

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 Eclipse 編

1. 目的

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 RESTful 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 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。サポートしているプラットフォームであれば Linux/UNIX でも利用可能です。

- 開発クライアント ソフトウェア

OS	Windows Server 2022 Standard Edition (64bit)
----	--

COBOL 開発環境製品	Visual COBOL 9.0 for Eclipse
--------------	------------------------------

- 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

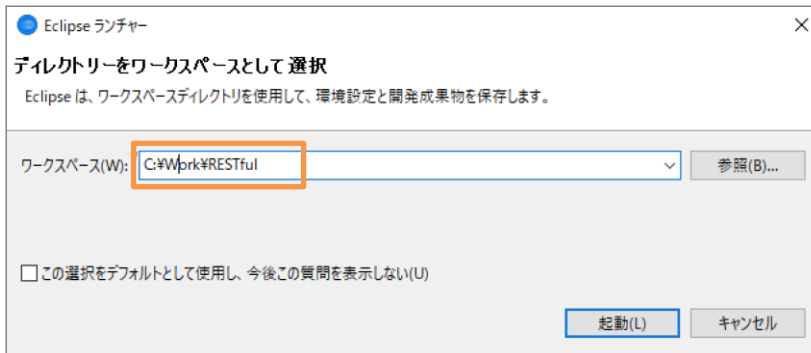
1. 目的
2. 前提条件
3. チュートリアル手順の概要
 - 3.1. Windows クライアントでの開発準備作業
 - 3.2. Enterprise Server に登録するサービスのディプロイ先変更
 - 3.3. RESTful Web サービスの開発作業
 - 3.4. コンパイルした COBOL アプリケーションを Enterprise Server へディプロイ
 - 3.5. RESTful Web サービスのテスト
 - 3.6. RESTful Web サービスのデバッグ
 - 3.7. インスタンスの停止

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 Eclipse を起動

- ① [スタート] メニュー > [すべてのアプリ] > [Micro Focus Visual COBOL] > [Visual COBOL for Eclipse] を選択します。
- ② ワークスペースの選択画面にて “C:¥Wprk¥RESTful” を指定し、[起動] ボタンをクリックします。

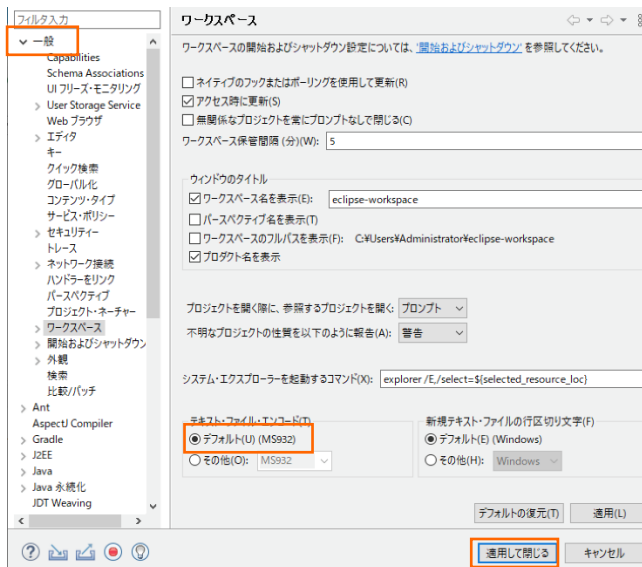


2) ネイティブ COBOL プロジェクトの作成

- ① [ファイル]メニュー > [新規] > [COBOL プロジェクト] を選択し、プロジェクト名に “NativeCOBOL” を指定して、[終了(F)] ボタンをクリックします。※テンプレートは 32 ビットで構いません。

3) 文字コードの指定を行います。

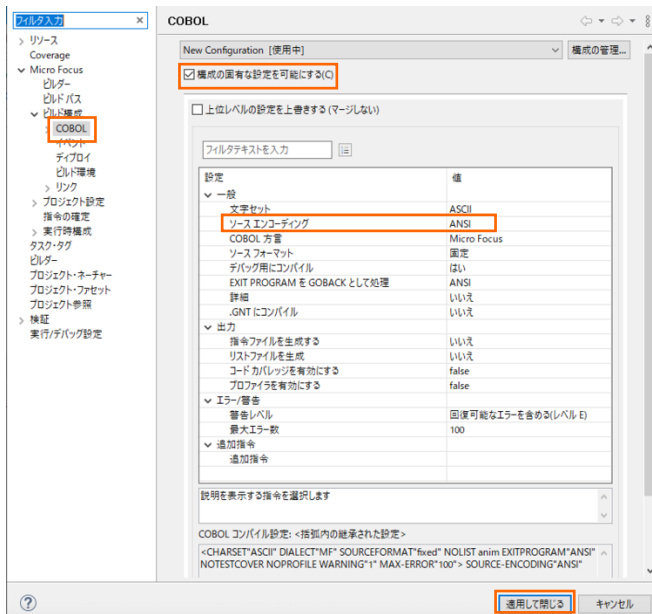
- ① Visual COBOL 9.0 より Shift-JIS を指定して日本語を表示する場合、文字コードの指定を明確に行う必要があります。最初に、[Window]メニュー > [設定]より[一般] > [ワークスペース]とナビゲートし、テキストファイルエンコードを「MS932」に変更し、[適用して閉じる]ボタンをクリックします。



※Preference Recorder のダイアログが表示されたら [キャンセル] を選択してください。

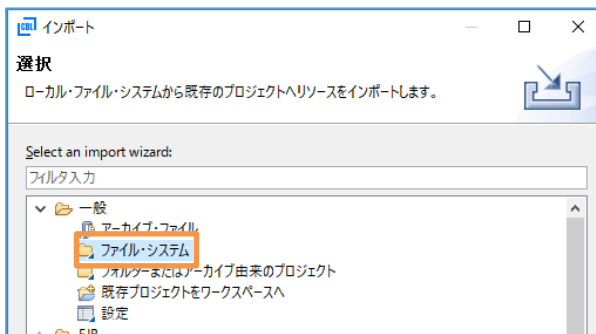
- ② 次に作成した COBOL プロジェクトを選択した状態で、マウスの右クリックにてコンテキストメニューを表示し、「プロパティ」を選択します。[Micro Focus] > [ビルド構成] > [COBOL]とナビゲートし、「構成の固有な設定を可能にする

(C)]にチェックを入れて[一般] > [ソース エンコーディング]を”UTF-8”から”ANSI”に変更し、[適用して閉じる]ボタンをクリックします。

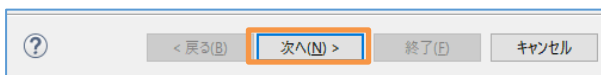


4) プログラムソースのインポート

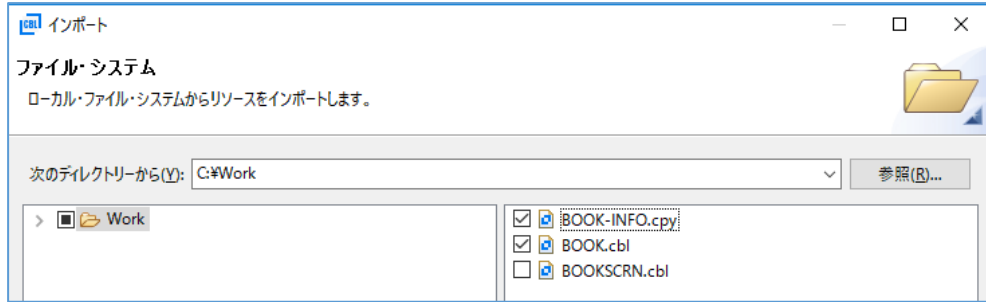
- ① COBOL エクスプローラーのパーズペクティブを開き、COBOL エクスプローラービューにて プロジェクトフォルダを右クリックし、コンテキストメニューから [インポート] > [インポート] を選択します。
- ② 既存のソースコードをロードします。一般のフォルダを展開し、[ファイル・システム] を選択し、[次へ(N)] ボタンをクリックします。



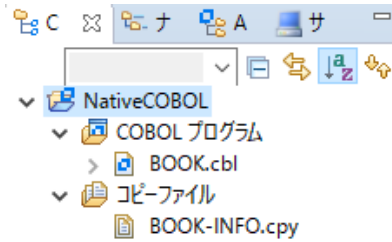
途中省略



- ③ チュートリアル用のファイルをインポートします。インポートダイアログが表示されるので [参照(R)] ボタンをクリックし、任意のフォルダにダウンロードしたファイルを解凍し、そのフォルダを指定します。下図では ”C:\work” に解凍したファイルを指定しています。ここで ”BOOK-INFO.cpy” と ”BOOK.cbl” を指定し、[終了(F)] ボタンをクリックします。

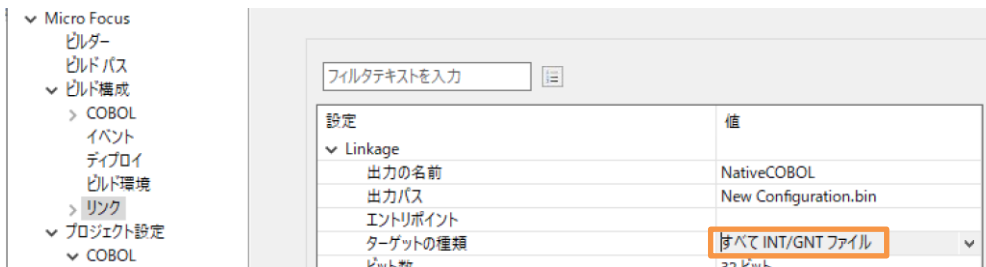


- ④ プロジェクトフォルダを展開し、2つのファイルが正常にロードされていることを確認します。

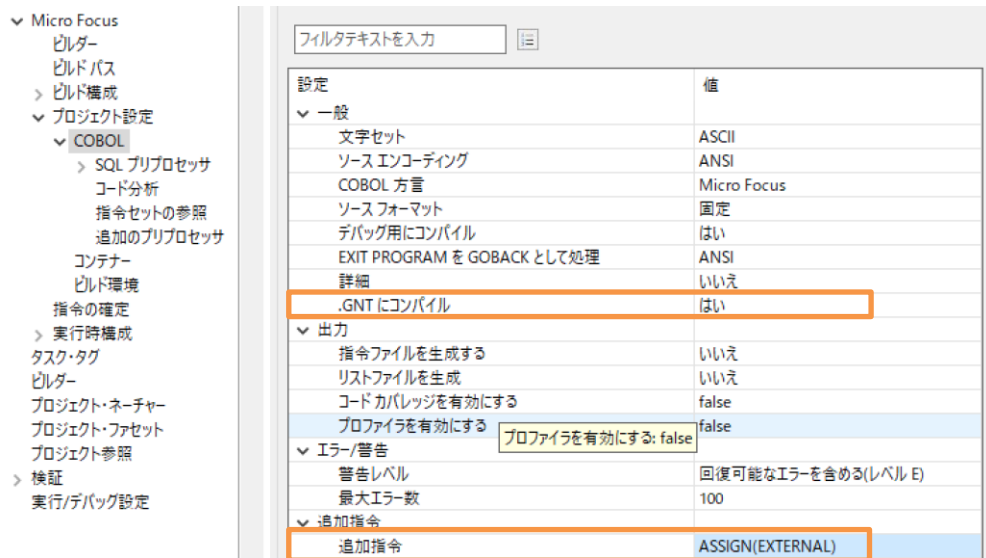


5) ビルドオプションの変更

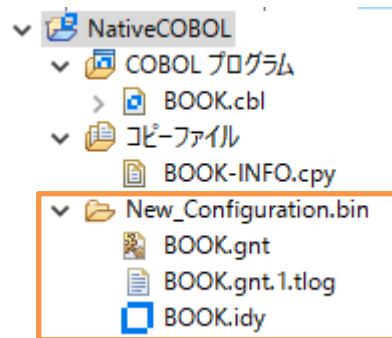
- ① プロジェクトの構成を変更します。COBOL エクスプローラーにて作成した「NativeCOBOL」プロジェクトを右クリックし、コンテキストメニューから [プロパティ] を選択します。
- ② プロパティ設定ダイアログが表示されます。[Micro Focus] > [ビルド構成] > [リンク] をクリックし、[ターゲットの種類] を「すべて INT/GNT ファイル」に変更します。



- ③ 次に[Micro Focus] > [プロジェクト設定] > [COBOL] をクリックし、[.GNTにコンパイル] を「はい」に変更し、[追加指令]に ”ASSIGN(EXTERNAL)” を指定し、[適用して閉じる] ボタンをクリックします。

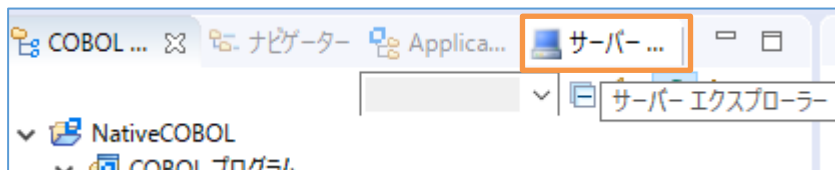


- ④ COBOL エクスプローラーにて ”New_Configuration.bin” を展開して下記のファイルが作成されていることを確認します。

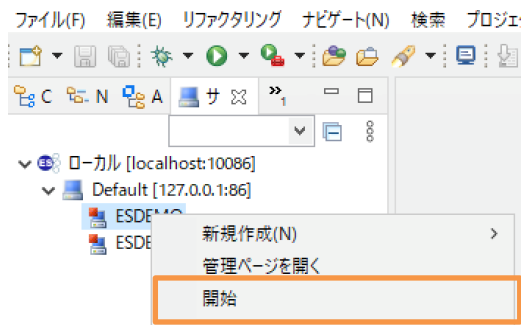


6) Enterprise Server の起動

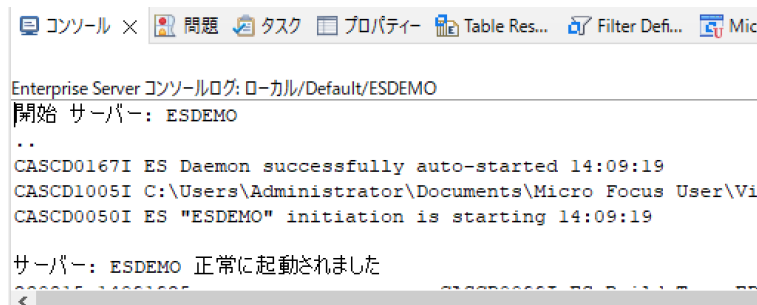
- ① [サーバーエクスプローラー] タブを選択します。



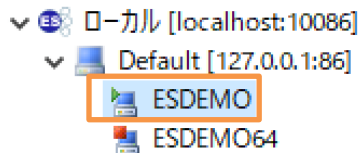
- ② Visual COBOL にビルドインされている Enterprise Server 「ESDEMO」 を右クリックし、コンテキストメニューから [開始] を選択します。もし、ダイアログが表示されたらそのまま[OK]ボタンを押してください。



- ③ Eclipse の Secure Storage に関するダイアログが表示された場合、[いいえ] を選択してください。開始処理の状況は、[コンソール] ビューでモニターできます。

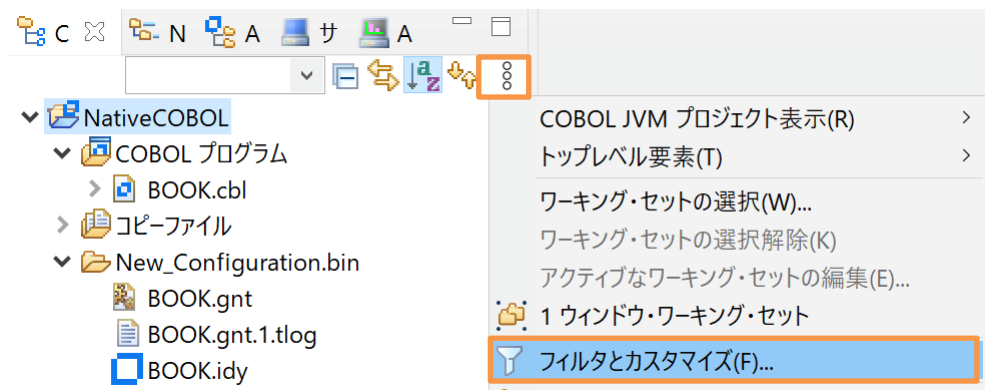


- ④ 正常に開始されると [サーバーエクスプローラー] ビュー上の「ESDEMO」アイコンが起動されたことを示す緑色のアイコンに切り替わります。

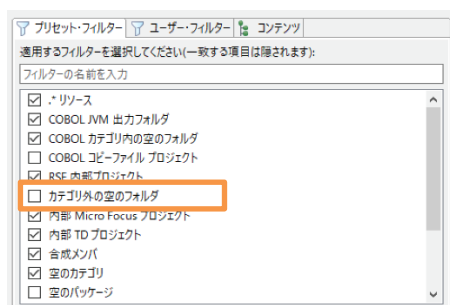


7) COBOL エクスプローラー表示設定の変更

- ① COBOL エクスプローラーに戻ります。
 ② COBOL エクスプローラー右上の「↑↓」アイコンの右横にあるアイコンをクリックし、[フィルタとカスタマイズ] を選択します。



- ③ [プリセットフィルター]タブの[カテゴリ外の空のフォルダ] にチェックされている場合は、チェックを外したのち、[OK] ボタンをクリックします。



8) デploy用フォルダの作成

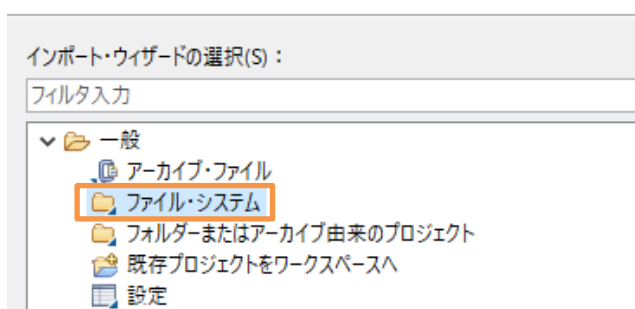
- ① 「NativeCOBOL」プロジェクト上で右クリックし、コンテキストメニューから [新規作成]→[フォルダ] を選択します。
- ② 「NativeCOBOL」プロジェクトが選択されていることを確認の上、フォルダ名に "deploy" を指定し、[終了(F)] ボタンをクリックします。

9) 「.mfdeploy」ファイルのインポート

- ① 作成した「deploy」フォルダ上で右クリックし、コンテキストメニューから [インポート]→[インポート] を選択します。
- ② 「一般」配下の「ファイル・システム」を選択し、[次へ(N)] ボタンをクリックします。

選択

ローカル・ファイル・システムから既存のプロジェクトへリソースをインポートします。

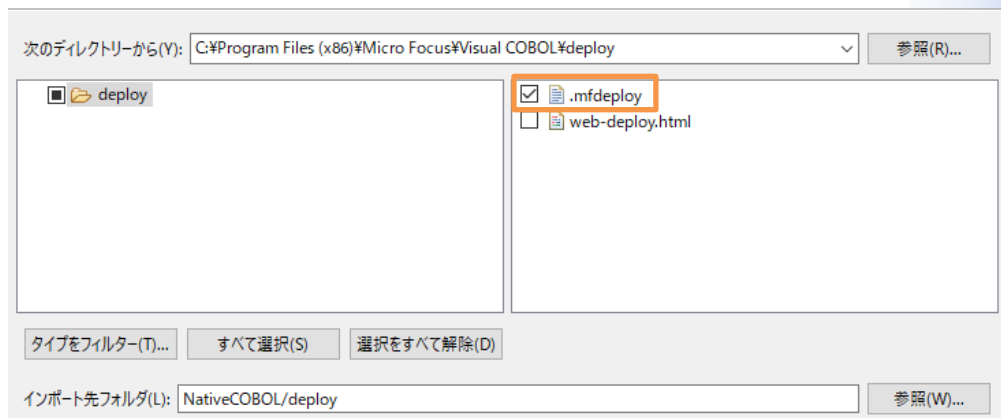


- ③ [参照(R)] ボタンをクリックし、Visual COBOL インストールフォルダ¥deploy を選択します。

- ④ [.mfdeploy] ファイルにチェックを行い、[終了(F)] ボタンをクリックします。

ファイル・システム

ローカル・ファイル・システムからリソースをインポートします。



補足)

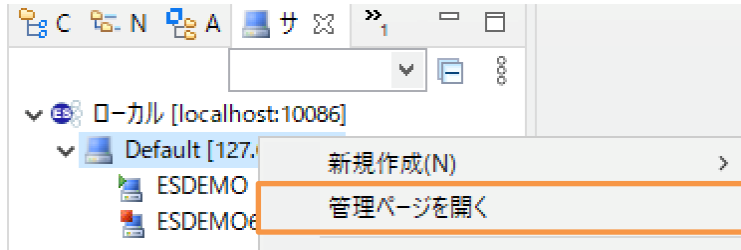
上記手順完了後も、COBOL エクスプローラー上の NativeCOBOL\deploy フォルダ配下に .mfdeploy ファイルは表示されませんが、これはフィルタによるものです。

正しくインポートされたことを確認する場合は、さきほど同様、[フィルタとカスタマイズ] を選択し、[.* リソース] のチェックを外すことで表示されるようになります。しかし、リソースに関する設定情報が表示されるようになるため、通常はチェックを行い、非表示状態とすることを推奨します。

3.2. Enterprise Server に登録するサービスのディプロイ先変更

1) RESTful Web サービスのディプロイ先フォルダの変更

- ① [サーバーエクスプローラー] タブを選択します。
- ② [ローカル localhost:86] 上で右クリックし、コンテキストメニューから [管理ページを開く] を選択します。



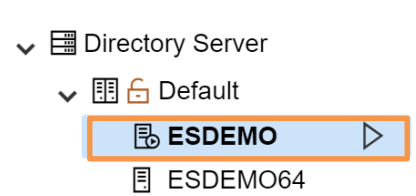
③ ESCWA (Enterprise Server Common Web Administration) 画面にてダッシュボードが表示されます。



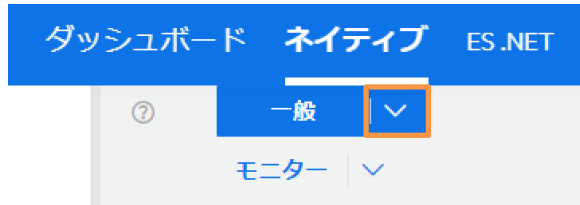
④ [ネイティブ]タブメニューをクリックします。



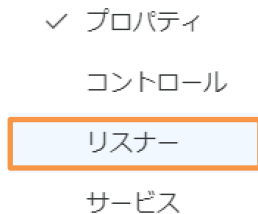
⑤ 次に左側メニューの[Directory Server]をドリルダウンして、[Default] → [ESDEMO]をクリックします。



⑥ [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。



- ⑦ 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。



- ⑧ [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。



- ⑨ 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の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/NativeCOBOL/deploy

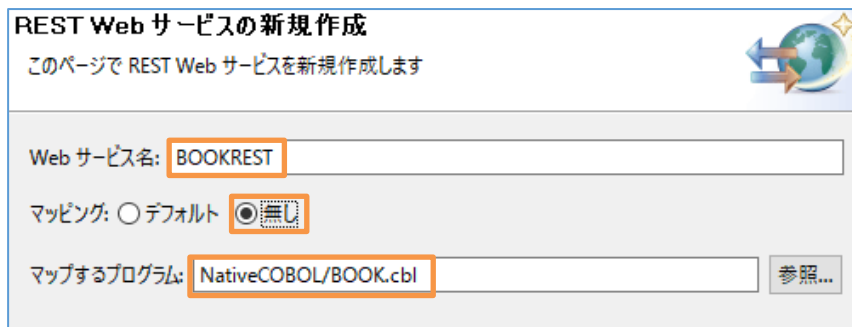
- ⑩ 正常に更新された旨のポップアップが表示されます。

※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

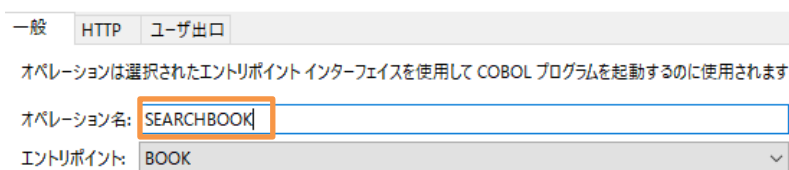
1) RESTful Web サービスのプロファイル作成

- ① Eclipse IDE の COBOL エクスプローラーに戻ります。
- ② RESTful Web サービスとして利用するビジネスロジックを処理するプログラム「BOOK.cb1」を右クリックし、コンテキストメニューから [新規作成] > [REST Web サービス] を選択します。
- ③ REST Web サービスの新規作成ウィザードが表示されます。[Web サービス名] 欄に "BOOKREST" を指定します。[マッピング] 欄は「無し」を選択、[マップするプログラム] 欄には「NativeCOBOL/BOOK.cb1」が選択されていることを確認し [終了(F)] ボタンをクリックします。



2) 書籍データ検索機能のオペレーションを作成

- ① 1) で作成した Web サービスプロファイル「BOOKREST」を右クリックし、コンテキストメニューから [新規作成] > [オペレーション] を選択します。
- ② オペレーションプロパティウィンドウが表示されます。[オペレーション名] 欄に “SEARCHBOOK” を入力します。

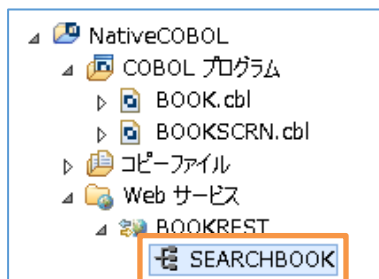


- ③ 次に [HTTP] タブを選択します。HTTP メソッドが [POST] になっていることを確認し、[OK] ボタンをクリックします。

3) 書籍データ検索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

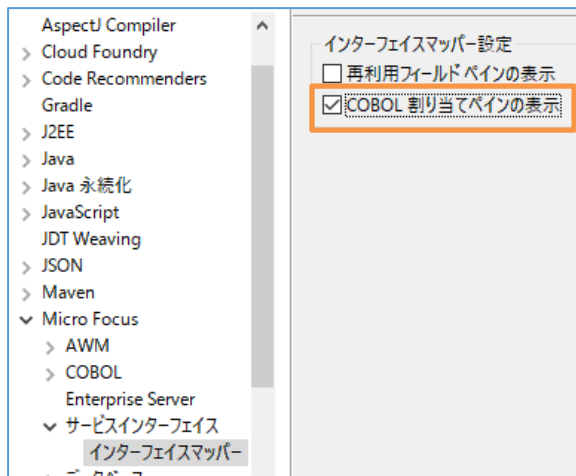
Enterprise Server は Web サービス側のデータ型と COBOL のデータ型を相互に自動変換させる機能を装備しています。この機能により Web サービスコンシューマー側と COBOL 側はそれぞれ相手のデータ型を意識することなく透過的にデータ変換が処理されやりとりができます。

- ① 「SEARCHBOOK」オペレーションをダブルクリックします。

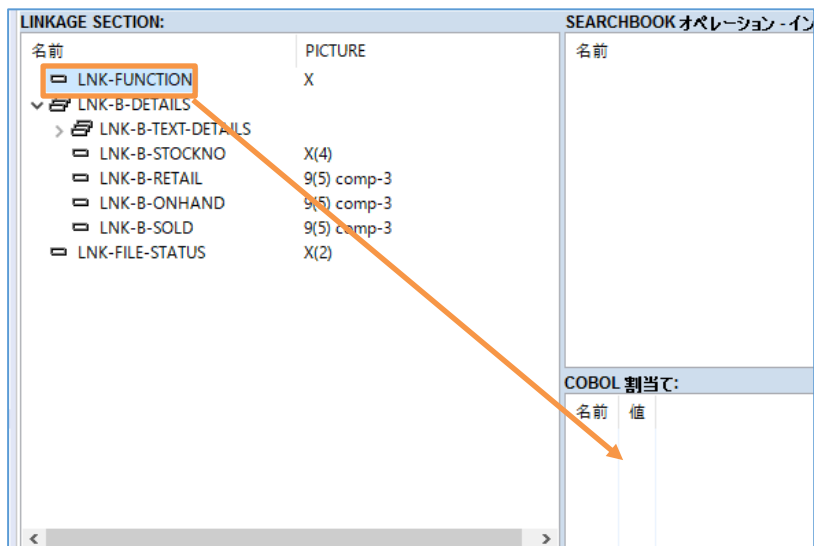


- ② [ウィンドウ] メニューから [設定] を選択し、[Micro Focus] > [サービスインターフェース] > [インターフェイスマッパー] を選択します。[COBOL 割り当てペインの表示] にチェックを入れ [適用して閉じる] ボタンをクリックします。

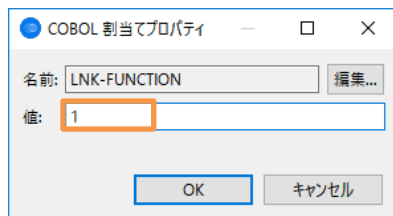
※Preference Recorder のダイアログが表示されたら [キャンセル] を選択してください。



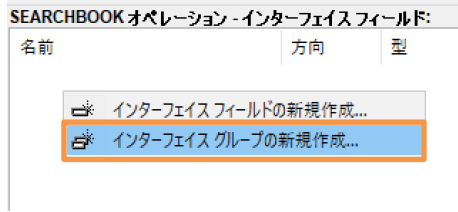
- ③ LINKAGE SECTION の COBOL の変数「LNK-FUNCTION」を [COBOL 割り当て] にドラッグ&ドロップします。



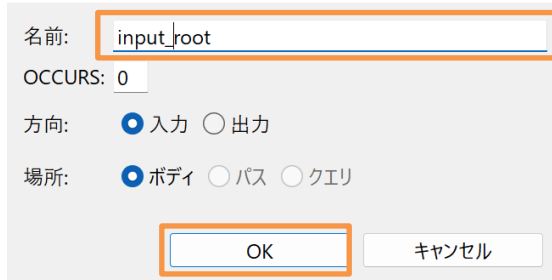
- ④ [COBOL 割当てプロパティ] ダイアログが表示されるので [値] に "1" を設定して [OK] ボタンをクリックします。



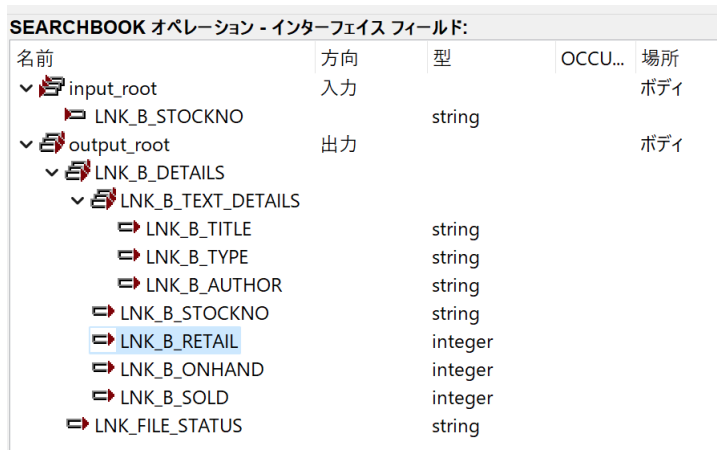
- ⑤ 「SEARCHBOOK オペレーション - インターフェイス フォールド」にて右クリックし、コンテキストメニューから [インターフェイス グループの新規作成] をクリックします。



- ⑥ [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。



- ⑦ 「SEARCHBOOK オペレーション - インターフェイス フィールド」にて右クリックし、コンテキストメニューから「インターフェイスグループの新規作成」をクリックします。
- ⑧ [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。
- ⑨ [LINKAGE SECTION] 側にある「LNK_B_STOCKNO」を「input_root」にドラッグ&ドロップします。
- ⑩ [LINKAGE SECTION] 側にある「LNK-B-DETAILS」を「output_root」にドラッグ&ドロップします。
- ⑪ 最後に [LINKAGE SECTION] 側にある「LNK-FILE-STATUS」を「output_root」にドラッグ&ドロップします。最終的には下のイメージのような構成になります。



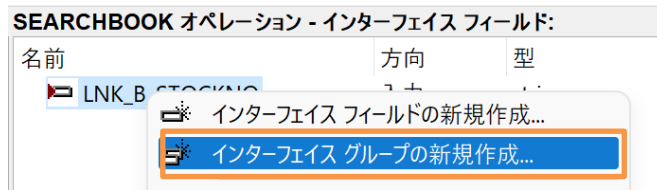
4) 書籍データ追加機能のオペレーションを追加

- ① 1) で作成した Web サービスプロファイル「BOOKREST」を右クリックし、コンテキストメニューから [新規作成] > [オペレーション] を選択します。
- ② オペレーションプロパティウィンドウが表示されます。[オペレーション名] 欄に “ADDBOOK” を入力します。
- ③ 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。

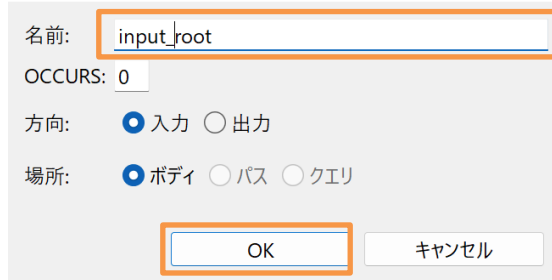
5) 書籍データ登録機能オペレーションのインターフェイスマッピングを定義

- ① 「ADDBOOK」オペレーションをダブルクリックします。

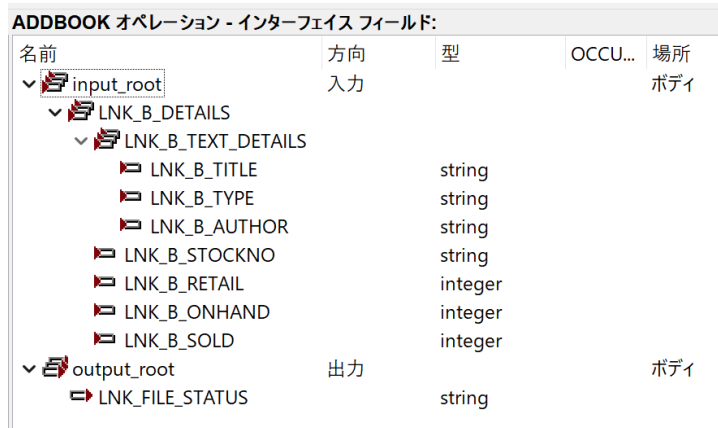
- ② 「LNK-FUNCTION」を COBOL 割当てにドラッグ&ドロップし、[値] には “2” を指定します。
- ③ 「ADDBOOK オペレーション - インターフェイス フォールド」にて右クリックし、コンテキストメニューから「インターフェイスグループの新規作成」をクリックします。



- ④ [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。



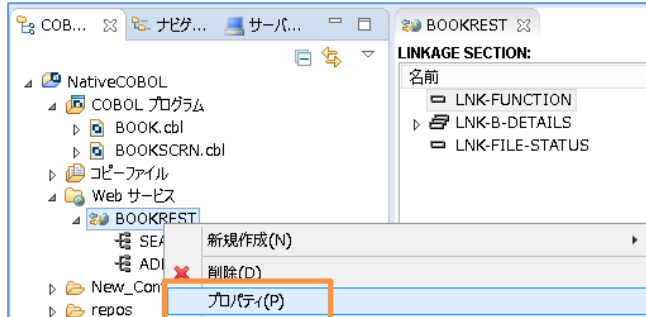
- ⑤ 「ADDBOOK オペレーション - インターフェイス フォールド」にて右クリックし、コンテキストメニューから「インターフェイスグループの新規作成」をクリックします。
- ⑥ [グループプロパティ] ウィンドウにて “output_root” と入力、[方向] を [出力] にして [OK] ボタンをクリックします。
- ⑦ [LINKAGE SECTION] 側にある「LNK-B-DETAILS」を「input_root」にドラッグ&ドロップします。
- ⑧ [LINKAGE SECTION] 側にある「LNK-FILE-STATUS」を「output_root」にドラッグ&ドロップします。最終的には下のイメージのような構成になります。



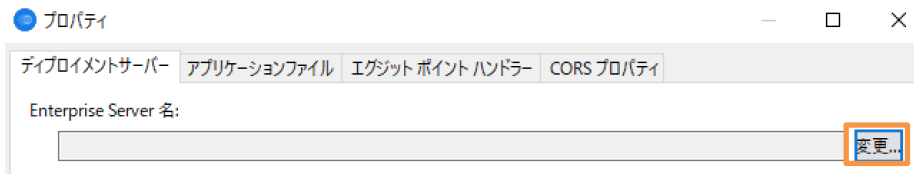
3.4. コンパイルした COBOL アプリケーションを Enterprise Server ヘッドプロイ

1) Enterprise Server へのディプロイ情報を指定

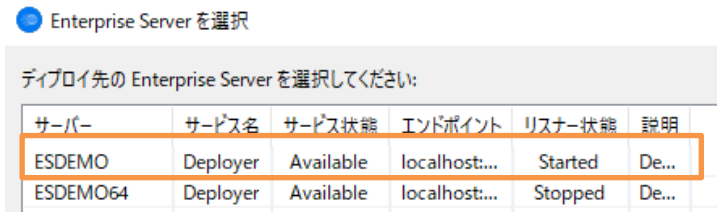
- ① COBOL エクスプローラーにて追加した Web サービス「BOOKREST」を右クリックし、コンテキストメニューから「プロパティ(P)」を選択します。



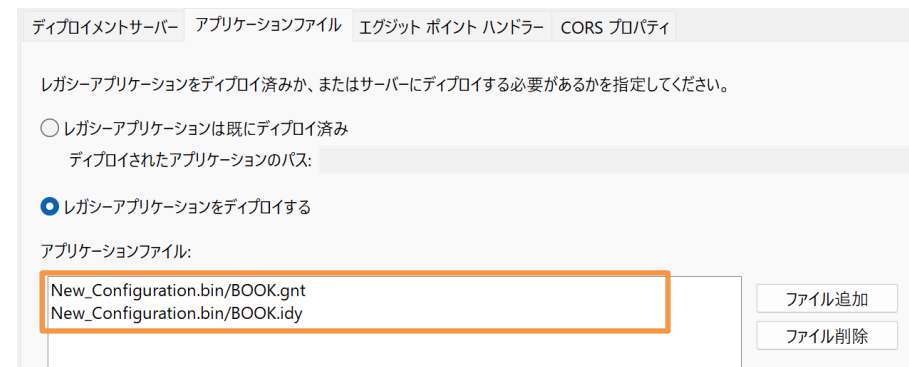
- ② [ディプロイメントサーバー] タブを選択し、[変更] ボタンをクリックします。



- ③ 起動済みの Enterprise Server 「ESDEMO」を選択し、[OK] ボタンをクリックします。



- ④ 次に [アプリケーションファイル] タブを選択し、「レガシーアプリケーションをディプロイする」を選択します。
- ⑤ [ファイル追加] ボタンを押して、プロジェクトディレクトリ配下の「New_Configuration.bin」に生成された「BOOK.gnt」および「BOOK.idy」を選択し、[開く(O)] ボタンをクリックします。
- ⑥ アプリケーションファイルが入ったマッピングプロパティ画面に戻るので [OK] ボタンをクリックします。

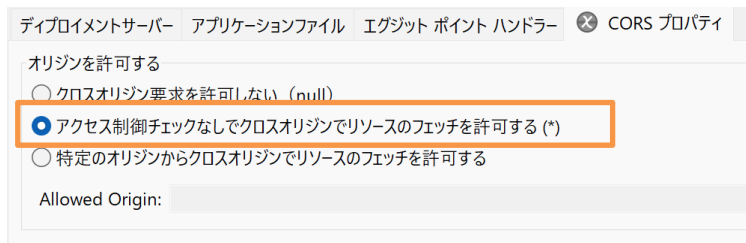


2) オリジン間リソース共有 (CORS) を許可

- ① 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。

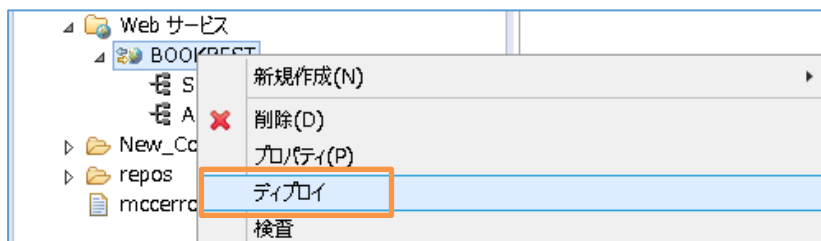
- ② 「BOOKREST」 Web サービスを選択して、右クリックしコンテキストメニューから[プロパティ]を選択します。
- ③ [CORS プロパティ]タブを選択し、運用用途に合わせた設定を行います。ここでは「アクセス制限チェックなしでクロスオリジンでリソースのフェッチを許可する(*)」を選択しています。



- ④ [OK] ボタンをクリックします。

3) RESTful Web サービスを Enterprise Server へデプロイ

- ① COBOL エクスプローラーにて作成した Web サービス「BOOKREST」を右クリックし、コンテキストメニューから [検査] を選択し、エラーが出ていないことを確認します。
- ② COBOL エクスプローラーにて作成した Web サービス「BOOKREST」を右クリックし、コンテキストメニューから [デプロイ] を選択します。



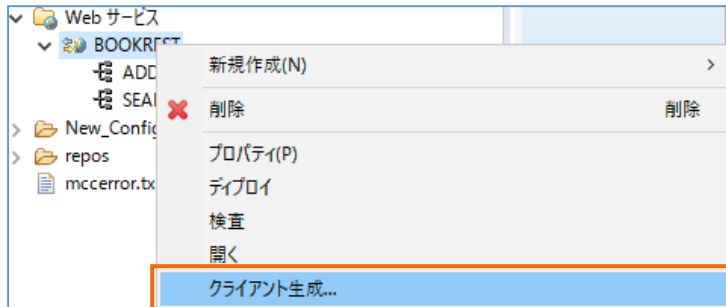
- ③ ブラウザー上の「ESCWA サーバー」に切り替えます。
- ④ [一般]メニューから[サービス]をクリックします。
- ⑤ 画面を下にスクロールしていくと最下行にデプロイした RESTful Web サービスが追加されていることを確認します。

サービス名	ステータス	タイプ	パス	フォーマット	作成日時
#SEARCHBOOK	Available	Web ...	/temppath/BOOKREST/1.0	MFRHJSON	created 10:
#ADDBOOK	Available	Web ...	/temppath/BOOKREST/1.0	MFRHJSON	created 10:

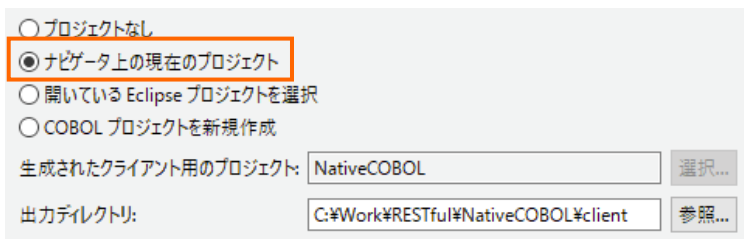
3.5. RESTful Web サービスのテスト

1) デプロイした RESTful Web サービスをテストするための COBOL アプリケーションを生成する

① 「BOOKREST」を右クリックし、コンテキストメニューから「クライアント生成...」を選択します。



② 「クライアントを生成」ダイアログが表示されるので「ナビゲータ上の現在のプロジェクト」を選択し、[OK] ボタンをクリックします。



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

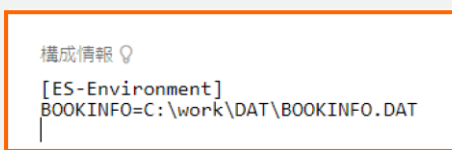
① ブラウザー上の ESCWA 画面に切り替えます。

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。※ファイルはあらかじめこのディレクトリに必ずコピーしておいてください。

[ES-Environment]

BOOKINFO=C:\work\DAT\BOOKINFO.DAT

追加設定



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。

3) Enterpriser Server の再起動

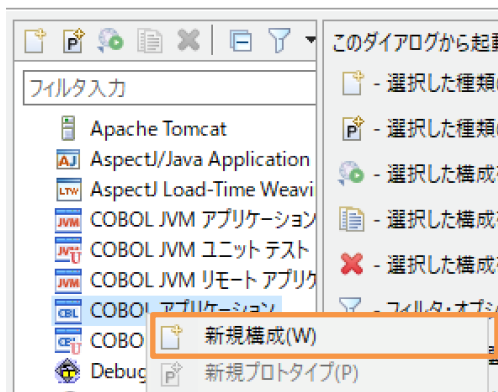
- ① Eclipse IDE に戻り、「サーバーエクスプローラー」に切り替えます。
- ② 「ESDEMO」上で右クリックし、コンテキストメニューから [再起動] を選択し、Enterprise Server を再起動します。

4) 生成したテスト用 COBOL クライアントの実行（登録処理）

- ① COBOL エクスプローラーにて「NativeCOBOL」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 [実行] > [実行の構成] を選択します。
- ② 左側より「COBOL アプリケーション」を右クリックし、コンテキストメニューから [新規構成] を選択します。

構成の作成、管理、および実行

COBOL プログラムを実行します



- ③ [一般] タブの 以下の入力を行い、[実行] ボタンをクリックします。
[名前] にて “NativeCOBOL-client”
[主プログラム] にて “New_Configuration.bin/BOOKREST-app.gnt”
- ④ 「Service Address」は、今回変更不要のため、Enter 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、[リスナー]メニュー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。

- ⑤ Supplemental Query String (Optional) は変更しないのでそのまま Enter を押します。
- ⑥ 「Username」、「Password」は変更しないのでそのまま 2 回 Enter を押します。
- ⑦ 「Operation (1 = SEARCHBOOK, 2 = ADDBOOK)」は、“2” を入力して Enter を押します。
- ⑧ 「LNK_B_TITLE」は “PLANET OF THE APES” を入力して Enter を押します。
- ⑨ 「LNK_B_TYPE」は “SCIENCE FICTION” を入力して Enter を押します。
- ⑩ 「LNK_B_AUTHOR」は “PIERRE BOULLE” を入力して Enter を押します。
- ⑪ 「LNK_B_STOCKNO」は、“5555” を入力して Enter を押します。
- ⑫ 「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。
- ⑬ 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。

```

Service Address (Enter = http://localhost:9003):
Supplemental Query String (optional):
Username (optional):
Password (optional):
Operation (1 = SEARCHBOOK, 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
続行するには何かキーを押してください . . .

```

5) 生成したテスト用 COBOL クライアントの実行 (検索処理)

- ① COBOL エクスプローラーにて「ネイティブ COBOL」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 [実行] > [実行の構成] を選択します。
- ② 左側の [COBOL アプリケーション] 配下の「NativeCOBOL-client」を選択し、[実行] ボタンをクリックします。
- ③ 「Service Address」は、デフォルトのまま使いたいので Enter キーを押します。
- ④ 「Username」、「Password」は変更しないのでそのまま 2 回 Enter を押します。
- ⑤ 「Operation (1 = SEARCHBOOK, 2 = ADDBOOK)」は、「1」を入力して Enter を押します。
- ⑥ 「LNK_B_STOCKNO」は、「5555」を入力して Enter を押します。
- ⑦ RESTful の Web サービスが実行され、データが返ってきます。以下のような内容が確認できます

```

Service Address (Enter = http://localhost:9003):
Supplemental Query String (optional):
Username (optional):
Password (optional):
Operation (1 = SEARCHBOOK, 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
続行するには何かキーを押してください . . .

```

6) curl コマンドによる RESTful Web サービスのテスト

作成した RESTful Web サービスは、一般的な仕様であるため、これまで確認した COBOL のクライアントプログラムだけではなく、Java などの他の開発言語などからもアクセスできます。

ここでは、curl コマンドを用いてリクエストが行えることを確認します。

- ① Windows のコマンドプロンプト画面を開き、チュートリアル用のファイルを解凍したフォルダに移動します。ここまでの手順では、C:\¥Work に解凍しています。
- ② 以下のコマンドを実行し、さきほど登録した書籍情報を検索します。

```
curl http://localhost:9003/temp/path/BOOKREST/1.0/SEARCHBOOK -d @json¥search5555.txt
```

```
C:\¥Work>curl http://localhost:9003/temp/path/BOOKREST/1.0/SEARCHBOOK -d
@json¥search5555.txt
```

```

{
  "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"
}
C:¥Work>

```

3.6. RESTful Web サービスのデバッグ

- 1) ブラウザー上の「ESCWA サーバー」に切り替えます。
- 2) 画面左側より「ESDEMO」を選択し、上部より [一般] をクリックします。
- 3) [動的デバッグを許可] にチェックを行い、[適用] をクリックします。

一般的なプロパティ |  適用 削除

開始オプション

* 入力必須の項目です

名前*  システムディレクトリ 

共有メモリ ページ数*   ページ数(4k): 共有メモリ クッション*   ページ数(4k):

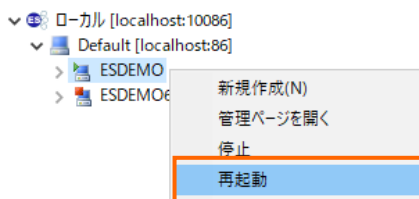
SEP数*  

コンソールログサイズ*   k

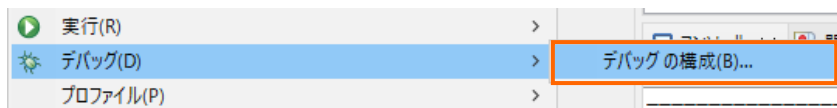
ローカルコンソールを表示  動的デバッグを許可  システム起動時に開始する 

64ビット作業モード  以前のログを削除 

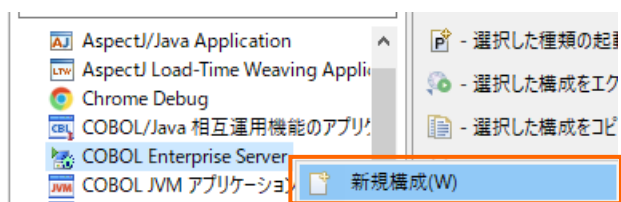
- 4) Eclipse IDE に戻り、サーバーエクスプローラーを開きます。
- 5) [ESDEMO] を選択し、マウスの右クリックにてコンテキストメニューを開いたうえで、[再起動] を選択します。



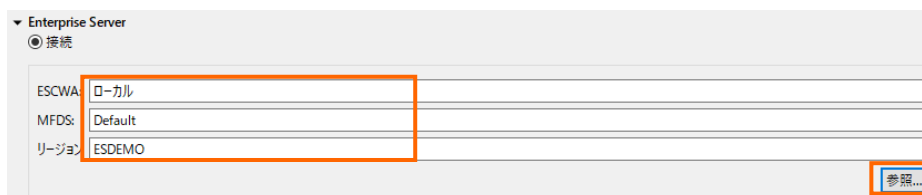
- 6) Eclipse IDE からのデバッグアタッチ
 - ① Eclipse IDE の COBOL エクスプローラーに戻ります。
 - ② 「NativeCOBOL」プロジェクトを選択し、マウスの右クリックにてコンテキストメニューを開いたうえで、[デバッグ(D)] > [デバッグの構成(B)] を選択します。



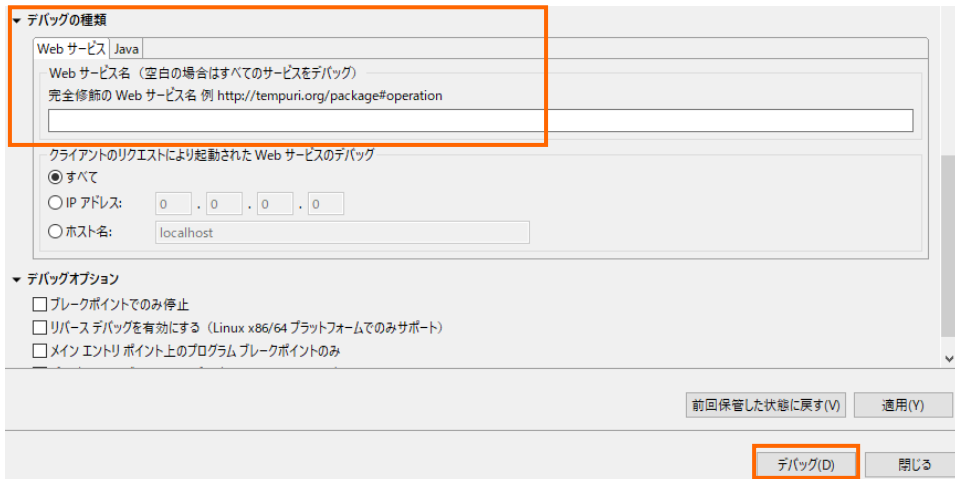
- ③ 画面左側より、[COBOL Enterprise Server] を選択し、マウスの右クリックにてコンテキストメニューを開いたうえで、[新規構成]を選択します。



- ④ 「Enterprise Server」セクションの [参照] をクリックして、[ESDEMO] を選択します。選択後は、次のような表示になります。

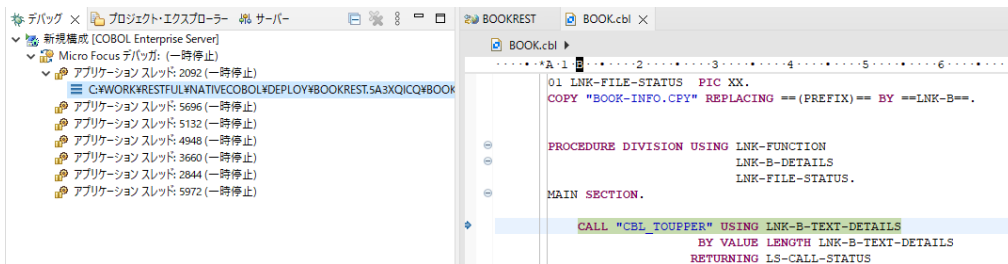


- ⑤ 「デバッグの種類」セクションで [Web サービス] タブを選択し、「Web サービス名」が空欄であることを確認したうえで、[デバッグ] をクリックします。



パースペクティブの切り替え確認ダイアログが表示された場合は、[はい(Y)] をクリックします。

- ⑥ 3.5 で実施した curl コマンドを用いた検索処理のリクエストを行ってください。
デバッグ停止するため、結果がすぐ戻りません。
- ⑦ Eclipse IDE に戻ると、デバッグ停止中になっています。



- ⑧ コンソールアプリケーションのデバッグと同様、[実行] メニュー配下の [再開] や [ステップイン] の操作が行えます。



ステップインでは、プログラムを 1 行ずつ進めることができます。

再開、もしくは、プログラム終了まで進めると、デバッグ中断が終了し、curl コマンドの結果が前回同様に戻されます。

- ⑨ [実行] > [終了] を選択して、デバッグを終了します。



3.7. インスタンスの停止

- 1) Enterpiser Server の停止
 - ① 「COBOL パースペクティブ」に切り替えます。
 - ② 「サーバーエクスプローラー」に切り替えます。
 - ③ 「ESDEMO」上で右クリックし、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。