

Micro Focus Server Express 5.1

JBoss Enterprise Application
Platform 5.1

Red Hat Enterprise Linux Server
release 6.0

動作検証報告書

1 COBOL資産の JBOSS環境での活用動作検証報告書

2011年2月

マイクロフォーカス株式会社

協力：レッドハット株式会社

2 検証概要、目的及びテスト方法

1 検証概要

Micro Focus Server Express 5.1 J の Enterprise Server が提供する J2EE Connector 機能は、JCA 仕様準拠のコンテナとして多くの J2EE 準拠アプリケーションサーバーについて動作検証がなされています。本報告書は、Red Hat JBoss Enterprise Application Platform 5.1 との J2EE Connector の接続性を検証し、報告するものです。

※ 各製品のインストールや環境設定は、本報告書の範囲外です。

2 目的及びテスト方法

Micro Focus Server Express 5.1 J の Enterprise Server が提供する J2EE Connector は、現在 WebSphere 7, WebLogic 9, JBoss 5 などとの連携が動作保証されています。また、Enterprise Server は JCA 仕様準拠のコンテナとして、JCA 仕様に準拠したすべてのアプリケーションサーバーとの連携が可能です。

Red Hat JBoss Enterprise Application Platform 5.1 は、JCA 1.4 仕様に準拠し EJB 3.0 をサポートするコンテナを装備しており、理論的には Micro Focus Enterprise Server の EIS とも連携できます。また、Micro Focus Server Express の COBOL デプロイツールが自動生成する EJB3.0 準拠のラッパーコードも JBoss EAP 5.1 上で動作します。今回、以下のテストプログラムを実行することによって、このことを実際に検証しました。

- Oracle データベースアクセスへの XA 接続による照会プログラム

3 使用ハードウェア及びソフトウェア一覧

1 ハードウェア

Dell Latitude D520

Intel Core2Duo

2 OS

Linux 2.6.32-71.14.1.el6.x86_64

Red Hat Enterprise Linux Server release 6.0 (Santiago)

3 ソフトウェア

OpenJDK Runtime Environment (IcedTea6 1.7.5) (rhel-1.31.b17.el6_0-x86_64)

OpenJDK 64-Bit Server VM (build 14.0-b16, mixed mode)

JBoss Enterprise Application Platform 5.1.0

Micro Focus Server Express 5.1 WrapPack 5

Oracle Database Client Release 11.2.0.1.0 x86

4 検証結果

以下の「補足」で示す通り、以下の一連の作業が問題なく実行できることが検証できました：

- Micro Focus リソースアダプタの JBoss EAP へのディプロイ
- Red Hat EL 6.0 での Server Express 5.1 のインストールとコンパイル・実行
- Red Hat EL 6.0 での Enterprise Server の稼働と Oracle XA リソースの稼働
- Server Express で自動生成した EJB3.0 準拠ラッパーの JBoss EAP へのディプロイ
- Server Express で自動生成したテスト用 Web アプリケーションの JBoss EAP へのディプロイと運用

5 補足. COBOLデータ照会プログラムのディプロイとJCA連携

1

Micro Focus リソースアダプタ `mfcobol-notx.rar` と `mfcobol-notx-ds.xml` を **Server Express** のマニュアル記載の通り **JBoss Enterprise Application Platform** 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 Enterprise Application Platform** のコンソールに以下のメッセージが表示されます。

```
INFO [ConnectionFactoryBindingService] Bound ConnectionManager
'JBoss.jca:service=ConnectionFactoryBinding,name=eis/MFCobol_v1.5' to JNDI name
'java:eis/MFCobol_v1.5'
```

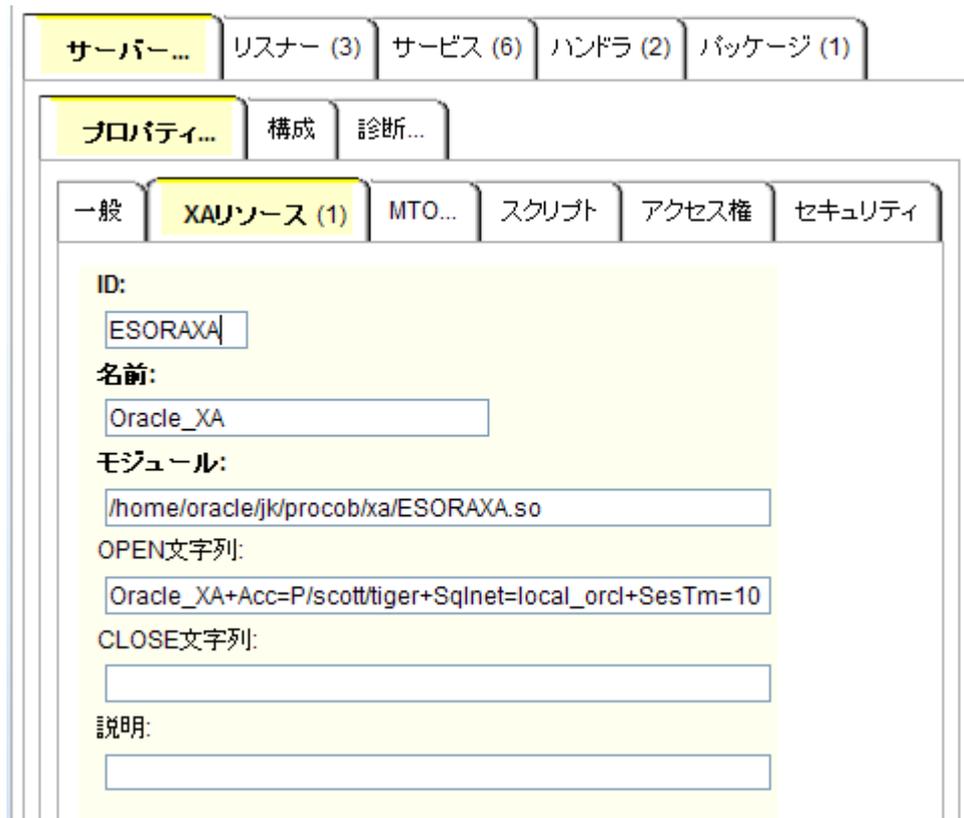
2

Oracle データベースへの接続をつかさどる **XA** スイッチモジュールを以下のようにビルドします。**Server Express** 製品にこれをビルドするための **COBOL** ソースとビルドスクリプトが提供されています。

```
$ export ORACLE_HOME=/opt/app/oracle/product/11.2.0/client32_1
$ cp $COBDIR/src/enterpriseserver/xa/* .
$ ./build ORA
$ ls -l ESORAXA.so
-rwxr-xr-x. 1 oracle oinstall 28236  2月  3 15:50 2011 ESORAXA.so
$
```

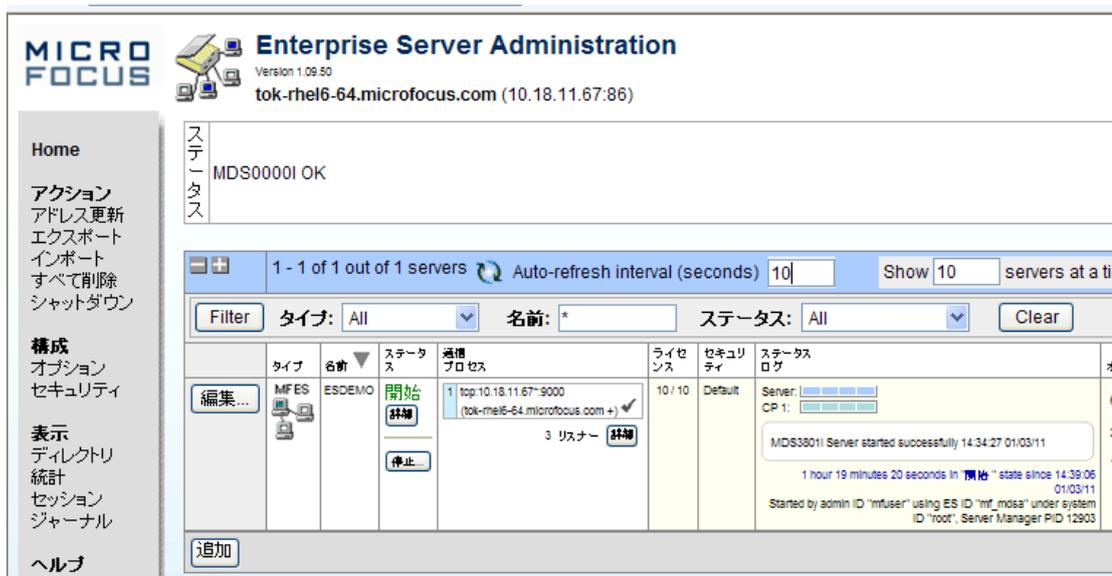
3

Micro Focus Enterprise Server Admin を開き、デフォルトのサーバーインスタンスである **ESDEMO** に Oracle **XA** リソースマネージャの登録を行います。**ESDEMO** の [サーバー] > [プロパティ] > [XA リソース] で [追加] ボタンをクリックし、以下のように新たな **XA** リソースの設定を行います。



4

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。



5

ESDEMO の [サーバー] > [診断] > [ES Console] に以下のように XA リソースの定義が正しく設定されたログが記録されていることを確認します。

```
110203 14390534 12907 ESDEMO CASX00020I ESORAXA XA interface loaded. Name(Oracle_XA),  
Registration Mode(Static) 14:39:05
```

```
110203 14390632 12907 ESDEMO CASX00015I ESORAXA XA interface initialized successfully  
14:39:06
```

6

Server Express を使用するための環境変数 COBDIR, PATH, LD_LIBRARY_PATH を正しく設定します。

7

以下の Oracle テーブル照会の 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 EMP は Oracle のサンプルデータベースにあるものを使用しています。

```
$ cat ReadEmp.cbl  
  
    WORKING-STORAGE SECTION.  
  
        EXEC SQL BEGIN DECLARE SECTION END-EXEC.  
01  EMP-NAME      PIC X(10).  
01  EMP-NUMBER    PIC S9(4) COMP VALUE ZERO.  
        EXEC SQL END DECLARE SECTION END-EXEC.  
        EXEC SQL INCLUDE SQLCA END-EXEC.  
  
    LINKAGE SECTION.  
01  LK-EMPNO      PIC 9(4).  
01  LK-ENAME      PIC X(10).  
  
    PROCEDURE DIVISION USING LK-EMPNO LK-ENAME.  
  
        MOVE LK-EMPNO TO EMP-NUMBER.  
        EXEC SQL SELECT ENAME  
            INTO :EMP-NAME FROM EMP WHERE EMPNO = :EMP-NUMBER  
        END-EXEC.  
        MOVE EMP-NAME TO LK-ENAME.  
        EXIT PROGRAM.  
  
$
```

8

Server Express で、上記プログラムをコンパイルします。

```
$ cob ReadEmp.cbl -C "P(cobsq1)"
```

9

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。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、ReadEmp.xml と ReadEmpS.xml が作成されます。

```
$ imtkmake -defmap src=ReadEmp.cbl service=ReadEmpS type=ejb
Micro Focus Interface Mapping Toolkit v5.1.00
Copyright (C) 2004-2010 Micro Focus (IP) Ltd. All rights reserved.
```

10

プログラム ReadEmp.cbl は二つのパラメータを受けとしますが、そのうち第一パラメータの LK-EMPNO は入力パラメータであり、第二パラメータの LK-ENAME は出力パラメータです。生成されたデフォルトマッピング ReadEmpS.xml ではどちらもデフォルトで入出力パラメータの扱いとなっていますので、テキストエディタで ReadEmpS.xml を以下のように修正します。

```
<parameter byType="in" dataType="BigInteger" name="lk_empno_in" occurs="0">
  <mappings>
    <mapping item="lk-empno"/>
  </mappings>
</parameter>
<parameter byType="out" dataType="String" name="lk_ename_out" occurs="0">
  <mappings>
    <mapping item="lk-ename"/>
  </mappings>
</parameter>
```

11

ディプロイメントパッケージの生成・コンパイルに必要な以下の Java クラスを CLASSPATH に追加します。

```
$COBDIR/lib/javaee5/jboss5/mfejblib.jar
$COBDIR/lib/javaee5/jboss5/mfconnector.jar
$COBDIR/lib/mfcobolpure.jar
/opt/jboss-eap-5.1/jboss-as/lib/jboss-javaee.jar
/opt/jboss-eap-5.1/jboss-as/common/lib/servlet-api.jar
```

12

以下のように COBOL 側と EJB 側のデプロイメントパッケージを生成します。

```
$ imtkmake -generate service=ReadEmpS type=ejb j2eeVersion=5 appServer="JBoss 5.0"
ejbversion=3

Micro Focus Interface Mapping Toolkit v5.1.00
Copyright (C) 2004-2010 Micro Focus (IP) Ltd. All rights reserved.

[com/mypackage/ReadEmpS/ReadEmpS.java を構文解析開始]

[331ms で構文解析完了]

[ソースファイルの検索パス:
/opt/mf/SE51WS5/lib/mfj2se.jar, /opt/mf/SE51WS5/lib/javaee5/jboss5/mfcci.jar, /opt/mf/SE51WS5/lib/javaee5/jboss5/mftransport.jar, /opt/mf/SE51WS5/lib/javaee5/jboss5/mfcobolpure.jar, /opt/mf/SE51WS5/lib/javaee5/jboss5/log4jpure.jar, /opt/mf/SE51WS5/lib/javaee5/jboss5/mfejblib.jar, /opt/mf/SE51WS5/lib/mfcobolpure.jar, /opt/mf/SE51WS5/lib/mfcobol.jar, /usr/java/jdk1.6.0_21_32/jre/lib/rt.jar, /opt/mf/SE51WS5/lib/mfimt.jar, /opt/mf/SE51WS5/lib/xerces.jar, /opt/mf/SE51WS5/lib/castor-0.9.4.1-xml.jar, /opt/mf/SE51WS5/lib/javaee5/jboss5/mfconnector.jar, /opt/jboss-eap-5.1/jboss-as/lib/jboss-javaee.jar, /opt/jboss-eap-5.1/jboss-as/common/lib/servlet-api.jar]
0.5.GA/server/default/lib/JBoss-j2ee.jar (javax/ejb/EJBHome.class) を読み込み中]

----- 途中省略 -----

[com/mypackage/ReadEmpS/ReadEmpSBean.class を書き込み完了]

[合計 1554ms]

注: com/mypackage/ReadEmpS/ReadEmpSBean.java の操作は、未チェックまたは安全ではありません。

注: 詳細については、-Xlint:unchecked オプションを指定して再コンパイルしてください。

マニフェストが追加されました。

com/mypackage/ReadEmpS/ReadEmpS.class を追加中です。(入 = 242) (出 = 202) (16% 収縮されました)

com/mypackage/ReadEmpS/ReadEmpSBean.class を追加中です。(入 = 4442) (出 = 2038) (54% 収縮されました)

com/mypackage/ReadEmpS/ReadEmpS.class を追加中です。(入 = 242) (出 = 202) (16% 収縮されました)

com/mypackage/ReadEmpS/ReadEmpSBean.class を追加中です。(入 = 4442) (出 = 2038) (54% 収縮されました)
```

```
minizip: Micro Focus zip utility version 1.1.3
Based on MiniZip 0.15 by Gilles Vollant
Copyright (C) 2002-2007 Micro Focus (IP) Limited.

$
```

これによって、ReadEmpS.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：

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

ReadEmp.jar : EJB ラッパー。JBoss 下の J2EE アプリケーションから利用して COBOL サービスを呼び出すことができます。

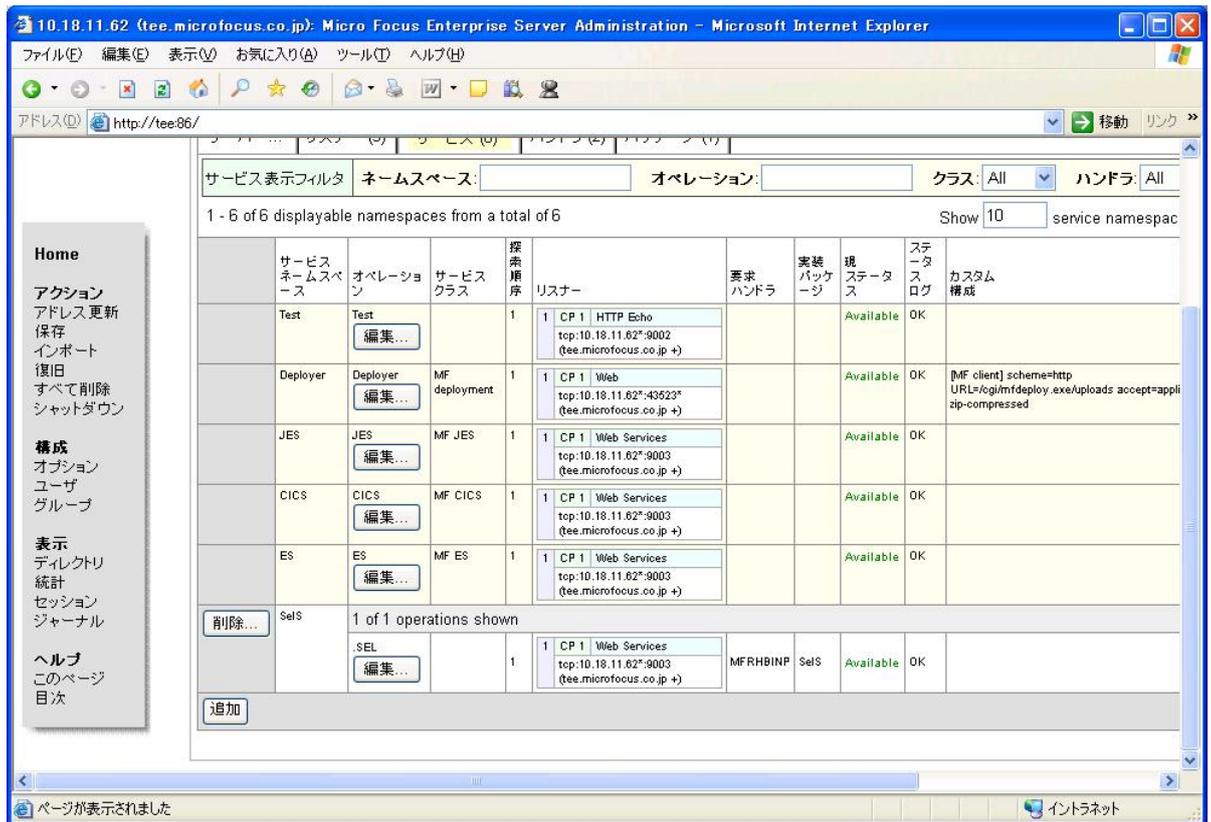
13

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

```
$ mkdir $COBDIR/deploy/ReadCustS
$ cp ReadEmp.int $COBDIR/deploy/ReadCustS
$ cp ReadEmpS.deploy/ReadEmpS.car $COBDIR/deploy/ReadCustS
$ cd $COBDIR/deploy/ReadCustS
$ mfdepinst ReadEmpS.car
```

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。



15

同じく Server Express 環境下で、以下のように、JSP テストクライアントの ear を自動生成します。

```

$ imtkmake -genclient service=ReadEmpS type=ejb j2eeVersion=5 appServer="JBoss 5.0"
ejbversion=3
Micro Focus Interface Mapping Toolkit v5.1.00
Copyright (C) 2004-2010 Micro Focus (IP) Ltd. All rights reserved.
[com/mypackage/ReadEmpS/ReadEmpSHome.java を構文解析開始]
[101ms で構文解析完了]
----- 途中省略 -----
[WEB-INF/classes/com/mypackage/ReadEmpS/ReadEmpSSessionMonitor.class を書き込み完了]
[合計 359ms]
WEB-INF/web.xml を追加中です。(入 = 735) (出 = 308) (58% 収縮されました)
WEB-INF/jboss-classloading.xml を追加中です。(入 = 180) (出 = 141) (21% 収縮されました)
WEB-INF/lib/mfejlib.jar を追加中です。(入 = 3027) (出 = 1983) (34% 収縮されました)
ReadEmpSMain.jsp を追加中です。(入 = 600) (出 = 374) (37% 収縮されました)
READEMP.jsp を追加中です。(入 = 1498) (出 = 652) (56% 収縮されました)

```

```
READEMP0. jsp を追加中です。(入 = 315) (出 = 198) (37% 収縮されました)
WEB-INF/classes/com/mypackage/ReadEmpS/ReadEmpSServlet.class を追加中です。(入 = 9538)
(出 = 3406) (64% 収縮されました)
WEB-INF/classes/com/mypackage/ReadEmpS/ReadEmpSjspBean.class を追加中です。(入 = 1965)
(出 = 880) (55% 収縮されました)
WEB-INF/classes/com/mypackage/ReadEmpS/ReadEmpSSessionMonitor.class を追加中です。(入 =
1307) (出 = 611) (53% 収縮されました)
マニフェストが追加されました。
ReadEmpS.jar を追加中です。(入 = 2951) (出 = 2634) (10% 収縮されました)
ReadEmpS.war を追加中です。(入 = 10001) (出 = 9121) (8% 収縮されました)
mfejblib.jar を追加中です。(入 = 3027) (出 = 1983) (34% 収縮されました)
META-INF/application.xml を追加中です。(入 = 516) (出 = 257) (50% 収縮されました)
クライアント生成が正常終了しました。
$
```

16

生成された ReadEmpS.ear を JBoss Enterprise Application Platform 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 Enterprise Application Platform 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:/opt/jboss-eap-5.1/jboss-as
/server/default/deploy/SelS.ear
INFO [EjbModule] Deploying SelSEJB
INFO [ProxyFactory] Bound EJB Home 'SelSEJB' to jndi 'ejb/SelSEJB'
INFO [EJBDeployer] Deployed: file:/opt/jboss-eap-5.1/jboss-as
/server/default/tmp/deploy/tmp15091SelS.ear-contents/SelS.jar
INFO [TomcatDeployer] deploy, ctxPath=/SelS,
warUrl=.../tmp/deploy/tmp15091SelS.ear-contents/SelS-exp.war/
INFO [EARDeployer] Started J2EE application: file:/opt/jboss-eap-5.1/jboss-as
/server/default/deploy/SelS.ear
```

17

ブラウザから <http://サーバー:8080/ReadEmpS/READEMP.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。

Test client for ReadEmpS.READEMP

[Back](#)

Perform the test by entering values:

reademp_lk_empno_in :

[Back](#)

18

テキストボックスに、データファイル STAFF 中に存在するキー値を入力し、[Go!] をクリックします。

19

COBOL サービスが呼び出され、照会されたデータが JSP 画面上に表示されます。

Test client for ReadEmpS.READEMP

[Back](#)

Perform the test by entering values:

reademp_lk_empno_in :

Result:

| Variable | Value |
|----------|-------|
| Result | JAMES |

[Back](#)