

## OServ (オブジェクトサーバー)

### オブジェクトサーバーの機能

OServ(オブジェクトサーバー)は、従来CORBA製品などで見られる面倒なサーバー検索処理を無くし、単体マシン上でJava言語のクラスを利用するのと同じようにサーバーに登録されたサービスクラスを利用するための仕組みを提供するソフトウェアシステムです。利用者は、システムに添付されているツールで、スタブクラスを生成、リンクするだけで、ネットワーク経由でのサービスクラスの利用が可能です。

また、パーシステント機能をサポートしているため、登録されるサービス機能は、複数の要求にまたがって、継続動作が可能です。

### ソフトウェアの著作権

本ソフトウェアは、シェアウェア（有料ソフトウェア）として販売されるもので、正規の使用（機能、仕様の評価のための試用でないもの。）の場合には、購入手続きを行ってください。

### 問い合わせ

問い合わせは下記のメールアドレスへ直接メールを送信して下さい。

uewxiun@pb3.so-net.ne.jp

### 動作環境

Java 1.5.0\_13

OS - Mac OS Darwin 9.6.1

Unix, Linux, Solaris

Windows

### インストール

ファイルは、zipコマンドによりアーカイブされています。Windowsのフォルダとしてオープンして内容をコピーして御使用下さい。

特別にファイルを配置する必要は無く、同梱されているjarファイルをライブラリとして指定して頂ければ動作します。

分散処理実現のためのミドルウェアで、サーバーサイドとクライアントサイドに分かれて動作します。

```
# mkdir SERVER
# cd SERVER
# copy ../OServ.jar .
# java -classpath ../OServ.jar oserv.ObjectServer
java -classpath ../OServ.jar oserv.ObjectServer
```

```
# mkdir CLIENT
# cd CLIENT
# copy ../OServ.jar .
```

```
# move ../SampleServer.jar .      (一般処理のサンプル)
# move ../PersistentServer.jar .  (パーシステント処理のサンプル)
```

(サプルサービスのサーバーへの登録。)

```
# java -classpath ../OServ.jar oserv.ObjectTool sample.SampleServer
# java -classpath ../OServ.jar oserv.ObjectTool test.PersistentServer
# java -classpath ../OServ.jar oserv.ObjectTool -l
(ホスト名を指定する場合には次のように指定して下さい。)
```

(登録パッケージの削除をするときは、次のように -d オプションを使用して下さい。)

```
# java -classpath ../OServ.jar oserv.ObjectTool -d sample.SampleServer
```

(登録パッケージの更新は、削除せずにそのまま上書きが可能です。)

```
# java -classpath ../OServ.jar -Doserv.service.hostname=localhost
oserv.ObjectTool -l
```

(遠隔では、localhostの所にネットワーク上のObjectServerを起動した  
ホスト名を指定して下さい。)

```
# move ../userclient .
# cd userclient
# java -classpath ../OServ.jar;./ sample.SampleClient
( java -Doserv.service.hostname=server_hostname
    -classpath ../OServ.jar;./ sample.SampleClient)
# java -classpath ../OServ.jar;./ test.PersistentClient
```

上記のクライアント要求では、既にスタブクラスが作成されています。実際にユーザが自分自身のサービスを作成して使用する場合には、実装サービスを定義するInterfaceファイルを基に、スタブクラスを生成する必要があります。  
生成されたスタブクラスをサーバーに登録されているサービスクラスと同様に利用できるので、クライアント側からは、サービスを意識する事無く簡単に使用する事ができます。下記がその構築例になります。

```
# del test¥PersistentServer.java
# java -classpath ../OServ.jar;¥ oserv.ObjectGenerator test/
PersistentService PersistentServer
(PersistentServiceは、インタフェースクラス名です。testの下に
PersistentService.classとしてコンパイルしたライブラリを配置します。)
```

#### Windowsでの操作の注意点

javaのクラスパス（classpath）の指定で、複数のパスを指定する場合には、コロ  
ン（:）ではなく、  
セミコロン（;）を使用して下さい。

```
# java -classpath ../OServ.jar;./ test.PersistentClient
```

削除コマンドは、rmではなく、delを、移動コマンドは、mvではなく、move、複  
製コマンドは、cpではなく、copyを使用して下さい。