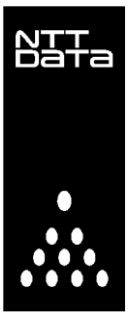


TERASOLUNA Server Framework for Java

環境設定手順書

第 2.0.6.2 版

NTT DATA

株式会社 NTTデータ

本ドキュメントを使用するにあたり、以下の規約に同意していただく必要があります。同意いただけない場合は、本ドキュメント及びその複製物の全てを直ちに消去又は破棄してください。

(1)本ドキュメントの著作権及びその他一切の権利は、NTT データあるいは NTT データに権利を許諾する第三者に帰属します。

(2)本ドキュメントの一部または全部を、自らが使用する目的において、複製、翻訳、翻案することができます。ただし本ページの規約全文、および NTT データの著作権表示を削除することはできません。

(3)本ドキュメントの一部または全部を、自らが使用する目的において改変したり、本ドキュメントを用いた二次的著作物を作成することができます。ただし、「参考文献:TERASOLUNA Server Framework for Java (Web 版) 機能説明書」あるいは同等の表現を、作成したドキュメント及びその複製物に記載するものとします。

(4)前2項によって作成したドキュメント及びその複製物を、無償の場合に限り、第三者へ提供することができます。

(5)NTT データの書面による承諾を得ることなく、本規約に定められる条件を超えて、本ドキュメント及びその複製物を使用したり、本規約上の権利の全部又は一部を第三者に譲渡したりすることはできません。

(6)NTT データは、本ドキュメントの内容の正確性、使用目的への適合性の保証、使用結果についての的確性や信頼性の保証、及び瑕疵担保義務も含め、直接、間接に被ったいかなる損害に対しても一切の責任を負いません。

(7)NTT データは、本ドキュメントが第三者の著作権、その他如何なる権利も侵害しないことを保証しません。また、著作権、その他の権利侵害を直接又は間接の原因としてなされる如何なる請求(第三者との間の紛争を理由になされる請求を含む。)に関しても、NTT データは一切の責任を負いません。

本ドキュメントで使用されている各社の会社名及びサービス名、商品名に関する登録
商標および商標は、以下の通りです。

・Apache、Tomcat は、Apache Software Foundation の登録商標または商標です。
・Java, JDK, J2SE, J2EE, JSP, Servlet は、米国 Sun Microsystems, Inc.の米国及びその他の国における登録商標または商標です。
・TERASOLUNA は、株式会社 NTT データの登録商標です。
・Windows は、米国 Microsoft Corp.の米国及びその他の国における登録商標または商標です。
・その他の会社名、製品名は、各社の登録商標または商標です。

本ドキュメントは、表紙の版数と同じバージョンの TERASOLUNA Server Framework for Java に対応しています。

変更履歴

バージョン	日付	改訂箇所	改訂内容
2.0.2.0	2009/10/30	新規作成	
2.0.2.1	2010/02/26	フッター	フッター情報変更
2.0.3.0	2010/04/01	全体	バージョン情報変更
2.0.3.1	2011/06/29	フッター	フッター情報変更
〃	〃	全体	バージョン情報変更
2.0.4.0	2010/04/01	全体	バージョン情報変更
〃	〃	フッター	フッター情報変更
2.0.5.0	2012/12/17	全体	バージョン情報変更
〃	〃	フッター	フッター情報変更
2.0.5.1	2014/03/31	全体	バージョン情報変更
〃	〃	フッター	フッター情報変更
2.0.5.2	2014/05/23	全体	バージョン情報変更
2.0.5.3	2014/08/20	全体	バージョン情報変更
2.0.6.1	2015/07/02	全体	バージョン情報変更
〃	〃	フッター	フッター情報変更
2.0.6.2	2016/08/31	全体	バージョン情報変更
〃	〃	フッター	フッター情報変更

第1章　概要

1.1 はじめに

本資料では、TERASOLUNA Server Framework for Java(以下、TERASOLUNA フレームワークと略す)を動作させるための必要な環境設定について解説する。サンプルアプリケーションとしてはTERASOLUNA で提供しているチュートリアルアプリケーションを用いる。
動作環境として想定している AP サーバは、オープンソースで最もよく利用されている Apache Tomcat（以下、Tomcat と略す）を利用する。開発環境である Eclipse 上で Tomcat を動作させ、デプロイからデバッグまで実施できる WTP[1]を利用した環境設定方法と WTP を利用できない環境での環境設定方法について解説していく。

1.2 本書の目的

本資料は TERASOLUNA フレームワークを利用して作成した Web アプリケーションの動作確認をする際、開発環境を構築するために AP サーバ等のプラットフォームおよび Web アプリケーションに必要な設定のほか、簡単なセットアップ手順を記述した「開発者向け」のお試しの設定方法を記述したものであるため、実運用に向けた環境構築については、AP サーバ等のプラットフォームを提供している各ベンダの関連ドキュメントを参照すること。

1.3 本書の範囲

本資料は、以下の目的で読むことを想定している。

- Tomcat 上で TERASOLUNA フレームワークを動かす際に必要な環境設定を理解する。
- 開発環境のセットアップの手順を理解する。
- TERASOLUNA の動作検証をする際に、機能網羅サンプルアプリケーション、およびチュートリアルアプリケーションの動作に必要な環境設定を理解する。

[1] WTP とは Web Tools Platform といい、サーバサイドで開発を行うための Eclipse プラグインのセットである。サーバ管理プラグインや JavaEE アプリケーションを作成するためのプラグインなど、一通りの機能がそろっている。

第2章　環境設定

2.1 想定環境

本資料では以下の環境を例にして解説していく。他の環境で設定する際は本書をベースに適宜読み替えて設定していくこと。

- OS：Microsoft Windows XP Professional SP3
- JDK：Java SE 6.0
- データベース：H2 DB Engine
- Web アプリケーションサーバ：Apache Tomcat 6.0.xx
- 統合開発環境：Eclipse SDK 3.4.1
- Eclipse plugin：WTP 3.0.4

なお、DB 接続設定は以下の通りとする。

- JNDI 名：TerasolunaSampleDataSource
- データソース URL：jdbc:h2:tcp://localhost/~/terasoluna
- JDBC ドライバクラス：org.h2.Driver
- ユーザ名：sa
- パスワード：(設定なし)

※TERASOLUNA で動作検証済みでない JDK,AP サーバ,DB サーバに関しては各自の責任で動作確認をして下さい。

2.2 統合開発環境 Eclipse と JDK の設定

2.1 節で用意した Eclipse と JDK をインストールする。それぞれ以下のディレクトリにインストールすると想定して記述している。他のディレクトリにインストールした場合は適宜読み替え設定していくこと。

- Java SE 6.0：C:¥Program Files¥java¥jdk1.6.0_x x (x はバージョン番号)
- Eclipse SDK 3.4.1：C:¥Eclipse
- Workspace ディレクトリ：C:¥Eclipse¥workspace

1. Java のホームディレクトリの設定
   各端末の環境変数を以下のように設定する。環境変数を設定するには「マイ コンピュータ」を右クリックし、「プロパティ(R)」を選択する。システムのプロパティ画面の「詳細設定」タブを選択し、「環境変数」をクリックする。ユーザ環境変数の「新規(N)」から変数、値を以下のように設定する。

変数：値
JAVA_HOME ： C:¥Program Files¥Java¥jdk1.6.0_xx (x はバージョン番号)
path ： %JAVA_HOME%¥bin;

2. JDK の設定
   Eclipse 上で利用できる JDK を設定する
   Eclipse のメニューバーにある「ウィンド(W)」-「設定(P)」を選択する。「java」-「インストール済みのJR

E」で、「jdk1.6.0_xx (x はバージョン番号)」にチェックがついていることを確認する。続けて「java」-「コンパイラー」で、「JDK 準拠」のコンパイラ準拠レベルが“1.5”になるように設定する。

3. プロジェクトのインポート
チュートリアルアプリケーションのプロジェクトである“terasoluna-server4jweb-tutorial_2.0.x.x.zip”ファイルを“C:¥Eclipse¥workspace”直下に展開し、“C:¥Eclipse¥workspace¥tutorial-thin”をを Eclipse 上に設定し、編集、実行が行えるようにする。
※Rich 版の場合は”terasoluna-server4jrich-tutorial_2.0.x.x.zip”、”tutorial-rich”となるので以降は適宜読み替えること。

手順は以下の通りである。

① Eclipse を起動する。
② 「ファイル(F)」-「インポート(I)」を選択する。
③ 選択画面では「既存のプロジェクトをワークスペースへ」を選択して、次へを押下する。
④ 「プロジェクトのインポート」画面ではルートディレクトリの選択欄に“C:¥Eclipse¥workspace¥tutorial-thin”を指定し、終了を押下する。
⑤ 「パッケージ・エクスプローラ」よりインポートしたプロジェクトを開き、Web アプリケーション・ライブラリー[{0}]内に jar ファイルが存在しているかを確認する。存在しない場合は、プロジェクトを右クリックして、「プロジェクトを閉じる」を選択し、再度プロジェクトを選択し「プロジェクトを開く」を選択する。それでも表示されない場合は、Eclipse を再起動する。
⑥ Eclipse の「プロジェクト(P)」-「クリーン(N)」を選択し、すべてのプロジェクトをクリーンにチェックを入れて「OK」を押下する。

2.3 DB サーバの設定

サンプルアプリケーションで利用する DB の設定を行う。2.1 節で用意した DB サーバである H2 DB Engine の初期 DB データを設定する。手順は以下の通りである。

1. DB サーバの起動
“C:¥Eclipse¥workspace¥tutorial-thin¥h2db”配下にある“h2db_start.bat”を実行する。H2Database Console が起動しない場合は、“h2db_console.bat”を実行する。

2. USER_LIST テーブルの作成
続けて、“C:¥Eclipse¥workspace¥tutorial-thin¥h2db”配下にある“h2db_init.bat”を実行する。(script が実行され初期データが設定される)

3. データベースへの接続
H2Database Console のログイン画面で、以下を入力する。
- 保存済設定 ： Generic H2 (Server)
- ドライバクラス ： org.h2.Driver
- JDBC URL ： jdbc:h2:tcp://localhost/~/terasoluna
- User ： sa
- Password ： (設定なし)

4. データベースの確認
データベースの接続後、画面左のテーブル一覧に以下が存在することを確認する。
テーブル名 ： USERLIST
カラム ： ID、NAME、AGE、BIRTH

2.4 AP サーバの設定

チュートリアルアプリケーションで利用する AP サーバの設定を行う。まずは Tomcat を http://tomcat.apache.org/index.html からダウンロードして、各自の端末にインストールする。そして、Eclipse 上から Tomcat を起動したり、プロジェクトのデプロイ、デバッグ等を行いたい場合は Eclipse WTP プラグインを利用して Tomcat の設定を行う。WTP 環境が利用できない場合、Antを利用して直接 Tomcat にデプロイする方法を説明していく。

1. Tomcat のダウンロード/インストール

ダウンロードサイトから「apache-tomcat-6.0.xx.zip」をダウンロードし、解凍ソフトを利用して各自端末に展開する。以下、C 直下に展開することを想定して進めていく

2. Eclipse 上での Tomcat の設定方法

開発者が簡単にデプロイやデバッグ等を行える環境を構築するために、WTP プラグインを利用して Eclipse 上で Tomcat の環境を設定していく。

1. サーバーの設定
   ① Eclipse の「ファイル(F)」-「新規(O)」-「その他(N)」メニューを選択する。
   ② ウィザードの選択から「サーバー」-「サーバー」を選択し次へを押下する。新規サーバーから「Apache」-「Tomcat v6.0 サーバー」を選択して終了する。

2. プロジェクトをサーバーへ追加
   ① Eclipse のサーバービューから上記の「Eclipse の設定」で追加したサーバーを選択して右クリックする。
   ② 右クリックメニューの中から「プロジェクトの追加と除去」を選択する。使用可能プロジェクトの中に”tutorial-thin”があるので、構成プロジェクトに追加する。

3. データソースの設定
   WTP の Tomcat は設定情報を Eclipse 内部でコピーして管理している。よって、管理ツールで設定したデータソースの設定を WTP へ反映させる必要がある。反映させる方法には以下の2種類がある。

   - WTP で管理している Tomcat の“server.xml”ファイルを手動で変更する方法
   - プロジェクトの META-INF 配下に“context.xml”ファイルを配置する方法

   どちらを適用するかは以下の手順を参考にして各自で判断してもらいたい。
   META-INF 配下に“context.xml”ファイルが配置してある場合は、“server.xml”ファイルに設定をしても“context.xml”ファイルの設定を優先するので注意が必要である。

   【server.xml を手動で変更する方法】
   Eclipse のパッケージエクスプローラー上で、「サーバ」-「Tomcat v6.0 サーバー @ localhostconfig」-「server.xml」を選択する。右画面に server.xml の内容が表示されたら、適宜自分の環境に合わせて修正する。(本手順書の想定環境では以下の網掛け部分のようになる。)

```
<?xml version="1.0" encoding="UTF-8"?>
<Server port="8005" shutdown="SHUTDOWN">
    <Listener SSLEngine="on" className="org.apache.catalina.core.AprLifecycleListener"/>
```

```
    <Listener className="org.apache.catalina.core.JasperListener"/>
    <Listener className="org.apache.catalina.mbeans.ServerLifecycleListener"/>
    <Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener"/>
    <GlobalNamingResources>
      <Resource auth="Container"
          description="User database that can be updated and saved"
          factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
          name="UserDatabase" pathname="conf/tomcat-users.xml"
          type="org.apache.catalina.UserDatabase"/>
    </GlobalNamingResources>
    <Service name="Catalina">
        <Connector connectionTimeout="20000" port="8080"
                                    protocol="HTTP/1.1" redirectPort="8443"/>
        <Connector port="8009" protocol="AJP/1.3" redirectPort="8443"/>
        <Engine defaultHost="localhost" name="Catalina">
            <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
                resourceName="UserDatabase"/>
            <Host appBase="webapps" autoDeploy="true"
                name="localhost" unpackWARs="true"
                xmlNamespaceAware="false"
                xmlValidation="false">
              <Context docBase="tutorial-thin"
                  path="/tutorial-thin"
                  reloadable="true"
                  source="org.eclipse.jst.j2ee.server:tutorial-thin">
                 <Resource name="TerasolunaDataSource"
                      type="javax.sql.DataSource"
                      password=""
                      driverClassName="org.h2.Driver"
                      maxIdle="2"
                      maxWait="5000"
                      username="sa"
                      url="jdbc:h2:tcp://localhost/~/terasoluna"
                      maxActive="4"/>
              </Context>
            </Host>
        </Engine>
    </Service>
</Server>
```

※JNDI データソースを設定したプロジェクトを WTP の Tomcat プロジェクトから除去した場合、同じプロジェクトを Tomcat へ追加した場合に再度この手順が必要となる。

【META-INF 配下に“context.xml”ファイルを配置する方法】

Eclipse にて“プロジェクト名/webapps/META-INF/context.xml”ファイルを、適宜各自の環境に合わせて修正する。（本手順書の想定環境では以下の網掛け部分のようになる。）

```
<?xml version="1.0" encoding="UTF-8"?>
    <Context>
    <Resource
        name="TerasolunaDataSource"
        type="javax.sql.DataSource"
        password=""
        driverClassName="org.h2.Driver"
        maxIdle="2"
        maxWait="5000"
        username="sa"
        url="jdbc:h2:tcp://localhost/~/terasoluna"
        maxActive="4"/>
</Context>
```

※WTP 環境以外の Tomcat へデプロイする場合は、JNDI の設定の注意点があるので、後述する「Tomcat を利用する際の JNDI 設定の注意点」を参照のこと。

4. データベースの接続設定
   ① H2DB のドライバ“h2.jar を”C:¥Eclipse¥workspace¥tutorial-thin¥webapps¥WEB-INF¥lib”配下からコピーして、”C:¥Program Files¥Apache Software Foundation¥Tomcat 6.0￥lib”配下にコピーする。
   ② “applicationContext.xml”ファイルの TerasolunaDataSource の Bean 定義については、以下のようにすること。

```
<bean id="TerasolunaDataSource"
    class="org.springframework.jndi.JndiObjectFactoryBean">
    <property name="jndiName" value="java:comp/env/TerasolunaDataSource" />
</bean>
```

5. Tomcat の起動
   「サーバー」を右クリックして、「開始(S)」をクリックし Tomcat の起動を行う。

   起動に失敗する場合は、検証環境、もしくは前節のアプリケーションの設定に不備がないかを確認すること。主に次の原因が考えられる。
   - DB が起動していない、もしくは接続できない URL である
   - DB には接続できるが、DB のアカウント、DB 名などのデータソース設定が不正である
   - JNDI 名を変更すべき Bean 定義ファイルに漏れがある
   - JDBC の jar ファイルを AP サーバのライブラリフォルダに格納していない
   - WAR ファイルの生成方法が不適切である

3. Ant を利用した Tomcat の設定方法

WTP プラグインを利用できず、Eclipse 上で Tomcat の環境設定が出来ない場合は直接 Tomcat にデプロイし、動作確認を行う。チュートリアルプロジェクト内には Ant の設定ファイルが用意されているので、Ant ビルドを実行し、WAR ファイルを生成し、デプロイを行う。注意点として Ant を実行する前に前節の 4.データベースの接続設定を済ませておくこと。

1. Ant 設定ファイルの修正

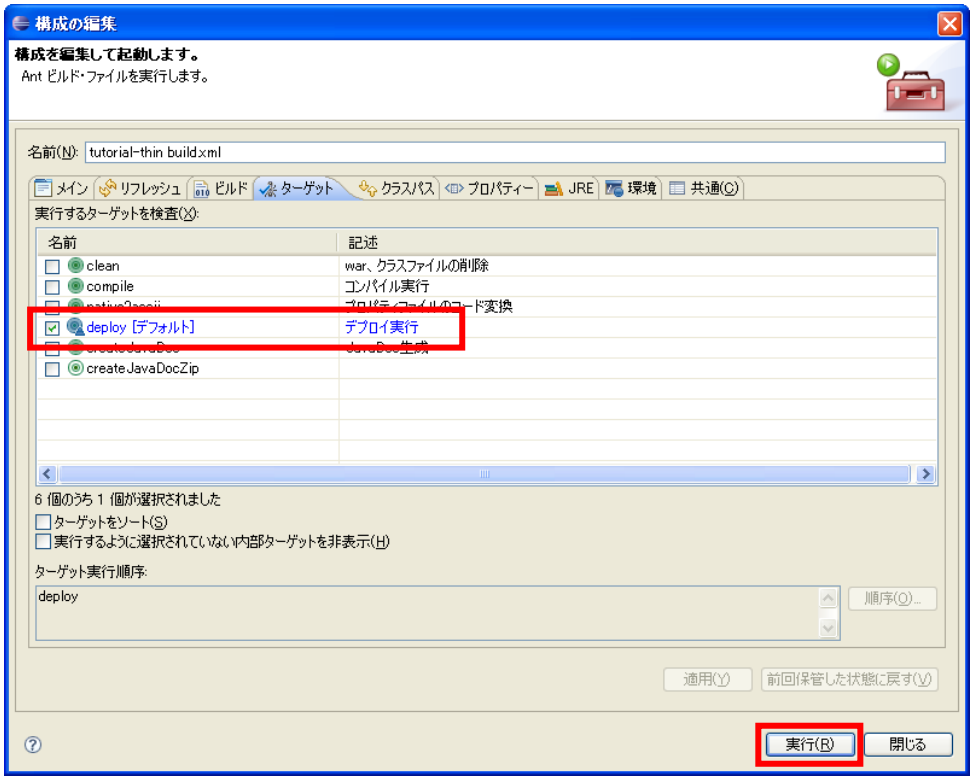
tutorial-thin 直下の ant フォルダにある「build.properties」を修正する。
上記に示した想定環境では、以下のように設定項目を変更する。

```
# Web アプリケーションサーバのホームディレクトリ
webapsvr.home=C:/apache-tomcat-6.0. xx

# Web アプリケーションサーバの lib ディレクトリ
webapsvr.lib.dir=C:/apache-tomcat-6.0. xx /lib

#  デプロイ先ディレクトリ
deploy.dir=C:/apache-tomcat-6.0. xx /webapps

# JDBC Jar のパス
jdbc.driver=C:/apache-tomcat-6.0. xx /lib/h2.jar
```

2. Ant の実行

コマンドプロンプトから Ant を実行する場合、各自端末でコマンドプロンプトを開き、ant フォルダ（C:¥terasoluna¥workspace¥tutorial-thin¥ant）まで移動する。コマンドに「ant」と入力して実行する。

Eclipse 上で Ant を実行する場合、パッケージエクスプローラー上の ant フォルダを開き、「build.xml」を右クリックして「実行」-2 つ目の「Ant ビルド」を選択する。下図のように構成の編集画面が開き、「deploy」にチェックを入れて「実行」をクリックする。



成功すれば C:¥apache-tomcat-6.0.xx¥webapps 直下に tutorial-thin.war が生成される。

Windows 環境上で Eclipse 3.4.1 の場合、ワークスペースのエンコーディング設定が Ant のコンソールログ出力の際に適用されるため、ログが出なかったり途中で止まったりすることが起きます。
その場合以下の2つの方法で回避してください。

- 「build.xml」を右クリックし「実行」-2 つ目の「Ant ビルド」で「構成の編集」画面の開く
  - コンソール・エンコードの設定を、その他 MS932 に変更して「適用」-「実行」をクリックする
- 「build.xml」の XML ヘッダ部を、<?xml version="1.0" encoding="Windows-31J" ?>に修正する

3. Tomcat の起動

JDK の設定と同様にして、「CATALINA_HOME」の環境変数を以下のように設定する。

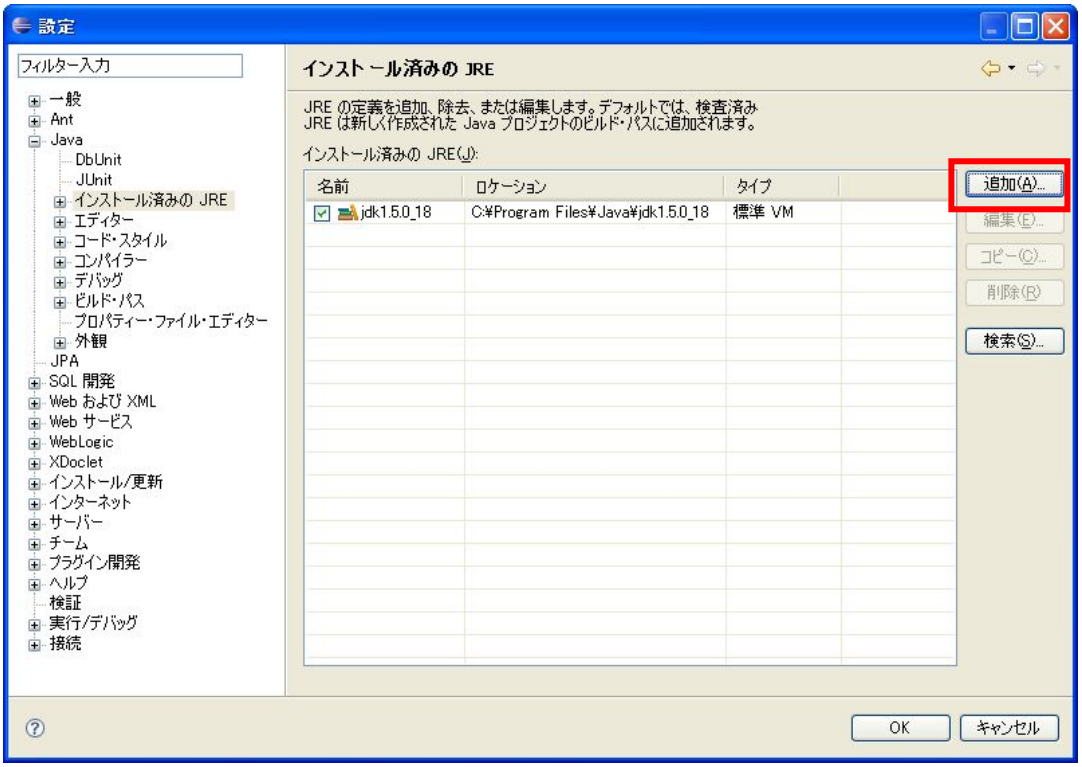
変数:値
CATALINA_HOME ： C:¥apache-tomcat-6.0.xx (x はバージョン番号)
path ： % CATALINA_HOME%¥bin;

Tomcat は、${Catalina.home}/webapps フォルダに WAR ファイルを格納し Tomcat を起動することで、自動的にデプロイされる。Tomcat を起動するにはコマンドプロンプトで C:¥apache-tomcat-6.0.xx¥bin に移動し、startup.bat を実行する。

第3章　APPENDIX

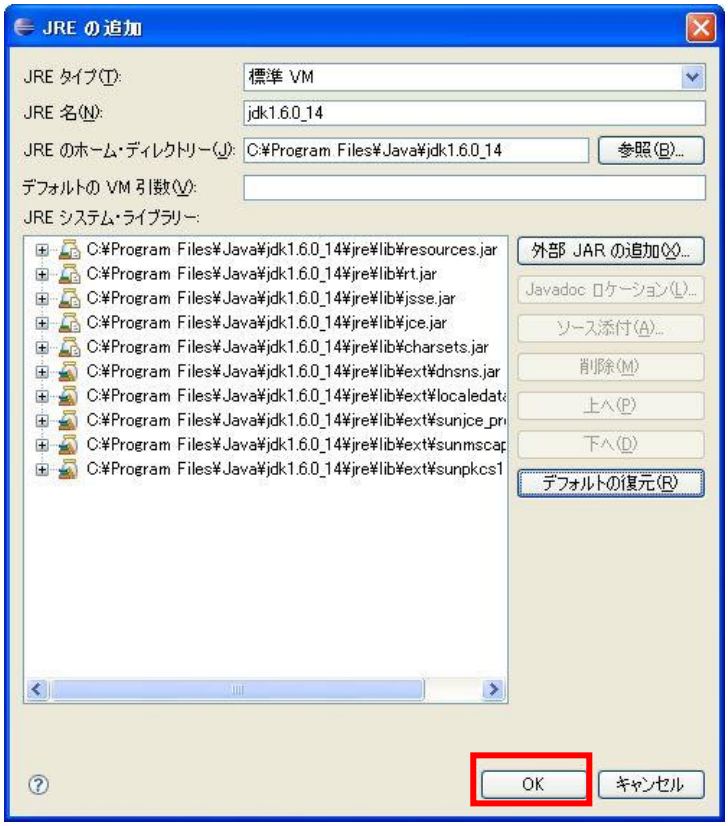
3.1 JDK のバージョンを変更する手順

1. 「インストール済み JRE」の設定の変更
   ① Eclipse のメニューより「ウィンドウ」→「設定」を選択する。
   ② 設定ウィンドウより「Java」→「インストール済み JRE」を選択し、「追加」ボタンを押下する。

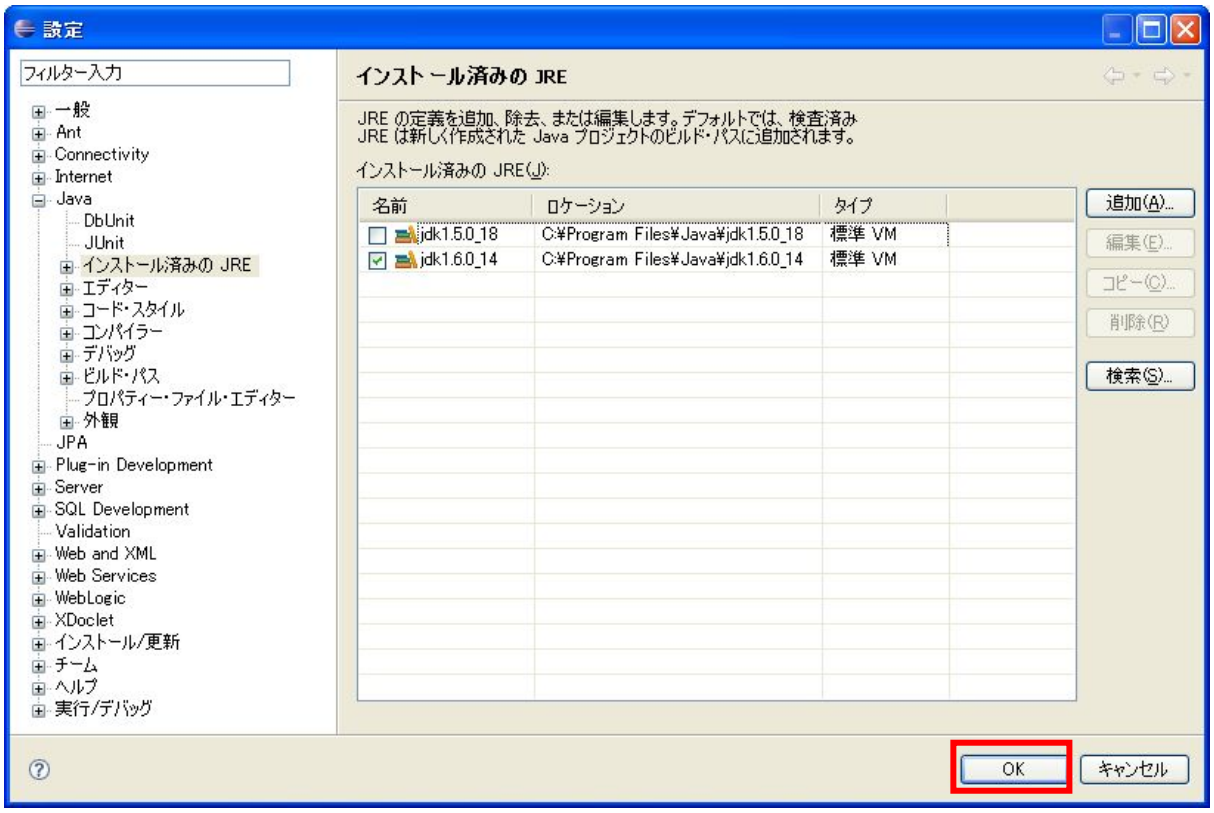


③ 表示された「JRE を追加」の設定ウィンドウより以下の内容を入力し、「OK」ボタンを押する。

- JRE タイプ：標準 VM
- JRE 名：jdk1.6.0_xx　<任意の名前でよいがバージョンを表す文字列とした方がよい>
- JRE のホーム・ディレクトリー：<jdk1.6.0_xx のインストールディレクトリ>
- デフォルト VM 引数：(設定なし)



④　新しく追加した JRE 名にチェックを入れ、「OK」ボタンを押下する。

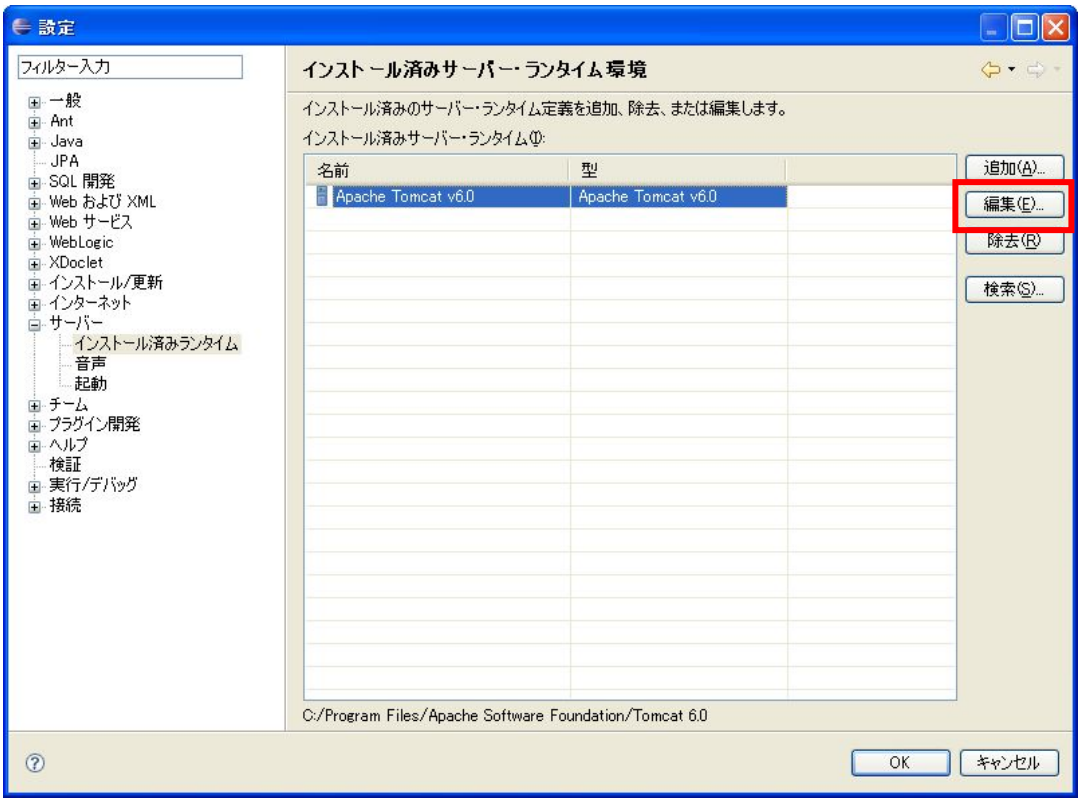


2.　「インストール済みサーバ・ランタイム」の JRE の設定の変更

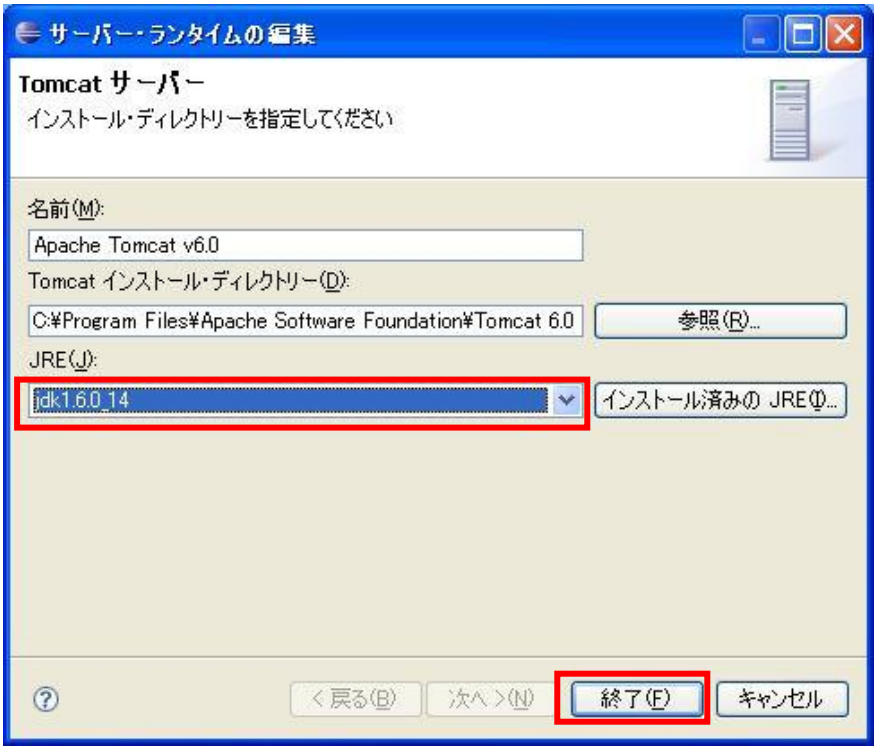
WTP として動作している Eclipse 上の Web アプリケーションサーバのランタイム設定を変更する。

①　Eclipse のメニューより「ウィンドウ」→「設定」を選択する。

②　設定ウィンドウより「サーバ」→「インストール済みランタイム」を選択し、既に設定されているサーバを選択し、「編集」ボタンを押下する。



③　表示された「サーバー・ランタイムの編集」の JRE のセレクトボックスを新しく追加した JRE 名に変更し、「終了」ボタンを押下する。以下は Tomcat の設定例である。



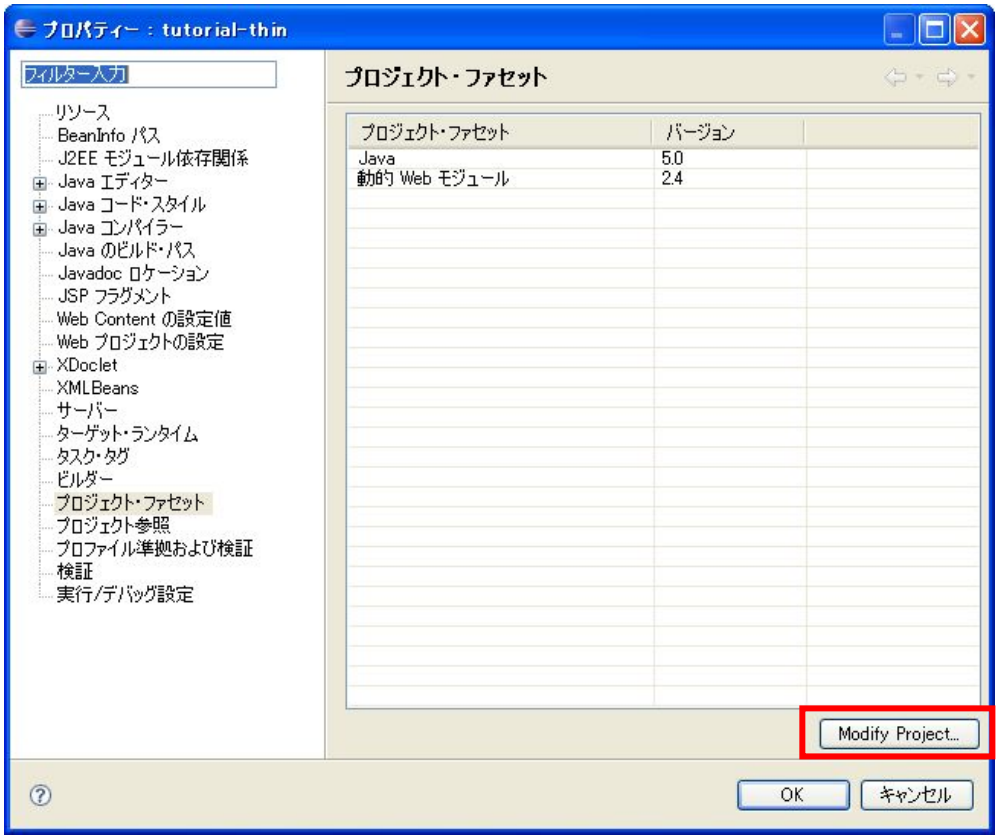
④　「OK」ボタンを押下し、設定ウィンドウを閉じる。

3. 「プロジェクト・ファセット」のランタイムの JRE の設定の変更

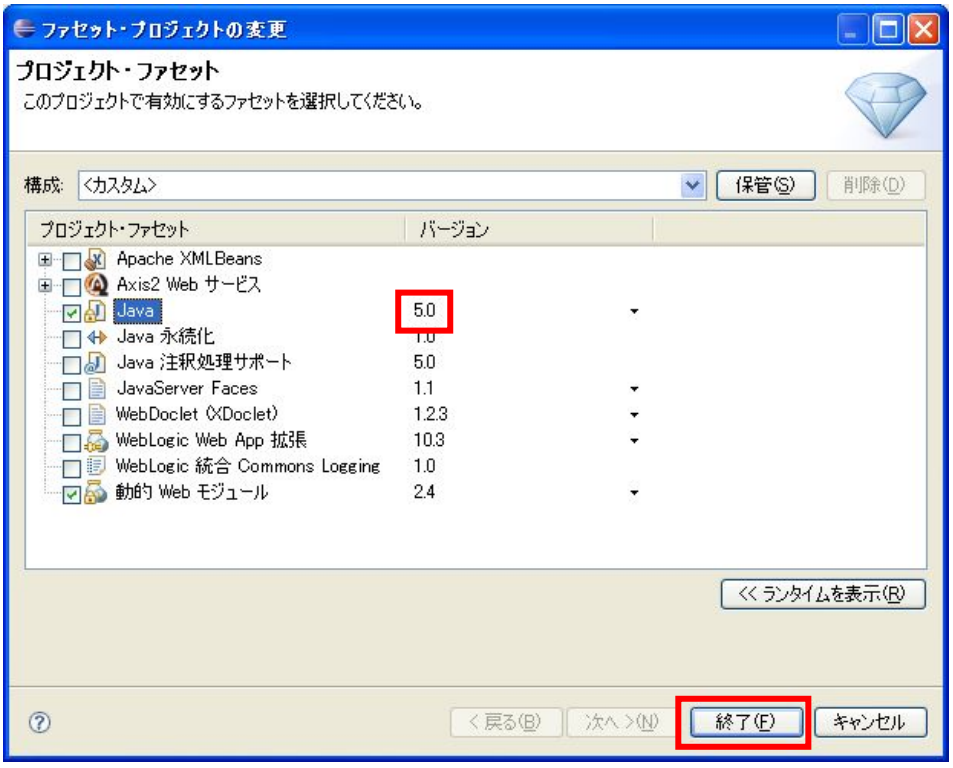
WTP プロジェクトは Eclipse 上では動的 Web プロジェクトとして動いているので、そこの設定を変更する。

① 「パッケージ・エクスプローラ」より該当のプロジェクトを選択し、右クリックメニューより「プロパティ」を選択する。

② 「プロジェクト・ファセット」を選択し、「Modify Projects...」ボタンを押下する。

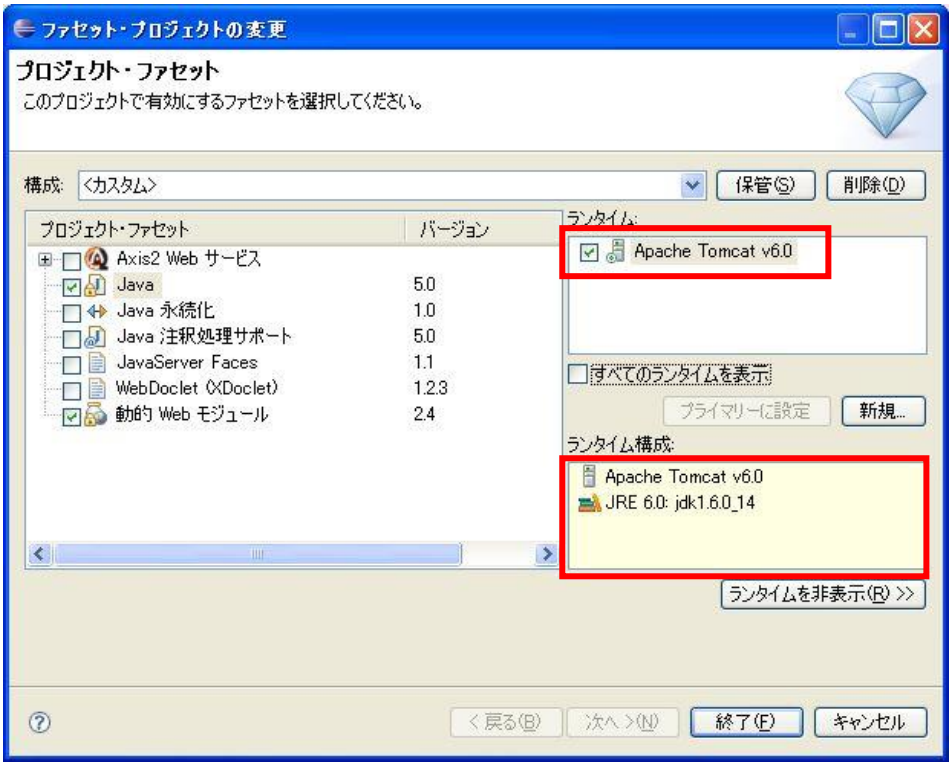


③ 表示された「Modify Projects...」の設定ウィンドウより「Java」のバージョンが「5.0」になっていることを確認し、「<<ランタイムを表示(R)」ボタンを押下する。



④ 「ランタイム」の設定内容が表示され、サーバ名を選択し、「ランタイム構成」に新しく追加した

JRE 名が表示されることを確認する。

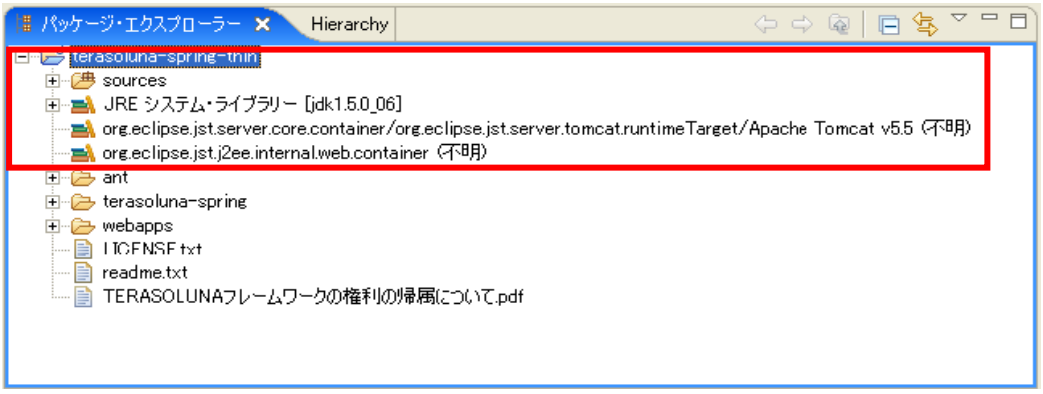


⑤　確認したら「終了」ボタンを押下し、「OK」ボタンを押下し、プロパティウィンドウを閉じる。

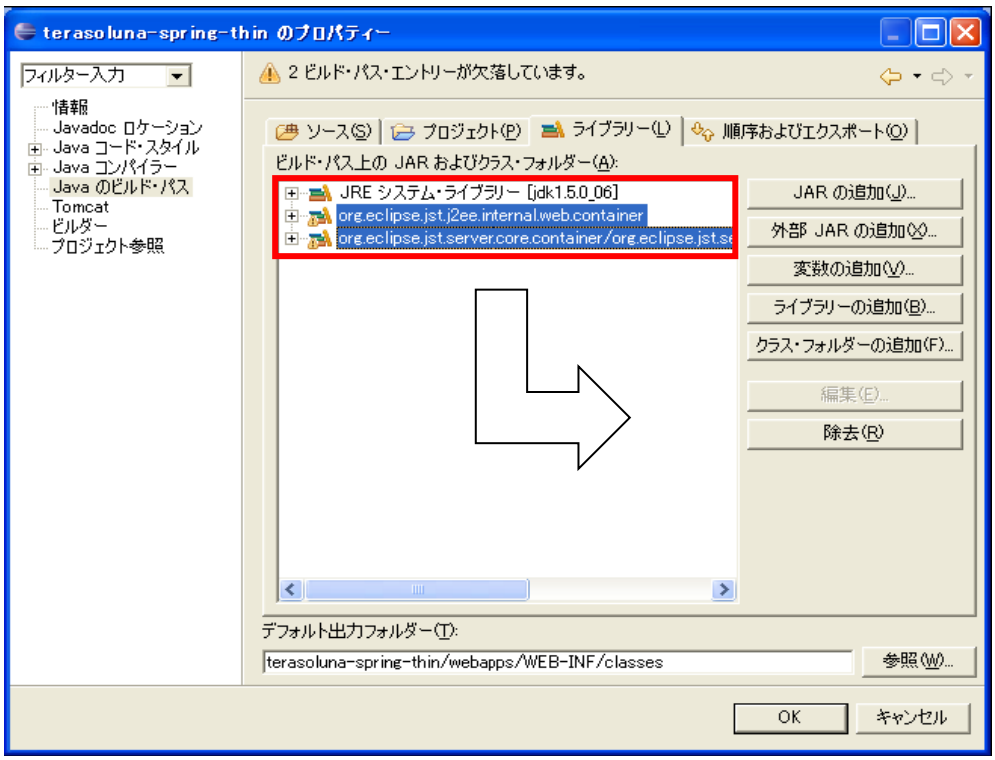
3.2 WTP 環境から非 WTP 環境への切り替え手順

1. インポート手順

① WTP を用いたプロジェクトを非 WTP 環境へインポートする。すると、ビルドパスの表示部分に（不明）のエラーが表示される。非 WTP 環境のため WTP の設定情報が判別できないため出力されるエラーである。



② （不明）を削除する。プロジェクトを右クリックして「プロパティー」から「Java のビルドパス」のページを開き、「ライブラリー」タブを選択し、エントリを除去する。



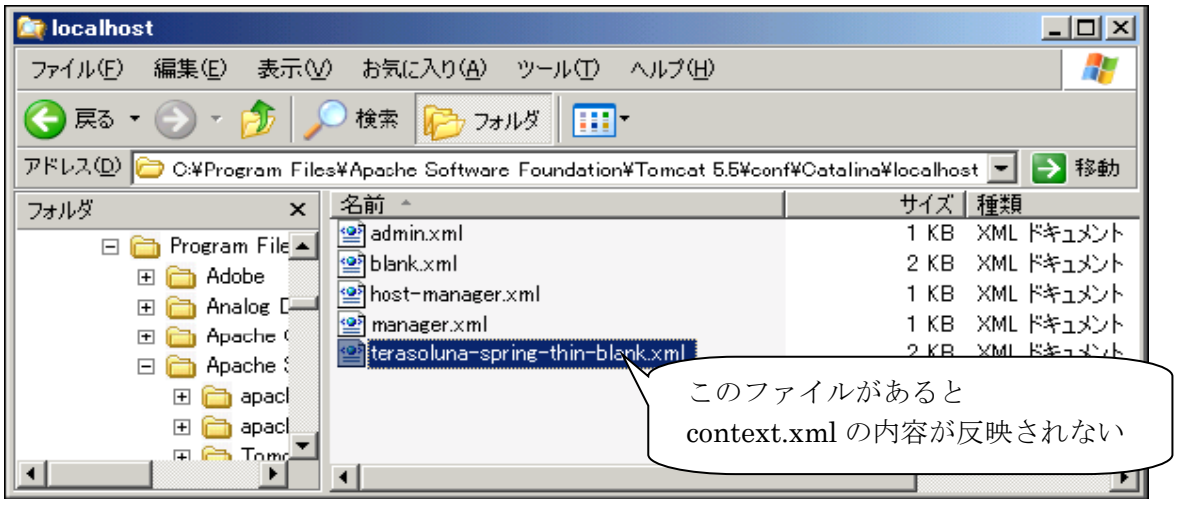
③ 続けて必要なライブラリをビルド・パスに追加する。プロジェクトの WEB-INF/lib に配置した JAR ファイルおよび各 AP サーバで参照する JAR ファイル(Tomcat の場合だと”TOMCAT_HOME/common/lib”にある JAR ファイル)を追加する。

2. デプロイ

/antフォルダにあるbuild.propertiesを自分の環境にあうように設定する。その後でbuild.xmlのdeployタスクを実行する。

3. Tomcat を利用する際の JNDI 設定の注意点

Eclipse プロジェクトの/webapps/META-INF/context.xml ファイルに JNDI の設定を記述してある場合は、TOMCAT_HOME/conf/Catalina/localhost/に<context.name>.xml ファイルが存在しないことを確認する。ある場合は、削除すること。なお、デプロイする度に削除する必要は無く context.xml ファイルを修正した場合のみ必要な作業である。



さらに、TOMCAT_HOME/conf/server.xml ファイルに<context>・・・</context>タグが存在する場合は、削除すること。なお、この作業は一度すればよい。

```
:
略
:
<Host>
      <Context path="/tutorial-thin">・・・</Context>
</Host>
:
略
:
```