



Xone インストール／設定ガイド

Xone Personal Version1.0.5

(有)ファイバーワークス

目次

1.動作環境.....	3
2.インストール.....	3
3.データベースの初期化.....	3
4.初期化の情報を書き込む.....	3
5.システム構成と設定.....	5
5.1 構成別の設定.....	6
5.2 接続テスト.....	9
5.3 ログ出力.....	10
ソフトウェア情報.....	11

1.動作環境

Xone(ゾーンと読みます)のパーソナル版はすべて **Java** で書かれているので、複数の OS で動作します。ただし、**Java** のバージョンは **1.4.0** 以上が必要です。また、**RDB** でデータを管理するので、**RDB** とその **RDB** 用の **JDBC** ドライバが必要です。これらは添付していないので、それぞれのウェブサイトからダウンロードしてください。

対応している **RDB** は次のとおりです。これ以降のバージョンの **RDB** では動作するはずですが、これ以前のバージョンでは動作は保証できません。

- PostgreSQL 7.3.3
- MySQL 4.0.18
- Oracle8i 8.1.7
- MS SQL Server 2000
- IBM DB2 8.1

また、システム構成によってはアプリケーションサーバ(J2EE 対応)用のライブラリも必要です。

2.インストール

ダウンロードした **Xone** の **zip** ファイルを適当なフォルダ(ディレクトリ)に解凍してください。これ以降、この解凍したフォルダを **XONE_HOME** として説明します。

3.データベースの初期化

RDB は必ず「日本語を使える」ように設定しておいてください。そうでないと、動作が不定になる場合があります。

最初に **Xone** で利用するデータベースとテーブルを **RDB** 上に作成します(*1)。 **XONE_HOME** の **sql** フォルダの中に各 **RDB** のフォルダがあります。その中に **create-db.sql** というファイルがあるので、これを対応する **RDB** 上で実行してください(*2)。これで、“**xone**”という名前でデータベースが作成されます。次に、このデータベースに対して **xone-schema.sql** を実行してください(*3)。これで **Xone** で利用するテーブルが作成されます。

*1:以前のバージョンを使っている方は、この「データベースの初期化」は必要ありません。ここは読み飛ばしてください。

*2:**RDB** によっては **create-db.sql** が存在しない場合もあります。その場合は手動で“**xone**”という名前のデータベースを作成してください。また、**create-db.sql** はデータベースを **drop** するので、必要なデータを入れた後では実行しないでください。

*3:**xone-schema.sql** ではエラーが出ることもありますが、内容によってはまったく問題ない場合もあります。また、**xone-schema.sql** はテーブルを **drop** するので、必要なデータを入れた後では実行しないでください。

4.初期化の情報を書き込む

最初に **XONE_HOME** の **bin** フォルダにある、**setup.bat** を実行してください(これは **Windows** の場合です。他の **OS** の場合は同名のシェルスクリプトを実行してください。これ以降もバッチファイルと記述しますが、適宜読みかえてください)。 **setup.bat** を実行すると、**XONE_HOME** の **config** フォルダのパスを **Preferences** を使って書き込みます。これにより **config** フォルダ内のファイルは、クラスパス、**/config/**ファイル名、**{Preferences の内容}/**ファイル名の順に検索されます。

次にデータベースに初期化のための情報を書き込みます。これには RDB に対応した JDBC ドライバ(*1)が必要です(ドライバは添付していないので、それぞれできるだけ新しいドライバを入手してください)。そのドライバをクラスパスに加えておいてください。

*1: DB2 の場合、現在のバージョンで対応しているのは、db2java.zip ファイルで提供されている COM.ibm.db2.jdbc.app.DB2Driver だけです。その他の com.ibm.db2.jcc.DB2Driver や COM.ibm.db2.jdbc.net.DB2Driver には対応していません。DB2 のドライバについては、<http://www-6.ibm.com/jp/software/data/developer/library/techdoc/java.html> を参照してください。

次に RDB への接続情報を設定します。XONE_HOME の config フォルダに torque.properties というファイルがありますが、それを環境に合わせて変更します。まず、

```
torque.database.xone.adapter=postgresql
```

という箇所を RDB 名に合わせてください。デフォルトでは postgresql になっていますが、その他の RDB の設定はコメントアウト(先頭に#記号が付いた行)されているので、それを利用してください(IBM DB2 の場合は db2app2 です)。

さらに、以下の箇所も変更してください。

(下記は PostgreSQL の例です)

```
# for postgresql
```

```
torque.dsfactory.xone.connection.driver = org.postgresql.Driver
```

```
torque.dsfactory.xone.connection.url = jdbc:postgresql://127.0.0.1/xone
```

```
torque.dsfactory.xone.connection.user = *****
```

```
torque.dsfactory.xone.connection.password = ***
```

変更したら XONE_HOME の bin フォルダにある、dbSetup.bat を実行してください。これで、エラーが出たときは、上記のファイルの間違いか、JDBC ドライバがクラスパスに加えられていないかだと思われます。再度確認してください。実行できたときは、次のようなダイアログが表示されます。



ここで、ユーザ名とパスワードをそれぞれ3文字以上で入力してください(以降、ここで指定した指定したユーザ名とパスワードでログインするので、忘れないようにしてください)。

OK ボタンをクリックすると、DB 上に root フォルダ、system フォルダ、および systemProperty オブジェクトなどを書き込みます。

5.システム構成と設定

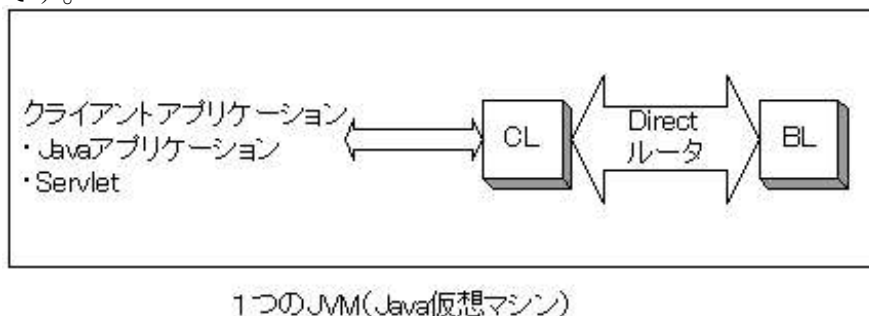
Xone は多層のシステム構成に対応しており、また構成の変更も容易に行うことができます。ここでは、ビジネスロジック層 (以下 BL) とクライアント層 (以下 CL) に分けて説明します。

BL はデータベースとのやりとりを行う層です。この層で必要なのは RDB との接続情報と JDBC ドライバです。接続情報については、「4.初期化の情報を書き込む」で記述した `torque.properties` の内容を必要に応じて変更してください。BL とデータベースサーバは物理的に別の PC にしてもよいですし、同じ PC でもかまいません。

CL はクライアントアプリケーションから使われる層です。クライアントアプリケーションは、Java のアプリケーションまたは Servlet (その Bean) です。BL と CL は、これらの層を接続する手段によって複数のシステム構成があります。この層を接続するモジュールをルータ (中継器) と名付けており、現在は直接接続 (Direct)、アプリケーションサーバ (SessionBean)、Web サービス (Soap) の3種類のルータがあります。

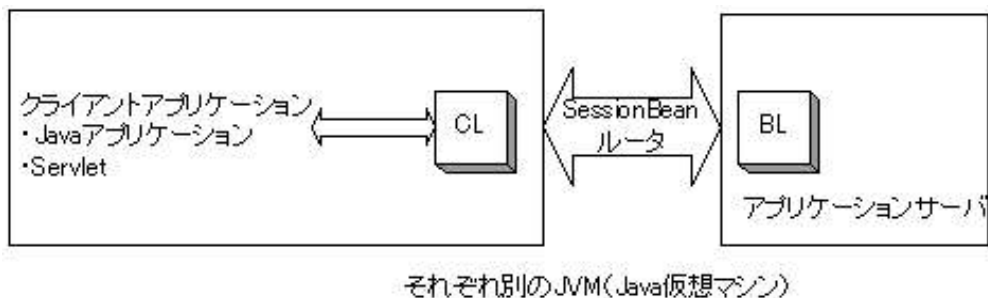
•Direct

Direct ルータは、BL と CL が同じ JVM (Java 仮想マシン) 上で動作するものです。単に直接呼び出しているだけです。



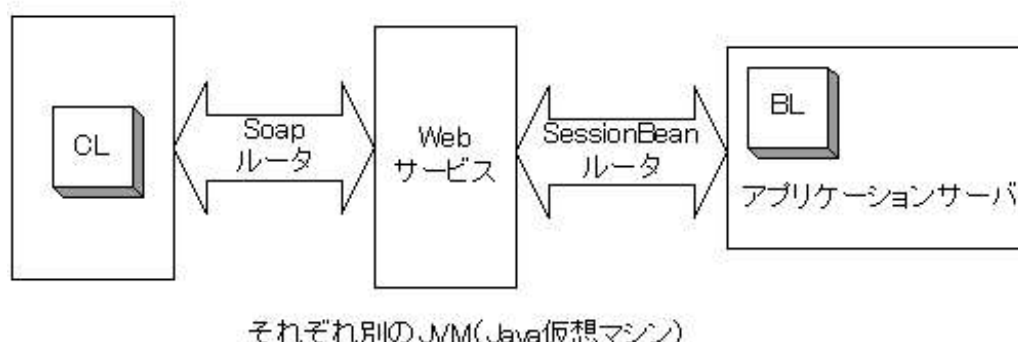
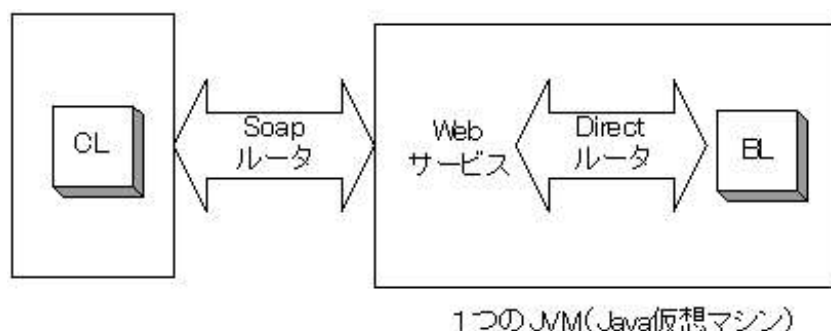
•アプリケーションサーバ (SessionBean)

J2EE1.3 以上に対応しているアプリケーションサーバの SessionBean (ステートレスセッションビーン) を利用するルータです。



•Web サービス (Soap)

Web サービス用のルータで、Xone では ApacheAxis1.1 を利用しています。このルータは他のルータと違い、数珠繋ぎにすることができますが、BL に接続することができません。BL への接続は、このルータからさらに Direct ルータあるいは SessionBean ルータに接続する必要があります。Soap ルータ自身はいくつでも接続することができますが、実用的には3段階ぐらいまででしょう。



※上記以外にもルータを追加すれば、CORBA、RMI、HORB などの分散オブジェクト環境で動作させることも可能です。必要であれば、info@fiverworks.com までメールしてください。また、パスワードは暗号化されますが、データは暗号化されません。Web サービス用のルータは LAN だけでなく、インターネット環境で利用できますが、安全のためには VPN の導入を検討するか、暗号化して送受信するルータが必要になります。これについても必要であれば上記アドレスにメールしてください。

5.1 構成別の設定

ルータの設定は、XONE_HOME の config フォルダにある xone.properties というファイルに記述します。xone.properties は、Shift-JIS で記述されています (日本語が使われているのはコメントの部分だけです)。Windows 以外のシステムの方はご注意ください。

•Direct ルータの場合

Direct ルータの場合、xone.properties の router という項目の値を「Direct」と記述します (デフォルトの設定)。この router の値は大文字と小文字を区別するので、正確に指定してください。CL が Java アプリケーション、Servlet のいずれの場合でも XONE_HOME の bin フォルダにある setup.bat を実行してください。

Direct ルータを使って、Java アプリケーションを作成するときは、各 RDB 用の JDBC ドライバもクラスパスに追加します。さらに XONE_HOME の lib フォルダにある Web サービス関連以外の次の jar ファイルをすべてクラスパスに追加します (*1)。

avalon-framework-4.1.4.jar
commons-beanutils-1.6.1.jar
commons-collections-2.1.jar
commons-configuration-1.0-dev-3.20030607.194155.jar
commons-dbcp-20030825.184428.jar

commons-lang-2.0.jar
commons-logging-1.0.3.jar
commons-pool-20030825.183949.jar
commons-validator.jar
csv.jar
highstandard.jar
jakarta-oro-2.0.7.jar
jcs-20030822.182132.jar
jdbc-2.0.jar
jdom.jar
jndi-1.2.1.jar
junit-3.8.1.jar
log4j-1.2.8.jar
logkit-1.0.1.jar
stratum-1.0-b3.jar
torque-3.1-xone1.0.jar (*2)
village-2.0-dev-20030825.jar
xercesImpl-2.0.2.jar
xmlParserAPIs-2.0.2.jar
xonePersonal1.0.5.jar

Servlet から利用する場合は、各 Servlet コンテナの流儀に従って、これらの jar ファイルを配備してください。

*1: クラスパスに追加するときは、XONE_LIB という環境変数に XONE_HOME の lib フォルダを設定します。その後、以下のテキストをコピー&ペースとして利用してください (Windows 用)。

```
%XONE_LIB%\avalon-framework-4.1.4.jar;%XONE_LIB%\commons-beanutils-1.6.1.jar;%  
XONE_LIB%\commons-collections-2.1.jar;%XONE_LIB%\commons-configuration-1.0-dev-  
3.20030607.194155.jar;%XONE_LIB%\commons-dbc-20030825.184428.jar;%XONE_LIB%\  
\commons-lang-2.0.jar;%XONE_LIB%\commons-logging-1.0.3.jar;%XONE_LIB%\commons-  
pool-20030825.183949.jar;%XONE_LIB%\commons-validator.jar;%XONE_LIB%\csv.jar;%  
XONE_LIB%\highstandard.jar;%XONE_LIB%\jakarta-oro-2.0.7.jar;%XONE_LIB%\jcs-  
20030822.182132.jar;%XONE_LIB%\jdbc-2.0.jar;%XONE_LIB%\jdom.jar;%XONE_LIB%\jndi-  
1.2.1.jar;%XONE_LIB%\junit-3.8.1.jar;%XONE_LIB%\log4j-1.2.8.jar;%XONE_LIB%\logkit-  
1.0.1.jar;%XONE_LIB%\stratum-1.0-b3.jar;%XONE_LIB%\torque-3.1-xone1.0.jar;%  
XONE_LIB%\village-2.0-dev-20030825.jar;%XONE_LIB%\xercesImpl-2.0.2.jar;%XONE_LIB%\  
\xmlParserAPIs-2.0.2.jar;%XONE_LIB%\xonePersonal1.0.5.jar
```

*2: torque-3.1-xone1.0.jar は、torque-3.1.jar に DB2 用のアダプタを追加したものです。

• SessionBean ルータの場合

SessionBean ルータの場合、J2EE1.3 以降に対応しているアプリケーションサーバが必要です。BL はアプリケーションサーバ上で実行されるので、これと通信するための設定が必要です。まず、xone.properties の router という項目の値を「SessionBean」と記述します。さらに、以下の項目をそれぞれのアプリケーションサーバに合わせた値に設定してください (次のこの例は JBoss3.2.3 を利用するときのものです)。

```
router.java.naming.factory.initial=org.jnp.interfaces.NamingContextFactory  
router.java.naming.provider.url=localhost:1099  
router.java.naming.factory.url.pkgs=org.jboss.naming:org.jnp.interfaces
```

さらに、次の項目を、「remote」または「local」のいずれかに設定してください。

`router.view-type=remote`

これは、**SessionBean** に接続するときリモートまたはローカルのどちらで接続するかを指定するものです。**Servlet** から利用する場合、アプリケーションサーバ上の **EJB** コンテナが同じ **JVM** 上で実行されるときは「**local**」にします。ネットワーク越し(たとえ同じホストでも)に利用するときは、「**remote**」を指定します。一般にローカルで利用できるときは、こちらの方がパフォーマンスがあがります。

CL で必要なのは **XONE_HOME** の **lib** フォルダ中の **Direct** ルータと同じ **jar** ファイルと各アプリケーションサーバに付属のクライアント用の **jar** ファイルです。これらをクラスパスに追加、あるいは各 **Servlet** コンテナの流儀に従って **jar** ファイルを配備してください(*3)。

*3: 正確に言えば必要のない **jar** ファイルもありますが、煩雑さを避けるために同じ設定にしています。

BL で必要なのは、**XONE_HOME** の **lib** フォルダ中の **Direct** ルータと同じ **jar** ファイルです。これらの **jar** ファイルを各アプリケーションサーバの流儀に従って配備してください。さらに、**XONE_HOME** の **ejb** フォルダに、各アプリケーションサーバのフォルダがあります。その中の **jar** ファイルをアプリケーションサーバに配備してください。該当するアプリケーションサーバが見つからないとき、あるいはうまく配備できないときは、**ejb/other** フォルダに **class** ファイルと **ejb-jar.xml** があるので、これから **jar** ファイルを作成してから配備してみてください。このとき **ejb** の **jndi** 名は、**remote** は

`ejb/com/fiverworks/xone/XoneSBRouter`

local は

`ejb/com/fiverworks/xone/XoneSBRouterLocal`

としてください。

最後に **XONE_HOME** の **bin** フォルダにある **setup.bat** を実行してください。CL と BL で物理的に異なるホストの場合は、各ホストでダウンロードした **Xone** の **zip** ファイルを適当なフォルダに解凍してから、その **setup.bat** を実行してください。また、各ホスト上の **xone.properties** を同じものにしてください。

• **Soap** ルータの場合

Soap ルータの場合、**Web** サービスの実行環境が必要です。**Xone** では CL に **ApacheAxis1.1** を利用しています。

まず、**xone.properties** の **router** という項目の値を「**Soap**」と記述します。さらに、以下の項目を **Web** サービスのサーバ側 (**Web** サービスプロバイダ) に合わせてホスト名とポートを設定してください。

`router.url=http://localhost:8080/axis/services/XoneWSRouter`

また、インターネットの接続に **proxy** サーバを使っている場合、以下の項目にそれぞれ適切な値を設定してください。

`router.http.proxyHost=xxx.xxx.xxx`

`router.http.proxyPort=xxxxx`

proxy サーバを使わないときは、これらの値を空欄にするか、コメントアウトしておいてください。

Soap ルータの場合、Direct ルータまたは SessionBean ルータにさらに接続する必要があります。
router1 という項目の値に「Direct」または「SessionBean」を指定してください。SessionBean ルータを指定した場合、前述したように

```
router1.java.naming.factory.initial=  
router1.java.naming.provider.url=  
router1.java.naming.factory.url.pkgs=
```

の各項目も設定が必要です。

なお、Soap ルータは数珠つなぎにすることもできます。router1 を「Soap」とし、router1.url を設定してください。次は router2、router3・・・routerN として、同様に設定を行っていきます。そして、最後の router を routerN+1 として「Direct」または「SessionBean」を設定します。このとき、各 Web サービスのサーバで xone.properties の内容を同じものしてください。

Web サービスのサーバ側では、XONE_HOME の bin フォルダにある deploy.bat で配備してください。なお、このバッチファイルは ApacheAxis1.1 用です。これをインストールしてからバッチファイルを実行してください。配備を取り消すときは、undeploy.bat を実行します。

CL で必要なのは XONE_HOME の lib フォルダにある Direct ルータと同じ jar ファイルに加えて、

```
axis.jar  
commons-discovery.jar  
jaxrpc.jar  
saaj.jar
```

が必要です。これらをクラスパスに追加、あるいは各 Servlet コンテナの流儀に従って jar ファイルを配備してください。Web サービスのサーバ側では、次に使うルータによって必要な jar ファイルをサーバ上に配備してください。

最後に XONE_HOME の bin フォルダにある setup.bat を実行してください。CL や BL で物理的に異なるホストの場合は、各ホストでダウンロードした Xone の zip ファイルを適当なフォルダに解凍してから、その setup.bat を実行してください。また、各ホスト上の xone.properties を同じものにしてください。

なお、Web サービスのサーバへのアクセスに Proxy サーバを使う場合は、router.http.proxyHost、router.http.proxyPort に値を設定しなくても、Java を起動するときに D オプションで以下のようにして起動することもできます。

```
-Dhttp.proxyHost=xxx.xxx.xxx -Dhttp.proxyPort=xxxx  
-Dhttp.nonProxyHosts="localhost"
```

また、D オプションではなくクライアントプログラム中で設定する場合は、

```
System.setProperty("http.proxyHost", "xxx.xxx.xxx");  
System.setProperty("http.proxyPort", xxxx);
```

を Xone を利用する前に実行しておいてください。

5.2 接続テスト

最後に接続テストを行います。クライアントのホストで、XONE_HOME の bin フォルダにある connTest.bat を実行してみてください。これで、ユーザ名とパスワードを入力し、モデルマネージャが起動するはずです。うまく起動できないときは、例外のメッセージをよく見て、クラスパス、各サーバへの jar ファイルの配備、setup.bat を実行したかどうかなどを確認してください。

なお、Soap ルータを使っており、さらに Proxy サーバを使っている場合は、このバッチファイルでは動きません。バッチファイル中の `java` コマンドで起動する箇所に、前述したような `D` オプションを付けてから実行してください。

5.3 ログ出力

Xone ではログ出力に Apache-jakarta の `commons-logging` を使っています。このロガーとして、Log4j を使いたいときは、XONE_HOME の `config` フォルダにある `log4j.xml` をクラスパスが通っているところに置いてください。

ソフトウェア情報

著作権

- 著作権について

本ソフトウェアの実行ファイル、ドキュメントに関する著作権を含む一切の権利は以下のとおり著作権者が有します。

有限会社 ファイバーワークス All rights reserved.

- 免責

本プログラムを使用した上で生じたいかなる損害についても、当社は責任を負いません。

サポート

- サポート

有限会社 ファイバーワークス

〒182-0011 東京都調布市深大寺北町 7-12-7

電話: 0424-98-1690

本プログラムに関するお問い合わせは E-mail にて受け付けています

Email: info@fiverworks.com

謝辞

本ソフトウェアの開発にあたり、メインの開発環境として NetBeansIDE3.5.1、Torque3.1 や Jakarta Project のいくつかのライブラリおよびツール、JDOM、csv.jar、highstandard.jar 等を使わせていただきました。その他、各種の RDB およびその JDBC ドライバ、JBoss、Eclipse、Ant、JUnit、XDoclet など使わせていただきました。末筆ではありますが、こうした素晴らしいソフトウェアを作られた多くの方々に感謝の意を表します。

『Java』は米国および、その他の国における米国 Sun Microsystems, Inc.の商標です。

Windows は Microsoft Corp.の商標です。

その他本文に記載されている会社名、商品名は、各社の登録商標または商標です。