

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

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。

- 開発環境

OS	Windows 11
COBOL 開発環境製品	Visual COBOL 11.0 Patch Update 01 for Eclipse

- 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、任意のフォルダー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
- 3 チュートリアル手順
 - 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
 - 3.5.1 サービス実行時にアプリケーションがアクセスするデータファイルの設定
 - 3.5.2 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

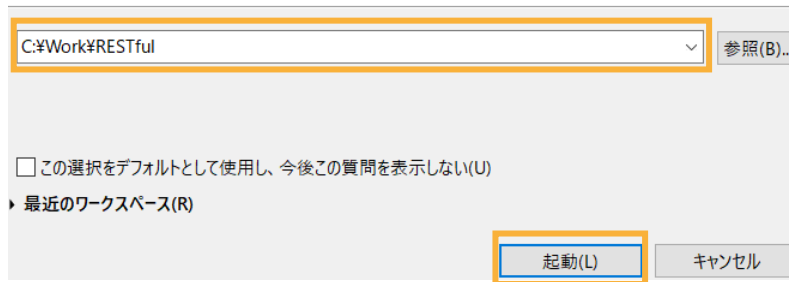
3.1 Windows クライアントでの開発準備作業

1) Visual COBOL for Eclipse を起動

- ① [スタート] メニュー > [すべてのアプリ] > [Rocket 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:¥Work¥RESTful” を指定し、[起動(L)] をクリックします。

ディレクトリをワークスペースとして選択

Eclipse は、ワークスペースディレクトリを使用して、環境設定と開発成果物を保存します。



起動後、ようこそ画面は閉じてください。

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)] 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。

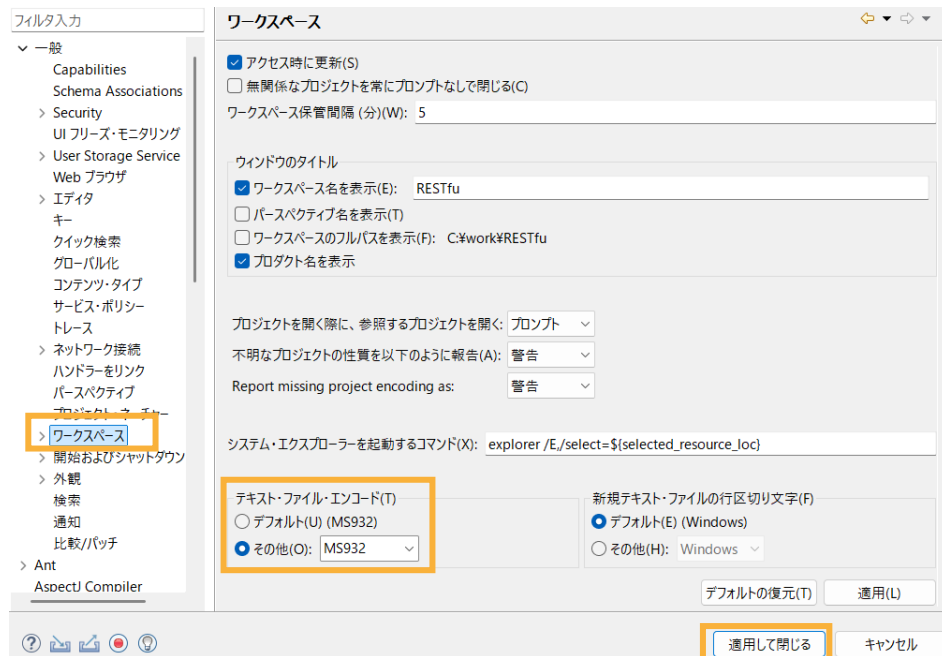
COBOL プロジェクト

ワークスペースまたは外部の場所に COBOL プロジェクトを作成します。




3) 文字コードの指定を行います。

- ① 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が“MS932”となっているかを確認してください。異なる値が設定されている場合は、“MS932”を選択の上、[適用して閉じる] をクリックします。



補足)

環境によっては、Windows-31J と表示されることがあります。その場合は、MS932 を Windows-31J に読み替えてください。

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。[Rocket Software] > [プロジェクト設定] > [COBOL] とナビゲートし、[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] をクリックします。

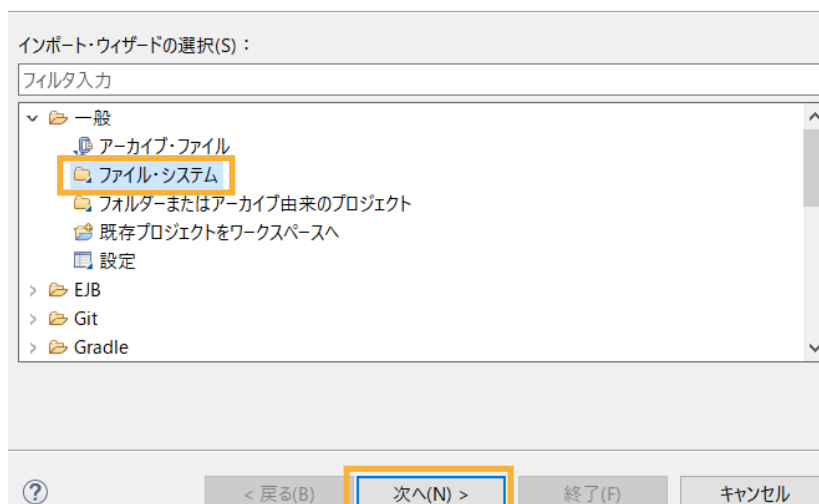


4) プログラムソースのインポート

- ① COBOL エクスプローラーのパースペクティブを開き、COBOL エクスプローラービューにて プロジェクトフォルダーを右クリックし、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。一般のフォルダーを展開し、[ファイル・システム] を選択し、[次へ(N)] 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ヘリソースをインポートします。



- 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)] をクリックし、任意のフォルダーにダウンロードしたファイルを解凍し、そのフォルダーを指定します。下図では "C:\%vc-tutorial" 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)] 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。



次のディレクトリーから(Y: C:\vc-tutorial) 参照(R)...

> **vc-tutorial**

- ☒ BOOK.cbl
- ☒ BOOK-INFO.cpy
- ☐ BOOKSCRN.cbl
- ☐ Readme.txt

タイプをフィルター(T)... すべて選択(S) 選択をすべて解除(D)

インポート先フォルダ(L): NativeCOBOL 参照(W)...

オプション

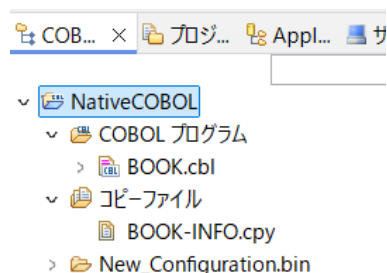
☐ 警告を出さずに既存リソースを上書き(O)

☐ トップ・レベルのフォルダーを作成(C)

拡張 >> (A)

? < 戻る(B) 次へ(N) > 終了(F) キャンセル

プロジェクトフォルダーを展開し、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。[Rocket Software] > [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。

フィルタ入力

- > リソース
 - Coverage
 - > Rocket Software
 - ビルダー
 - ビルドパス
 - > ビルド構成
 - > COBOL
 - イベント
 - ディプロイ
 - ビルド環境
 - > リンク**
- > プロジェクト設定
 - 指令の確認

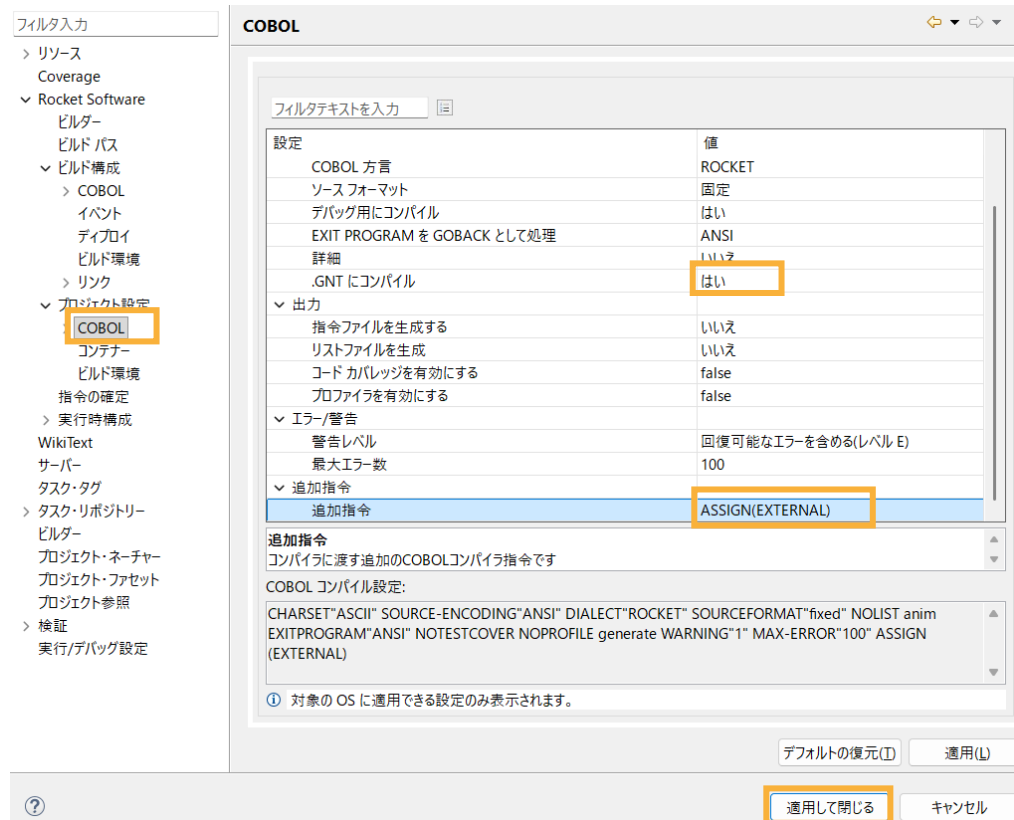
リンク

New Configuration [使用中] 構成の管理

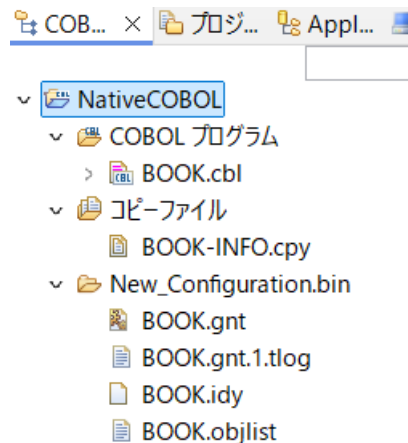
フィルタテキストを入力

設定	値
出力の名前	NativeCOBOL
出力パス	New Configuration.bin
エントリポイント	
ターゲットの種類	すべて INT/GNT ファイル
ビット数	32 ビット

- ③ 次に [Rocket Software] > [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] をクリックします。

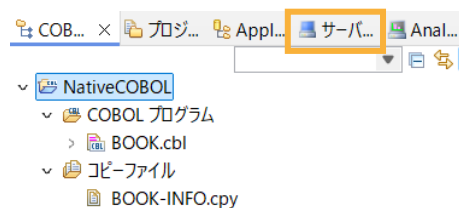


- ④ COB OL エクスプローラーにて “New_Configuration.bin” を展開して下記のファイルが作成されていることを確認します。



6) Enterprise Server の起動

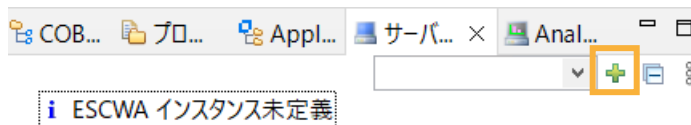
- ① [サーバーエクスプローラー] タブを選択します。



- ② [+] アイコンを押します。

補足)

すでに作成済みの場合は、②～③はスキップしてください。



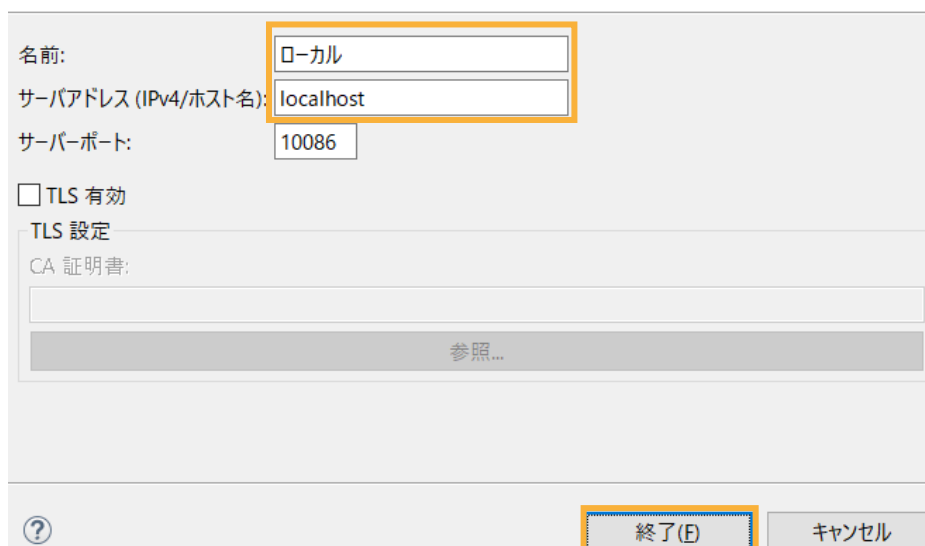
- ③ 以下の入力を行い、[終了(F)] をクリックします。

名前: “ローカル”

サーバアドレス(IPv4/ホスト名): “localhost”

接続の新規作成

既存の Enterprise Server Common Web Administration インスタンスへの接続を新規作成します

The screenshot shows a dialog box titled '接続の新規作成' (New Connection). It contains the following fields and controls:

- 名前:** A text box containing 'ローカル' (Local), highlighted with an orange box.
- サーバアドレス (IPv4/ホスト名):** A text box containing 'localhost', highlighted with an orange box.
- サーバポート:** A text box containing '10086'.
- TLS 有効:** A checkbox that is currently unchecked.
- TLS 設定:** A section containing a text box for 'CA 証明書:' and a '参照...' (Browse...) button.
- Buttons:** At the bottom right, there are two buttons: '終了(F)' (End) and 'キャンセル' (Cancel). The '終了(F)' button is highlighted with an orange box.

ESCWA のセキュリティが有効な場合は、以下のダイアログが表示されますので、以下の入力を行い、[OK] をクリックします。

ユーザー名 / パスワード: 設定した情報

[認証情報の保存] にチェック

ESCWA: ローカル の接続の詳細を入力してください

☒ サーバーに認証情報が必要

認証情報

ユーザー名: SYSAD

パスワード: *****

☒ 認証情報の保存

☐ 認証情報がクリアされるまで、再度プロンプトを表示しない

OK キャンセル

補足)

インストール直後に有効となっている際の認証情報は以下の手順で確認できます。

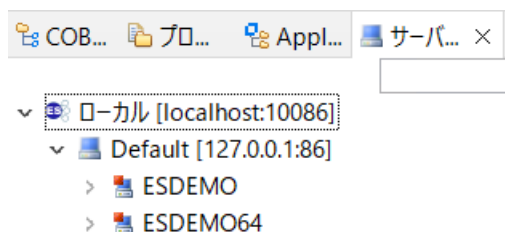
スタートメニューより、[Rocket Visual COBOL] > [Visual COBOL コマンドプロンプト(64 ビット)] を選択します。

コマンドライン上で “mfsecretsadmin read microfocus/temp/admin” のコマンドを実行します。

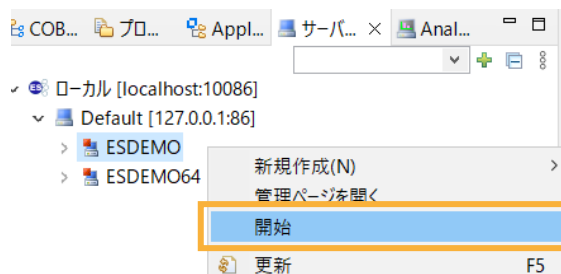
以下の場合、ユーザー名は “SYSAD”、パスワードは “LZQe4VS3” です。

```
C:\xa>mfsecretsadmin read microfocus/temp/admin
{"mfUser":"SYSAD", "mfPassword":"LZQe4VS3"}
C:\xa>
```

Enterprise Server インスタンスが追加されます。



- ④ 「ESDEMO」を選択し、マウスの右クリックでコンテキストメニューを開き、[開始] をクリックします。



認証情報には、さきほどの情報を入力してください。

サーバー ESDEMO の ESMAC 認証情報を入力してください

☒ サーバーに認証情報が必要

認証情報

ユーザー名:

パスワード:

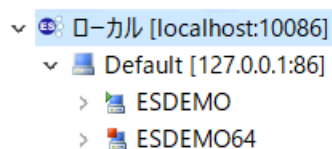
グループ: デフォルト グループは空白

☒ 認証情報の保存

☐ 認証情報がクリアされるまで、再度プロンプトを表示しない

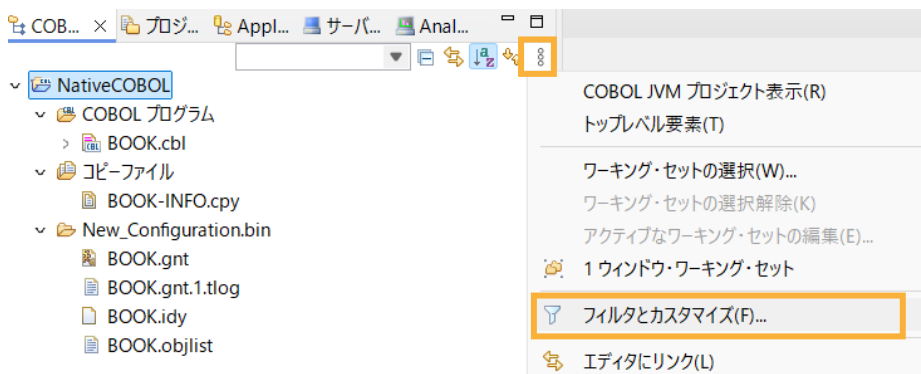
OK キャンセル

- ⑤ 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。

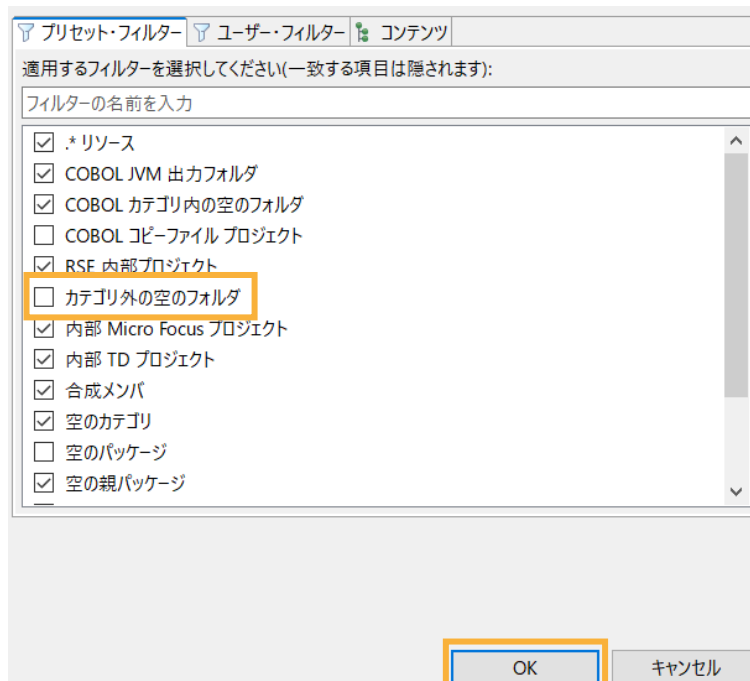


7) COBOL エクスプローラー表示設定の変更

- ① [COBOL エクスプローラー] タブをクリックします。
- 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。

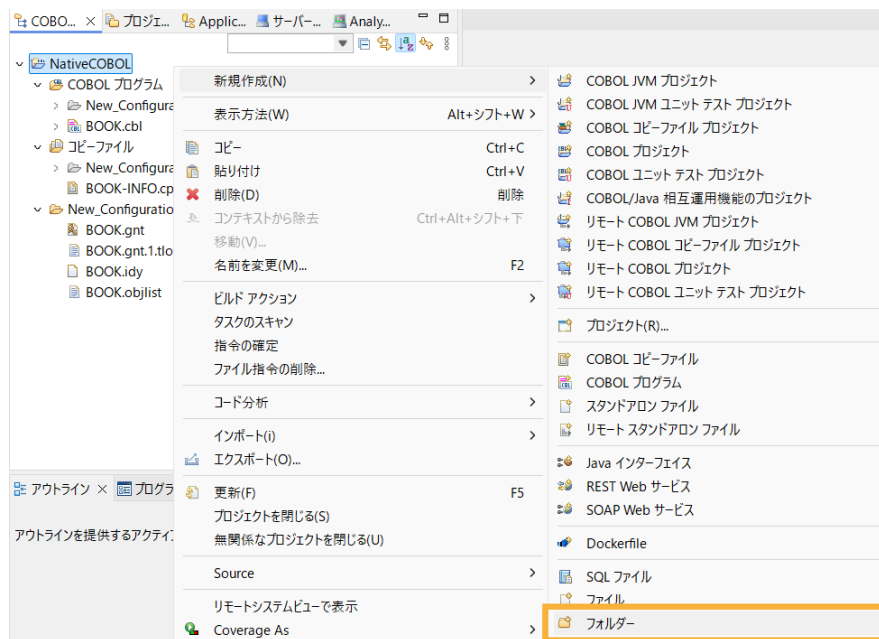


- ③ [プリセットフィルター] タブの[カテゴリ外の空のフォルダ] にチェックされている場合は、チェックを外したのち、[OK] をクリックします。



8) デploy用フォルダーの作成

- ① 「NativeCOBOL」プロジェクトを選択し、マウスの右クリックでコンテキストメニューを開き、[新規作成] > [フォルダー] 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)] をクリックします。

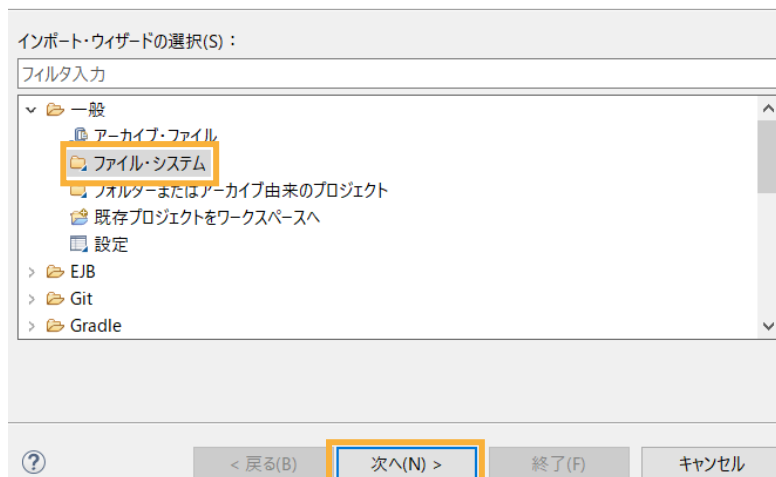


9) 「.mfdeploy」ファイルのインポート

- ① 作成した「deploy」フォルダーを選択し、右クリックでコンテキストメニューを開き、[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)] をクリックします。

選択

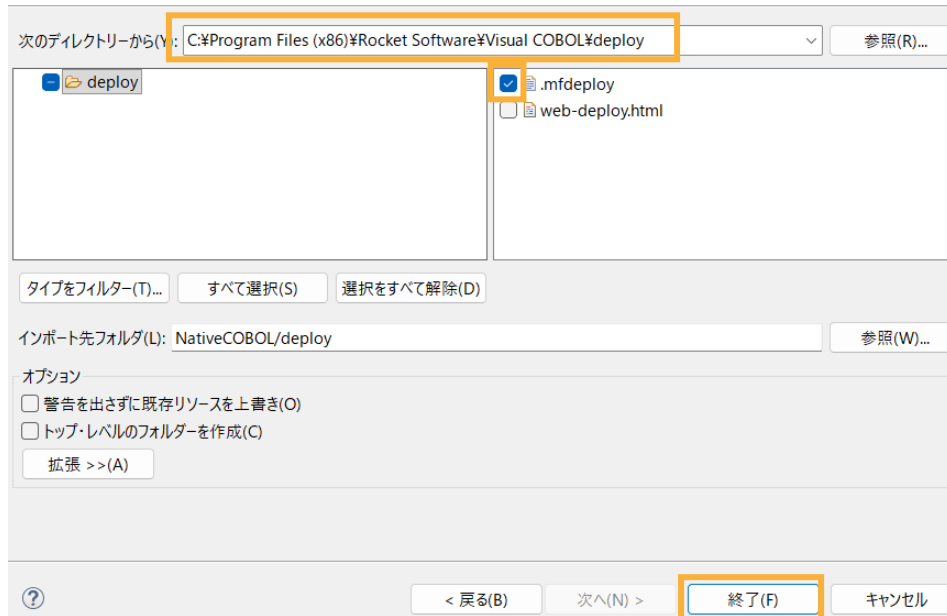
ローカル・ファイル・システムから既存のプロジェクトリソースをインポートします。



- ③ [参照(R)] をクリックし、Visual COBOL インストールフォルダー¥deploy を選択したうえで、[.mfdeploy] ファイルにチェックを行い、[終了(F)] 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)

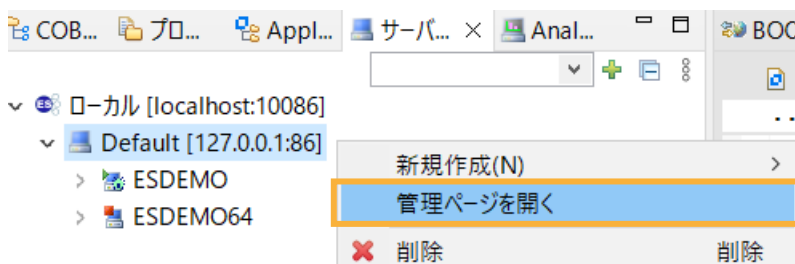
上記手順完了後も、COBOL エクスプローラー上の NativeCOBOL¥deploy フォルダ配下に .mfdeploy ファイルは表示されませんが、これはフィルタによるものです。

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ーの変更

- 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:10086] > [Default [127.0.0.1:86]] 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。



管理画面のログインには、さきほどの認証情報を使用してください。

Enterprise Server Administration

⚠ Rocket Software Enterprise Serverでは、インストール後に基本的なセキュリティ機能がデフォルトで有効になっていません。

[詳細情報](#)

ユーザー名

SYSAD

パスワード

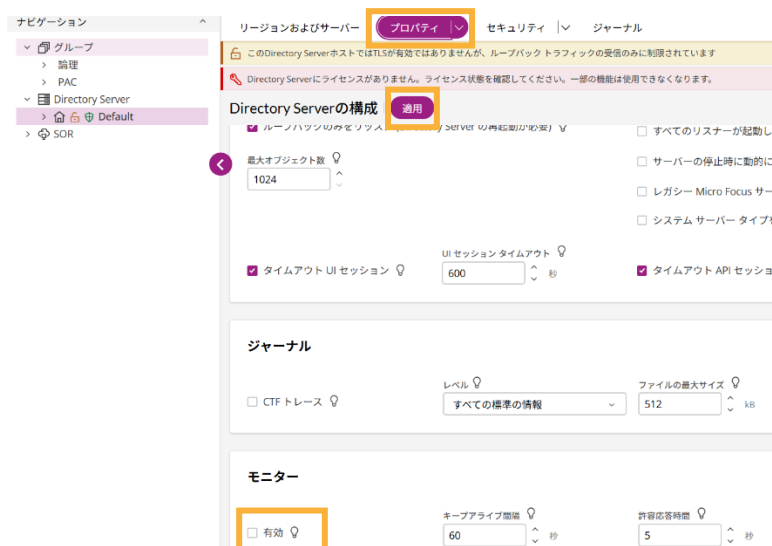
[パスワード変更](#)

[ログイン](#)

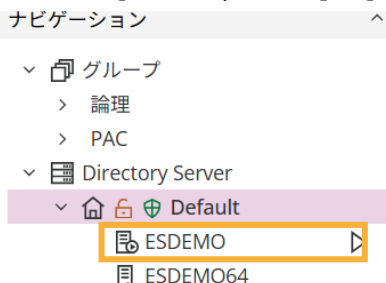
認証情報は、次のセキュリティ マネージャを使用
して検証されます: VSAM ESM

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

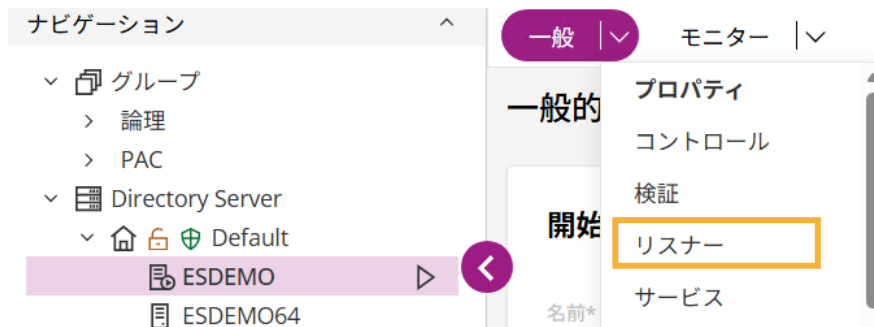
- ③ [\[プロパティ\]](#) をクリックし、[\[モニター\]](#) > [\[有効\]](#) のチェックをはずしたうえで、[\[適用\]](#) をクリックします。



- ④ 画面左より、[\[Directory Server\]](#) > [\[ESDEMO\]](#) を選択します。

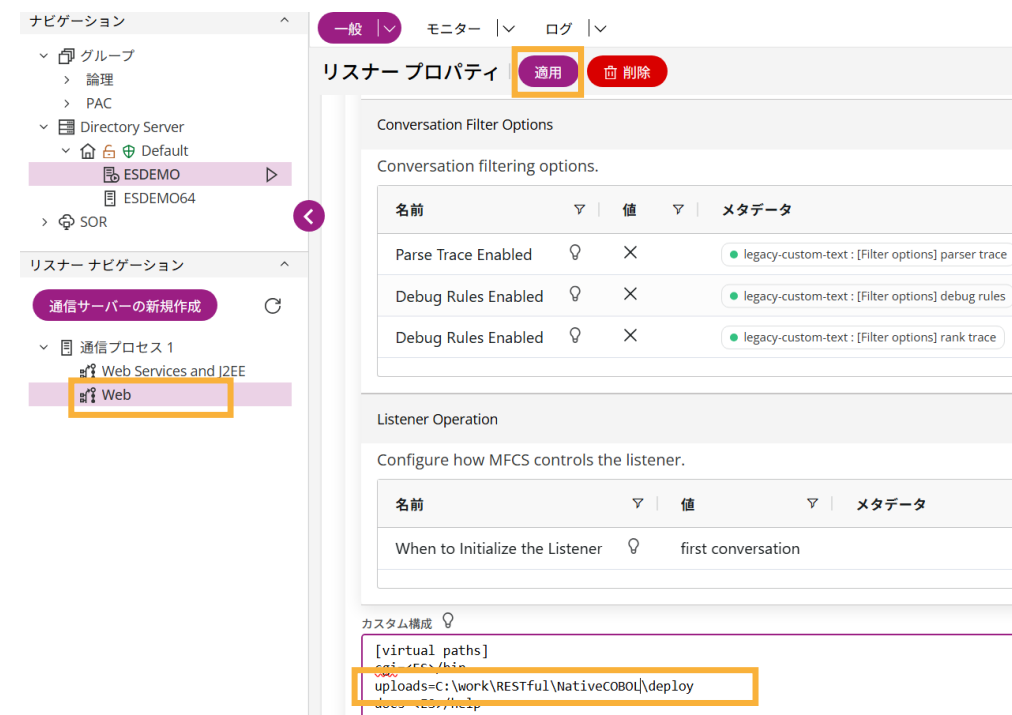


- ⑤ [\[一般\]](#) の横にある下向き記号をクリックし、ドロップダウンメニューから[\[リスナー\]](#) をクリックします。



- ⑥ 左側より [Web] を選択し、右側のスライダーを下にスクロールし、[高度] をクリックするとカスタム構成情報が表示されます。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。
- 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] をクリックします。

例 : uploads=c:/work/RESTful/NativeCOBOL/deploy



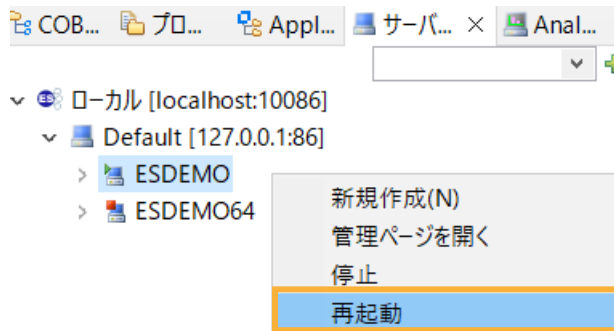
- ⑦ 正常に更新された旨のポップアップが表示されます。

Enterprise Server を自分で作成した場合、[ステータスの設定] を「Disable」から「Started」に変更してください。

2) 設定内容の反映

- ① Eclipse に戻り、[サーバーエクスプローラー] タブを選択します。
- ② インスタンスの再起動を行います。

[ESDEMO] を選択し、マウスの右クリックでコンテキストメニューを開き、[再起動] を選択します。



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 Eclipse の [COBOL エクスプローラー] タブに戻ります。
- 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.cbl」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.cbl」が選択されていることを確認し [終了(F)] をクリックします。

REST Web サービスの新規作成

このページで REST Web サービスを新規作成します



Web サービス名

マッピング: ☐ デフォルト ☒ 無し

マップするプログラム:

2) 書籍データ検索機能のオペレーションを作成

- ① さきほど作成した 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。

一般 HTTP ユーザ出口

オペレーションは選択されたエントリポイント インターフェイスを使用して COBOL プログラムを起動するのに使用されます

オペレーション名:

エントリポイント:

- 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] をクリックします。

一般 HTTP ユーザ出口

REST サービスのオペレーションの HTTP パスを更新/削除します

パス:

REST サービスのオペレーションの HTTP メソッドを選択します

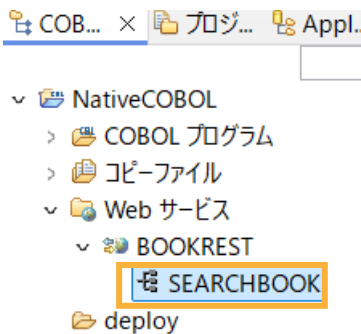
メソッド: ☐ GET ☒ POST ☐ PUT ☐ DELETE

OK キャンセル

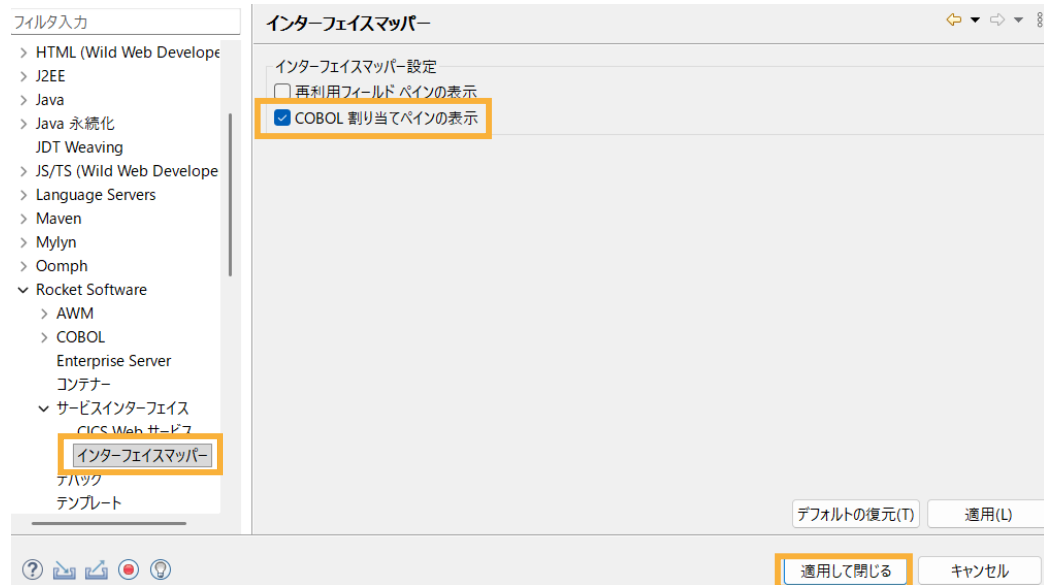
3) 書籍データ検索機能オペレーションのインターフェイスマッピングを定義

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。

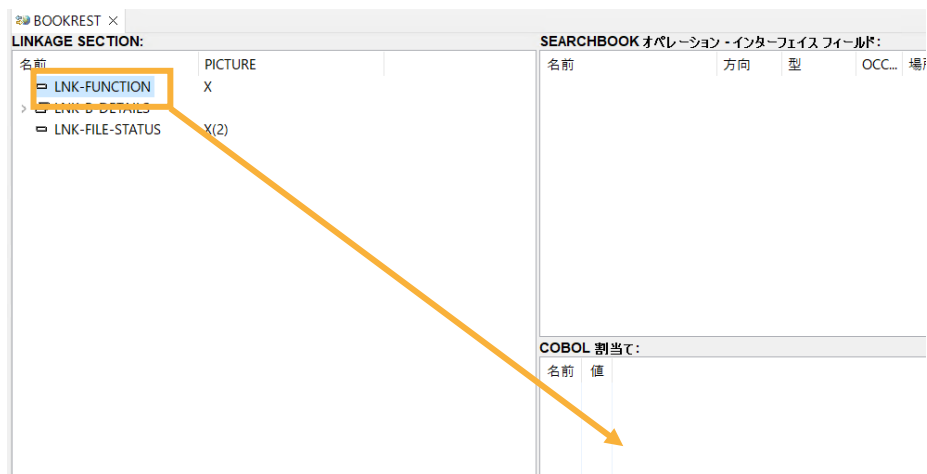


- ② Eclipse の [ウィンドウ] メニューから [設定] を選択し、[Rocket Software] > [サービスインターフェイス] > [インターフェイスマッパー] を選択します。[COBOL 割り当てペインの表示] にチェックを入れ [適用して閉じる] をクリックします。

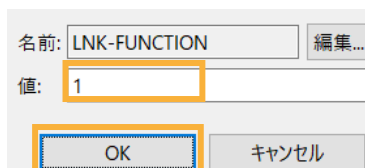


Preference Recorder のダイアログが表示されたら [キャンセル] をクリックしてください。

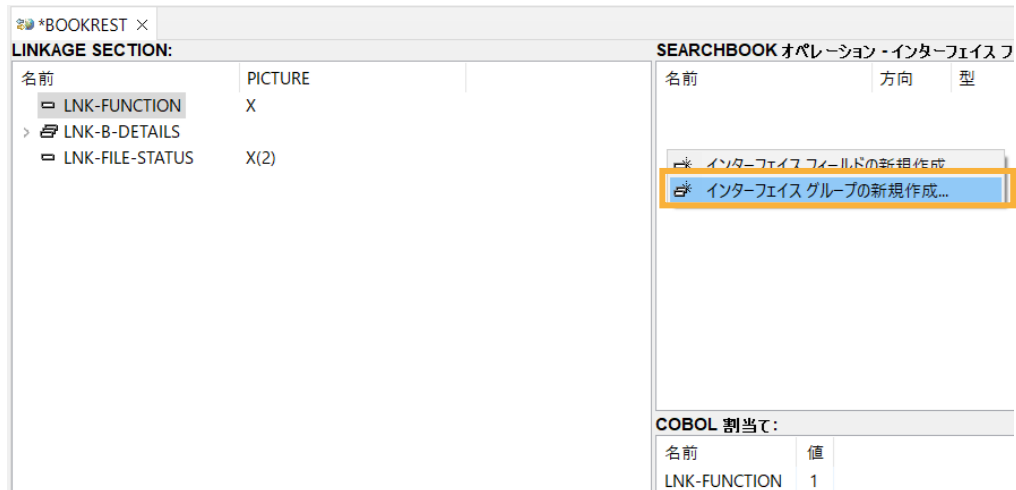
- ③ LINKAGE SECTION の COBOL の変数「LNK-FUNCTION」を [COBOL 割り当て] にドラッグ&ドロップします。



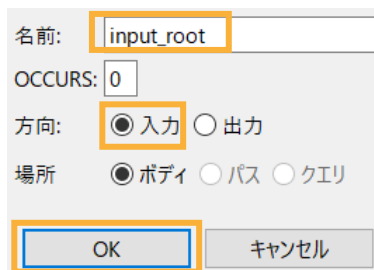
- ④ [COBOL 割当てプロパティ] ダイアログが表示されるので [値] に "1" を設定して [OK] をクリックします。



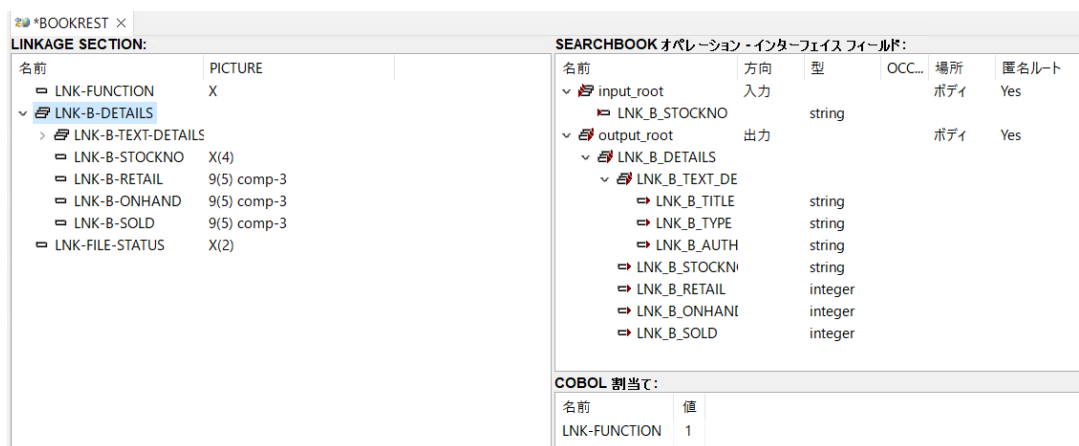
- ⑤ 「SERCHBOOK オペレーション - インターフェイス フールド」にて右クリックし、コンテキストメニューから [インターフェイス グループの新規作成] をクリックします。



- 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] をクリックします。



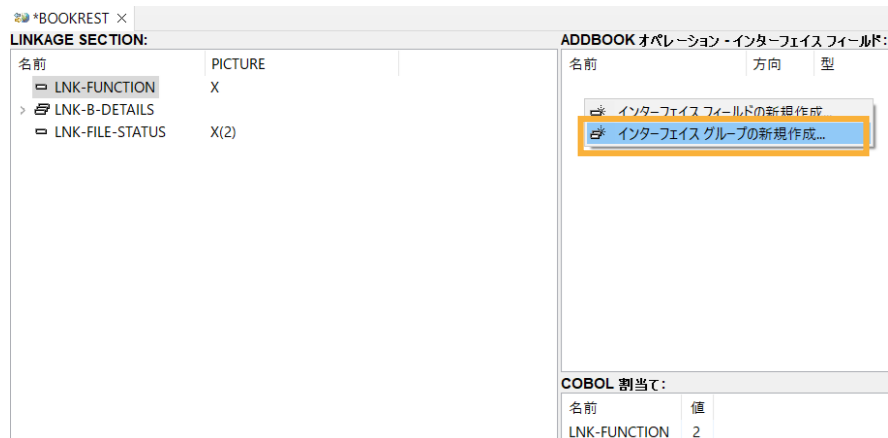
- ⑦ 「SERCHBOOK オペレーション - インターフェイス フールド」にて右クリックし、コンテキストメニューから「インターフェイスグループの新規作成」をクリックします。
- ⑧ [グループプロパティ] ウィンドウにて “output_root” と入力、[方向] を [出力] にして [OK] をクリックします。
- ⑨ [LINKAGE SECTION] 側にある「LNK_B_STOCKNO」を「input_root」にドラッグ&ドロップします。
- ⑩ [LINKAGE SECTION] 側にある「LNK-B-DETAILS」を「output_root」にドラッグ&ドロップします。
- 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っていることを確認して、[ファイル(F)] > [保存(S)] を選択します。



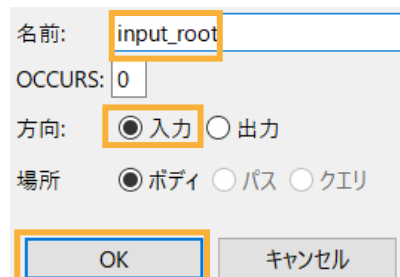
4) 書籍データ追加機能のオペレーションを追加

- ① 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。

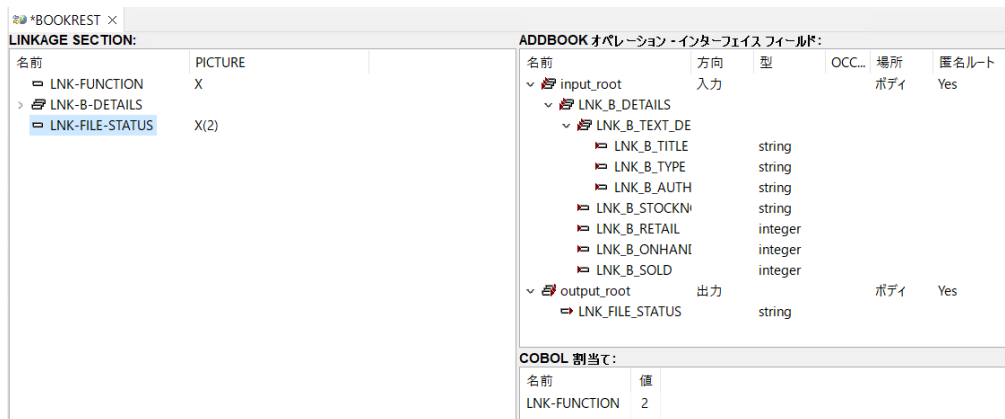
- ② オペレーションプロパティウィンドウが表示されます。[オペレーション名] 欄に “ADDBOOK” を入力します。
 - ③ 次に [HTTP] タブを選択します。HTTP メソッドが [POST] になっていることを確認し、[OK] をクリックします。
- 5) 書籍データ登録機能オペレーションのインターフェイスマッピングを定義
- ① 「ADDBOOK」 オペレーションをダブルクリックします。
 - ② 「LNK-FUNCTION」 を COBOL 割当てにドラッグ&ドロップし、[値] には “2” を指定します。
 - ③ 「ADDBOOK オペレーション - インターフェイス フォールド」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] をクリックします。



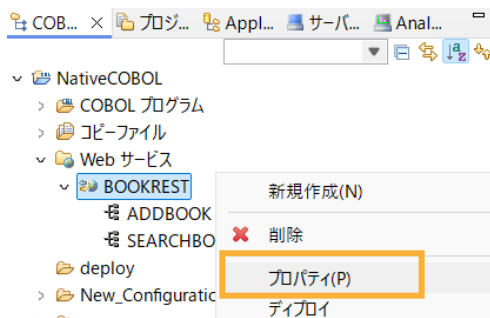
- ⑤ 「ADDBOOK オペレーション - インターフェイス フォールド」にて右クリックし、コンテキストメニューから [インターフェイスグループの新規作成] をクリックします。
- ⑥ [グループプロパティ] ウィンドウにて “output_root” と入力、[方向] を [出力] にして [OK] をクリックします。
- ⑦ [LINKAGE SECTION] 側にある「LNK-B-DETAILS」を「input_root」にドラッグ&ドロップします。
- ⑧ [LINKAGE SECTION] 側にある「LNK-FILE-STATUS」を「output_root」にドラッグ&ドロップし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



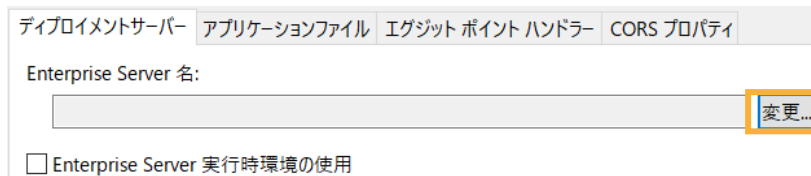
3.4 コンパイルした COBOL アプリケーションを Enterprise Server へデプロイ

1) Enterprise Server へのデプロイ情報を指定

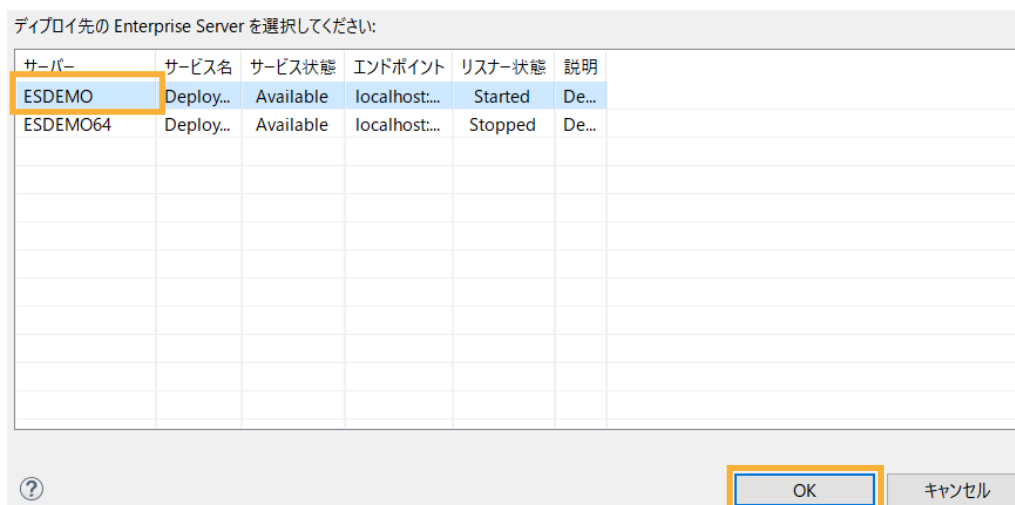
- ① COBOL エクスプローラーにて追加した Web サービス「BOOKREST」を右クリックし、コンテキストメニューから [プロパティ(P)] を選択します。



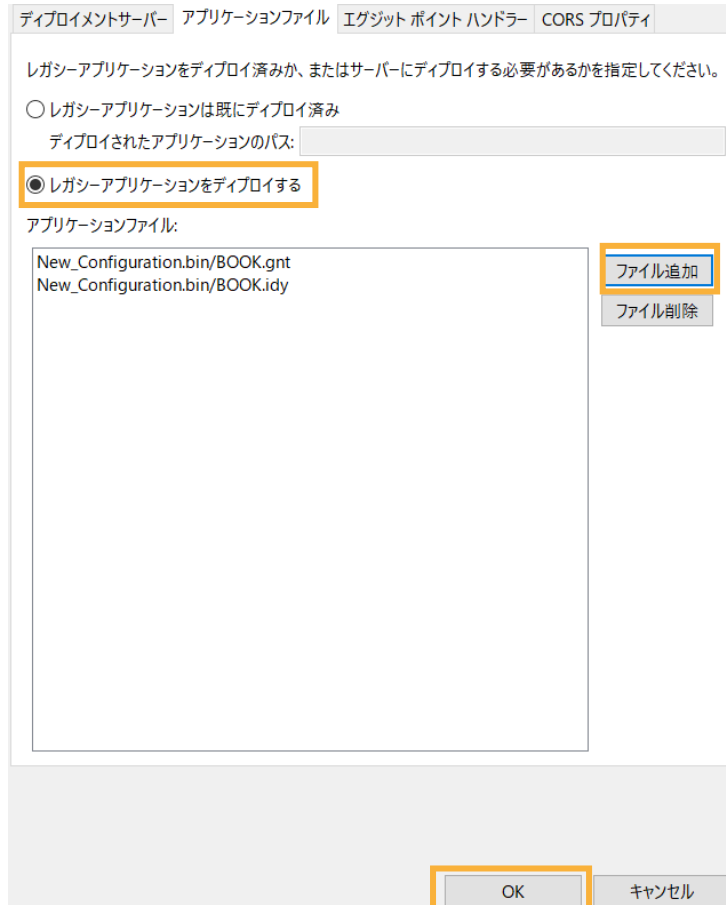
- ② 「ディプロイメントサーバー」タブを選択し、「変更」をクリックします。



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

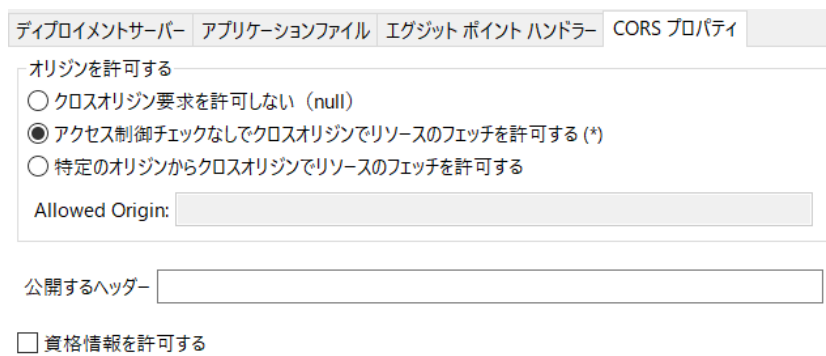


- ④ 次に [アプリケーションファイル] タブを選択し、以下の作業を実施したのち、[OK] をクリックします。
- [レガシーアプリケーションをデプロイする] を選択
 - [ファイル追加] をクリックして、プロジェクトディレクトリ配下の「New_Configuration.bin」に生成された「BOOK.gnt」および「BOOK.idy」を選択し、[開く(O)] をクリック



2) オリジン間リソース共有 (CORS) を許可

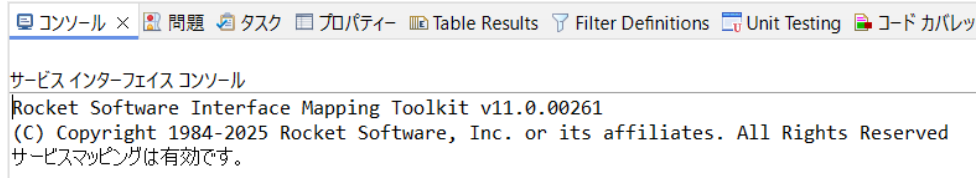
- ① デフォルトではオリジン間リソース共有は許可されていません。もしこれに関するエラーが発生する場合、許可設定を行います。
- ② 「BOOKREST」 Web サービスを選択して、右クリックしコンテキストメニューから[プロパティ]を選択します。
- ③ [CORS プロパティ]タブを選択し、運用用途に合わせた設定を行います。ここでは「アクセス制限チェックなしでクロスオリジンでリソースのフェッチを許可する(*)」を選択しています。



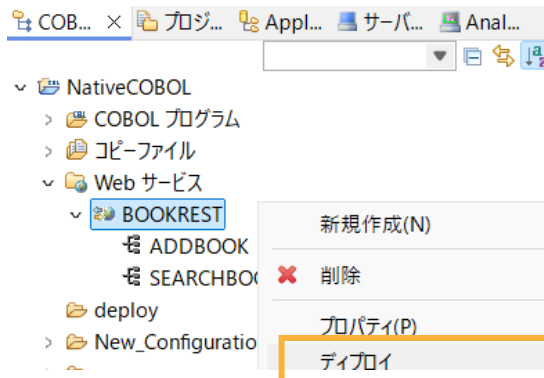
- ④ [OK] をクリックします。

3) RESTful Web サービスを Enterprise Server ヘッドプロイ

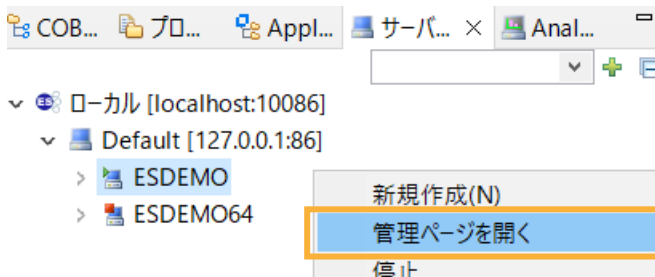
- ① Web サービス「BOOKREST」を右クリックし、コンテキストメニューから [検査] を選択し、サービスインターフェイス
コンソール上に
- ② エラーが出ていないことを確認します。



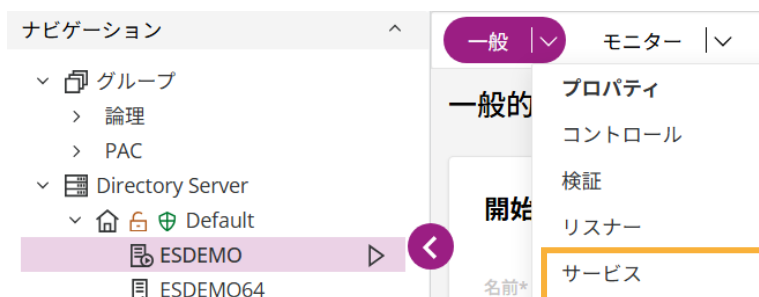
- ③ Web サービス「BOOKREST」を右クリックし、コンテキストメニューから [ディプロイ] を選択します。



- ④ [サーバーエクスプローラー] タブを選択し、[ESDEMO] を選択したうえで、マウスの右クリックでコンテキストメニューを開き、[管理ページを開く] を選択します。



- ⑤ [一般] > [サービス] をクリックします。



- ⑥ 画面を下にスクロールしていくと最下行にディプロイした RESTful Web サービスが表示されます。

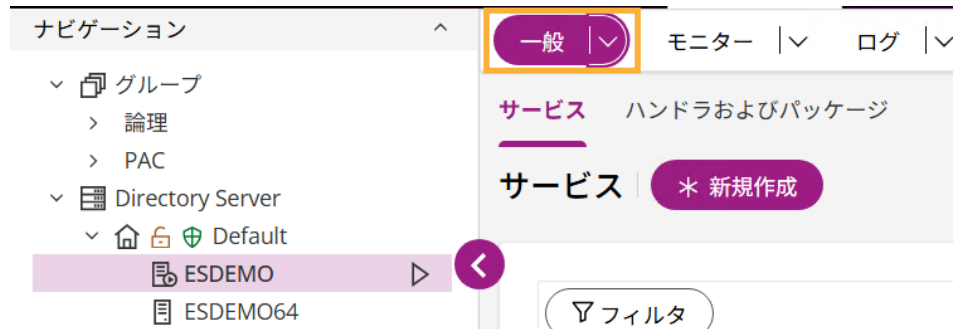
▼ /temppath/BOOKREST/1.0	
#SEARCHBOOK	Available
#ADDBOOK	Available

3.5 RESTful Web サービスのテスト

3.5.1 サービス実行時にアプリケーションがアクセスするデータファイルの設定

サンプルアプリケーションは、環境変数で書籍情報を管理するデータファイルを指定しています。テスト実行前にこの設定を行います。

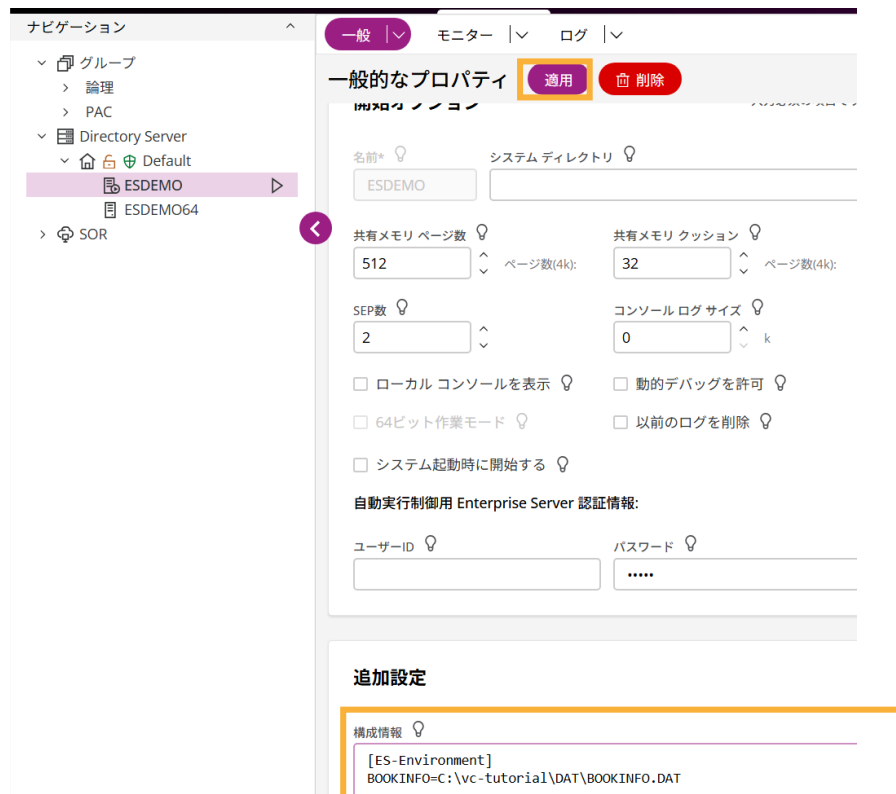
- 1) [一般] をクリックします。



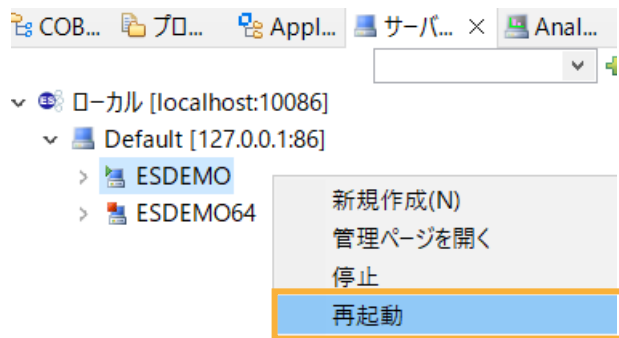
- 2) [追加設定] > [構成情報] に以下を入力したうえで、[適用] をクリックします。

[ES-Environment]

BOOKINFO=C:¥vc-tutorial¥DAT¥BOOKINFO.DAT



- 3) Eclipse に戻り、[サーバーエクスプローラー] タブを選択します。
- 4) [ESDEMO] を選択し、マウスの右クリックでコンテキストメニューを開き、[再起動] を選択します。

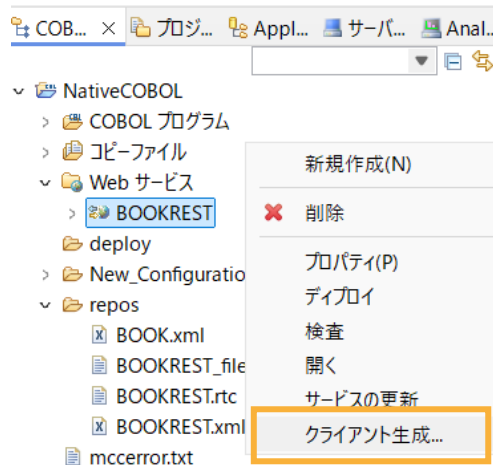


3.5.2 サービスのテスト

3.5.2.1 テスト用 COBOL アプリケーションの利用

登録処理

- 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] をクリックします。

☐ プロジェクトなし
☒ ナビゲータ上の現在のプロジェクト
☐ 開いている Eclipse プロジェクトを選択
☐ COBOL プロジェクトを新規作成

生成されたクライアント用のプロジェクト: NativeCOBOL 選択...

出力ディレクトリ: C:\work\RESTful\NativeCOBOL 参照...

☐ JSON/YAML を使用
 JSON/YAML ファイル: 参照...

オペレーション
☐ すべてのオペレーション

サイズ
 文字列サイズ: 100
 インライン配列サイズの制限: 個別の構造体なし (0)
 配列サイズ: 100

文字集合: ☒ ASCII ☐ EBCDIC
 ビットモード: ☒ 32 ☐ 64

OK キャンセル

3) 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(W)] 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



フィルタ入力

- Apache Tomcat
- AspectJ/Java Application
- AspectJ Load-Time Weaving A
- COBOL/Java 相互運用機能のア
- COBOL JVM アプリケーション
- COBOL JVM ユニットテスト
- COBOL JVM リモート アプリケーシ
- COBOL アプリケーション**
- COBOL ユニットテスト
- Debug Adapter Launch
- Eclipse アプリケーション

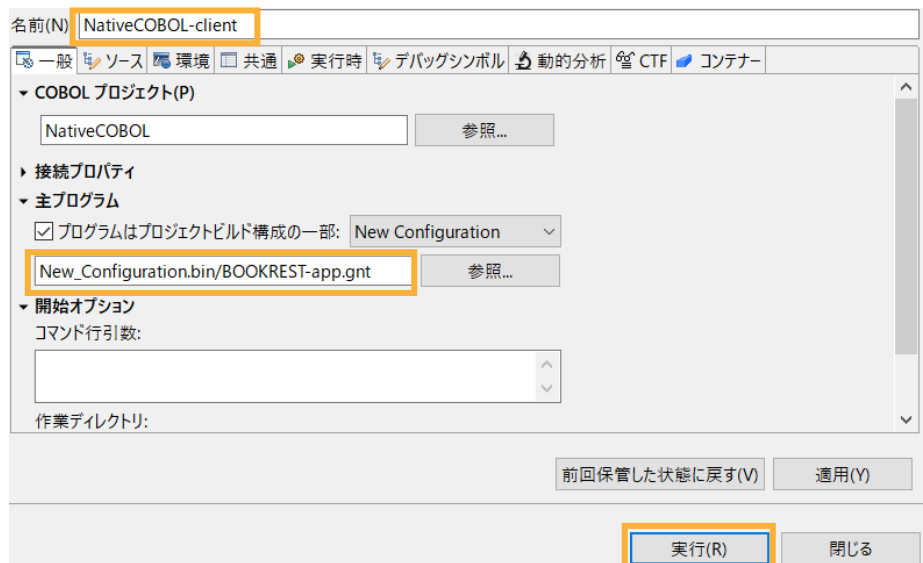
このダイアログから起動設定を構成します:

- 選択した種類の構成を作成するには、「新規構成」ボタンを押します。
- 選択した種類の起動構成プロトタイプ...「新規プロトタイプ」ボタンを押します。
- 選択した構成をエクスポートするには、「エクスポート」ボタンを押します。
- 選択した構成をコピーするには、「複製」ボタンを押します。
- 選択した構成を削除するには、「削除」ボタンを押します。
- フィルタ・オプションを構成するには、「フィルタ」ボタンを押します。
- プロトタイプをリセットするには、起動構...リンク」メニュー項目を選択してください。
- 削除するには、...を解除」メニュー項目を選択してください。
- ...するには、起...セット」メニュー項目を選択してください。

- ③ [一般] タブの 以下の入力を行い、[実行(R)] をクリックします。

名前: "NativeCOBOL-client"

主プログラム: "New_Configuration.bin/BOOKREST-app.gnt"

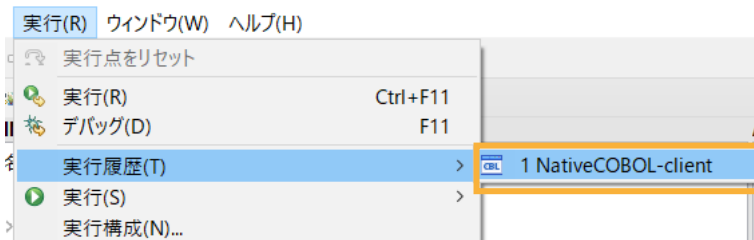


- ④ 「Service Address」は、今回変更不要のため、Enter を押します。
Web サービスの要求を受け付けるリスナーがポート 9003 でリスンしているため、変更せず、クライアントの実行が行えます。このポートは、[リスナー] メニューをクリックした後の画面にて任意に変更できます。変更した場合は、クライアント実行時に変更後のポートを指定する必要があります。
- ⑤ Supplemental Query String (Optional) は変更しないのでそのまま Enter を押します。
- ⑥ 「Username」、「Password」は変更しないのでそのまま 2 回 Enter 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):
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 . . .
```

検索処理

- 1) Eclipse メニューより [実行(R)] > [実行履歴(T)] > [1 NativeCOBOL-client] を選択します。



- 2) 以下の入力を行います。

Service Address : そのまま Enter を入力

Supplemental Query String : そのまま Enter を入力

User: そのまま Enter を入力

Password : そのまま Enter を入力

Operation : “1” を入力

LNK_B_STOCKNO : “5555” を入力

先ほど登録した情報が戻されます。

```
Service Address (Enter = http://localhost:9003):
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 . . .
```

3.5.2.2 curl コマンドの利用

作成した RESTful Web サービスは、一般的な仕様であるため、テスト用の COBOL アプリケーションからだけではなく、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ーに移動します。ここまでの手順では、C:\%vc-tutorial 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
```

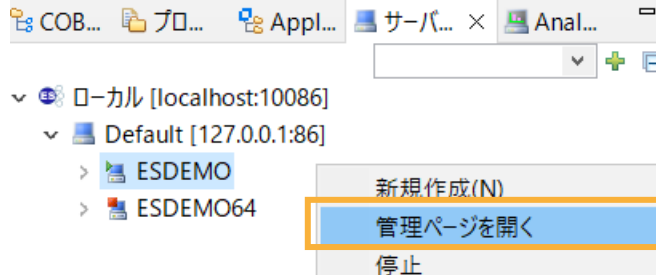
コマンドは 1 行で入力してください。

```
コマンド プロンプト
C:\vc-tutorial>curl http://localhost:9003/temppath/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
  }
}
C:\vc-tutorial>_
```

3.6 RESTful Web サービスのデバッグ

- 1) Eclipse に戻り、[サーバーエクスプローラー] タブを選択します。
- 2) デバッグ設定

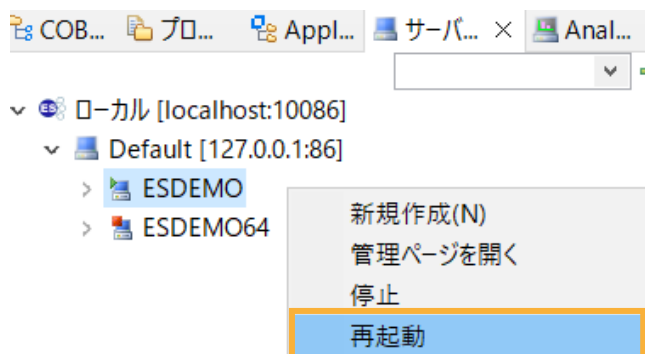
- ① [ESDEMO] を選択したうえで、マウスの右クリックでコンテキストメニューを開き、[管理ページを開く] を選択します。



- ② [一般] をクリックしたうえで、[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。



- ③ Eclipse に戻り、[サーバーエクスプローラー] タブを選択します。
- ④ [ESDEMO] を選択し、マウスの右クリックでコンテキストメニューを開き、[再起動] を選択します。



- 3) Eclipse IDE からのデバッグアタッチ

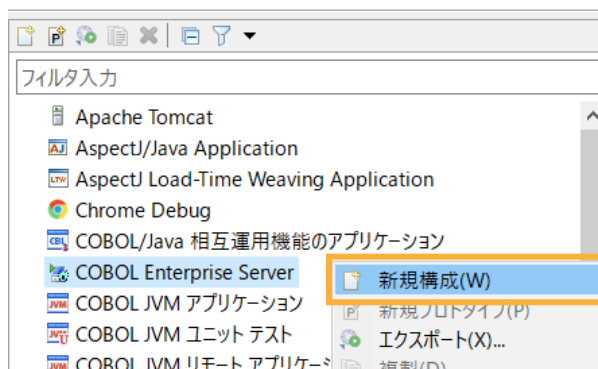
- ① Eclipse に戻り、[COBOL エクスプローラー] タブを選択したうえで、「NativeCOBOL」プロジェクトを選択し、マウスの右クリックにてコンテキストメニューを開いたうえで、[デバッグ(D)] > [デバッグの構成(B)] を選択します。



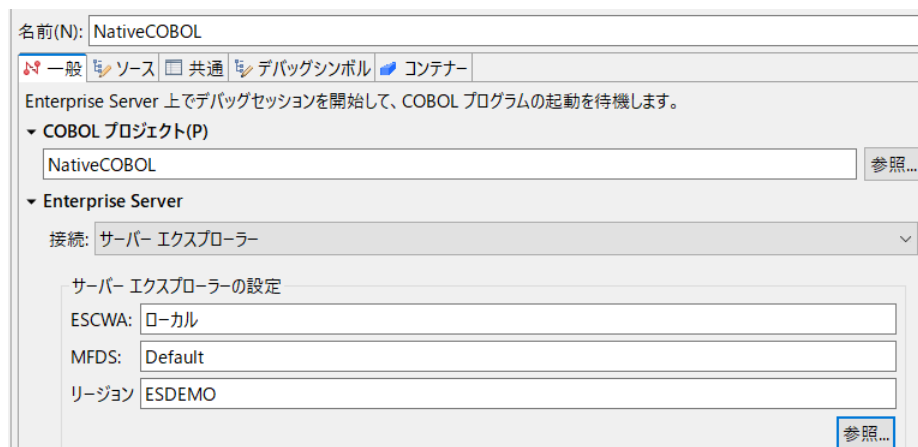
- ② 画面左側より、[COBOL Enterprise Server] を選択し、マウスの右クリックにてコンテキストメニューを開いたうえで、[新規構成]を選択します。

構成の作成、管理、および実行

Enterprise Server アプリケーションへの接続とデバッグ



- ③ 名前に “NativeCOBOL” を入力し、「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。

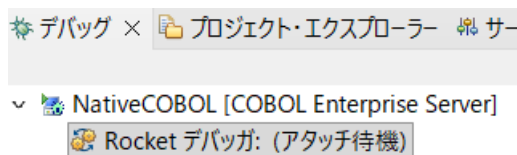


- ④ 画面を下にスクロールし、「デバッグの種類」セクションを表示します。
タイプに [Web サービス] が選択されていること、[Web サービス名] が空欄であることを確認したうえで、[デバッグ (D)] をクリックします。



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。

アタッチ待機状態になります。



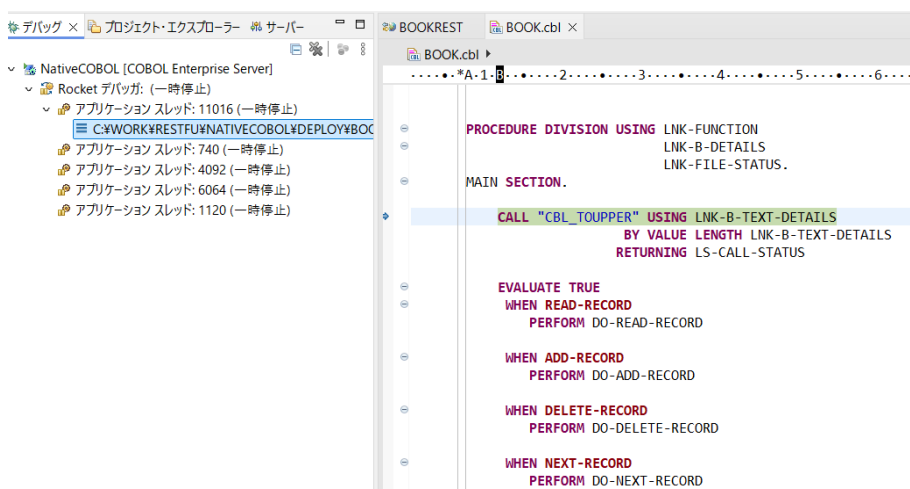
- ⑤ 3.5.2.2 で実施した curl コマンドを用いた検索処理のリクエストを行ってください。

デバッグ停止するため、さきほどと異なり、結果がすぐ戻ってきません。

補足)

localhost がデバッグ対象にならないことがあります。この場合は、localhost を 127.0.0.1 に変更してください。

- ⑥ Eclipse IDE に戻ると、デバッグ停止中になっています。

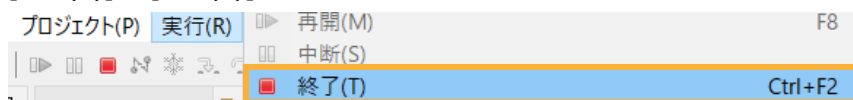


コンソールアプリケーションのデバッグと同様、[実行] メニュー配下の [再開] や [ステップイン] の操作が行えます。

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。

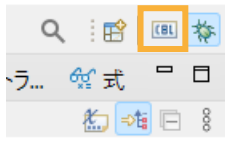
再開、もしくは、プログラム終了まで進めると、デバッグ中断が終了し、curl コマンドの結果が前回同様に戻されます。

- ⑦ [実行(R)] > [終了(T)] を選択して、デバッグを終了します。

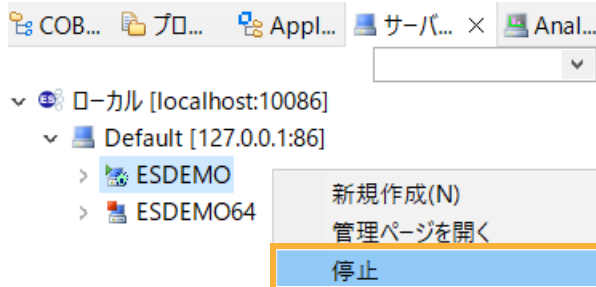


3.7 インスタンスの停止

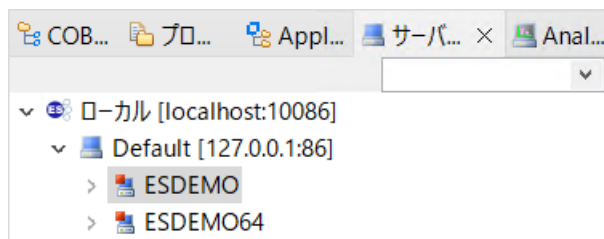
- 1) 画面右上のアイコンをクリックして、「COBOL パースペクティブ」を表示します。



- 2) 「サーバーエクスプローラー」タブに切り替えます。
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。



停止すると、赤色のアイコンが表示されます。



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。