



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が、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

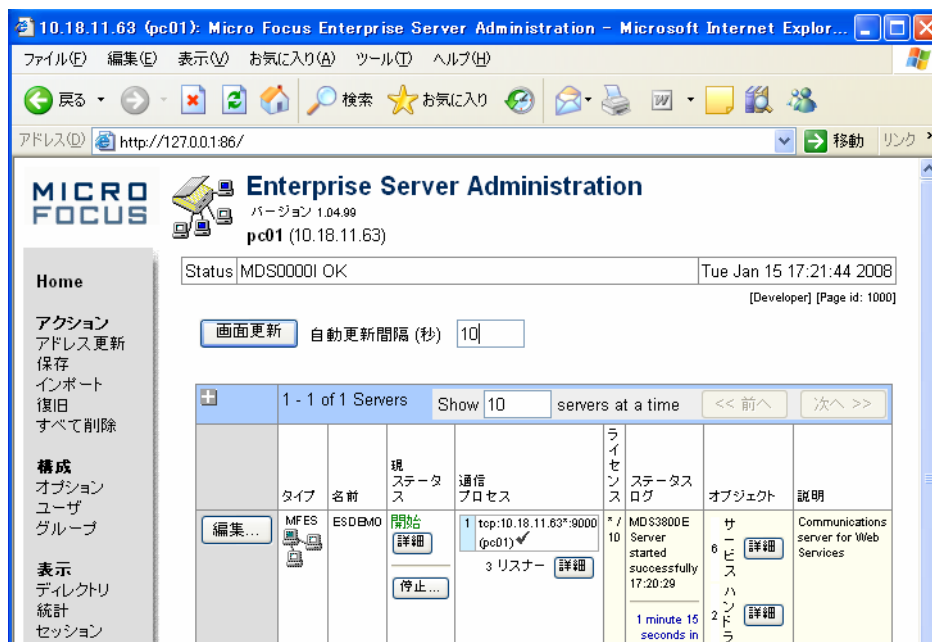
高操作性、高速なレスポンスを実現するリッチクライアントツール **Biz/Browser** のアプリケーションから、既存の **Micro Focus Net Express 5.0J** の Web サービスを問題なく使用できることが検証できました。

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

- 1) 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。



- 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。

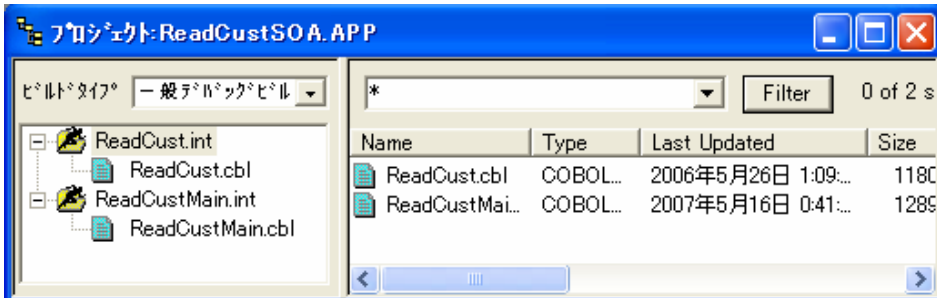
```
FILE-CONTROL.  
    SELECT CUST-MASTER ASSIGN TO "CUST.dat"  
    ORGANIZATION INDEXED RECORD KEY FS-CUSTID  
    ACCESS MODE RANDOM.  
DATA          DIVISION.  
FILE          SECTION.  
FD CUST-MASTER.  
01 CUST-REC.  
    05 FS-CUSTID      PIC X(4)   COMP-5.  
    05 FS-CUSTNAME   PIC X(30).  
    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
```

```

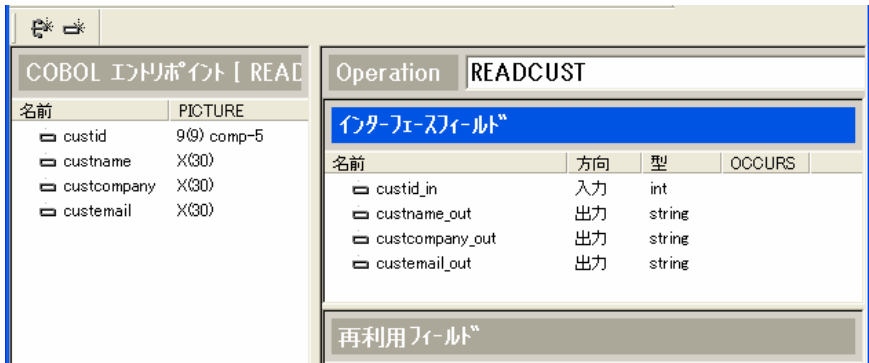
                END-READ.
CLOSE          CUST-MASTER.
MOVE          FS-CUSTNAME   TO  CUSTNAME.
MOVE          FS-CUSTCOMPANY TO  CUSTCOMPANY.
MOVE          FS-CUSTEMAIL  TO  CUSTEMAIL.
EXIT          PROGRAM.

```

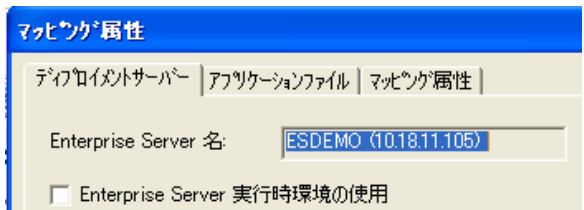
- 3) Net Express でプロジェクトを新規作成し、このプログラムを追加します。
- 4) [一般デバッグビルド] でプロジェクトをビルドします。



- 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。



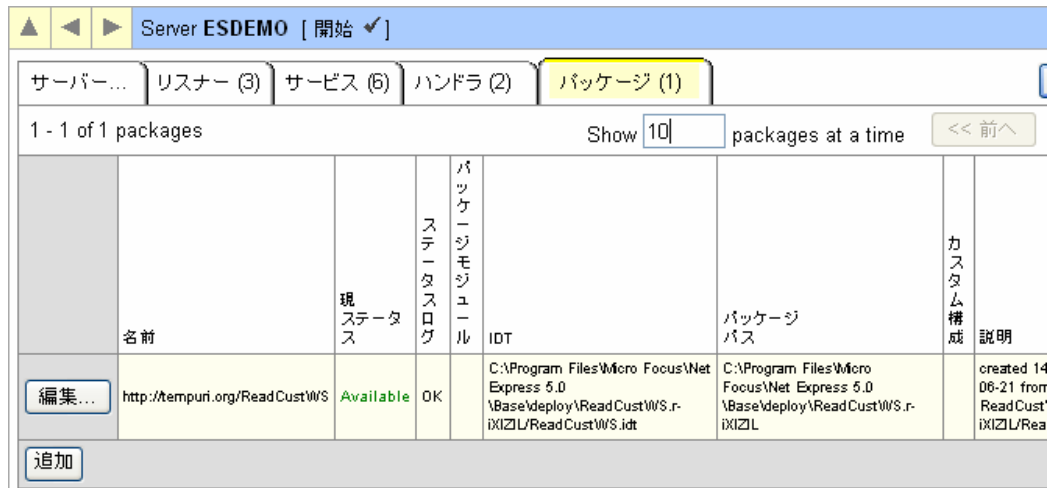
- 6) マッピング属性では、Enterprise Server 名を 1) で指定したローカルの ESDemo を指定し、「アプリケーションファイル」でコンパイル済みの .int コードと、索引データファイルを指定します。



7) サービスをデプロイします。



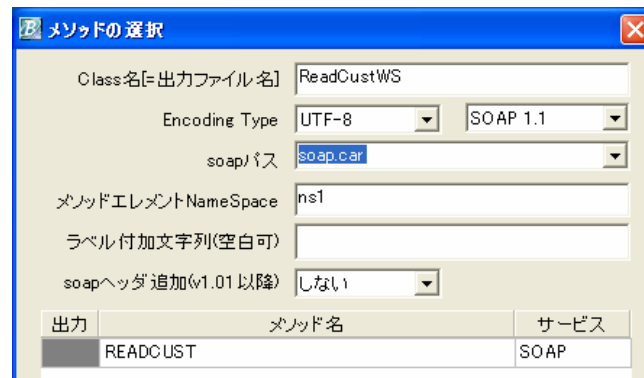
8) Micro Focus Enterprise Server Admin でサービスが正しくデプロイされていることを確認します。



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。



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;
        function OnTouch(e) {
            /* 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);
    }
}
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 = "キャンセルしました";
    }
    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;
    }

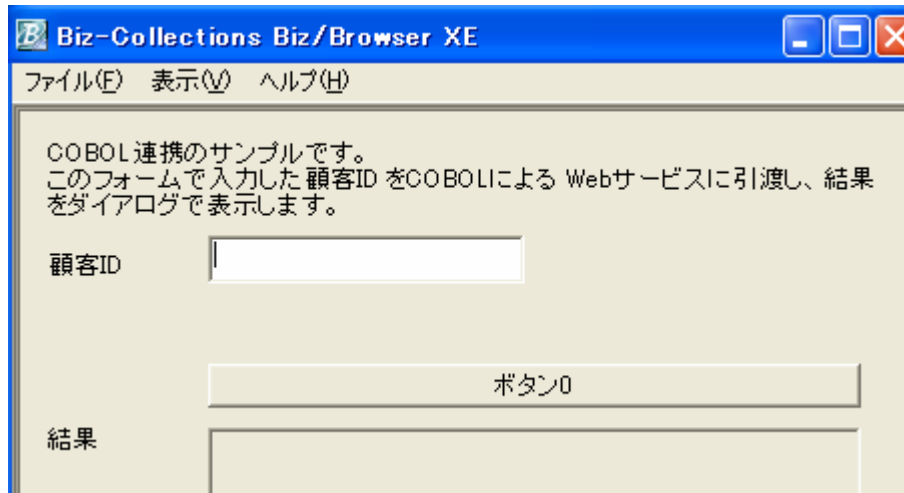
    Button btnOK {
        Height = 30;
        Title = "OK";
        Width = 102;
        X = 288;
        Y = 152;
        function OnTouch(e) {
            var ev = new Event {
                eventName = "DialogEvent";
                String code = "OK";
                String result;
            }
            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 = "キャンセルしました";
            }
            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。



- 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。



以上