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 リモート開発編

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 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

本チュートリアルでは、Linux サーバーとのリモート開発を行います。リモート開発については、「Visual COBOL チュートリアル COBOL 開発： Linux/UNIX 版 リモート開発編」を参照してください。

また、本資料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(Windows) ソフトウェア

OS	Windows Server 2019 Standard Edition（64bit）
COBOL 開発環境製品	Visual COBOL 9.0 for Eclipse + PU01

- 開発クライアント(Linux) ソフトウェア

OS	Red Hat Enterprise Linux 8.4
COBOL 開発環境製品	Visual COBOL Development Hub 9.0 + PU01

- 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

3. チュートリアル手順の概要

3.1. Linux リモートサーバーを起動

1) 管理者権限をもったユーザーに切り替え、ユーザーロケールを SJIS に変更します。

2) Visual COBOL の利用に必要な環境変数を整えます。

```
#. /opt/microfocus/VisualCOBOL/bin/cobsetenv
COBDIR set to /opt/microfocus/VisualCOBOL
```

3) デーモンを起動します。

```
$COBDIR/remotedev/startrdodaemon
Starting RSE daemon...
Reading "/opt/microfocus/VisualCOBOL/remotedev/rdo.cfg" ...
Daemon port loaded from "/opt/microfocus/VisualCOBOL/remotedev/rdo.cfg": 4075
Server port range loaded from "/opt/microfocus/VisualCOBOL/remotedev/rdo.cfg": 10000-10003
[root@myhost tmp]# Daemon running on: myhost.localdomain, port: 4075
```

3.2. ディレクトリサーバーの起動

1) ディレクトリサーバーを起動します。

```
# mfds&
[2] 10800
```

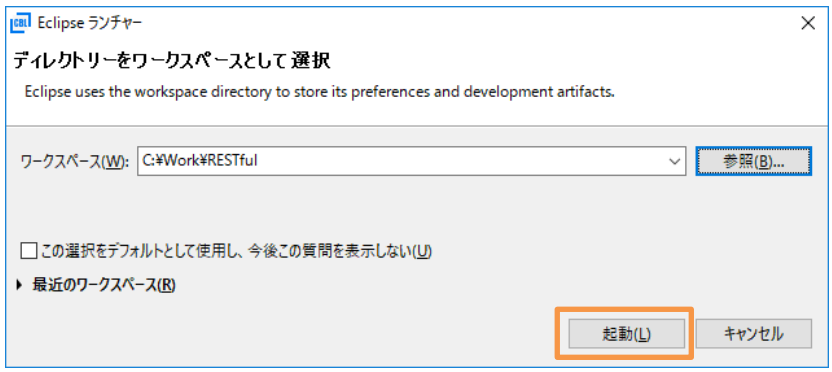
補足）
１つの管理画面上で、異なるサーバー上で稼働するディレクトリサーバーを管理することができます。
本チュートリアルでは、localhost 上（Windows 上）の管理画面を用いてリモートサーバー上で稼働するディレクトリサーバーを管理するため、この補足に記述されたコマンドを実行する必要はありませんが、別途、リモートサーバー上で管理画面を起動し、そちらを利用することもできます。
その場合は、リモートサーバーの管理画面機能を有効にするため、escwa コマンドの実行が必要です。
escwa &
localhost 以外からのアクセスの許可を行う場合は、以下のようなオプションをつけて実行します。
escwa --BasicConfig.MfRequestedEndpoint=tcp:*:10086 &
起動した場合は、作業終了後に以下のコマンドで終了してください。
escwa -p

3.3. 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 > [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:¥Work¥RESTful” を指定し、[起動] ボタンをクリックします。

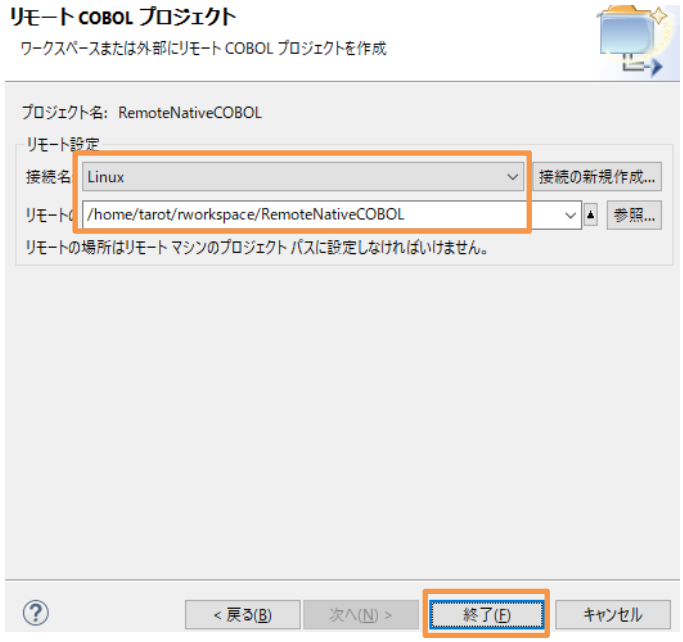


2) ネイティブ COBOL プロジェクトの作成とプログラムソースのインポート

① [ファイル(F)]メニュー > [新規(N)] > [リモート COBOL プロジェクト] を選択し、プロジェクト名に “RemoteNativeCOBOL” を入力、ファイル システムに “リモートファイル・システム(RSE)”を選択して、[次へ(N)] ボタンをクリックします。

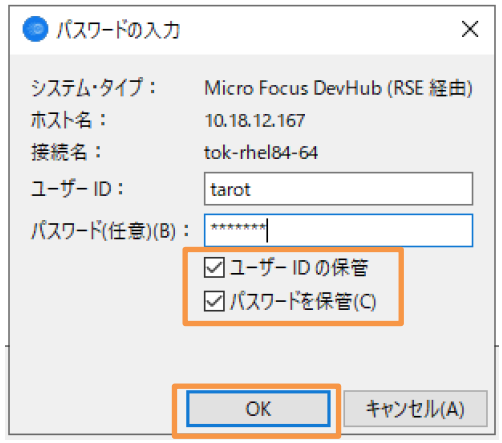
② プロジェクトテンプレートは “64 ビット” を選択して、[次へ(N)] をクリックします。

③ リモートサーバーの Linux への接続設定を選択し、プロジェクトの保存ディレクトリを入力したうえで、[終了(F)] をクリックします。

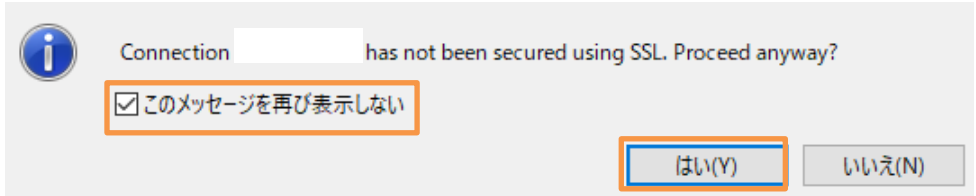


注意）
リモート接続設定が未作成の場合は、前提条件に記載したチュートリアル「Visual COBOL チュートリアル COBOL 開発： Linux/UNIX 版 リモート開発編」を参照して作成してください。

パスワードの入力画面が表示されましたら、パスワードを入力し、「ユーザー ID の保管」、「パスワードを保管」にチェックしたうえで、[OK] をクリックします。

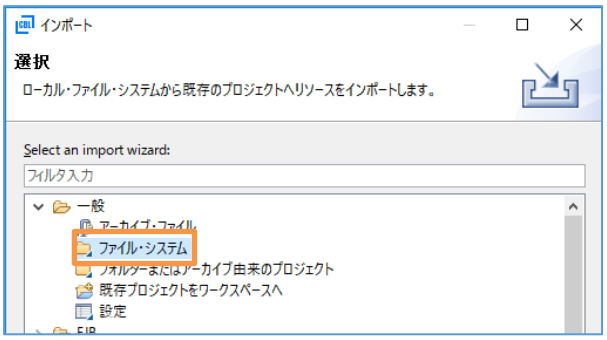


このようなダイアログが表示された場合は、「このメッセージを再び表示しない」にチェックを入れて、[はい(Y)] をクリックします。

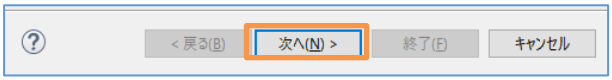


④ COBOL エクスプローラーのパースペクティブを開き、COBOL エクスプローラービューにて プロジェクトフォルダを右クリックし、コンテクストメニューから［インポート(I)］>［インポート(I)］を選択します。

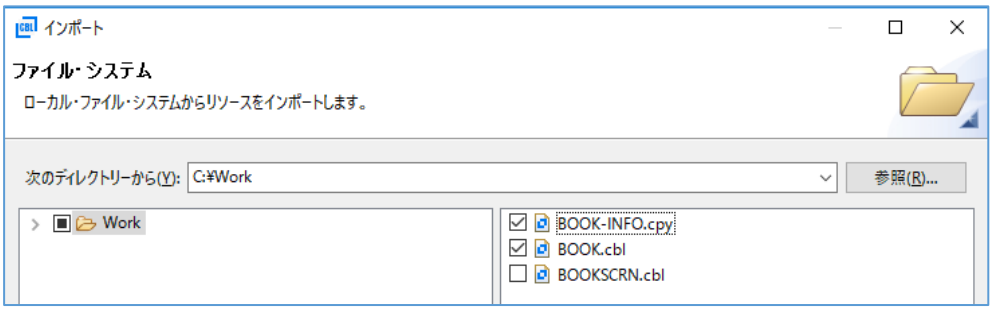
⑤ 既存のソースコードをロードします。一般のフォルダを展開し、［ファイル・システム］を選択し、［次へ(N)］ボタンをクリックします。



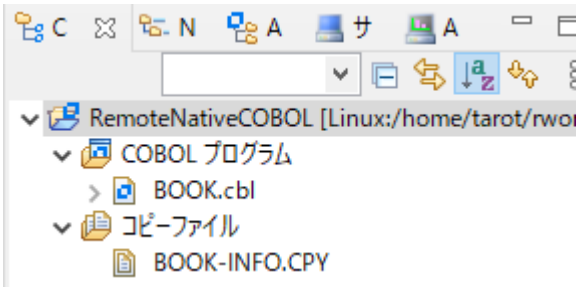
途中省略

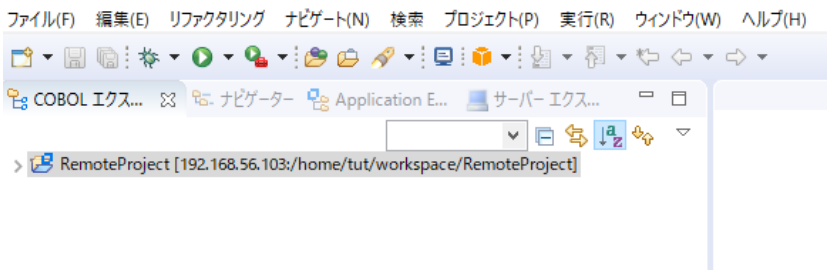


⑥ チュートリアル用のファイルをインポートします。インポートダイアログが表示されるので［参照(R)］ボタンをクリックし、任意のフォルダにダウンロードしたファイルを展開し、そのフォルダを指定します。下図では "C:¥work" に展開したファイルを指定しています。ここで "BOOK-INFO.cpy" と "BOOK.cbl" を指定し、［終了(F)］ボタンをクリックします。



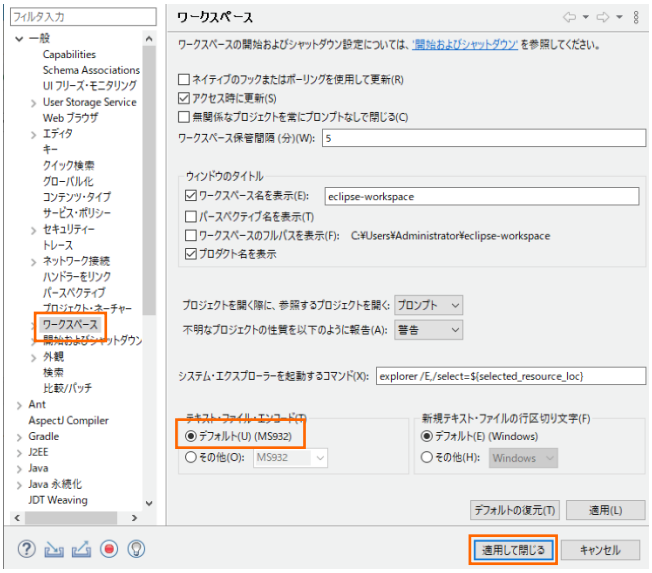
⑦ プロジェクトフォルダを展開し、２つのファイルが正常にロードされていることを確認します。



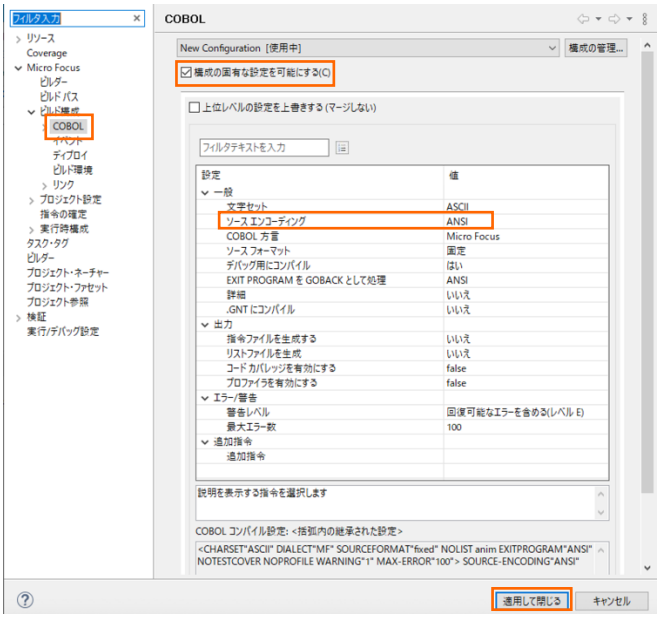


3) 文字コードの指定を行います。

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(W)]メニュー ＞ [設定(P)]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。



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ク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」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。



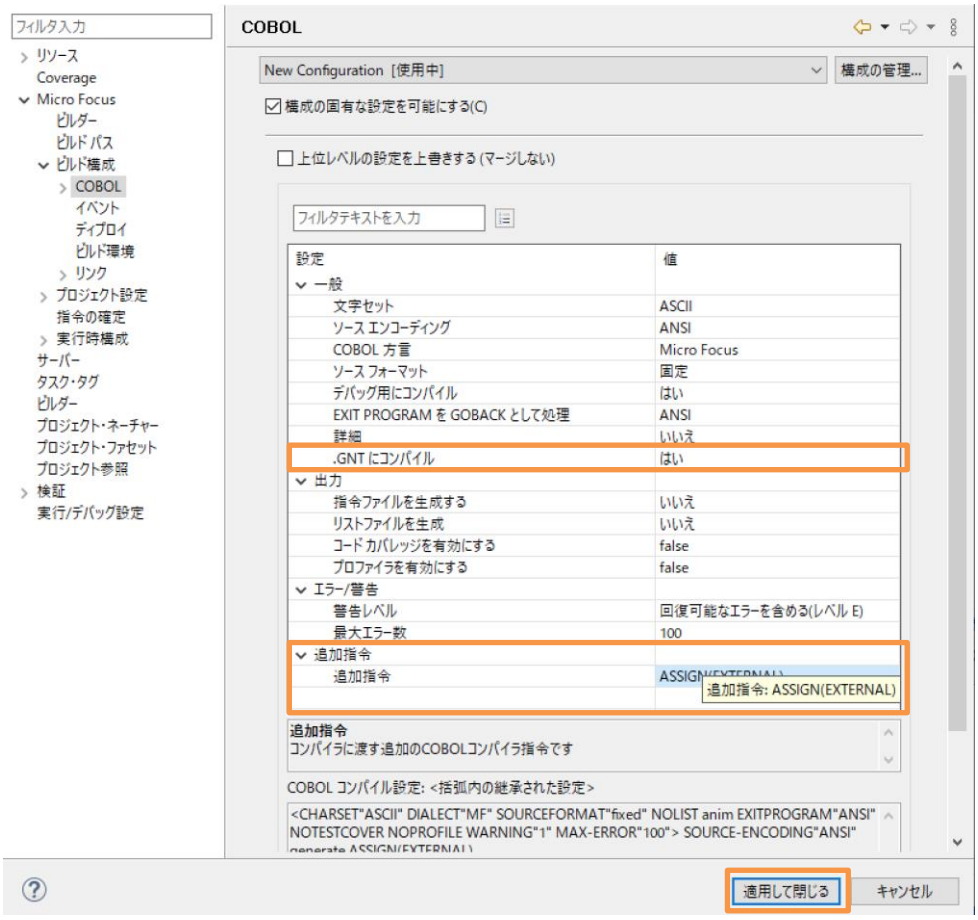
4) ビルドオプションの変更

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「RemoteNativeCOBOL」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［プロパティ(R)］を選択します。

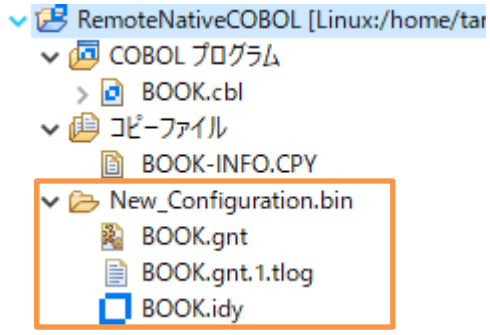
② プロパティ設定ダイアログが表示されます。[Micro Focus] ＞ [ビルド構成] ＞ [リンク] をクリックし、[ターゲットの種類] を

「すべて INT/GNT ファイル」に変更し、[適用(L)] をクリックします。

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。

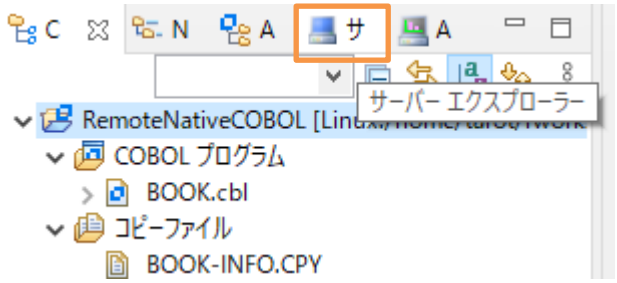


④ COBOL エクスプローラーにて ”New_Configuration.bin” を展開して下記のファイルが作成されていることを確認します。

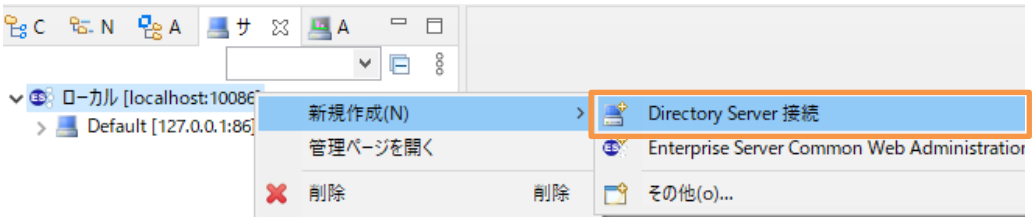


5) リモートサーバーの Enterprise Server の追加と起動

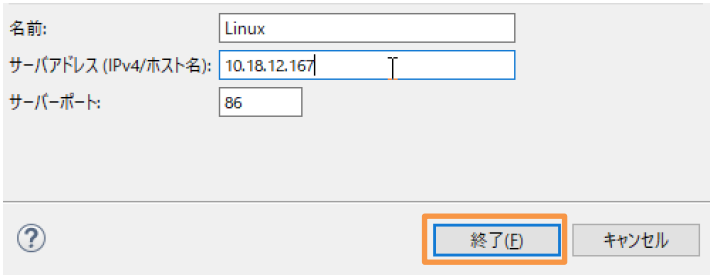
① [サーバーエクスプローラー] タブを選択します。



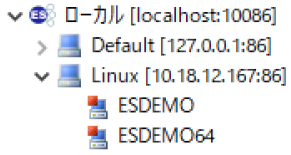
② 「ローカル [localhost:10086]」上で右クリックし、コンテクストメニューから [新規作成(N)] > [Directory Server 接続] をクリックします。



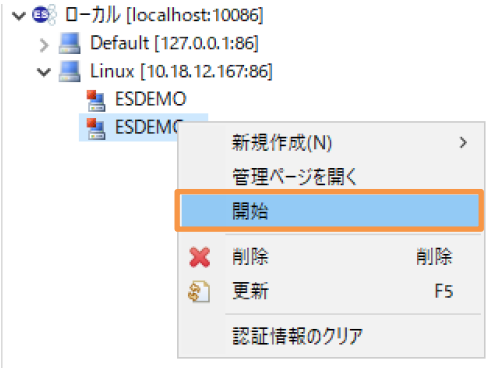
③ リモートサーバーの情報を入力し、[終了(F)] をクリックします。



追加後、サーバーエクスプローラーのリモートサーバー名（以降では、Linux とします）を展開すると リモートサーバーにインストールされた Visual COBOL Development Hub 製品にビルドインされている Enterprise Server インスタンス ESDEMO, ESDEMO64 が表示されます。



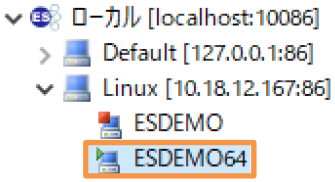
④ リモートサーバー配下の「ESDEMO64」を右クリックし、コンテクストメニューから［開始］を選択します。もし、ダイアログが表示されたらそのまま[OK]ボタンを押してください。



⑤ 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。

```
サーバー: ESDEMO64 正常に起動されました
```

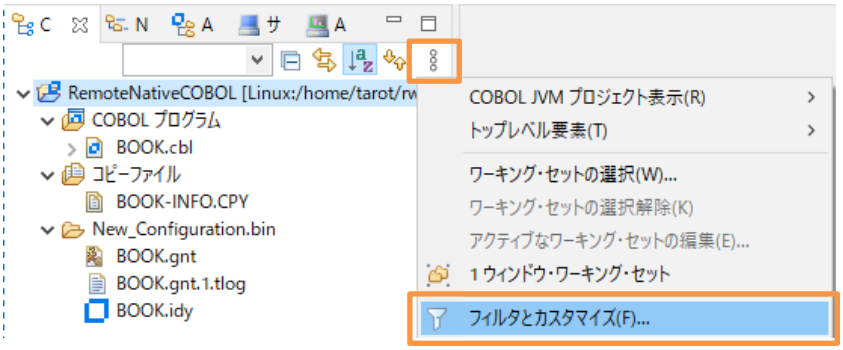
⑥ 正常に開始されると［サーバーエクスプローラー］ビュー上の「ESDEMO64」アイコンが起動されたことを示す緑色のアイコンに切り替わります。



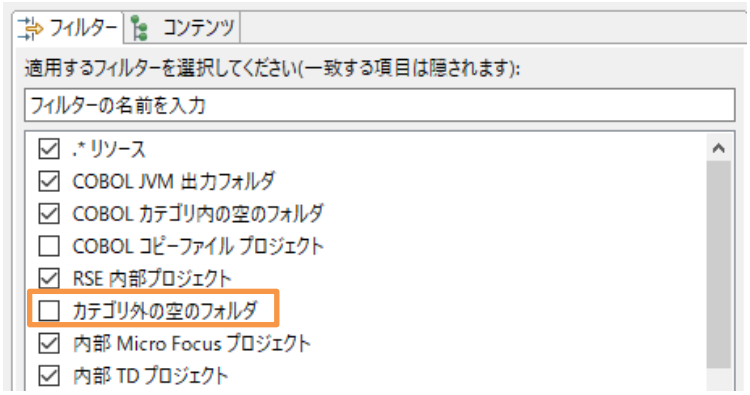
6) COBOL エクスプローラー表示設定の変更

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(F)] を選択します。



③ [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。



7) ディプロイ用フォルダの作成

① 「RemoteNativeCOBOL」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 [新規作成(N)] > [フォルダー] を選択します。

② 「RemoteNativeCOBOL」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。

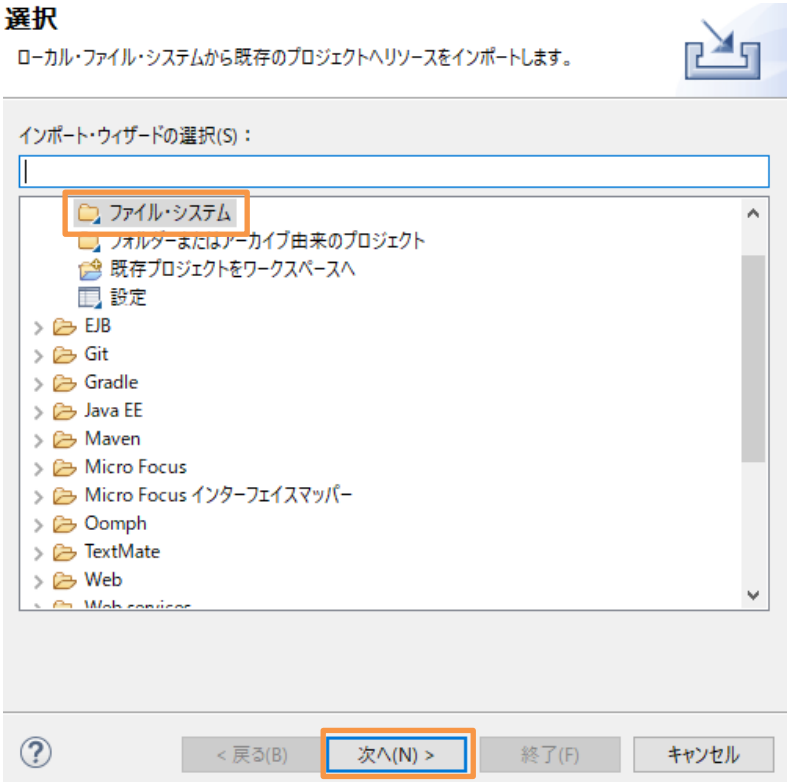
8) データファイル保存フォルダの作成

① 7) と同様の手順で “DAT” フォルダを作成してください。

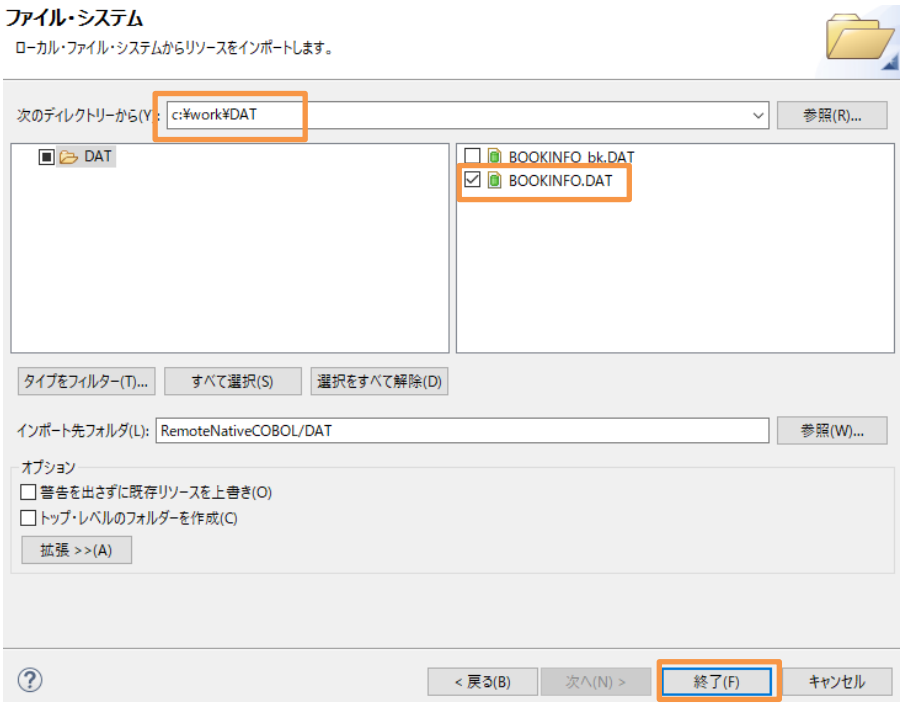
9) データファイルのインポート

① 作成した「DAT」フォルダ上で右クリックし、コンテクストメニューから［インポート(I)］>［インポート(I)］を選択します。

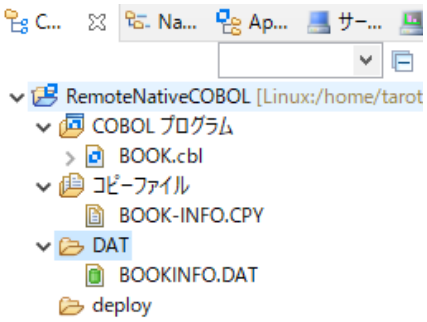
② 「一般」配下の「ファイル・システム」を選択し、［次へ(N)］ボタンをクリックします。



③ チュートリアル用ファイルを展開したフォルダ配下（ここでは、C:¥Work）配下の DAT を指定して、BOOKINFO.DAT にチェックを行い、［終了(F)］をクリックします。



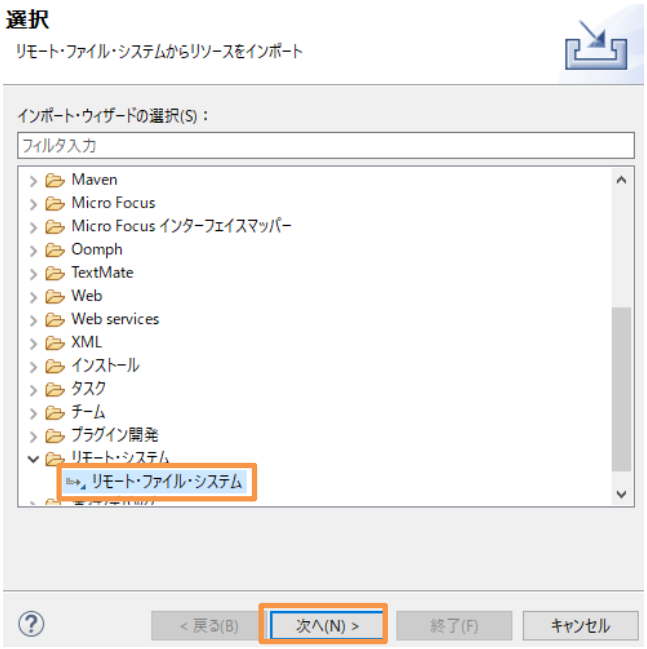
BOOKINFO.DAT が DAT フォルダ配下に表示されます。



10)「.mfdeploy」ファイルのインポート

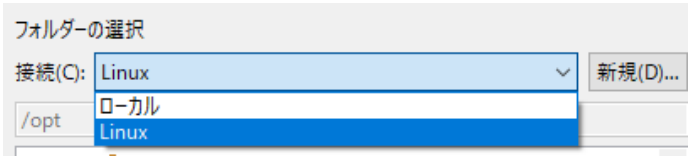
① 作成した「deploy」フォルダ上で右クリックし、コンテクストメニューから［インポート(I)]→[インポート(I)］を選択します。

② 「リモート・システム」配下の「リモート・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。

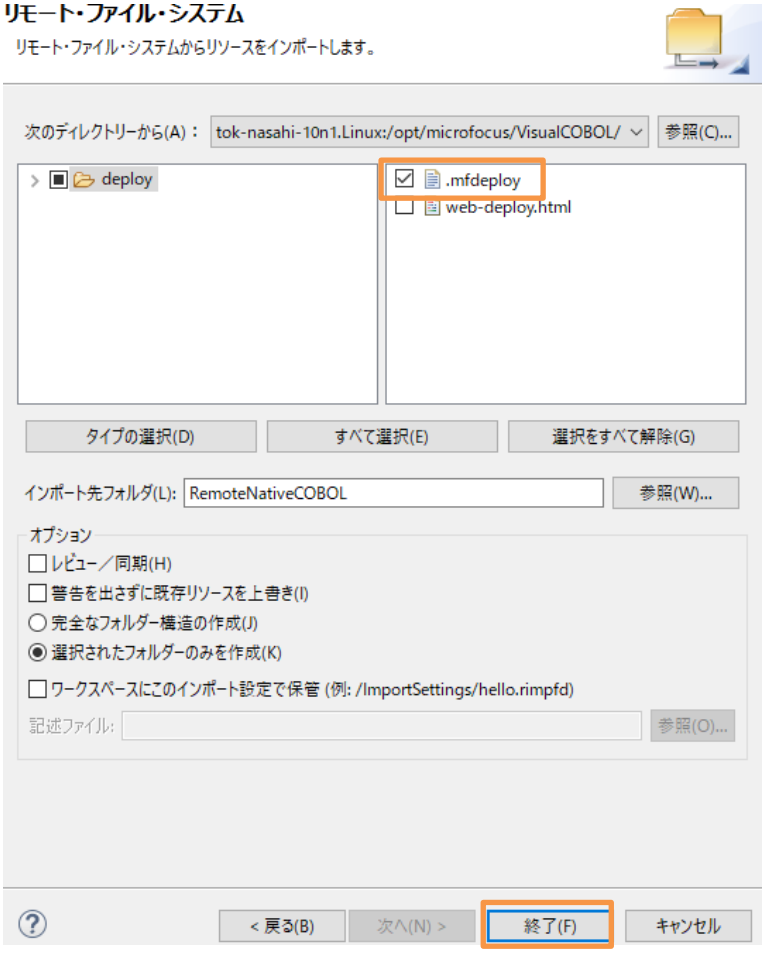


③ [参照(C)] ボタンをクリックし、接続を Linux に変更したうえで、Visual COBOL インストールフォルダ/deploy を選択します。

デフォルトの Visual COBOL インストールフォルダは /opt/microfocus/VisualCOBOL になります。



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。



補足）
上記手順完了後も、COBOL エクスプローラー上の RemoteNativeCOBOL/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。

3.4. 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 エクスプローラーを表示します。

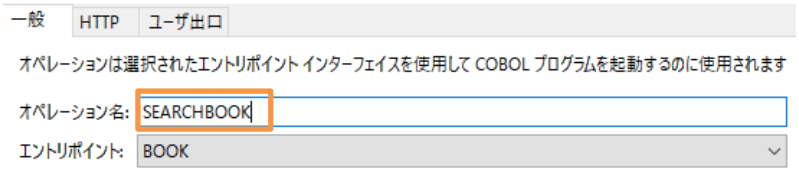
② RESTful Web サービスとして利用するビジネスロジックを処理するプログラム「BOOK.cbl」を右クリックし、コンテクストメニューから［新規作成(N)］>［REST Web サービス］を選択します。

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)］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、コンテクストメニューから［新規作成(N)］>［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。

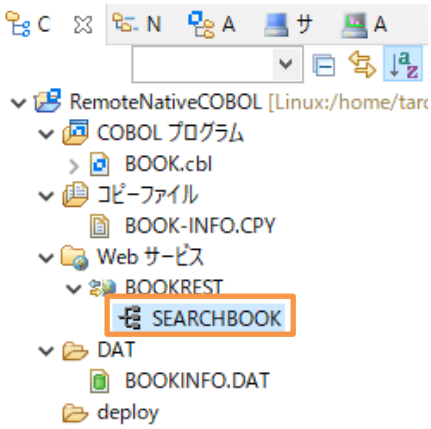


③ 次に［HTTP］タブを選択します。HTTP メソッドが［POST］になっていることを確認し、［OK］ボタンをクリックします。

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

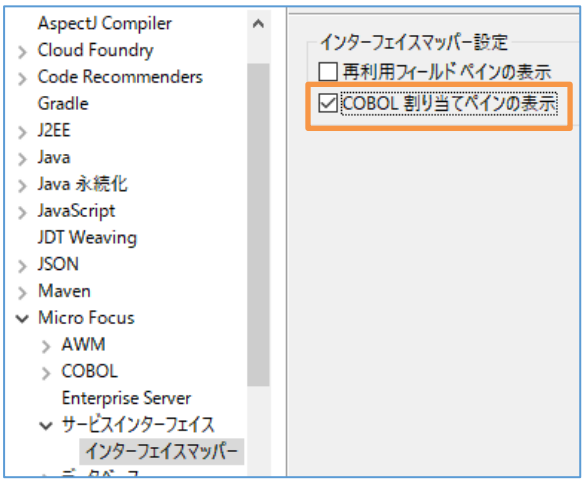
Enterprise Server は Web サービス側のデータ型と COBOL のデータ型を相互に自動変換させる機能を装備しています。この機能により Web サービスコンシューマー側と COBOL 側はそれぞれ相手のデータ型を意識することなく透過的にデータ変換が処理されやりとりができます。

① 「SEARCHBOOK」オペレーションをダブルクリックします。

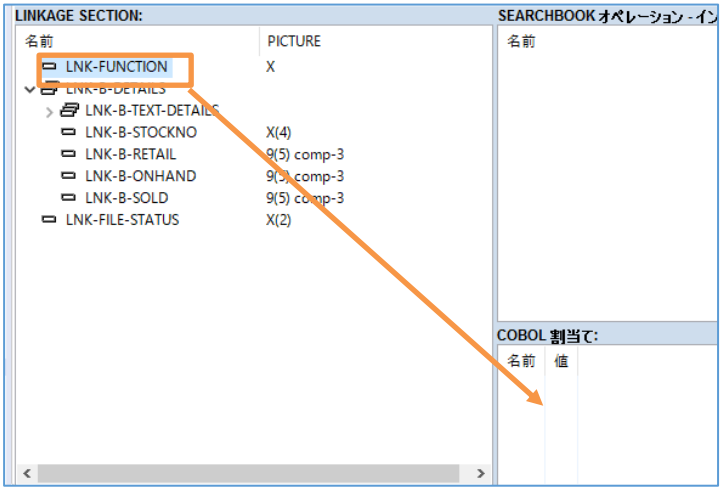


② Eclipse IDE メニューから［ウィンドウ(W)］-> [設定(P)]を選択し、[Micro Focus] > [サービスインターフェース] > [インターフェイスマッパー］を選択します。[COBOL 割り当てペインの表示］にチェックを入れ［適用して閉じる］ボタンをクリックします。

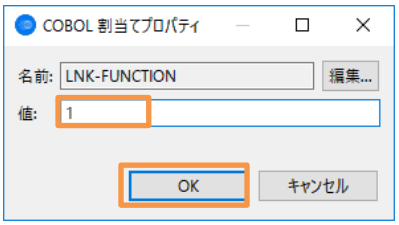
※Preference Recorder のダイアログが表示されたら［キャンセル］を選択してください。



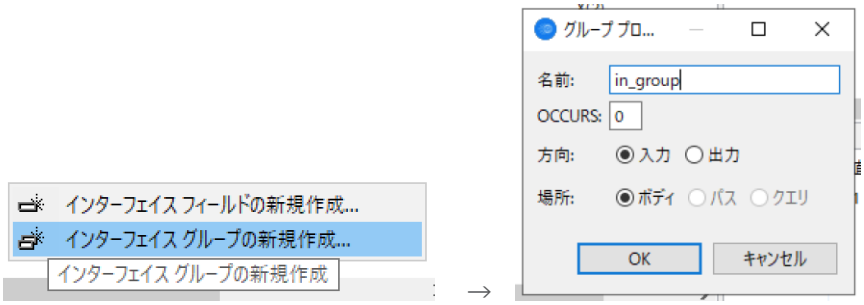
③ LINKAGE SECTION の COBOL の変数「LNK-FUNCTION」を［COBOL 割り当て］にドラッグ＆ドロップします。



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。

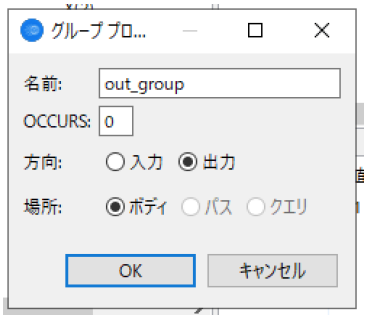


⑤ 次に [SEARCHBOOK オペレーション – インターフェースフィールド] にて右クリックから[インターフェースグループの新規作成]を選択し、[名前]に"in_group"と入力し、[OK]ボタンをクリックします。



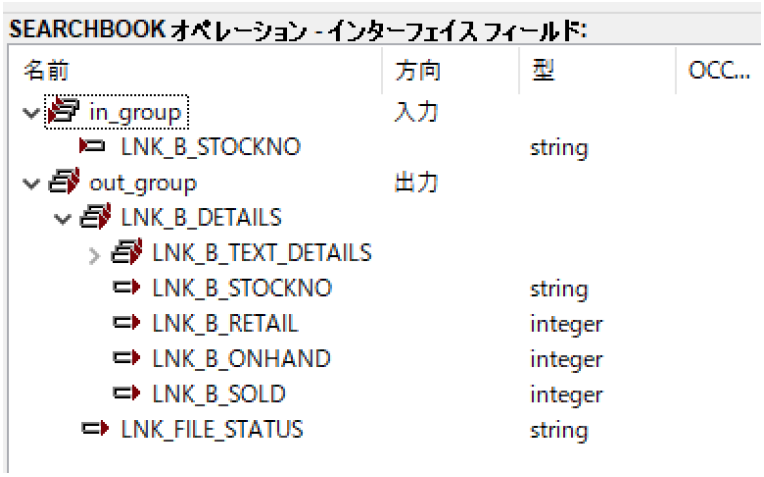
⑥ 作成したグループに「LNK-B-STOCKNO」をドラッグ＆ドロップします。

⑦ 同じく [SEARCHBOOK オペレーション – インターフェースフィールド] にて右クリックから[インターフェースグループの新規作成]を選択し、[名前]に"out_group"と入力し、[方向]を「出力」に変更し、[OK]ボタンをクリックします。



⑧ 作成したグループに「LNK-B-DETAILS」をドラッグ＆ドロップします。

⑨ 最後に「LNK-FILE-STATUS」をドラッグ＆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、コンテクストメニューから［新規作成(N)］>［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

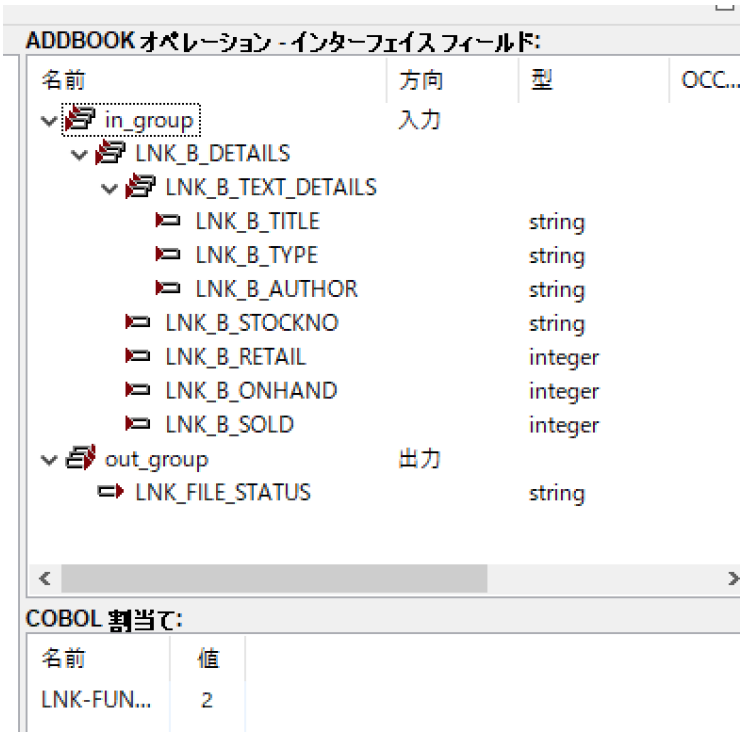
① 「ADDBOOK」オペレーションをダブルクリックします。

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 同様の手順で「in_group」を作成し、「LNK-B-DETAILS」をドラッグ＆ドロップします。

④ 同様の手順で「out_group」を作成し［方向］を出力に変更してから、「LNK_FILE_STATUS」をドラッグ＆ドロップします。

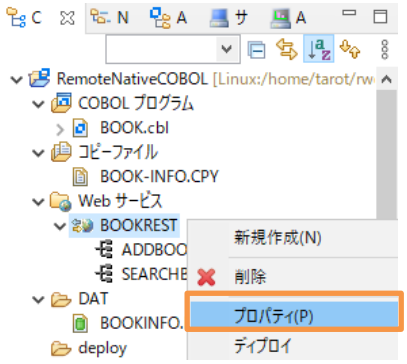
⑤ 最終的には下のイメージのような構成になります。



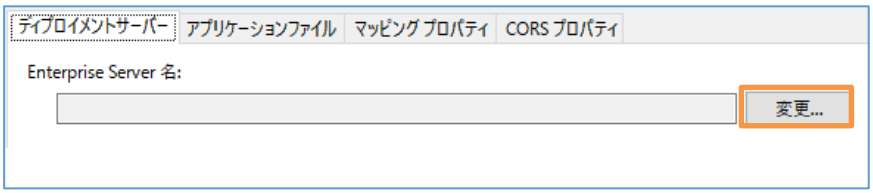
3.5. コンパイルした COBOL アプリケーションを Enterprise Server へディプロイ

1) Enterprise Server へのディプロイ情報を指定

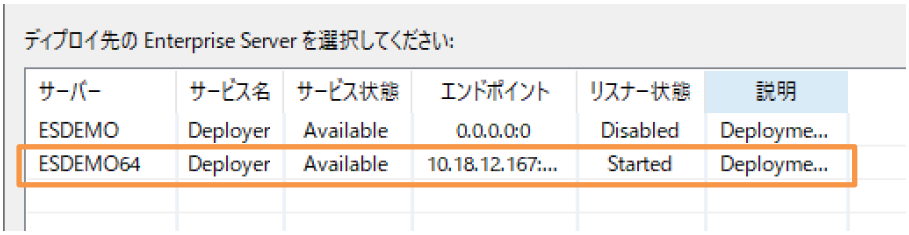
① COBOL エクスプローラーにて追加した Web サービス「BOOKREST」を右クリックし、コンテクストメニューから［プロパティ(P)］を選択します。



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。

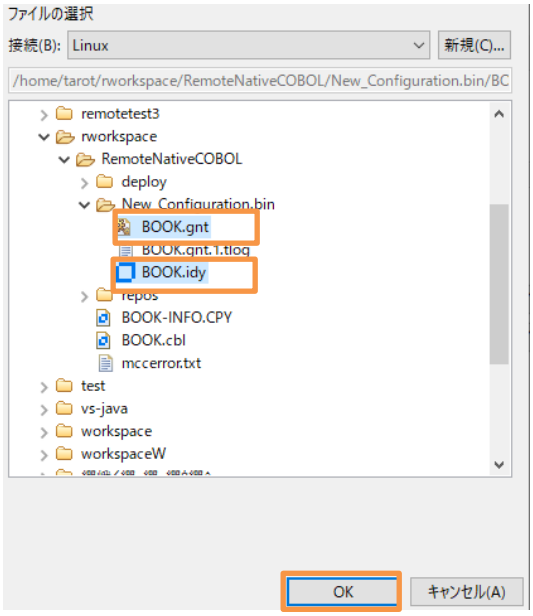


③ 起動済みの Enterprise Server「ESDEMO64」を選択し、［OK］ボタンをクリックします。

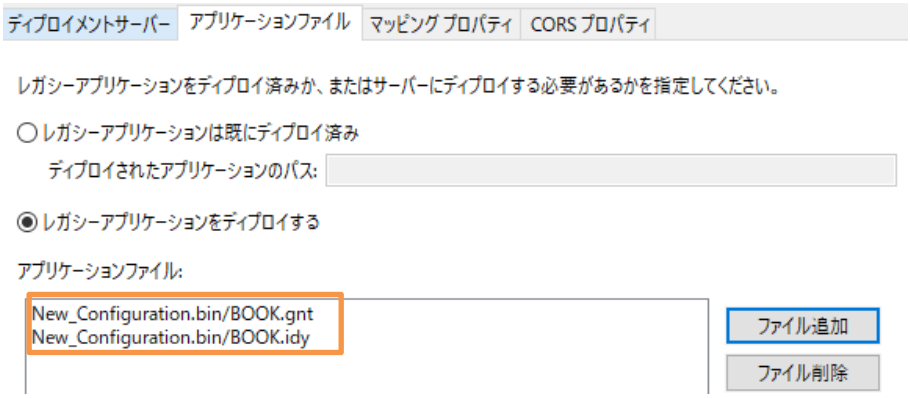


④ 次に［アプリケーションファイル］タブを選択し、「レガシーアプリケーションをディプロイする」を選択します。

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、［OK］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。

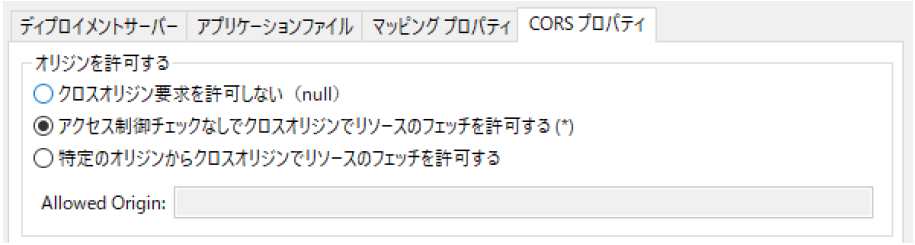


2) オリジン間リソース共有（CORS）を許可

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クストメニューから[プロパティ(P)]を選択します。

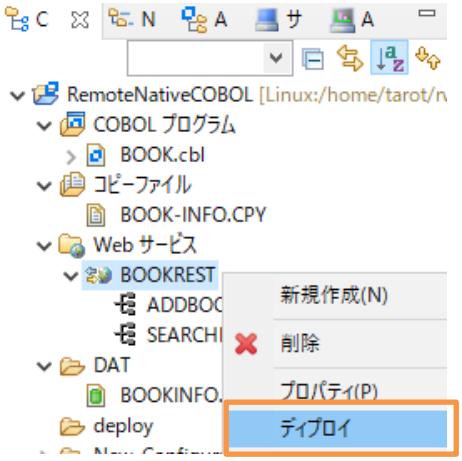
③ [CORS プロパティ]タブを選択し、運用用途に合わせた設定を行います。ここでは「アクセス制限チェックなしでクロスオリジンでリソースのフェッチを許可する(*)」を選択しています。



④ [OK] ボタンをクリックします。

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

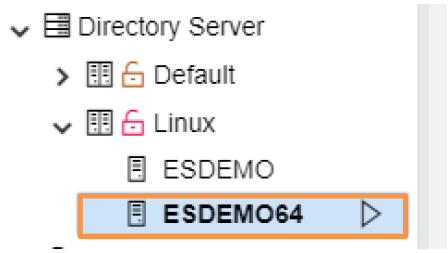
① COBOL エクスプローラーにて作成した Web サービス「BOOKREST」を右クリックし、コンテクストメニューから［ディプロイ］を選択します。



② Eclipse IDE 上からサーバーエクスプローラーを選択します。

③ 「ローカル［localhost:10086］」を選択し、右クリックでコンテクストメニューを表示して、［管理ページを開く］をクリックし、ブラウザーで管理画面を開きます。

④ ESCWA 管理画面にて［ネイティブ］タブを選択し、画面左より［Linux］>［ESDEMO64］をクリックします。



⑤ [一般]メニューから[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 サービスが追加されていることを確認します。

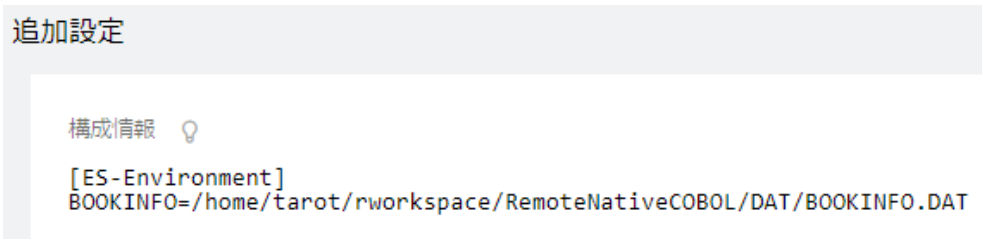
3.6. Enterprise Server インスタンスへの環境変数設定と有効化

1) ブラウザー上の ESCWA 管理画面より、画面上部のネイティブをクリックしたのち、左側より「ESDEMO64」をクリックします。

2) 一般メニューが選択されていることを確認の上、［構成情報］に以下の情報を設定したうえで、画面上部の［適用］をクリックします。

［ES-Environment］

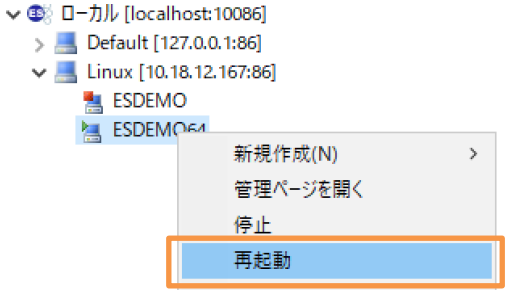
BOOKINFO=/home/tarot/rworkspace/RemoteNativeCOBOL/DAT/BOOKINFO.DAT



注意）
BOOKINFO.dat へのパスは、リモート COBOL プロジェクトの保存先のパスによって調整してください。
Eclipse IDE の COBOL エクスプローラー上で BOOKINFO.dat を選択し、マウスの右クリックでコンテクストメニューを表示し、[プロパティ(P)] を表示することでパスを確認することができます。

3) Eclipse IDE に戻り、サーバーエクスプローラーを選択します。

4) Linux > ESDEMO64 を選択し、右クリックでコンテクストメニューを表示し、[再起動] を選択します。



再起動の状態はコンソールビューにて確認できます。

```
サーバー: ESDEMO64 正常に停止しました
開始 サーバー: ESDEMO64

サーバー: ESDEMO64 正常に起動されました
```

3.7. RESTful Web サービスのテスト

1) Windows のスタートメニューより、コマンドプロンプトを起動し、チュートリアル用ファイルを解凍したフォルダに移動します。

2) 検索機能をテストするため、以下のコマンドをコマンドプロンプト上で実行します。

curl http://IP アドレス:9003/temppath/BOOKREST/1.0/SEARCHBOOK -d @json¥search1111.txt

注意）
サーバーアドレスやポート番号は実際の環境に合わせて変更してください。

```
C:¥work>curl http://10.18.12.167:9003/temppath/BOOKREST/1.0/SEARCHBOOK -d
@json¥search1111.txt
{
  "LNK_B_DETAILS" :
  {
    "LNK_B_TEXT_DETAILS" :
    {
      "LNK_B_TITLE" : "LORD OF THE RINGS",
      "LNK_B_TYPE" : "FANTASY",
```

```
      "LNK_B_AUTHOR" : "TOLKIEN"
    },
    "LNK_B_STOCKNO" : "1111",
    "LNK_B_RETAIL" : 1500,
    "LNK_B_ONHAND" : 4000,
    "LNK_B_SOLD" : 3444
  },
  "LNK_FILE_STATUS" : "00"
}
```

3) 追加機能をテストするため、以下のコマンドをコマンドプロンプト上で実行します。

まずは、既存データがない事を検索機能で確認します。

curl http://IP アドレス:9003/temppath/BOOKREST/1.0/SEARCHBOOK -d @json¥search9999.txt

```
C:¥work>curl http://10.18.12.167:9003/temppath/BOOKREST/1.0/SEARCHBOOK -d
@json¥search9999.txt
{
  "LNK_B_DETAILS" :
  {
    "LNK_B_TEXT_DETAILS" :
    {
      "LNK_B_TITLE" : "**************************************************",
      "LNK_B_TYPE" : "********************",
      "LNK_B_AUTHOR" : "**************************************************"
    },
    "LNK_B_STOCKNO" : "",
    "LNK_B_RETAIL" : 0,
    "LNK_B_ONHAND" : 0,
    "LNK_B_SOLD" : 0
  },
  "LNK_FILE_STATUS" : "23"
}
```

続いて、追加機能でデータを追加します。

curl http://IP アドレス:9003/temppath/BOOKREST/1.0/ADDBOOK -d @json¥add9999.txt

```
C:¥work>curl http://10.18.12.167:9003/temppath/BOOKREST/1.0/ADDBOOK -d
@json¥add9999.txt
{
  "LNK_FILE_STATUS" : "00"
}
```

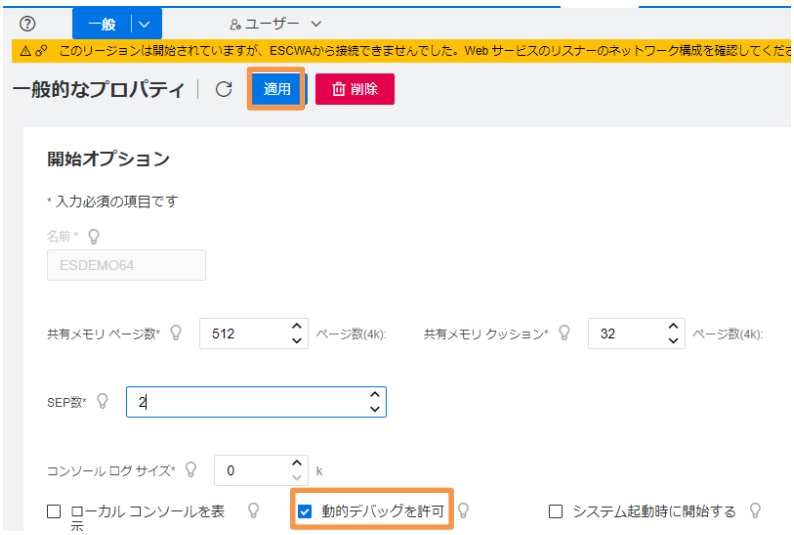
検索機能で、正しく追加されたことを確認します。

curl http://IP アドレス:9003/temppath/BOOKREST/1.0/SEARCHBOOK -d @json¥search9999.txt

```
C:¥work>curl http://10.18.12.167:9003/temppath/BOOKREST/1.0/SEARCHBOOK -d
@json¥search9999.txt
{
  "LNK_B_DETAILS" :
  {
    "LNK_B_TEXT_DETAILS" :
    {
      "LNK_B_TITLE" : "ALICE'S ADVENTURES IN WONDERLAND",
      "LNK_B_TYPE" : "FANTASY",
      "LNK_B_AUTHOR" : "LEWIS CARROLL"
    },
    "LNK_B_STOCKNO" : "9999",
    "LNK_B_RETAIL" : 100,
    "LNK_B_ONHAND" : 200,
    "LNK_B_SOLD" : 300
  },
  "LNK_FILE_STATUS" : "00"
}
```

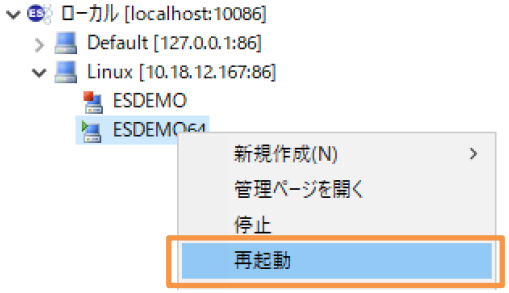
3.8. RESTful Web サービスのデバッグ

1) ブラウザー上の ESCWA 管理画面より、画面上部のネイティブをクリックしたのち、左側より「ESDEMO64」をクリックします。一般メニューが選択されていることを確認の上、[動的デバッグを許可] にチェックをいれたうえで、画面上部の [適用] をクリックします。



2) Eclipse IDE に戻り、サーバーエクスプローラーを選択します。

3) Linux > ESDEMO64 を選択し、右クリックでコンテクストメニューを表示し、[再起動] を選択します。



再起動の状態はコンソールビューにて確認できます。

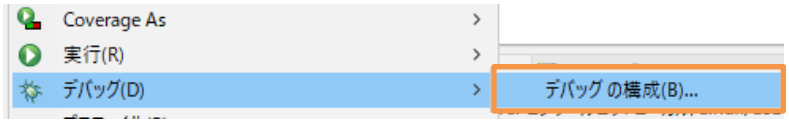
```
サーバー: ESDEMO64 正常に停止しました
開始 サーバー: ESDEMO64

サーバー: ESDEMO64 正常に起動されました
```

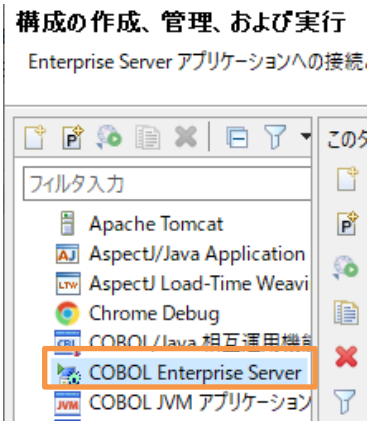
4) デバッガーのアタッチ

① Eclipse IDE の COBOL エクスプローラーを表示します。

② RemoteNativeCOBOL プロジェクトを選択し、マウスの右クリックにてコンテクストメニューを表示した上で、[デバッグ(D)] > [デバッグの構成(B)] をクリックします。

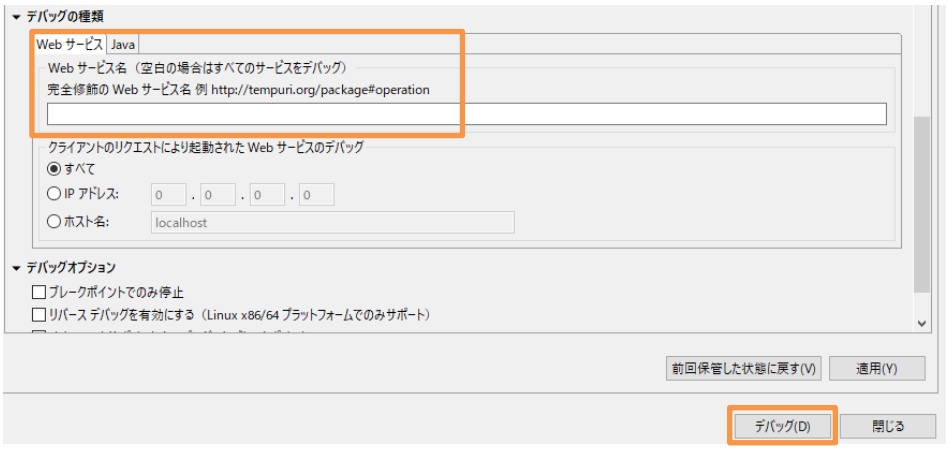


③ 画面左側より、[COBOL Enterprise Server] をダブルクリックします。



④ [Enterprise Server] セクションの [参照] をクリックして、Linux > ESDEMO64 を選択します。選択後は、以下のように表示されます。

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] が空欄になっていることを確認したうえで、[デバッグ(D)] をクリックします。

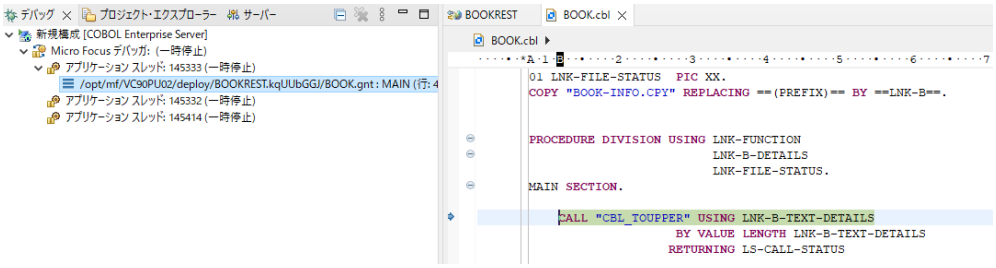


⑥ 3.7 と同様、検索処理のリクエストを行ってください。実行するコマンドは、以下になります。

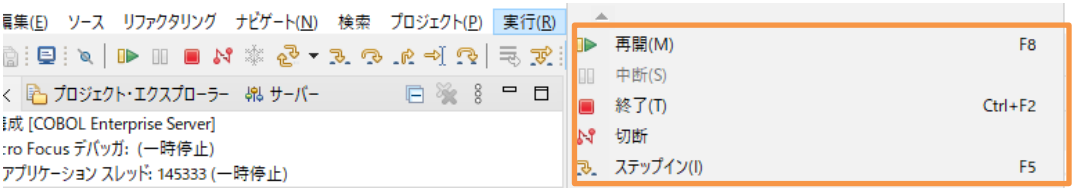
curl http://IP アドレス:9003/temppath/BOOKREST/1.0/SEARCHBOOK -d @json¥search1111.txt

「IP アドレス」は、環境に合わせて変更してください。。

⑦ Eclipse IDE に戻ると、デバッガーが起動し、プログラムが停止しています。



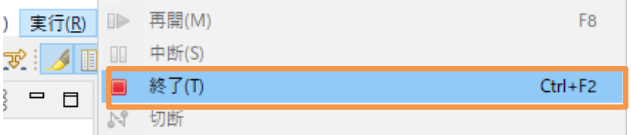
通常のコンソールアプリケーションのデバッグと同様、[実行] メニュー配下の [再開] や [ステップイン] などといった操作が利用できます。



[ステップイン] にて、1 行毎にプログラムを進めることができます。

[再開] などで、デバッグが終了すると、コマンドラインに結果が戻されます。

⑧ デバッグの終了を行う場合は、[実行(R)] > [終了(T)] をクリックします。

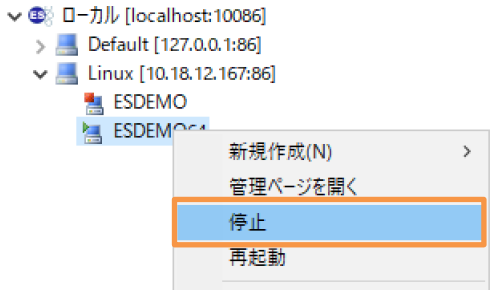


3.9. インスタンスの停止

1) Enterprise Server インスタンスの停止

① 「サーバーエクスプローラー」に切り替えます。

② Linux > ESDEMO64 上で右クリックし、コンテクストメニューから［停止］を選択し、Enterprise Server インスタンスを停止します。



3.10. サービス、デーモンの停止

1) 3.1 同様、root ユーザーで、ターミナル画面を開き、cobsetenv を実行したうえで、以下のコマンドを実行します。

① ディレクトリサーバーの停止

mfds -s 2

② Linux リモートサーバーの停止

$COBDIR/remotedev/stoprdodaemon

停止してよいかの確認が求められますので、”y” を入力します。

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