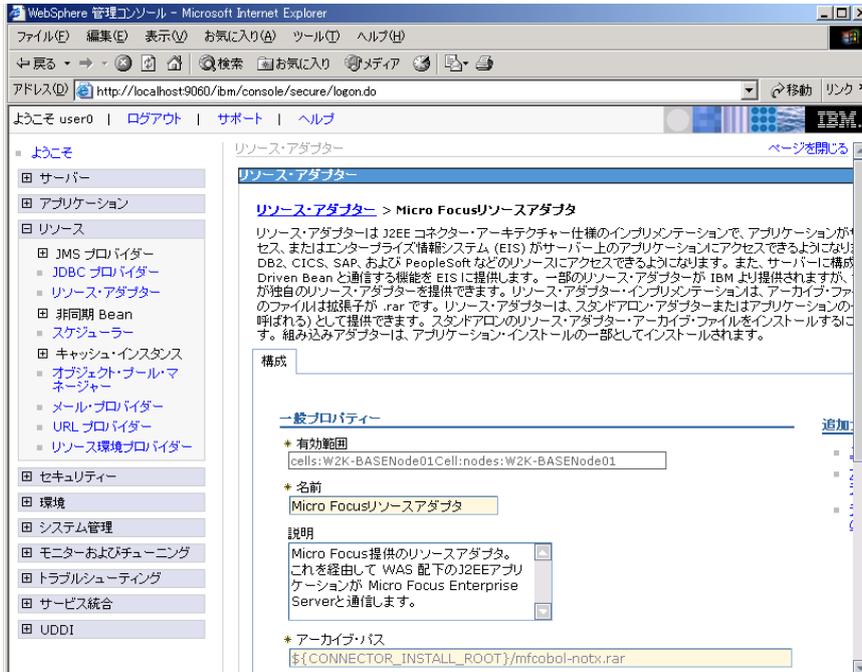


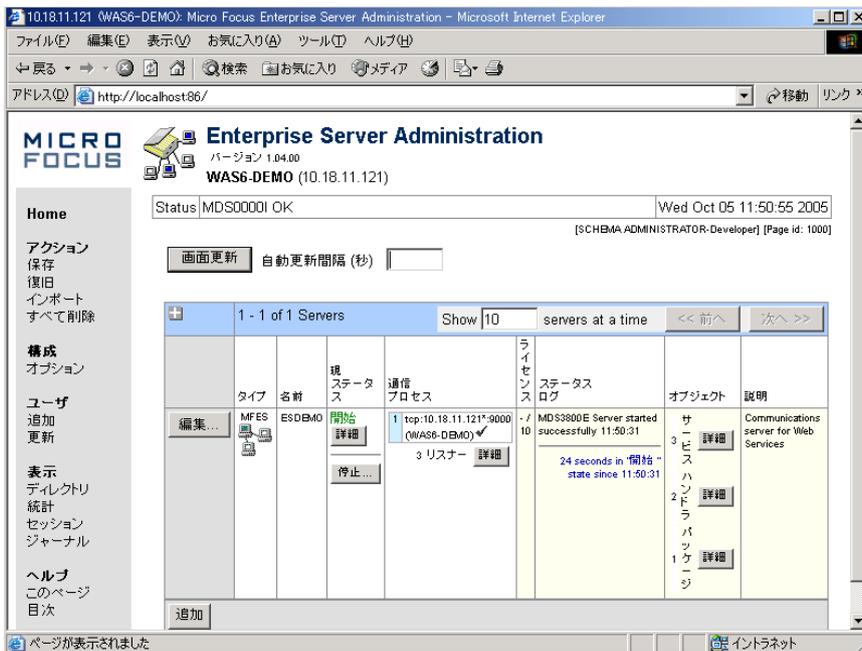
Micro Focus Net Express 4.0J
IBM WebSphere Application Server 6.0 動作検証報告書 別紙

補足1. 通常の COBOL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、Net Expressヘルプの記載の通りインストールします。以下のように登録されます：



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 COBOLプログラムを用意します。テスト用のデータファイル CUST.dat はあらかじめデータを投入して

用意しておきます。

ReadCust.cbl

```
FILE-CONTROL.  
  SELECT CUST-MASTER ASSIGN TO "CUST.dat"  
  ORGANIZATION INDEXED RECORD KEY FS-CUSTID  
  ACCESS MODE RANDOM.  
DATA          DIVISION.  
FILE          SECTION.  
FD CUST-MASTER.  
01 CUST-REC.  
  05 FS-CustId      PIC X(4)  COMP-5.  
  05 FS-CustName    PIC X(30).  
  05 FS-CustCompany PIC X(30).  
  05 FS-CustEmail   PIC X(30).  
LINKAGE      SECTION.  
01 CustId     PIC X(4)  COMP-5.  
01 CustName   PIC X(30).  
01 CustCompany PIC X(30).  
01 CustEmail  PIC X(30).  
PROCEDURE    DIVISION  
  USING CustId CustName CustCompany CustEmail.  
1.  
  OPEN       I-O CUST-MASTER.  
  MOVE       CustId      TO FS-CustId.  
  READ       CUST-MASTER INVALID CONTINUE  
            END-READ.  
  CLOSE     CUST-MASTER.  
  MOVE      FS-CustName  TO CustName.  
  MOVE      FS-CustCompany TO CustCompany.  
  MOVE      FS-CustEmail TO CustEmail.  
  EXIT      PROGRAM.
```

4) Net Express コマンドプロンプトを開き、上記の例題プログラムを以下のようにコンパイルします。

```
C:¥ReadCust>cobolc ReadCust.cbl gnt;  
Micro Focus Net Express V4  
Version 4.0.38 Copyright (C) 1984-2004 Micro Focus International Ltd.  
URN AXCGG/AA0/00000  
* チェック終了 : エラーはありません - コード生成を開始します  
* Generating ReadCust  
* Data:          480   Code:          984   Literals:          384  
  
C:¥ ReadCust >
```

5) 同じ Net Express コマンドプロンプトから、以下のようにデフォルトマッピングを作成します。

```
C:¥ReadCust>imtkmake -defmap src=ReadCust.cbl service=ReadCustS type=ejb  
Micro Focus Interface Mapping Toolkit v1.0.00  
Copyright (C) 2004 Micro Focus International Ltd. All rights reserved.  
  
C:¥ReadCust>
```

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、ReadCust.xml と ReadCustS.xml が作成されます。

- 6) 同じ Net Express コマンドプロンプトから、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。

```
C:\>ReadCust>set CLASSPATH=%CLASSPATH%;C:\Program Files\IBM\WebSphere\AppServer1\lib\j2ee.jar

C:\>ReadCust>imtkmake -generate service=ReadCustS type=ejb
Micro Focus Interface Mapping Toolkit v1.0.00
Copyright (C) 2004 Micro Focus International Ltd. All rights reserved.
"C:\Program Files\IBM\WebSphere\AppServer1\java\bin\javac.exe" -g:none -verbose
com\mypackage\ReadCustS\ReadCustSHome.java
[com\mypackage\ReadCustS\ReadCustSHome.java を構文解析開始]
[230ms で構文解析完了]
[C:\Program Files\IBM\WebSphere\AppServer1\lib\j2ee.jar (javax/ejb/EJBHome.class)
を読み込み中]

..... 途中省略 .....

[com\mypackage\ReadCustS\ReadCustSBean.class を書き込み完了]
[合計 3104ms]
マニフェストが追加されました。
META-INF/ejb-jar.xml を追加中です。(入 = 1372) (出 = 512) (62% 収縮されました)
META-INF/weblogic-ejb-jar.xml を追加中です。(入 = 839) (出 = 339) (59% 収縮されました)
com/mypackage/ReadCustS/ReadCustS.class を追加中です。(入 = 265) (出 = 188) (29% 収縮されました)
com/mypackage/ReadCustS/ReadCustSBean.class を追加中です。(入 = 3436) (出 = 1513) (55% 収縮されました)
com/mypackage/ReadCustS/ReadCustSHome.class を追加中です。(入 = 257) (出 = 176) (31% 収縮されました)
minizip.exe: Micro Focus zip utility version 1.1.1
Based on MiniZip 0.15 by Gilles Vollant
Copyright (C) 2002-2003 Micro Focus International Limited

C:\>ReadCust>
```

これによって、ReadCustS.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：

- ReadCustS.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
- ReadCustS.jar : ラッパーEJB のパッケージ

- 7) 同じ Net Express コマンドプロンプトから、以下のように ESDEMO へディプロイします。

```
C:\>ReadCust2>imtkmake -deploy carname=ReadCustS.deploy\ReadCustS.car
Micro Focus Interface Mapping Toolkit v1.0.00
Copyright (C) 2004 Micro Focus International Ltd. All rights reserved.
Found 1 deployment services:

1. Deployer (Deployment file-upload service) is Available at tcp:10.18.11.121:1206
Listener (Basic HTTP web server) is Started
Server ESDEMO (Communications server for Web Services) on -unknown- (-unknown)
```

n-) is Started

Select a service provider (1-1) or 0 to exit: 1

Sending ReadCustS.deploy¥ReadCustS.car to ESDEMO's Deployer at tcp:10.18.11.121:1206...

Received 52 bytes:

http://10.18.11.121:1206/uploads/ReadCustS.EpQq0_i4/

Server returned 202

Retrieving deployment log file http://10.18.11.121:1206/uploads/ReadCustS.EpQq0_i4/deploylog.txt...

1000 (2005/10/05 13:11:06): mfdepinst 1.2.4 starting deployment of COBOL archive "ReadCustS.car"

0010 (2005/10/05 13:11:06): Extracting and parsing manifest file

0011 (2005/10/05 13:11:06): Processing Manifest element in the manifest file

0012 (2005/10/05 13:11:06): Processing "Application" section of Manifest

0012 (2005/10/05 13:11:06): Processing "FileList" section of Manifest

2003 (2005/10/05 13:11:06): File pathname "C:¥ReadCust2¥ReadCustS.deploy¥ReadCustS.idt" is absolute; reducing to filename "ReadCustS.idt"

0012 (2005/10/05 13:11:06): Processing "Service" section of Manifest

0019 (2005/10/05 13:11:06): Manifest processing completed

0020 (2005/10/05 13:11:06): Adding service and package objects to directory

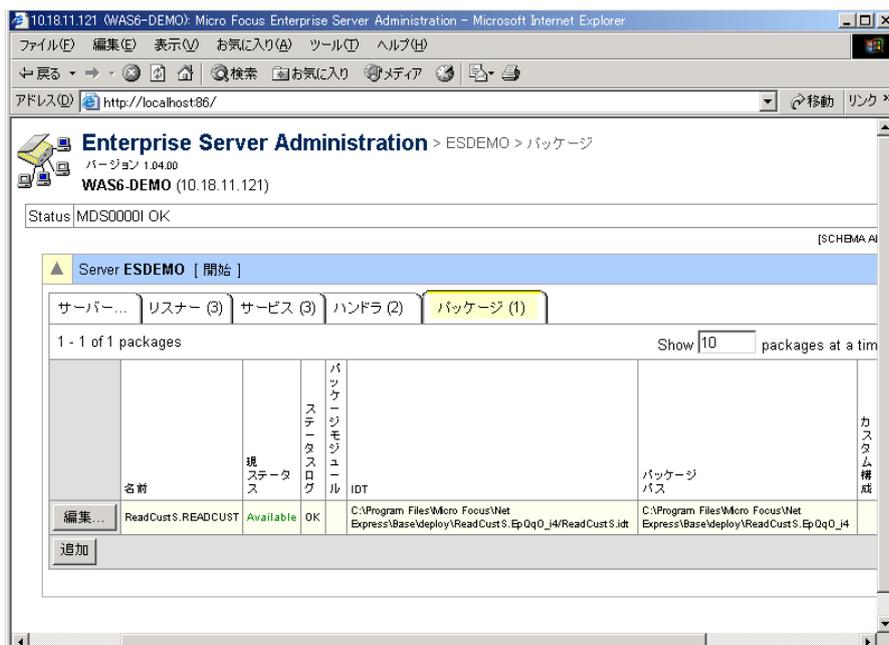
0030 (2005/10/05 13:11:13): ES server "ESDEMO" notified service "ReadCustS.READCUST" is available

0002 (2005/10/05 13:11:13): Installation of package "ReadCustS.car" finished with 1 warnings

Deployment completed with warnings

C:¥ReadCust>

8) 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。



9) 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 コードと、索引編成データファイルをコピーします。

10) 同じ Net Express コマンドプロンプトから、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。

```
C:\¥ReadCust>imtkmake -genclient service=ReadCustS type=ejb appserver="WebSphere
5. x"

..... 途中省略 .....

[C:\¥Program Files¥IBM¥WebSphere¥AppServer1¥lib¥jee.jar (javax/ejb/RemoveExceptio
n.class) を読み込み中]
[WEB-INF¥classes¥com¥mypackage¥ReadCustS¥ReadCustSSessionMonitor.class を書き込
み完了]
[合計 2904ms]
WEB-INF/web.xml を追加中です。(入 = 1078) (出 = 420) (61% 収縮されました)
WEB-INF/ibm-web-bnd.xmi を追加中です。(入 = 517) (出 = 255) (50% 収縮されました)
WEB-INF/ibm-web-ext.xmi を追加中です。(入 = 296) (出 = 176) (40% 収縮されました)
META-INF/Manifest.mf を追加中です。(入 = 87) (出 = 85) (2% 収縮されました)
WEB-INF/lib/mfejblib.jar を追加中です。(入 = 2984) (出 = 1958) (34% 収縮されまし
た)
ReadCustSMain.jsp を追加中です。(入 = 549) (出 = 316) (42% 収縮されました)
READCUST.jsp を追加中です。(入 = 2193) (出 = 697) (68% 収縮されました)
WEB-INF/classes/com/mypackage/ReadCustS/ReadCustServlet.class を追加中です。(入
= 9134) (出 = 3300) (63% 収縮されました)
WEB-INF/classes/com/mypackage/ReadCustS/ReadCustSjspBean.class を追加中です。(入
= 2927) (出 = 907) (69% 収縮されました)
WEB-INF/classes/com/mypackage/ReadCustS/ReadCustSSessionMonitor.class を追加中だ
す。(入 = 1631) (出 = 762) (53% 収縮されました)
マニフェストが追加されました。
ReadCustS.jar を追加中です。(入 = 5045) (出 = 4067) (19% 収縮されました)
ReadCustS.war を追加中です。(入 = 10480) (出 = 9510) (9% 収縮されました)
mfejblib.jar を追加中です。(入 = 2984) (出 = 1958) (34% 収縮されました)
META-INF/application.xml を追加中です。(入 = 506) (出 = 283) (44% 収縮されまし
た)

META-INF/ibm-application-bnd.xmi を追加中です。(入 = 392) (出 = 191) (51% 収縮さ
れました)
META-INF/ibm-application-ext.xmi を追加中です。(入 = 817) (出 = 265) (67% 収縮さ
れました)

C:\¥ReadCust>
```

これによって、ReadCustS.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：

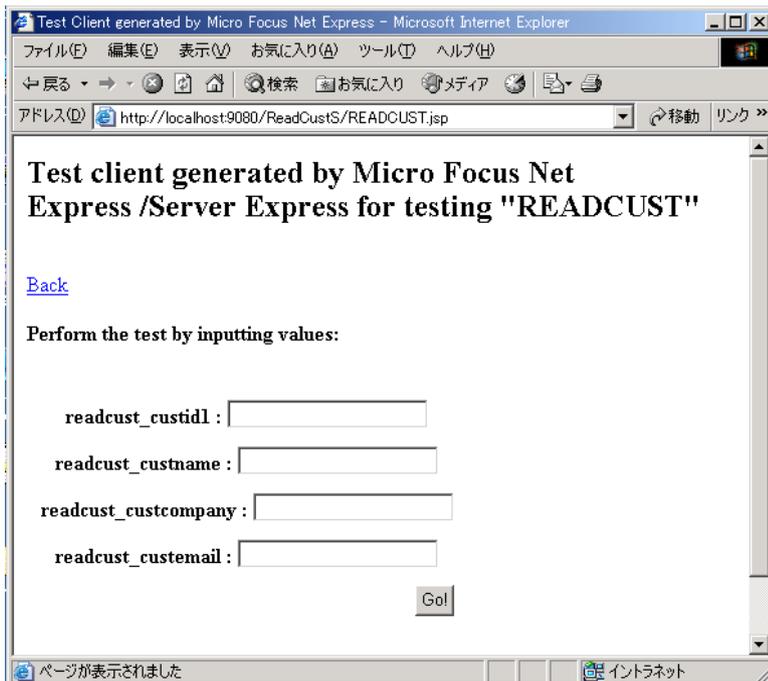
ReadCustS.ear : ラッパーEJBのパッケージと、自動生成されたJSPクライアント含む、エンタープライズアーカイブ

11) 生成された ReadCustS.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。

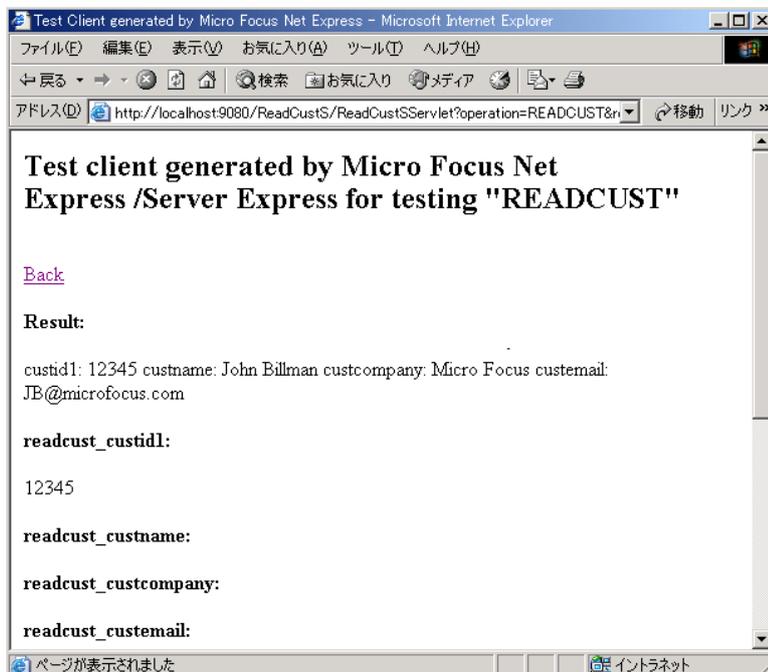
```
インストール中...
アプリケーションに Enterprise Bean がある場合、EJB デプロイメント処理に数分間かかる場合があります。処理が完了するまでは、構成を保管しないでください。
EJB デプロイメント処理に関する特定情報については、アプリケーションがデプロイされている Deplo
yment Manager またはサーバーの SystemOut.log をチェックしてください。
ADMA50161: ReadCustS のインストールが開始されました。
ADMA50671: アプリケーション ReadCustS のリソース検証は正常に完了しました。
```

```
ADMA5058I: アプリケーションとモジュールのバージョンは、デプロイメント・ターゲットのターゲットと検証されました。
ADMA5018I: EJBDeploy コマンドが、エンタープライズ・アーカイブ (EAR) ファイル
C:\Program
Files\IBM\WebSphere\AppServer1\profiles\default\wstemp\111578565\upload\ReadCustS. ear
について実行中です。
Starting workbench.
Creating the project.
Building: /ReadCustS_EJB
Deploying jar ReadCustS_EJB
Generating deployment code
Refreshing: /ReadCustS_EJB/ejbModule.
Building: /ReadCustS_EJB
Invoking RMIC.
Generating DDL
Building: /WsaEJBDeployUtility
Writing output file
Shutting down workbench.
EJBDeploy complete.
0 Errors, 0 Warnings, 0 Informational Messages
ADMA5007I: EJBDeploy コマンドが C:\WINNT\TEMP\app_106bf177a44\dpl\dpl_ReadCustS. ear で完了
しました。
ADMA5005I: アプリケーション ReadCustS が WebSphere Application Server リポジトリに構成さ
れます。
ADMA5053I: インストール済みオプション・パッケージのライブラリー参照が作成されます。
ADMA5005I: アプリケーション ReadCustS が WebSphere Application Server リポジトリに構成さ
れます。
ADMA5001I: アプリケーション・バイナリーは C:\Program Files\IBM\WebSphere\AppServer1
/profiles/default\wstemp\111578565\workspace\cells\W2K-BASENode01Cell\applications\
ReadCustS. ear\ReadCustS. ear に保管されます。
ADMA5005I: アプリケーション ReadCustS が WebSphere Application Server リポジトリに構成さ
れます。
SECJ0400I: アプリケーション ReadCustS が appContextIDForSecurity 情報で正常に更新されまし
た。
ADMA5011I: アプリケーション ReadCustS の一時ディレクトリーのクリーンアップが完了しました。
ADMA5013I: アプリケーション ReadCustS は正常にインストールされました。
アプリケーション ReadCustS は正常にインストールされました。
アプリケーションを開始するには、最初にマスター構成への変更を保管する必要があります。
```

- 12) WebSphere 管理コンソールから、インストールしたエンタープライズアプリケーション ReadCustS を開始しま
す。
- 13) ブラウザから <http://localhost:9080/ReadCustS/READCUST.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。



- 14) 最初のテキストボックスに、用意した索引編成ファイル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、索引編成ファイル中のデータが 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. DB2 照会プログラムのデプロイと、EJB経由の JCA呼び出し

- 1) 補足1で使用したインストール済みのリソースアダプタと、ESDEMOをそのまま使用します。
- 2) 以下の COBOLプログラムを用意します。DB2に添付のsampleデータベースを使用しますので、作成しておきます。

SelStaff.cbl

```
WORKING-STORAGE SECTION.  
  copy "sqlca.cbl".  
  EXEC SQL BEGIN DECLARE SECTION END-EXEC.  
01 STAFF-ID          PIC X(4)  COMP-5.  
01 STAFF-NAME        PIC X(10).  
01 STAFF-DEPT        PIC X(4)  COMP-5.  
01 STAFF-SALARY      PIC S9(5)V9(2) COMP-3.  
  EXEC SQL END DECLARE SECTION END-EXEC.  
LINKAGE SECTION.  
01 LK-STAFF-ID       PIC X(4)  COMP-5.  
01 LK-STAFF-NAME     PIC X(10).  
01 LK-STAFF-DEPT     PIC X(4)  COMP-5.  
01 LK-STAFF-SALARY   PIC S9(11) COMP-3.  
PROCEDURE DIVISION USING LK-STAFF-ID LK-STAFF-NAME  
                        LK-STAFF-DEPT LK-STAFF-SALARY.  
1.  MOVE LK-STAFF-ID TO STAFF-ID  
  EXEC SQL CONNECT TO sample END-EXEC  
  EXEC SQL SELECT NAME, DEPT, SALARY  
    INTO :STAFF-NAME, :STAFF-DEPT, :STAFF-SALARY  
  FROM STAFF  
  WHERE ID=:STAFF-ID END-EXEC.  
  EXEC SQL CONNECT RESET END-EXEC.  
  MOVE STAFF-NAME TO LK-STAFF-NAME.  
  MOVE STAFF-DEPT TO LK-STAFF-DEPT.  
  MOVE STAFF-SALARY TO LK-STAFF-SALARY.  
  EXIT PROGRAM.
```

- 3) DB2 コマンドウィンドウを開きます。ここで COBOL のコンパイルやデプロイの作業も行うために、環境変数の設定が必要です。Net Express のインストール先の %Bin ディレクトリに、SETENV.bat というバッチがありますので、これを作業ディレクトリにコピーし、

```
CALL SETENV.bat
```

を実行すれば、以降 DB2 環境の中で COBOL の作業も行えるようになります。

- 4) DB2 コンパイル指令をデフォルトで設定しておくために、カレントディレクトリに cobol.dir というテキストファイルを作成し、その中に

```
DB2(db=sample,bind)
```

を記述しておきます。

- 5) コマンドプロンプトから、以下のようにデフォルトマッピングを作成します。

```
C:¥SelStaff>imtkmake -defmap src=SelStaff.cbl service=SelStaffS type=ejb
Micro Focus Interface Mapping Toolkit v1.0.00
Copyright (C) 2004 Micro Focus International Ltd. All rights reserved.

C:¥SelStaff>
```

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、SelStaff.xml と SelStaffS.xml が作成されます。

- 6) 同じコマンドプロンプトから、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。

```
C:¥SelStaff>set CLASSPATH=%CLASSPATH%;C:¥Program Files¥IBM¥WebSphere¥AppServer1¥
lib¥j2ee.jar

C:¥SelStaff>imtkmake -generate service=SelStaffS type=ejb
Micro Focus Interface Mapping Toolkit v1.0.00
Copyright (C) 2004 Micro Focus International Ltd. All rights reserved.
"C:¥Program Files¥IBM¥WebSphere¥AppServer1¥java¥bin¥javac.exe" -g:none -verbose
com¥mypackage¥SelStaffS¥SelStaffSHome.java
[com¥mypackage¥SelStaffS¥SelStaffSHome.java を構文解析開始]
[180ms で構文解析完了]
[C:¥Program Files¥IBM¥WebSphere¥AppServer1¥lib¥j2ee.jar (javax/ejb/EJBHome.class)
を読み込み中]

..... 途中省略 .....

[com¥mypackage¥SelStaffS¥SelStaffSBean.class を書き込み完了]
[合計 2564ms]
マニフェストが追加されました。
META-INF/ejb-jar.xml を追加中です。(入 = 1372) (出 = 512) (62% 収縮されました)
META-INF/weblogic-ejb-jar.xml を追加中です。(入 = 839) (出 = 340) (59% 収縮されま
した)
com/mypackage/SelStaffS/SelStaffS.class を追加中です。(入 = 231) (出 = 185) (19%
収縮されました)
com/mypackage/SelStaffS/SelStaffSBean.class を追加中です。(入 = 3453) (出 = 1539
) (55% 収縮されました)
com/mypackage/SelStaffS/SelStaffSHome.class を追加中です。(入 = 257) (出 = 177) (
31% 収縮されました)
minizip.exe: Micro Focus zip utility version 1.1.1
Based on MiniZip 0.15 by Gilles Vollant
Copyright (C) 2002-2003 Micro Focus International Limited

C:¥SelStaff>
```

これによって、SelStaffS.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：

SelStaffS.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
SelStaffS.jar : ラッパーEJB のパッケージ

- 7) 同じ Net Express コマンドプロンプトから、以下のように ESDEMO へディプロイします。

```
C:¥SelStaff>imtkmake -deploy carname=SelStaffS.deploy¥SelStaffS.car
Micro Focus Interface Mapping Toolkit v1.0.00
```

Copyright (C) 2004 Micro Focus International Ltd. All rights reserved.
Found 1 deployment services:

1. Deployer (Deployment file-upload service) is Available at tcp:10.18.11.121:1206
Listener (Basic HTTP web server) is Started
Server ESDEMO (Communications server for Web Services) on -unknown- (-unknown-) is Started

Select a service provider (1-1) or 0 to exit: 1

Sending SelStaffS.deploy¥SelStaffS.car to ESDEMO's Deployer at tcp:10.18.11.121:1206...

Received 52 bytes:

http://10.18.11.121:1206/uploads/SelStaffS.AB23hYW6/

Server returned 202

Retrieving deployment log file http://10.18.11.121:1206/uploads/SelStaffS.AB23hYW6/deploylog.txt...

1000 (2005/10/05 15:23:38): mfdepinst 1.2.4 starting deployment of COBOL archive "SelStaffS.car"

0010 (2005/10/05 15:23:38): Extracting and parsing manifest file

0011 (2005/10/05 15:23:38): Processing Manifest element in the manifest file

0012 (2005/10/05 15:23:38): Processing "Application" section of Manifest

0012 (2005/10/05 15:23:38): Processing "FileList" section of Manifest

2003 (2005/10/05 15:23:38): File pathname "C:¥SelStaff¥SelStaffS.deploy¥SelStaffS.idt" is absolute; reducing to filename "SelStaffS.idt"

0012 (2005/10/05 15:23:38): Processing "Service" section of Manifest

0019 (2005/10/05 15:23:38): Manifest processing completed

0020 (2005/10/05 15:23:38): Adding service and package objects to directory

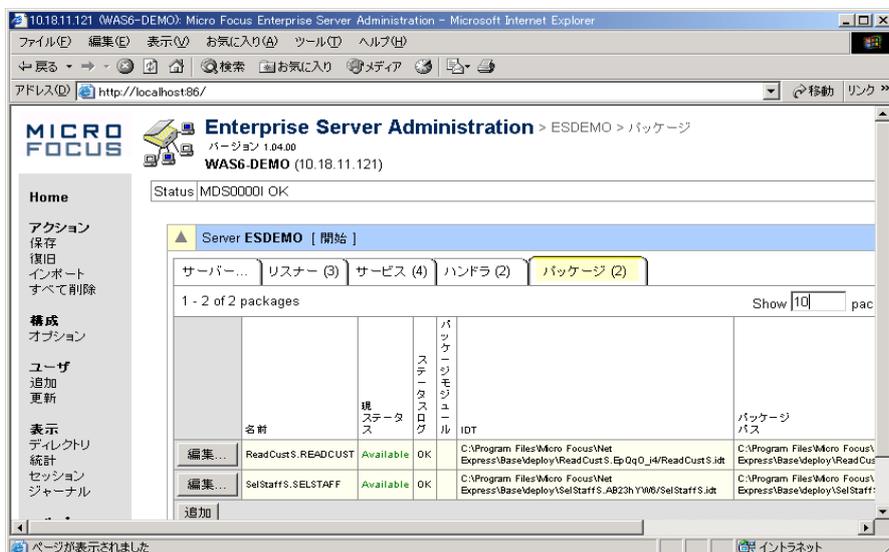
0030 (2005/10/05 15:23:45): ES server "ESDEMO" notified service "SelStaffS.SELSTAFF" is available

0002 (2005/10/05 15:23:45): Installation of package "SelStaffS.car" finished with 1 warnings

Deployment completed with warnings

C:¥ReadCust>

8) 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。



- 9) 上記の例題プログラムを以下のように DB2 指令でコンパイルします。

```
C:¥SelStaff>cobolc SelStaff.cbl DB2(db=sample,bind) gnt:
Micro Focus Net Express V4
Version 4.0.38 Copyright (C) 1984-2005 Micro Focus International Ltd.
URN AXCGG/AA0/00000
* Micro Focus SQL External Compiler Module for IBM DB2 UDB
* Version 4.0.05 (C) copyright 1997-2005 Micro Focus International Ltd.
* チェック終了 : エラーはありません - コード生成を開始します
* Generating SelStaff
* Data:      1312      Code:      2936      Literals:      728

C:¥SelStaff>
```

- 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。DB2 プログラムは、コンパイル時にアクセスプランを定義するバインドが行われます。一方マッピング定義の際にも内部的にコンパイラが起動されているため、そこでもバインドがなされています。このため、上記のように、マッピングを完了した後で改めてコンパイルを行い、コンパイル済みのコードをコピーしておくことが重要です。

- 11) コマンドプロンプトから、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。

```
C:¥SelStaff>imtkmake -genclient service=SelStaffS type=ejb appserver="WebSphere
5. x"
Micro Focus Interface Mapping Toolkit v1.0.00
Copyright (C) 2004 Micro Focus International Ltd. All rights reserved.
    1 個のファイルをコピーしました。
    1 個のファイルをコピーしました。
"C:¥Program Files¥IBM¥WebSphere¥AppServer1¥java¥bin¥javac.exe" -g:none -verbose
com¥mypackage¥SelStaffS¥SelStaffSHome.java
[com¥mypackage¥SelStaffS¥SelStaffSHome.java を構文解析開始]
[160ms で構文解析完了]

..... 途中省略 .....

[WEB-INF¥classes¥com¥mypackage¥SelStaffS¥SelStaffSSessionMonitor.class を書き込
み完了]
[合計 2333ms]
WEB-INF/web.xml を追加中です。(入 = 1078) (出 = 418) (61% 収縮されました)
WEB-INF/ibm-web-bnd.xmi を追加中です。(入 = 517) (出 = 255) (50% 収縮されました)
WEB-INF/ibm-web-ext.xmi を追加中です。(入 = 296) (出 = 176) (40% 収縮されました)
META-INF/Manifest.mf を追加中です。(入 = 87) (出 = 85) (2% 収縮されました)
WEB-INF/lib/mfejlib.jar を追加中です。(入 = 2984) (出 = 1958) (34% 収縮されまし
た)
SelStaffMain.jsp を追加中です。(入 = 544) (出 = 327) (39% 収縮されました)
SELSTAFF.jsp を追加中です。(入 = 2273) (出 = 701) (69% 収縮されました)
WEB-INF/classes/com/mypackage/SelStaffS/SelStaffServlet.class を追加中です。(入
= 9229) (出 = 3327) (63% 収縮されました)
WEB-INF/classes/com/mypackage/SelStaffS/SelStaffJspBean.class を追加中です。(入
= 2876) (出 = 981) (65% 収縮されました)
WEB-INF/classes/com/mypackage/SelStaffS/SelStaffSSessionMonitor.class を追加中デ
ス。(入 = 1631) (出 = 761) (53% 収縮されました)
マニフェストが追加されました。
SelStaffS.jar を追加中です。(入 = 5069) (出 = 4093) (19% 収縮されました)
```

```
SELSTAFFS.war を追加中です。(入 = 10593) (出 = 9623) (9% 収縮されました)
mfejblib.jar を追加中です。(入 = 2984) (出 = 1958) (34% 収縮されました)
META-INF/application.xml を追加中です。(入 = 506) (出 = 284) (43% 収縮されました)

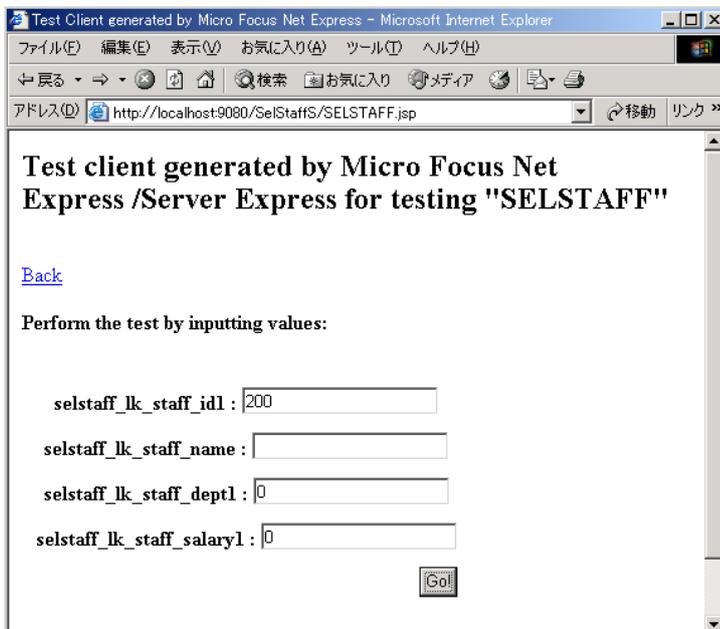
META-INF/ibm-application-bnd.xmi を追加中です。(入 = 392) (出 = 191) (51% 収縮さ
れました)
META-INF/ibm-application-ext.xmi を追加中です。(入 = 817) (出 = 265) (67% 収縮さ
れました)

C:\SelStaff>
```

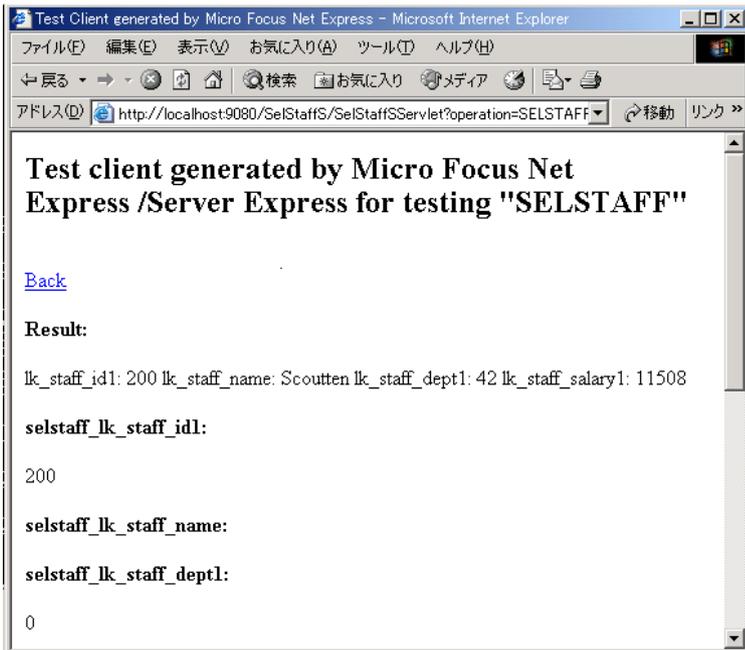
これによって、SelStaffS.deploy ディレクトリ下に以下のパッケージが作成されています：

SelStaffS.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 にデプロイします

- 12) 生成された SelStaffS.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。管理コンソールでインストールが完了したことを確認します。
- 13) WebSphere 管理コンソールから、インストールしたエンタープライズアプリケーション SelStaffS を開始します。
- 14) ブラウザから <http://localhost:9080/SelStaffS/SELSTAFF.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。

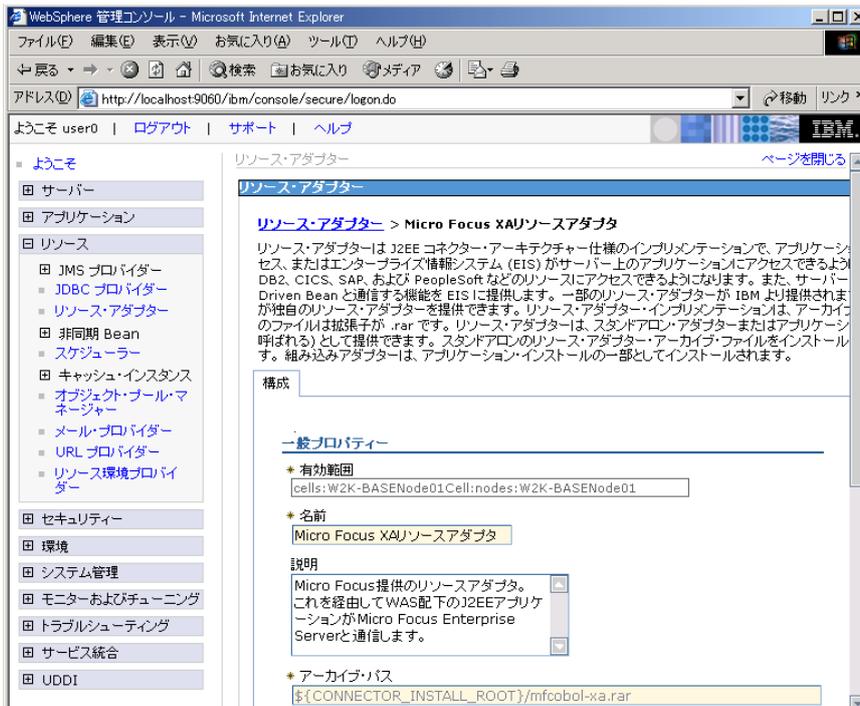


- 15) 最初のテキストボックスに 200、3 番目と 4 番目のテキストボックスに 0 を入力し、[Go!] をクリックします。3 番目と 4 番目のパラメタは入力パラメタではありませんが、数字型であるため何らかの数字を入力しないと、COBOL マッピングの型変換でエラーが発生します。
- 16) COBOL サービスが呼び出され、DB2 テーブルのデータが JSP 画面上に表示されます。



補足3. DB2 更新プログラムのデプロイと、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 を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、Net Expressヘルプの記載の通りインストールします。以下のように登録されます：



- 3) 新しいリソースアダプタを有効にするために、WebSphere Application Server を再起動します。
- 4) Micro Focus Enterprise Server に、DB2 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、Net Express ヘルプに記載された手順でスイッチモジュールをビルドします。Net Express のインストール先ディレクトリ下の Base¥Source¥EnterpriseServer¥XA の下のファイルをすべて作業ディレクトリにコピーした上で、以下のようにビルドします。

```
C:¥UpdStaff¥XA>call build.bat db2
Ensure that the DB2 LIB directory is included in "C:¥Program Files¥Micro Focus¥Net
Express¥Unix¥DSCHAR¥LIB";"C:¥Program Files¥Micro Focus¥Net Express¥DialogSyst
em¥LIB";"C:¥Program Files¥Micro Focus¥Net Express¥Base¥LIB";C:¥Program Files¥Micro
Focus¥Net Express¥Base¥LIB;C:¥PROGRA~1¥IBM¥SQLLIB¥LIB
Micro Focus Net Express - CBLINK 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.

ESDB2XA.obj
oblls0000067C.obj
    Creating library ESDB2XA.lib and object ESDB2XA.exp

C:¥UpdStaff¥XA>dir
```

```

ドライブ C のボリューム ラベルがありません。
ボリューム シリアル番号は 0000-1641 です

C:\¥UpdStaff¥XA のディレクトリ

2005/10/05 16:57 <DIR>      .
2005/10/05 16:57 <DIR>      ..
2003/04/11 01:38             1,029 BUILD.BAT
2003/04/11 01:38             647 ESDB2XA.CBL
2003/04/11 01:38             882 ESORA8XA.CBL
2003/04/11 01:38             870 ESORA9XA.CBL
2005/10/05 17:09             1,572 ESDB2XA.obj
2005/10/05 17:10             20,480 ESDB2XA.dll
                6 個のファイル                25,480 バイト
                2 個のディレクトリ    9,799,991,296 バイトの空き領域

C:\¥UpdStaff¥XA>

```

以上でスイッチモジュール ESDB2XA.dll がビルドされました。

- 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) Net Express ヘルプの記述と、DB2 XA リソースマネージャのマニュアル記述にしたがって、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作していることを確認してください。

051005 19055982 1772 ESDEMO CASXO0015I XADB2 XA interface initialized successfully 19:05:59

- 8) 以下の COBOL プログラムを用意します。DB2 に添付の sample データベースを使用しますので、作成しておきます。

UpdStaff.cbl

```
WORKING-STORAGE SECTION.  
  copy "sqlca.cbl".  
  EXEC SQL BEGIN DECLARE SECTION END-EXEC.  
01 STAFF-ID          PIC X(4)  COMP-5.  
01 STAFF-NAME       PIC X(10).  
  EXEC SQL END DECLARE SECTION 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-Rollback PIC X.  
PROCEDURE DIVISION USING LK-STAFF-ID LK-STAFF-NAME  
                        LK-Commit-Or-Rollback.  
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.  
    IF LK-Commit-Or-Rollback = '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、その後で意図的にテーブル項目の添え字参照範囲の例外が発生しますので、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) DB2 コマンドウィンドウを開きます。ここで COBOL のコンパイルやディプロイの作業も行うために、環境変数の設定が必要です。Net Express のインストール先の %Bin ディレクトリに、SETENV.bat というバッチがありますので、これを作業ディレクトリにコピーし、

```
CALL SETENV.bat
```

を実行すれば、以降 DB2 環境の中で COBOL の作業も行えるようになります。

- 10) DB2 コンパイル指令をデフォルトで設定しておくために、カレントディレクトリに cobol.dir というテキストファイルを作成し、その中に

```
DB2(db=sample,bind)
```

を記述しておきます。

11) コマンドプロンプトから、以下のようにデフォルトマッピングを作成します。

```
C:\UpdStaff>imtkmake -defmap src=UpdStaff.cbl service=UpdStaffS type=ejb
Micro Focus Interface Mapping Toolkit v1.0.00
Copyright (C) 2004 Micro Focus International Ltd. All rights reserved.

C:\UpdStaff>
```

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、UpdStaff.xml と UpdStaffS.xml が作成されます。

12) 同じコマンドプロンプトから、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。コマンド行にてコンテナ管理トランザクションとして生成することを指定していることに注意してください。

```
C:\UpdStaff>imtkmake -generate service=UpdStaffS type=ejb transaction=container
Micro Focus Interface Mapping Toolkit v1.0.00
Copyright (C) 2004 Micro Focus International Ltd. All rights reserved.
"C:\Program Files\IBM\WebSphere\AppServer1\java\bin\javac.exe" -g:none -verbose
com\mypackage\UpdStaffS\UpdStaffSHome.java
[com\mypackage\UpdStaffS\UpdStaffSHome.java を構文解析開始]
[190ms で構文解析完了]

..... 途中省略 .....

[C:\Program Files\IBM\WebSphere\AppServer1\java\jre\lib\core.jar (java/lang/Boolean.class) を読み込み中]
[com\mypackage\UpdStaffS\UpdStaffSBean.class を書き込み完了]
[合計 2603ms]
マニフェストが追加されました。
META-INF/ejb-jar.xml を追加中です。(入 = 1372) (出 = 512) (62% 収縮されました)
META-INF/weblogic-ejb-jar.xml を追加中です。(入 = 839) (出 = 340) (59% 収縮されました)
com\mypackage\UpdStaffS\UpdStaffS.class を追加中です。(入 = 247) (出 = 187) (24% 収縮されました)
com\mypackage\UpdStaffS\UpdStaffSBean.class を追加中です。(入 = 3401) (出 = 1509) (55% 収縮されました)
com\mypackage\UpdStaffS\UpdStaffSHome.class を追加中です。(入 = 257) (出 = 177) (31% 収縮されました)
minizip.exe: Micro Focus zip utility version 1.1.1
Based on MiniZip 0.15 by Gilles Vollant
Copyright (C) 2002-2003 Micro Focus International Limited

C:\UpdStaff>
```

これによって、UpdStaffS.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：

UpdStaffS.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
UpdStaffS.jar : ラッパーEJB のパッケージ

13) 同じ Net Express コマンドプロンプトから、以下のように ESDEMO へディプロイします。

```

C:¥UpdStaff>imtkmake -deploy carname=UpdStaffS.deploy¥UpdStaffS.car
Micro Focus Interface Mapping Toolkit v1.0.00
Copyright (C) 2004 Micro Focus International Ltd. All rights reserved.
Found 1 deployment services:

  1. Deployer (Deployment file-upload service) is Available at tcp:10.18.11.121:1070
    Listener (Basic HTTP web server) is Started
    Server ESDEMO (Communications server for Web Services) on -unknown- (-unknown-) is Started

Select a service provider (1-1) or 0 to exit: 1

Sending UpdStaffS.deploy¥UpdStaffS.car to ESDEMO's Deployer at tcp:10.18.11.121:1070...
Received 52 bytes:
http://10.18.11.121:1070/uploads/UpdStaffS.mufmSKhQ/

Server returned 202
Retrieving deployment log file http://10.18.11.121:1070/uploads/UpdStaffS.mufmSKhQ/deploylog.txt...
1000 (2005/10/06 20:48:14): mfdepinst 1.2.4 starting deployment of COBOL archive "UpdStaffS.car"
0010 (2005/10/06 20:48:14): Extracting and parsing manifest file
0011 (2005/10/06 20:48:15): Processing Manifest element in the manifest file
0012 (2005/10/06 20:48:15): Processing "Application" section of Manifest
0012 (2005/10/06 20:48:15): Processing "FileList" section of Manifest
2003 (2005/10/06 20:48:15): File pathname "C:¥UpdStaff¥UpdStaffS.deploy¥UpdStaffS.idt" is absolute; reducing to filename "UpdStaffS.idt"
0012 (2005/10/06 20:48:15): Processing "Service" section of Manifest
0019 (2005/10/06 20:48:15): Manifest processing completed
0020 (2005/10/06 20:48:15): Adding service and package objects to directory
0030 (2005/10/06 20:48:23): ES server "ESDEMO" notified service "UpdStaffS.UPDSTAFF" is available
0002 (2005/10/06 20:48:23): Installation of package "UpdStaffS.car" finished with 1 warnings

Deployment completed with warnings

C:¥UpdStaff>

```

14) 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。

Server ESDEMO [開始]					
サーバー...		リスナー (3)	サービス (5)	ハンドラ (2)	パッケージ (3)
1 - 3 of 3 packages					Show 10 packages
名前	視ステータス	パッケージモジュール	IDT	パッケージパス	
編集... ReadCustS.READCUST	Available	OK	C:\Program Files\Micro Focus\Net Express\Base\deploy\ReadCustS.Ep.Qq0_4\ReadCustS.idt	C:\Program Files\Micro Focus\Net Express\Base\deploy\ReadCustS.Ep.Qq0_4	
編集... SelStaffS.SELSTAFF	Available	OK	C:\Program Files\Micro Focus\Net Express\Base\deploy\SelStaffS.AB23h.YW6\SelStaffS.idt	C:\Program Files\Micro Focus\Net Express\Base\deploy\SelStaffS.AB23h.YW6	
編集... UpdStaffS.UPDSTAFF	Available	OK	C:\Program Files\Micro Focus\Net Express\Base\deploy\UpdStaffS.mufmSKhQ\UpdStaffS.idt	C:\Program Files\Micro Focus\Net Express\Base\deploy\UpdStaffS.mufmSKhQ	
追加					

15) 上記の例題プログラムを以下のように DB2 指令でコンパイルします。

```
C:¥UpdStaff>cobolc UpdStaff.cbl DB2(db=sample,bind);
Micro Focus Net Express V4
Version 4.0.38 Copyright (C) 1984-2005 Micro Focus International Ltd.
URN AXCGG/AAO/00000
* Micro Focus SQL External Compiler Module for IBM DB2 UDB
* Version 4.0.05 (C) copyright 1997-2005 Micro Focus International Ltd.
* チェック終了 : エラーはありません - コード生成を開始します
* Generating UpdStaff
* Data:          1376      Code:          1768      Literals:          628

C:¥UpdStaff>
```

16) 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。

17) コマンドプロンプトから、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。

```
C:¥UpdStaff>imtkmake -genclient service=UpdStaffS type=ejb appserver="WebSphere
5.x"
Micro Focus Interface Mapping Toolkit v1.0.00
Copyright (C) 2004 Micro Focus International Ltd. All rights reserved.
    1 個のファイルをコピーしました。
    1 個のファイルをコピーしました。
"C:¥Program Files¥IBM¥WebSphere¥AppServer1¥java¥bin¥javac.exe" -g:none -verbose
com¥mypackage¥UpdStaffS¥UpdStaffSHome.java
[com¥mypackage¥UpdStaffS¥UpdStaffSHome.java を構文解析開始]
[431ms で構文解析完了]

..... 途中省略 .....

[WEB-INF¥classes¥com¥mypackage¥UpdStaffS¥UpdStaffSSessionMonitor.class を書き込
み完了]
[合計 12518ms]

マニフェストが追加されました。
UpdStaffS.jar を追加中です。(入 = 5042) (出 = 3996) (20% 収縮されました)
UpdStaffS.war を追加中です。(入 = 10378) (出 = 9406) (9% 収縮されました)
mfejblib.jar を追加中です。(入 = 2984) (出 = 1958) (34% 収縮されました)
META-INF/application.xml を追加中です。(入 = 506) (出 = 284) (43% 収縮されました)

META-INF/ibm-application-bnd.xmi を追加中です。(入 = 392) (出 = 191) (51% 収縮さ
れました)
META-INF/ibm-application-ext.xmi を追加中です。(入 = 817) (出 = 266) (67% 収縮さ
れました)

C:¥UpdStaff>
```

これによって、UpdStaffS.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：

UpdStaffS.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 にデプロイします

- 18) 生成された UpdStaffS.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。管理コンソールでインストールが完了したことを確認します。
- 19) WebSphere 管理コンソールから、インストールしたエンタープライズアプリケーション SelStaffS を開始します。
- 20) 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=200 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。まず、DB2 コマンドプロンプトから、以下のように現在のレコードの内容を確認します。

```

C:\¥UpdStaff>db2 connect to sample

データベース接続情報

データベース・サーバー           = DB2/NT 8.2.0
SQL 許可 ID                       = ADMINIST...
ローカル・データベース別名       = SAMPLE

C:\¥UpdStaff>db2 "select name from staff where ID=200"

NAME
-----
Scoutten

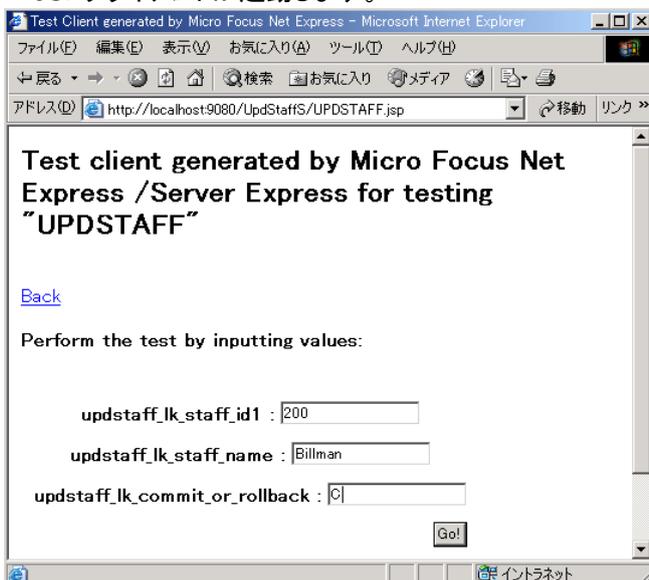
1 レコードが選択されました。

C:\¥UpdStaff>

```

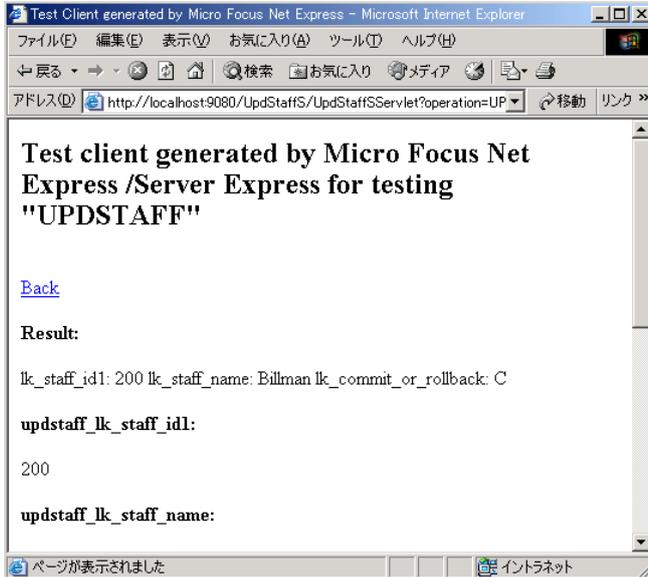
現在は Scoutten さんの名前が格納されています。

- 21) ブラウザから <http://localhost:9080/UpdStaffS/UPDSTAFF.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。



最初のテキストボックスに IDとして “200”、2 番目のテキストボックスに更新したい Scoutten 以外の名前を入力します。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。

22) 以下のように結果が返ります。



23) ここで DB2 コマンドプロンプトから、STAFF テーブルの ID=200 のレコードの内容を確認します。

```
C:¥UpdStaff>db2 "select name from staff where ID=200"

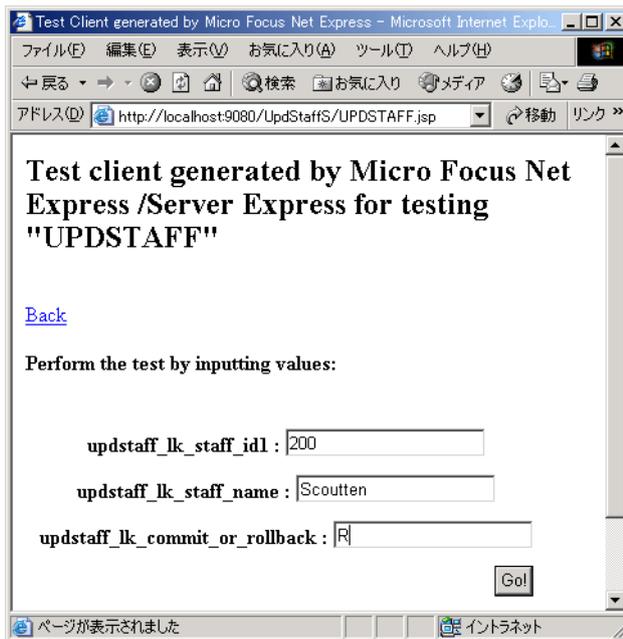
NAME
-----
Billman

  1 レコードが選択されました。

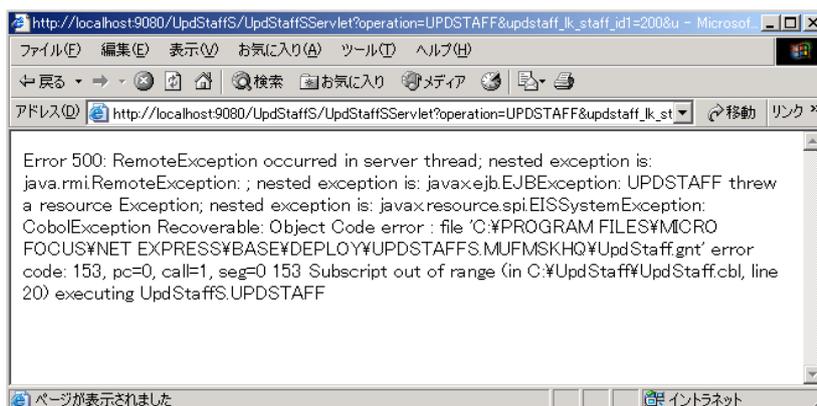
C:¥UpdStaff>
```

更新がコミットされ、先ほど入力した名前に更新されていることが確認できました。

24) 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 “Scoutten” 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” を入力します。



25) [Go!] をクリックします。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。



UpdStaff.cbl の 20 行目で添え字範囲の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わかります。

26) ここで、再び DB2 コマンドプロンプトで STAFF テーブルの ID=200 のレコードの内容を確認します。

```
C:\UpdStaff>db2 "select name from staff where ID=200"

NAME
-----
Billman

    1 レコードが選択されました。

C:\UpdStaff>
```

27) 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。