Visual COBOL チュートリアル

JCA による JBOSS EAP 連携の設定と開発

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 EJB として再利用し、Java EE ライアントから呼び出す機能を提供しています。EJB コンポーネントとして呼び出しを行う場合、Java アプリケーションサーバー上の Java EE クライアントは JCA の仕様にもとづいたリクエストを Enterprise Server に渡し、 COBOL のビジネスロジックが処理をして結果を返します。また、Enterprise Server は、XA に準拠しているので同じく XA に準拠しているデータベースや他のシステムと協調してトランザクション処理を行うことができます。

このドキュメントでは Windows Server 上の JBOSS EAP 7.4 と Enterprise Server を JCA による連携を行い、Eclipse で開発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他の Linux/UNIX でも利用可能です。

● アプリケーションサーバー ソフトウェア

OS	Windows 10
Java	Adoptium 17.0.11.9
AP サーバー製品	JBoss Enterprise Application Server 7.4.16

● 開発クライアント ソフトウェア

OS	Windows 10
COBOL 環境製品	Visual COBOL 10.0 for Eclipse
PostgreSQL ODBC Driver	psqlodbc 16.00.00.00
Windows SDK	10.0.26100.1

本チュートリアルでは、アプリケーションサーバーと開発クライアントは同一環境としていますが、別環境構成も可能です。

なお、本チュートリアルでは、データベースに PostgreSQL を使用していますので、PostgreSQL 環境が必要です。

チュートリアルを実施するにあたり、必要なテーブル、レコードについては、以下のチュートリアル用サンプルプログラム配下の setup_database.sql に記載しておりますので、事前にテーブル作成、レコード挿入を実施してください。

- 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

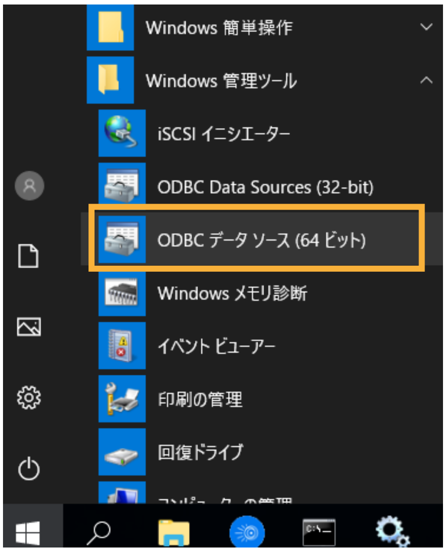
内容

3 チュートリアル手順

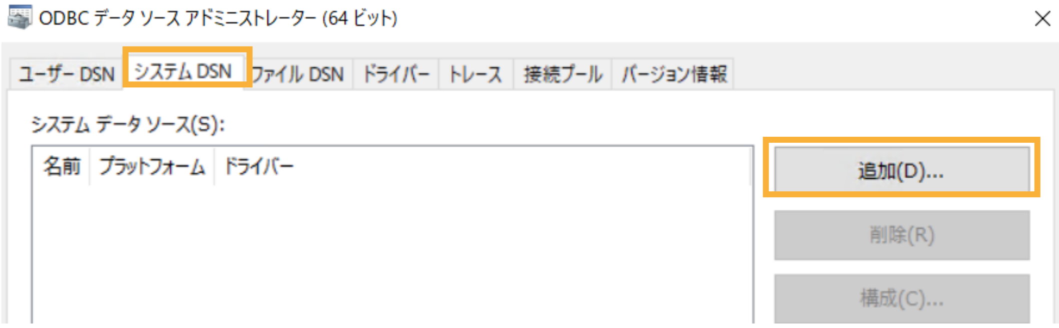
3.1 データソースの作成

1) データソースの作成

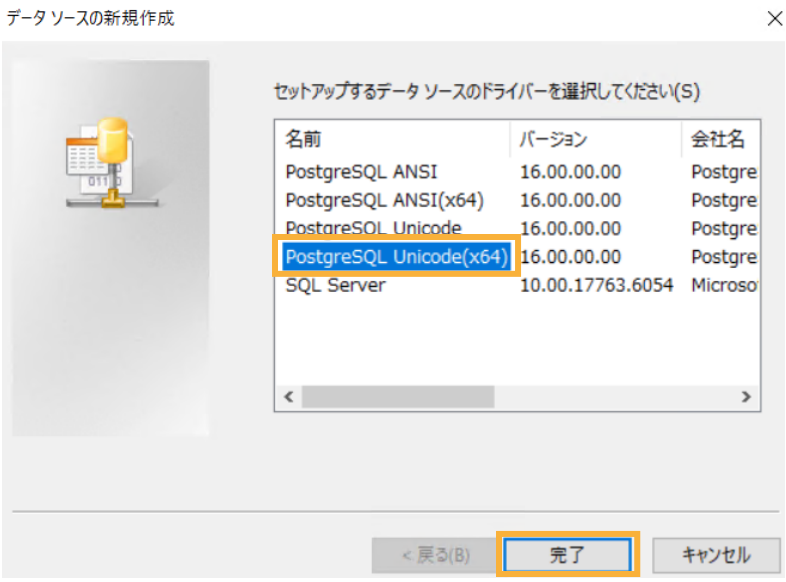
① [スタート] メニュー > [Windows 管理ツール] > [ODBC データ ソース (64 ビット)] を選択します。



② [システム DSN] をクリックし、[追加(D)] をクリックします。



[PostgreSQL Unicode(x64)] を選択し、[完了] をクリックします。



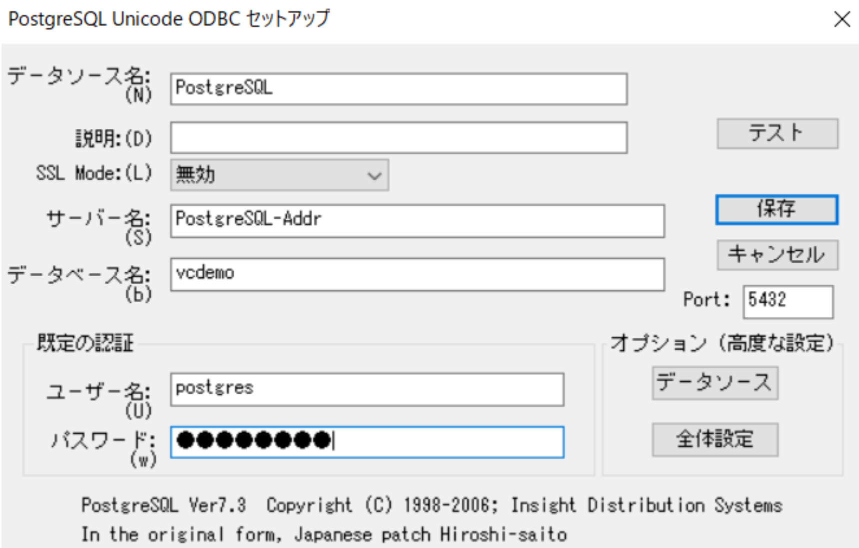
③ 以下の入力を行い、[作成(C)] をクリックします。

データソース名： PostgreSQL

サーバー名： PostgreSQL サーバーのアドレス

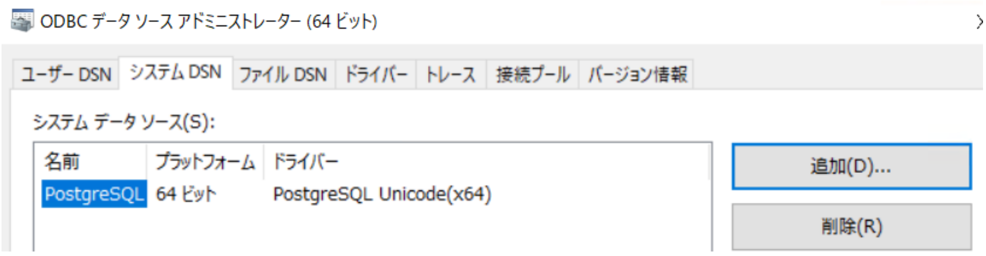
Port： PostgreSQL サーバーのリッスンポート番号

データベース名： チュートリアルで使用するサンプルテーブルの格納データベース名



なお、[保存] 前に [テスト] で構成情報に誤りがないかを確認ください。

データソースが登録されたことを確認したのち、本画面を閉じます。



3.2 ODBC 用の XA スイッチモジュールの作成

2) XA スイッチモジュールの作成

① [スタート] メニュー > [Micro Focus Visual COBOL] > [Visual COBOL コマンドプロンプト(64-bit)] を選択します。

② <製品のインストールフォルダー>¥src¥enterpriseserver 配下の xa フォルダーを任意のフォルダーにコピーします。本手順では C ドライブ直下にコピーします。

補足）
デフォルトのインストールフォルダーは、C:¥Program Files (x86)¥Micro Focus¥Visual COBOL です。

③ ① のコマンドプロンプト上で、以下のコマンドを実行して XA スイッチモジュールをビルドします。

注意）
各箇条書きのコマンドはそれぞれ 1 行で入力してください。LIB 環境変数に、Windows SDK へのパスを設定し

ていますが、このパスはデフォルトのインストールフォルダーです。異なるフォルダーを指定した場合はそちらに修正してください。
また、環境変数設定時にはパス内に含まれる半角スペースにご注意ください。

- cd ¥xa
- set LIB="c:¥Program Files (x86)¥Windows Kits¥10¥Lib¥10.0.26100.0¥um¥x64";"c:¥Program Files (x86)¥Windows Kits¥10¥Lib¥10.0.26100.0¥um¥x86";%LIB%
- build odbc

以下のように、「ESODBCXA_D.dll」と「ESODBCXA.dll」ファイルが作成されます。

```
C:¥xa>dir |sort /R
2024/08/05  11:50    <DIR>          ..
2024/08/05  11:50    <DIR>          .
2024/08/05  11:50            96,768 ESODBCXA_D.dll
2024/08/05  11:50            88,064 ESODBCXA.dll
2024/04/23  15:36           121,877 ESORAXA.CBL
2024/04/23  15:36            10,852 build.bat
2024/02/08  17:44            48,866 xapd.cpy
（以下、省略）
```

④

3) リソースアダプターファイルの構成

① <製品のインストールフォルダー>¥javaee フォルダーを任意のフォルダーにコピーします。本チュートリアルでは、C ドライブ直下にコピーします。

② さきほど使用したコマンドプロンプト画面で上記フォルダーに移動したのち、”ravaluesupdater.bat” を実行します。実行後は以下の入力を行い、Enter キーを押します。

Please enter the application server you would like to update: “jboss74EAP”

ServerHost に対する Please enter the resource adapter you would like to update: “mfcobol-xa.rar”

ServerPort に対する Would you like to change the value of ServerHost? (y/n/reset to default x to exit): “n”

Trace に対する Would you like to change the value of ServerHost? (y/n/reset to default x to exit): “x”

Any changes have already been saved. Are you sure you want to exit? Please enter y to confirm: “y”

```
C:¥javaee>ravaluesupdater.bat
Your available application servers are:
ibmwebsphere855
ibmwebsphere90
ibmwebsphereliberty
jboss74EAP
oracleweblogic1221
oracleweblogic1411
```

```
Please enter the application server you would like to update: jboss74EAP
Your available resource adapters are:
mfcobol-localtx.rar
mfcobol-notx.rar
mfcobol-xa.rar

Please enter the resource adapter you would like to update: mfcobol-xa.rar
ServerHost is currently set to: localhost (Default: localhost)
Would you like to change the value of ServerHost?  (y/n/reset to default x to exit)
n
ServerPort is currently set to: 9003 (Default: 9003)
Would you like to change the value of ServerPort?  (y/n/reset to default x to exit)
n
Trace is currently set to: false (Default: false)
Would you like to change the value of Trace?  (y/n/reset to default x to exit)
x
Any changes have already been saved. Are you sure you want to exit? Please enter
y to confirm:
y
C:¥javaee>
```

javaee7¥jboss74EAP¥mfcobol-xa.rar が更新されます。

3.3 Visual COBOL for Eclipse を起動してソースコードのインポート

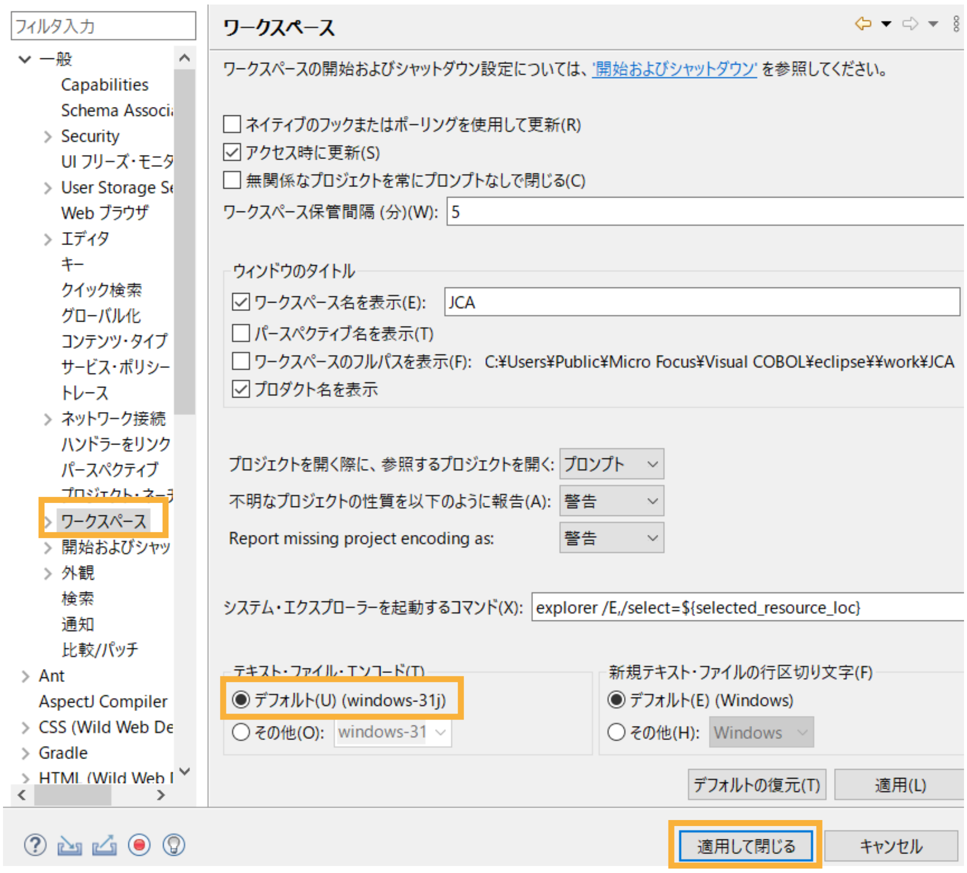
1) Visual COBOL for Eclipse を起動

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 任意の作業フォルダーを指定します。ここでは "C:¥work¥JCA" を指定します。

2) 文字コードの指定

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(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「デフォルト(windows-31j)」に変更し、[適用して閉じる] をクリックします。

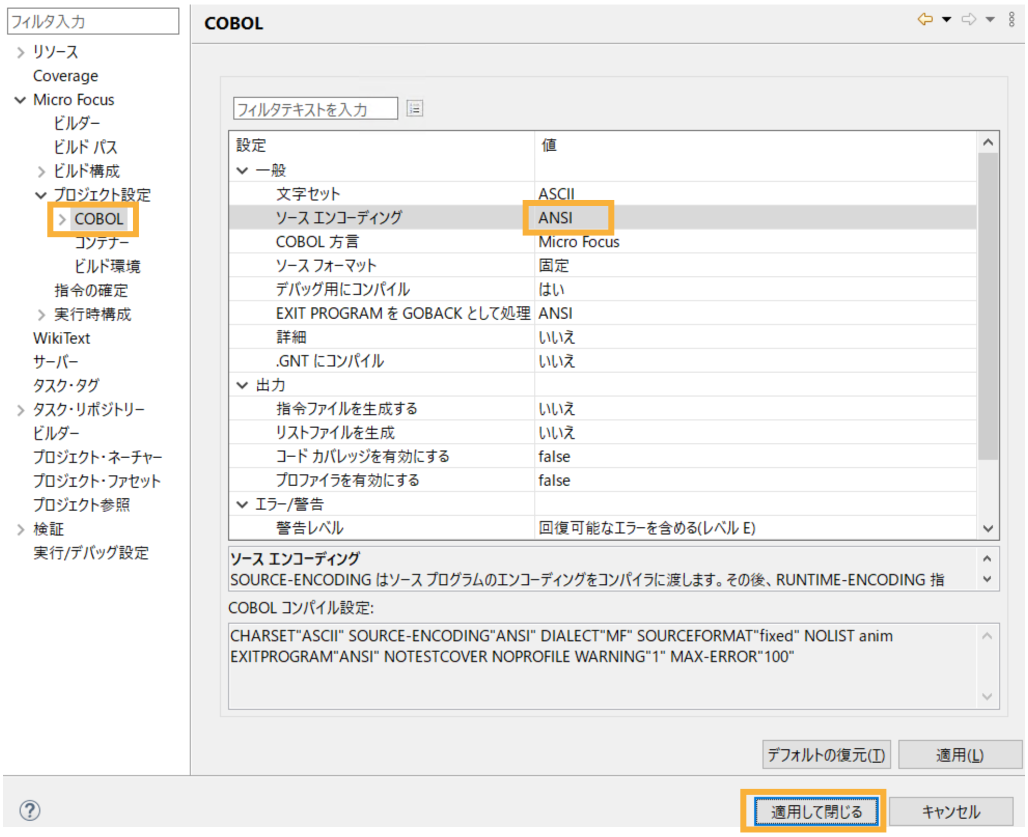


※Preference Recorder のダイアログが表示されたら [キャンセル] を選択してください。

3) ネイティブ COBOL プロジェクトの作成

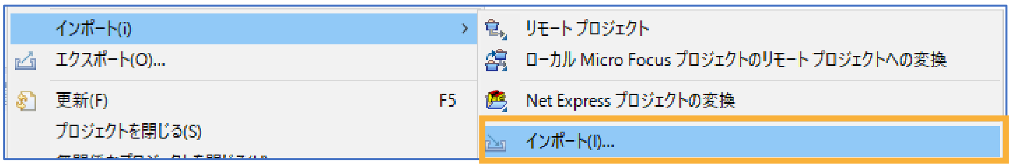
① [ファイル(F)] メニュー > [新規(N)] > [COBOL プロジェクト] を選択し、プロジェクト名に "NativeCOBOL" を入力し、プロジェクトテンプレートに「64 ビット」を指定して、[終了(F)]を押します。

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クストメニューを表示し、[プロパティ(R)] 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。

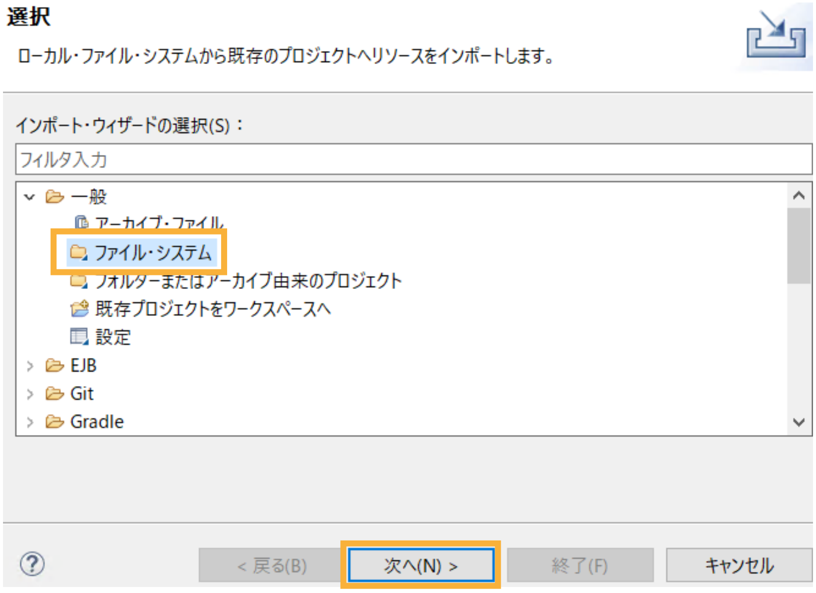


4) プログラムソースのインポート

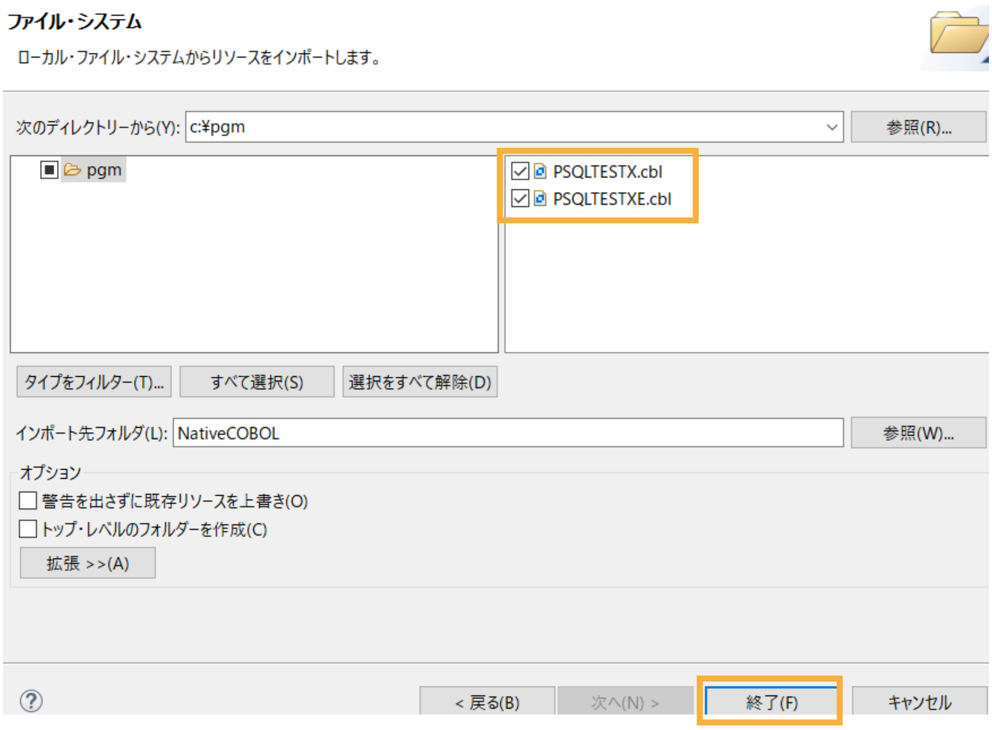
① COBOL エクスプローラーのパースペクティブを開き、COBOL エクスプローラービューにて プロジェクトフォルダーを右クリックし、コンテクストメニューから [インポート(I)] > [インポート(I)] を選択します。



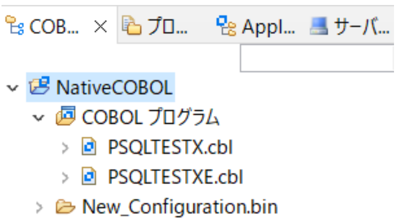
② サンプルファイルをインポートします。一般のフォルダーを展開し、[ファイル・システム] を選択し、[次へ(N)] ボタンを押します。



③　インポートダイアログが表示されるので［参照(R)］をクリックしてソースコードを解凍したフォルダーを指定します。ここでは“C:¥pgm”を指定しています。配下の「PSQLTESTX.cbl」と「PSQLTESTXE.cbl」を指定し、[終了(F)]を押します。



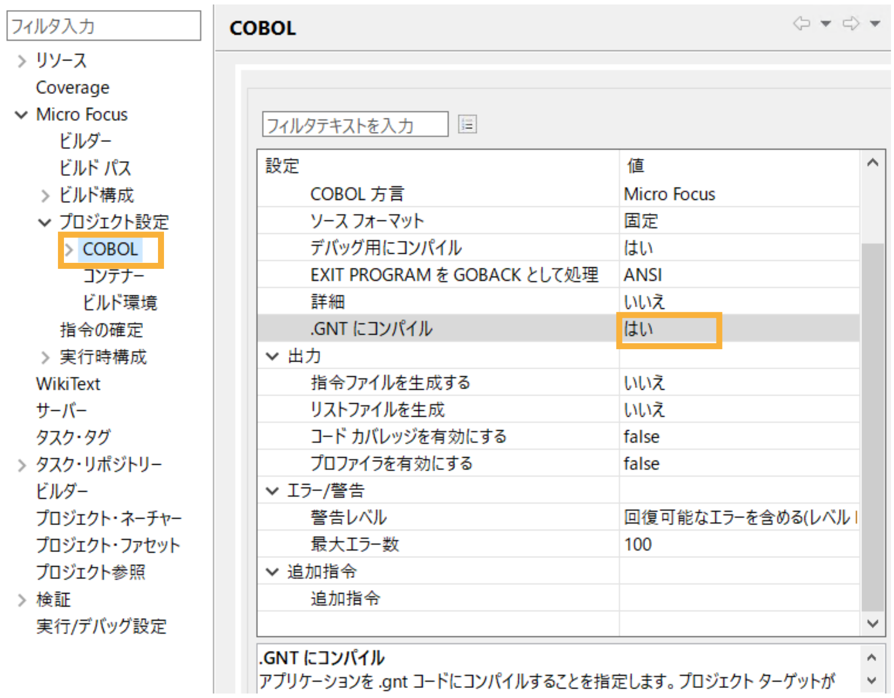
④　プロジェクトフォルダーを展開し、２つのファイルが正常にロードされていることを確認します。



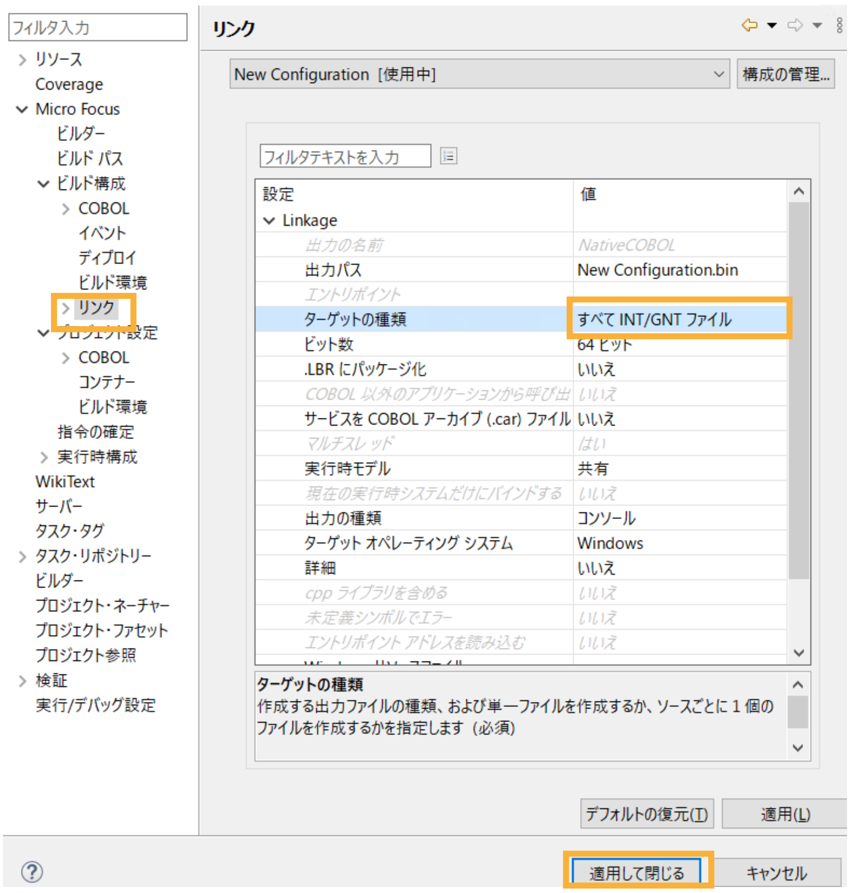
5)　ビルドオプションの変更

① COBOL エクスプローラーにて作成したプロジェクトを右クリックし、[プロパティ(R)] を選択します。

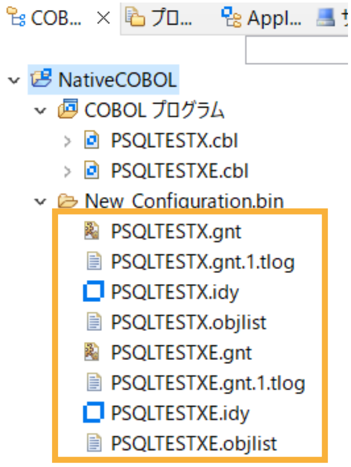
② プロパティ設定ダイアログが表示されます。[Micro Focus] > [プロジェクト設定] > [COBOL] を選択、[.GNT にコンパイル] を “いいえ” から “はい” に変更したうえで、[適用(L)] をクリックします。



③ [Micro Focus] > [ビルド構成] > [リンク] を選択し、[ターゲットの種類] を「すべて INT/GNT ファイル」に変更し、[適用して閉じる] をクリックします。



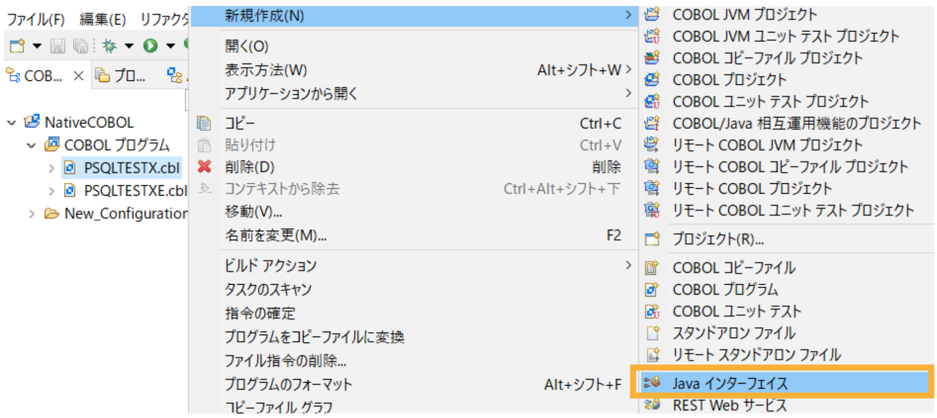
④ COBOL エクスプローラーにて”New_Configuration.bin”を展開して下記のファイルが作成されていることを確認します。



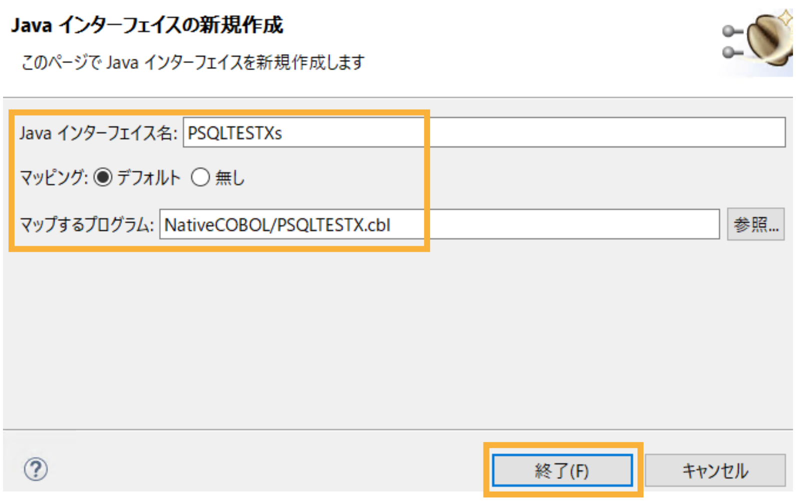
3.4 JCA 用のアプリケーションプロファイルの作成

1) EJB/JCA アプリケーションプロファイルの作成

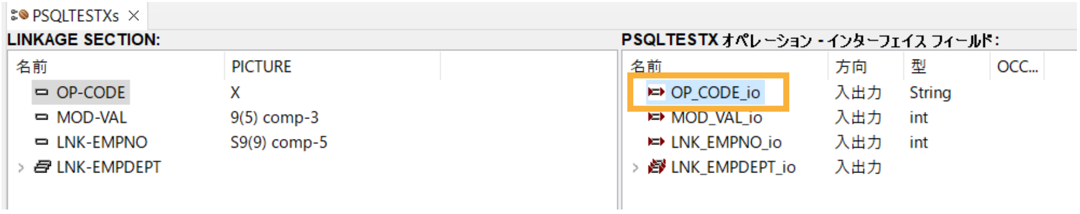
① EJB/JCA アプリケーションとして利用する「PSQLTESTX.cbl」を右クリックし、コンテクストメニューから［新規作成(N)］>［Java インターフェイス］を選択します。

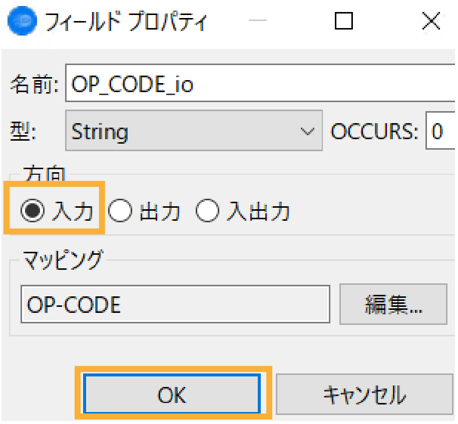


② ［Java インターフェイス名］欄には ”PSQLTESTXs” を入力し、［マッピング］欄は “デフォルト” を選択し、マッピングするプログラムは “PSQLTESTX.cbl” が選択されていることを確認し［終了(F)］を押します。

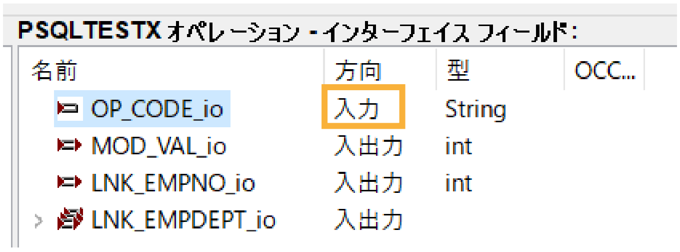


③ [PSQLTESTXs オペレーション・インターフェイス フィールド] 欄の [OP_CODE_io] をダブルクリックしたうえで、[方向] に ”入力” を選択し [OK] をクリックします。





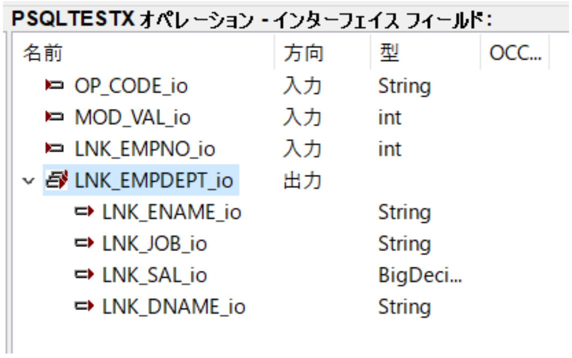
[OP_CODE_io] の方向が “入力” になったことを確認します。



④ 上記と同じ手順で、以下の項目の方向を変更し、保存します。

名前	方向
MOD_VAL_IO	入力
LNK_EMPNO_io	入力
LNK_EMPDEPT_io	出力

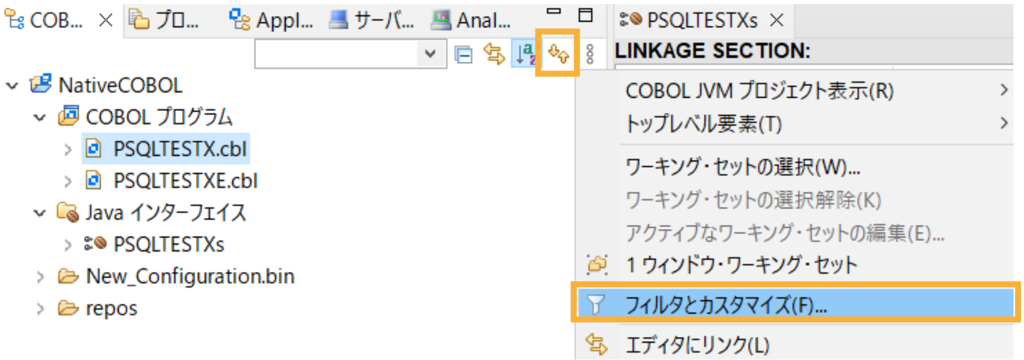
変更後は以下のようになります。



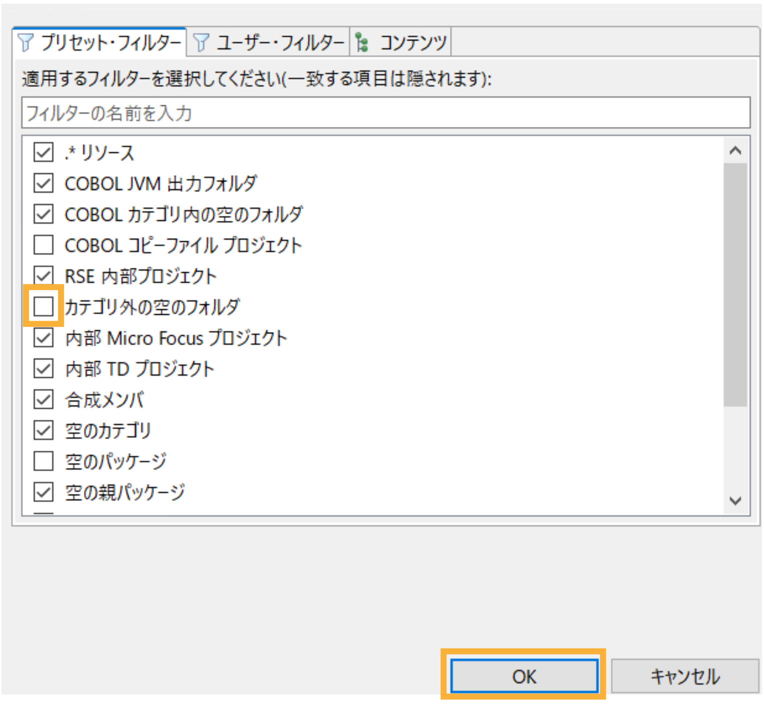
3.5 ディプロイ先の変更

1) 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。



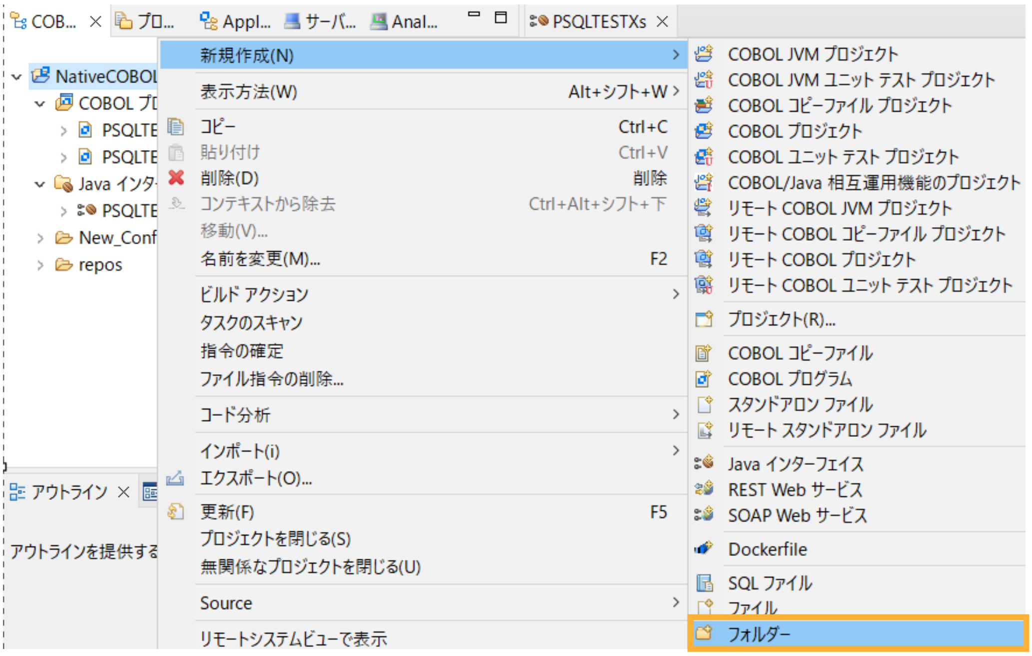
② [カテゴリ外の空のフォルダ] にチェックされている場合は、チェックを外したのち、[OK] ボタンをクリックします。



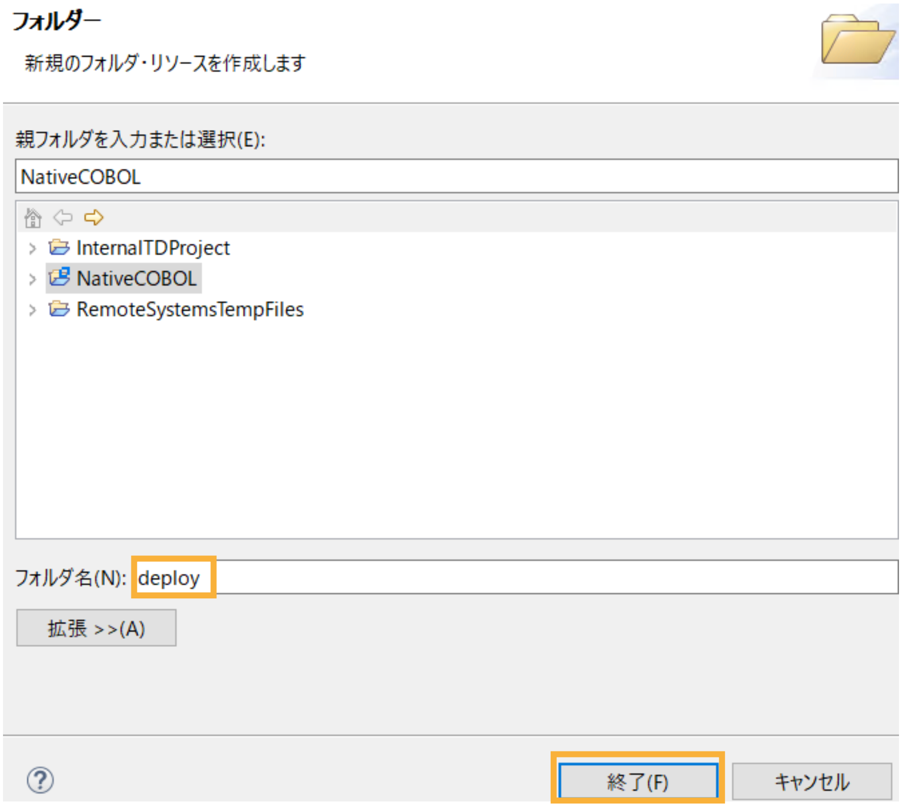
2) ディプロイ用フォルダーの作成

① 「NativeCOBOL」プロジェクト上で右クリックし、コンテクストメニューから [新規作成(N)]→[フォルダー] を選択し

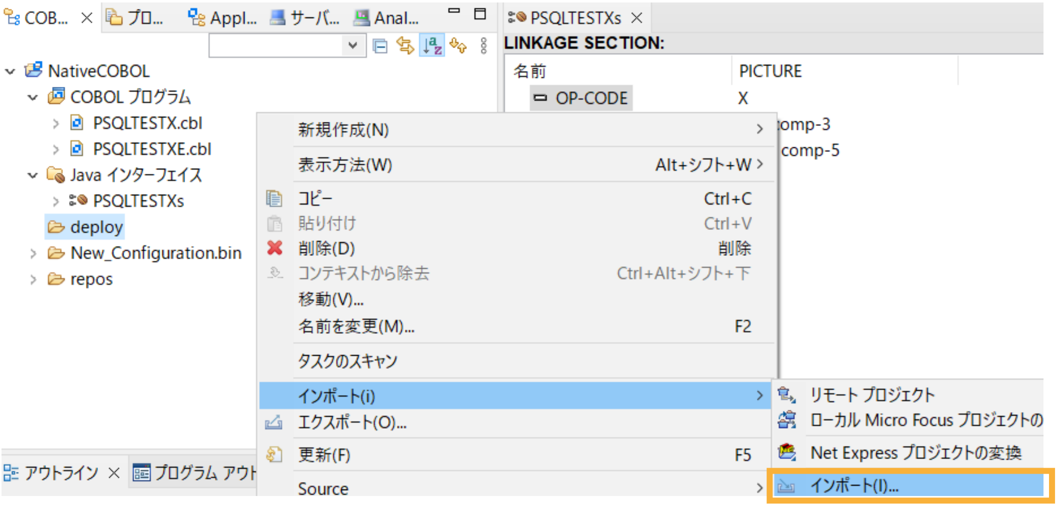
ます。



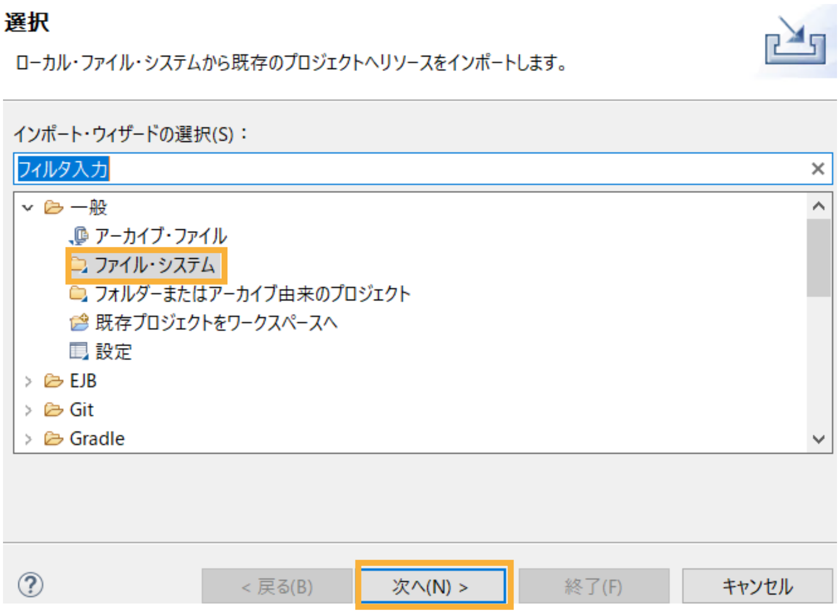
② 「NativeCOBOL」プロジェクトが選択されていることを確認の上、フォルダー名に "deploy" を指定し、[終了(F)] をクリックします。



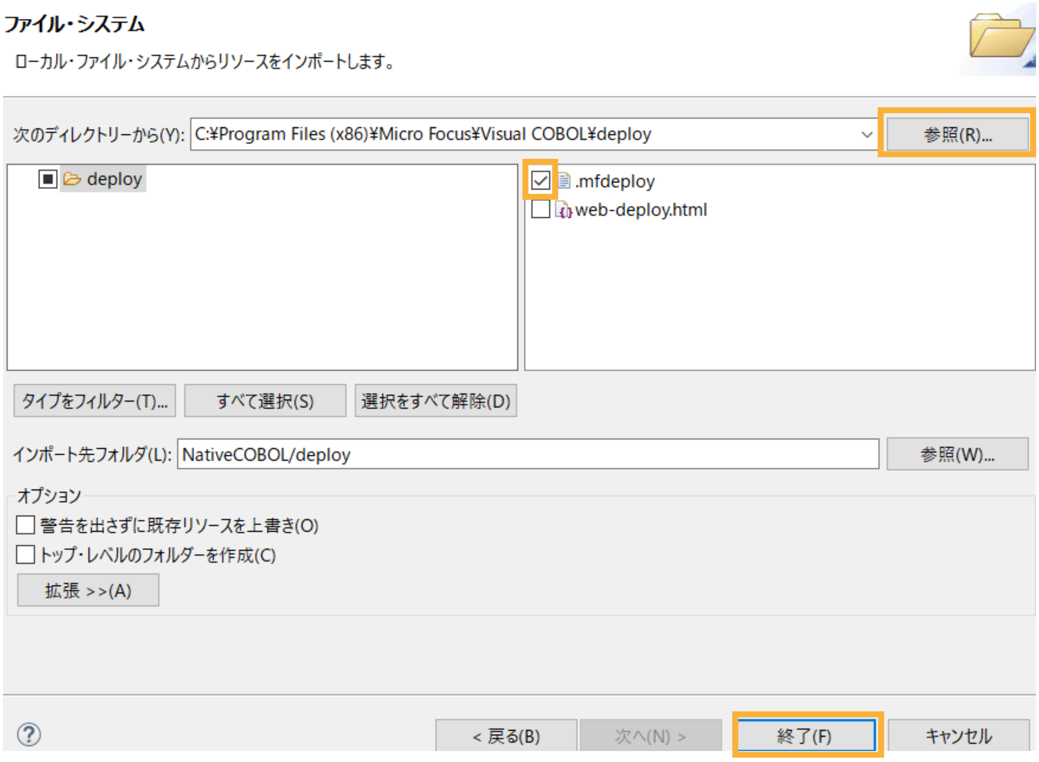
③ 作成した「deploy」フォルダー上で右クリックし、コンテクストメニューから [インポート] > [インポート] を選択します。



④ [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。



⑤ [参照(R)] をクリックし、<製品のインストールフォルダー¥deploy> を選択し、[.mfdeploy] ファイルにチェックしたうえで、[終了(F)] をクリックします。



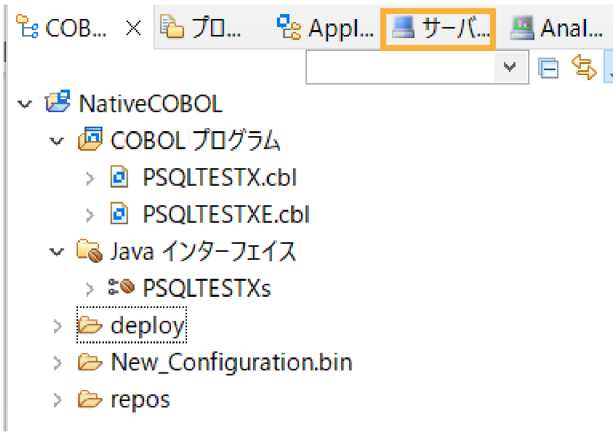
補足）

上記手順完了後も COBOL エクスプローラー上の NativeCOBOL¥deploy フォルダー配下に .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) Enterprise Server インスタンスの設定

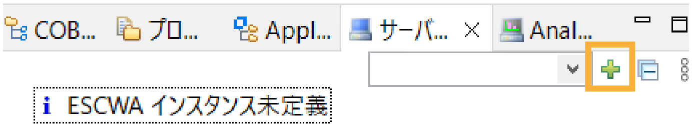
① Eclipse に戻り、[サーバーエクスプローラー] を開きます。



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。

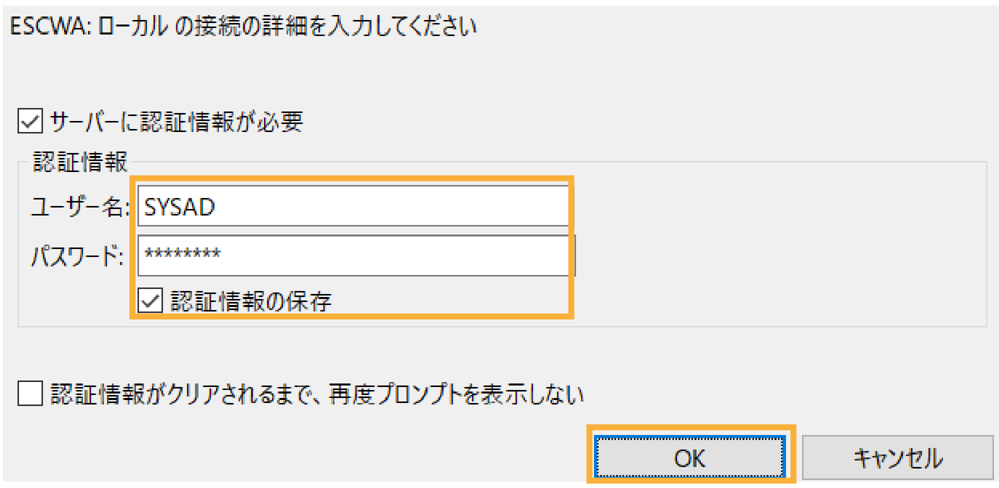
名前： “ローカル”

サーバアドレス(IPv4/ホスト名)： “localhost”

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



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。

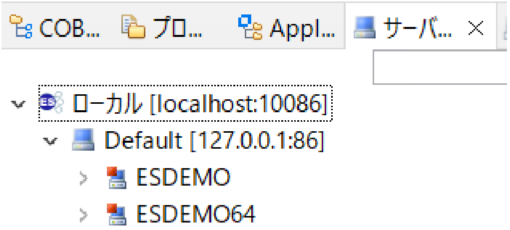
スタートメニューより、[Micro Focus Visual COBOL] > [Visual COBOL コマンドプロンプト(64 ビット)] を選択します。

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。

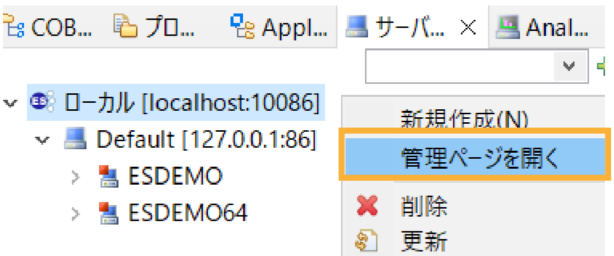
```
C:¥xa>mfsecretsadmin read microfocus/temp/admin
{"mfUser":"SYSAD", "mfPassword":"LZQe4VS3"}
C:¥xa>
```

Enterprise Server インスタンスが追加されます。

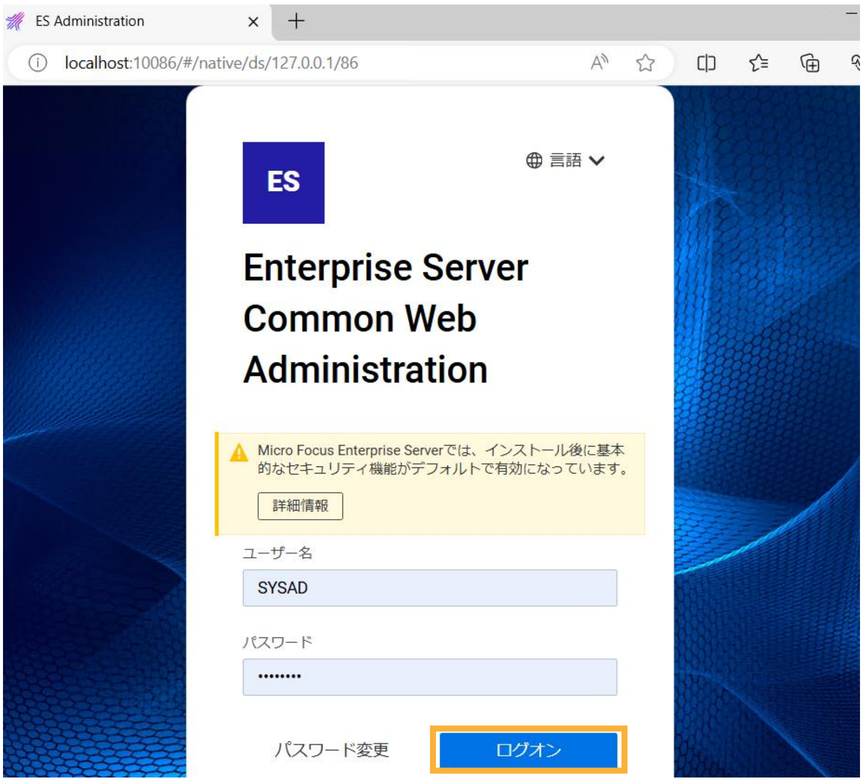


4) ESCWA 管理画面の表示

① [ローカル] > [Default] を選択したうえで、マウスの右クリックでコンテクストメニューを開き、[管理ページを開く] を選択します。



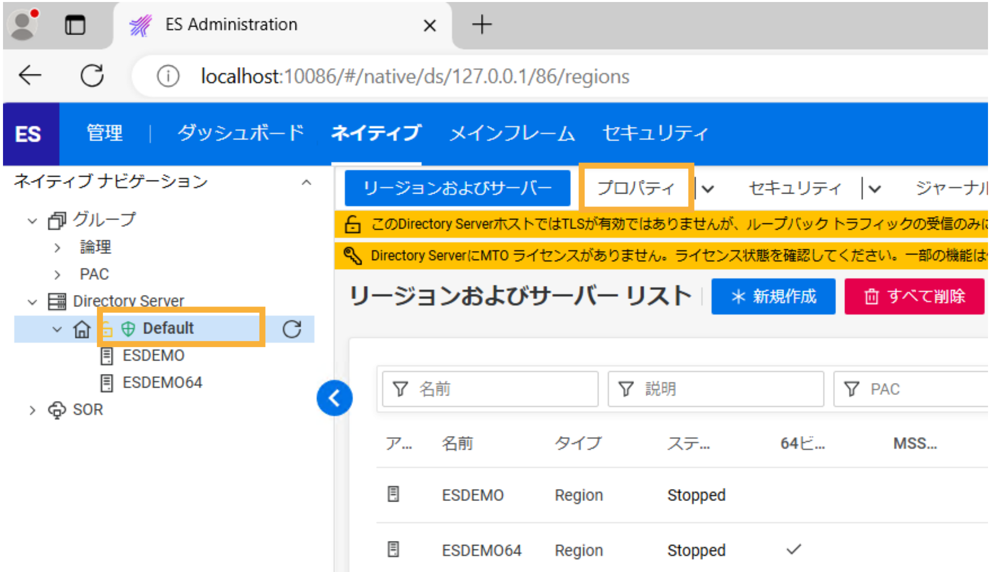
② ブラウザーが開いたら、さきほどの認証情報を入力して、[ログオン] をクリックします。



ログインに成功すると、ESCWA (Enterprise Server Common Web Administration) 画面にてダッシュボードが表示されますので、[ネイティブ] をクリックします。

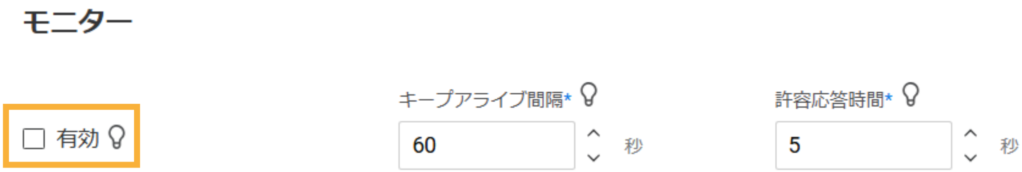
5) 不要ログの抑止

① ダッシュボード画面左側より、[Directory Server] > [Default] をクリックし、上部のメニューより [プロパティ] をクリックします。



② 以下の入力を行い、[適用] をクリックします。

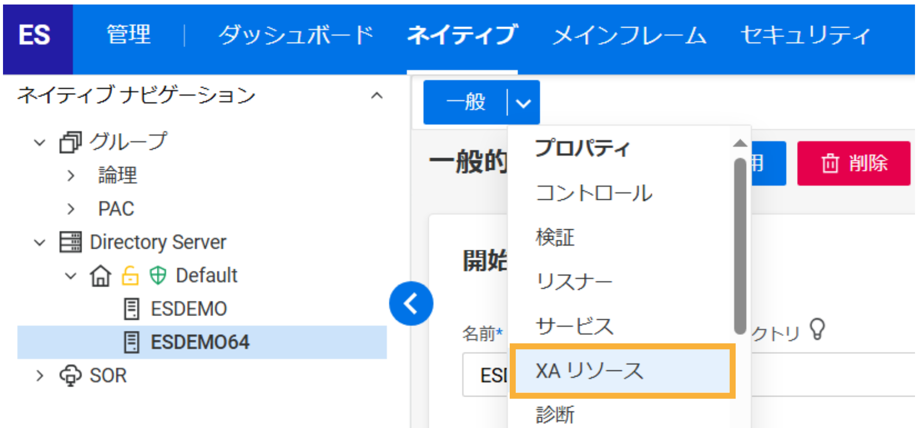
モニター ＞ 有効のチェックをはずす



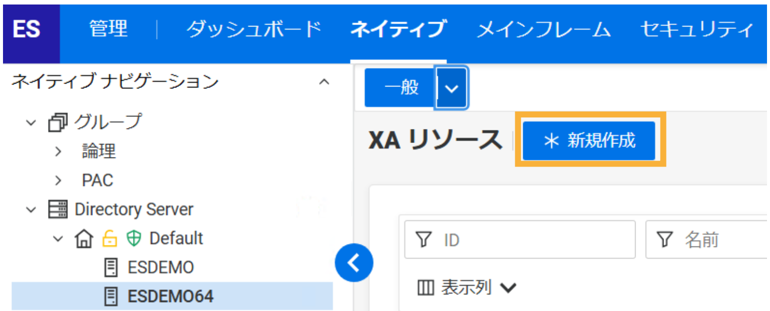
6) XA スイッチモジュールの構成

① ESCWA 画面左のツリーより、[Directory Server] > [Default] > [ESDEMO64] をクリックします。

② ［一般］をクリックすると表示される［XA リソース］をクリックします。



［新規作成］をクリックします。



③ 以下の入力を行い、[OK］をクリックします。

ID: “PSQLXA”

名前： “PSQLXA”

モジュール： “c:¥xa¥ESODBCXA_D.dll”

OPEN 文字列： “DSN=PostgreSQL”

補足）

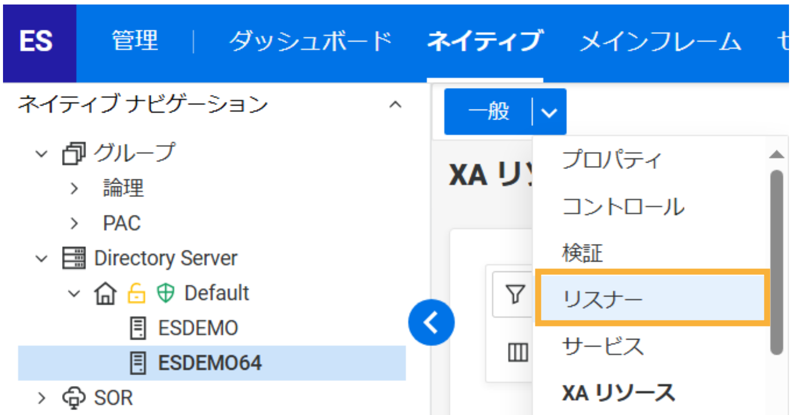
モジュールには、3.2 で作成した ESODBCXA_D.dll ファイルを絶対パスで指定します。

OPEN 文字列の DSN= の後は、3.1 で作成したデータソース名を入力します。

XA リソースが登録されます。

7) リスナーの構成

① ESCWA 画面の [一般] をクリックしたのちに表示される [リスナー] を選択します。

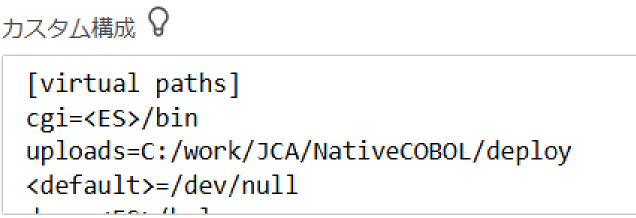


② 画面左下より、[通信プロセス１] > [Web] をクリックします。

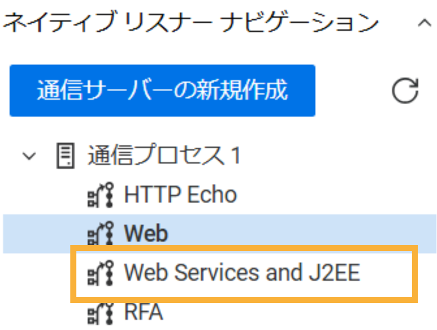
③ リスナープロパティのカスタム構成情報が表示されます。アプリケーションのディプロイ先は uploads 項目で定義されますが、現在は「uploads=<ES>/deploy」となっています。この場合、Visual COBOL インストールディレクトリ配下の deploy フォルダーがディプロイ用フォルダーとして使用されます。通常、Program Files (x86)等のフォルダーは管理者権限を持つユーザーフォルダー書き込みできフォルダー変更を行います。下記のように変更し、[適用] をクリックします。

uploads=C:/work/JCA/NativeCOBOL/deploy

> 補足）
> ここで指定しているフォルダーパスは、2) で作成した deploy フォルダーに対応します。



④ 画面左下部より、[通信プロセス１] > [Web Services and J2EE] を選択します。

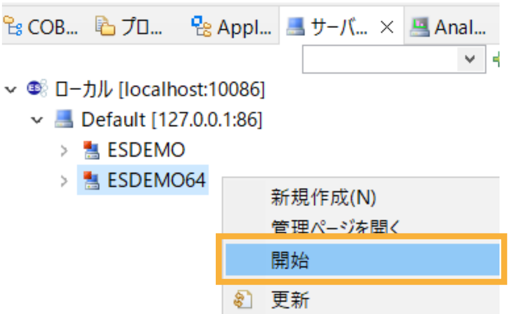


⑤ ［ポート］ に “9003” を入力して、［適用］ をクリックします。

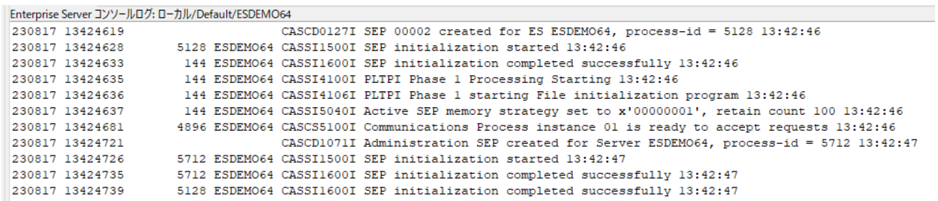
8) Enterprise サーバーインスタンスの起動

① Eclipse に戻り、［サーバーエクスプローラー］ を開きます。

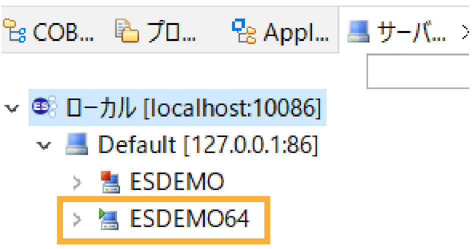
② ［ESDEMO64］ を選択したうえでマウスの右クリックでコンテクストメニューを開き、［開始］ を選択します。



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。



正常に起動しますと、緑のアイコンになります。



9) JBoss へリソースアダプタのディプロイ

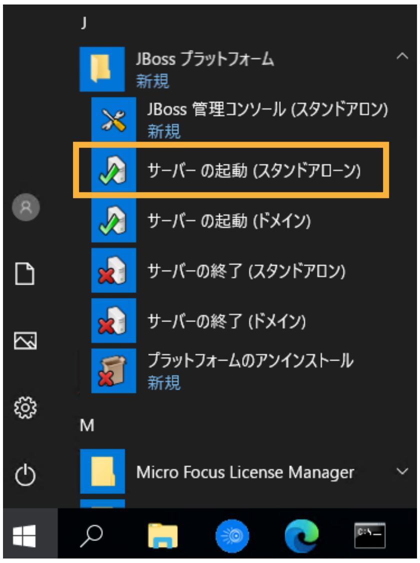
① 構成ファイル「standalone.xml」の変更を行います。

<JBoss インストールフォルダー>¥standalone¥configuration¥standalone.xml をエディターで開き、下記手順に従って、mfcobol-xa.rar の設定を行います。

https://www.amc.rocketsoftware.co.jp/manuals/VC100/Eclipse/vc100indx.html

[ディプロイ] > [Enterprise Server へのディプロイ] > [モダナイズ済みのアプリケーションのディプロイおよび構成] > [EJB およびリソース アダプターのディプロイ] > [EJB のディプロイ - 概要] > [JBoss へのディプロイ]

② Windows メニューより [JBoss プラットフォーム] > [サーバーの起動（スタンドアローン] を選択します。



コンソールログからエラーがないことを確認してください。

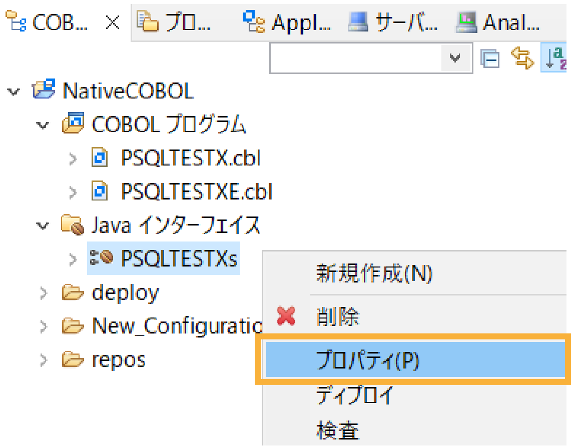
③ <JBoss インストールフォルダー>¥standalone¥deployments 配下に「リソースアダプターファイルの構成」で作成された mfcobol-xa.rar をコピーします。

コンソールログから mfcobol-xa.rar のデプロイがエラーとなっていないことを確認します。

```
16:44:36,690 INFO  [org.jboss.as.repository] (DeploymentScanner-threads - 1) WFLYDR0001: ロケーション C:¥EAP-7.4¥standalone¥data¥content¥7d¥8b6b53fa83a874cea0279ae2ea9d945c1e82cf¥content にコンテンツが追加されました。
16:44:36,709 INFO  [org.jboss.as.server.deployment] (MSC service thread 1-1) WFLYSRV0027: "mfcobol-xa.rar" (runtime-name: "mfcobol-xa.rar") のデプロイメントを開始しました。
16:44:37,814 INFO  [org.jboss.as.connector.deployment] (MSC service thread 1-3) WFLYJCA0007: 接続ファクトリ java:/eis/MFCobol_v1.5 を登録しました
（中略）
16:44:38,358 INFO  [org.jboss.as.clustering.infinispan] (ServerService Thread Pool -- 72) WFLYCLINF0002: ejb コンテナー から http-remoting-connector キャッシュを開始しました。
16:44:38,443 INFO  [org.jboss.as.server] (DeploymentScanner-threads - 1) WFLYSRV0010: "mfcobol-xa.rar" (runtime-name : "mfcobol-xa.rar") をデプロイしました。
```

10) JCA アプリケーションのディプロイ

① Eclipse にもどり、COBOL エクスプローラー上から NativeCOBOL プロジェクト配下の [Java インターフェイス] > [PSQLTESTXs] を選択、マウスの右クリックでコンテクストメニューを開き、[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、以下の作業を行います。

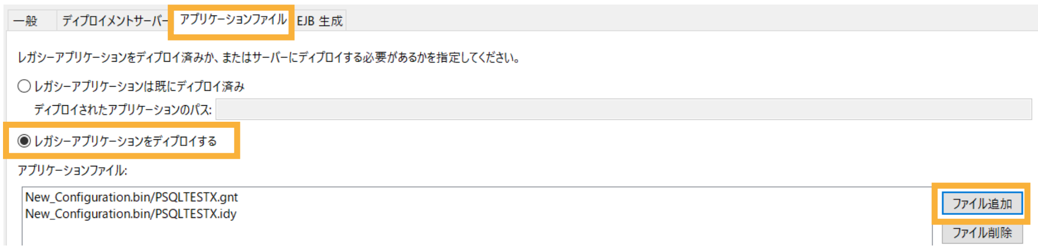
Enterprise Server 名：[変更] をクリックして、[ESDEMO64] を選択

トランザクション管理： “コンテナ管理” を選択

③ [アプリケーションファイル] を選択し、以下の作業を行います。

[レガシーアプリケーションをディプロイします] を選択

[ファイル追加] をクリックし、C:¥work¥JCA¥NativeCOBOL¥New_Configuration.bin 配下の「PSQLTESTX.gnt」と「PSQLTESTX.idy」を選択



④ [EJB 生成] を選択し、以下の作業を行ったうえで [OK] をクリックします。

アプリケーションサーバー： “JEE 7” / “JBoss EAP 7.4” を選択

Java コンパイラ： 使用している Java コンパイラへのパス

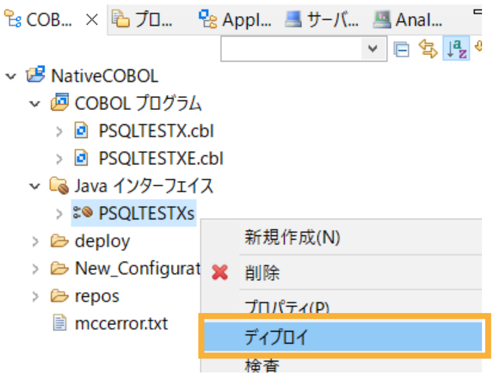
Java EE クラスパス： 以下の jar ファイルを指定

いずれも <JBoss EAP インストールフォルダー>¥modules¥system¥layers¥base¥javax

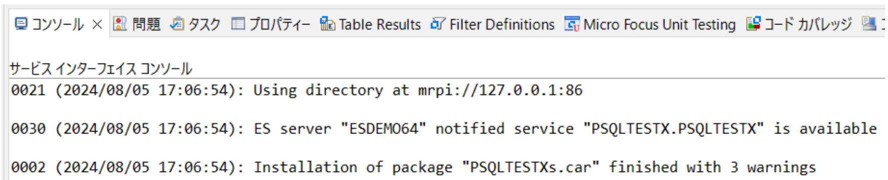
- annotation/api/main/jboss-annotations-api_1.3_spec-2.0.1.Final-redhat-00001.jar
- ejb/api/main/jboss-ejb-api_3.2_spec-2.0.0.Final-redhat-00001.jar
- resource/api/main/jboss-connector-api_1.7_spec-2.0.0.Final-redhat-00001.jar

- servlet/api/main/jboss-servlet-api_4.0_spec-2.0.0.Final-redhat-00001.jar

⑤ COBOL エクスプローラー上の NativeCOBOL プロジェクト配下の ［Java インターフェイス］ > ［PSQLTESTXs］ を選択、マウスの右クリックでコンテクストメニューを開き、［ディプロイ］ を選択します。



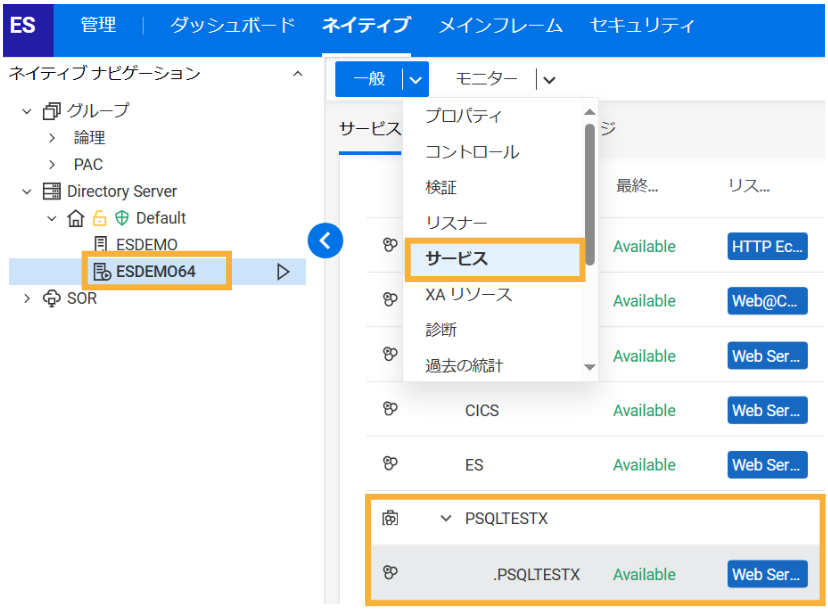
成功すると、［サービス インターフェイス コンソール］ ビューでは、以下のログが出力されます。



また、ESCWA 画面からも以下の手順で確認できます。

- 左側より ［ESDEMO64］ を選択
- ［一般］ > ［サービス］ を選択

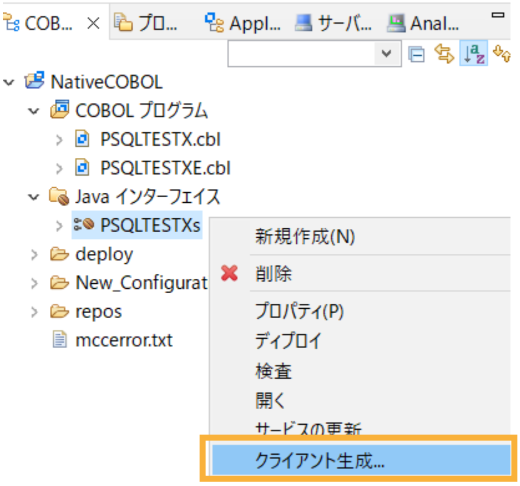
  PSQLTESTX がサービスとして登録されています。



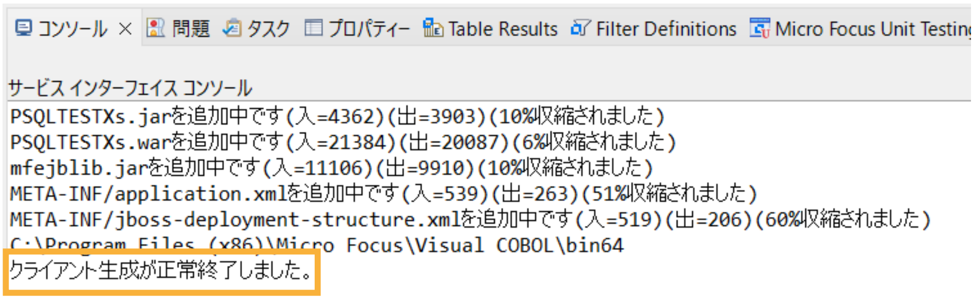
11）テストクライアントアプリケーションの生成

Visual COBOL 製品にはテストクライアントを生成する機能があり、それを用いてテストを行います。

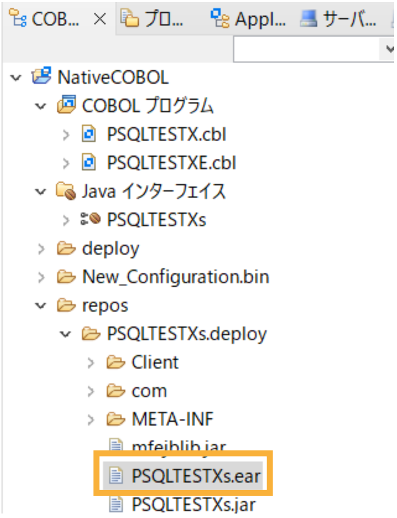
① COBOL エクスプローラー上の NativeCOBOL プロジェクト配下の [Java インターフェイス] > [PSQLTESTXs] を選択、マウスの右クリックでコンテクストメニューを開き、[クライアント生成] を選択します。



[サービス インターフェイス コンソール] ビューに結果が出力されます。



② COBOL エクスプローラー上の NativeCOBOL プロジェクト配下の repos¥を選択し、<JBoss EAP インストールフォルダー>¥standalone¥deployments にコピーします。



JBoss EAP のコンソールに、以下のようなログが出力されます。

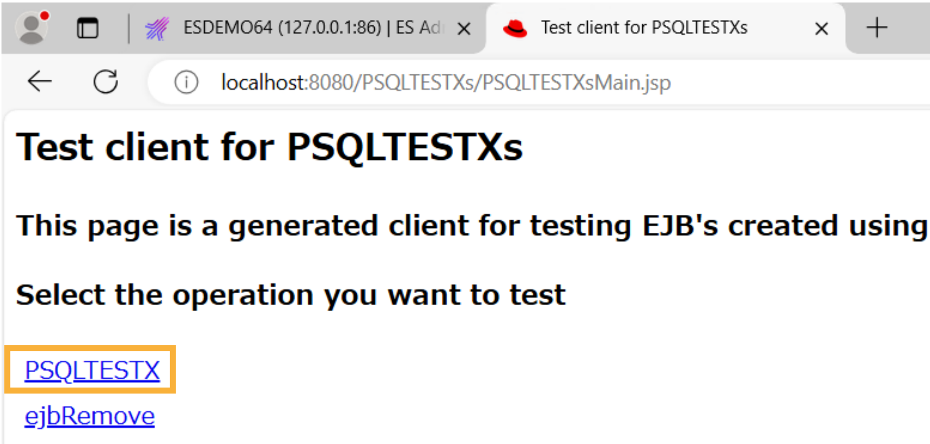
```
（省略）
17:21:39,796 INFO  [org.jboss.weld.deployer] (MSC service thread 1-8) WFLYWELD0003: Weld デプロイメント PSQLTESTXs.ear  を処理しています
17:21:39,994 INFO  [org.hibernate.validator.internal.util.Version] (MSC service thread 1-8) HV000001: Hibernate Validator 6.0.22.Final-redhat-00002
17:21:40,122 INFO  [org.jboss.weld.deployer] (MSC service thread 1-1) WFLYWELD0003: Weld デプロイメント PSQLTESTXs.war  を処理しています
17:21:40,130 INFO  [org.jboss.weld.deployer] (MSC service thread 1-4) WFLYWELD0003: Weld デプロイメント PSQLTESTXs.jar  を処理しています
17:21:40,133 INFO  [org.jboss.as.ejb3.deployment] (MSC service thread 1-4) WFLYEJB0473: デプロイメントユニット 'subdeployment "PSQLTESTXs.jar" of deployment "PSQLTESTXs.ear"'の 'PSQLTESTXsEJB' という名前のセッション Bean の JNDI バインディ ングは次のとおりです:
        java:global/PSQLTESTXs/PSQLTESTXs.jar/PSQLTESTXsEJB!com.mypackage.PSQLTESTXs.PSQLTESTXsLocal
        java:app/PSQLTESTXs.jar/PSQLTESTXsEJB!com.mypackage.PSQLTESTXs.PSQLTESTXsLocal
        java:module/PSQLTESTXsEJB!com.mypackage.PSQLTESTXs.PSQLTESTXsLocal
        java:global/PSQLTESTXs/PSQLTESTXs.jar/PSQLTESTXsEJB
        java:app/PSQLTESTXs.jar/PSQLTESTXsEJB
        java:module/PSQLTESTXsEJB

17:21:40,191 INFO  [org.jboss.weld.Version] (MSC service thread 1-5) WELD-000900: 3.1.6 (redhat)
17:21:40,856 INFO  [org.wildfly.extension.undertow] (ServerService Thread Pool -- 72) WFLYUT0021: 登録された web コンテ キスト: '/PSQLTESTXs' (サーバー 'default-server' 用)
17:21:40,877 INFO  [org.jboss.as.server] (DeploymentScanner-threads - 1) WFLYSRV0010: "PSQLTESTXs.ear" (runtime-name : "PSQLTESTXs.ear") をデプロイしました。
```

12) テストアプリケーションの実行

① ブラウザーを開き、以下のアドレスにアクセスし、[PSQLTESTX] リンクをクリックします。

http://localhost:8080/PSQLTESTXs/PSQLTESTXsMain.jsp



② データを更新します。

サンプルレコードの EMPNO: 7934 の情報は以下の通りです。

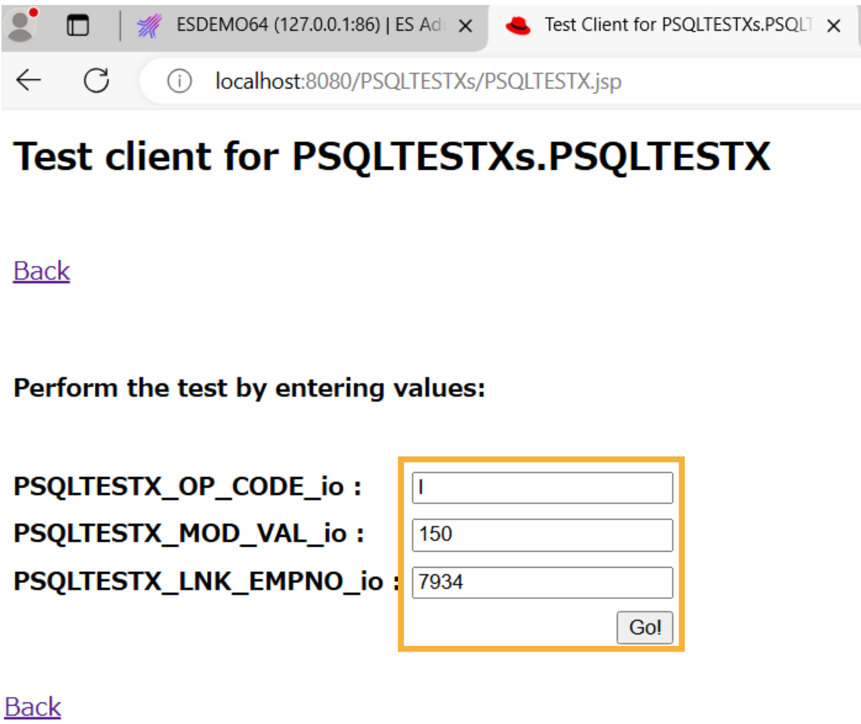
EMPNO	7934
ENAME	CMILLER
JOB	CLEARK
SAL	1300.00
DEPTNO	3

Web 画面で以下の項目を入力して、[Go!] をクリックします。

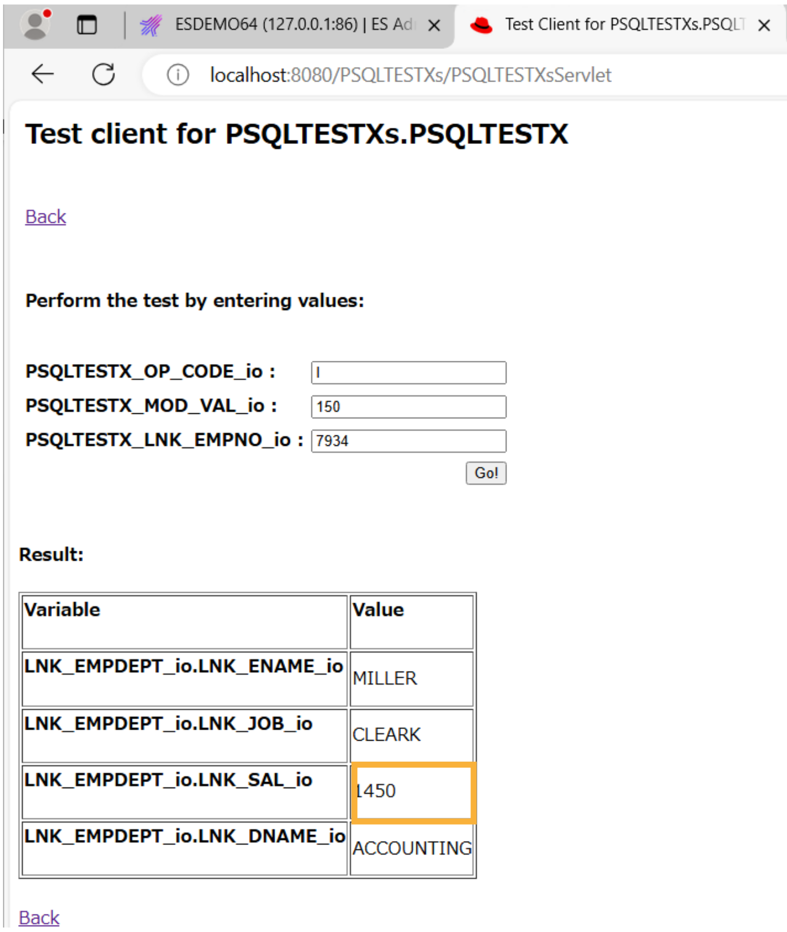
PSQLTESTX_OP_CODE_io: “I”

PSQLTESTX_MOD_VAL_io: “150”

PSQLTESTX_LNK_EMPNO_io: “7934”



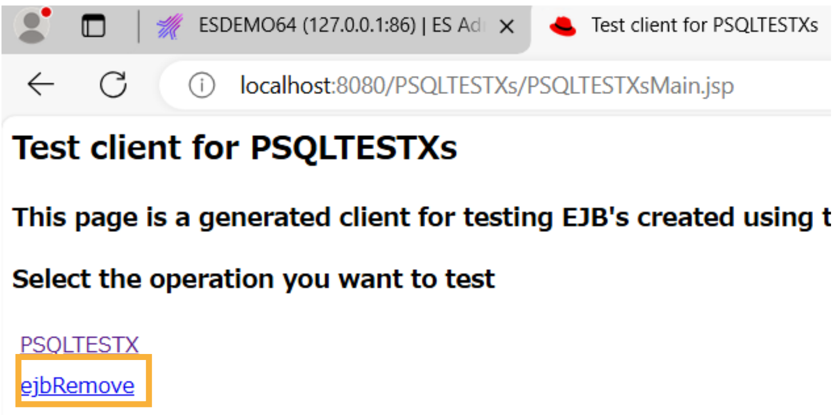
結果が表示され、1450.00 になっていることがわかります。



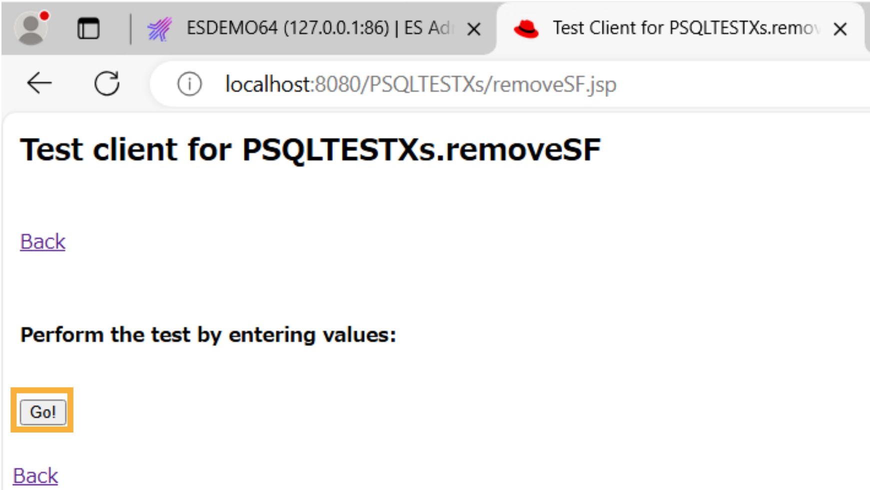
実際のデータベースのレコードも 1450.00 に更新されています。

③ EJB の削除処理を行います。

画面下部に表示されている [Back] リンクをクリックし、[ejbRemove] リンクをクリックします。

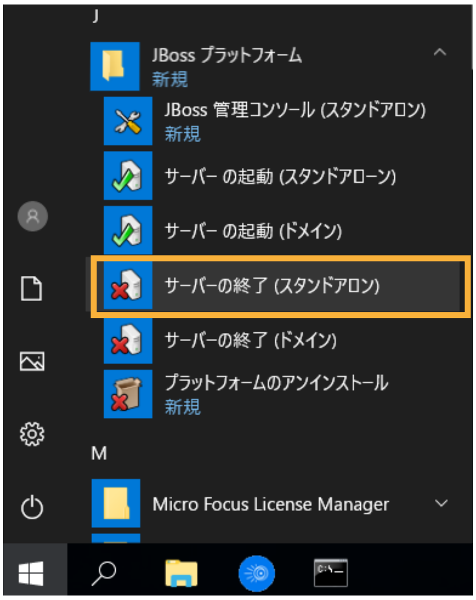


[Go!] をクリックし、ブラウザーを閉じます。



13) JBoss EAP, Enterprise Server インスタンスの停止

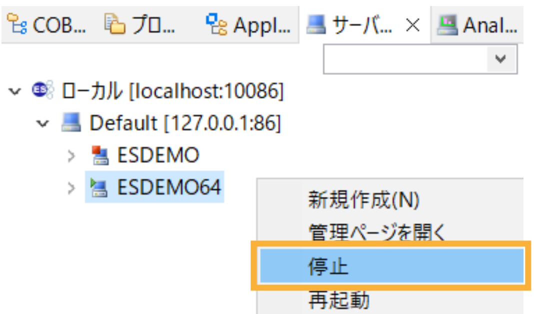
① Windows スタートメニューより、[JBoss プラットフォーム] > [サーバーの終了 (スタンドアロン)] をクリックします。



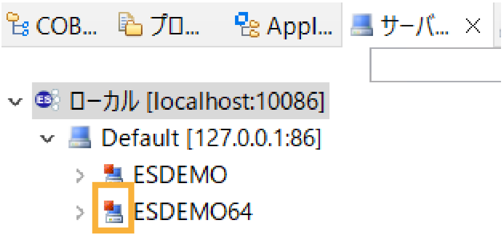
“続行するには何かキーを押してください . . .” の指示に従い、何かキーを押してプロンプト画面を終了します。

本作業により、JBoss EAP の起動時に使用していたプロンプト画面にも同様のメッセージが表示されますので、そちらも同様に終了します。

② Eclipse に戻り、[サーバー エクスプローラー] を開き、[ESDEMO64] を選択、マウスの右クリックでコンテクストメニューを開き、[停止] をクリックします。



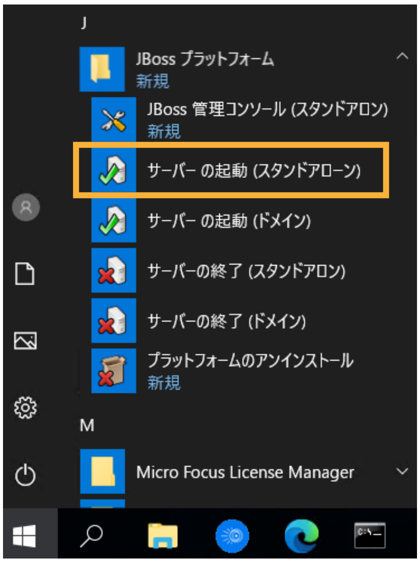
停止すると、赤いアイコンが表示されます。



3.6　ディプロイしたアプリケーションのデバッグ

1)　JBoss EAP の起動

①　Windows メニューより［JBoss プラットフォーム］＞［サーバーの起動（スタンドアローン］を選択します。

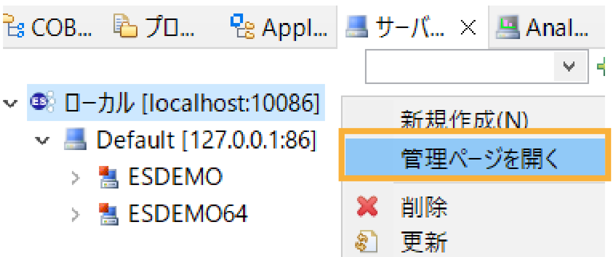


コンソールログからエラーがないことを確認してください。

2)　Enterprise Server インスタンスのデバッグ設定

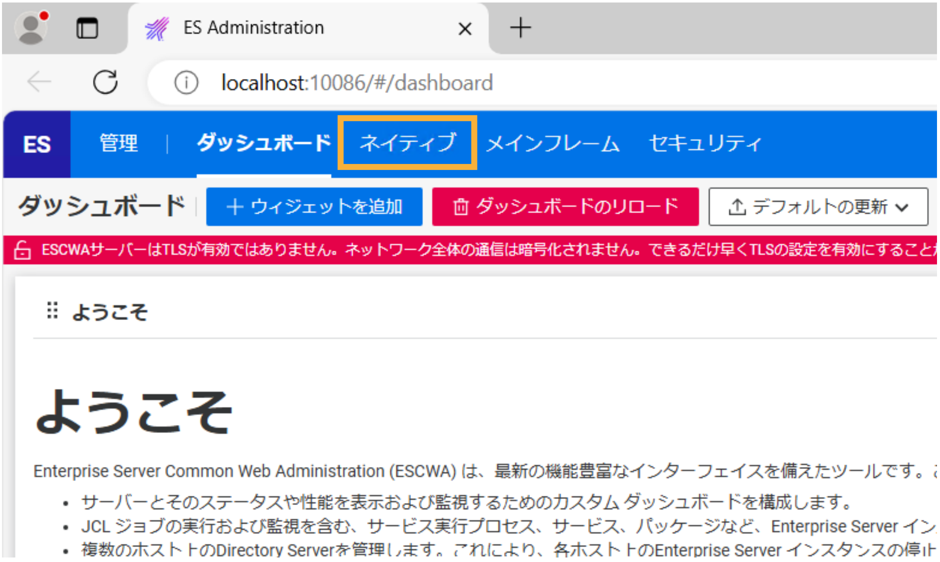
①　Eclipse に戻り、［サーバーエクスプローラー］を開きます。

②　［ローカル［localhost:10086］を選択したうえで、マウスの右クリックでコンテクストメニューを開き、［管理ページを開く］を選択します。

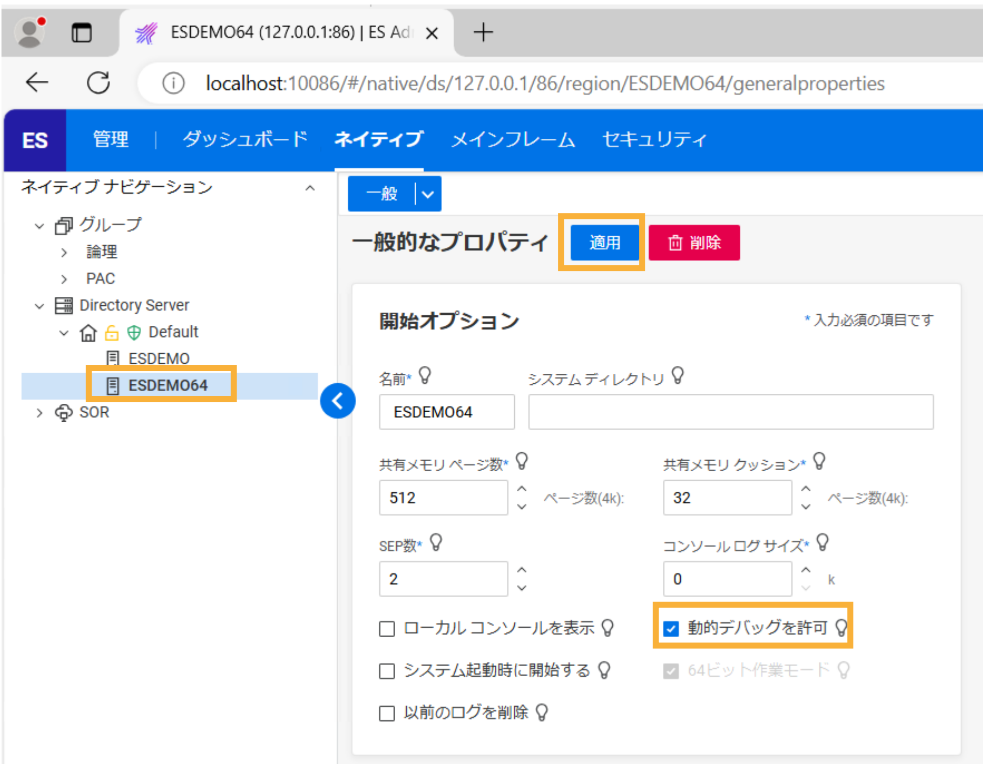


ブラウザーが開いたら、さきほどの認証情報を入力して、［ログオン］をクリックします。

ダッシュボード画面が表示されたら、［ネイティブ］をクリックします。

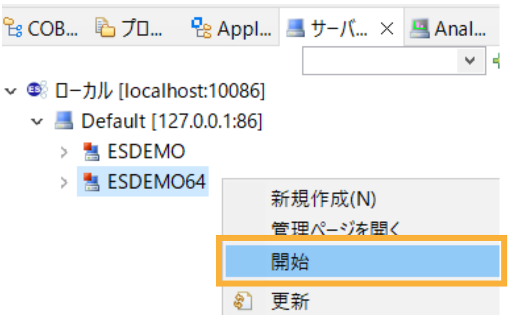


③　左側より、[Directory Server] ＞ [Default] ＞ [ESDEMO64] を選択し、[動的デバッグを許可] にチェックを行ったうえで、[適用] をクリックします。



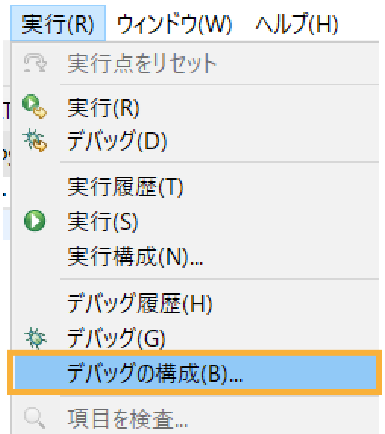
3)　Enterprise Server インスタンスの起動

①　[ESDEMO64] を選択したうえでマウスの右クリックでコンテクストメニューを開き、[開始] を選択します。

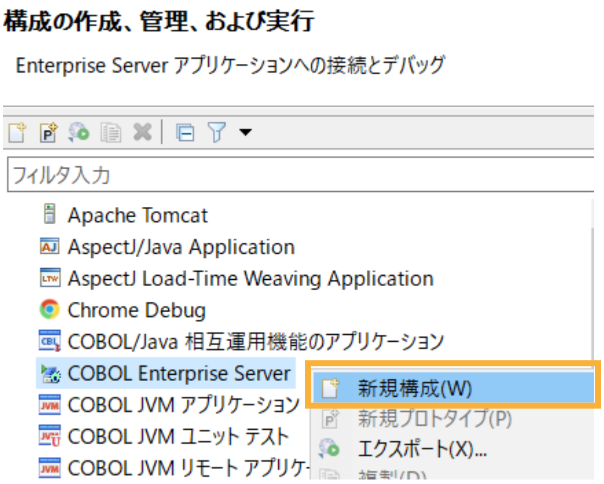


4) アプリケーションのデバッグ

① Eclipse に戻り、[実行(R)] > [デバッグの構成(B)] を選択します。



② 左側より、「COBOL Enterprise Server」上で右クリックし、コンテクストメニューから [新規構成(W)] を選択します。

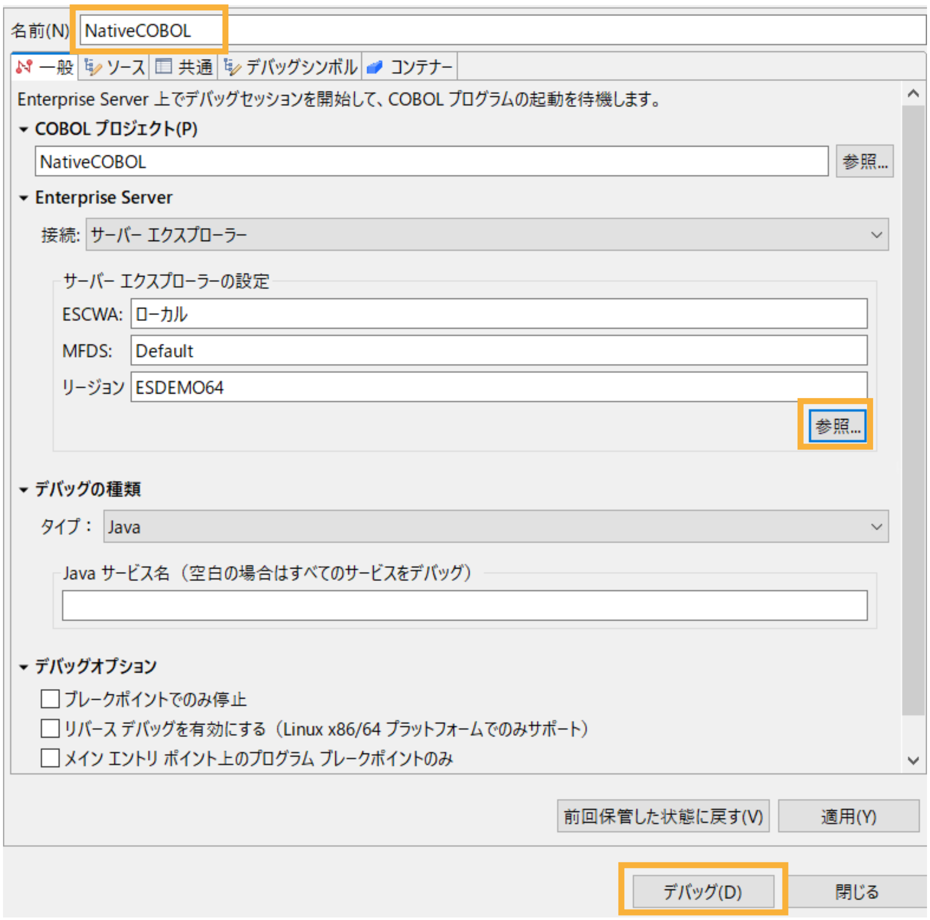


③ 以下の入力を行い、[デバッグ(D)] をクリックします。

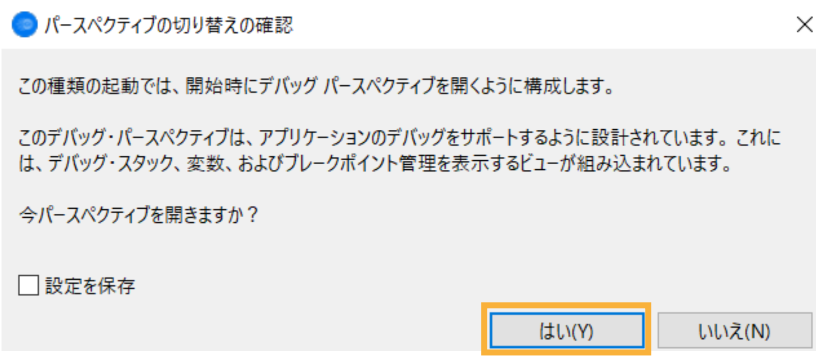
名前： “NativeCOBOL”

Enterprise Server： [参照] をクリックし、[ESDEMO64] を選択

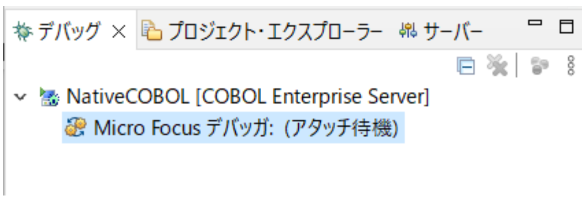
デバッグの種類 > タイプ： “Java”



以下の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。

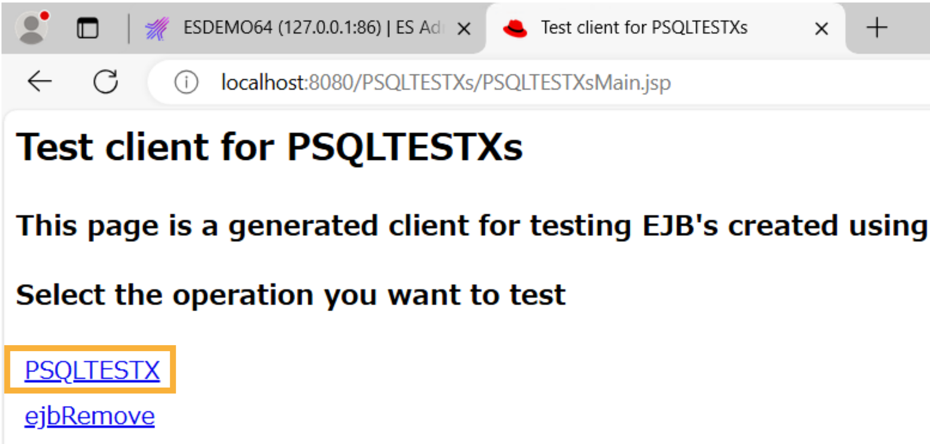


デバッガが以下のようにアタッチ待機になります。



④ ブラウザーを開き、以下のアドレスにアクセスし、[PSQLTESTX] リンクをクリックします。

http://localhost:8080/PSQLTESTXs/PSQLTESTXsMain.jsp



⑤　データを更新します。

現在、サンプルレコードの EMPNO: 7934 の情報は以下の通りです。

EMPNO	7934
ENAME	CMILLER
JOB	CLEARK
SAL	1450.00
DEPTNO	3

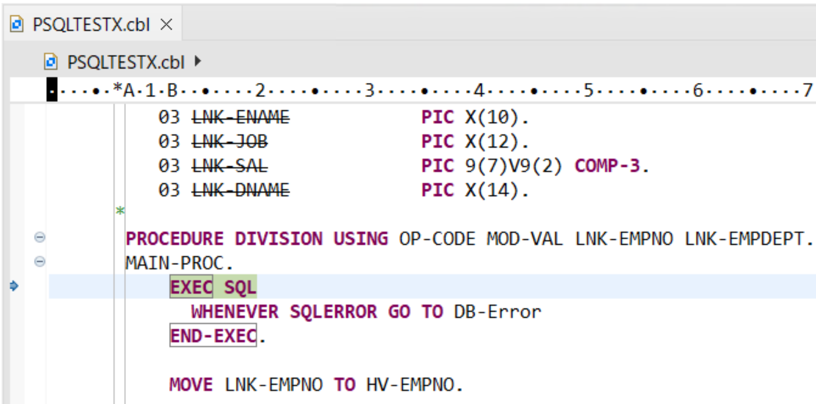
Web 画面で以下の項目を入力して、[Go!] をクリックします。

PSQLTESTX_OP_CODE_io: “I”

PSQLTESTX_MOD_VAL_io: “150”

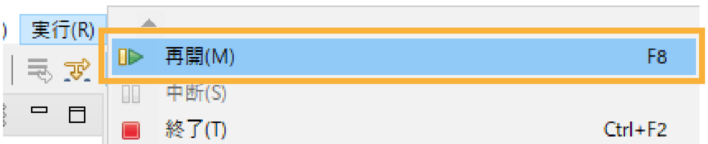
PSQLTESTX_LNK_EMPNO_io: “7934”

⑥　Eclipse IDE を見ると、デバッグパースペクティブ上で、PSQLTESTX.cbl で停止していることが分かります。



通常のコンソールアプリケーションと同様、項目値の確認や、ステップ実行などが行えます。

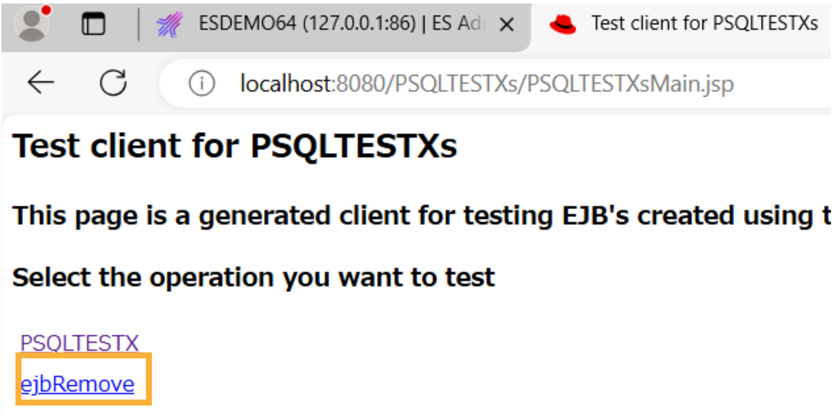
デバッグを終え、画面に応答結果を戻すためには、[実行(R)] > [再開(M)] を選択してください。



⑦ デバッグ作業を終了する際は、[実行(R)] メニューより [切断] を選択してください。

⑧ EJB の削除処理を行います。

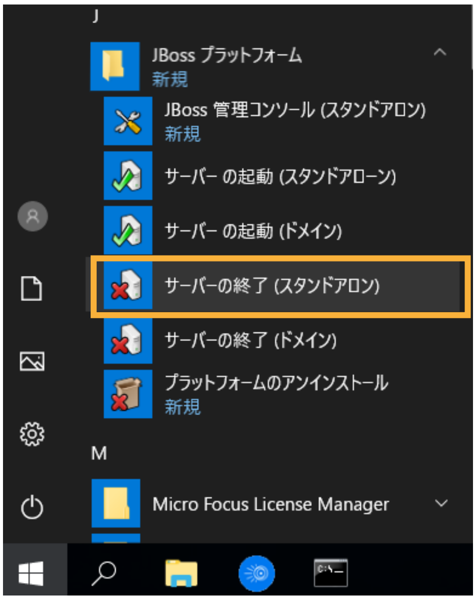
画面下部に表示されている [Back] リンクをクリックし、[ejbRemove] リンクをクリックします。



[Go!] をクリックし、ブラウザーを閉じます。

5) JBoss EAP, Enterprise Server インスタンスの停止

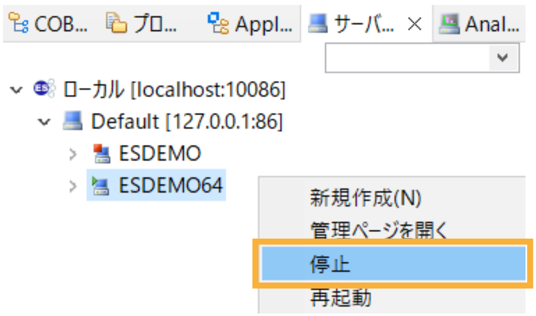
① Windows スタートメニューより、[JBoss プラットフォーム] > [サーバーの終了 (スタンドアロン)] をクリックします。



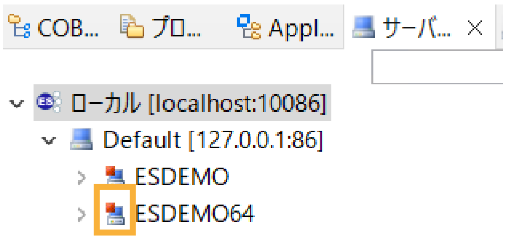
“続行するには何かキーを押してください . . .” の指示に従い、何かキーを押してプロンプト画面を終了します。

本作業により、JBoss EAP の起動時に使用していたプロンプト画面にも同様のメッセージが表示されますので、そちらも同様に終了します。

② Eclipse に戻り、[サーバー エクスプローラー] を開き、[ESDEMO64] を選択、マウスの右クリックでコンテクストメニューを開き、[停止] をクリックします。



停止すると、赤い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。