

Micro Focus Server Express 5.1 J Red Hat Enterprise Linux 6.2 (x86_64) 動作検証 検証結果報告書

第2部: JBoss 連携編

平成 24 年 4 月 13 日

マイクロフォーカス株式会社



1. 検証概要、目的及びテスト方法

1.1 検証概要

既に Red Hat Enterprise Linux 6.1 x86-64 で動作保証されている Micro Focus Server Express 5.1 J を Red Hat Enterprise Linux Server 6.2 x86_64 上の JBoss Enterprise Application Platform 5.1 との連携で動作検証しました。

1.2 目的及びテスト方法

Micro Focus Server Express 5.1 J は、現在 Red Hat Enterprise Linux 5.1 x86-64 で動作保証済 みです。Red Hat Enterprise Linux Server 6.2 x86_64 は、旧バージョンの Red Hat Enterprise Linux からのバイナリ互換性をサポートしているため、この環境でもそのまま動作する はずです。既に本検証の Part 1 にて、COBOL 言語の機能を網羅的に実行するテストス ートを実行することによって、このことを実際に検証しています。 ここでは、さらに Enterprise Server 上の COBOL サービスを Red Hat Enterprise Linux

Server 6.2 x86_64 上の JBoss Enterprise Application Platform 5.1 上の Java EE アプリケーションから利用できることを検証し、その内容を報告します。

2. 検証環境

ハードウェア

富士通 PRIMERGY TX200 S3 (KVM イメージ)

ソフトウェア

Red Hat Enterprise Linux 6.2 x86_64 JDK 1.6.0_31 64-Bit JBoss Enterprise Application Platform 5.1 Micro Focus Server Express 5.1 J WrapPack 6

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. 検証概要、目的及びテスト方法

以下のテストプログラムを実行することによって検証しました。

- (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JBoss AP上で運用し、COBOLを呼び出す



3.1 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/jboss5/mfcobol-notx.rar
- (3) 使用した Enterprise Server 既定義の ESDEMO をそのまま使用。
- (4) 使用したJBossサーバー既定義のデフォルトサーバーをそのまま使用。
- (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。

3.2 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。

以上、



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。

JBoss EAP もデフォルトでインストールされており、出荷時設定のデフォルトサーバー が開始され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

1. root ユーザでログインし、以下の環境変数を設定します:

COBDIR

```
Server Express のインストール先ディレクトリ
```

JBOSS_HOME

```
JBoss のインストール先ディレクトリ
```

JAVA_HOME

```
JDK のインストール先ディレクトリ
```

PATH

\$COBDIR/bin, \$JAVA_HOME/bin, \$JAVA_HOME/jre/bin, \$JBOSS_HOME/bin を追加

LD_LIBRARY_PATH

\$COBDIR/lib, \$JAVA_HOME/jre/lib/amd64, \$JAVA_HOME/jre/lib/amd64/server を追加



CLASSPATH

\$COBDIR/bin/cobsje スクリプトによって自動設定されるパス、

\$COBOL/lib/mfcobolpure.jar

\$COBOL/lib/javaee5/jboss5/mfconnector.jar

\$JBOSS_HOME/common/lib/jbos-javaee.jar

\$JBOSS_HOME/common/lib/servlet-api.jar

を追加

2. 以下のようにリソースアダプタを JBoss サーバーの deploy ディレクトリにコピーします:

\$ cp \$COBDIR/lib/javaee5/jboss5/mfcobol-notx.rar \$COBDIR/lib/javaee5/jboss5/mfcobol-notxds.xml \$JBOSS_HOME/server/default/deploy

3. JBoss コンソールに以下のようなメッセージが出力されます:

11:56:20,569 INFO [ConnectionFactoryBindingService] Bound ConnectionManager 'jboss.jca:service=ConnectionFactoryBinding,name=eis/MFCobol_v1.5' to JNDI name 'java:eis/MFCobol_v1.5'

 JBoss 管理コンソールのツリービューの [Resource Adapter Archive] 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。



5. また、[Resources] > [Connection Factories] > [No Tx ConnectionFactory] を展開すると以下のように JNDI 名 eis/MFCobol_v1.5 が登録されています。

🍄 Embedded Jopr Core	ログイン中: admin (<u>ログアウ</u>)
 in thel6 i BossAS Servers i Boss EAP 5 (default) Service Binding Manager i Service Binding Sets Applications i Applications i Resources i Boss Messaging i JBoss Web i Scripts i Connection Factories i No Tx ConnectionFactories i Ix ConnectionFactories i Datasources 	: thel5 : JBossAS Servers : JBoss EAP 5 (default) : Resources : Connection Factories : No Tx ConnectionFactories : eis/MFCobol v1.5 サマリ 設定 メトリクス 管理 コンテント Status ー般情報 名前: eis/MFCobol_v1.5 パージョン: 説明: A ConnectionFactory that does not take part in JTA transactions. リンースの特性 Run State: RUNNING Pool JNDI Name: eis/MFCobol_v1.5

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 32-Bit モードでコンパイルします。

\$ cob32 -ug Calcu.cbl

- 4. 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。
- 5. [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。

🙀 Server Express - Interface Mapping Toolkit					
ファイル サービス ヘルプ					
管 サービスインターフェイスの新規作成	Ctrl-N				
🚰 開く	Ctrl=0				
🔄 保存	Ctrl-S				
サービスインターフェイスのエクスポート	`				
サービスインターフェイスのインポート					
閉じる					
終了					

6. 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。

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、 歴聞できます。サービスインターフェイスグループの名 必要があります。	? 関連するサービスを簡単に 弥と置き場所を指定する
名前: SIGroup1	
場所:	
/home/tarot/SIGroup1	参照
	了解 取消し

7. [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。

Server Express - [: SIGroup1]
ファイル サービス ヘルプ
ー図 Webサービス
Javaインターフェイスの新規作成



8. 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。

đ.	
	< 戻る 次へ > 取消し へルプ(+)



9. 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。

8		. 🗆 🛛
	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	?
*	< 戻る 次へ > 取消し へル	ップ(H)

10.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。



ši	
	 マッパーを起動する前に、プログラムのすべてのエントリポイントの外部インターフェイス定義を 設定する省略時マッピングを実行できます。このオプションを選択しない場合は、 マッパーの起動時に空の外部インターフェイス定義を聞きます。 ④ 省略時マッピング ○ None
	< 戻る 次へ > 取消し ヘルプ(H)

11.以下のダイアログで[完了]をクリックします。



12.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。

🚺 Java -	[CalcuServ]					
ファイル	オペレーション	フィールド	削り当て			
	siate Notation cui		8			
		Picture	オペレーション CALCU	Transa	tionOption.	NotSupported
• 8 CA	LCULATOR	ricture	インターフェイスフィール	۳		
	ARG1	9(5) comp-3	名前	方向	Туре	Occurs
	ARG2	9(5) comp-3		人出刀	int	
	RESULT	9(5) comp-3	- ARG2_io		int	
			RESULT_io		int	
			东北田市 2 1118			
			再利用フィールト		Tuno	Occure
			-084		Туре	occurs
			<u></u>			
			COROI割り当て			
			COBOLフィールド		値	
			§]			

13. 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] を選択します。

🊺 Jar	va – [CalcuServ]				
ファイ	(ルーオペレーション	フィールド	割り当て		
	6* # #				
COB	DL入り口点 [CALCU]	/ プログラム	オペレーション CAL	.CU 🔻 TransactionOption.NotSupported	
- 名肌		Picture	インターフェイスフ-	r-ሥሥ	
	- ARG1	9(5) comp-3		方向 Type Occurs	
	- 🗖 AKG2 - 🗖 RESULT	9(5) comp-3 9(5) comp-3	ARG1_io	□☆インターフェイスフィールドの新規作成 □☆インターフェイスフィールドグループの新規作成	
			- C RESULT_io	よ 切り取り Ctrl	-X
				Line Line Ctrl Ctrl	-V
			再利用フィールド	11.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1	
			名前	グループ化	
				グループ化解除	
				マッピング	
			COBOL割り当て	プロパティ	
			COBOLフィールド	値	
II			ali		

14. 以下のダイアログでは [はい] をクリックしてください。

1	
?	選択されたフィールドを本当に削除しますか?
	はいのいいえて

15. 右ペインがクリアされます。ここで左ペインの ARG1, ARG2, RESULT を右ペインにドラッグします。このとき デフォルトではすべて「入力」フィールドとしてマップされます。RESULT は「出力」フィールドなので、以下のよ



うにプロパティを変更します。

オペレーション CALCU 🔽 TransactionOption.NotSupported					
インターフェイスフィールト	۴.				
名前	方向	Type	Occurs		
- 🗖 ARG1	入力	int			
- 🗖 ARG2	入力	int			
RESULT	入力	lint			
		ば インター:	フェイスフィールドの新規作成		
		書 インター:	フェイスフィールドグループの新規作成		
-		👗 切り取り		Ctrl-X	
再利用フィールド		Line (1997) Line (1997) Lin			L
名前		削除			┝
		グループィ	ˈk		1
		グループィ	化解除		
COBOL割り当て マッピング					-
COBOLフィールド		プロパテ	ſ		
					1

16.RESULT フィールドのプロパティダイアログで、以下のように「方向」を「出力」に変更します。

8	
	?
名前: RESULT	
型: int 🔽 Occurs: 0	
方向: 〇 入力 (⑧ 出力) 〇 入出力 「マッピング	
CALCULATOR.RESULT	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。

🚺 Java – [CalcuServ]*			
ファイル オペレーション	フィールド 割	り当て	

COBOL入り口点 [CALCU] /	プログラム	オペレーション CALCU 🔻 Transaction	iOption.NotSupported
	Picture	インターフェイスフィールド	
- ARG1 S	9(5) comp-3	名前 万向 T	ype Occurs
- 🗖 ARG2 🛛 🦻	9(5) comp-3	ーロARG1 入力 int	
□ □ RESULT 9	9(5) comp-3	「日 ARG2 人力 Init	
		重利用フィールド	
		名前 1	ype Occurs
		COBOL割り当て	
		COBOLフィールド	<u>ē</u>
μ]	

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。

😽 Java – [CalcuServ]*		
ファイル オペレーション	ソ フィールド	割り当て
C 保存 L入り口点 [CALCU	l] / プログラム	オペレーション
名前	Picture	110 7 7 7 7
🛛 👇 🗗 CALCULATOR		1/9-711/
- 🗖 ARG1	9(5) comp-3	<u>名前</u>
- 🗖 ARG2	9(5) comp-3	- 🗖 ARG1
	9(5) comp-3	- 🗖 ARG2
		🗆 🗖 RESULT

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。

Server Expres	s - [: SIGroup1]*
ファイル サービン	ス ヘルプ
12 🚅 📕	
🔚 🚾 Webサービス	ζ
🛉 🗖 Java-インター	-フェイス
🔶 🗠 🎾 CalcuSe	
	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。

<u>8</u>	. 🗆 🖂
	?
────────────────────────────────────	
● EJB 生成	
○ Java Bean生成 (J2SE使用)	
了解目	以消し

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。

			?	
一般 ティブロイ	先サーバー アプリケーションフ	アイル EJB生成		
Enterprise Server4	j:			
ESDEMO (10.18	.12.227)	麥更		
🔲 Enterprise Serv	🚺 Enterprise Server			
Enterprise S	Current Development Working	Mode: 32 hit		?
サービス名:	サーバー名 サーバー状	嵬 Deployer名 [Deployerアドレス D	eployer状態
CalcuServ	ESDEMO Started	Deployer 1	0.18.12.227: Avai	lable
「トランザクション				
◎ アプリケーショ				
○ コンテナ管理				
🔲 Username/pas				
			了解	取消し

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。



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。

	, ou
- 〒 ディプロイ先サーバー アプリケーションファイル EIB生成	
レガシーアプリケーションがディプロイメントサーバーにインストール済みか、またはこうからサーバーにディプロイメントサーバーにインストール済みか、または	
ディプロイされたアプリケーションのパス:	
● レガシーアプリケーションをディプロイする:	
アプリケーションファイル	
ファイルを追加	
ファイルを削除	
参照: □ work	
Calcu.gnt	
Calcu.int	
	当し
	?
(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	
レガシーアプリケーションがディプロイメントサーバーにインストール済みか、または	
これからサーバーにティブロイするかを選択してください。	
○ レカシーアプリケーションは既にティブロイ済み ニィブロイまれたアプリケーションのパフ・	
ティンロコ でんしテン ファーションのハス:	
アプリケーションファイル:	
/home/tarot/work/Calcu.cbl /home/tarot/work/Calcu.got	
/ home/tarot/work/Calcu.gnt / home/tarot/work/Calcu.idy ファイルを削除	
「新 取消し	

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 [JBoss 5.1.0] 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 JBoss の \$JBOSS_HOME/common/lib/jbos-javaee.jar と \$JBOSS_HOME/common/lib/servlet-api.jar をコロンで



区切って指定します。それ以外はデフォルト値のままにします。

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
Application server	: Java EE 5 💌 JBoss 5.1.0 💌
EJB 属性	
EJB Version 3 (Ja	wa Compiler version 1.5 or greater is required) 🗹
Bean 名:	CalcuServ
パッケージ名:	com.mypackage.CalcuServ
セッション永続性	³ ③ ステートレス ○ ステートフル
EJB Location:	Local O Remote
ディプロイメント	
CJD -C.	Calcuserveja
Artifive frame.	Calcuserv
J2SE and J2EE At	tributes
Java compiler:	(ont (iava (idk (bin
Classnath for th	a selected application server's EIP and lava connector implementation
Servlet imnleme	e selected application server's EJB and Java connector implementation
J2EE Class Path:	boss-javaee.jar:/opt/java/jboss/common/lib/servlet-api.jar
	了解の間に

- 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。





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。。

	×
ディプロイメントログ	
0011 (2012年03月29日 16時41分01秒): Processing Manifest element in the manifest file	7
0012 (2012年03月29日 16時41分01秒): Processing "Application" section of Manifest	1
0012 (2012年03月29日 16時41分01秒): Processing "FileList" section of Manifest	1
2003 (2012年03月29日 16時41分01秒): File pathname "/home/tarot/work/SIGroup1/repos/CalcuServ.deploy/CalcuServ.idt" is absolute; reducing to [1
2003 (2012年03月29日 16時41分01秒): File pathname "/home/tarot/work/Calcu.cbl" is absolute; reducing to filename "Calcu.cbl"	ł
2003 (2012年03月29日 16時41分01秒): File pathname "/home/tarot/work/Calcu.gnt" is absolute; reducing to filename "Calcu.gnt"	1
2003 (2012年03月29日 16時41分01秒): File pathname "/home/tarot/work/Calcu.idy" is absolute; reducing to filename "Calcu.idy"	ł
0012 (2012年03月29日 16時41分01秒): Processing "Service" section of Manifest	ł
0019 (2012年03月29日 16時41分01秒): Manifest processing completed	ł
0020 (2012年03月29日 16時41分01秒): Adding service and package objects to directory	ł
0021 (2012年03月29日 16時41分02秒): Using directory at mrpi://10.18.12.227:86	ł
0030 (2012年03月29日 16時41分03秒): ES server "ESDEMO" notified service "CalcuServ.CALCU" is available	
0002 (2012年03月29日 16時41分03秒): Installation of package "CalcuServ.car" finished with 4 warnings	1
	٢
(1) ディプロイメント正常終了	
0	
ОК	

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確認してください。

	Server	ESDEMO [開始 🔨]							
サーバー.	דגען	(3) <mark>-</mark>	+ービス (6)		ハンドラ (2) パッケージ (1)	<u>ا</u>				
サービス表	表示フィルタ	ネームス	<mark>ペース:</mark>		オペレー	ジョン:			ク	57:
1 - 6 of 6	displayabl	e namespa	ces from a	tota	l of 6				S	how
	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 構成
	Test	Test 編集…		1	1 CP 1 HTTP Echo top:10.18.12.227*:9002 (rhel6)			Available	ок	
	Deployer	Deployer 編集…	MF deployment	1	1 CP 1 Web top:10.18.12.227*:49800* (rhel6)			Available	ок	[MF acce liste
	JES	JES 編集…	MF JES	1	1 CP 1 Web Services and J2EE top:10.18.12.227*:9003 (rhel6)			Available	ок	
	CICS	cics 編集…	MF CICS	1	1 CP 1 Web Services and J2EE top:10.18.12.227*:9003 (rhel6)			Available	ок	
	ES	ES 編集	MF ES	1	1 CP 1 Web Services and J2EE top:10.18.12.227*:9003 (rhel6)			Available	ок	
削除	CalcuServ) of 1 open	ations sho	wn						
		.CALCU 編集		1	1 CP 1 Web Services and J2EE top:10.18.12.227*:9003 (rhel6)	MFRHBINP	CalcuServ	Available	ок	
追加		·								

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 JBoss 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. 以下のように生成が正常終了することを確認します。



- 3. 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。
- 4. 生成された .ear を JBoss サーバーの deploy ディレクトリにコピーします。

\$ cp CalcuServ.ear \$JBOSS_HOME/server/default/deploy

5. JBoss 管理コンソールの [Applications] > [Enterprise Application (EAR)] の下に CalcuServ.ear が登録さ れたことを確認します。



😵 Embedded Jopr Core	ロヴイン中: admin (<u>ロジアウト</u>)
 minute in the initial initial	: thel6 : JBossAS Servers : JBoss EAP 5 (default) : Applications : Enterprise Application (EAR)s : CalcuServ.ear CalcuServ.ear サマリ 設定 メトリクス 管理 Status: ↓A コンテント 一般情報 名前: CalcuServ.ear バージョン: 説明: JEE enterprise application (EAR) リソースの特性

- 6. 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。
- 7. Web ブラウザを開き、"http://<Red Hat サーバーアドレス>:8080/CalsuServ/CALCU.jsp" を開きます。

🗿 Test Client for CalcuServ.CALCU – Microsoft Internet Explorer 📃 🗖	$\mathbf{ imes}$					
ファイル(E) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルブ(H)						
중 戻る ・ ② - 💌 🖉 🏠 🔎 検索 🌟 お気に入り 🧐 🙆 - 🌺 🚍 ・	**					
アドレス(1) 🛃 http://tok-wedge/CalcuServ/CALCU.jsp	麵					
Search Web 🔹 🛕 Upgrade your Toolbar Now 🔹 🖉 🚯	>>					
Test client for CalcuServ.CALCU	~					
Back						
Perform the test by entering values:						
calcu_arg1 : 0 calcu_arg2 : 0 Gol						
Back	>					

8. 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。



9. 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						
Result:						
Variable Value						
Result 33						

以上、