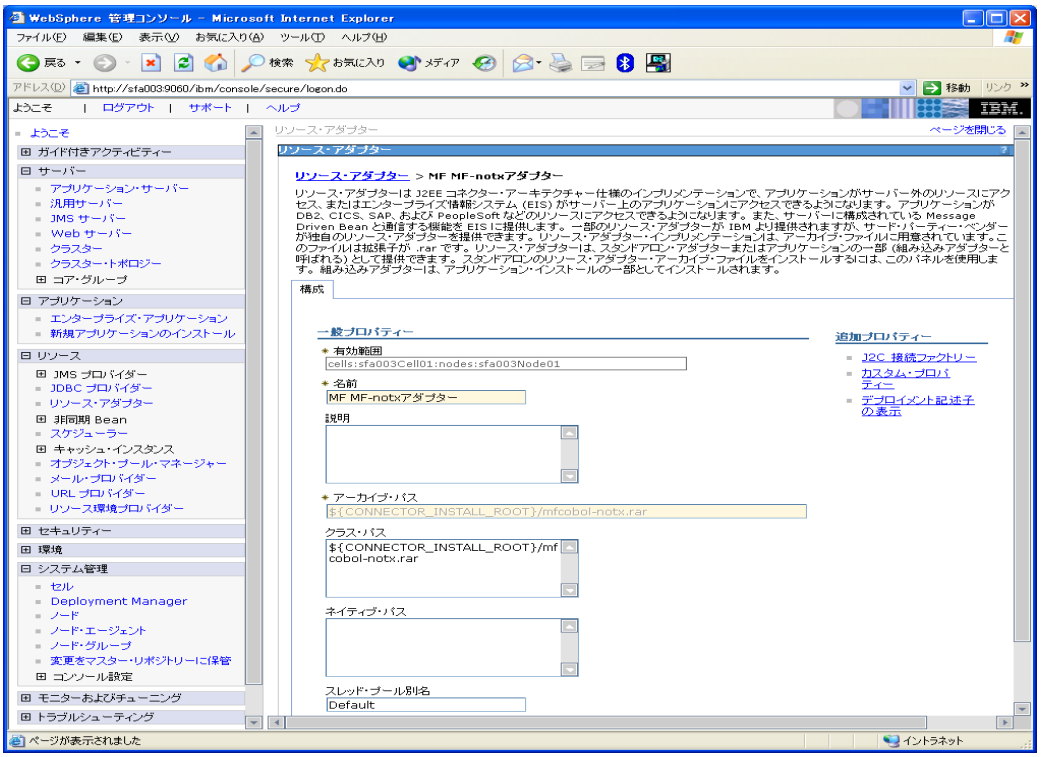
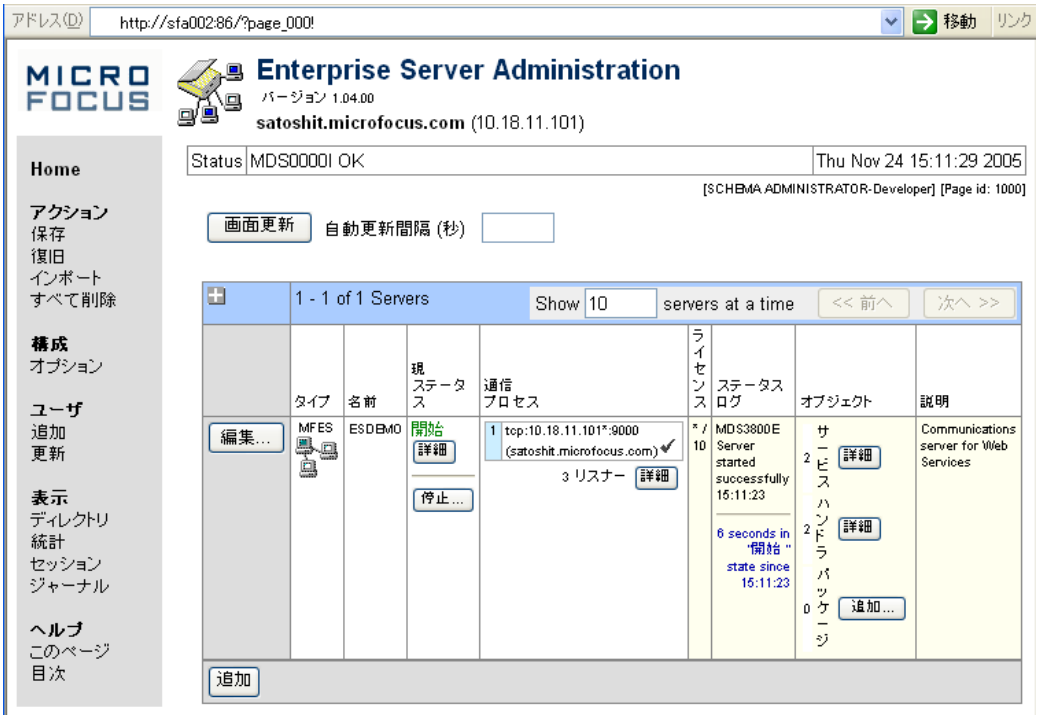
補足1. Oracle 照会プログラムのディプロイと、EJB経由の JCA呼び出し

1) WebSphere管理コンソールから、Micro Focusリソースアダプタ mfcobol-notx.rar を、
Server Expressのマニュアル記載の通りインストールします。



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



3) 以下の OracleのPro*COBOL によるアクセスのCOBOLプログラムを用意します。
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。

```
     $ cat Sel.pco
       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) Oracle の Pro*COBOL でこのプログラムをプレコンパイルします。今回は、Oracle 提供の Pro*COBOL1.8 を使用しました。

```
$ procob18_32  iname=Sel.pco
Pro*COBOL: Release 1.8.77.0.0 - Production on 木 Nov 24 10:51:58 2005
Copyright (c) 1982, 2002, Oracle Corporation.  All rights reserved.
システムのデフォルト・オプション値: /space/ora9i/product/9.2.0.1/precomp/admin/pcccob.cfg
$
```

5) Server Express で、上記で生成された Sel.cob プログラムを以下のようにコンパイルします。

```
$ cob -Utgu  Sel.cob -P -C "INITCALL(orainst.so),NOREENTRANT"
```

INITCALL コンパイラ指令で呼び出す Oracle DBMS ルーチンの共有ライブラリ orainst.so は下記のように生成しました。

```
$ cd $ORACLE_HOME/precomp/lib
$ make LOC_RTSORA32=orainst.so -f ins_precomp.mk orainst.so
```

この時の ins_precomp.mk の中で参照している env_precomp.mk で RTSORA_LINKLINE ではじまる行の -xe を -ze に置き換えます。これは orainst.so を共有ライブラリにするための指令です。

```
RTSORA_LINKLINE=$(COB) -o $(LOC_RTSORA) -t -xe "" $(RTSPORTFLAGS) ¥
                        ↓
RTSORA_LINKLINE=$(COB) -o $(LOC_RTSORA) -t -ze "" $(RTSPORTFLAGS) ¥
```

なお、orainst.so の生成が終了した後は忘れずに –xe に戻してください。

6) 同じく Server Express 環境下で、以下のようにデフォルトマッピングを作成します。

```
$ imtkmake -defmap src=Sel.cob service=SelS type=ejb
Micro Focus Interface Mapping Toolkit v1.0.00
Copyright (C) 2004 Micro Focus International Ltd. All rights reserved.
$
```

これでカレントディレクトリ下にマッピング定義ファイル、Sel.xml と SelS.xml が作成されます。

7) 同じく Server Express 環境下で、以下のように、j2ee.jar を CLASSPATH に張った上で、ディプロイメントパッケージと EJB ラッパーを生成します。

```
$ imtkmake -generate service=SelS type=ejb
Micro Focus Interface Mapping Toolkit v1.0.00
Copyright (C) 2004 Micro Focus International Ltd. All rights reserved.
[parsing started com/mypackage/SelS/SelSHome.java]
[parsing completed 76ms]
[loading /opt/IBM/WebSphere/AppServer/lib/j2ee.jar(javax/ejb/EJBHome.class)]
[loading /opt/IBM/WebSphere/AppServer/java/jre/lib/rt.jar(java/rmi/Remote.class)
]
    ------ 途中省略 ---------
[loading /opt/IBM/WebSphere/AppServer/java/jre/lib/rt.jar(java/lang/Boolean.clas
s)]
[wrote com/mypackage/SelS/SelSBean.class]
[total 1199ms]
added manifest
adding: META-INF/ejb-jar.xml(in = 1290) (out= 496)(deflated 61%)
adding: META-INF/weblogic-ejb-jar.xml(in = 807) (out= 333)(deflated 58%)
adding: com/mypackage/SelS/SelS.class(in = 214) (out= 174)(deflated 18%)
adding: com/mypackage/SelS/SelSBean.class(in = 3320) (out= 1480)(deflated 55%)
adding: com/mypackage/SelS/SelSHome.class(in = 237) (out= 172)(deflated 27%)
minizip: Micro Focus zip utility version 1.1.1
Based on MiniZip 0.15 by Gilles Vollant
Copyright (C) 2002-2003 Micro Focus International Limited
$
```

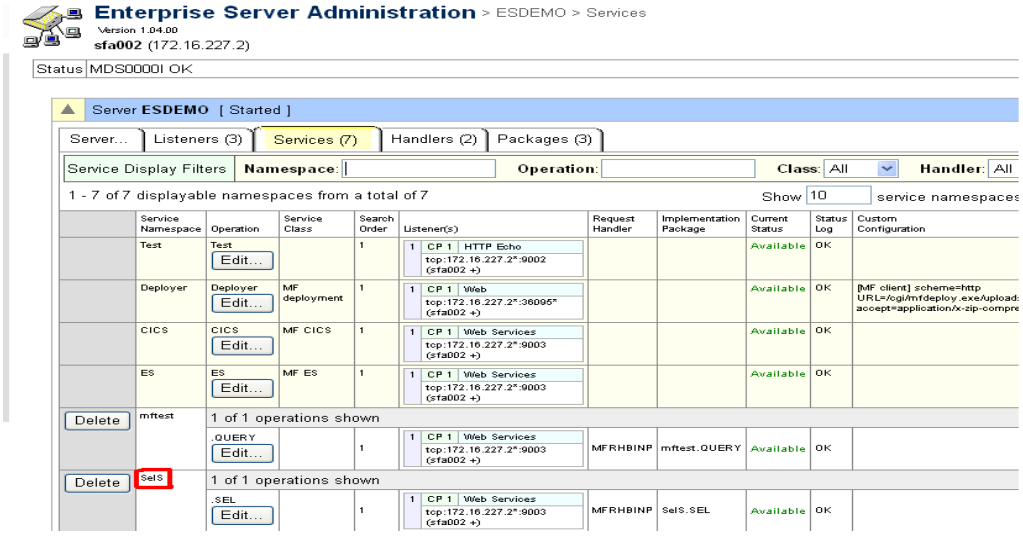
これによって、SelS.deploy ディレクトリ下に以下のパッケージが作成されています:

SelS.car　:COBOL ディプロイメントパッケージ。Enterprise Server にディプロイします

8) 同じく Server Express 環境下で、以下のように ESDEMO へディプロイします。

```
$ imtkmake -deploy carname=SelS.deploy/SelS.car server=ESDEMO,Deployer
Micro Focus Interface Mapping Toolkit v1.0.00
Copyright (C) 2004 Micro Focus International Ltd. All rights reserved.
Sending SelS.deploy/SelS.car to ESDEMO's Deployer at tcp:172.16.227.2:34767...
Received 48 bytes:
http://172.16.227.2:34767/uploads/SelS.mrSdSvWR/

Retrieving deployment log file http://172.16.227.2:34767/uploads/SelS.mrSdSvWR/d
eploylog.txt...
1000 (Tue Nov 22 13:45:25 2005): mfdepinst 1.2.4 starting deployment of COBOL ar
chive "SelS.car"

0010 (Tue Nov 22 13:45:25 2005): Extracting and parsing manifest file

0011 (Tue Nov 22 13:45:25 2005): Processing Manifest element in the manifest fil
e
0012 (Tue Nov 22 13:45:25 2005): Processing "Application" section of Manifest
0012 (Tue Nov 22 13:45:25 2005): Processing "FileList" section of Manifest
2003 (Tue Nov 22 13:45:25 2005): File pathname "/export/home/oracle/stdemo/SEL3/
SelS.deploy/SelS.idt" is absolute; reducing to filename "SelS.idt"
0012 (Tue Nov 22 13:45:25 2005): Processing "Service" section of Manifest
0019 (Tue Nov 22 13:45:25 2005): Manifest processing completed
0020 (Tue Nov 22 13:45:25 2005): Adding service and package objects to directory

0030 (Tue Nov 22 13:45:30 2005): ES server "ESDEMO" notified service "SelS.SEL"
is available
0002 (Tue Nov 22 13:45:30 2005): Installation of package "SelS.car" finished wit
h 1 warnings
Deployment completed
$
```

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

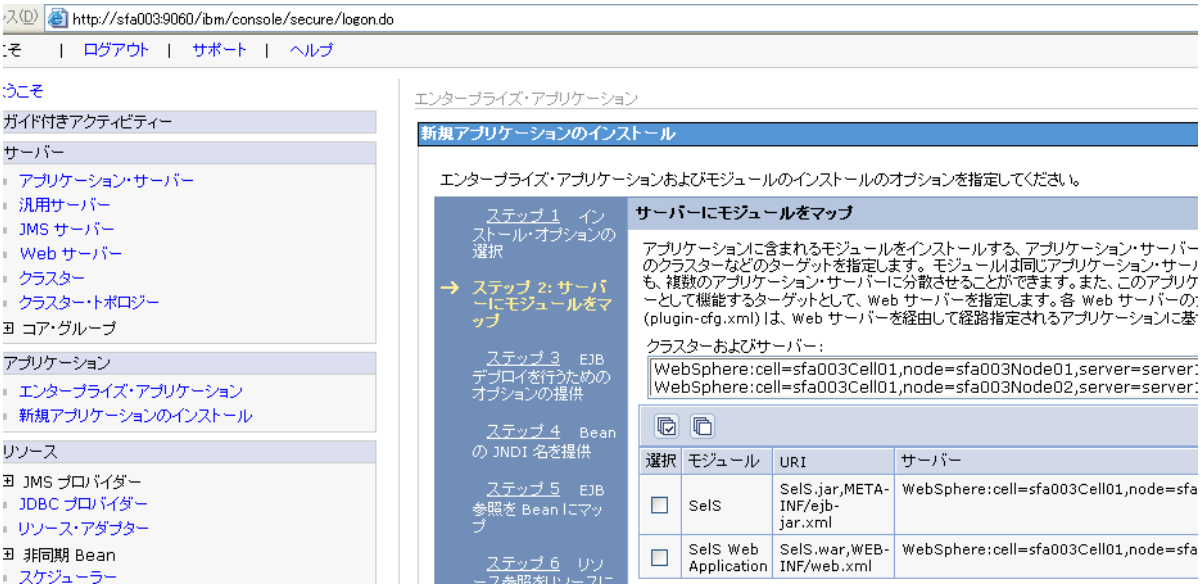


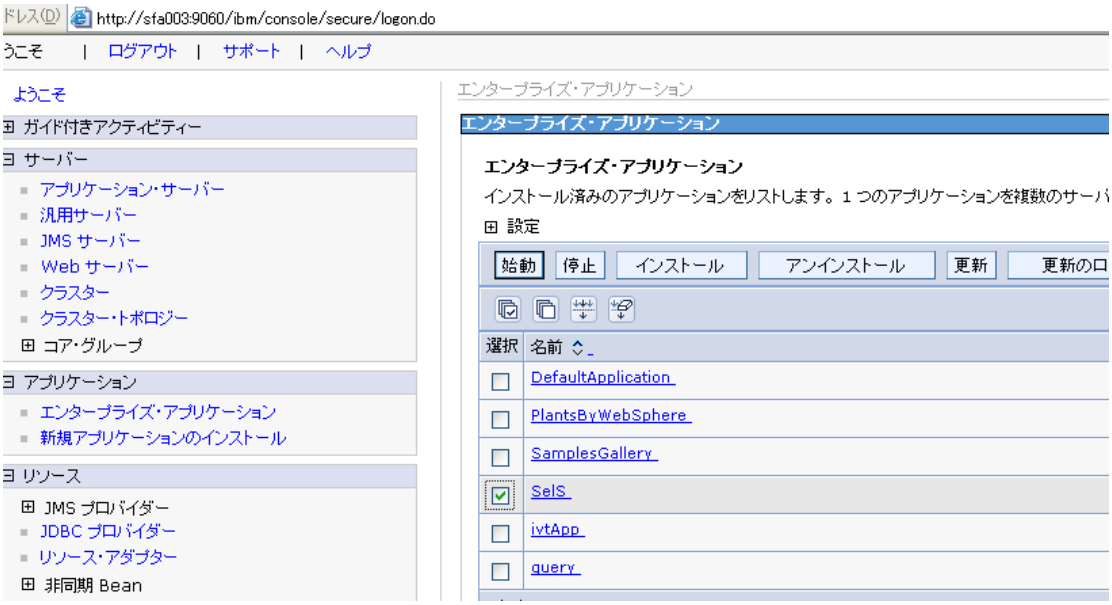
10) 上記に表示されているパッケージパスに、COBOL プログラムのコンパイル済み .gnt コードをコピーします。Oracle DBMS ルーチンの共有ライブラリ orainst.so もコピーします。

11) 同じく Server Express 環境下で、以下のように、WebSphere 向け JSP をインストールするための ear を自動生成します。

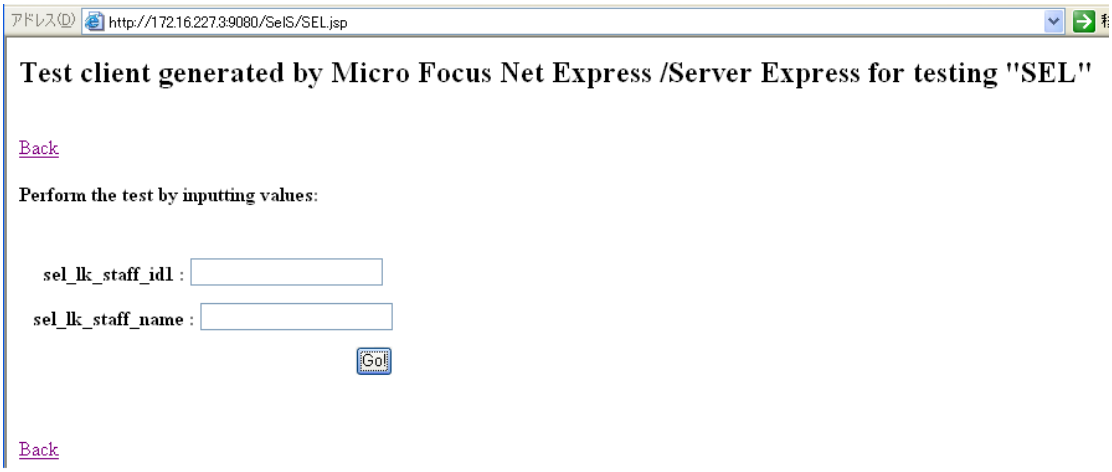
```
$ imtkmake -genclient service=SelS type=ejb appserver="WebSphere 5.x"
Micro Focus Interface Mapping Toolkit v1.0.00
Copyright (C) 2004 Micro Focus International Ltd. All rights reserved.
[parsing started com/mypackage/SelS/SelSHome.java]
[parsing completed 78ms]
[loading /opt/IBM/WebSphere/AppServer/lib/j2ee.jar(javax/ejb/EJBHome.class)]
[loading /opt/IBM/WebSphere/AppServer/java/jre/lib/rt.jar(java/rmi/Remote.class)
]
[loading /opt/IBM/WebSphere/AppServer/java/jre/lib/rt.jar(java/lang/Object.class
)]
[loading /export/home/oracle/stdemo/SEL3/SelS.deploy/com/mypackage/SelS/SelS.jav
a]

        ---- 途中省  -------

adding: WEB-INF/classes/com/mypackage/SelS/SelSServlet.class(in = 8656) (out= 31
77)(deflated 63%)
adding: WEB-INF/classes/com/mypackage/SelS/SelSJspBean.class(in = 1891) (out= 72
3)(deflated 61%)
adding: WEB-INF/classes/com/mypackage/SelS/SelSSessionMonitor.class(in = 1561) (
out= 756)(deflated 51%)
added manifest
adding: SelS.jar(in = 4906) (out= 3989)(deflated 18%)
adding: SelS.war(in = 9970) (out= 9065)(deflated 9%)
adding: mfejblib.jar(in = 2982) (out= 1950)(deflated 34%)
adding: META-INF/application.xml(in = 473) (out= 275)(deflated 41%)
adding: META-INF/ibm-application-bnd.xmi(in = 387) (out= 188)(deflated 51%)
adding: META-INF/ibm-application-ext.xmi(in = 797) (out= 259)(deflated 67%)
$
```

12) 生成された SelS.ear を、WebSphere 管理コンソールからインストールします。設定はすべてデフォルトで行います。以下のように表示されインストールが完了したことを確認します。

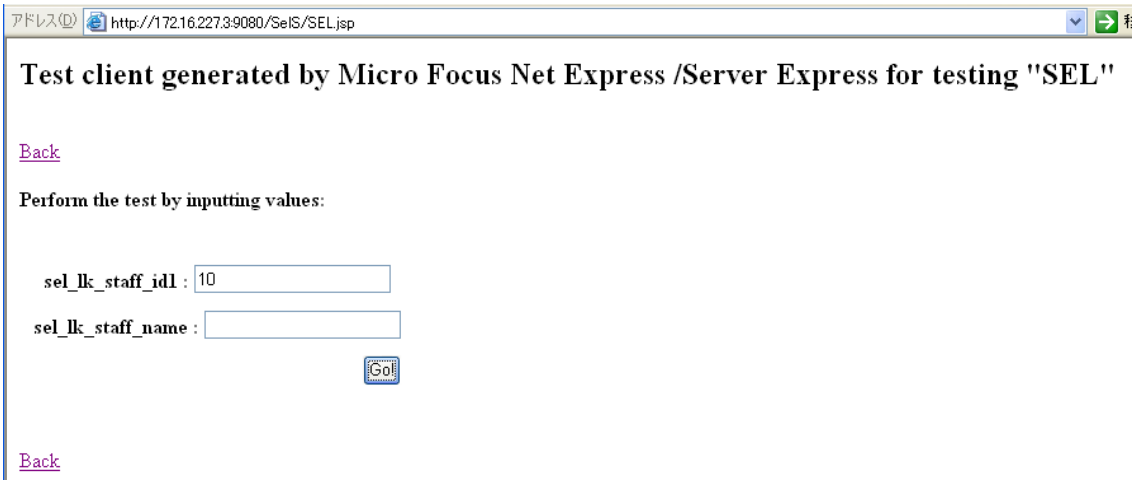




13) ブラウザから http://sfa003:9080/SelS/SelS.jsp を開きます。以下のように自動生成された JSP クライアントが起動します。（図は sfa003 の IP アドレスで http://172.16.226.3:9080/SelS/SelS.jsp ）



14) 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。



15) 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。自動生成された JSP クライアントにはコードセットの明示指定が含まれていませんので日本語データは、"?" に変換されて表示されます。必要に応じて適宜 WebSphere にインストールされた JSP を改造し Shift_JIS コードセットの指定を追加してください。

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) WebSphere 管理コンソールから、インストール済みのMicro Focus リソースアダプタ mfcobol-notx.rar アンインストールします。

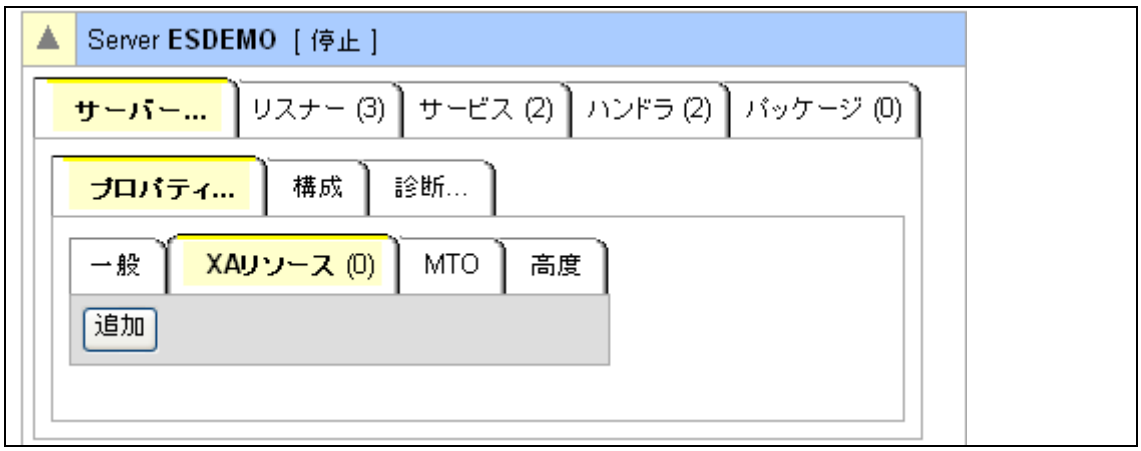
2) 同じく WebSphere 管理コンソールから、Micro Focus リソースアダプタ mfcobol-xa.rar を、Server Express マニュアルの記載の通りインストールします。

3) 新しいリソースアダプタを有効にするために、WebSphere Application Server を起動します。

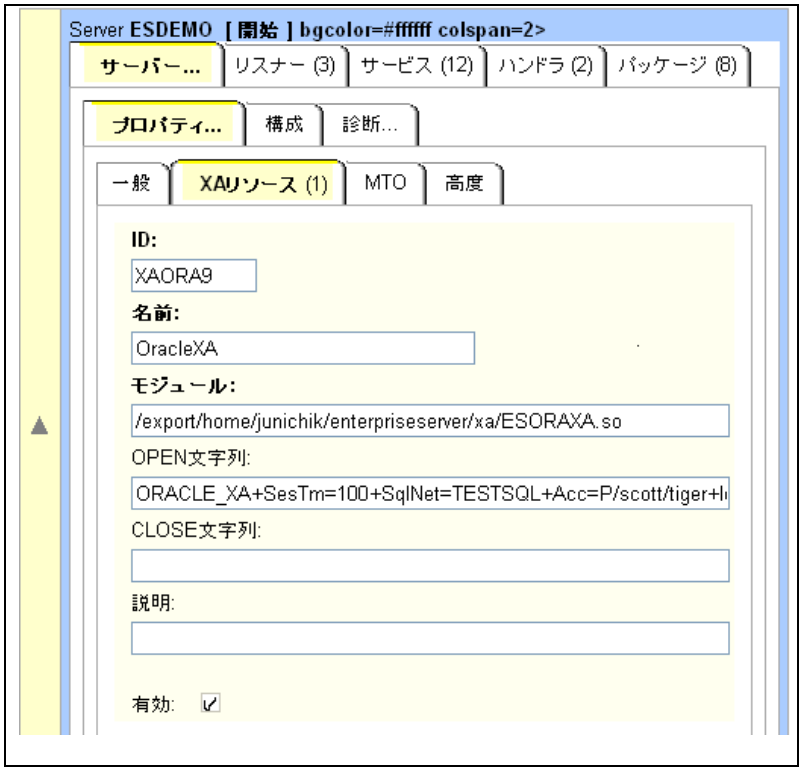
4) Micro Focus Enterprise Server に、Oracle の XA スイッチモジュールを XA リソースとして追加する必要があります。まず、Server Express マニュアルの記載された手順で $COBDIR/src/enterpriseserver/xa に格納されている ESORAXA.CBL ソースをビルドしスイッチモジュール ESORAXA.so を作成します。

```
$ build ora
Linking ESORAXA.so
$ ls
ESORAXA.CBL   ESORAXA.so    build
$
```

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よ、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。



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。

051005 14563620 802 ESDEMO CASXO0015I XAORA9 XA interface initialized successfully 14:56:36

8) 以下の COBOL プログラムを用意します、Oracle

UPP.pco

```
       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ィプロイする 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が、 cob コマンドのオプションから INITCALL を外します。データベースへの接続は Enterprise Server 側で行われるためです。

```
$ procob18_32  iname=UPP.pco

Pro*COBOL: Release 1.8.77.0.0 - Production on 木 Nov 24 10:58:50 2005

Copyright (c) 1982, 2002, Oracle Corporation.  All rights reserved.

システムのデフォルト・オプション値: /space/ora9i/product/9.2.0.1/precomp/admin/pcccob.cfg
```

```
$ cob -Utgu  UPP.cob -P -C "NOREENTRANT"
```

10) コマンドプロンプトから、以下のようにディフォルトマッピングします。

```
$ imtkmake -defmap src=UPP.cob service=UPPS type=ejb
Micro Focus Interface Mapping Toolkit v1.0.00
Copyright (C) 2004 Micro Focus International Ltd. All rights reserved.
$
```

11) これでカレントディレクトリ下にマッピング定義ファイル UPP.xml と UPPS.xml が作成されます。

```
$ imtkmake -generate service=UPPS transaction=container type=ejb
imtkmake -deploy carname=UPPS.deploy/UPPS.car server=ESDEMO,Deployer
imtkmake -genclient service=UPPS type=ejb appserver="WebSphere 5.x"
$ sh ./imtk.sh
Micro Focus Interface Mapping Toolkit v1.0.00
Copyright (C) 2004 Micro Focus International Ltd. All rights reserved.
Micro Focus Interface Mapping Toolkit v1.0.00
Copyright (C) 2004 Micro Focus International Ltd. All rights reserved.
[parsing started com/mypackage/UPPS/UPPSHome.java]
[parsing completed 77ms]

   ------ 途中省略 ------

[total 1196ms]
added manifest
adding: META-INF/ejb-jar.xml(in = 1290) (out= 496)(deflated 61%)
adding: META-INF/weblogic-ejb-jar.xml(in = 807) (out= 334)(deflated 58%)
adding: com/mypackage/UPPS/UPPS.class(in = 232) (out= 179)(deflated 22%)
adding: com/mypackage/UPPS/UPPSBean.class(in = 3361) (out= 1495)(deflated 55%)
adding: com/mypackage/UPPS/UPPSHome.class(in = 237) (out= 173)(deflated 27%)
minizip: Micro Focus zip utility version 1.1.1
Based on MiniZip 0.15 by Gilles Vollant
Copyright (C) 2002-2003 Micro Focus International Limited
$
```

これによって、UPPS.deploy ディレクトリ下に以下のパッケージが作成されます。

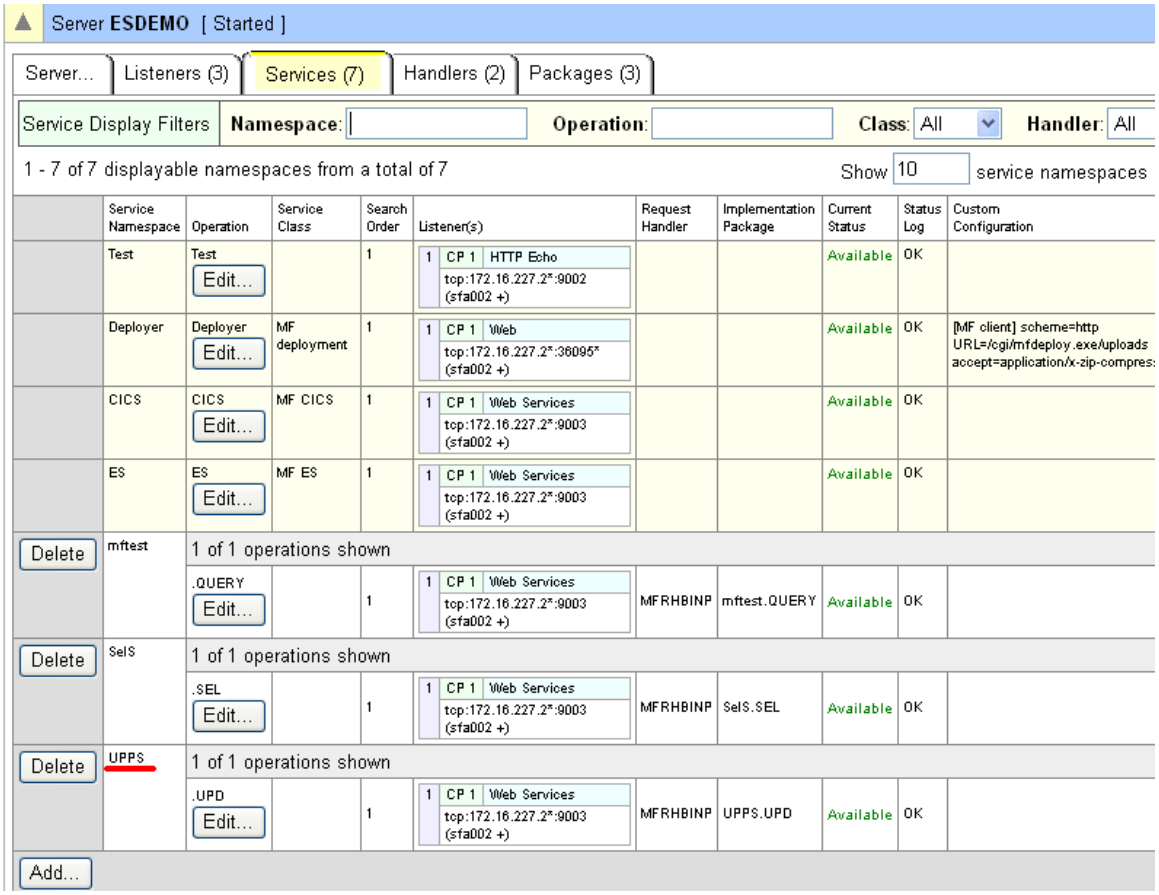
UPPS.car :COBOL ディプロイメントパッケージ。Enterprise Server にディプロイします。
UPPS.jar :ラッパーEJB のパッケージ

12) 同じく Server Express 環境下で、以下のように ESDEMO へディプロイします。

```
$ imtkmake -deploy carname=UPPS.deploy/UPPS.car transaction=container
server=ESDEMO,Deployer

Micro Focus Interface Mapping Toolkit v1.0.00
Copyright (C) 2004 Micro Focus International Ltd. All rights reserved.
Sending UPPS.deploy/UPPS.car to ESDEMO's Deployer at tcp:172.16.227.2:36095...
Received 48 bytes:
http://172.16.227.2:36095/uploads/UPPS.gHcltyPt/

    ---- 途中省略 -----

0012 (Tue Nov 22 15:05:27 2005): Processing "Service" section of Manifest
0019 (Tue Nov 22 15:05:27 2005): Manifest processing completed
0020 (Tue Nov 22 15:05:27 2005): Adding service and package objects to directory
```

```
0030 (Tue Nov 22 15:05:32 2005): ES server "ESDEMO" notified service "UPPS.UPD"
is available
0002 (Tue Nov 22 15:05:32 2005): Installation of package "UPPS.car" finished wit
h 1 warnings
Deployment completed
```

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



14) 上記に表示されているパッケージパスに、COBOL プログラムのコンパイル済み .gnt コードをコピーします。

15) 同じく Server Express 環境下で、以下のように JSP クライアント Web モジュールを生成します。

```
$ imtkmake -genclient service=UPPS type=ejb appserver="WebSphere 5.x"
Copyright (C) 2004 Micro Focus International Ltd. All rights reserved.
[parsing started com/mypackage/UPPS/UPPSHome.java]
[parsing completed 77ms]
[loading /opt/IBM/WebSphere/AppServer/lib/j2ee.jar(javax/ejb/EJBHome.class)]

  ------ 途中省略 ------

adding: WEB-INF/classes/com/mypackage/UPPS/UPPSSessionMonitor.class(in = 1561) (
out= 758)(deflated 51%)
added manifest
adding: UPPS.jar(in = 4927) (out= 3919)(deflated 20%)
adding: UPPS.war(in = 10190) (out= 9282)(deflated 8%)
adding: mfejblib.jar(in = 2982) (out= 1950)(deflated 34%)
adding: META-INF/application.xml(in = 473) (out= 276)(deflated 41%)
```

```
adding: META-INF/ibm-application-bnd.xmi(in = 387) (out= 188)(deflated 51%)
adding: META-INF/ibm-application-ext.xmi(in = 797) (out= 260)(deflated 67%)
$
```

これによって、UPPS.deploy ディレクトリ下に以下のパッケージが作成されています。

UPPS.ear　:ラッパーEJB と JSP クライアントを含むエンタープライズアーカイブ。WebSphere にディプロイします。

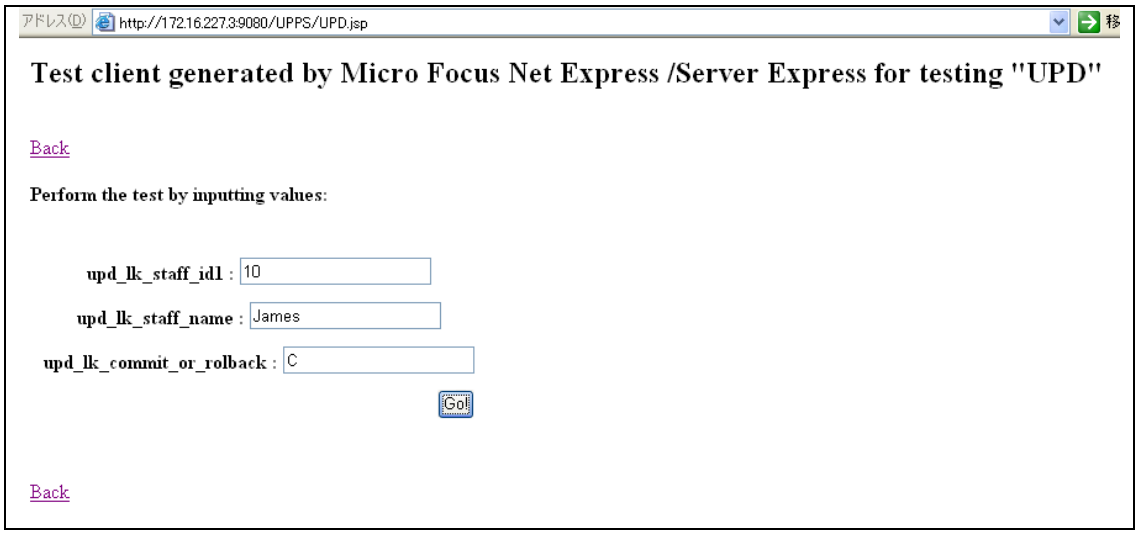
16) 生成された UPPS.ear を、WebSphere 管理コンソールからインストールします。設定は全てディフォルトで行います。管理コンソールでインストールしたエンタープライズアプリケーション UPPS を開始します。

17) 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 Marenghi            38 Mgr              5    17506.75
        40 Brien               38 Sales            6     18006       846.55

SQL>
```

現在は Smith さんの名前が格納されています。

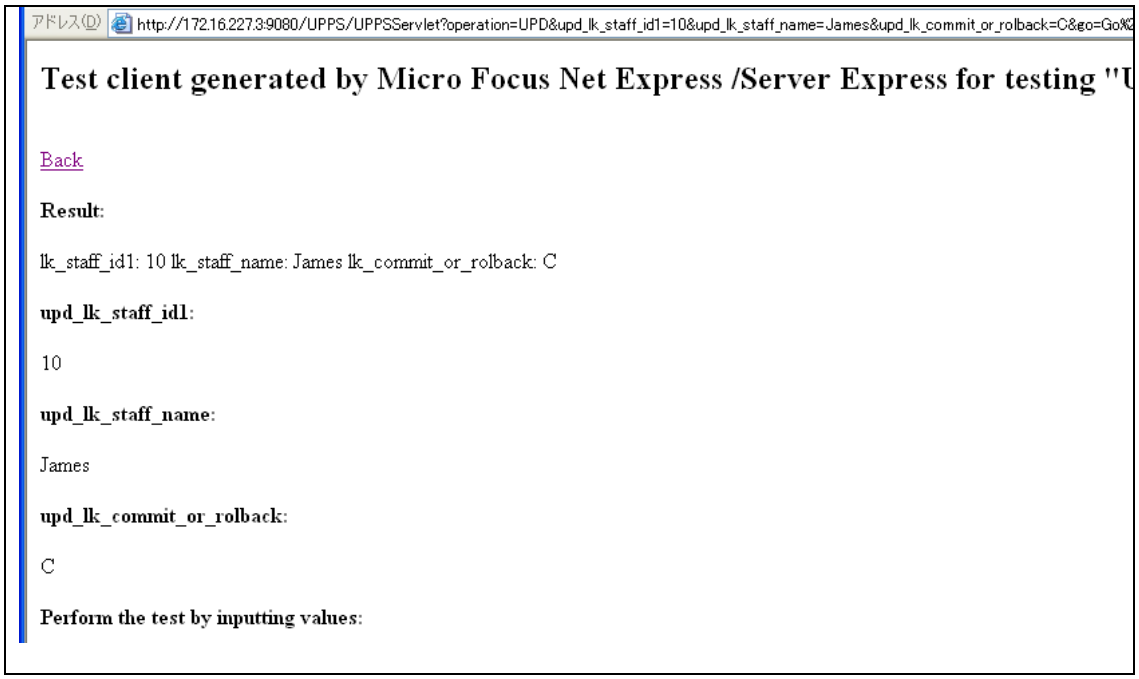
18) ブラウザから、 http://sfa003:9080/UPPS/UPPS.jsp を開きます。以下のように自動生成された JSP クライアントが起動します。(図は sfa003 の IP アドレスで http://172.16.226.3:9080/UPPS/UPPS.jsp)



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”を入力し[Go！] をクリックします。

19) 以下のように結果が返ります。



20) 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
        40 Brien              38 Sales               6      18006     846.55

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。

21) 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。

```
051122 15193577        803 ESDEMO   CASKC0027E Error executing service 'UPPS.UPD'
Object Code error : file '/opt/microfocus/cobol/deploy/UPPS.gHcltyPt/UPP.gnt'
error code: 153, pc=0, call=1, seg=0
153     Subscript out of range (in UPP.cob, line 202) 15:19:35
```

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
```

22) 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
        40 Brien                38 Sales                6       18006        846.55

SQL>
```

23) 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> INSERT INTO STAFF VALUES (40, 'Brien', 38, 'Sales', 6, 18006.00, 846.55);
```

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