
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 11 Professional Edition (64bit)
COBOL 開発環境製品	Micro Focus Visual COBOL 8.0 for Visual Studio 2022
IDE	Microsoft Visual Studio Professional 2022 (17.2.2)

- 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

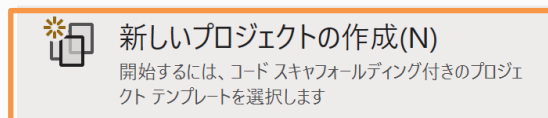
3.1. Windows クライアントでの開発準備作業

1) Visual COBOL for Visual Studio を起動

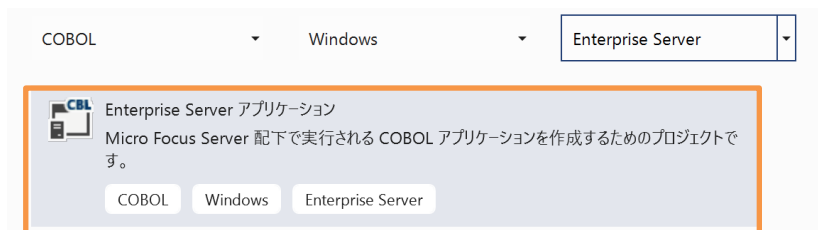
- ① [スタート] メニュー > [すべてのアプリ] > [Visual Studio 2022] を選択します。

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

- ① 「作業の開始」より「新しいプロジェクトの作成」を選択します。



- ② 「新しいプロジェクトの作成」ウィザードが表示されるので [言語] を「COBOL」、[プラットフォーム] を「Windows」、[プロジェクト タイプ] に「Enterprise Server」を選択します。

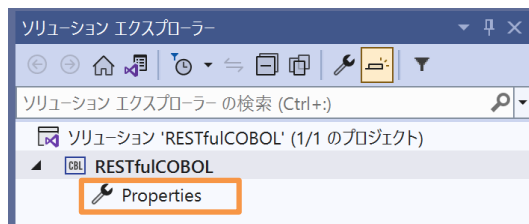


- ③ [名前(N)] フィールドに “RESTfulCOBOL” を入力して、[OK] ボタンをクリックします。[場所(L)] は任意のフォルダを指定し、[ソリューション名] は、デフォルトのままとし、[作成(C)] ボタンをクリックします。ます。

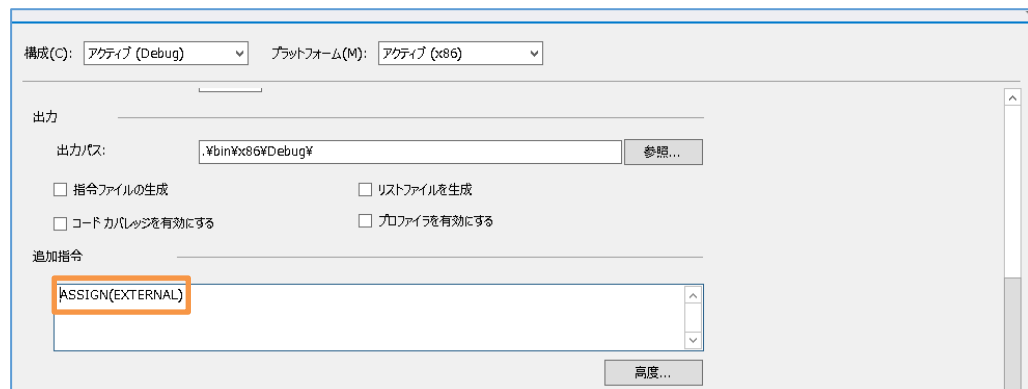


3) コンパイラオプションの指定とソースコードのインポート

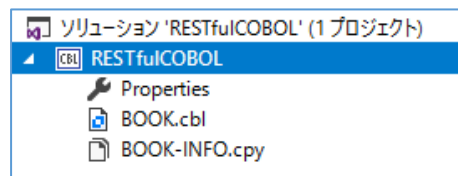
- ① 作成されたプロジェクトの Properties をダブルクリックします。



- 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。

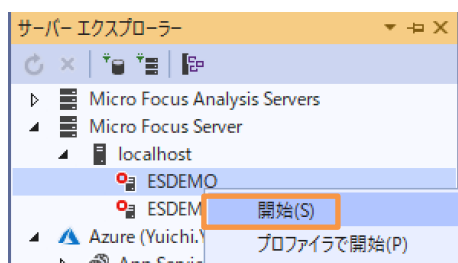


- ③ エクスプローラを起動し、サンプルのソースコードを解凍したフォルダから “BOOK-INFO.cpy” と “BOOK.cbl” をプロジェクトフォルダにドラッグアンドドロップします。
- ④ ソリューションエクスプローラーから、2つのファイルが正常にロードされていることを確認します。

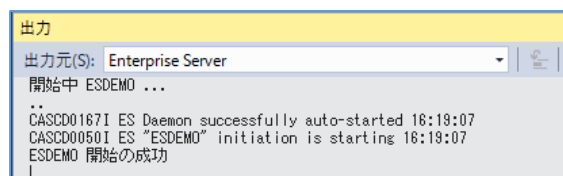


4) Enterprise Server の起動

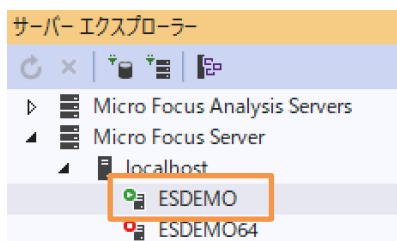
- ① [サーバーエクスプローラー] ペインを選択、[Micro Focus Server] > [localhost] > [ESDEMO] と展開します。もし表示されていないければ、[表示]メニューから[サーバーエクスプローラー]を選択すれば表示されます。「ESDEMO」の上で右クリックし、コンテキストメニューから[開始(S)]を選択します。



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

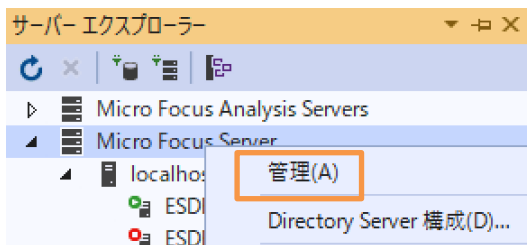


- ③ 正常に開始されると [サーバーエクスプローラー] 上の ESDEMO アイコンが起動されたことを示す緑色のアイコンに切り替わります。

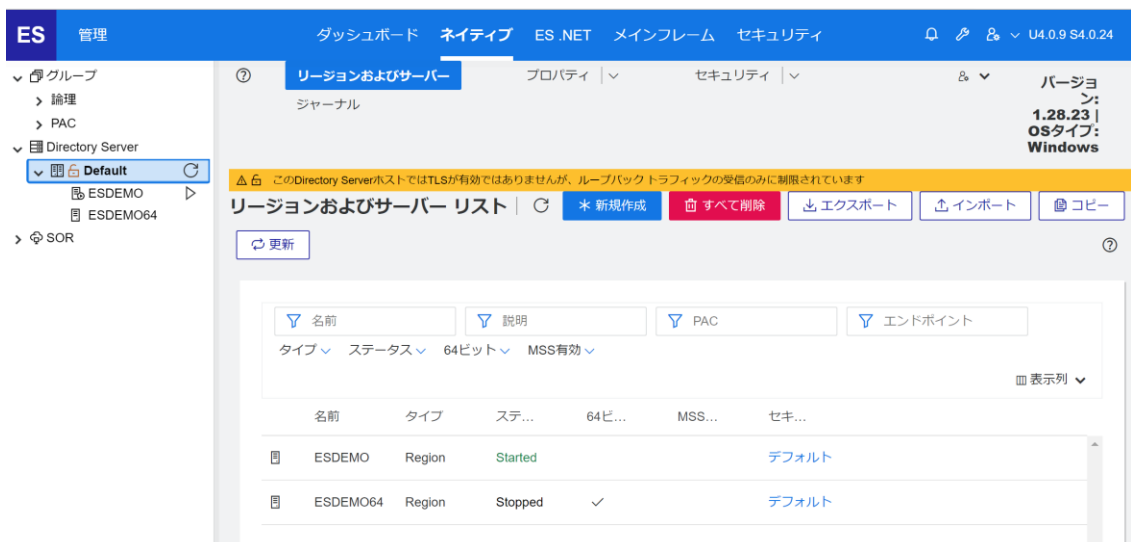


3.2. Enterprise Server の設定変更

- 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、コンテキストメニューから [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、コンテキストメニューから [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、コンテキストメニューから [管理(A)] を選択します。



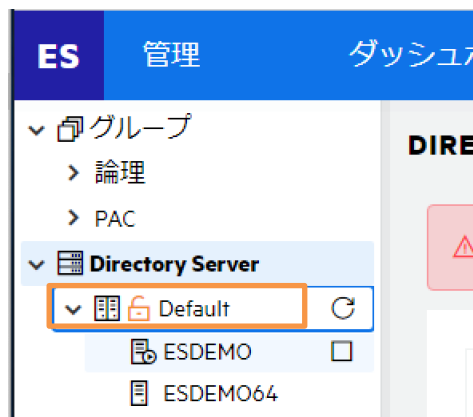
- ① ESCWA (Enterprise Server Common Web Administration) 画面にてダッシュボードが表示されます。



- 4) リスナーの構成を変更します。
 - ① [ネイティブ]タブメニューをクリックします。



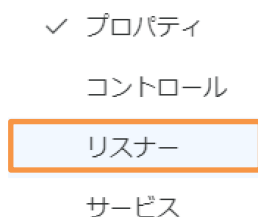
- ② 次に左側メニューの[Directory Server]をドリルダウンして、[Default] → [ESDEMO]をクリックします。



- ③ [一般]メニューが表示されるので横にある下向き記号をクリックします。



- ④ ドロップダウンメニューから[リスナー] をクリックします。



- ⑤ [通信プロセス 1]の下に見えている[Web]リスナーをクリックします。

リスナープロパティが表示されます。横のスライダーを下に下げていくとカスタム構成情報が表示されています。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。

下記の例のように設定を変更し、スライダーを上を移動して、[適用] ボタンをクリックします

例：uploads= C:\work\RESTful\RESTfulCOBOL\deploy

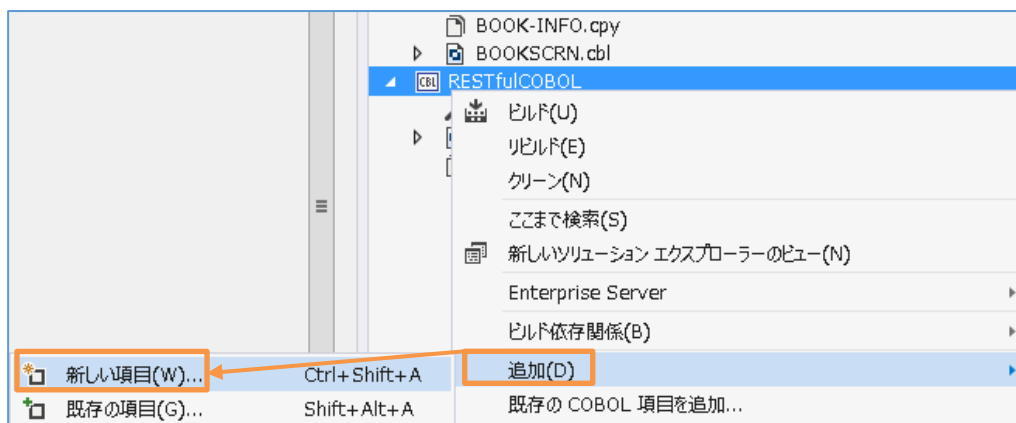
- ⑥ 正常に更新された旨のポップアップが表示されます。

※Enterprise Server を自分で作成した場合、[Web リスナー] を「Disable」から「Started」に変更してください。

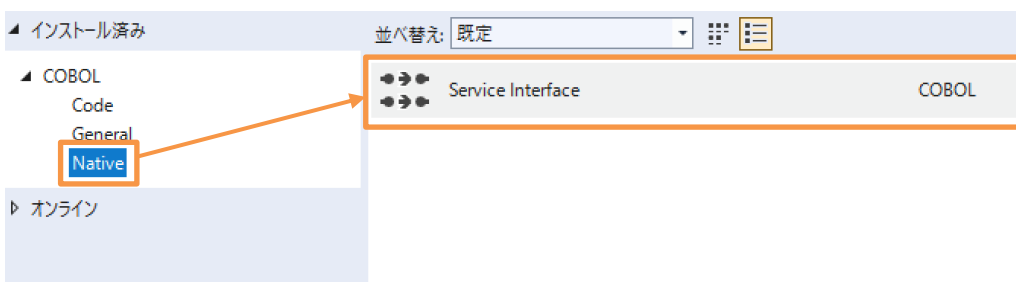
3.3. RESTful Web サービスの開発作業

- 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、コンテキストメニューから [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。

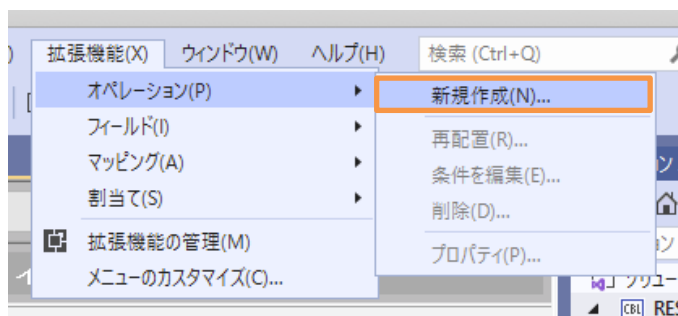


- ③ [名前] に "RESTfulBOOK.svi" を指定し、[追加(A)] ボタンをクリックします。
- ④ [サービスインターフェイス] 画面が表示されるので、インターフェイスの種類はデフォルトの「Web サービス」を選択し、[転送形式]は、「JSON (RESTful)」を選択し [OK] ボタンをクリックします。

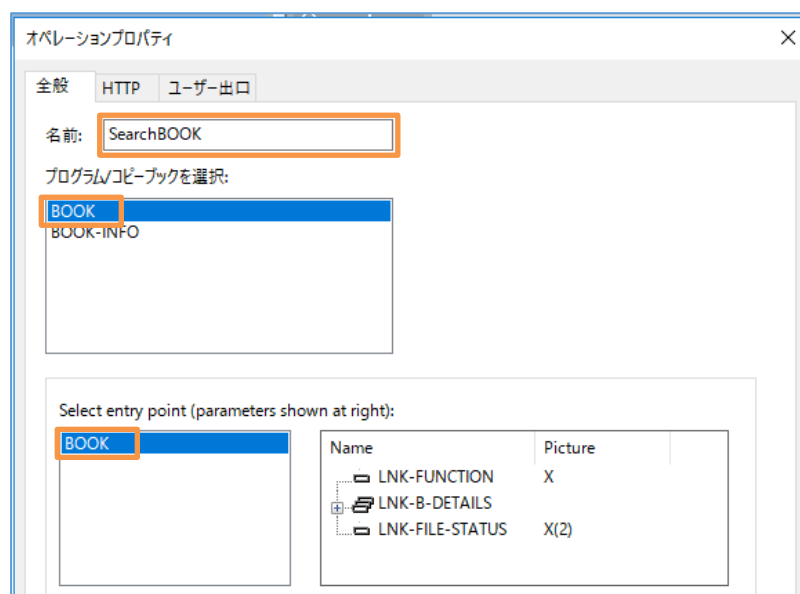


2) 書籍検索用のオペレーションを作成

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

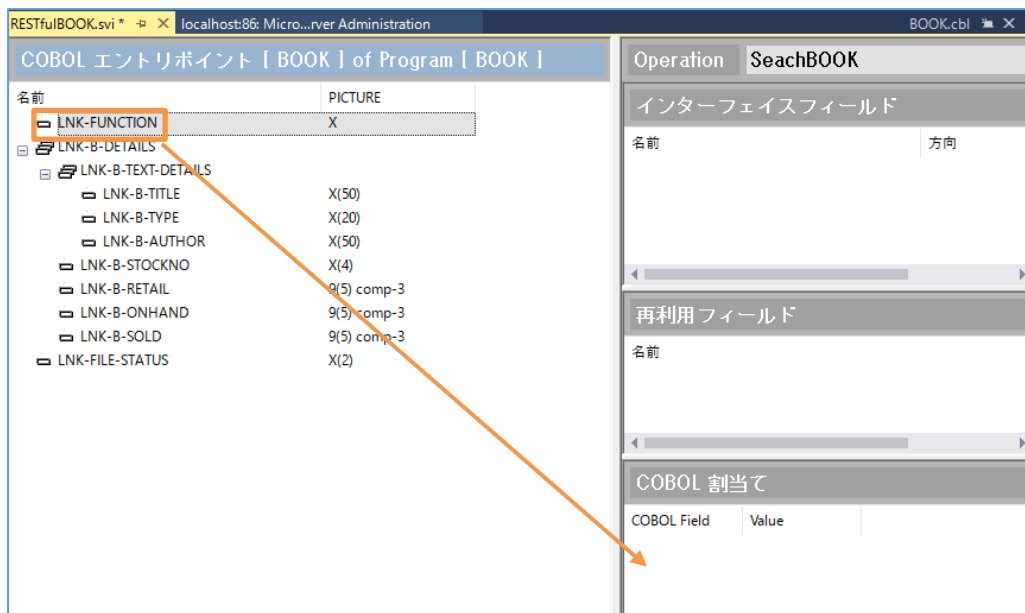


- ② [オペレーションプロパティ] ダイアログが表示されるので [名前] に "SeachBOOK" を入力し、[プログラム/コピーブック] では、「BOOK」を選択、[Select entry point] も「BOOK」を選択し、[OK] ボタンをクリックします。

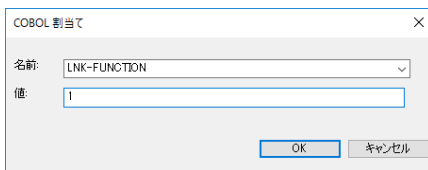


3) COBOL と RESTful Web サービス間の変数型変換マッピングを定義（書籍情報検索用オペレーション）

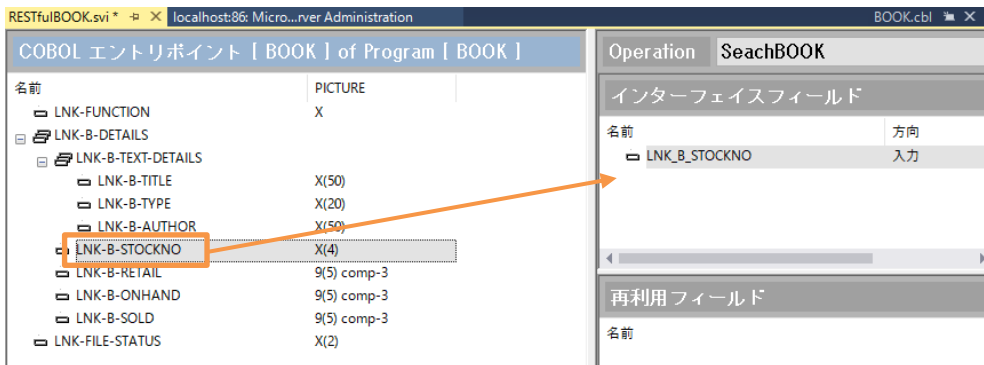
- ① 「LNK-FUNCTION」を [COBOL 割当て] ペインにドラッグ&ドロップします。



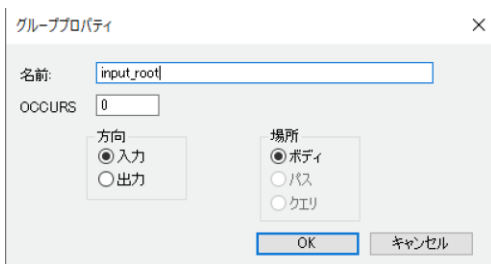
- ② [値] フィールドには “1” を指定します。



- 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。



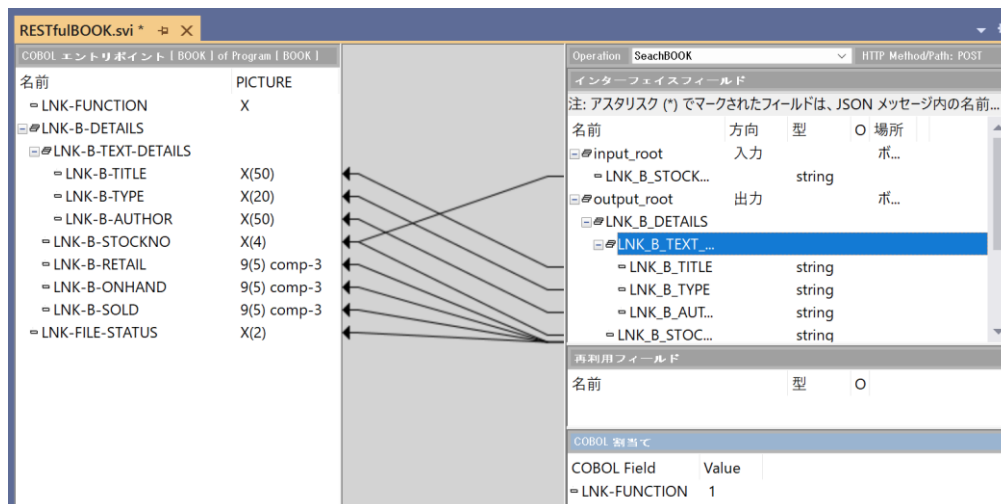
- ④ インターフェイスフィールドにある「LNK-B-STOCKNO」上で右クリックから[グループ化]を選択し、[グループプロパティ] ウィンドウにて “input_root” と入力、[方向] を [入力] にして [OK] ボタンをクリックします。



- ⑤ 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

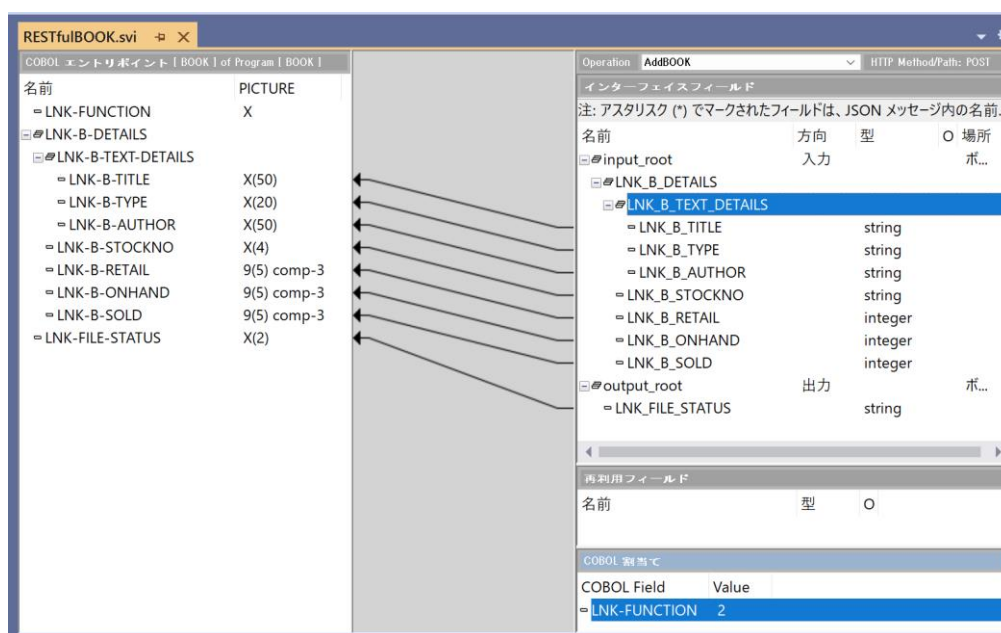
ロップします。

- ⑥ インターフェイスフィールドにある「LNK-B-DETAILS」及び「LNK-FILE-STATUS」を選択した状態で右クリックから [グループ化] を選択し、[グループプロパティ] ウィンドウにて "output_root" と入力、[方向] を [出力] にして [OK] ボタンをクリックします。
- ⑦ 下記の図のようになっていることを確認します。



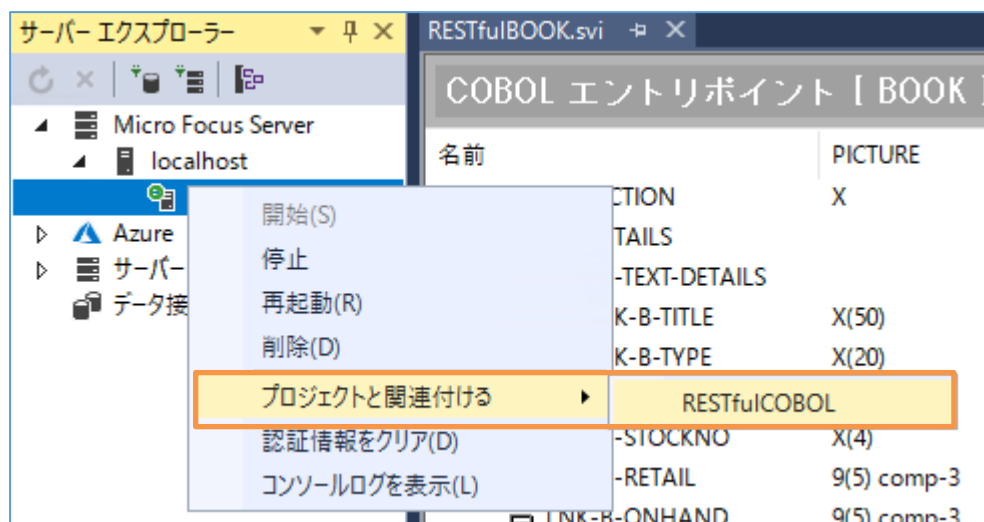
4) 書籍データ追加機能のオペレーションを追加

- 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 サーバークリエイターにて、[Micro Focus Servers] > [localhost] > [ESDEMO] を右クリックし、コンテキストメニューから [プロジェクトと関連付ける] > [RESTfulCOBOL] を選択します。



3.4. Enterprise Server ソリューションのビルド作業

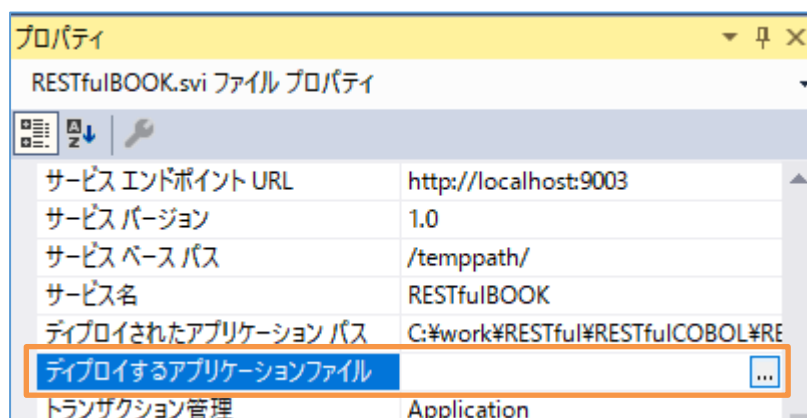
1) RESTful プロジェクトのソースコードをコンパイル

- ① [ビルド(B)]メニューから [ソリューションのビルド(B)]を選択します。

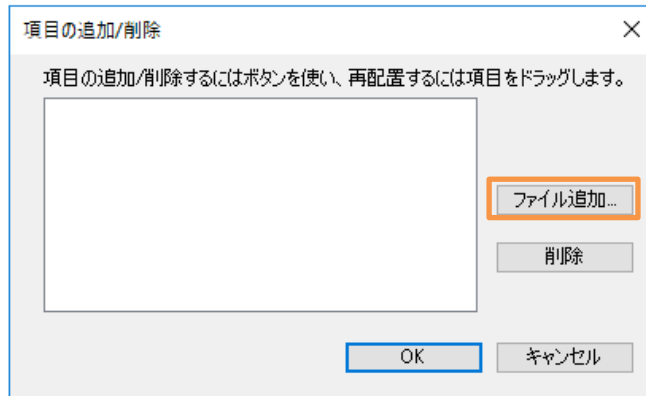
3.5. コンパイルした COBOL アプリケーションを Enterprise Server ヘッドアップ

1) デプロイする COBOL プログラムの指定

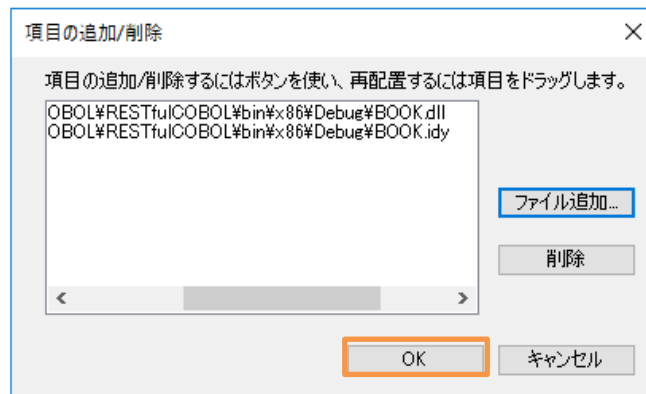
- ① ソリューションエクスプローラーにて「RESTfulBOOK.svi」を右クリックし、コンテキストメニューから [プロパティ] を選択します。
- ② プロパティペインの [デプロイするアプリケーションファイル] 右横にある [...] ボタンをクリックします。



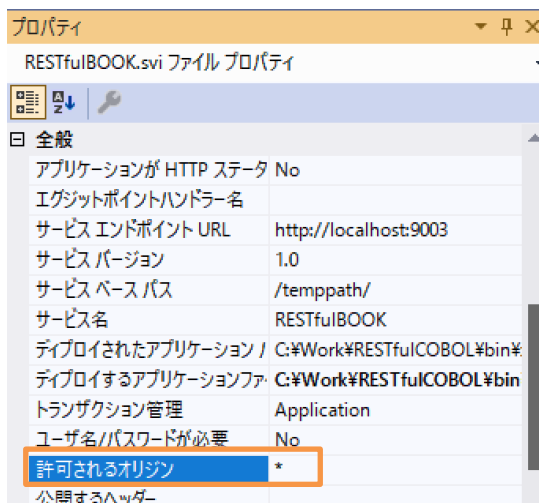
- ③ [項目の追加/削除] ウィンドウが表示されるので [ファイル追加] ボタンを押します。



- 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。

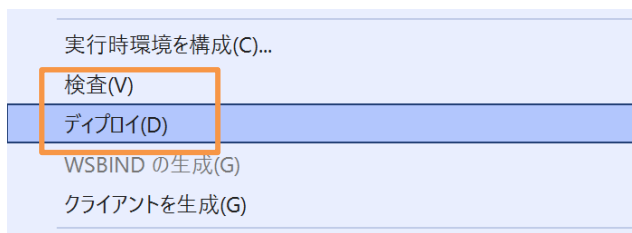


- ⑥ デフォルトではオリジン間リソース共有は許可されていません。もしこれに関するエラーが発生する場合、許可設定を行います。ここでは[許可されるオリジン]に "*" を入力します。

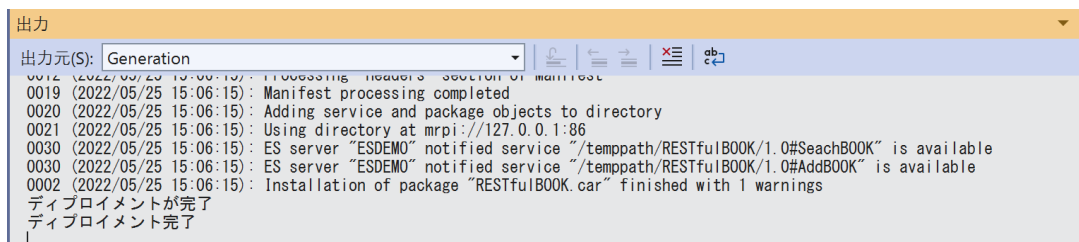


- 2) RESTful Web サービスのコンポーネント一式を Enterprise Server ヘッドアップロードする
- ① 「RESTfulBOOK.svi」を右クリックし、コンテキストメニューから [検査(V)]を実行し、問題ないことを確認したら[デバッグ]

ロイ(D)] を選択します。

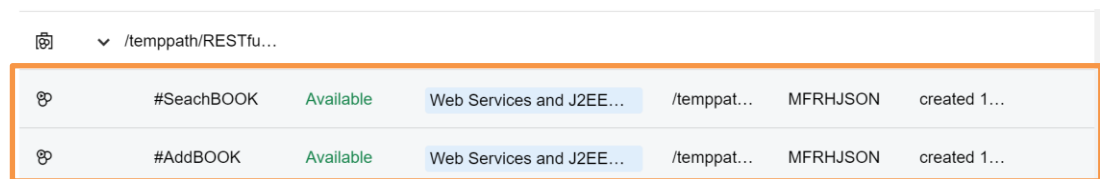


- ② デプロイが完了すると下図のようなメッセージが出力されます（ここでは警告は無視して構いません）。



- 3) ESCWA(Enterprise Server Common Web Administration) からデプロイされたことを確認

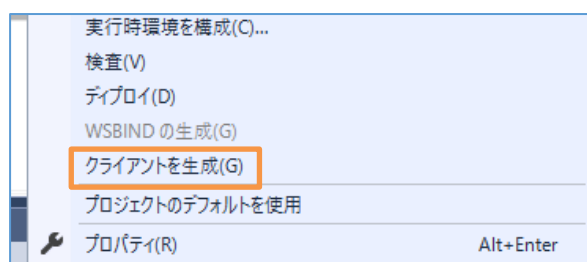
- ① ブラウザ上の ESCWA に切り替えます。
- ② [一般メニュー] から [サービス] をクリックします。
- ③ 画面下にスクロールしていくと最下行にデプロイした RESTful Web サービスが追加されていることを確認します。



3.6. 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、コンテキストメニューから [クライアントの生成(G)] を選択します。

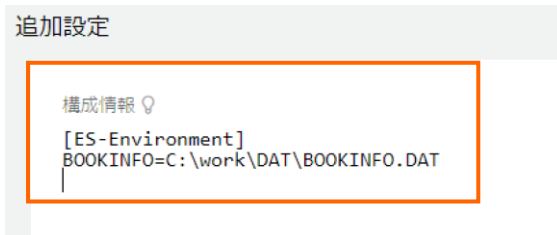


- 2) Enterprise Server がアクセスするデータファイルを指定

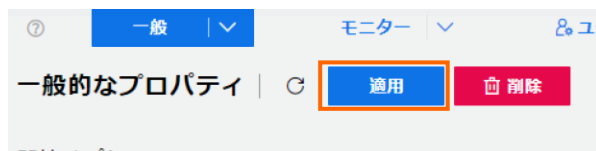
- ① ブラウザ上の ESCWA 画面に切り替えます。
- ② [一般]をクリックし、[追加設定]フィールドに下記の命令を追加します。

[ES-Environment]

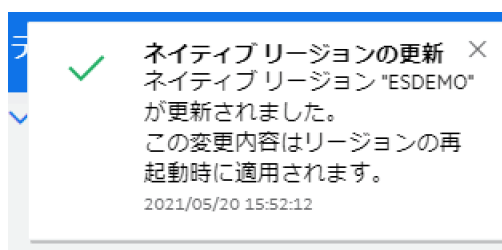
BOOKINFO=C:\¥work¥DAT¥BOOKINFO.DAT



- ③ 上にスクロールし、[適用]ボタンをクリックします。



- ④ 更新された旨のポップアップメッセージが表示されます。

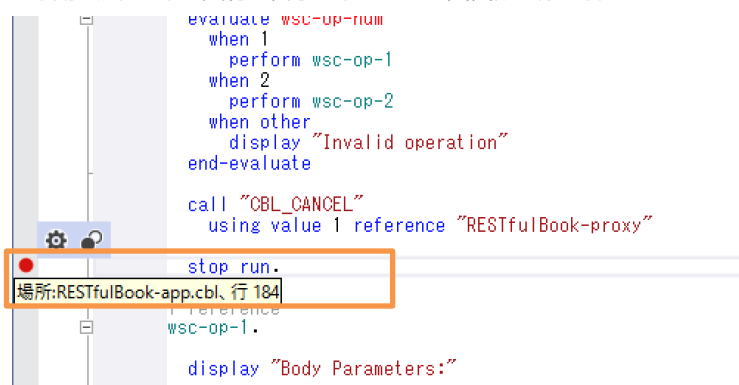


1) Enterpriser Server の再起動

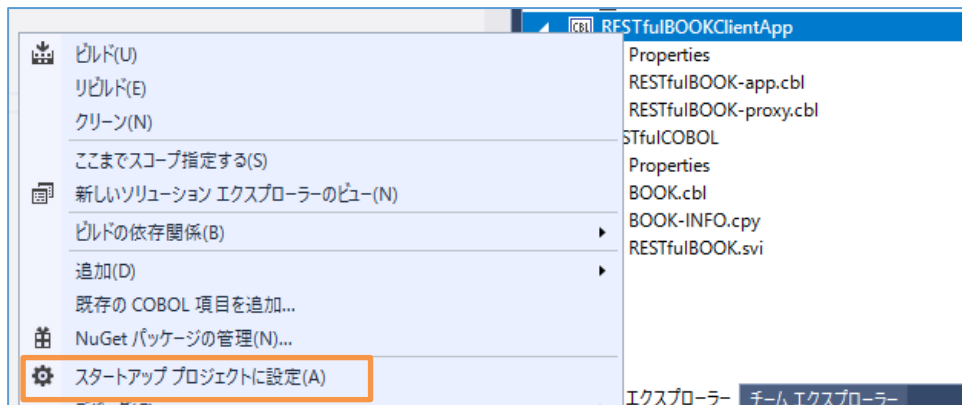
- ① Visual Studio に戻り、「サーバーエクスプローラー」に切り替えます。
- ② 「ESDEMO」上で右クリックし、コンテキストメニューから[再起動]を選択し、Enterprise Server を停止/起動します。

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、184 行目にブレークポイントを設定します。

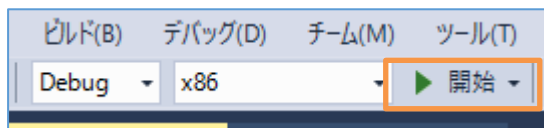


- ③ [ビルド(B)]メニュー > [ソリューションのリビルド(R)] を選択します。
- ④ 「RESTfulBOOKClientApp」プロジェクトを右クリックし、コンテキストメニューから[スタートアッププロジェクトに設定(A)]を選択します。



4) 生成したテスト用 COBOL クライアントの実行（登録処理）

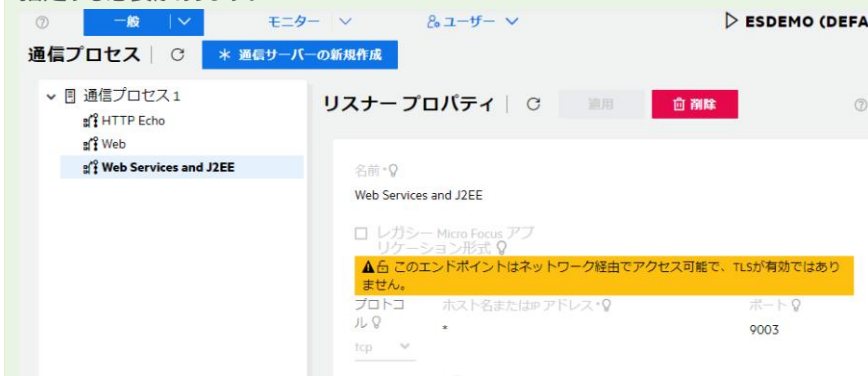
- ① ツールバーにて「開始」アイコンをクリックし、アプリケーションを起動します。DOS プロンプトでアプリケーションが起動します。



- 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、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 「Supplemental Query String (optional)」、 「Username」、 「Password」 は変更しないのでそのまま 3 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。


```

RESTfulBOOKClientApp
Service Address (Enter = http://localhost:9003):
Supplemental Query String (optional):
Username (optional):
Password (optional):
Operation (1 = SeachBOOK, 2 = AddBOOK): 2
Body Parameters:
.
. LNK_B_DETAILS:
. LNK_B_TEXT_DETAILS:
. . LNK_B_TITLE: PLANET OF THE APES
. . LNK_B_TYPE: SCIENCE FICTION
. . LNK_B_AUTHOR: PIERRE BOULLE
. LNK_B_STOCKNO: 5555
. LNK_B_RETAIL: 1000
. LNK_B_ONHAND: 3000
. LNK_B_SOLD: 2333
.
. LNK_FILE_STATUS: 00

```

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 「Supplementa Query String (optional)」、「Username」、「Password」は変更しないのでそのまま 3 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。

```

RESTfulBOOKClientApp
Service Address (Enter = http://localhost:9003):
Supplemental Query String (optional):
Username (optional):
Password (optional):
Operation (1 = SeachBOOK, 2 = AddBOOK): 1
Body Parameters:
.
. LNK_B_STOCKNO: 5555
.
. LNK_B_DETAILS:
. LNK_B_TEXT_DETAILS:
. . LNK_B_TITLE: PLANET OF THE APES
. . LNK_B_TYPE: SCIENCE FICTION
. . LNK_B_AUTHOR: PIERRE BOULLE
. LNK_B_STOCKNO: 5555
. LNK_B_RETAIL: 01000
. LNK_B_ONHAND: 03000
. LNK_B_SOLD: 02333
. LNK_FILE_STATUS: 00

```

- 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。

3.7. 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切り替えます。

- ② [ESDEMO] を右クリックし、コンテキストメニューから [停止] を選択し、Enterprise Server を停止します。

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。