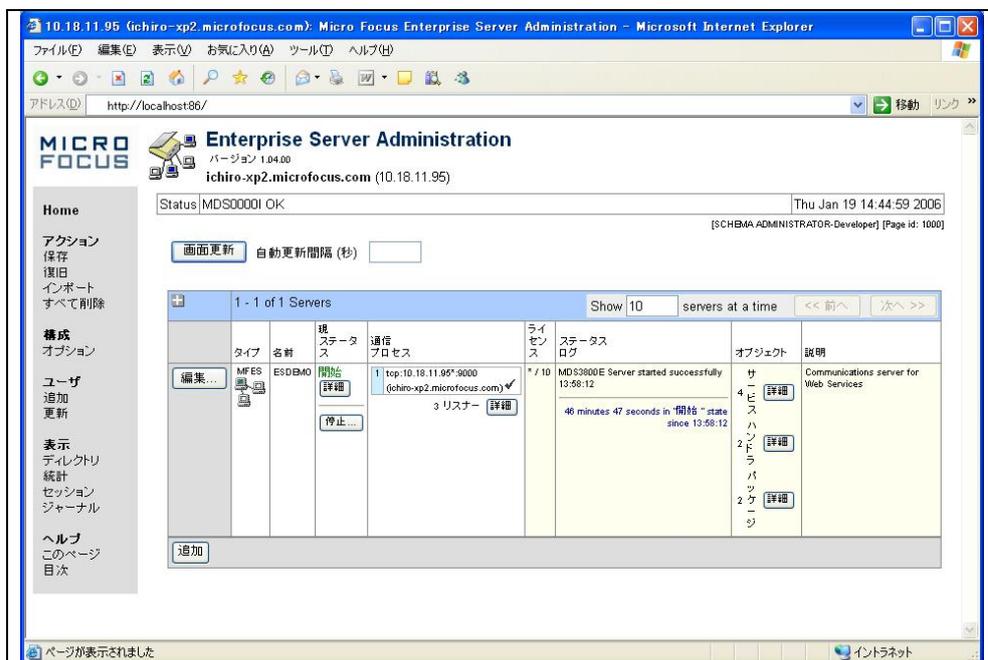


補足1. Oracle 照会プログラムのデプロイと、EJB経由の JCA呼び出し

- 1) Micro Focusリソースアダプタ mfcobol-notx.rar と mfcobol-notx-ds.xml を JBossインストールディレクトリ ¥server¥default¥deploy へコピーします。正しくインストールされると JBossのコンソール以下のメッセージが表示されます。

```
INFO [0] Bound connection factory for resource adapter for ConnectionManager
' jboss.jca:service=NoTxCM,name=eis/MFCobol_v1.0 to JNDI name
' java:/eis/MFCobol_v1.0'
```

- 2) Micro Focus Enterprise Server Admin から、出荷時設定の ESDEMOサーバを開始します。以下のよう
に開始状態となります。



- 3) Net Express のプロジェクトを作成し、以下の Oracle の Pro*COBOL による選択の COBOL プログラム: Sel.pco を用意します。このデモプログラムでアクセスするテーブル staff の 作成SQLは巻末の備考に記載させていただきます。

```
IDENTIFICATION DIVISION.
PROGRAM-ID. Sel.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
    EXEC SQL BEGIN DECLARE SECTION END-EXEC.
01 PASSWD          PIC X(20) VARYING.
01 STAFF-ID        PIC S9(4) COMP-3.
01 STAFF-NAME     PIC X(10).
01 STAFF-DEPT     PIC S9(4) COMP-3.
01 STAFF-SALARY   PIC S9(10)V9(2) COMP-3.
    EXEC SQL END DECLARE SECTION END-EXEC.
    EXEC SQL INCLUDE SQLCA END-EXEC.
```

```

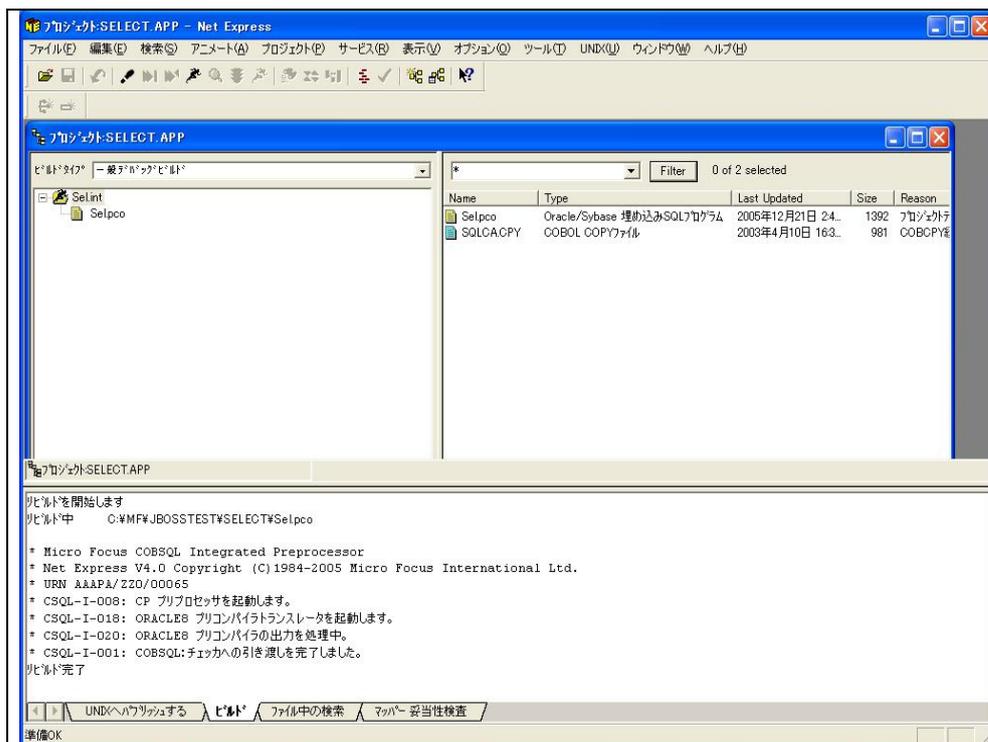
LINKAGE SECTION.
01 LK-STAFF-ID PIC x(4) comp-5.
01 LK-STAFF-NAME PIC X(10).
PROCEDURE DIVISION USING LK-STAFF-ID LK-STAFF-NAME.
1.
  MOVE "SCOTT/TIGER@orcl" TO PASSWD-ARR.
  MOVE 16 TO PASSWD-LEN.
  DISPLAY "CONNECT STEP" UPON CONSOLE.
  EXEC SQL
    CONNECT :PASSWD
  END-EXEC.
  MOVE LK-STAFF-ID TO STAFF-ID.
  EXEC SQL SELECT NAME, DEPT, SALARY
    INTO :STAFF-NAME
    FROM STAFF
    WHERE ID=:STAFF-ID
  END-EXEC.
  MOVE STAFF-NAME TO LK-STAFF-NAME.
  EXEC SQL COMMIT WORK RELEASE END-EXEC.
  EXIT PROGRAM.

```

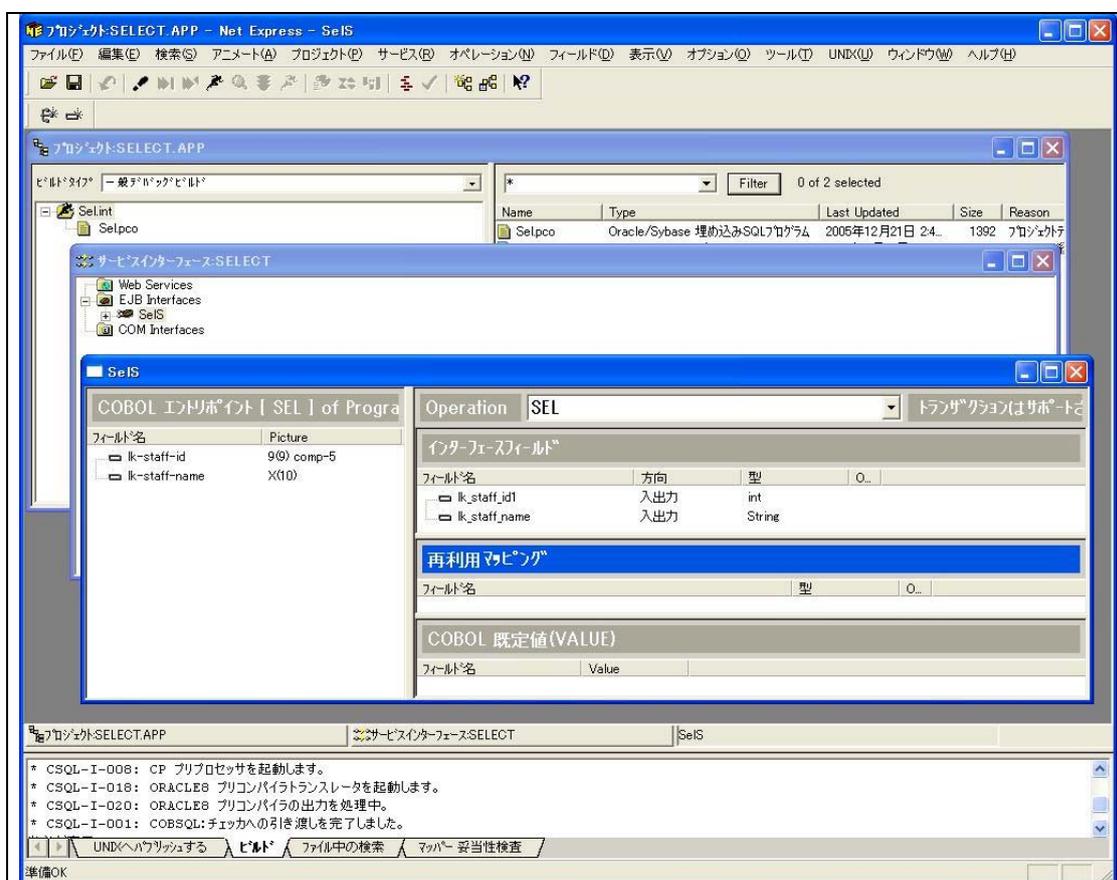
- 4) Net Express 付随の SQL プリプロセッサ COBSQL を使用して、このプログラムをコンパイルします。COBSQL は、Oracle 提供の Pro*COBOL を内部で使用してプリコンパイルしています。COBSQL を有効にするために以下の内容をプロジェクト指令に追加します。

```
p(cobsql) CSQL=ORACLE8 END-COBSQL mode=ANSI comp5=yes P(CP) ENDP
```

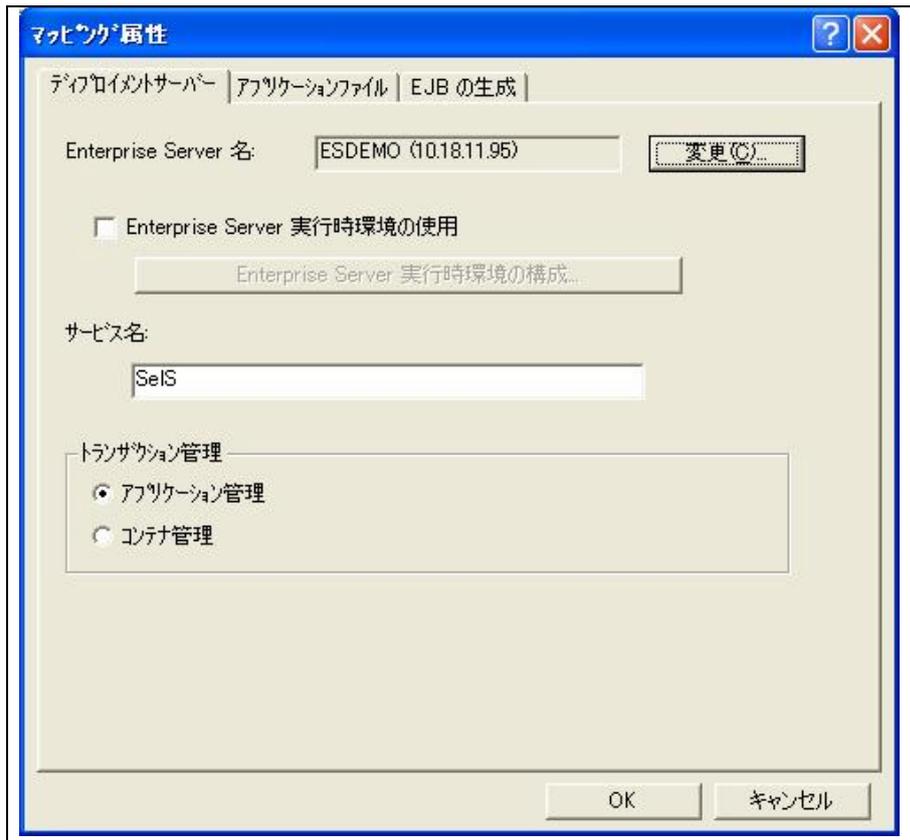
- 5) Net Express で、Sel.pco プログラムをコンパイルします。



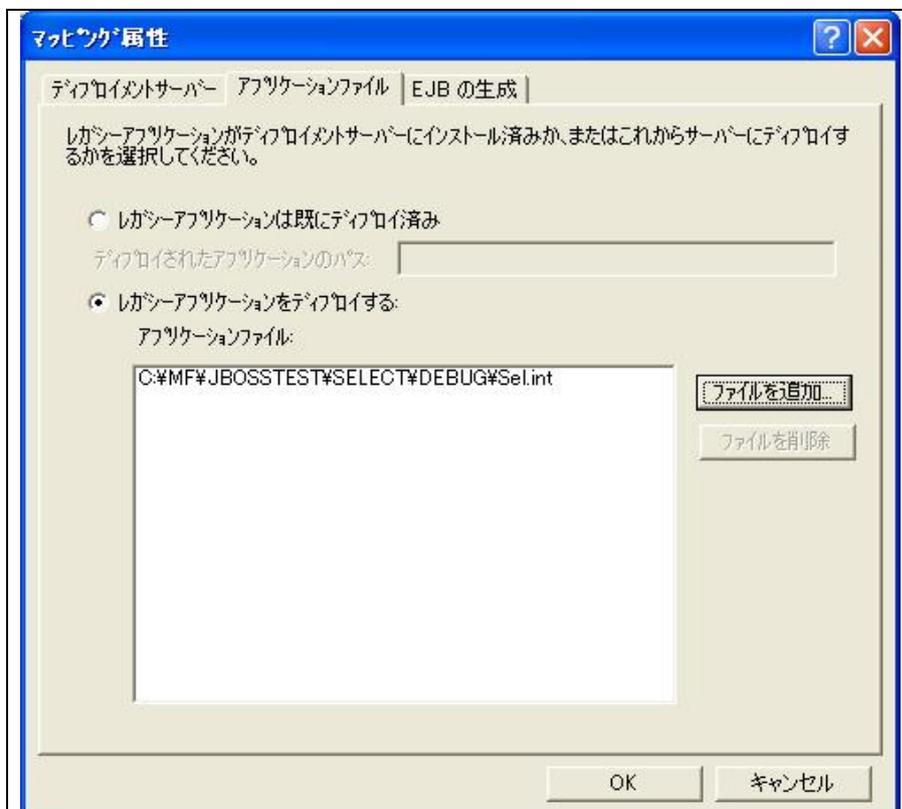
- 6) 同じく Net Express 環境下で、サービスインターフェイスを使用してデフォルトマッピングを作成します。
 - ① ファイル > 新規作成 をクリックします。
 - ② 「新規作成」ダイアログ ボックスで、「サービス インターフェイス」をクリックして、「OK」をクリックします。その結果、マッピング ウィザードが起動されます。
 - ③ ウィザードに従って、EJB インターフェイスを作成します。
 - ④ マッピングを選択では、「COBOL を EJB としてマップ」を選択します。
 - ⑤ COBOL インターフェイスマッピングでは、「現在の Net Express プロジェクトを使用する」を選択します。
 - ⑥ プログラムの選択で「Sel.pco」を選択します。
 - ⑦ サービスマッピング名に「SelS」をタイプします。
 - ⑧ 省略時マッピングの実行で「省略時マッピング」を選択します。
 - ⑨ 完了をクリックすると、Net Express によってインターフェイス マッパーが起動します。
 - ⑩ 今回は、特に変更しませんのでインターフェイス マッパーを終了します。



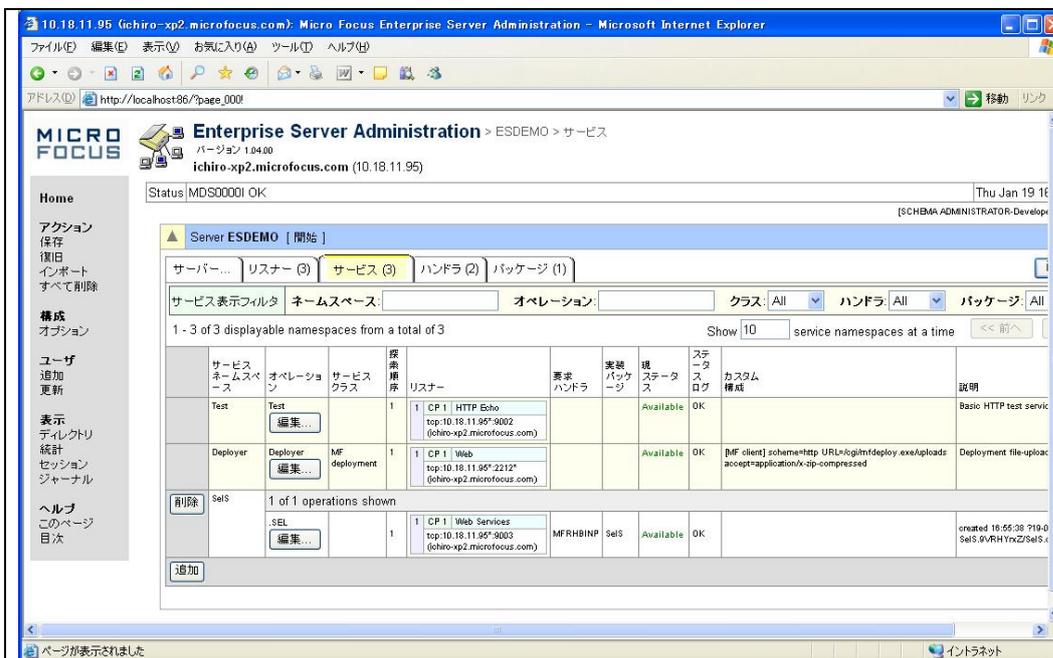
- 7) サービスインターフェイスより、サービスをデプロイする設定を行います。
 - ① サービスインターフェイスの「SelS」を右クリックし、「設定」を選択します。
 - ② 「デプロイメントサーバー」タブで、「変更」ボタンをクリックしサーバを選択します。
 - ③ サービス名には、「SelS」をタイプします。



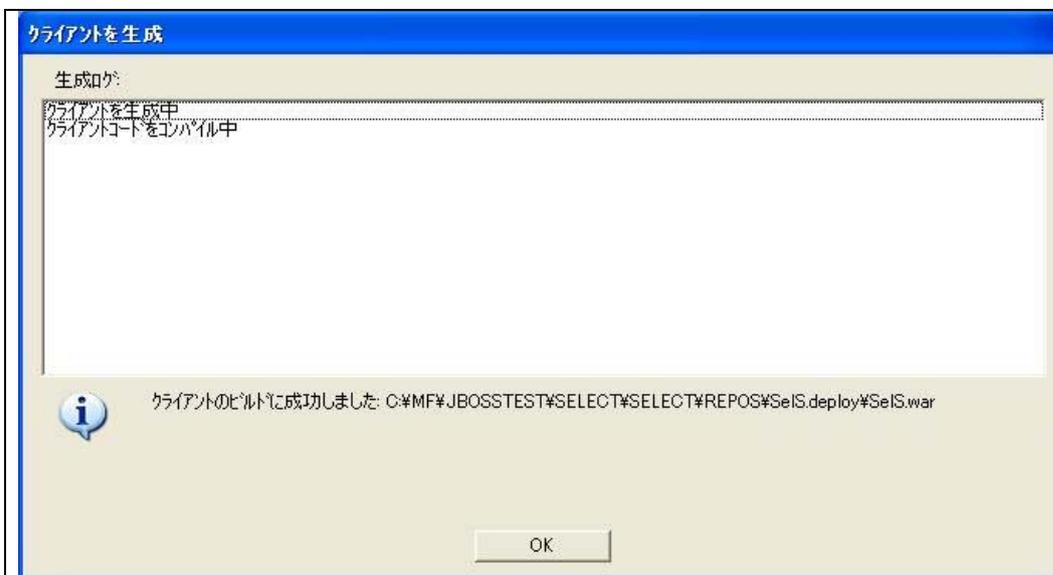
- ④ 「アプリケーションファイル」タブを選択で、「レガシーアプリケーションをデプロイする」を選択し、「ファイルを追加」ボタンをクリックし、「Sel.int」を選択します。



- 9) Micro Focus Enterprise Server Admin でサービスが正しくデプロイされていることを確認します。



- 10) サービスインターフェイスより、JBoss 向け JSP をインストールするための ear を自動生成します。サービスインターフェイスの「SelS」を右クリックし「クライアントを生成」を選択します。

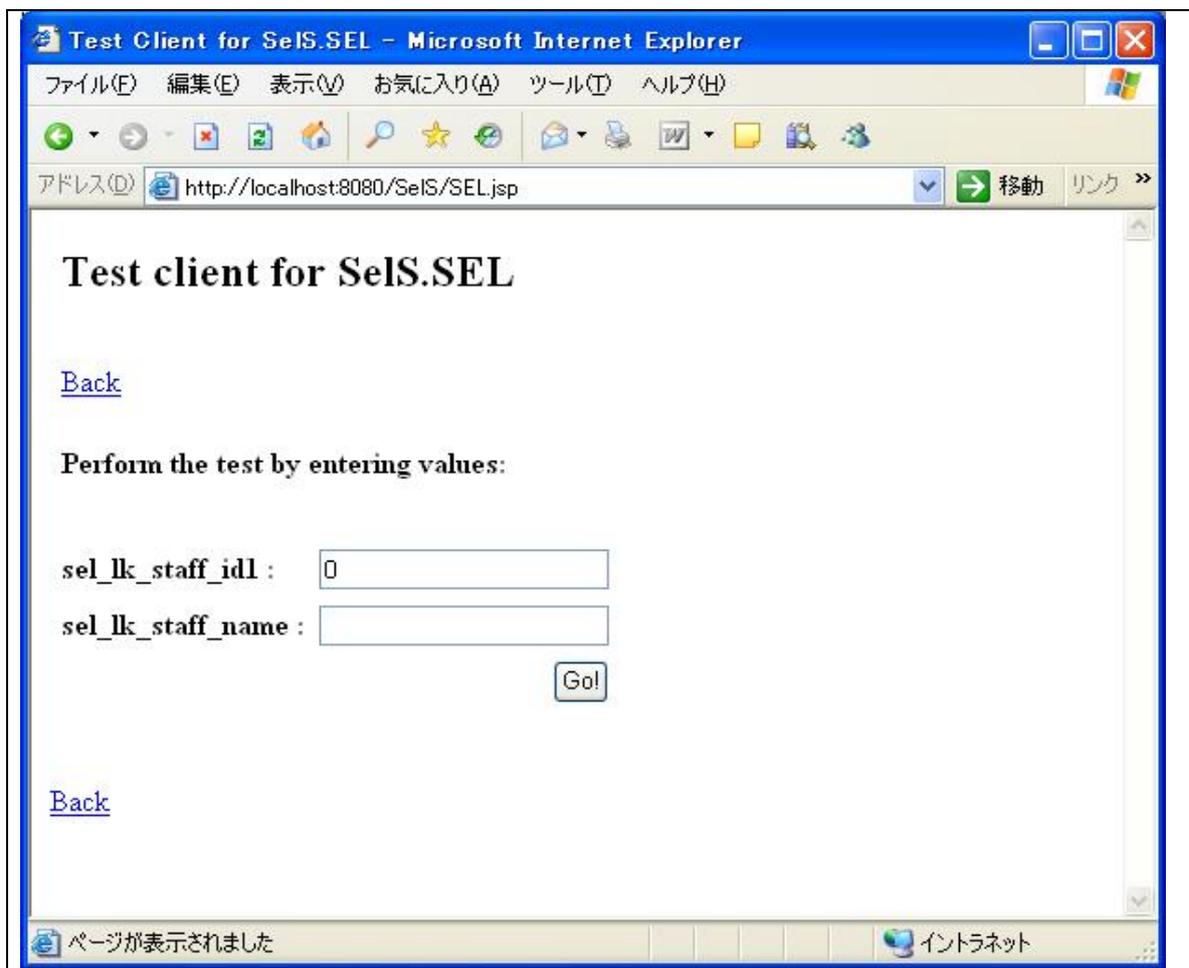


- 11) 生成された SelS.ear を JBoss インストールディレクトリ¥server¥default¥deploy へコピーします。以下のようように JBoss のコンソールに表示されインストールが完了したことを確認します。

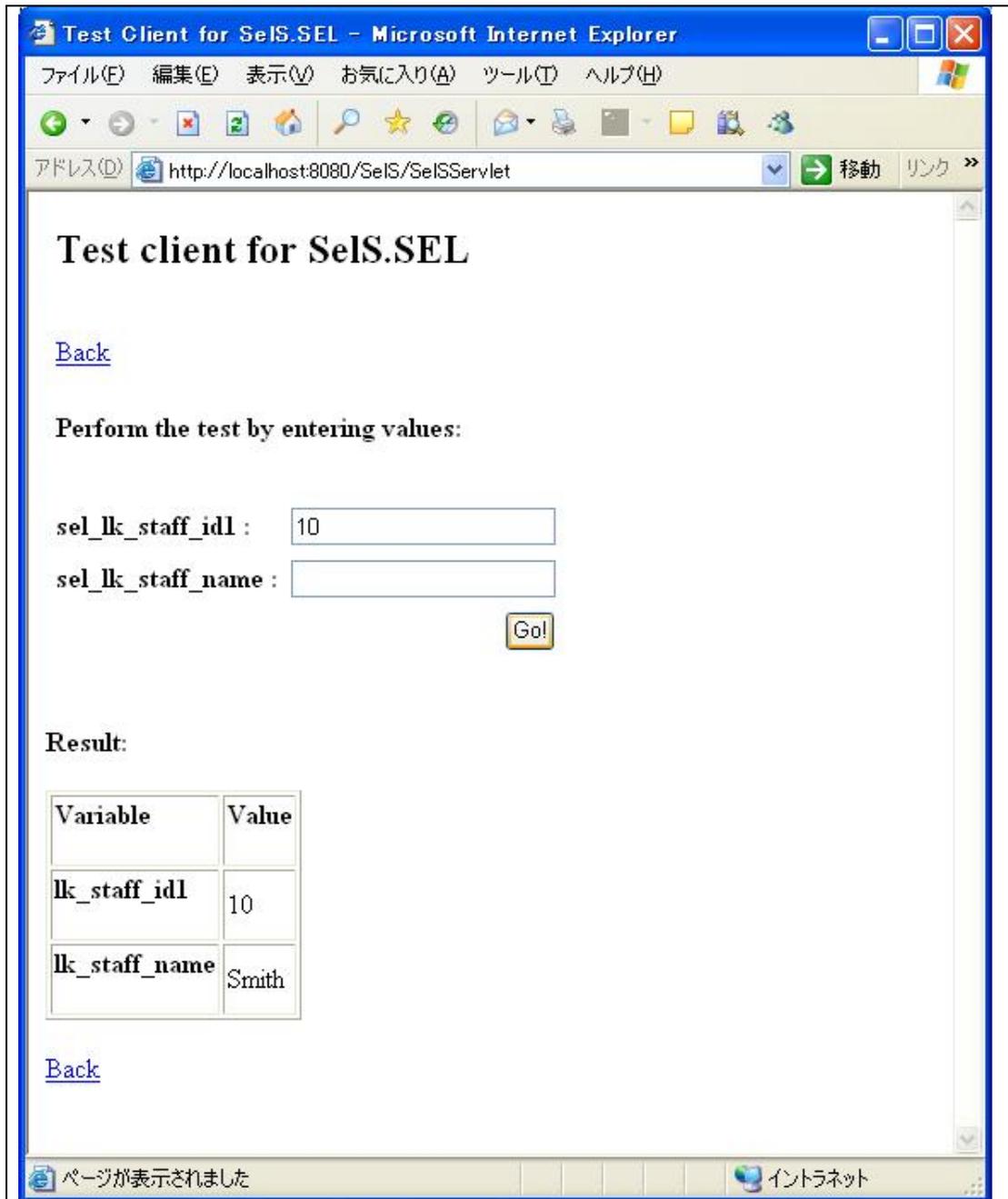
```
INFO [EARDeployer] Init J2EE application: file:/C:/jboss-3.2.7/server/default/
deploy/SelS.ear
INFO [EjbModule] Deploying SelSEJB
```

```
INFO [EJBDeployer] Deployed: file:/C:/jboss-3.2.7/server/default/tmp/deploy/
tmp10582SelS.ear-contents/SelS.jar
INFO [TomcatDeployer] deploy, ctxPath=/SelS,
warUrl=file:/C:/jboss-3.2.7/server/default
/tmp/deploy/tmp10582SelS.ear-contents/SelS.war/
INFO [EARDeployer] Started J2EE application: file:/C:/jboss-3.2.7/server/
default/deploy/SelS.ear
```

- 12) ブラウザから `http://localhost:8080/SelS/SelS.jsp` を開きます。以下のように自動生成された JSP クライアントが起動し、最初のテキストボックスに、Oracle の SAMPLE 中に存在するキー値を入力し、[Go!] をクリックします。



- 13) COBOL サービスが呼び出され、Oracle を検索したデータが JSP 画面上に表示されます。



補足2. Oracle 更新プログラムのディプロイと、EJB経由の JCA呼び出しにおけるコンテナ管理 トランザクション

- 1) JBossインストールディレクトリ¥server¥default¥deployから、インストール済みのMicro Focus リソースアダプタ mfcobol-notx.rar と mfcobol-notx-ds.xml を削除します。
- 2) 同じく Micro Focusリソースアダプタ mfcobol-xa.rar と mfcobol-xa-ds.xml を Server Expressのマニュアル記載の通りJBossインストールディレクトリ¥server¥default¥deployへコピーします。正しくインストールされるとJBossのコンソール以下のメッセージが表示されます。

```
INFO [0] Bound connection factory for resource adapter for ConnectionManager  
' jboss. jca:service=TxCM, name=eis/MFCobol_v1.0 to JNDI name ' java:/eis/MFCobol_v1.0'
```

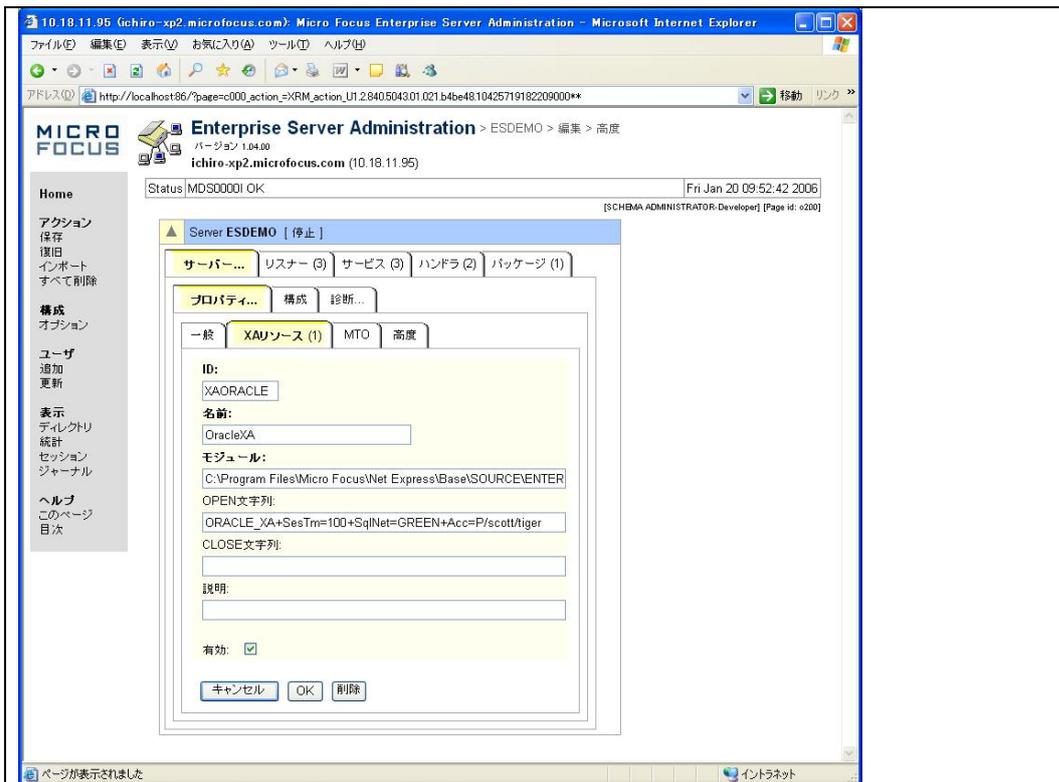
- 3) 新しいリソースアダプタを有効にするために、JBoss を再起動します。
- 4) Micro Focus Enterprise Server に、Oracle の XA スイッチモジュールを作成します。
 - ① Net Express コマンドプロンプトを開き、Net Express インストールディレクトリ ¥Base¥SOURCE¥ENTERPRISESERVER¥XA に移動します。
 - ② バッチファイル: BUILD.BAT を利用して、ESORA9XA.CBL をビルドしスイッチモジュール: ESORA9XA.dll を作成します。
 - ③ 詳細は、Net Express ヘルプ > Enterprise Server > 構成と管理 > 2:構成 > リソースマネージャ をご覧ください。

```
C:¥Micro Focus¥Net Express¥Base¥SOURCE¥ENTERPRISESERVER¥XA>build ora9  
Micro Focus Net Express - CBLLINK utility  
Version 4.0.38 Copyright (C) 1984-2003 Micro Focus International Ltd.  
  
Microsoft (R) Incremental Linker Version 6.00.8168  
Copyright (C) Microsoft Corp 1992-1998. All rights reserved.  
  
ESORA9XA.obj  
cbllids00000C4C.obj  
Creating library ESORA9XA.lib and object ESORA9XA.exp
```

- 5) Micro Focus Enterprise Server の ESDEMO サーバーにスイッチモジュールを XA リソースとして追加します。まず、ESDEMO を一旦停止し、ESDEMO の[編集]ボタンをクリックします。[XA リソース]タブの[追加]ボタンをクリックします。



- 6) Server Express のマニュアルの記述どおり、Oracle XA リソースマネージャのマニュアルに従って、XA リソース定義を入力します。
- ① ID、名前は、ご自由に設定できます。
 - ② モジュールは、作成した Oracle の XA スイッチモジュールをディレクトリも含めて指定します。
 - ③ OPEN 文字列は、Oracle の仕様にあわせて設定します。詳細は、Oracle9i アプリケーション開発者ガイド-基礎編 リリース 2(9.2)「Oracle XA でのトランザクション・モニターの操作」をご覧ください。 <http://otn.oracle.co.jp/document/products/oracle9i/920/develop.html#app>



- 7) Enterprise Server Admin 画面から ESDEMO をスタートします。コンソールログに以下のようなメッセージが出て、XA スイッチモジュールが正しく動いていることを確認してください。

```
ESDEMO CASX00015I XAORACLE XA interface initialized successfully
```

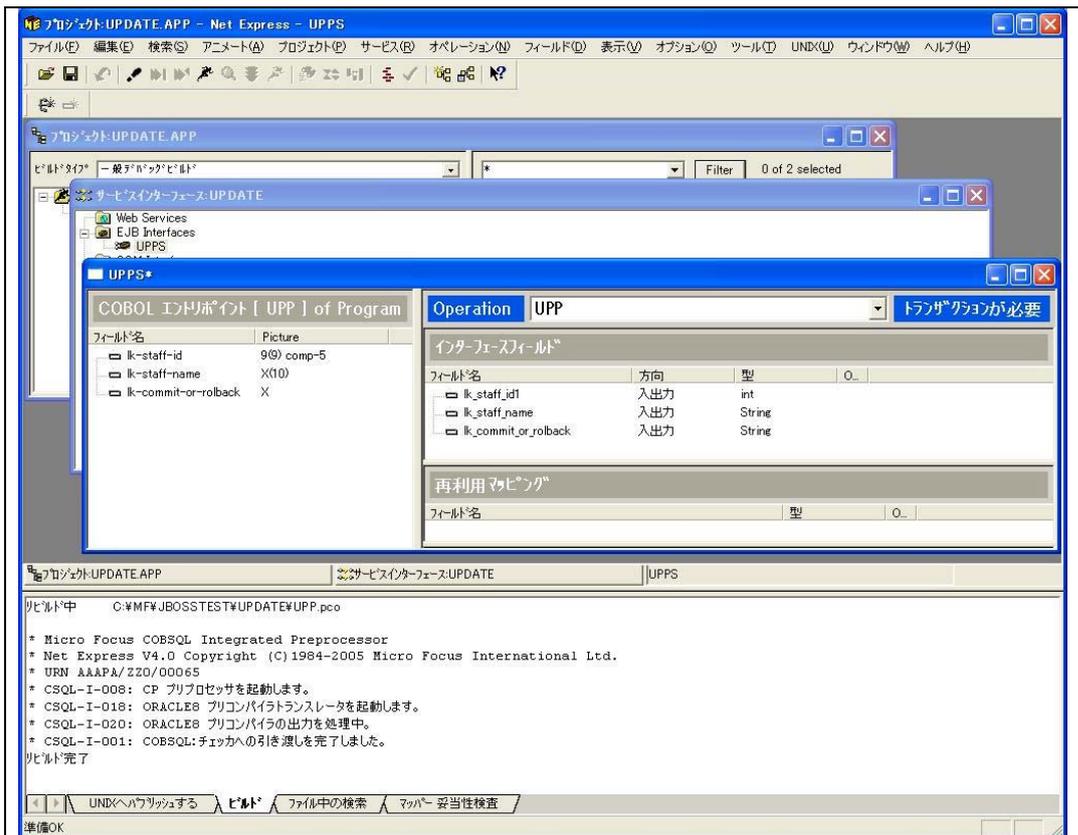
- 8) Net Express のプロジェクトを作成し、以下の Oracle の Pro*COBOL によるテーブル更新の COBOL プログラム: UPP.pco を用意します。

```
IDENTIFICATION DIVISION.
PROGRAM-ID. UPP.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
    EXEC SQL BEGIN DECLARE SECTION END-EXEC.
01 PASSWD          PIC X(20) VARYING.
01 STAFF-ID PIC S9(4) COMP-3.
01 STAFF-NAME PIC X(10).
    EXEC SQL END DECLARE SECTION END-EXEC.
    EXEC SQL INCLUDE SQLCA END-EXEC.
01 TABLE-ITEM PIC X OCCURS 10 TIMES INDEXED BY IDX.
LINKAGE SECTION.
01 LK-STAFF-ID PIC X(4) comp-5.
01 LK-STAFF-NAME PIC X(10).
01 LK-Commit-Or-Rolback PIC X.
PROCEDURE DIVISION USING LK-STAFF-ID LK-STAFF-NAME
    LK-Commit-Or-Rolback.
1.
    MOVE LK-STAFF-ID TO STAFF-ID.
    MOVE LK-STAFF-NAME TO STAFF-NAME.
    EXEC SQL UPDATE STAFF SET NAME=:STAFF-NAME
        WHERE ID=:STAFF-ID
    END-EXEC.
    DISPLAY "UPDATE = " SQLCODE UPON CONSOLE.
    IF LK-Commit-Or-Rolback = 'R'
        SET IDX TO 11
        MOVE SPACE TO TABLE-ITEM(IDX)
    END-IF.
EXIT PROGRAM.
```

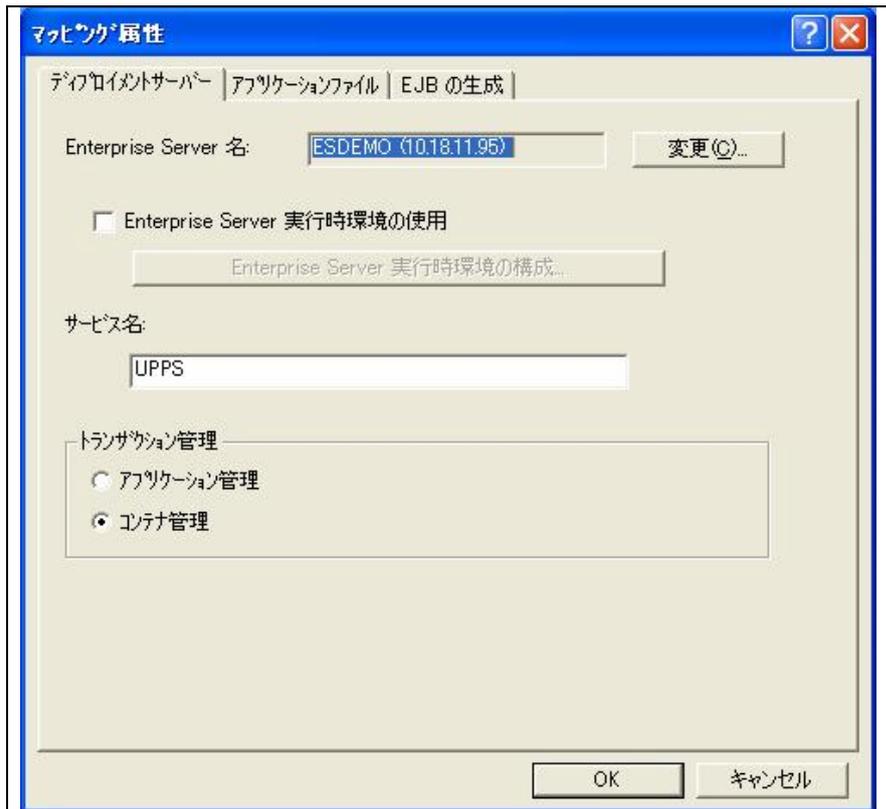
コンテナ管理トランザクションとしてデプロイする COBOL サービスの場合、データベースへの接続は Enterprise Server 側で行われますので、このプログラムには CONNECT 文が無いことに注意してください。プログラムは、第1 パラメタで渡されたキー値のレコードの NAME カラムを、第2 パラメタで渡された値で UPDATE します。第3 パラメタの値が 'R' の場合は、その後の意図的にテーブル項目の添え字参照範囲の例外を発生しますので、この更新トランザクションは Enterprise Server によって自動的に ROLLBACK されます。そうでなければ自動的に COMMIT されます。

- 9) このプログラムを Sel.pco の時と同様の手順でコンパイルします。
- 10) 同じく Net Express 環境下で、サービスインターフェイスを使用してデフォルトマッピングを作成します。
- ① ファイル > 新規作成 をクリックします。
 - ② 「新規作成」ダイアログ ボックスで、「サービス インターフェイス」をクリックして、「OK」をクリックします。その結果、マッピング ウィザードが起動されます。
 - ③ ウィザードに従って、EJB インターフェイスを作成します。
 - ④ マッピングを選択では、「COBOL を EJB としてマップ」を選択します。

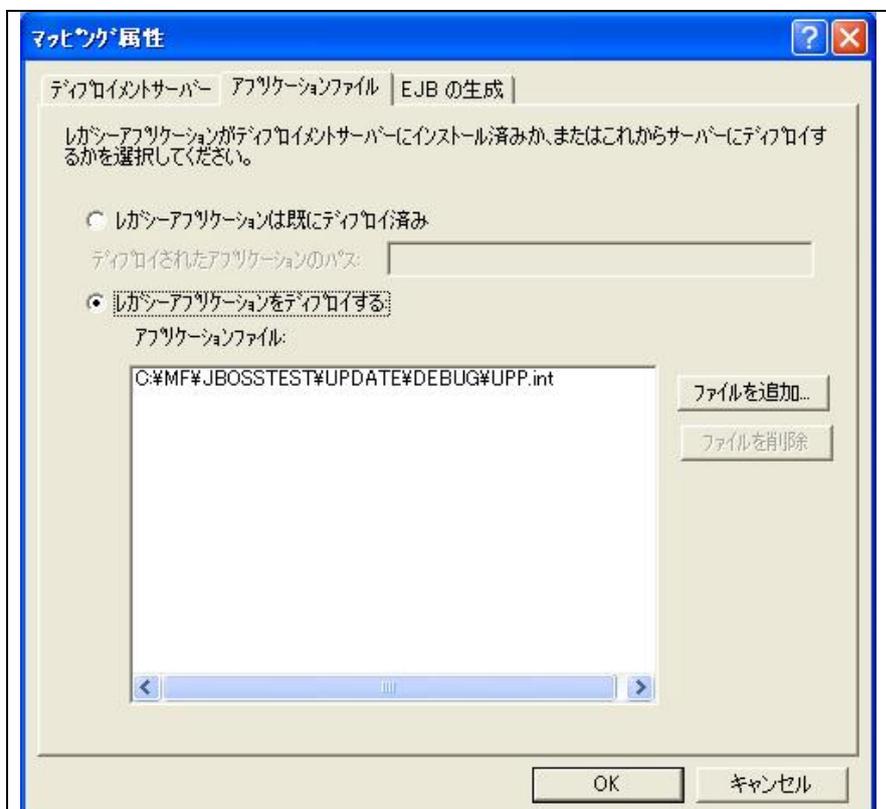
- ⑤ COBOL インターフェイスマッピングでは、「現在の Net Express プロジェクトを使用する」を選択します。
- ⑥ プログラムの選択で「UPP.pco」を選択します。
- ⑦ サービスマッピング名に「UPPS」をタイプします。
- ⑧ 省略時マッピングの実行で「省略時マッピング」を選択します。
- ⑨ 完了をクリックすると、Net Express によってインターフェイス マッパーが起動します。
- ⑩ 右上の「トランザクションはサポートされない」と右クリックし、「プロパティ」を選択します。
- ⑪ オペレーションプロパティダイアログが起動されます。トランザクションタグのコンテナ管理属性で「トランザクションが必要」を選択します。
- ⑫ インターフェイス マッパーを終了します。



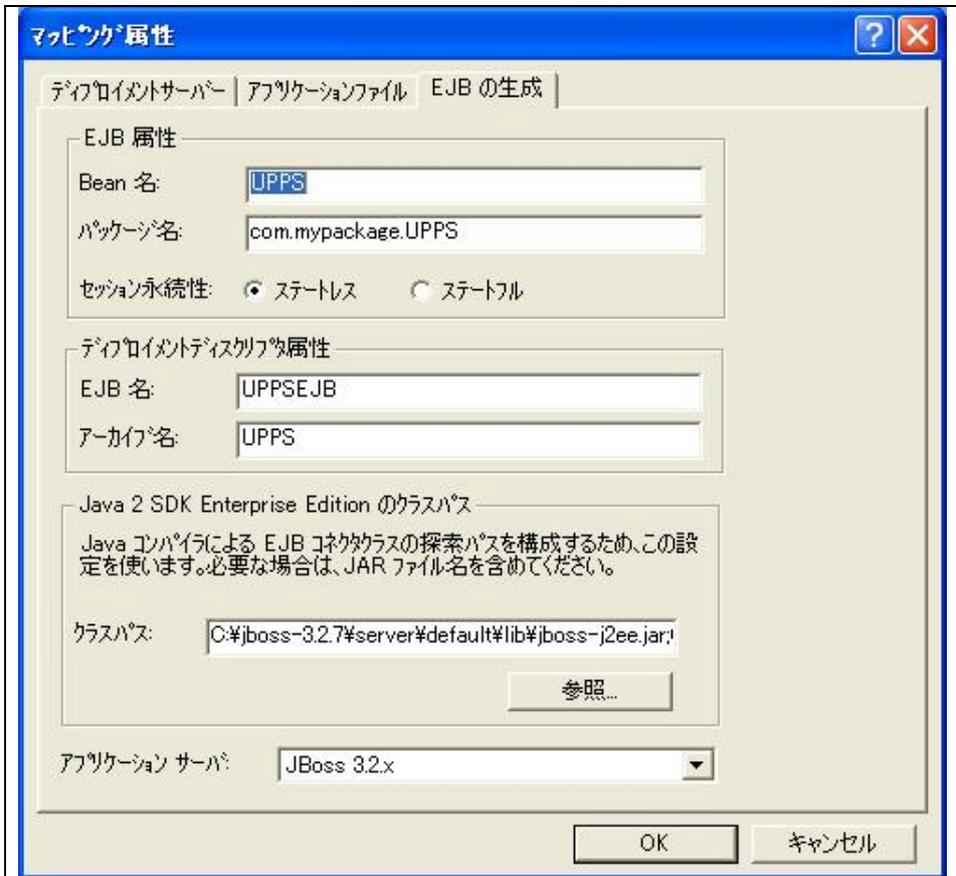
- 11) サービスインターフェイスより、サービスをデプロイする設定を行います。
 - ① サービスインターフェイスの「UPPS」を右クリックし、「設定」を選択します。
 - ② 「デプロイメントサーバー」タブで、「変更」ボタンをクリックしサーバを選択します。
 - ③ サービス名には、「UPPS」をタイプします。
 - ④ トランザクション管理で、「コンテナ管理」を選択します。



- ⑤ 「アプリケーションファイル」タブを選択で、「レガシーアプリケーションをデプロイする」を選択し、「ファイルを追加」ボタンをクリックし、「Sel.int」を選択します。



- ⑥ 「EJB の生成」タグで、JAVA 2 SDK Enterprise Edition のクラスパスに JBoss 付随の jboss-j2ee.jar と servlet-api.jar を指定します。
- ⑦ アプリケーションサーバに「JBoss 3.2.X」を選択します。

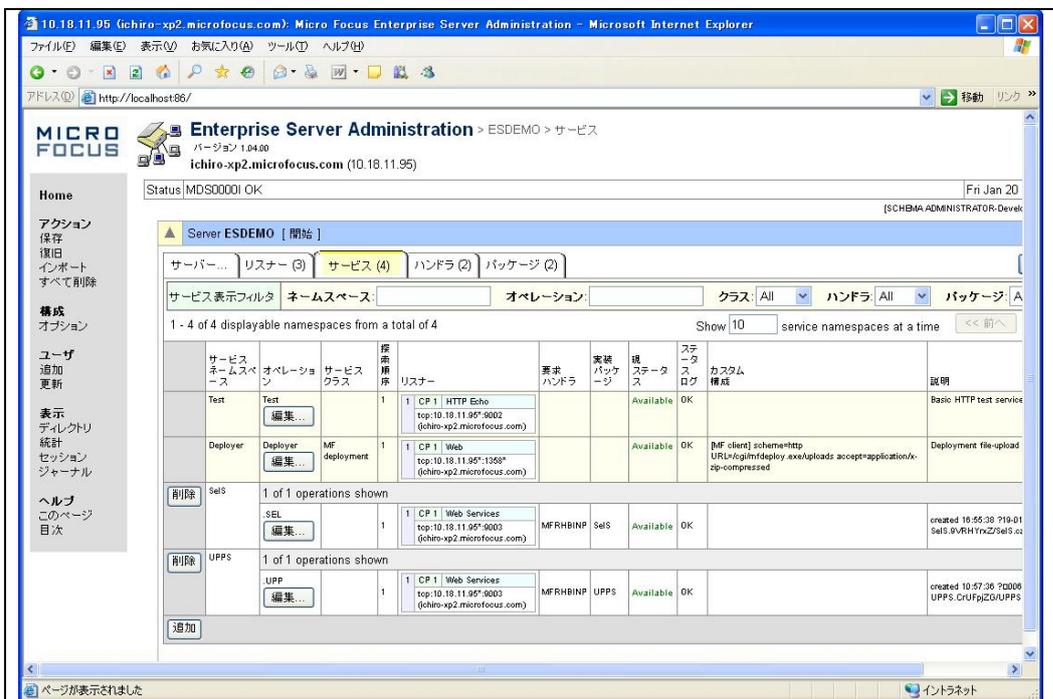


- ⑧ 「OK」をクリックして設定を保存します。

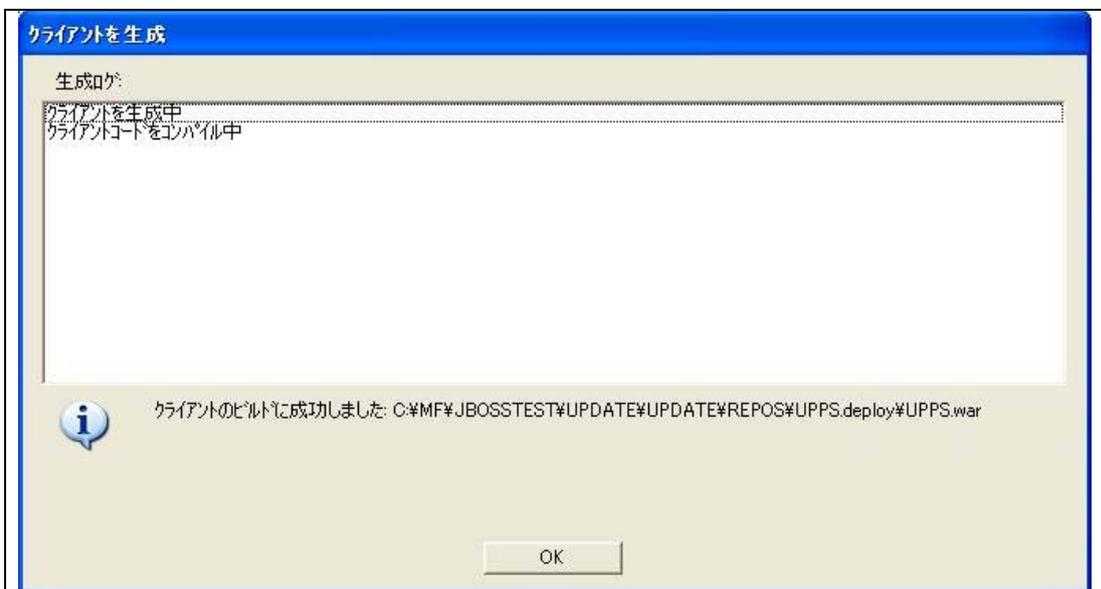
- 12) サービスインターフェイスより、サービスのデプロイを行います。
サービスインターフェイスの「UPPS」を右クリックし「デプロイ」を選択します。



- 13) Micro Focus Enterprise Server Admin でサービスが正しくデプロイされていることを確認します。



- 14) サービスインターフェイスより、JBoss 向け JSP をインストールするための ear を自動生成します。サービスインターフェイスの「UPPS」を右クリックし「クライアントを生成」を選択します。



- 15) 生成された UPPS.ear JBoss インストールディレクトリ%server%default%deploy へコピーします。以下のように JBoss のコンソールに表示されインストールが完了したことを確認します。

```
INFO [EARDeployer] Init J2EE application: file:/C:/jboss-3.2.7/server/default/
deploy/UPPS.ear
```

```

INFO [EjbModule] Deploying UPPSEJB
INFO [EJBDeployer] Deployed: file:/C:/jboss-3.2.7/server/default/tmp/deploy/
tmp60408UPPS.ear-contents/UPPS.jar
INFO [TomcatDeployer] deploy, ctxPath=/UPPS, warUrl=file:/C:/jboss-3.2.7/server/
default/tmp/deploy/tmp60408UPPS.ear-contents/UPPS.war/
INFO [EARDeployer] Started J2EE application: file:/C:/jboss-3.2.7/server/
default/deploy/UPPS.ear

```

これでテストプログラムの実行準備が整いました、staff テーブルの ID = 10 のレコードを使用してトランザクションの動作検証を行います。Oracle の SQLPLUS から、以下のように現在のレコード内容を確認します。

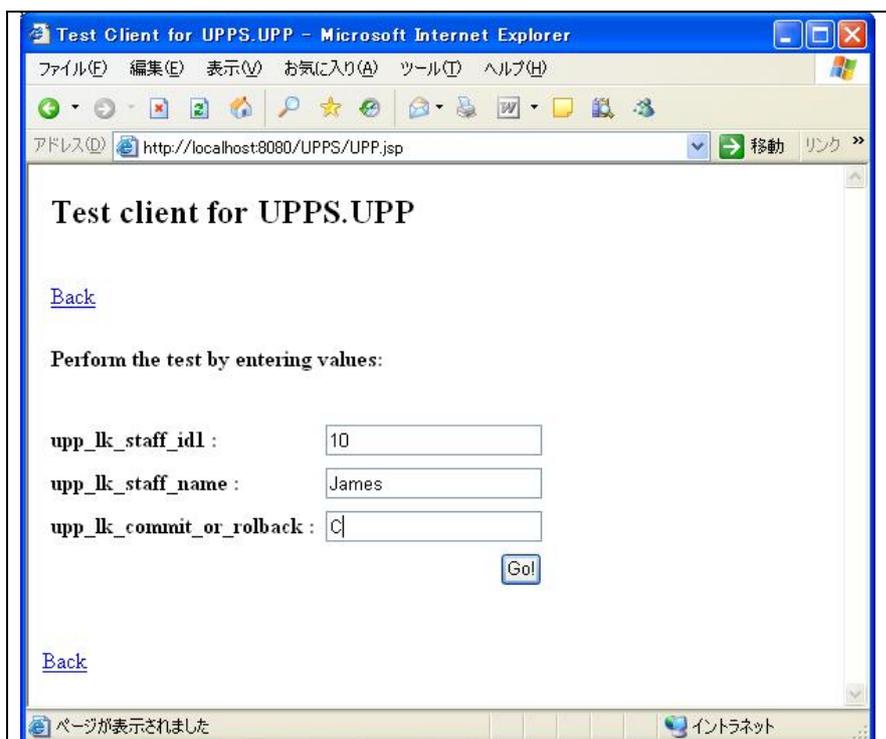
```
SQL> select * from staff;
```

ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
10	Smith	333	Mgr	7	18357.5	
20	TAKESI	333	Sales	8	18171.25	612.45
30	Marengi	38	Mgr	5	17506.75	

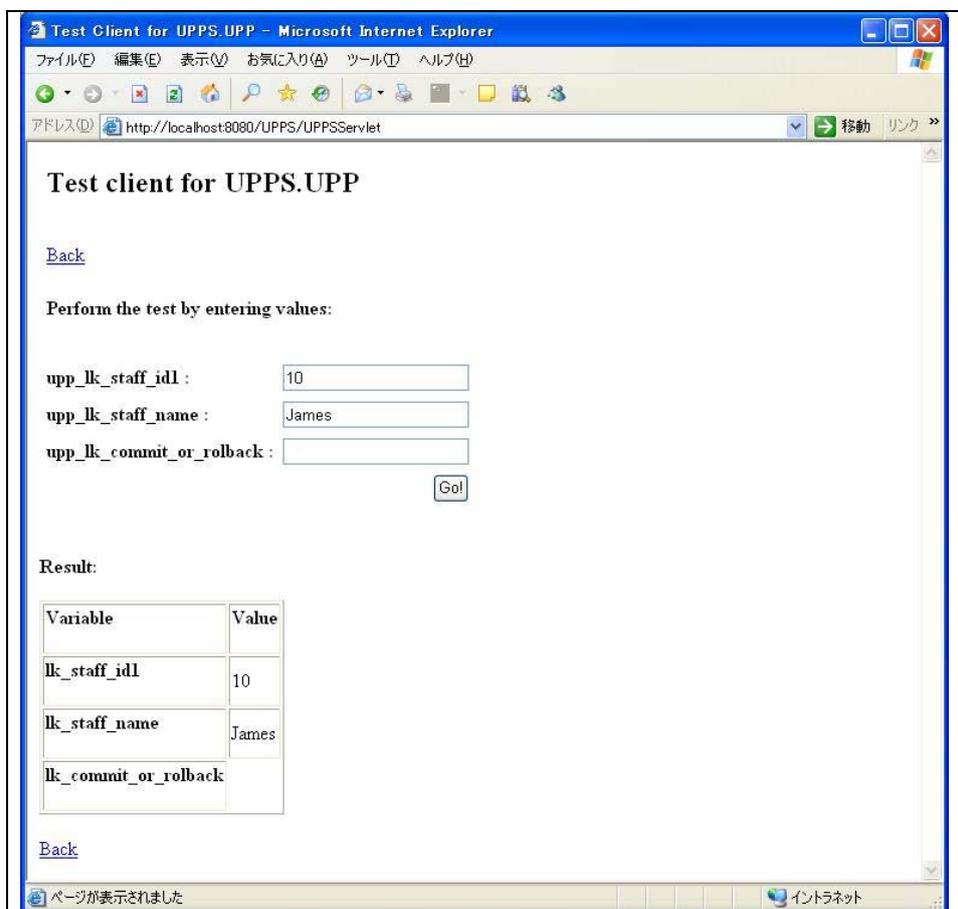
```
SQL>
```

現在は Smith さんの名前が格納されています。

- ブラウザから、`http://localhost:8080/UPPS/UPPS.jsp` を開き、最初のテキストボックスに ID として “10”、2 番目のテキストボックスに更新した名前 James を入力します。
3 番目のテキストボックスには、コミットさせることを示す “C” を入力す、[Go] をクリックします。



- 17) COBOL サービスが呼び出され、Oracle を更新したデータが JSP 画面上に表示されます。



- 18) ここで SQLPLUS から、staff テーブルの更新内容を確認します。

```
SQL> select * from staff;
```

ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
10	James	333	Mgr	7	18357.5	
20	TAKESI	333	Sales	8	18171.25	612.45
30	Marenghi	38	Mgr	5	17506.75	

```
SQL>
```

更新が COMMIT され、先ほど入力した名前に更新されていることが確認されました。

- 19) 再度、同じテストプログラムを起動し、今度は名前を最初から格納されていた“Smith”と入力し、3 番目のテキストボックスには、ロールバックさせることを示す“R”を入力し、[Go]をクリックします。COBOL のサービスで意図的なアプリケーション例外が発生し、ES コンソールに下記のようにエラーが帰ります。

```
ESDEMO CASKC0027E Error executing service 'UPPS.UPP'
Execution error : file 'C:\PROGRAM FILES\MICRO FOCUS\NET
```

```
EXPRESS¥BASE¥DEPLOY¥UPPS. CRUFPJZG¥UPP. int'
error code: 153, pc=1BF, call=1, seg=0
153 Subscript out of range 11:57:26
```

UPP.pco の下記の箇所で例外が発生した旨のエラーメッセージが表示されていることが分かります。

```
IF LK-Commit-Or-Rolback = 'R'
    SET IDX TO 11
    MOVE SPACE TO TABLE-ITEM(IDX)
END-IF
```

20) ここで、再び SQLPLUS で staff テーブルを参照します。

```
SQL> select * from staff;
```

ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
10	James	333	Mgr	7	18357.5	
20	TAKESI	333	Sales	8	18171.25	612.45
30	Marenghi	38	Mgr	5	17506.75	

```
SQL>
```

21) 更新トランザクションが自動的にロールバックされて、以前の値がそのまま残っていることが確認できました。

備考: Oracle 上で使用するテーブル staff の SQL 文

```
SQL> CREATE TABLE STAFF (ID      NUMBER(4) NOT NULL,
2      NAME  VARCHAR2(10),
3      DEPT  NUMBER(4),
4      JOB   VARCHAR2(10),
5      YEARS NUMBER(4),
6      SALARY NUMBER(10,2),
7      COMM  NUMBER(10,2));
```

表が作成されました。

```
SQL> DESC STAFF;
```

名前	NULL?	型
ID	NOT NULL	NUMBER(4)
NAME		VARCHAR2(10)
DEPT		NUMBER(4)
JOB		VARCHAR2(10)
YEARS		NUMBER(4)
SALARY		NUMBER(10,2)

COMM

NUMBER(10,2)

```
SQL> INSERT INTO STAFF VALUES (10, 'Smith', 333, 'Mgr', 7, 18357.50, Null);
```

1行が作成されました。

```
SQL> INSERT INTO STAFF VALUES (20, 'TAKESI', 333, 'Sales', 8, 18171.25, 612.45);
```

1行が作成されました。

```
SQL> INSERT INTO STAFF VALUES (30, 'Marenghi', 38, 'Mgr', 5, 17506.75, Null);
```

1行が作成されました。

```
SQL> COMMIT;
```

コミットが完了しました。

以上