

Micro Focus Net Express 5.0 J BizBrowser 4.1 Web サービス連携 動作検証結果報告書

平成20年2月5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

Micro Focus Net Express 5.0 J の Enterprise Server が提供する Web サービス機能は、多くの SOAP 準拠アプリケーションからの利用実績があります。開発者はこの機能を用いて、COBOL で記述された堅牢なビジネスロジックを Web サービスによって分散環境で幅広く活用してゆく ことができます。

Biz/Browser は高操作性、高速なレスポンスを実現するリッチクライアントツールであり、Web サービスによるアプリケーション連携もサポートしています。

本報告書は、Biz/Browser 4.1 での Web サービス連携機能を使用して、Net Express 5.0 J による COBOL Web サービスとの連携を検証し、その結果を報告するものです。

1.2 目的及びテスト方法

Micro Focus Server Express 5.0 J の Enterprise Server が提供する Web サービス機能は、 SOAP 1.1 仕様に準拠した設計となっており、この互換性から Biz/Browser 4.1 からの呼び出し は可能であるはずです。今回は、簡単な COBOL テストプログラムを Enterprise Server 配下で Web サービス化し、Micro Focus のツールによって自動生成された WSDL を、Biz/Designer の WSDL for CRS ユーティリティを使用して、CRS に自動変換して呼び出すことによって、このこ とを実際に検証しました。

2. 使用ハードウェア及びソフトウェア一覧

(1) HTTP サーバ

Dell Dimension 4600C Intel Pentium 4 2.8GHz 1Gb memory Windows XP SP2 Internet Information Service 5.1

(2) COBOL サーバ

Dell Dimension 4600C Intel Pentium 4 2.8GHz 1Gb memory Windows XP SP2 Micro Focus Net Express 5.0 J WrapPack 4

(3) クライアント

Dell Dimension 4600C Intel Pentium 4 2.8GHz 1Gb memory Windows XP SP2 アクシスソフト Biz/Browser 4.1

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

COBOL の索引編成ファイルから指定されたキーのレコードを READ しその内容を返す、 簡単な COBOL サブルーチン。Enterprise Server へはすべてデフォルトの指定で Web サ ービスとしてディプロイした。

(2) 使用した Web サービス種別

Doc-Literal 形式

- (3) 使用した Enterprise Server既定義の ESDEMO をそのまま使用した。
- (4) 使用した Biz/Browser クライアント

COBOL Web サービスに引き渡すパラメタを入力するための簡単なウィンドウを作成し、 ここから WSDL for CRS が自動生成した CRS のスタブメソッドを呼び出すことで、 COBOL サービスを呼び出し、COBOL から返された結果を簡単なダイアログ上に表示す る。

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。 なお、WSDL for CRS ユーティリティの現状の制限事項のため、自動生成された CRS を一箇所 手作業で修正する必要がありました。

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

高操作性、高速なレスポンスを実現するリッチクライアントツール Biz/Browser のアプリケーションから、既存の Micro Focus Net Express 5.0J の Web サービスを問題なく使用できることが検証できました。

この方法により、Biz/Browser の持つ豊かな対話操作機能・管理の容易さと、COBOL 言語の持つ強力なビジネスロジック記述力の双方を活かして、堅牢かつ変化に強い企業情報システムを構築してゆくことが可能となります。

以上

補足: 検証手順の詳細

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

🕘 10.18.11.63 🤤	pc01): Micro Focus Enterprise Server Administration – Microsof	t Internet Explor 🔳 🗖 🔀							
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) 🦧									
Ġ 戻る 🔹 🕥	- 🖹 🙆 🏠 🔎 検索 🌟 お気に入り 🧐 😒 婱 📝	- 🔜 🛍 🦓							
アドレス(D) 🙆 http:/	//127.00.1:86/	🔽 🔁 移動 リンク 🎽							
MICRO FOCUS Server Administration pc01 (10.18.11.63)									
Home	Status MDS0000I OK	Tue Jan 15 17:21:44 2008							
nome		[Developer] [Page id: 1000]							
アクション アドレス更新 保存 インポート 復回	アクション アドレス更新 保存 インボート								
すべて削除		S BU S DX S Z							
構成 オプション ユーザ グループ	タイプ 名前 現テータ ステータス ブロセス ブロセス ブロセス ブロセス 編集< 単価 単価 ESDEMO 開始 詳細 1 top:10.18.11.637:9000 (pe01) * 1 MDS30000 Server started 1 MDS30000 Server started	オブジェクト 説明 サ Communications 8 ビ 詳細 Services							
表示 ディレクトリ 統計 セッション	「登止 3 リスリー 第4項 「登止 「日本」	y ス - 2 5 2 f 詳細 in ラ							

2) 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。

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 INPUT CUST-MASTER.						
MOVE CUSTID TO FS-CUSTID.						
READ CUST-MASTER INVALID CONTINUE						

	END-READ.		
CLOSE	CUST-MASTER.		
MOVE	FS-CUSTNAME	T0	CUSTNAME.
MOVE	FS-CUSTCOMPANY	T0	CUSTCOMPANY.
MOVE	FS-CUSTEMAIL	T0	CUSTEMAIL.
EXIT	PROGRAM.		

- 3) Net Express でプロジェクトを新規作成し、このプログラムを追加します。
- 4) [一般デバッグビルド] でプロジェクトをビルドします。

ቼ <mark>⊨</mark> ን¶ን⁵ェንŀ:ReadCustSOA.A	P P			
ヒ°ルト°タイフ° - 般デパック°ヒ°ル -	*		▼ Filter	0 of 2 s
🖃 🥭 ReadCust.int	Name	Туре	Last Updated	Size
ReadCust.cbl	ReadCust.cbl ReadCustMai	COBOL COBOL	2006年5月26日 1:09: 2007年5月16日 0:41:	118C 1289
1	<			>

5) Interface Mapping Toolkit を利用して、Web サービスのデフォルトマッピングを作成します。 その後、第一パラメタの CUSTID のみを「入力」にし、残りの3パラメタをすべて「出力」に変更します。

COBOL IOFU	ポイント [REAC	Operation READCU	JST								
名前 由 custid	PICTURE 9(9) comp-5	ſ <u>Ⴢ</u> ┩ーフェースフィールド									
i custname i custcompany i custemail	×(30) ×(30) ×(30)	名前 custid_in custname_out custcompany_out custemail_out	<u>方向</u> 入力 出力 出力 出力	型 int string string string	OCCURS						
		再利用フィールド									

6) マッピング属性では、Enterprise Server 名を 1) で指定したローカルの ESDEMO を指定し、 「アプリケーションファイル」でコンパイル済みの .int コードと、索引データファイルを指定します。

マッピング属性					
	ディプロイメントサーバー アフッリケーションファイル マッピック 属性				
	Enterprise Server 名: ESDEMO (10.18.11.105)				
	厂 Enterprise Server 実行時環境の使用				

7) サービスをディプロイします。



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。

🔺 🚽 🕨 Server ESDEMO [開始 🗸]										
サーバー… 】 リスナー (3)】 サービス (6)】 ハンドラ (2) <mark>パッケージ (1)</mark>										
1 - 1 of 1 packages at a time <<前へ										
	名前	現 ステータ ス	ステータスログ	パッケージモジュール	ЮТ	パッケージ バス	カスタム構成	説明		
編集	http://tempuri.org/ReadCustW/S	Available	ок		C:\Program Files\Wicro Focus\Net Express 5.0 \Base\deploy\ReadCust\WS.r- iXIZIL/ReadCust\WS.idt	C:\Program Files\Micro Focus\Net Express 5.0 \Base\deploy\ReadCust\WS.r- iXIZIL		created 14 06-21 from ReadCust iXIZIL/Rea		
追加										

- 9) Net Express プロジェクトの配下に WSDL が生成されていることを確認しておきます。
- 10) Biz/Designer の WSDL for CRS ユーティリティを起動します。
- 11) 「WSDL ファイルを開く(F1)」を選択し、Net Express が生成した WSDL を指定します。

🗷 メソッドの 選択	\mathbf{X}
Class名[=出力ファイル名]	ReadCustWS
Encoding Type	UTF-8 💌 SOAP 1.1 💌
soapパス	soap.car
メソッドエレメントNameSpace	ns1
ラベル付加文字列(空白可)	
soapヘッダ追加(v1.01以降)	Utati 💌
出力 メン	ルド名 サービス
READCUST	SOAP

12) 「soap パス」として soap.car を選択し、[変換] をクリックすると、以下のようにスタブクラスの CRS が生成されます。



- 13) これを CRS ファイルに保存します。
- 14) 現状の WSDL for CRS ユーティリティの制限によって、Net Express が生成した WSDL 中に記述された エンドポイントアドレスが正しく取り込まれていません。そこで9行目の

string host = "http:/";

の箇所を、以下のように Enterprise Server のホスト名とポート番号に書き換えておきます。

string host = "http:/localhost:9003";

15) Biz/Designer を使用して、この CRS クラスを呼び出す簡単なウィンドウを作成します。

ここでは以下のようなものを作成しました。

frmdialog.crs

Form:String frmDialog {
BgColor = \$STD;
Border = \$TRUE;
Height = 300;
Width = 450;
/*
ダイアログの CRS を Get し、制御を移します
*/
Button Button0 {
FontFace = \$STD;
FontKind = \$STD;
FontSize = 10;
Height = 22;
Title = "ボタン0";
Width = 326 ;
X = 94;
Y = 126;
<pre>function OnTouch(e) {</pre>
/* ダイアログの CRS をロードし */

```
frmDialog.Get("dlgDialog.crs");
                         /* 受け渡しパラメータをセットします */
                        frmDialog.dlgDialog.SetParam(frmDialog.txtName);
                }
        ł
        Label Label1 {
                X = 14;
                Y = 14;
                Width = 414;
                Height = 39;
                Value = "COBOL 連携のサンプルです。¥n このフォームで入力した顧客 ID
を COBOL による Web サービスに引渡し、結果をダイアログで表示します。";
        }
        Label Label2 {
                X = 14;
                Y = 70;
                Width = 62;
                Height = 15;
                Value = "顧客 ID";
        }
        TextBox txtName {
                X = 94;
                Y = 62;
                Width = 158;
                Height = 24;
        }
        Label Label4 {
                X = 13;
                Y = 157;
                Width = 56;
                Height = 15;
                Value = "結果";
        }
        Label labResult {
                X = 94;
                Y = 158;
                Width = 326;
                Height = 120;
                Border = $TRUE;
        }
        /*
        ユーザーイベントです。
        呼び出したダイアログからの戻り値をイベントとして
        受け取ります。
        */
        function OnDialogEvent(e) {
                if( e.code == "OK" ) {
                        frmDialog.labResult = e.result;
                } else {
                        frmDialog.labResult = "++vvvvlstub";
                }
                dlgDialog.Delete();
        }
}
```

dlgDialog.crs

import ReadCustWS; /*

```
frmdialog.crs から Get されます
*/
dialog dlgDialog {
         ReadCustWS ReadCustWS;
         Function SetParam(pID) {
                  READCUST_rvf READCUST_rvf;
                  READCUSTResponse READCUSTResponse;
                  READCUST_rvf.custid_in = 12345;
                  READCUSTResponse = ReadCustWS. READCUST(READCUST_rvf);
                  dlgForm. labName = READCUSTResponse.custname_out;
                  dlgForm. labCompany = READCUSTResponse.custcompany_out;
                  dlgForm.labEmail = READCUSTResponse.custemail_out;
         Form:String dlgForm {
                  Height = 231;
                  Width = 408;
                  Label Label1 {
                            X = 13;
                            Y = 18;
                            Width = 96;
                            Height = 22;
                            Value = "顧客 ID";
                  Label:String labID {
                            X = 125;
                            Y = 18;
                            Width = 235;
                            Height = 22;
                            Border = true;
                  Label Label2 {
                            X = 12;
                            Y = 49;
                            Width = 88;
                            Height = 24;
                            Value = "顧客名";
                  }
                  Label:String labName {
                            X = 125;
                            Y = 49;
                            Width = 158;
                            Height = 22;
                            Border = true;
                  Label Label3 {
                            X = 12;
                            Y = 80;
                            Width = 88;
                            Height = 24;
                            Value = "会社名";
                  Label:String labCompany {
                            X = 125;
                            Y = 80;
                            Width = 158;
                            Height = 22;
                            Border = true;
                  Label Label3 {
                            X = 12;
                            Y = 111;
```

```
Width = 88;
                           Height = 24;
                           Value = "Email";
                  }
                  Label:String labEmail {
                           X = 125;
                           Y = 111;
                           Width = 158;
                           Height = 22;
                           Border = true;
                  }
                  Button btnOK {
                           Height = 30;
                           Title = "OK";
                           Width = 102;
                           X = 288;
                           Y = 152;
                           function OnTouch(e) {
                                     var ev = new Event {
                                              eventName = "DialogEvent";
                                              String code = "OK";
                                              String result;
                                     }
                                     ev.result = "COBOLの呼び出し成功";
                                     postEvent(ev);
                           }
                  Button btnCancel {
                           Height = 30;
                           Title = "Cancel";
                           Width = 102;
                           X = 176;
                           Y = 152;
                           function OnTouch(e) {
                                     var ev = new Event {
                                              eventName = "DialogEvent";
                                              String code = "CANCEL";
                                              String result = "++vvtultite";
                                     }
                                     postEvent(ev);
                           }
                  }
         }
}
```

- 16) これらの CRS ファイルと soap.car モジュールを Internet Information Server の Web サイトの 仮想ディレクトリにコピーします。
- 17) Biz/Browser を起動し、上記の frmdialog.crs の URL を開きます。

Biz-Collect	tions Biz/Browser XE
ファイル(E) 表示	M VILA
COBOL連携の このフォームで をダイアログで)サンプルです。 『入力した顧客ID をCOBOLIこよる Webサービスに引渡し、結果 『表示します。
顧客ID	
	ボタン0
結果	

- 18) あらかじめ索引ファイルに投入しておいたデータの顧客 ID を入力し、[ボタン 0] をクリックします。
- 19) COBOL Web サービスが呼び出され、索引ファイルから READ されたデータが Biz/Browser の ダイアログ上に表示されます。

B		×				
顧客ID	12345	-				
顧客名	山田太郎					
会社名	山田産業株式会社					
Email	ty@yamada.com					
	Cancel OK					

以上