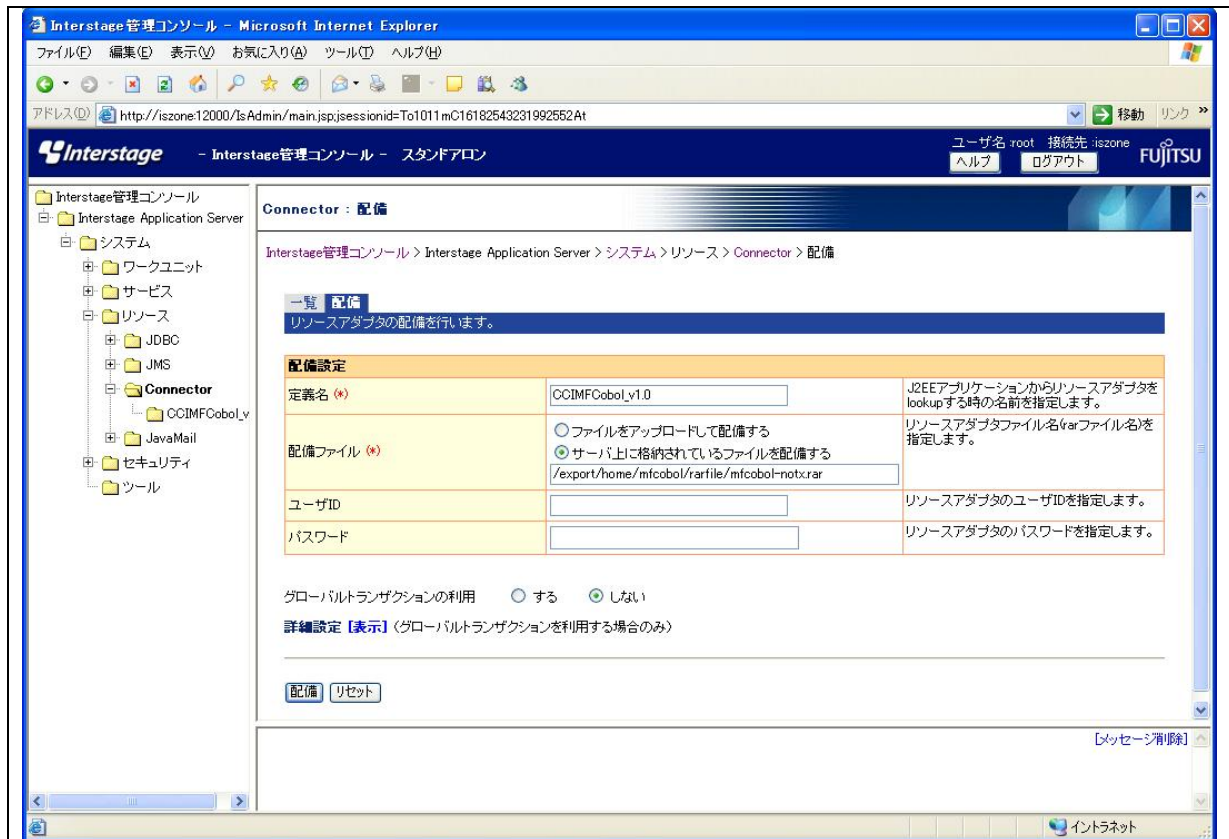


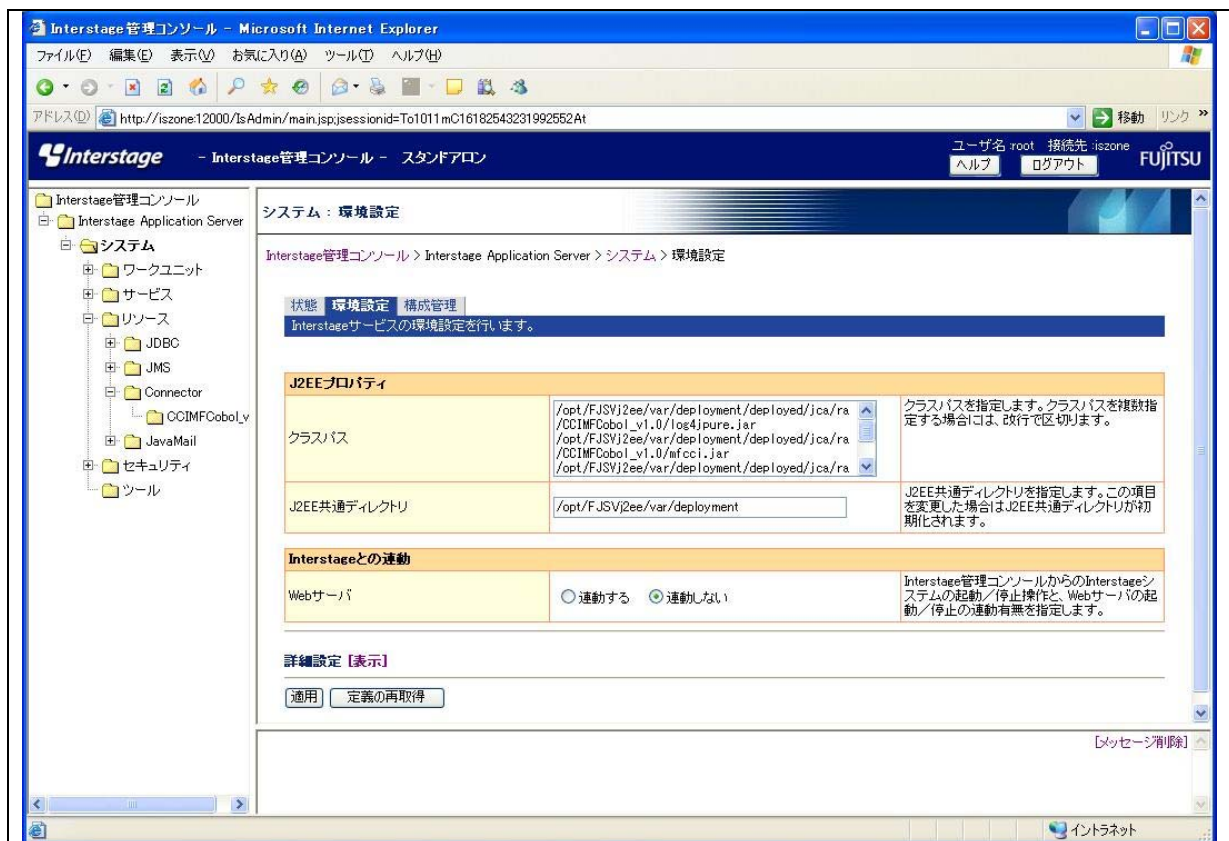
補足1. Oracle 照会プログラムのデプロイと、EJB経由の JCA呼び出し

- 1) Micro Focusリソースアダプタ mfcobol-notx.rar を Interstage 管理コンソールより登録します。
登録は、Interstage管理コンソール > Interstage Application Server > システム > リソース > Connector > 配備 で定義名に「CCIMFCobol_v1.0」、配置ファイルに mfcobol-notx.rar を指定し配置ボタンをクリックします。

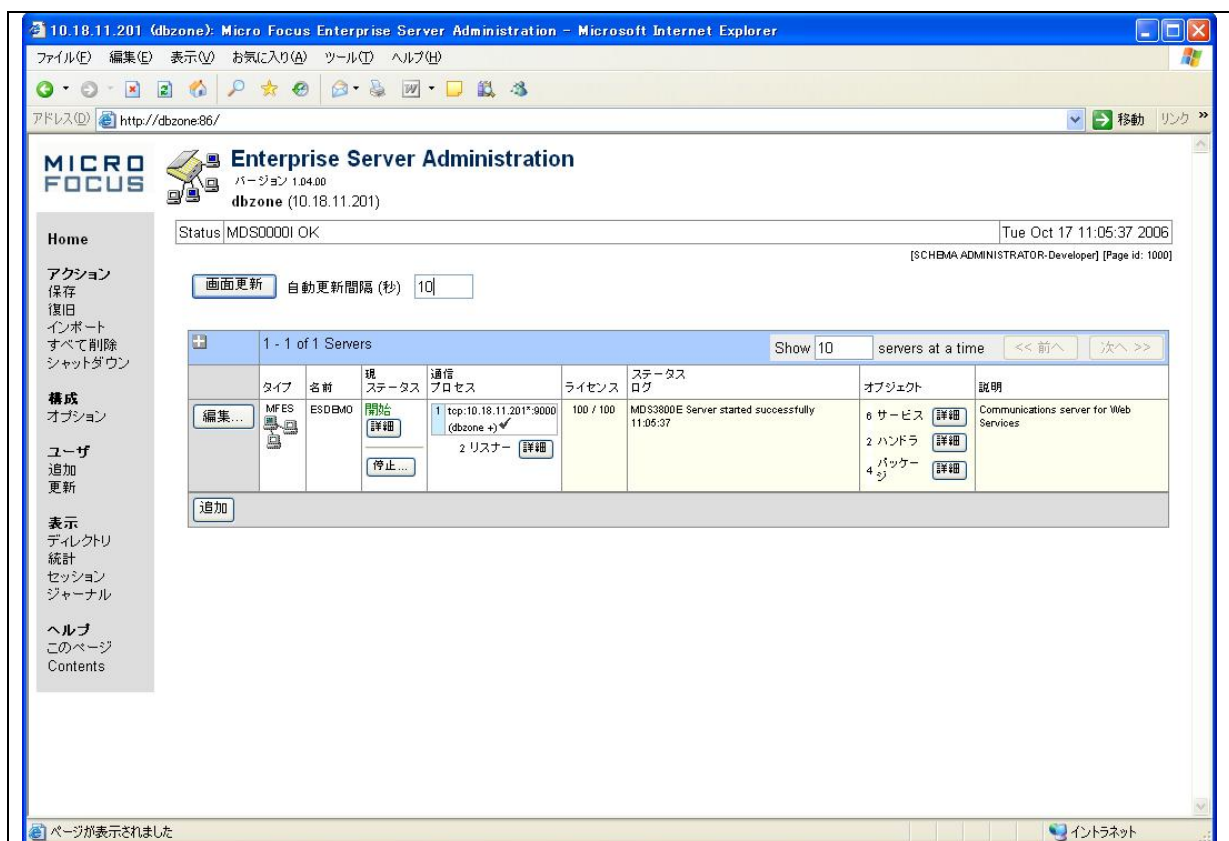


- 2) Interstage 管理コンソールよりクラスパスを登録します。
Interstage管理コンソール > Interstage Application Server > システム > 環境設定 の J2EE プロパティ・クラスパスに以下を追加します。
\$IS_HOME/FJSVj2ee/var/deployment/deployed/jca/ra/CCIMFCobol_v1.0/log4jpure.jar
\$IS_HOME/FJSVj2ee/var/deployment/deployed/jca/ra/CCIMFCobol_v1.0/mfccijar
\$IS_HOME/FJSVj2ee/var/deployment/deployed/jca/ra/CCIMFCobol_v1.0/mfcobolpure.jar
\$IS_HOME/FJSVj2ee/var/deployment/deployed/jca/ra/CCIMFCobol_v1.0/mftransport.jar

尚、\$IS_HOME は Interstage のインストールディレクトリに置き換えてください。



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



- 4) Server 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.
```

- 5) Oracle の Pro*COBOL でこのプログラムをプレコンパイルします。

```
$ procob32 iname=Sel.pco oname=Sel.cbl
```

Pro*COBOL: Release 10.2.0.1.0 – Production on 火 10 月 17 12:32:02 2006

Copyright (c) 1982, 2005, Oracle. All rights reserved.

システムのデフォルト・オプション値: /opt/app/oracle/product/10.2.0/Db_1/precomp/admin/pcbcfg.cfg

- 6) 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す。
作成した共有ライブラリを INITCALL コンパイラ指令で呼び出すようにします。

```
$ cob32 -z -e "" -o orainst.so $ORACLE_HOME/precomp/lib32/cobsqlintf.o -L$ORACLE_HOME/lib32
-lclntsh `cat $ORACLE_HOME/lib32/ldflags` `cat $ORACLE_HOME/lib32/sysliblist` -ldl -lm
```

- 7) Server Express で、このプリコンパイルされたプログラムをコンパイルします。

```
$ cob -UtgU Sel.cbl -P -C "INITCALL(oraInst.so), NOREENTRANT"
```

- 8) imtkmake コマンドを利用して、デフォルトマッピングを作成します。コマンドを実行すると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。

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

- 9) imtkmake コマンドを利用して、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。
CLASSPATH に、\$COBDIR/lib/mfcobolpure.jar、\$COBDIR/lib/mfconnector.jar、isj2ee.jar を設定しておきます。

SelS.deploy ディレクトリが作成され、配下に パッケージ: SelS.car されます。

SelS.car の中には、Sel.gnt、oraInst.so も含まれています。

```
$ imtkmake -generate service=SelS type=ejb appserver="WebLogic 8.1" /pg/Sel.gnt /pg/oraInst.so
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 24ms]
[loading /export/home/oracle/isj2ee.jar(java/ejb/EJBHome.class)]
[loading /usr/j2sdk1.4.2_08/jre/lib/rt.jar(java/rmi/Remote.class)]
[loading /usr/j2sdk1.4.2_08/jre/lib/rt.jar(java/lang/Object.class)]
[loading ./com/mypackage/SelS/SelS.java]
```

----- 途中省略 -----

```
[loading /usr/j2sdk1.4.2_08/jre/lib/rt.jar(java/lang/Boolean.class)]
[wrote com/mypackage/SelS/SelSBean.class]
[total 317ms]
added manifest
adding: META-INF/ejb-jar.xml(in = 1290) (out= 496)(deflated 61%)
adding: META-INF/weblogic-ejb-jar.xml(in = 807) (out= 332)(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
```

- 10) imtkmake コマンドを利用して、作成したパッケージ: SelS.car を Enterprise Server の ESDEMO ヘディブロイします。

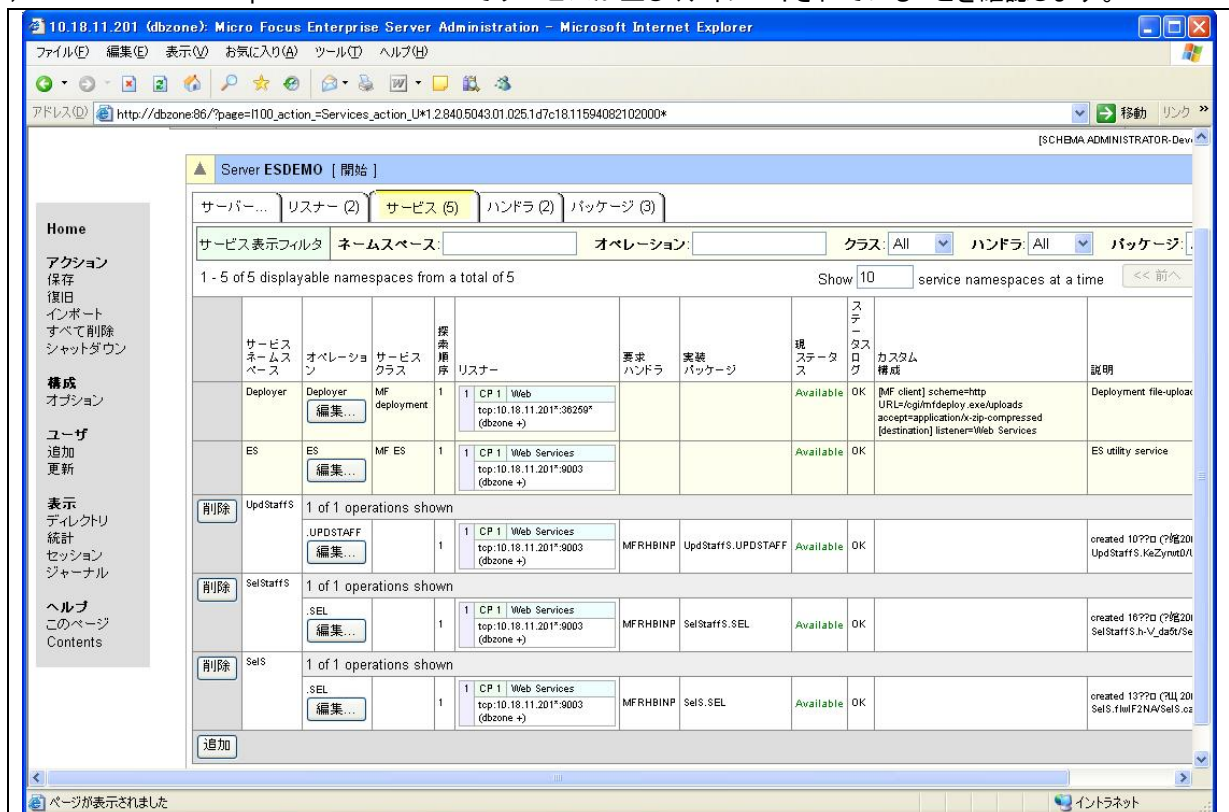
```
$ imtkmake -deploy carname=SelS.deploy/SelS.car server=ESDEMO,Deployer
Micro Focus Interface Mapping Toolkit v1.0.00
Copyright (C) 2004 Micro Focus International Ltd. All rights reserved.
Sending SelS.deploy/SelS.car to ESDEMO's Deployer at tcp:10.18.11.201:36259...
Received 48 bytes:
http://10.18.11.201:36259/uploads/SelS.gLo0tWvp/

Server returned 202
Retrieving deployment log file http://10.18.11.201:36259/uploads/SelS.gLo0tWvp/deploylog.txt...
1000 (2006 年 10 月 17 日 (火) 12 時 58 分 01 秒): mfdepinst 1.2.4 starting deployment of COBOL archive
"SelS.car"
```

0010 (2006 年 10 月 17 日 (火) 12 時 58 分 01 秒): Extracting and parsing manifest file
 0011 (2006 年 10 月 17 日 (火) 12 時 58 分 01 秒): Processing Manifest element in the manifest file
 0012 (2006 年 10 月 17 日 (火) 12 時 58 分 01 秒): Processing "Application" section of Manifest
 0012 (2006 年 10 月 17 日 (火) 12 時 58 分 01 秒): Processing "FileList" section of Manifest
 2003 (2006 年 10 月 17 日 (火) 12 時 58 分 01 秒): File pathname
 "/export/home/oracle/ESTEST/SelS.deploy/SelS.idt" is absolute; reducing to filename "SelS.idt"
 2003 (2006 年 10 月 17 日 (火) 12 時 58 分 01 秒): File pathname "/export/home/oracle/ESTEST/Sel.gnt" is
 absolute; reducing to filename "Sel.gnt"
 2003 (2006 年 10 月 17 日 (火) 12 時 58 分 01 秒): File pathname "/export/home/oracle/ESTEST/orainst.so"
 is absolute; reducing to filename "orainst.so"
 0012 (2006 年 10 月 17 日 (火) 12 時 58 分 01 秒): Processing "Service" section of Manifest
 0019 (2006 年 10 月 17 日 (火) 12 時 58 分 01 秒): Manifest processing completed
 0020 (2006 年 10 月 17 日 (火) 12 時 58 分 01 秒): Adding service and package objects to directory

0030 (2006 年 10 月 17 日 (火) 12 時 58 分 06 秒): ES server "ESDEMO" notified service "SelS.SEL" is
 available
 0002 (2006 年 10 月 17 日 (火) 12 時 58 分 06 秒): Installation of package "SelS.car" finished with 3 warnings
 Deployment completed with warnings

11) 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。



12) imtkmake コマンドを利用して、JSP をインストールするための ear を自動生成します。
 appserver パラメータで配備先の J2EE アプリケーションサーバーの種別を 指定できるようになっています
 が、現状ではこの選択肢の中に Interstage はありません。ここでは WebLogic 8.1 を指定して生成して
 おけば差し支えありません。

```
$ imtkmake -genclient service=SelS type=ejb appserver="WebLogic 8.1"
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 23ms]

[loading /export/home/oracle/isj2ee.jar(javax/ejb/EJBHome.class)]

[loading /usr/j2sdk1.4.2_08/jre/lib/rt.jar(java/rmi/Remote.class)]

[loading /usr/j2sdk1.4.2_08/jre/lib/rt.jar(java/lang/Object.class)]

----- 途中省略 -----

[loading /export/home/oracle/isj2ee.jar(javax/ejb/RemoveException.class)]

[wrote WEB-INF/classes/com/mypackage/SelS/SelSSessionMonitor.class]

[total 258ms]

adding: WEB-INF/web.xml(in = 898) (out= 374)(deflated 58%)

adding: WEB-INF/weblogic.xml(in = 522) (out= 259)(deflated 50%)

adding: META-INF/Manifest.mf(in = 79) (out= 78)(deflated 1%)

adding: WEB-INF/lib/mfejbllib.jar(in = 2982) (out= 1950)(deflated 34%)

adding: SelSMain.jsp(in = 460) (out= 296)(deflated 35%)

adding: SEL.jsp(in = 1559) (out= 637)(deflated 59%)

adding: WEB-INF/classes/com/mypackage/SelS/SelSServlet.class(in = 8656) (out= 3177)(deflated 63%)

adding: WEB-INF/classes/com/mypackage/SelS/SelSjspBean.class(in = 1891) (out= 723)(deflated 61%)

adding: WEB-INF/classes/com/mypackage/SelS/SelSSessionMonitor.class(in = 1561) (out= 756)(deflated 51%)

added manifest

adding: SelS.jar(in = 3773) (out= 3168)(deflated 16%)

adding: SelS.war(in = 9630) (out= 8811)(deflated 8%)

adding: mfejbllib.jar(in = 2982) (out= 1950)(deflated 34%)

adding: META-INF/application.xml(in = 419) (out= 252)(deflated 39%)

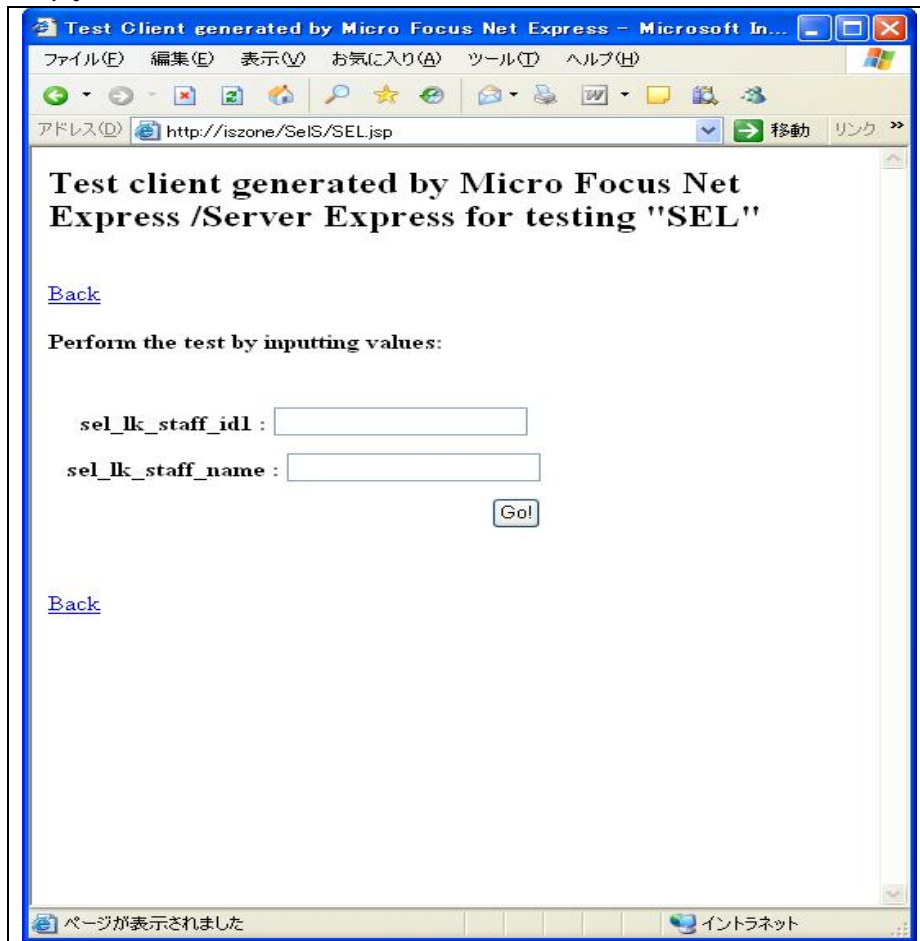
adding: META-INF/weblogic-application.xml(in = 301) (out= 184)(deflated 38%)

13) 生成された SelS.ear を Interstage 管理コンソールより配置する。

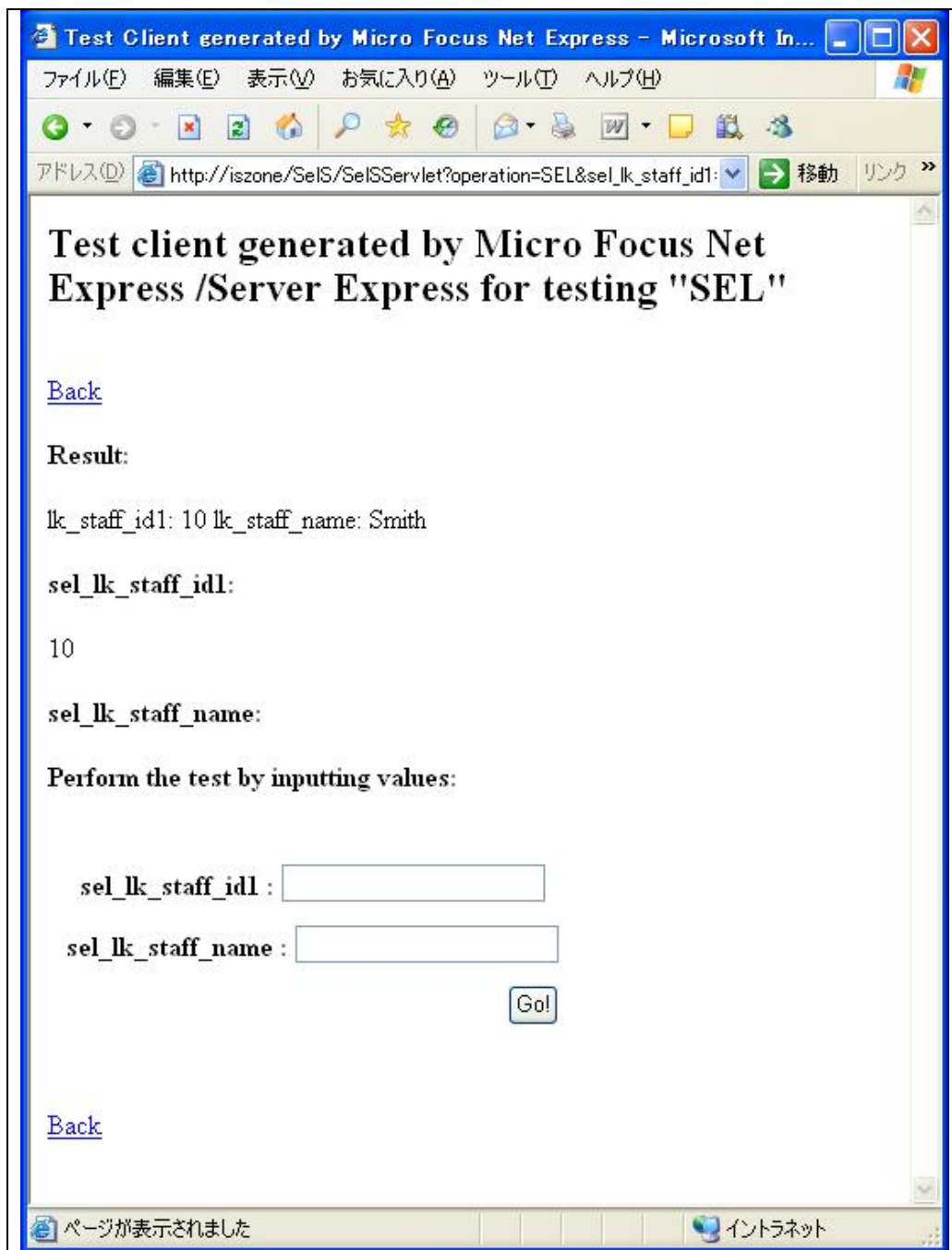
Interstage 管理コンソール > Interstage Application Server > システム > ワークユニット > SampleServer > 配備 で配備ファイルに生成された SelS.ear を指定して配置ボタンをクリックします。以下のように表示されインストールが完了したことを確認します。



- 14) ブラウザから <http://iszone/SelS/SEL.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。



- 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。



補足2. Oracle 更新プログラムのディプロイと、EJB経由の JCA呼び出しにおける例外処理

- 1) インストール済みのMicro Focus リソースアダプタ mfcobol-notx.rar を配備解除します。
- 2) 同じく Micro Focusリソースアダプタ mfcobol-xa.rar を Interstage 管理コンソールより登録します。
登録は、Interstage管理コンソール > Interstage Application Server > システム > リソース > Connector > 配備 で定義名に「CCIMFCobol_v1.0」、配置ファイルに mfcobol-xa.rar を指定し配置ボタンをクリックします。



- 3) 新しいリソースアダプタを有効にするために、Interstage Application Server 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 \$COBDIR/src/enterpriseserver/xa に移動します。
 - ② シェル: build を利用して、ESORA10XA.CBL をビルドしスイッチモジュール: ESORA10XA.dll を作成します。
 - ③ 詳細は、Server 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。

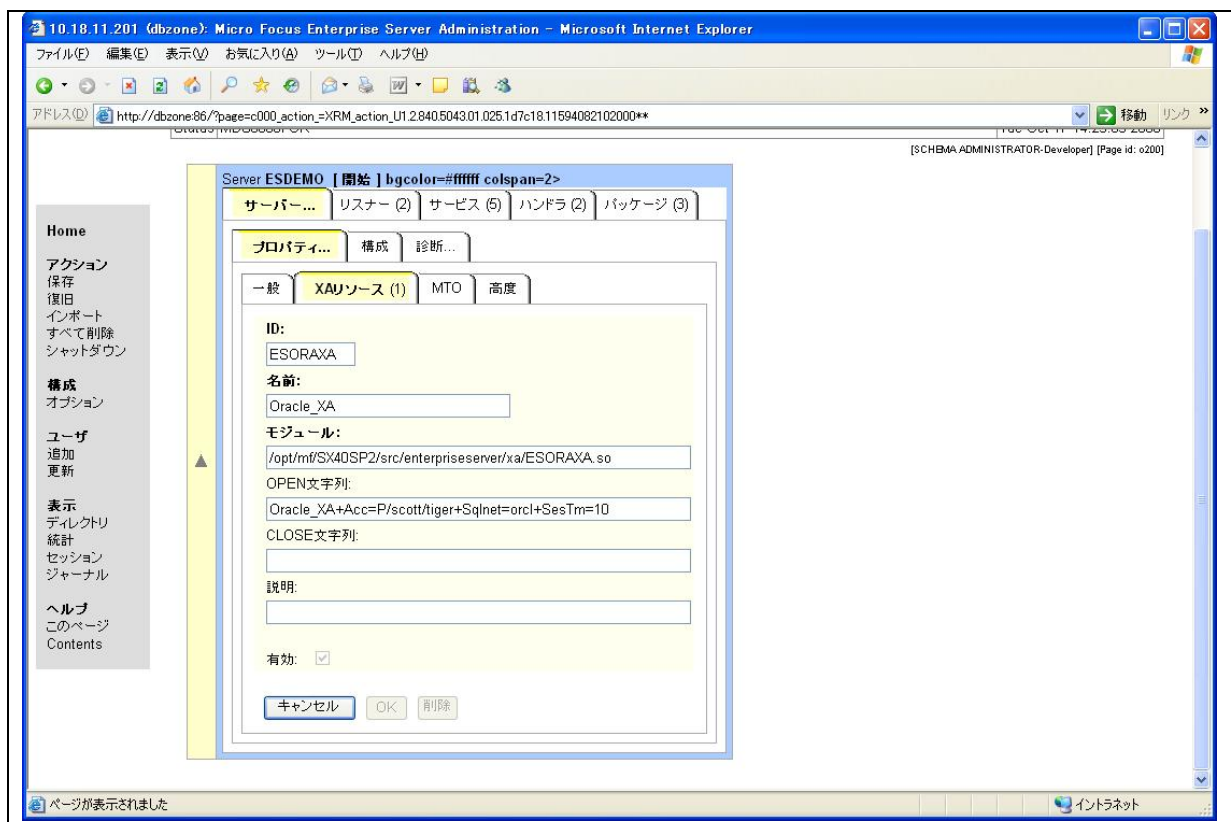
```
$ cd $COBDIR/src/enterpriseserver/xa
$ ./build ora
$ ls
ESDB2XA.CBL  ESORAXA.CBL  ESORAXA.so  build  build.ora9
```

- 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。

- 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、Oracle10g アプリケーション開発者ガイド - 基礎編 「15 Oracle XA を使用したアプリケーションの開発」をご覧ください。
http://otndnld.oracle.co.jp/document/products/oracle10g/102/doc_cd/nav/portal_3.htm
- ④ 設定した値の例
ID: ESORAXA
名前: Oracle_XA
モジュール: <インストールディレクトリ> /src/enterpriseserver/xa/ESORAXA.so
OPEN 文字列: Oracle_XA+Acc=P/ユーザ/パスワード+Sqlnet=サービス名+SesTm=10



- 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 CASX000151 XAORACLE XA interface initialized successfully

- 8) Server 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.
```

トランザクションが必要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。Oracle の Pro*COBOL でこのプログラムをプレコンパイルします。

```
$ procob32 iname=UPP.pco oname=UPP.cbl
```

Pro*COBOL: Release 10.2.0.1.0 - Production on 火 10 月 17 14:41:22 2006

Copyright (c) 1982, 2005, Oracle. All rights reserved.

システムのデフォルト・オプション値: /opt/app/oracle/product/10.2.0/Db_1/precomp/admin/pcbcfg.cfg

- 10) Server Express で、このプリコンパイルされたプログラムをコンパイルします。

```
$ cob -UtgU UPP.cbl -P -C "NOREENTRANT"
```

- 11) imtkmake コマンドを利用して、デフォルトマッピングを作成します。コマンドを実行するとカレントディレクトリ下にマッピング定義ファイル、UPPS.xml と UPPS.xml が作成されます。

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

- 12) トランザクションを有効にするために UPPS.xml の transactionType を "requires" に変更します。

修正前 <operation entryPoint="UPP" name="UPP" transactionType="notSupported">

修正後 <operation entryPoint="UPP" name="UPP" transactionType="requires">

- 13) imtkmake コマンドを利用して、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。
CLASSPATH に、\$COBDIR/lib/mfcobolpure.jar、\$COBDIR/lib/mfconnector.jar、isj2ee.jar を設定しておきます。

UPPS.deploy ディレクトリが作成され、配下に パッケージ: UPPS.car されます。

UPPS.car の中には、UPPS.gnt を含まれています。

```
$ imtkmake -generate service=UPPS type=ejb appserver="WebLogic 8.1" transaction=container
/pg/UPP.gnt
```

```
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 24ms]
[loading /export/home/oracle/isj2ee.jar(javax/ejb/EJBHome.class)]
[loading /usr/j2sdk1.4.2_08/jre/lib/rt.jar(java/rmi/Remote.class)]
[loading /usr/j2sdk1.4.2_08/jre/lib/rt.jar(java/lang/Object.class)]
[loading ./com/mypackage/UPPS/UPPS.java]
```

----- 途中省略 -----

```
[loading /usr/j2sdk1.4.2_08/jre/lib/rt.jar(java/lang/Boolean.class)]
[wrote com/mypackage/UPPS/UPPSBean.class]
[total 312ms]
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/UPPS/UPPS.class(in = 232) (out= 178)(deflated 23%)
adding: com/mypackage/UPPS/UPPSBean.class(in = 3361) (out= 1493)(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
```

- 14) imtkmake コマンドを利用して、作成したパッケージ: UPPS.car を Enterprise Server の ESDEMO へデプロイします。

```
$ imtkmake -deploy carname=UPPS.deploy/UPPS.car 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:10.18.11.201:37809...

Received 48 bytes:

http://10.18.11.201:37809/uploads/UPPS.Gj8IMgLv/

Server returned 202

Retrieving deployment log file http://10.18.11.201:37809/uploads/UPPS.Gj8IMgLv/deploylog.txt...

1000 (2006 年 10 月 17 日 (火) 14 時 50 分 00 秒): mfdepinst 1.2.4 starting deployment of COBOL archive "UPPS.car"

0010 (2006 年 10 月 17 日 (火) 14 時 50 分 00 秒): Extracting and parsing manifest file

0011 (2006 年 10 月 17 日 (火) 14 時 50 分 00 秒): Processing Manifest element in the manifest file

0012 (2006 年 10 月 17 日 (火) 14 時 50 分 00 秒): Processing "Application" section of Manifest

0012 (2006 年 10 月 17 日 (火) 14 時 50 分 00 秒): Processing "FileList" section of Manifest

2003 (2006 年 10 月 17 日 (火) 14 時 50 分 00 秒): File pathname "/export/home/oracle/ESTEST/UPPS.deploy/UPPS.idt" is absolute; reducing to filename "UPPS.idt"

2003 (2006 年 10 月 17 日 (火) 14 時 50 分 00 秒): File pathname "/export/home/oracle/ESTEST/UPP.gnt" is absolute; reducing to filename "UPP.gnt"

0012 (2006 年 10 月 17 日 (火) 14 時 50 分 00 秒): Processing "Service" section of Manifest

0019 (2006 年 10 月 17 日 (火) 14 時 50 分 00 秒): Manifest processing completed

0020 (2006 年 10 月 17 日 (火) 14 時 50 分 00 秒): Adding service and package objects to directory

0030 (2006 年 10 月 17 日 (火) 14 時 50 分 05 秒): ES server "ESDEMO" notified service "UPPS.UPP" is available

0002 (2006 年 10 月 17 日 (火) 14 時 50 分 05 秒): Installation of package "UPPS.car" finished with 2 warnings
Deployment completed with warnings

15) 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。

The screenshot shows the Micro Focus Enterprise Server Administration web interface in Microsoft Internet Explorer. The browser address bar shows 'http://dbzone86/?page_0001'. The interface has a sidebar on the left with navigation links like 'Home', 'アクション', '構成', 'ユーザー', '表示', 'ヘルプ'. The main content area displays a table of services. The table has columns for 'サービス名', 'オペレーション', 'サービスクラス', '探索順序', 'リスナー', '要求', 'インストール', 'ステータス', 'カスタム構成', and '説明'. The table lists several services including 'Deployer', 'ES', 'UpdStaffs', 'SelStaffs', 'SelS', and 'UPPS'. Each service row has a '削除' (Delete) button and a '追加' (Add) button. The 'UPPS' service is highlighted in yellow.

サービス名	オペレーション	サービスクラス	探索順序	リスナー	要求	インストール	ステータス	カスタム構成	説明
Deployer	Deployer	MF deployment	1	1 CP 1 Web top:10.18.11.201*:37809* (dbzone +)			Available	OK	[MF client] scheme=http URL=/cgi-bin/deploy.exe/uploads accept=application/x-zip-compressed [destination] listener=Web Services
ES	ES	MF ES	1	1 CP 1 Web Services top:10.18.11.201*:9003 (dbzone +)			Available	OK	ES utility service
UpdStaffs	1 of 1 operations shown	UPDSTAFF	1	1 CP 1 Web Services top:10.18.11.201*:9003 (dbzone +)	MFRHBINP	UpdStaffs.UPDSTAFF	Available	OK	created 10??0 (?9620) UpdStaffs.KeZymd0L
SelStaffs	1 of 1 operations shown	SEL	1	1 CP 1 Web Services top:10.18.11.201*:9003 (dbzone +)	MFRHBINP	SelStaffs.SEL	Available	OK	created 16??0 (?9620) SelStaffs.h-V_daf0/Se
SelS	1 of 1 operations shown	SEL	1	1 CP 1 Web Services top:10.18.11.201*:9003 (dbzone +)	MFRHBINP	SelS.SEL	Available	OK	created 13??0 (?11 20) SelS.flnf2NA/SelS.o2
UPPS	1 of 1 operations shown	UPP	1	1 CP 1 Web Services top:10.18.11.201*:9003 (dbzone +)	MFRHBINP	UPPS.UPP	Available	OK	created 14??0 (?11 20) UPPS.Gj8IMgLvUPPS

16) imtkmake コマンドを利用して、JSP をインストールするための ear を自動生成します。

appserver パラメータで配備先の J2EE アプリケーションサーバーの種別を 指定できるようになっていますが、現状ではこの選択肢の中に Interstage はありません。ここでは WebLogic 8.1 を指定して生成しておけば差し支えありません。

```
$ imtkmake -genclient service=UPPS type=ejb appserver="WebLogic 8.1"
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 23ms]
[loading /export/home/oracle/isj2ee.jar(javax/ejb/EJBHome.class)]
[loading /usr/j2sdk1.4.2_08/jre/lib/rt.jar(java/rmi/Remote.class)]
[loading /usr/j2sdk1.4.2_08/jre/lib/rt.jar(java/lang/Object.class)]



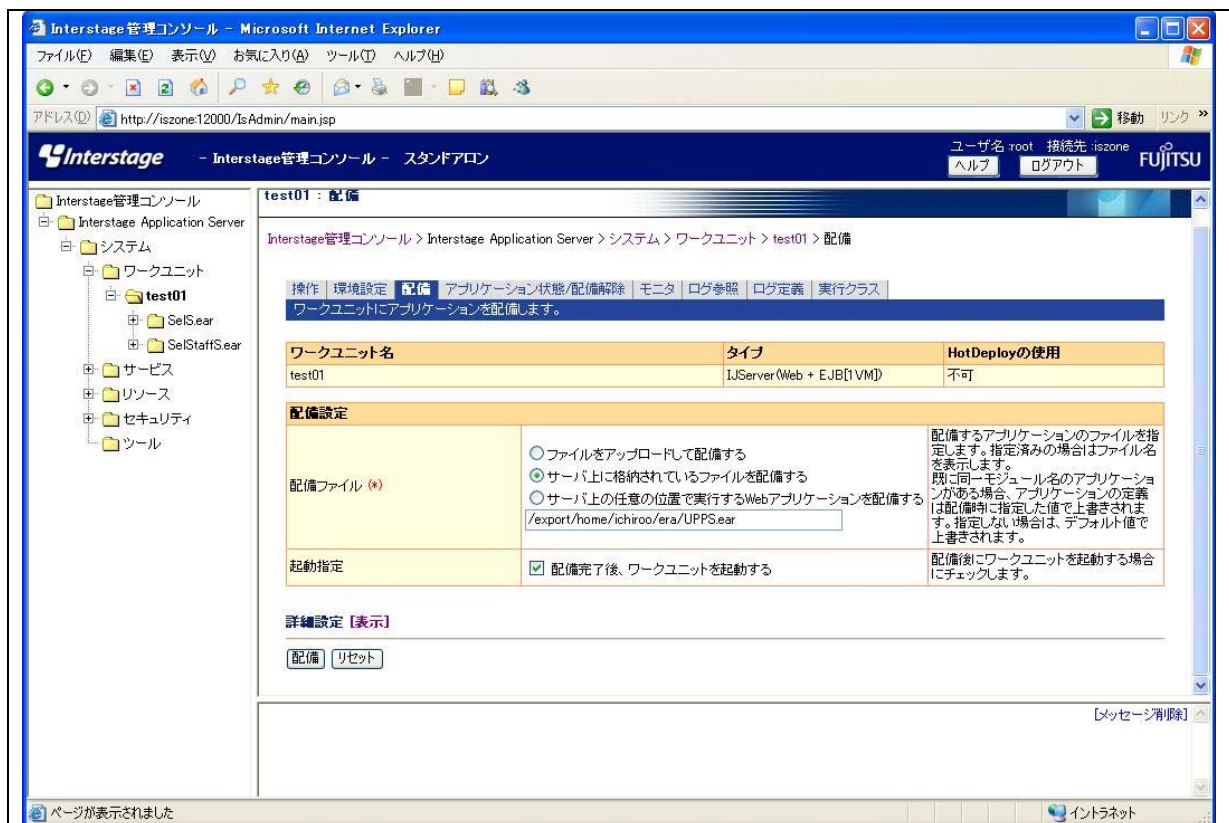
----- 途中省略 -----



[loading /export/home/oracle/isj2ee.jar(javax/ejb/RemoveException.class)]
[wrote WEB-INF/classes/com/mypackage/UPPS/UPPSSessionMonitor.class]
[total 259ms]
adding: WEB-INF/web.xml(in = 898) (out= 374)(deflated 58%)
adding: WEB-INF/weblogic.xml(in = 522) (out= 260)(deflated 50%)
adding: META-INF/Manifest.mf(in = 79) (out= 78)(deflated 1%)
adding: WEB-INF/lib/mfejblib.jar(in = 2982) (out= 1950)(deflated 34%)
adding: UPPSMain.jsp(in = 499) (out= 311)(deflated 37%)
adding: UPP.jsp(in = 1810) (out= 665)(deflated 63%)
adding: WEB-INF/classes/com/mypackage/UPPS/UPPSServlet.class(in = 8850) (out= 3247)(deflated 63%)
adding: WEB-INF/classes/com/mypackage/UPPS/UPPSJspBean.class(in = 2449) (out= 823)(deflated 66%)
adding: WEB-INF/classes/com/mypackage/UPPS/UPPSSessionMonitor.class(in = 1561) (out= 758)(deflated 51%)
added manifest
adding: UPPS.jar(in = 3792) (out= 3168)(deflated 16%)
adding: UPPS.war(in = 9846) (out= 9029)(deflated 8%)
adding: mfejblib.jar(in = 2982) (out= 1950)(deflated 34%)
adding: META-INF/application.xml(in = 419) (out= 252)(deflated 39%)
adding: META-INF/weblogic-application.xml(in = 301) (out= 184)(deflated 38%)
```

- 17) 生成された UPPS.ear を Interstage 管理コンソールより配置する。

Interstage 管理コンソール > Interstage Application Server > システム > ワークユニット > SampleServer
> 配備 で配備ファイルに生成された SelS.ear を指定して配置ボタンをクリックします。



- 18) 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。

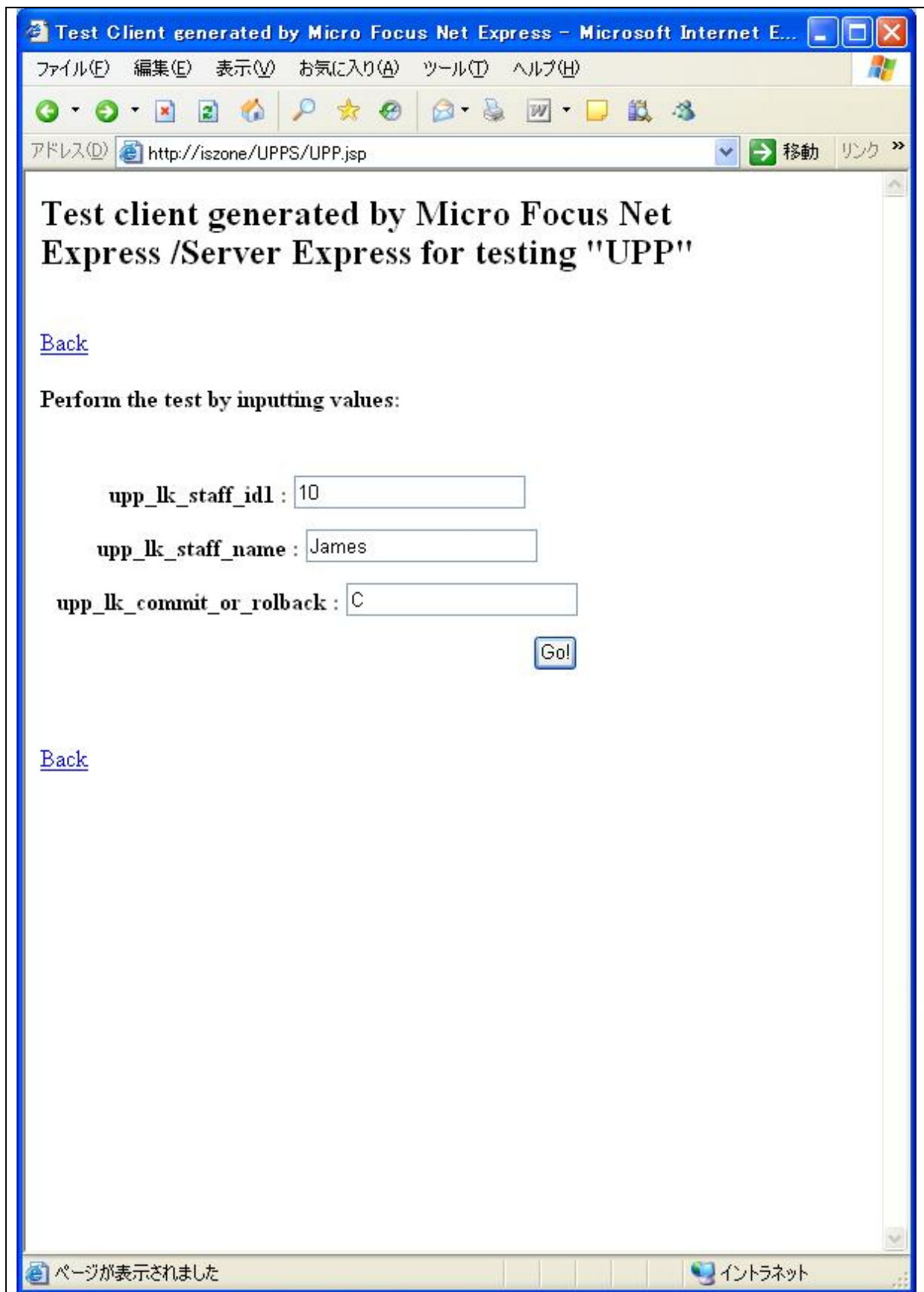
```
SQL> select * from staff;
```

ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
10	Smith	333	Mgr	7	18357.5	
20	TAKESI	333	Sales	8	18171.25	612.45
30	Marengni	38	Mgr	5	17506.75	
40	Brien	38	Sales	6	18006	846.55

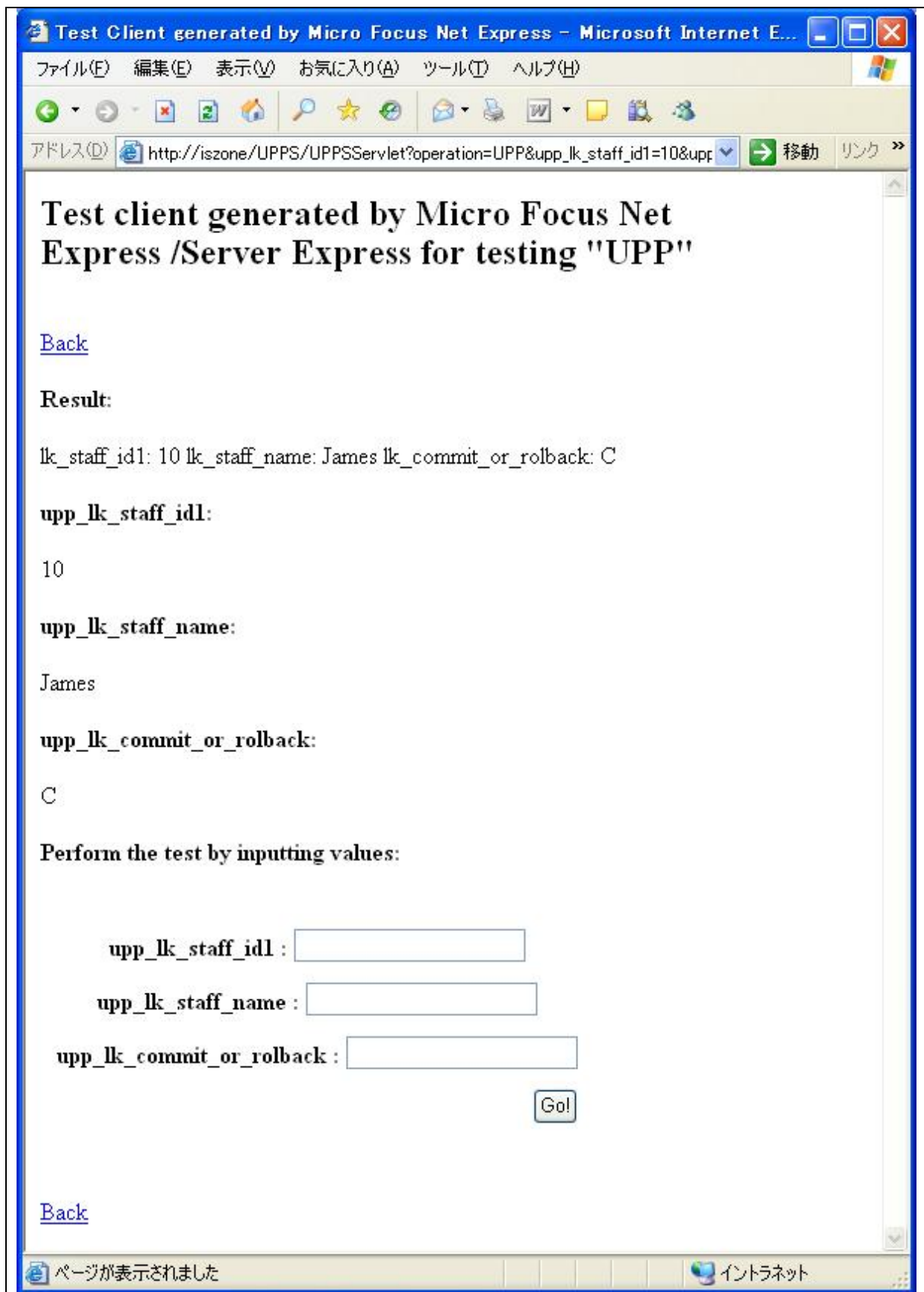
```
SQL>
```

現在は Smith さんの名前が格納されています。

- 19) ブラウザから、http://iszone/UPPS/UPP.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。



20) 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。



21) 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	DA	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。

- 22) 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“ABC”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。

Test client generated by Micro Focus Net Express /Server Express for testing "UPP"

[Back](#)

Perform the test by inputting values:

upp_lk_staff_id1 :

upp_lk_staff_name :

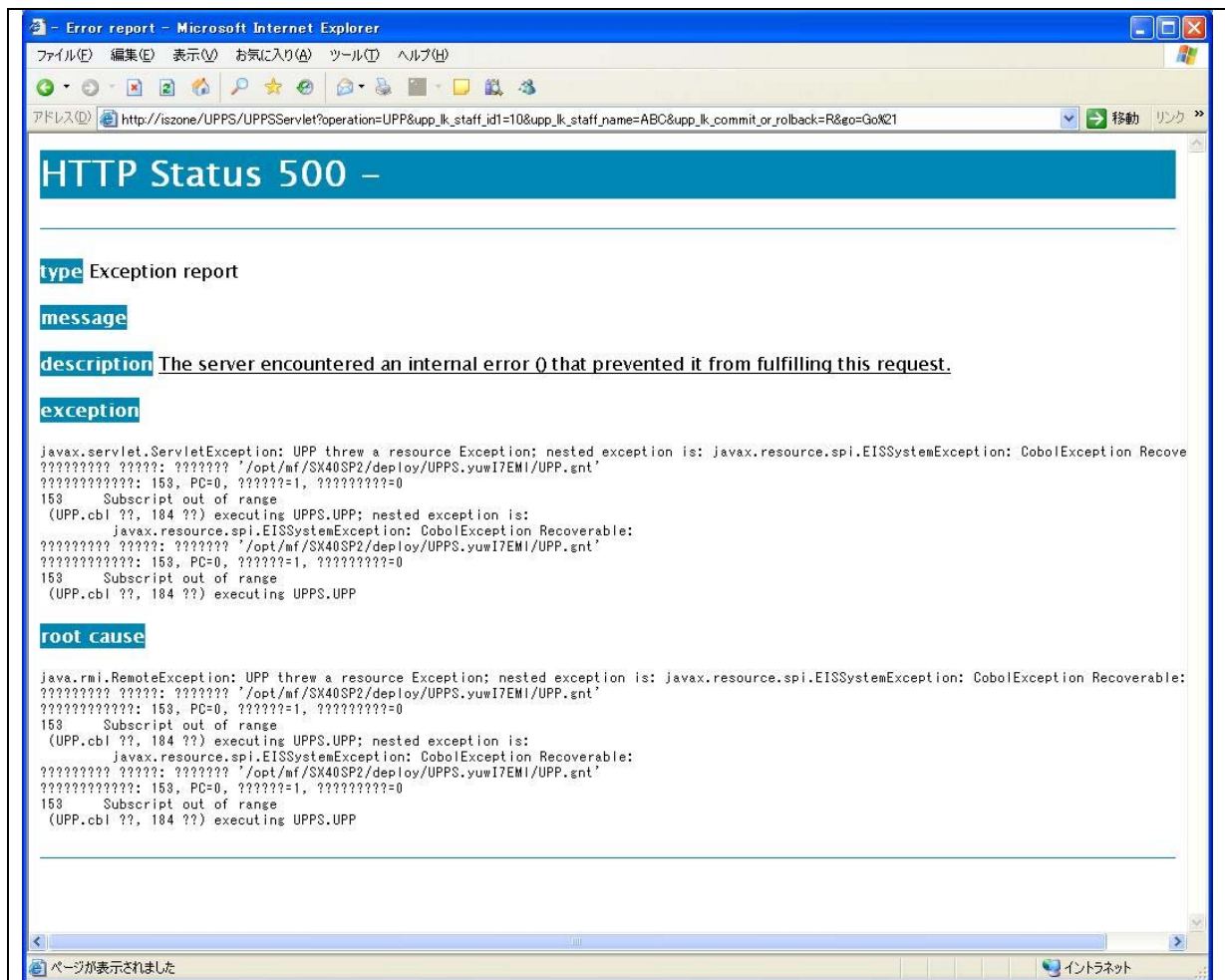
upp_lk_commit_or_rollback :

[Back](#)

ページが表示されました

イントラネット

- 23) COBOL サービス中で意図的なアプリケーション例外が発生し、ブラウザに以下のようにエラーが返ります。



- 24) これは、リソースアダプターがスローした例外を EJB がキャッチしたことを示しています。ES コンソールには、下記のようにエラーが返ります。

```

061019 16363279      628 ESDemo  CASKC0027E Error executing service 'UPPS.UPP'
目的コード エラ: ファイル '/opt/mf/SX40SP2/deploy/UPPS.yuwI7EMI/UPP.gnt'
エラーコード: 153, PC=0, コール=1, セグメント=0
153 添字が指定範囲外になっている
(UPP.cbl 内, 184 行) 16:36:32

```

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

```

- 25) 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>

26) 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