分で作成した Enteprise Server 名を「ESDEMO64」としています。

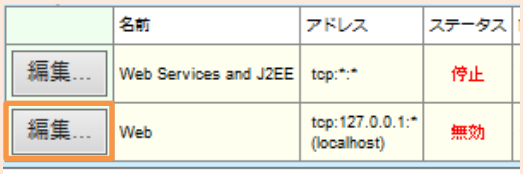
1) 「Micro Focus Server」を右クリックし、コンテクストメニューから [管理] を選択します。



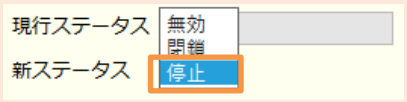
2) Enterprise Server Administration コンソール画面にて「ESDEMO2」サーバー欄の [通信プロセス] 列中の [リスナー] の横にある [詳細] ボタンをクリックします。



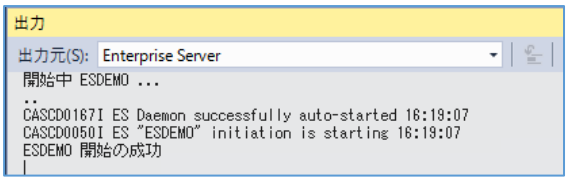
3) リスナーの構成情報が表示されるので [名前] 列中の [Web] の [編集] ボタンをクリックします。



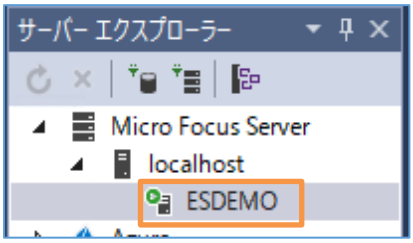
4) 新ステータスを「停止」に変更し、[OK] ボタンをクリックします



② Visual Studio の [出力] に起動メッセージが表示されます。

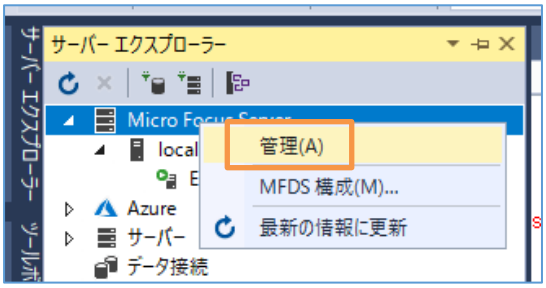


③ 正常に開始されると [サーバーエクスプローラー] 上の ESDEMO アイコンが起動されたことを示す緑色のアイコンに切り替わります。

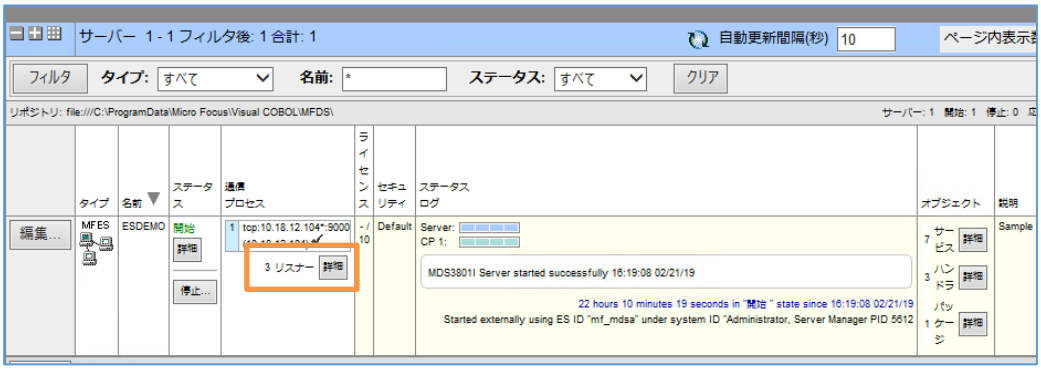


3.2. Enterprise Server の設定変更

1) ディプロイ用フォルダを作成します。
   ① 「RESTfulCOBOL」プロジェクト上で右クリックし、コンテクストメニューから［追加(D)]→[新しいフォルダー(D)］を選択します。
   ② フォルダー名に ”deploy” を指定します。
2) 「.mfdeploy」ファイルをインポートします。
   ① 作成した「deploy」フォルダー上で右クリックし、コンテクストメニューから［追加(D)]→[既存の項目(G)］を選択します。
   ② ［すべてのファイル(*.*)］に変更した上で、Visual COBOL インストールフォルダ¥deploy 配下にある［.mfdeploy］ファイルを指定します。
3) サーバーエクスプローラーを表示し、「Micro Focus Server」上で右クリックし、コンテクストメニューから［管理(A)］を選択します。



4) Enterprise Server Administration コンソール画面にて［通信プロセス］列中の［リスナー］の横にある［詳細］ボタンをクリックします。



5) リスナーの構成を変更します。

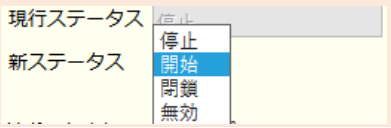
① リスナーの構成情報が表示されるので［名前］列中の［Web］の［編集］ボタンをクリックします。

② 構成情報フィールドを見るとデフォルトは「uploads=<ES>/deploy」となっています。この場合、Visual COBOL インストールディレクトリ配下の deploy フォルダがディプロイ用フォルダとして使用されます。通常、Program Files (x86) 等のフォルダは管理者権限を持つユーザーでないと書き込みできないので変更を行います。

③ 下記の例のように設定を変更し、[適用] ボタンをクリックします。

例：uploads=C:/work/Restful/RestfulCOBOL/RestfulCOBOL/deploy

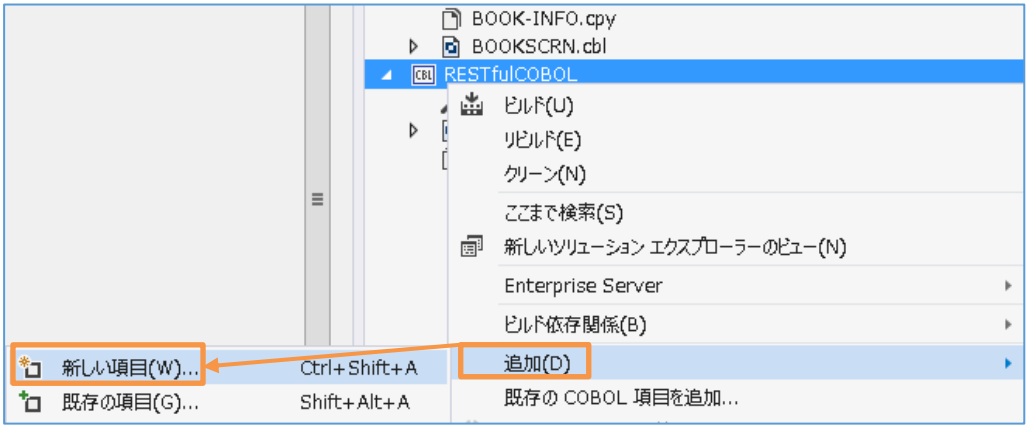
注意）
Web リスナーが開始状態で、この変更を行うと Web リスナーが「停止」状態になります。
このため、新ステータスを「開始」に変更した上で、[OK] ボタンをクリックします。



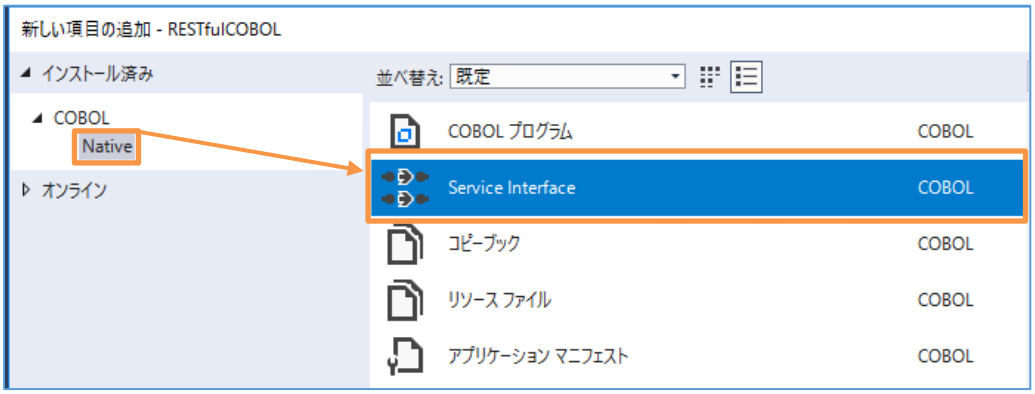
3.3. RESTful Web サービスの開発作業

1) RESTful Web サービスのプロファイル作成

① 「RESTfulCOBOL」プロジェクトを右クリックし、コンテクストメニューから［追加(D)]> [新しい項目(W)］を選択します。

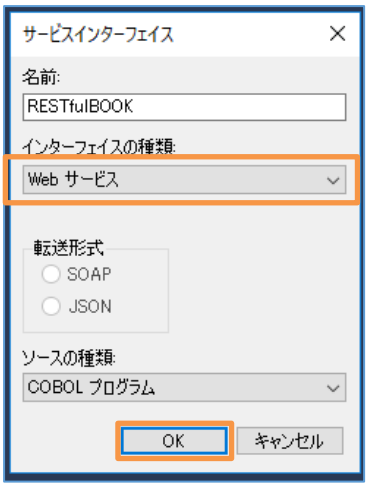


② [COBOL] > [Native] アイテムを選択し、 [Service Interface] を指定します。



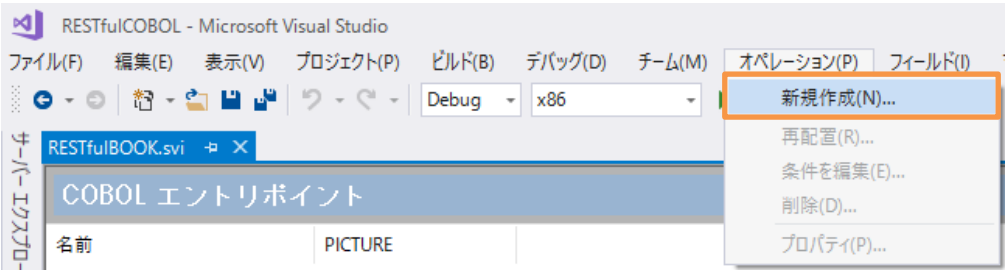
③ [名前] に ”RESTfulBOOK.svi” を指定し、[追加(A)] ボタンをクリックします。

④ [サービスインターフェイス] 画面が表示されるので、インターフェイスの種類はデフォルトの「Web サービス」を選択したまま [OK] ボタンをクリックします。

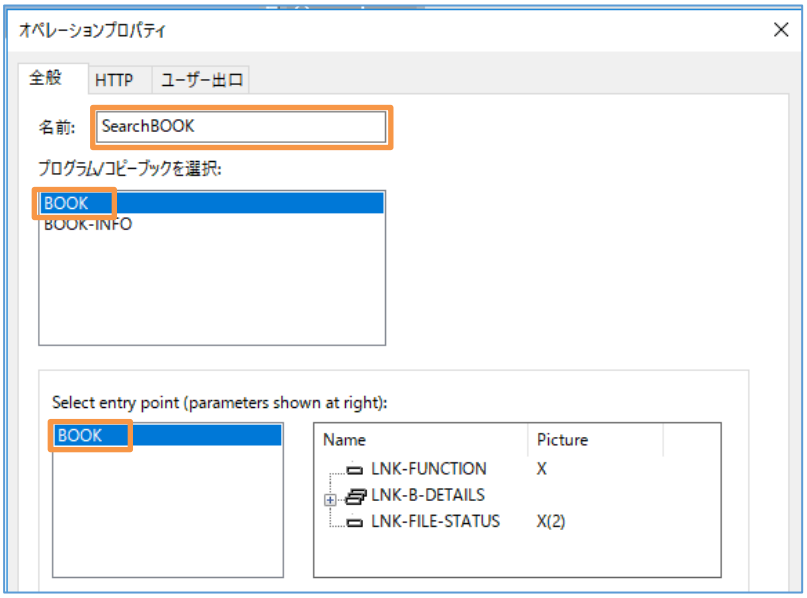


2) 書籍検索用のオペレーションを作成

① 下図のように Visual Studio 内で「RESTfulBOOK.svi」が開いている状態で [オペレーション(P)] メニュー > [新規作成(N)] を選択します。

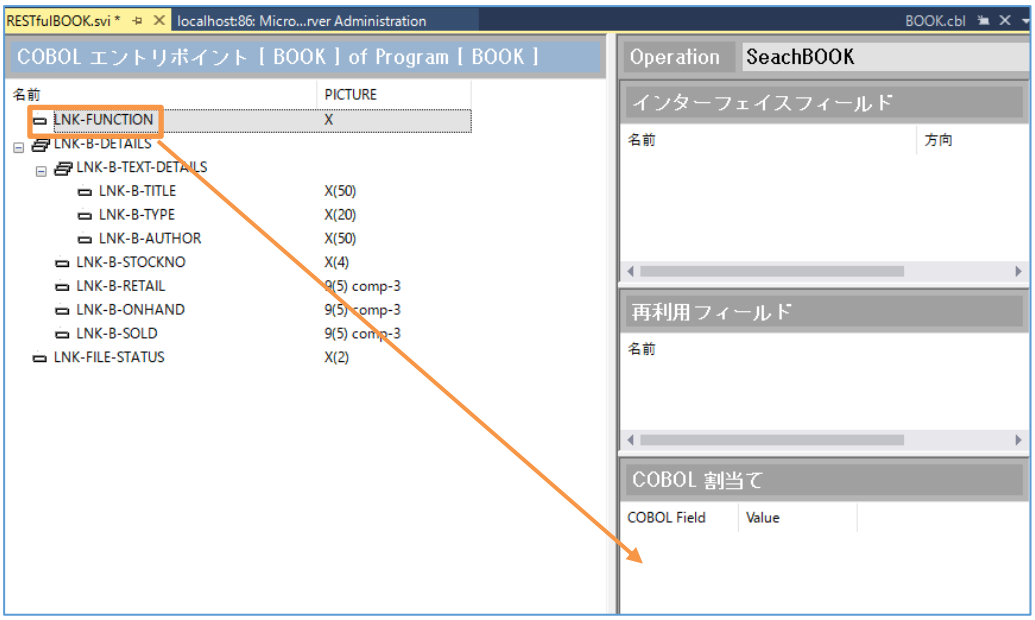


② [オペレーションプロパティ] ダイアログが表示されるので [名前] に "SeachBOOK" を入力し、[プログラム/コピーブックを選択] では、「BOOK」を選択、[Select entry point] も「BOOK」を選択し、[OK] ボタンをクリックします。

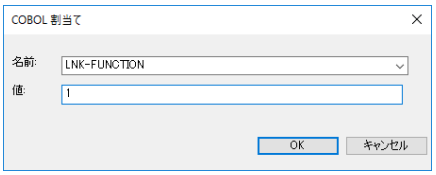


3) COBOL と RESTful Web サービス間の変数型変換マッピングを定義（書籍情報検索用オペレーション）

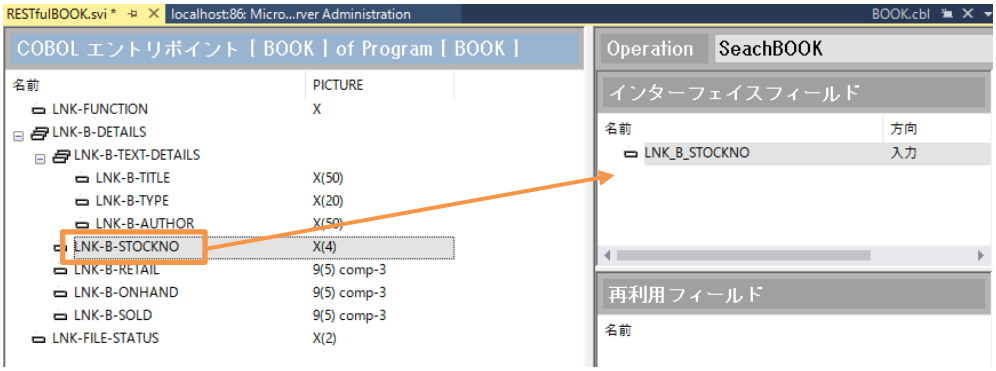
① 「LNK-FUNCTION」を［COBOL 割当て］ペインにドラッグ＆ドロップします。



② ［値］フィールドには “1” を指定します。



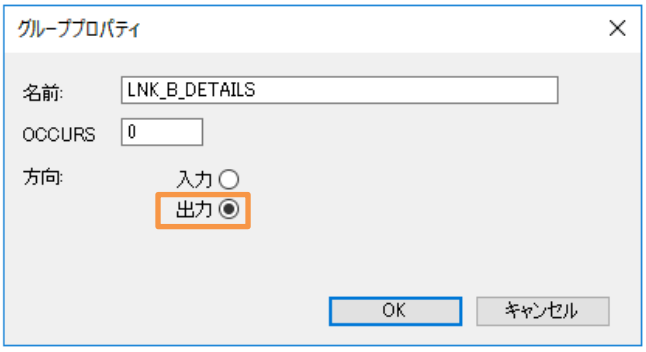
③ 次に「LNK-B-STOCKNO」を［インターフェイスフィールド］ペインにドラッグ＆ドロップします。



④ 次に「LNK-B-DETAILS」及び「LNK-FILE-STATUS」も［インターフェイスフィールド］ペインにドラッグアンドドロップします。

⑤ 次に「LNK-B-DETAILS」をダブルクリックします。

グループプロパティが表示されるので方向を「入力」から「出力」に変更します。



⑥ 同様に「LNK-FILE-STATUS」の方向も「入力」から「出力」に変更します。

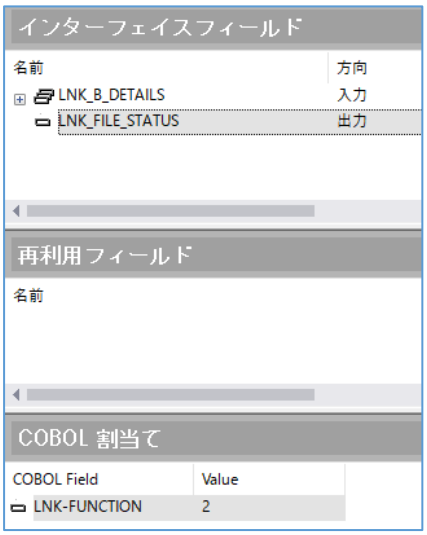
4) 書籍データ追加機能のオペレーションを追加

① 2）の手順で「AddBOOK」オペレーションを作成します。

② 次に［HTTP］タブを選択します。HTTP メソッドが［POST］になっていることを確認し、[OK］ボタンをクリックします。

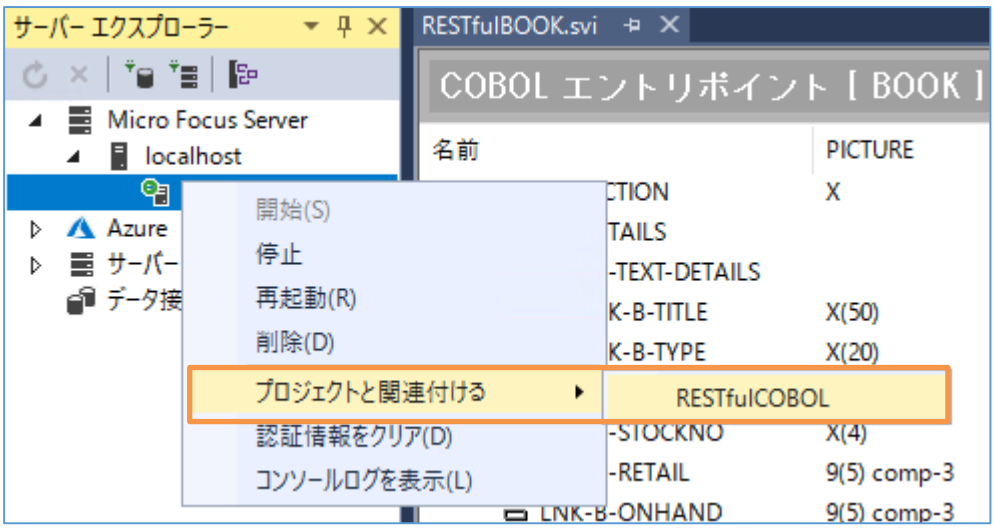
5) 書籍データ登録機能オペレーションのインターフェイスマッピングを定義

① 3）の手順で「AddBOOK」オペレーションの変換マッピングを定義します。最終的に下のイメージとなるように定義を行ってください。



6) プロジェクトと Enterprise Server「ESDEMO」を関連付ける

① サーバーエクスプローラーにて、[Micro Focus Servers] > [localhost] > [ESDEMO] を右クリックし、コンテクストメニューから［プロジェクトと関連付ける] > [RESTfulCOBOL］を選択します。

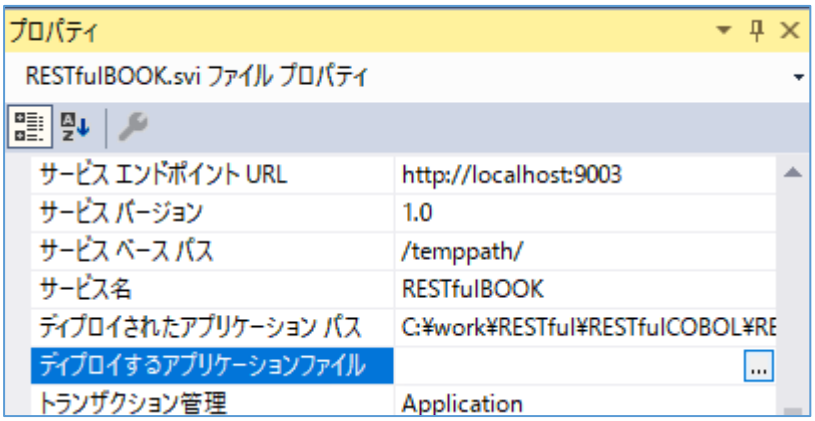


3.4. コンパイルした COBOL アプリケーションを Enterprise Server へディプロイ

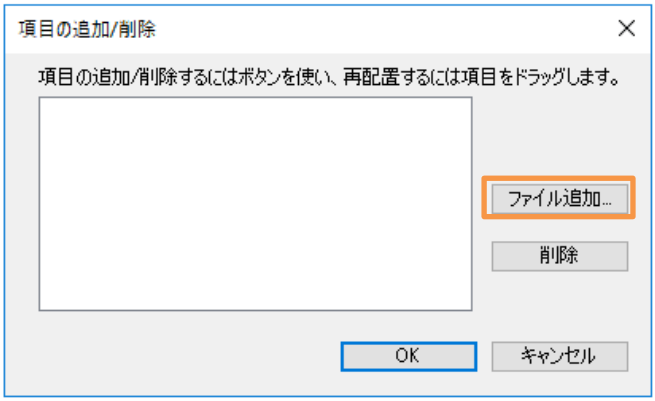
1) ディプロイする COBOL プログラムの指定

① ソリューションエクスプローラーにて「RESTfulBOOK.svi」を右クリックし、コンテクストメニューから［プロパティ］を選択します。

② プロパティペインの［ディプロイするアプリケーションファイル］右横にある［...］ボタンをクリックします。

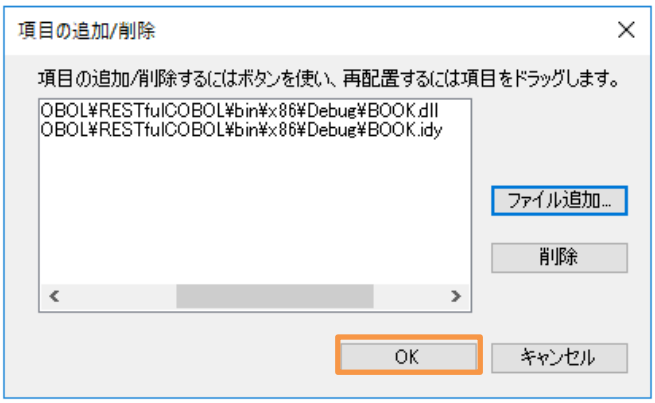


③ ［項目の追加/削除］ウィンドウが表示されるので［ファイル追加］ボタンを押します。



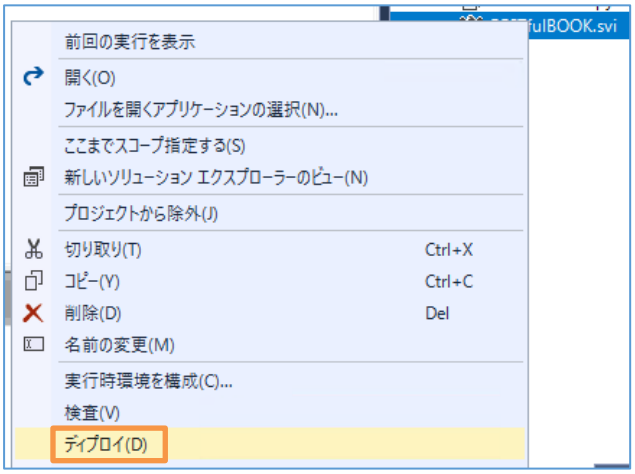
④ エクスプローラから「VisualStudio プロジェクトフォルダ¥RESTfulCOBOL¥bin¥x86¥Debug」まで移動し、「BOOK.dll」及び「BOOK.idy」を指定します。ファイルが存在しない場合は、リビルドを実行してください。リビルドは、[ビルド(B)]メニュー ＞ [ソリューションのリビルド(R)] を選択することで実行できます。

⑤ [項目の追加/削除] ウィンドウにファイルがセットされるので [OK] ボタンをクリックします。

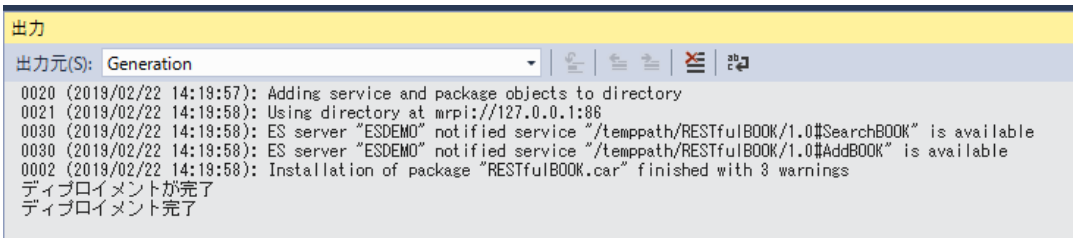


2) RESTful Web サービスのコンポーネント一式を Enterprise Server へディプロイする

① 「RESTfulBOOK.svi」を右クリックし、コンテクストメニューから [ディプロイ(D)] を選択します。

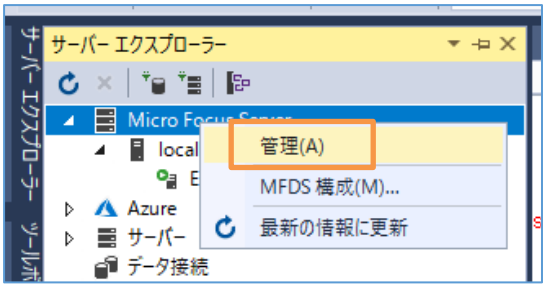


② ディプロイが完了すると下図のようなメッセージが出力されます（ここでは警告は無視して構いません）。

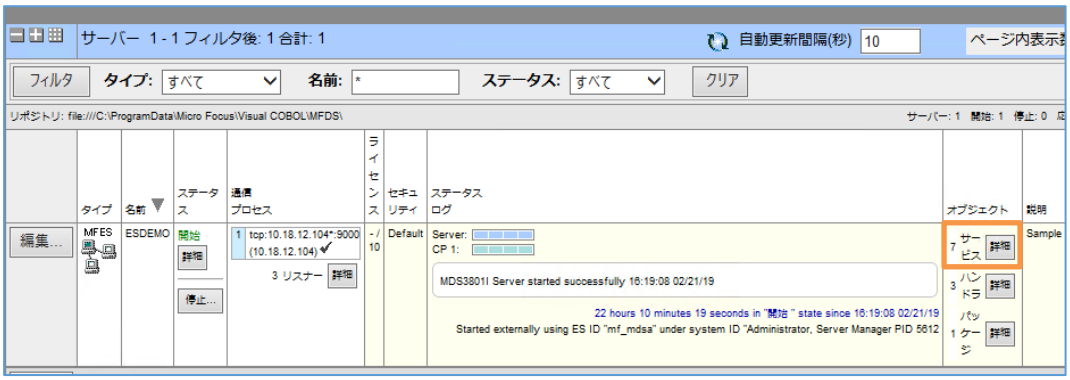


3) Enterprise Server Administration コンソール画面よりディプロイされたことを確認

① サーバーエクスプローラーを表示し、「Micro Focus Server」上で右クリックし、コンテクストメニューから [管理(A)] を選択します。

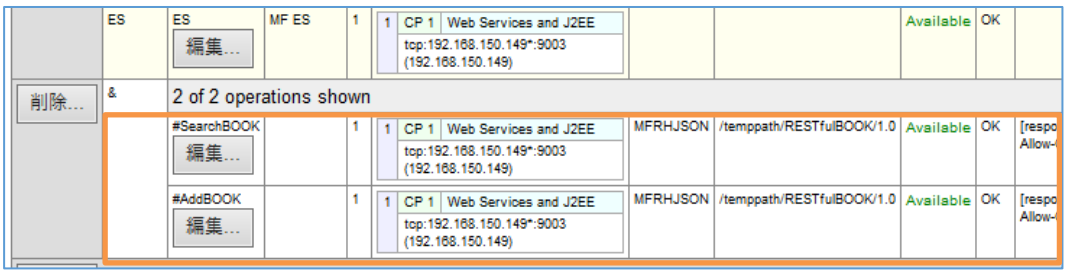


② Enterprise Server Administration コンソール画面にて［オブジェクト］列中の［サービス］の横にある［詳細］ボタンをクリックします。



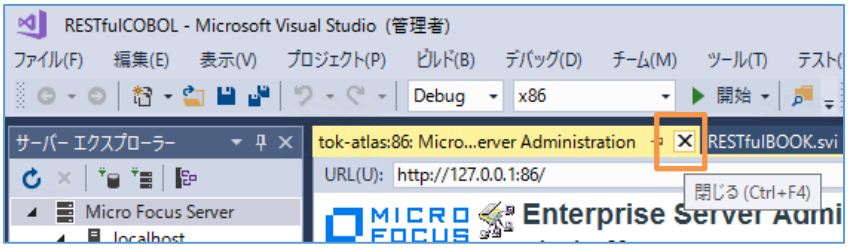
補足）
サービス数や下記のサービス一覧で表示される内容は、Enterprise Server に登録されたサービス内容により異なります。

③ 最下行にディプロイした RESTful Web サービスが追加されていることを確認します。



補足）
登録サービス数が多い場合は、画面右上の「ページ内表示数」の変更や、[<< 前へ]、[次へ >>] などで表示位置を最後に変更してください。

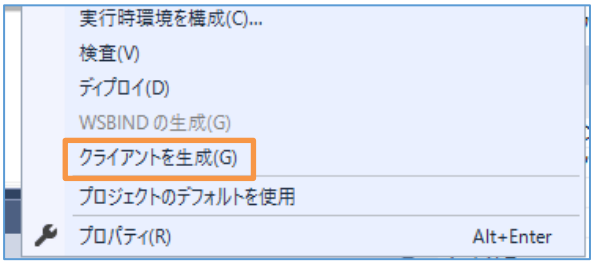
④ 追加されていることを確認したら、Enterprise Server Administration コンソール画面のペインを×マークをクリックして閉じます。



3.5. RESTful Web サービスのテスト

1) RESTful Web サービステスト用のクライアント生成

① 「RESTfulBOOK.svi」を右クリックし、コンテクストメニューから［クライアントの生成(G)］を選択します。

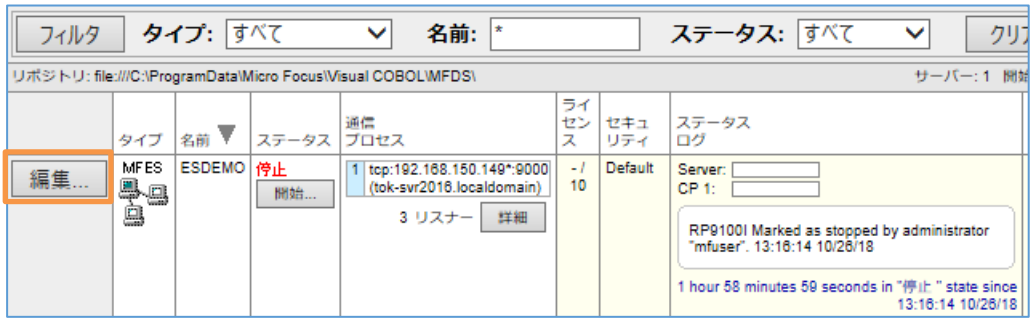


2) Enterprise Server がアクセスするデータファイルを指定

① 「サーバーエクスプローラー」に切り替えます。

② ［停止］ボタンをクリックし、Enterprise Server を停止します。

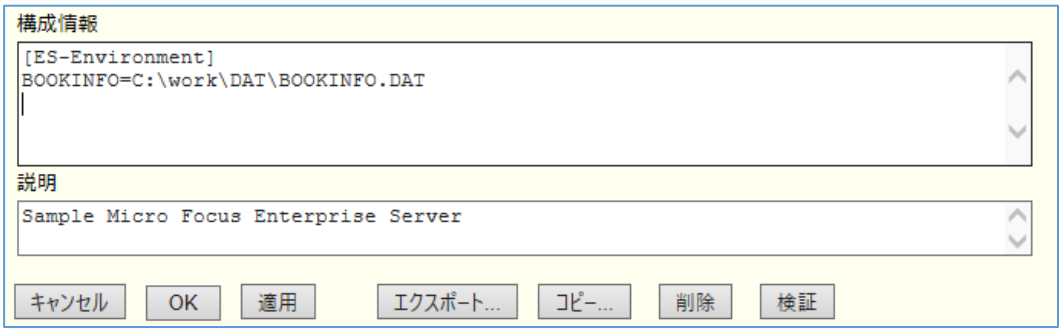
③ ［編集］ボタンを押して構成を変更します。



④ ［サーバー］> ［プロパティ］> ［一般］タブの 構成情報］にデータファイルの場所を示す以下の値を設定します。ディレクトリ名はデータファイルを解凍した任意のフォルダとなります。

［ES-Environment］

BOOKINFO=C:¥work¥DAT¥BOOKINFO.DAT



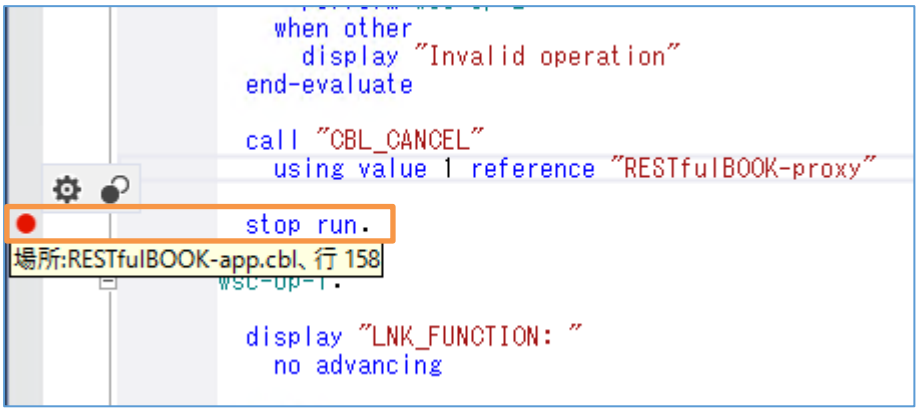
⑤ ［OK］ボタンをクリックします。

⑥ サーバーエクスプローラーを表示し、［Micro Focus Server］> ［localhost］> ［ESDEMO］を選択し、右クリックにてコンテクストメニューから ［開始(S)］をクリックします。

3) テスト用アプリケーションの実行準備

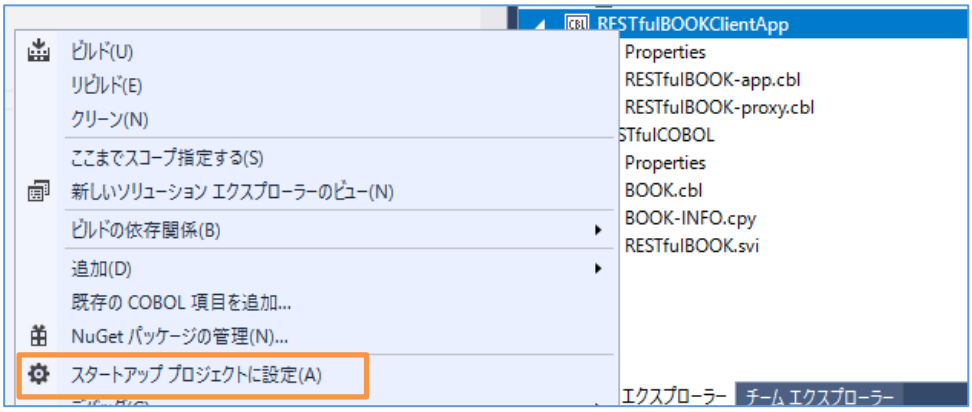
① RESTfulBOOK-app.cbl をダブルクリックして開きます。

② テスト用クライアントは実行を終了すると DOS 画面から消えてしまうため、158 行目にブレークポイントを設定します。



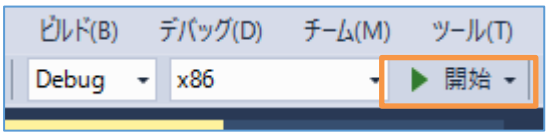
③ [ビルド(B)]メニュー ＞[ソリューションのリビルド(R)] を選択します。

④ 「RESTfulBOOKClientApp」プロジェクトを右クリックし コンテクストメニューから［スタートアッププロジェクトに設定(A)］を選択します。



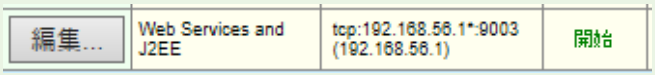
4) 生成したテスト用 COBOL クライアントの実行（登録処理）

① ツールバーにて［開始］アイコンをクリックし、アプリケーションを起動します。DOS プロンプトでアプリケーションが起動します。



② 「Service Address」は、デフォルトのままで Enter キーを押します。

補足）
Web サービスの要求を受け付けるリスナーが以下のようにポート 9003 でリッスンしているため、変更せず、クライアントの実行が行えます。このポートは、[編集] ボタンをクリックした後の画面にて任意に変更できます。変更した場合は、クライアント実行時に変更後のポートを指定する必要があります。



③ 「Username」、「Password」は変更しないのでそのまま 2 回 Enter を押します。

④ 「Operation (1 = SearchBOOK, 2 = AddBOOK」は、"2" を入力して Enter を押します。

⑤ 「LNK_B_TITLE」は "PLANET OF THE APES" を入力して Enter を押します。

⑥ 「LNK_B_TYPE」は "SCIENCE FICTION" を入力して Enter を押します。

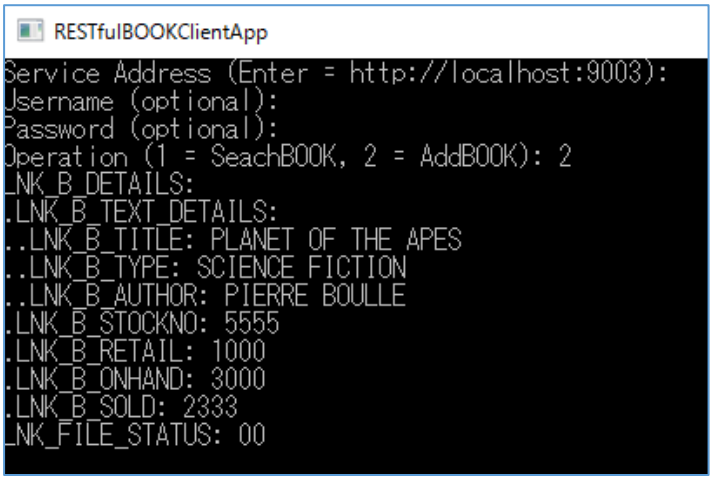
⑦ 「LNK_B_AUTHOR」は "PIERRE BOULLE" を入力して Enter を押します。

⑧ 「LNK_B_STOCKNO」は、"5555" を入力して Enter を押します。

⑨ 「LNK_B_RETAIL」は、"1000"、「LNK_B_ONHAND」は、"3000"、「LNK_B_SOLD」は、"2333"を入力して Enter を押します。

⑩ RESTful の Web サービスが実行されます。「lnk_FILE_STATUS」に "00" が返ってきたら登録成功です。

⑪ ブレークポイントで止まっているのでツールバーから［続行］をクリックして処理を終了させます。



5) 生成したテスト用 COBOL クライアントの実行（検索処理）

① ツールバーにて［開始］アイコンをクリックし、アプリケーションを起動します。DOS プロンプトでアプリケーションが起動します。

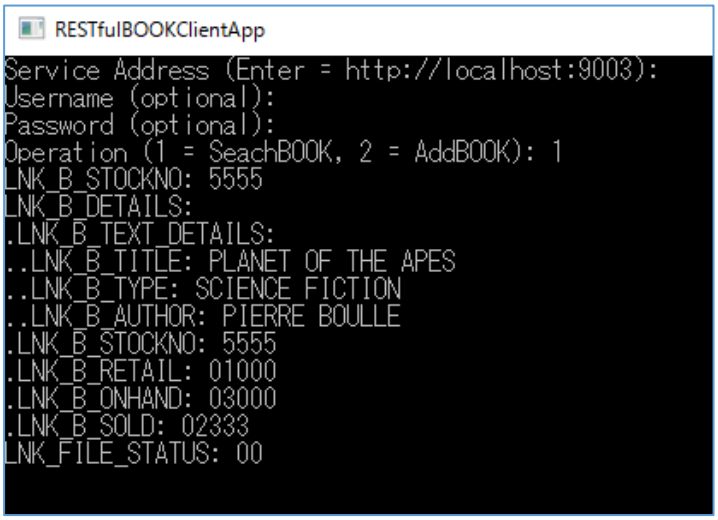
② 「Service Address」は、デフォルトのままで Enter キーを押します。

③ 「Username」、「Password」は変更しないのでそのまま 2 回 Enter を押します。

④ 「Operation (1 = SEARCHBOOK, 2 = ADDBOOK」は、"1" を入力して Enter を押します。

⑤ 「lnk_FUNCTION」は、 "5555" を入力して Enter を押します。

⑥ RESTful の Web サービスが実行され、データが返ってきます。「登録処理」で入力した内容が確認できます。

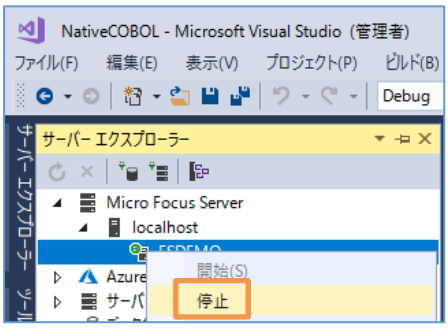


⑦ ブレークポイントで止まっているのでツールバーから［続行］をクリックして処理を終了させます。

3.6. インスタンスの停止

1) Enterprise Server の停止

① サーバーエクスプローラーにて［Micro Focus Server］>［localhost］>［ESDEMO］を右クリックし、コンテクストメニューから［停止］を選択します



WHAT'S NEXT

- 本チュートリアルで学習した技術の詳細については製品マニュアルをご参照ください。

免責事項

ここで紹介したソースコードは、機能説明のためのサンプルであり、製品の一部ではございません。ソースコードが実際に動作するか、御社業務に適合するかなどに関しまして、一切の保証はございません。 ソースコード、説明、その他すべてについて、無謬性は保障されません。
ここで紹介するソースコードの一部、もしくは全部について、弊社に断りなく、御社の内部に組み込み、そのままご利用頂いても構いません。
本ソースコードの一部もしくは全部を二次的著作物に対して引用する場合、著作権法の精神に基づき、適切な扱いを行ってください。