

1.1. サンプル 6

CORBA-BC を使用したサーバアプリケーションの呼び出しを SAAJ クライアントから実行する手順を、サンプルアプリケーションを例にして説明します。

このサンプルは、WebOTX ESB をマルチプロセスモードで動作させることを前提としています。

この項で説明する手順は次のとおりです。

- サンプルのインストール
- ドメインの起動
- アプリケーショングループとプロセスグループの作成と起動
- サーバアプリケーションの配備
- サブレットの配備
- サービスアセンブリの配備と開始
- クライアントアプリケーションのコンパイル実行

1.1.1. サンプルのインストール

CORBA-BC のサンプルアプリケーションは、WebOTX マニュアルのインストールに伴って、次の場所に配置されますので、zip ファイルを展開してください。

＜WebOTX マニュアルインストールディレクトリ＞¥samples¥esb¥esb.zip

ファイル構成

- ・ 6¥ConvertTest.cpk CORBA BC から呼び出すサーバアプリケーションです。
- ・ 6¥CORBABCSampleSOAPBinding.war SOAP BC 用のサブレットです。
- ・ 6¥artifacts¥corbabc_sample_convertSA.zip CORBA-BC のサービスアセンブリです。
- ・ 6¥clients¥saa_client.zip クライアントアプリケーションです。

1.1.2. ドメインの起動

ドメインの起動

ドメインが起動されていない場合は、起動してください。

```
otxadmin> start-domain domain1
```

1.1.3. アプリケーショングループとプロセスグループの作成と起動

WebOTX AS の Standard Edition または Enterprise Edition で、マルチプロセスモードにて ESB をご利用になる場合、アプリケーショングループとプロセスグループを作成します。

アプリケーショングループの作成

アプリケーショングループを作成します。

```
otxadmin> login --user admin --password adminadmin
otxadmin> create-apg apg1
Command create-apg executed successfully.
```

プロセスグループの作成

プロセスグループは CORBA アプリケーション用と JBI コンテナ用と 2 つ用意します。

プロセスグループ (CORBA アプリケーション用) を作成します。

```
otxadmin> create-pg --version 7 --kind java2 --apgroup apg1 pg1
Command create-pg executed successfully.
```

プロセスグループ (JBI コンテナ用) を作成します。

```
otxadmin> create-pg --version 7 --kind j2ee --apgroup apg1 pg2
Command create-pg executed successfully.
```

JBI コンテナの有効化

プロセスグループ起動時に JBI コンテナを起動するように設定します。

```
otxadmin> set tpsystem.applicationGroups.apg1.processGroups.pg2.
enabledJBIContainer=true
```

(※) 改行していますが実行する際は、1 行で記述して実行してください。

アプリケーショングループ プロセスグループの起動

作成したアプリケーショングループを起動します。

```
otxadmin> start-apg apg1
Command start-apg executed successfully.
```

1.1.4. サーバアプリケーションの配備

サーバアプリケーションの配備

ConvertTest.cpk があるディレクトリで、以下のコマンドを実行します。

```
otxadmin> deploy --apgroup apg1 --pgroup pg1 ConvertTest.cpk  
Command deploy executed successfully.
```

スタブクラスのクラスパスへの追加

ConvertTestBean.cpk が配備されると、次のディレクトリに「ConvertTest.jar」が展開されます。

%INSTANCE_ROOT%\applications\corba-apps\ConvertTest\ConvertTest.jar

この jar ファイルを、次のディレクトリにコピーしてください。

%INSTANCE_ROOT%\jbi\bindings\CORBABinding\install_root\workspace\lib

CORBABinding の再起動

以下のコマンドを実行して、CORBABinding の再起動を行ってください。ただし、WebOTX AS Standard Edition または Enterprise Edition で、マルチプロセスモードにて ESB をご利用になる場合は、プロセスグループの再起動を行ってください。

CORBABinding を停止します。

```
>jbiadmin.bat --user admin --password adminadmin --port 6212 stop-component  
CORBABinding  
Stopped Component CORBABinding
```

CORBABinding をシャットダウンします。

```
>jbiadmin.bat --user admin --password adminadmin --port 6212 shut-down-component  
CORBABinding  
Shut Down Component CORBABinding
```

CORBABinding を開始します。

```
>jbiadmin.bat --user admin --password adminadmin --port 6212 start-component  
CORBABinding  
Started Component CORBABinding
```

名前サーバへの登録

プロセスグループ起動時に CORBA アプリケーションの名前サーバへの登録を行うように設定します。

```
otxadmin> set applications.corba-applications.ConvertTest.ConvertTest.jar.  
function-ConvertTest.bindType=0
```

(※) 改行していますが実行する際は、1 行で記述して実行してください。

1.1.5. サブプレットの配備

サブプレットの配備

CORBABCSampleSOAPBinding.war があるディレクトリで、以下のコマンドを実行します。

```
otxadmin> deploy --apgroup apg1 --pgroup pg2 CORBABCsampleSOAPBinding.war  
Command deploy executed successfully.
```

(*) アプリケーショングループ名とプロセスグループ名は、WebOTX AS Standard Edition または Enterprise Edition で、マルチプロセスモードにて ESB をご利用になる場合にだけ指定してください。

1.1.6. サービスアセンブリの配備と開始

サービスアセンブリの配備

corbabc_sample_convertSA.zip があるディレクトリで、以下のコマンドを実行します。

```
>jbiadmin.bat --user admin --password adminadmin --port 6212 deploy-service-assembly  
corbabc_sample_convertSA.zip apg1 pg2  
Deployed Service Assembly corbabc_sample_convertSA
```

(*) --port には該当ドメインのポート番号を指定してください。アプリケーショングループ名とプロセスグループ名は、WebOTX AS Standard Edition または Enterprise Edition で、マルチプロセスモードにて ESB をご利用になる場合にだけ指定してください。

サービスアセンブリの開始

配備したサービスアセンブリを開始します。

```
> jbiadmin.bat --user admin --password adminadmin --port 6212 start-service-assembly  
corbabc_sample_convertSA  
Started Service Assembly corbabc_sample_convertSA
```

(*) --port には該当ドメインのポート番号を指定してください。

1.1.7. クライアントアプリケーションのコンパイルと実行

クライアントアプリケーション

clients¥saa_client.zip

を展開してください。

ファイル構成

- ・ compile.bat コンパイル実行バッチです。
- ・ run.bat クライアント実行バッチです。
- ・ src¥Test.java クライアントのソースです。
- ・ class¥Test.class クライアントのクラスです。
- ・ xml¥test_sample.xml、xml¥header_sample.xml SOAP のリクエストメッセージを作成するための xml です。

コンパイル

次のライブラリをアプリケーションの環境変数クラスパスに追加してください。

jar ファイル

%AS_INSTALL%¥lib¥j2ee.jar

%AS_INSTALL%¥lib¥saa-impl.jar

%AS_INSTALL%¥lib¥endorsed¥xercesImpl.jar

%AS_INSTALL%¥lib¥endorsed¥xml-apis.jar

コンパイル、および、実行時に使用する JDK のコマンドが環境変数パスに追加されていない場合は、追加してください。

compile.bat を実行し、コンパイルを行ってください。

```
>compile.bat
```

実行

クラスパスに class ディレクトリを追加した後、run.bat を実行し、クライアントを実行します。

```
>run.bat
```

呼び出しに成功すると、以下のように表示されます。

```
-----
<?xml version="1.0" encoding="UTF-8"?>
<convertResponse xmlns="http://www.nec.co.jp/jbi/sample1.wsdl"><result
xmlns="">ok</result></convertResponse>
times/messages = 4516.0
```