

Micro Focus Server Express 5.1 J for SPARC Solaris WebOTX Application Server V8.4 動作検証結果報告書

> 2011 年 10 月 24 日 マイクロフォーカス株式会社

Copyright © 2011 Micro Focus. All Rights Reserved. 記載の会社名、製品名は、各社の商標または登録商標です。

1 検証概要、目的及びテスト方法

1.1 検証概要

Micro Focus Server Express 5.1 J の Enterprise Server が提供する Java EE Connector 機能は、JCA仕様準拠のコンテナとして多くの Java EE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、NECのWebOTX Application Server(以降 WebOTX ASと略記)との JavaEE Connector の接続性を検証し、報告するものです。

1.2 目的及びテスト方法

Micro Focus Server Express 5.1 J の Enterprise Server が提供する JavaEE Connector は、現在 WebSphere, WebLogic, JBoss などとの連携が動作保証されています。しかし Enterprise Serverは、JCA仕様準拠のコンテナとして、設計上は 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。

WebOTX ASはJCAの仕様に準拠しており、理論的には Micro Focus Enterprise Serverの EISとも連携するはずです。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。

- (1) 渡された2つの数字パラメタを加算してその結果を返すCOBOLサブルーチンを使用
- (2) Interface Mapping Toolkit が自動生成した EJB と ServletクライアントをWebOTX AS上 で運用し、COBOLを呼び出す

2 使用ハードウェア及びソフトウェア一覧

Sun Fire V240 US3i-1.5GHz * 2 Memory 8Gb Solaris 10 Oracle Java DK 1.6.0_21 Micro Focus Server Express 5.1 WrapPack 6 WebOTX AS Express V8.41 (8.41.00.00 (build 20110714))

作業用環境として Windows XP PC を使用し、Internet Explorer 8及び CygWin X Serverを利用

3 テスト内容

以下に実施したテストの概要を述べます。詳細な手順については補足に記載します。

- (1) 使用した COBOLロジック 渡された2つの数字パラメタを加算してその結果を返す簡単なCOBOLサブルーチンを使用
- (2) 使用したリソースアダプタ
 \$COBOL/ lib/javaee5/oracleweblogic10/mfcobol-notx.rar
 WebLogic 10.x にディプロイするのに適した形式でパッケージされたものであり、JavaEE仕様に 照らして最も標準的な提供形態です。
- (3) 使用した Enterprise Server既定義の ESDEMO をそのまま使用。
- (4) 使用したWebOTX ASのドメイン 既定義の domain1 をそのまま使用。
- (5) 使用した JavaEEアプリケーション
 Server Express の Interface Mapping Toolkit がデプロイ時に自動生成する EJB と、自動生成される Webモジュールクライアントを使用。

4 結果

上記のテストを実行した結果、正常に実行されることを確認しました。詳細な結果については補足に記載します。

5 テスト結果及び考察

最新の JavaEE標準をサポートする WebOTX ASで、既存の Micro Focus Server Express 5.1 Jの

JavaEE Connector 接続を問題なく使用できることが検証できました。

補足. 検証の手順

1. 前提条件

本検証では、各ソフトウェアはデフォルトでインストールされたままの状態になっていることを仮定しています。 Server Express はデフォルトのインストール先に Enterprise Server も含めてインストールされており、 出荷時 設定のサーバー ESDEMO がそのままの状態で利用可能になっているものとします。 検証を始める前に ESDEMO を開始状態にしておきます。

WebOTX AS もデフォルトでインストールされており、管理者ユーザ admin/adminadmin で、出荷時設定のドメ イン domain1 が利用可能になっているものとします。

ここでは、以下の簡単な COBOL 例題プログラム CALCU.cbl を使用します。第一、第二の引数を加算し、結果 を RESULT に返すというだけのプログラムです:

```
LINKAGE SECTION.

01 CALCULATOR.

05 ARG1 pic 9(5) comp-3.

05 ARG2 pic 9(5) comp-3.

05 RESULT pic 9(5) comp-3.

procedure division using CALCULATOR.

move ARG1 to RESULT

add ARG2 to RESULT

exit program.
```

2. リソースアダプタの設定

Server Express5.1JのEnterprise ServerへのJCA接続は、WebLogic などのいくつかのJava EEアプリケーションサーバーで動作保証されており、それらのそれぞれに対応したリソースアダプタが個別に製品に添付されています。これらは基本的に同じ物ですが、アプリケーションサーバーの種類によって必要となるマニフェストやデプロイメントディスクリプタが個別にパッケージ化されています。

今回の検証対象である WebOTX に対応したものは用意されていませんので、ここでは比較的標準的な内 容を持っている WebLogic 向けのリソースアダプタを使用しました。

 Windows PC上の Internet Explorer で <u>http://<Solarisサーバーアドレス>:5858/</u>を指定し、WebOTX 統合運用管理コンソールを開きます。

🌀 WebOTX統合運用管理コンソール	👌 • 🗟 - 🖃 🖶 • ページ®・ セーフティ⑤・ ツール@・ 🎱
WebCTX	ようこそ, admin
ビューの表示(<u>W</u>) 🔹	於定(S) 10 ドメイン管理(Z) △ モニター(M) ▼ ◇ リフレッシュ(R) 13 ログアウト(L) ② ヘルブ(H)
アプリケーション	
■ □ リソース ■ 「 アプリケーションサーバ」	WebOTX はクラウドやSaaS仮想化を見捉えつつ 最新技術と豊富な道入実績をも
🗉 🖳 統計情報	とに「止まらない」「変化に強い」「先進技術を活用した」システムの構築をご支援し新
■ サーハー覧	
	ようこそ
	domain1@tok-wedge
検索: マビュー	Web版統合運用管理コンソール GUIの紹介

 左ペインのツリービューで [アプリケーション] をクリックします。以下のように、[アップロード] のチェック をオフにし、[コンポーネントタイプ] のプルダウンで [リソースアダプタ] を選択し、[アーカイブ名] に \$COBOL/ lib/javaee5/oracleweblogic10/mfcobol-notx.rar を指定します。(\$COBDIR の部分は Server Express のインストール先パス名に置き換えます)

Web	ようこそ, admin
	ビューの表示(W) 🔹 🏋 設定(S) 🛛 『ジャイン管理(Z) 🏻 🎱 モニター(M) 🔹 🗇 リフレッシュ(B) 🖥 ログアウト(L) 🖗 🥝 ヘルプ(Ł
domain1>>アプリケーション	
 	配備 置換 操作
🗷 🧵 アブリケーションサーバ	アップロード
■ 頃 統計情報	
□ 📮 ビーン □ 🔄 サーバー覧	サーバマシンに配備可能なファイルをアップロードする
	コンポーネントタイプ
	🖉 (リソースアダプタ)
	コンポーネントのタイプ
	ファイル
	参照
	配備可能なファイル名(アップロードオプションにチェックを入れている場合は、ローカルマシン上のファ イルバス)
	アーカイブ名
検索: ☑!	Za – 🖉 🖉 //opt/microfocus/cobol/lib/javaee5/oracleweblogic10/mfcobol-notx.rar
屛麻 恋面 ローカル絵表純	アップロードしたアーカイブのURI

3. [配備] をクリックします。以下のようにリソースアダプタの配備が SUCCESS になることを確認します。

	(配備(<u>P</u>)) 再表示(<u>B</u>)	
☑ ログ詳細		9
(情報): [2011/10/18 17:52:47]管理対象生成:mfcobol-notx (情報): [2011/10/18 17:52:42]処理成功 - アプリケーション mfcob	ool-notx を配備しました。: SUCCESS.	

 統合運用管理コンソールのツリービューの [アプリケーション] の下に以下のように mfcobol-notx が作 成されていることを確認します。これをクリックすると右ペインに以下のように配備されたリソースアダプ タの情報が表示されます。

Web TX		L	うこそ, adm
ビューの表示(W) * 🏠 設定(S) 🦷] ドメイン管理(Z) 🛛 🕸 モニター(M) 🔹 🗇 リフレッシュ(R) 🔹 ログアウト	(L) @ヘル
<u>domain1</u> >>アプリケーション >> <u>リソ</u>	<u>ースアダプタ</u> >>mfcobol	-notx	
□ ◆ アブリケーション	設定操作		
mfcobol-notx	コネクタモジュール		
🗉 🛄 リソース	名	mfcobol-notx	
🗷 🧵 アブリケーションサーバ	*	コネクタモジュール名を指定します。	?
🗉 🔍 統計情報			
■ (△) モニター			
💐 サーハ一覧	「ネクタモンユール」 格納場所	\${com.nec.webotx.instanceRoot}/applications/j2ee-module	
	*	コネクタモジュールの格納場所を指定します。	?
	リソースタイプ		9
		リノースダイノ右で指定します。	•
	リソースの有効		

5. 配備されたリソースアダプタのコネクションプールを登録します。統合運用管理コンソールのツリービュー で [リソース]を展開し、[コネクションプール]を選択します。右ペインに現れる [操作リスト] の中から [コネクションプールの登録] をクリックします。

■ ◆ アプリケーション ■ ■ リソースアダプタ	▲ コネクションプールの登録	
 mfcobol-notx リソース JMSリソース コネクタリソース コネクタコネクションブール JDBCデータソース 泳続化リソース JavaMailリソース JNDIリソース JNDIリソース リソースアダブタ コネクタモジュール構成情報 	操作リスト コネクションブールの登録 コネクションブールの削除 コネクションブールのテスト コネクションブールのテスト 名前: コネクションブールの登録 説明: コネクションブールを登録します。	

6. 以下の通りに登録するコネクションプールの情報を入力します。

一般	
リソースアダブタ 名 ☆	<mark>mfcobol-notx</mark> リソースアダプタ名を指定します。
コネクション定義 名 ☆	javax.resource.cci.ConnectionFactory リソースアダプタの配備記述子中のconnection-definiti を指定します。
コネクタコネクシ ョンブール名 ☆	CCIMFCobol_v1.5 コネクタコネクションブール名を指定します。

7. [実行] ボタンをクリックし、以下のように成功のログが表示されることを確認します。

(実行(E)) 再表示(B)	
▼ ログ詳細	0
 (情報): [2011/10/18 18:02:28]管理対象生成:CCIMFCobol_v1.5 (情報): [2011/10/18 18:02:20](コネクタコネクションブール」の「コネクションブールの登録」操作実行に成功しました。 返却値: domain1:type=connector-connection-pool,name=CCIMFCobol_v1.5,category=config (情報): [2011/10/18 17:52:47]管理対象生成:mfcobol-notx (情報): [2011/10/18 17:52:42]処理成功 - アプリケーション mfcobol-notx を配備しました。: SUCCESS. 	

 統合運用管理コンソールのツリービューの [コネクションプール] の下に以下のように CCIMFCobol_v1.5 が作成されていることを確認します。これをクリックすると右ペインに以下のように 登録されたコネクションプールの情報が表示されます。



9. 最後にコネクタリソースを登録します。管理コンソールのツリービューで [リソース] > [コネクタリソース] を選択します。右ペインに現れる [操作リスト] の中から [コネクタリソースの登録] をクリックします。

domain1 >>リソース >>コネクタリソース	· · · · · · · · · · · · · · · · · · ·	
 □ ◆ アブリケーション □ リソースアダブタ ■ リソースアダブタ ■ mfcobol-notx □ リソース □ JMSリソース □ コネクタリソース □ コネクタコネクションブール □ JDBCデータソース ■ ホ続化リソース □ JavaMailリソース ■ カスタムリソース ■ JNDIリソース ■ JNDIリソース 	コネクタリソースの登録 (コネクタリソースの登録) コネクタリソースの削除 名前: コネクタリソースの登録 説明: マネクタリソースを登録します。	

10. 以下の通りに登録するコネクションプールの情報を入力します。

一般		
コネクタコネクシ ョンブール名 ☆	CCIMFCobol_v1.5 コネクタコネクションプール名を指定します。	<u>I</u>
JNDI名	eis/MFCobol_v1.5 コネクタリソースのJND名を指定します。	Ľ
コネクタリソース の説明	コネクタリソースの説明を記述します。	

11. [実行] ボタンをクリックし、以下のように成功のログが表示されることを確認します。

	実行(<u>E</u>)	再表示(<u>B</u>)
☑ ログ詳細		ę
(情報): [2011/10/18 18:08:40]管理対象生成:eis/MFCobol_v1.5 (情報): [2011/10/18 18:08:40]管理対象生成:eis/MFCobol v1.5 (情報): [2011/10/18 18:08:32](コネクタリソース」の「コネクタリソースの登録」操作 返却値: domain1:type=connector-resource,jndi-name=eis/MFCobol_v1.5,cate	実行に成功しました。 egory=config	

 12. 統合運用管理コンソールのツリービューの [コネクションプール] の下に以下のように CCIMFCobol_v1.5 が作成されていることを確認します。これをクリックすると右ペインに以下のように登録されたコネクションプールの情報が表示されます。

<u>domain1>>リソース>>コネクタリソース</u> >>	eis/MFCobol_v1.5	
🗉 🔷 アプリケーション 🔼	設定 操作	
💷 🖿 リソースアダプタ		
🖿 mfcobol-notx	eis/MECohol v1.5	
■ ■ リソース	JNDI名 コネクタリソースのJNDI登録名です。	
🖿 JMSリソース		
🗉 🖿 コネクタリソース		
ជឿ eis/MFCobol_v1.5	コネクタコネクショ ンプール名 CCIMECohol v1.5	
💷 🖿 コネクタコネクションブール 📲		9
CCIMFCobol_v1.5		•
🖿 JDBCデータソース		
💼 永続化リソース	sr user	
🖿 JavaMailリソース	オブジェクトタイプを指定します。	
🖿 カスタムリソース		
🖿 JNDIリソース		
🖿 リソースアダプタ	ックースの有効/三のリソースの有効/無効を指定します。	
🖿 コネクタモジュール構成情報	(既定值:有効(true))	
🔲 🛲 🛲 An 🛏 (A. A. M. 1990)		

以上で、リソースアダプタの配備は完了しました。

3. COBOLプログラムの準備

ここでは、検証で使用する COBOL プログラムを Enterprise Server に配備し、 同時に EJB ラッパーを自動生成します。

- 1. 作業用ディレクトリを新規作成し、COBOL 例題プログラム Calcu.cbl をコピーします。
- 2. Server Express を使用する環境変数を設定します。 検証では ShiftJIS ロケールを使用しましたので LANG 環境変数は ja_JP.PCK に設定されています。
- 3. Calcu.cbl をコンパイルします。
- 4. \$ cob -ug Calcu.cbl
- 5. X Window をホストする環境を整え、cobimtk コマンド打鍵し、インターフェイスマッピングツールキットを 起動します。
- 6. [ファイル] > [サービスインターフェイスの新規作成]を選択します。



7. サービスインターフェイスグループの新規作成ダイアログが現れます。ここで [名前] には何でも良いで すがグループ名を命名し、[場所] に現在の作業用ディレクトリのパスを入力します。

SB New Service Interfaces Group	
複数のサービスインターフェイスをグループ化すると、関連する 展開できます。サービスインターフェイスグループの名称と置き 必要があります。	? サービスを簡単に ・場所を指定する
名前: work	
場所: /export/home/tarot/WebOTX_Demo/work	参照
了解	取消し

8. [Java インターフェイス] を選択して右クリック [Java インターフェイスの新規作成] を選択します。

SF
ファイル サービス ヘルプ
▲ Webサービス Javaインターフェイマ Javaインターフェイスの新規作成

9. マッピングするプログラムを選択する画面が表示されます。以下のように [COBOL プログラムの名前] の [参照...] から COBOL のプログラムソース Calcu.cbl を選択し、[次へ] をクリックします。

SE .		×
	1 サービスマッピングを作成するために、マップされるCOBOL プログラムおよびコンパイルに必要な指令と COPYメンバー の所在が必要です。 COBOLプログラムの名前:	
	/export/home/tarot/WebOTX_Demo/Calcu.cbl 参照 COPYメンバーの所在パス(COBCPY): 参照	
-	コンパイル時の指令: noanim noint preservecase	
	< 戻る 次へ > 取消し ヘルブ	

10.サービス名を入力する画面が表示されますので、ここでは CalcuServ と入力します。

SE		×
	l	?
	サービスマッピング名はサービスマッピング	
	のリストの中で識別する名称です。	
	サービスの名称を入力してください:	_
	Calcuserv	
		16
	_ <戻る 取消し _ へルブ	

11.以下のダイアログでは [省略時マッピング] がチェックされたままの状態で [次へ] をクリックします。



12.以下のダイアログで [完了] をクリックします。

88		
	完了ボタンを押すと、インターフェイスマッパーウィンドウが表示されます。	2
	≪戻る 完了 取消し	ヘルプ

13. マッパーウィンドウが現れます。 左ペインには CALCU.cbl の LINKAGE パラメタに書かれた宣言がその まま表示されています。 右ペインにはこれを EJB メソッドとしてマッピングする方法を示しています。 省 略時マッピングでは以下のように LINKAGE SECTION の集団項目がグループフィールドとしてマップ されています。

8 1				
ファイル オペレーション フィールド 割	リ当て			
	×.			
COBOL入り口点 [CALCU] / プログラム [C	オペレーション CALCU 7 Transaction	Option.NotSupported		
名前 Picture	インターフェイスフィールド			
	名前	方向	Туре	Occurs
ARG2 9(5) comp-3 RESULT 9(5) comp-3	 一日 CALCULATOR_io ARG1_io ARG2_io RESULT_io 	入出力	int int int	Occurs
	COBOL割り当て COBOLフィールド		値	

14. 一旦グループフィールドを削除します。右ペインで CALULATOR_io を右クリックし、[削除] を選択しま

インターフェイス	フィールド	
名前		方向
	🛋 インターフェイスフィールドの新規作成	
—□ ARG2_i	💣 インターフェイスフィールドグループの新	所規作成…
	<mark>湯</mark> 切り取り	Ct
	貼り付け	Ct
	肖JB余	
冉利用フィールド 々前	グループ化	
	//J.1	

15. 右ペインがクリアされます。ここで左ペインの ARG1 を右ペインにドラッグします。

COBOL入り口点[CALCU]/;	プログラム [C		オペレーション CALCU 7 Transaction
名前	Picture		インターフェイスフィールド
		ΙT	2前
ARG1	9(5) comp-3		
ARG2	9(5) comp-3		
	9(5) comp-3		-

16.引き続き ARG2 と RESULT も右ペインにドラッグします。デフォルトではすべてのフィールドが入力パ ラメタとしてマップされています。この例題では最後の RESULT は出力パラメタなので、これを以下のよ

うに [出力] に変更します。

COBOL入り口点 [CALCU] / プログラム [C	オペレーション CALCU / TransactionOption.Not	Supported
名前 Picture	インターフェイスフィールド	
	名前	方向
ARG1 9(5) comp-3	ARG1	入力
RESULT 9(5) comp-3	ARG2	入力
		人力
	SE SE	
	ー 再利用フィー	2
	名前: RESULT	
	型: <u>int </u>	
	方向: ○入力 ④出力 ○入出力	
	COBOL割년当	
		編集
	了解	取消し

17.マッピングが以下のようになっていることを確認してください。

COBOL入り口点 [CALCU] / プログラム [C			オペレーション CALCU 🗹 TransactionOption.NotSupporte	d		
名前	Picture		インターフェイスフィールド			
		Т	名前	方向	Туре	Occurs
	9(5) comp-3		r ⊐ ARG1	入力	int	
	9(5) comp-3		—⊐ ARG2	入力	int	
	0(0) 00mp 0			出力	int	

18.作成されたマッピングを以下の [保存] ボタンで保存しマッパーウィンドウを閉じます。



19.ツリービュー上で [CalcuServ] を右クリックし、[設定...] を選択します。

■ Webサービス ■ Javaインターフェイス ■ CalcuServ						
L-€ ca	ディプロイ					
	編集					
	削除					
	設定					
	クライアント生成					

20.以下のダイアログが現れます。[一般] タブでは [EJB 生成] をチェックします。

SF
一般 ディプロイ先サーバー アプリケーションファイル EJB生成
● E-IB生成
◯ Java Bean生成 (J2SE使用)
◯ Java Bean生成 (J2SE使用)

21. [ディプロイ先サーバー] タブで、[Enterprise Server 名] の [変更...] ボタンをクリックし、開始している ESDEMO を選択します。

S I				_	
					2
一般 ディプロ	イ先サーバー	アブリケーションファー	イル EJB生成		
Enterprise Se	rver名:				
			変更	[]	
			14 <u>1223102</u>		
Enterprise	Senver主行時段	関節の使用			
Ente					
サービス名 〇	urrent Develop	ment Working Mode: 32	? bit		
Calcu	サーバー名	サーバー状態	Deployer名	Deployerアドレス	Deployed
	SDEMO	Started	Deployer	10.18.11.28:41006	Available
「トランザ					
() アプロ					

22. [サービス名] に CalcuServ が入っていることを確認してください。入っていない場合は修正してください。

一般 ディプロイ先サーバー アプリケーションファイル EJB生&	*
Enterprise Server名:	
ESDEMO (10.18.11.28)	変更
「Enternrice Conver実行時環境の使用	
Enterprise Server + 10477# - 40712/H	
Enterprise Server與行時環境の構成…	
サービス名:	
CalcuServ	
「トランザクション管理	
● アプリケーション管理	
○ コンテナ管理	
Username/password required for deployment	

23.[アプリケーションファイル] タブでは、コンパイル済みの Calcu.gnt、Calcu.idy、Calcu.cbl を追加します。

を追加
を削除

24. [EJB 生成] タブでは以下のように指定します。

[Application Server] は [Java EE 5] と [Weblogic 10.3.4] を選択します。[Java Compiler] にお使いの Java DKの bin ディレクトリのパス名を入力します。[J2EE Class Path] には WebOTX の

ー般 ディブロイ先ち	ナーバー アプリケーションファイル EJB生成
Application server:	Java EE 5 🔽 WebLogic 10.3.4 🛛
EJB 属性	
EJB Version 3 (Ja	va Compiler version 1.5 or greater is required) 🗹
Bean 名:	CalcuServ
バッケージ名:	com.mypackage.CalcuServ
セッション永続性:	● ステートレス ○ ステートフル
EJB Location:	Local ORemote
ディプロイメント5	ディスクリプタ属性
EJB 名:	CalcuServEJB
Archive name:	CalcuServ
J2SE and J2EE Att	tributes
Select the path co	ntaining Java compiler
Java compiler:	/usr/jdk/instances/jdk1.6.0/bin 参照
Classpath for the (Servlet implement)	selected application server's EJB and Java connector implementation Itation also, if generating the client)
J2EE Class Path:	/optWebOTX/lib/javaee.jar 参照

/opt/WebOTX/javaee.jar を入力します。それ以外はディフォルト値のままにします。

25.[了解] をクリックしダイアログを閉じます。

26. [Java インターフェイス] > [CalcuServ] を選択して右クリックして [ディプロイ] を選択します。

← ≦ Webサービス ー ● ■ Javaインター ー ー ● CalcuServ	フェイス
	ディプロイ
	編集
	肖耶余
	設定
	クライアント生成

27.以下のダイアログでサービスのディプロイが完了することを確認してください。。

"ィブ	ロイメントログ:			
011	(2011年10月19日	(水)	12時54分57秒):	Processing Manifest element in the mani
012	(2011年10月19日	(水)	12時54分57秒):	Processing "Application" section of Mar
012	(2011年10月19日	(水)	12時54分57秒):	Processing "FileList" section of Manife
003	(2011年10月19日	(水)	12時54分57秒):	File pathname "/export/home/tarot/Web07
003	(2011年10月19日	(水)	12時54分57秒):	File pathname "/export/home/tarot/Web07
003	(2011年10月19日	(水)	12時54分57秒):	File pathname "/export/home/tarot/Web07
003	(2011年10月19日	(水)	12時54分57秒):	File pathname "/export/home/tarot/Web07
012	(2011年10月19日	(水)	12時54分57秒):	Processing "Service" section of Manifes
019	(2011年10月19日	(水)	12時54分57秒):	Manifest processing completed
020	(2011年10月19日	(水)	12時54分57秒):	Adding service and package objects to c
021	(2011年10月19日	(水)	12時54分58秒):	Using directory at mrpi://10.18.11.28:8
030	(2011年10月19日	(水)	12時54分59秒):	ES server "ESDEMO" notified service "Ca
002	(2011年10月19日	(水)	12時54分59秒):	Installation of package "CalcuServ.car'
1	1	-		
e	ディポロイスいたる	E-44-24	7	
T.	74744771	E UDIG.	1	

28.これで Enterprise Server 上に COBOL サービスマッピングが配備されました。 Enterprise Server Admin 画面で以下のようにディプロイが完了していることを各 y 人してください。

	サービス ネームス ペース	オペレーショ ン	サービス クラス	探索順序	リスナー	要求 ハンドラ	実装 パッケー ジ	現 ステータ ス	ステ ータ ス ログ	カスタム 構成
	Test	Test 編集…		1	1 CP 1 HTTP Echo top:10.18.11.28*:9002 (tok-wedge +)			Available	ок	
	Deployer	Deployer 編集…	MF deployment	1	1 CP 1 Web top:10.18.11.28*:41006* (tok-wedge +)			Available	ок	[MF client] scheme=l accept=application/x- listener=Web Service
	JES	JES 編集…	MFJES	1	1 CP 1 Web Services and J2EE top:10.18.11.28*:9003 (tok-wedge +) Cok-wedge +)			Available	ок	
	CICS	cics 編集…	MFCICS	1	1 CP 1 Web Services and J2EE top:10.18.11.28*:9003 (tok-wedge +) Cok-wedge +)			Available	ок	
	ES	ES 編集…	MF ES	1	1 CP 1 Web Services and J2EE top:10.18.11.28*:9003 (tok-wedge +) Cok-wedge +)			Available	ок	
削除	CalcuServ	1 of 1 oper	ations sho	wn						
		.CALCU 編集		1	1 CP 1 Web Services and J2EE top:10.18.11.28*:9003 (tok-wedge +) Cok-wedge +)	MFRHBINP	CalcuServ	Available	ОК	

29.この時作業用ディレクトリの repos/CalcuServ.deploy に EJB ラッパー CalcuServ.jar が自動生成されています。ソースファイルとともに生成されていますので確認してください。

4. テスト用 Webクライアントの生成

インターフェイスマッピングツールキットのクライアント生成機能を使用すると、対話型でパラメタの値を受け取り、 EJB のメソッドを呼び出して結果を表示するような、簡単な Servlet モジュールを含む、ear パッケージを作成で きます。ここでは、これを使用して WebOTX 上の Web クライアントからの呼び出しを行います。

1. COBOL サービスを配備したインターフェイスマッピングツールキットに戻り、[Java インターフェイス] > [CalcuServ] を右クリックして "クライアント生成" を選択します。



- 2. この時作業用ディレクトリの repos/CalcuServ.deploy に パッケージ CalcuServ.ear が自動生成され ています。ソースファイルとともに生成されていますので確認してください。
- WebOTX 統合運用管理コンソールに戻り、左ペインのツリービューで [アプリケーション] を選択します。 右ペインで以下のように、[アップロード] のチェックをオフにし、[コンポーネントタイプ] のプルダウンで "Java EE アプリケーション"を選択し、[アーカイブ名] に上記で生成された CulcuServ.ear のパス名 を入力します。

<u>domain1</u> >>アブリケーション	
 □ ◆ アプリケーション □ リソースアダプタ □ リソース □ アプリケーションサーバ □ ○ 統計情報 □ ◆ モニター □ サーバー覧 	健健 接作 アップロード レ ・ ・
検索: ▼ ビュー 履歴 変更 ローカル検索結果 ◆ アプリケーション	配備可能なファイル名(アップロードオブションにチェックを入れている場合は、ローカルマシン上のファイ ルパス) アーカイブ名 ℓ home/tarot/WebOTX_Demo/work/repos/CalcuServ.deploy/CalcuServ.ear アップロードしたアーカイブのURI 配備(<u>P</u>) 再表示(<u>B</u>)

4. [配備] ボタンをクリックし、以下のように配備が成功したログ表示を確認します。

	配備(<u>P</u>)	再表示(<u>B</u>)
 ログ詳細 		
(情報): [2011/10/19 13:12:18]管理対象生成: CalcuServ (情報): [2011/10/19 13:12:18]管理対象生成: app-configs (情報): [2011/10/19 13:12:18]管理対象生成: CalcuServ.war (情報): [2011/10/19 13:12:15]処理成功 - アプリケーション CalcuS	erv を配備しました。:	SUCCESS.

- 5. 以上で、EJB と Web アプリケーションが同時に配備されました。
- 6. Web ブラウザを開き、"http://<Solaris サーバーアドレス>:80/CalsuServ/CALCU.jsp"を開きます。

🗿 Test Client for CalcuServ.CALCU – Microsoft Internet Explorer 💦 📃	
ファイル(E) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)	2
Ġ 戻る 🔹 💿 - 💌 🗟 🏠 🔎 検索 🧙 お気に入り 🚱 😒 - 🌺 🔜 -	»
アドレス(1) 🥙 http://tok-wedge/CalcuServ/CALCU.jsp 🔹 🔁	移動
😒! - 🖉 Search Web - 🗛 Upgrade your Toolbar Now - 🖉 🥵	»
Test client for CalcuServ.CALCU	~
<u>Back</u>	
Perform the test by entering values:	
calcu_arg1 : 0 calcu_arg2 : 0 Go!	
Back	>

7. 入力フィールドに適宜数値を入力し、[Go!] をクリックします。

8. 以下のように結果が返ることを確認します。

Test client for CalcuServ.CALCU
Back
Perform the test by entering values:
calcu_argl : 11
calcu_arg2 : 22 Go!
Result:
Result
33

以上