

## 目次

1. Eclipse WTP.....	2
1.1. EclipseWTP の目的 .....	2
1.2. EclipseWTPプロジェクトの構成.....	2
2. WTP コンポーネント.....	4
2.1. 環境設定 .....	4
2.2. データツール.....	4
2.3. サーバツール.....	9
2.4. 構造化エディタ(SSE).....	16
2.5. XML・Webサービスツール .....	16
3. 参考資料.....	17
4. 最後に .....	18

## 1. Eclipse WTP

EclipseWTP(Web Tools Platform)は、Eclipse プロジェクトが進めている J2EE アプリケーション開発の共通インフラの提供し、Web アプリ開発のスタンダードを目指すプロジェクトです。

IDE としてデファクトスタンダードの地位を確立した Eclipse は、テストツールへ共通インフラを提供する Hyades など、商用も含めたツールの共通プラットフォームとしての動きが加速しているようです。

プロジェクトマイルストーンは以下の様になっており、急ピッチで開発が進んでいます。

- 2004 年 7 月 初期バージョンを IBM、ObjectWeb(Lomboz)が提供
- 2004 年 10 月 M1 リリース
- 2004 年 12 月 M2 リリース
- 2005 年中 Eclipse3.1 をベースとした V1.0 リリース予定

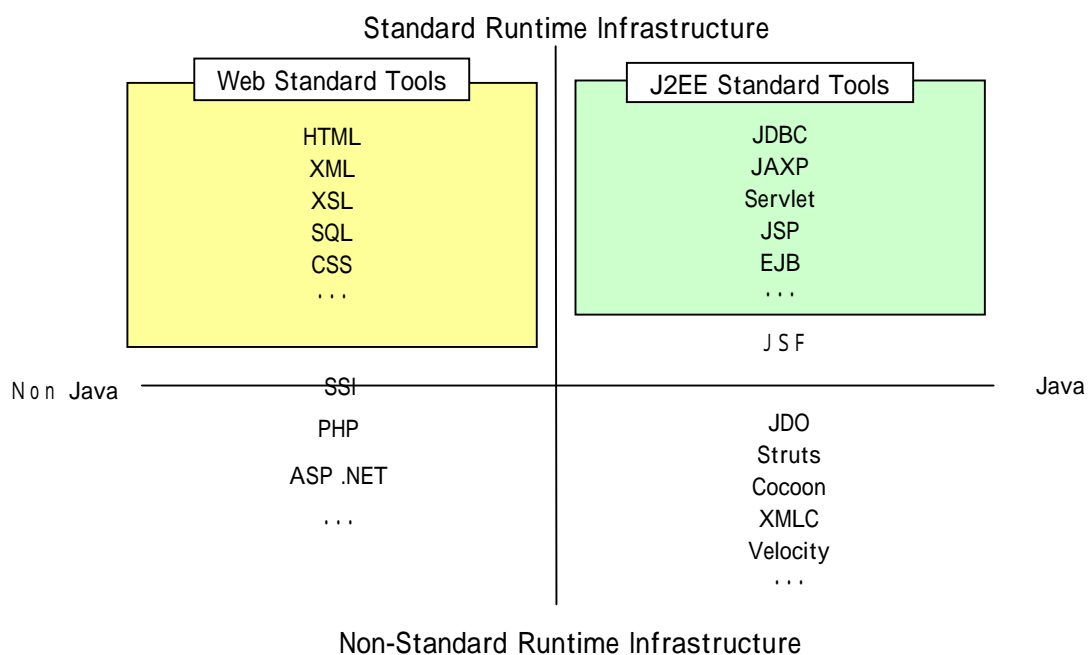
### 1.1. EclipseWTP の目的

『ユーザーインターフェイス、ツール API セットの統一を行い、ベース・テクノロジーのための統一なプラットフォームを提供する』を目的としている EclipseWTP ですが、開発者の視点からどのようなメリットがあるのでしょうか。

現在、多くの開発者が Eclipse 上で Web アプリケーションの開発を行っていますが、環境によってプラグインを使い分け、その都度使い方を覚えなければならないという状況にあります。EclipseWTP を使用すると、J2EE 開発に必要なセットが入手でき、プラグインやツールを統一されたインターフェイスで利用することが出来ます。

### 1.2. EclipseWTP プロジェクトの構成

EclipseWTP プロジェクトは『Web Standard Tools』と『J2EE Standard Tools』という2つのサブプロジェクトから構成されています。



**(1) Web Standard Tools**

Web 用プロジェクト

ビルダー、バリデータ

ソース・エディターフレームワーク

XML,HTML, CSS ソース・エディター

データベース開発

**(2) J2EE Standard Tools**

J2EE のための生成物の開発サポート

EJB,JSP,Servlet,JCA,JDBC,JMS

Web Services

アプリケーション・サーバーのサポート

起動、デバッグ

Web サービス開発サポート

WSDL エディタなど

現段階では全てのコンポーネントがそろっているわけではありません。

次章以降では EclipseWTP に含まれる各コンポーネントの使用方法について説明します。

## 2. WTP コンポーネント

### 2.1. 環境設定

Eclipse3.01 に以下の各プラグインのインストールが必要ですが、EclipseWTP 以外はメニューの[ヘルプ]-[ソフトウェア更新]-[検索とインストール]からもインストールできます。言語パックは日本語化しない場合は不要です。

- Eclipse 3.\*(eclipse-SDK-3.0.1-win32.zip)  
<http://download.eclipse.org/downloads/drops/R-3.0.1-200409161125/index.php>
- 言語パック(NLpack-eclipse-SDK-3.0.x-win32.zip)  
[http://download.eclipse.org/downloads/drops/L-3.0.1\\_Translations-200409161125/index.php](http://download.eclipse.org/downloads/drops/L-3.0.1_Translations-200409161125/index.php)
- EclipseWTP M2 (wtp-1.0M2.zip)  
<http://download.eclipse.org/webtools/downloads/>
- EMF 2.\* SDK, XSD 2.\* SDK (emf-sdo-xsd-SDK-2.0.1.zip)  
<http://download.eclipse.org/tools/emf/scripts/downloads.php>
- GEF 3.\* SDK(GEF-SDK-3.0.1.zip)  
<http://download.eclipse.org/tools/gef/downloads/>
- JEM SDK Jave Edit Model SDK (JEM-SDK-1.0.2.zip)  
<http://download.eclipse.org/tools/ve/downloads/drops/R-1.0.2-200412091401/index.html>

( )内は今回使用したファイル

また、後ほど使用しますので Tomcat もインストールしておきましょう。

Tomcat 5.0.28(<http://jakarta.apache.org/tomcat/index.html>)

### 2.2. データツール

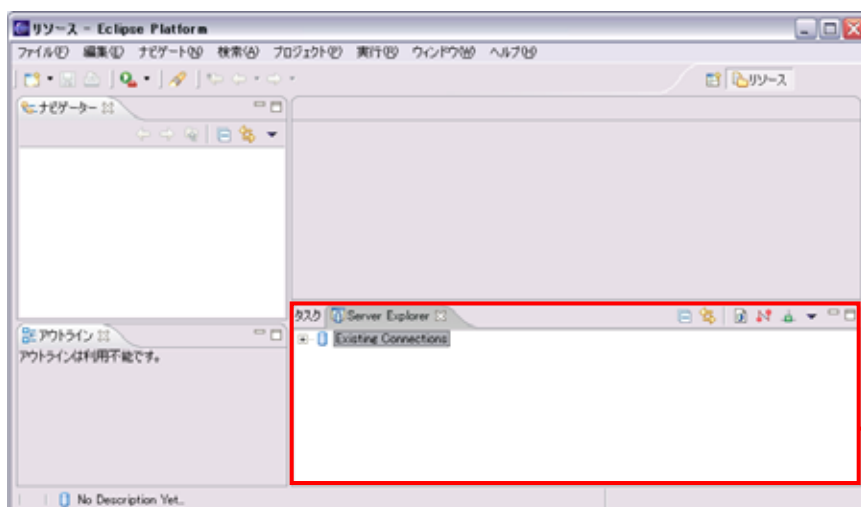
データツールはデータベースを操作するためのコンポーネントです。

#### (1) 接続設定

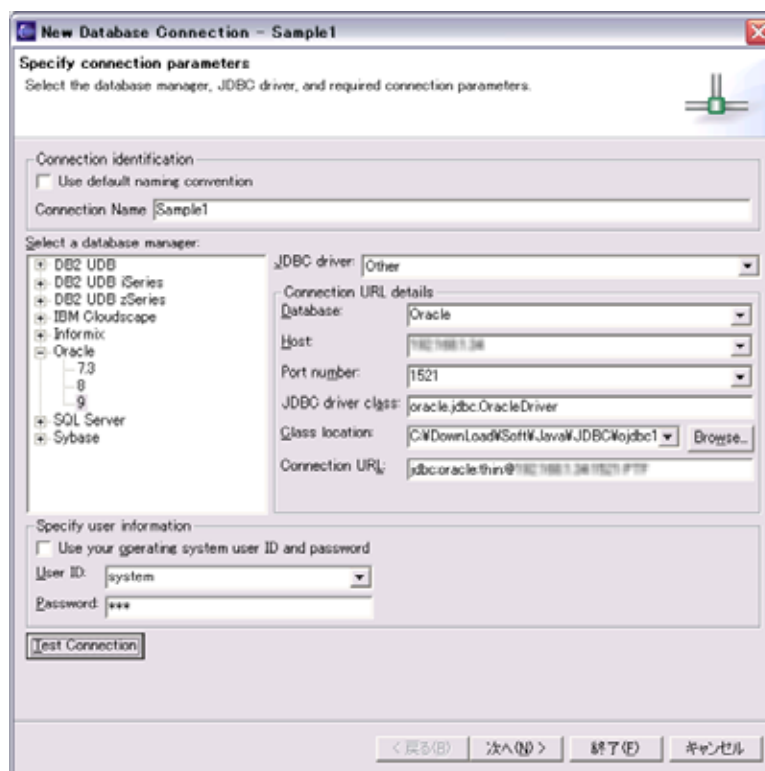
[ウィンドウ]-[ビューの表示]-[その他]を選択し、[Server Explorer Category]-[Server Explorer]を選択します。



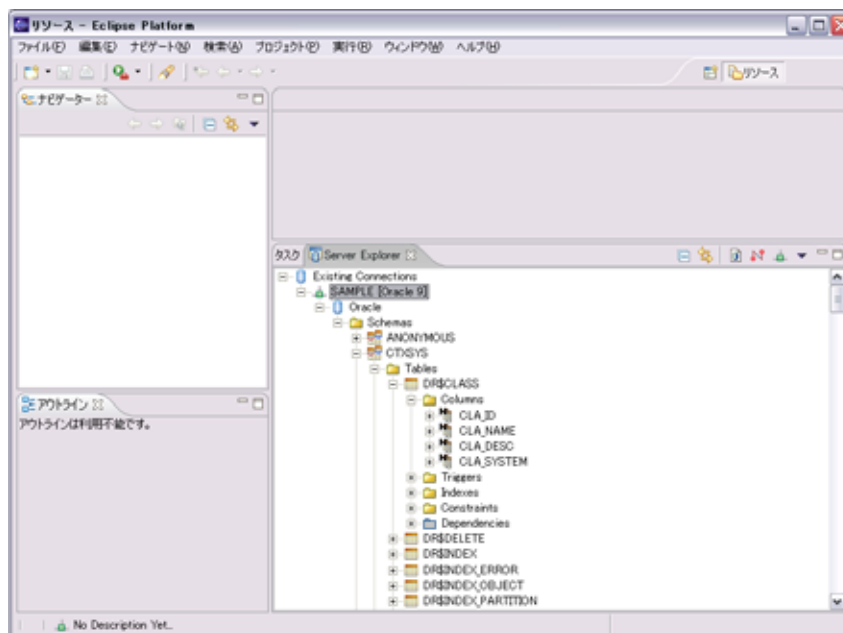
[Server Explorer]ビュー(下図中 )が表示されます。



[Server Explorer]ビューで[Existing Connection]を右クリックし、[New Connection...]を選択します。各パラメータを入力し[Test Connection]を押すと接続が確認できます。[次へ]を押すとフィルターの設定や接続情報の確認が出来ます。

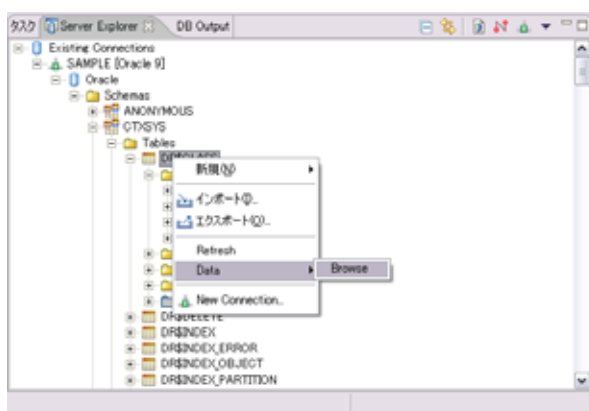


接続に成功していれば[Server Explorer]ビューからデータベースの構成が確認できます。

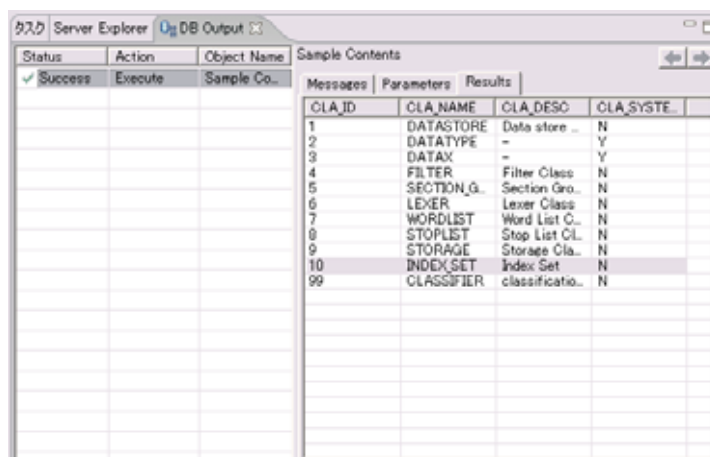


## (2) データ参照

[Server Explorer]ビューでテーブルを右クリックし、[Data]-[Browse]を選択します。

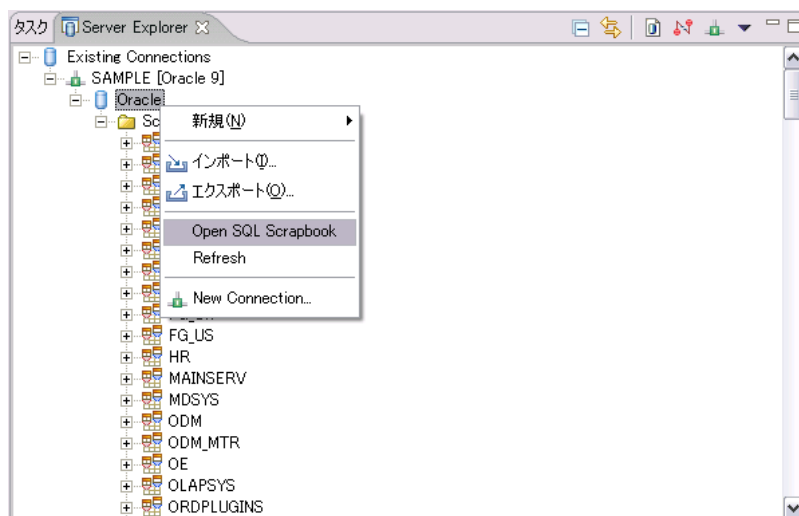


選択したテーブルのデータが表示されます。

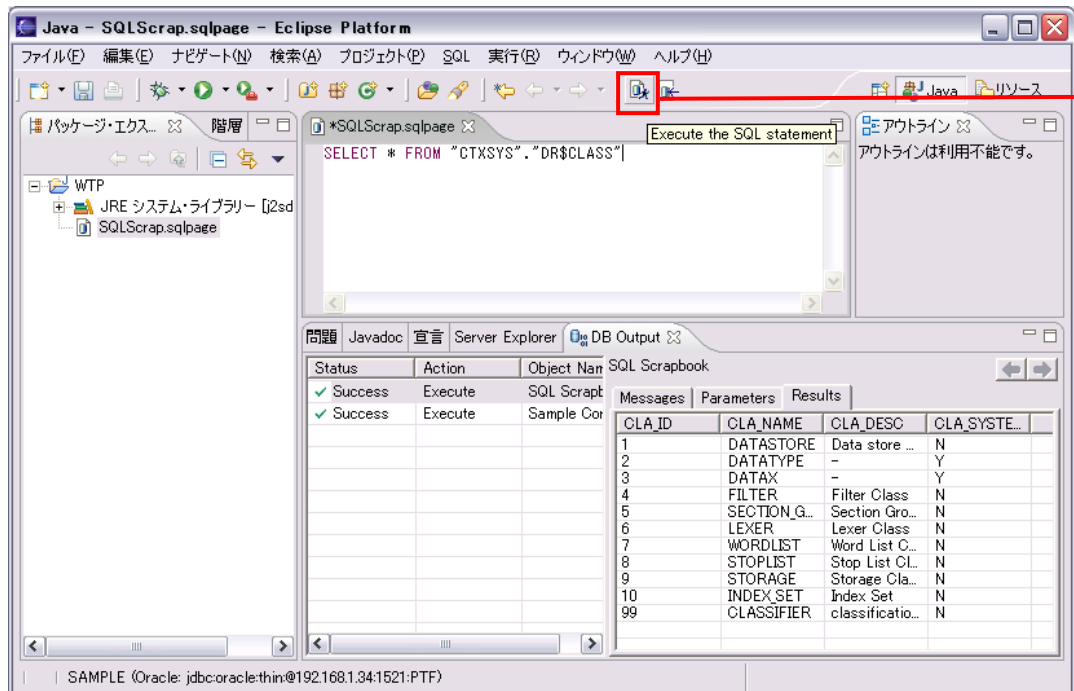


## (3) 任意のSQLの実行

接続ノードを右クリックし、[Open SQL Scrapbook]を選択し、任意のパスにファイルを作成します。



エディタが表示されるので、任意のSQLを入力し、[Execute the SQL statement(下図中)]を押します。





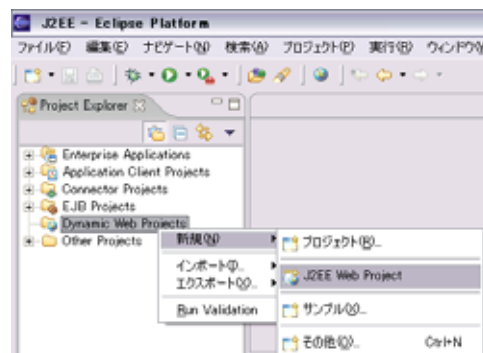
## 2.3. サーバツール

サーバツールでは、JSP のデバッグ、およびクライアント・サーバ間のデータの流を TCP/IP モニタで確認することができます。標準では Tomcat にのみ対応していますので、先ほどインストールした Tomcat 5.0.28 を使用します。

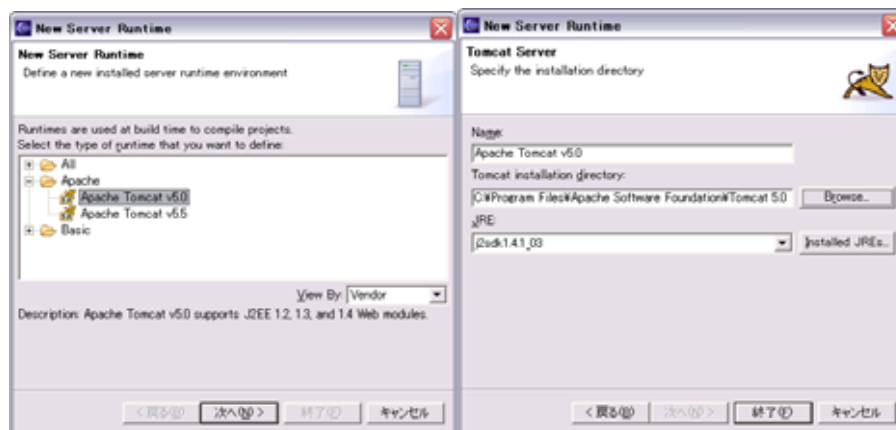
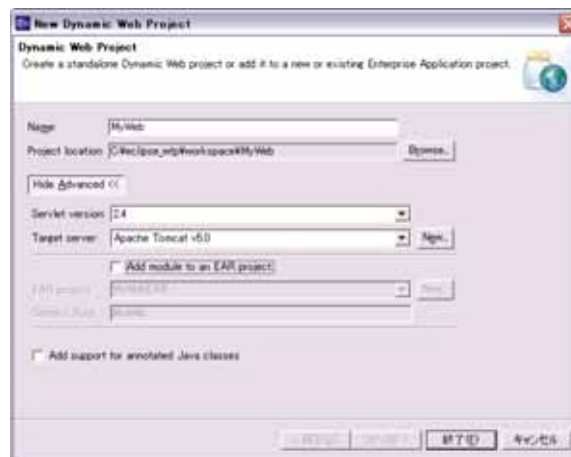
### (1) JSP のデバッグ

EclipseWTP の JSP デバッガではコンパイル済みの Java ソースではなく、JSP のソース上をステップ実行できます。まずデバッグ対象となる Web プロジェクトと JSP を作成します。

[J2EE パースペクティブ]を開き、[Project Explorer]で[Dynamic Web Project]を選択します。



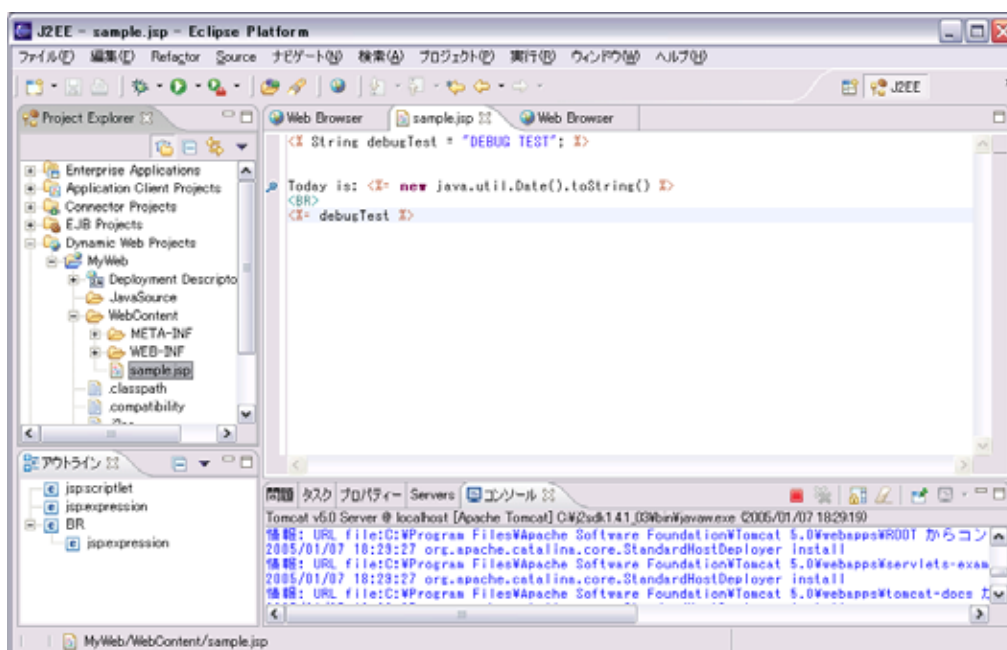
環境に合わせて各パラメータを入力し、[終了]を押下するとプロジェクトが生成されます。



以下の内容で”sample.jsp”を作成し、任意の場所にデバッグポイントを設定します。

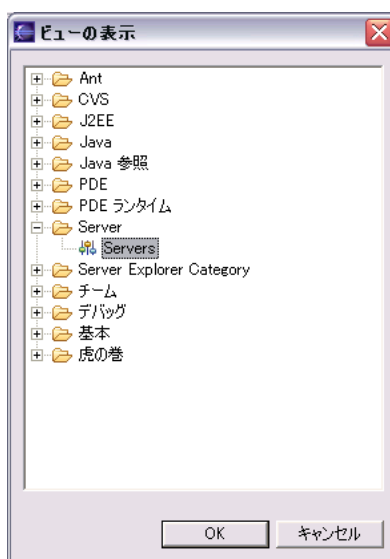
```
<% String debugTest = "DEBUG TEST"; %>

Today is: <%= new java.util.Date().toString() %>
<BR>
<%= debugTest %>
```

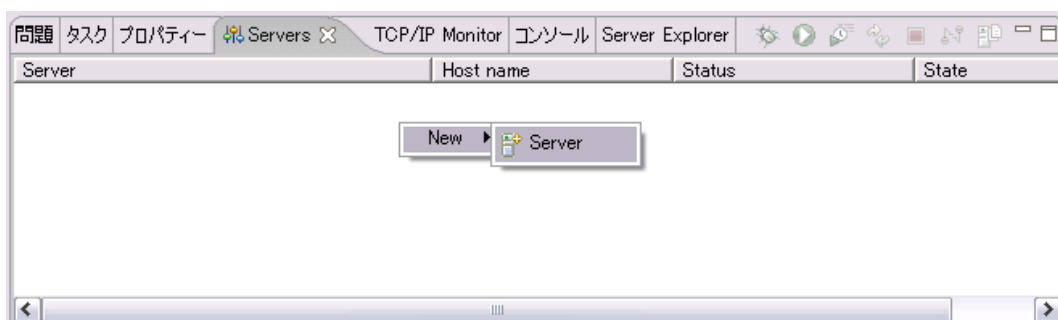


次に、デバッグ時に使用するサーバを登録します。

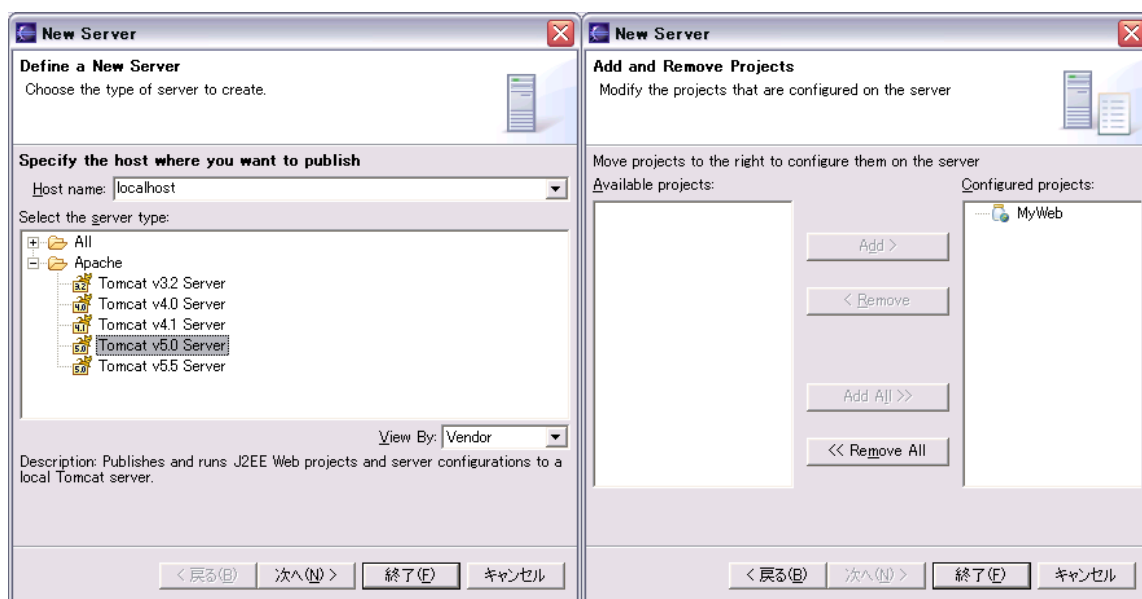
[ウィンドウ]-[ビューの表示]-[その他]を選択し、[Server]-[Servers]を選択します。



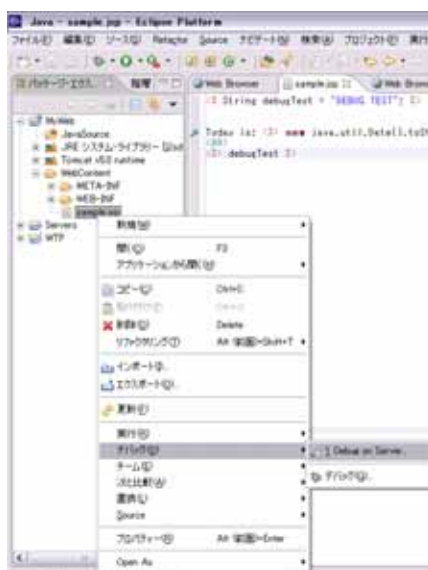
[Servers]ビューが開きますので、右クリックして[New]-[Server]を選択します。



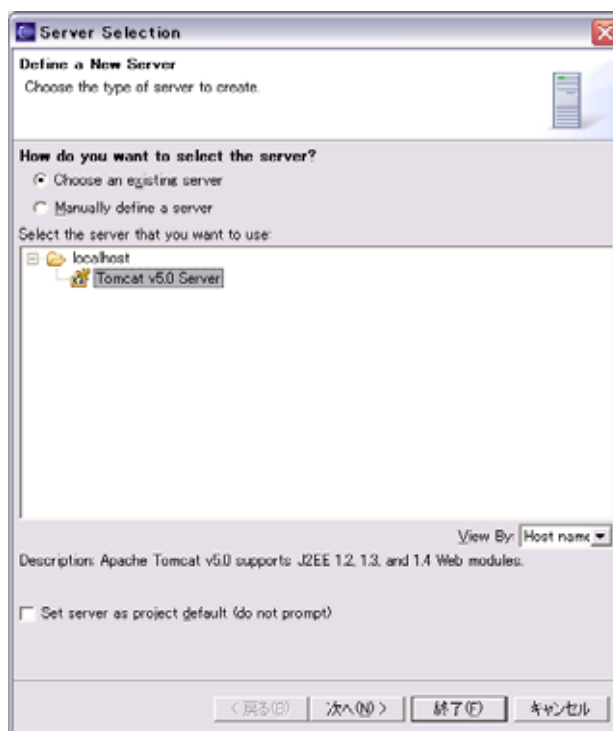
[New Server]ウィンドウが開きますので、Tomcat を選択し、[MyWeb]プロジェクトを登録します。



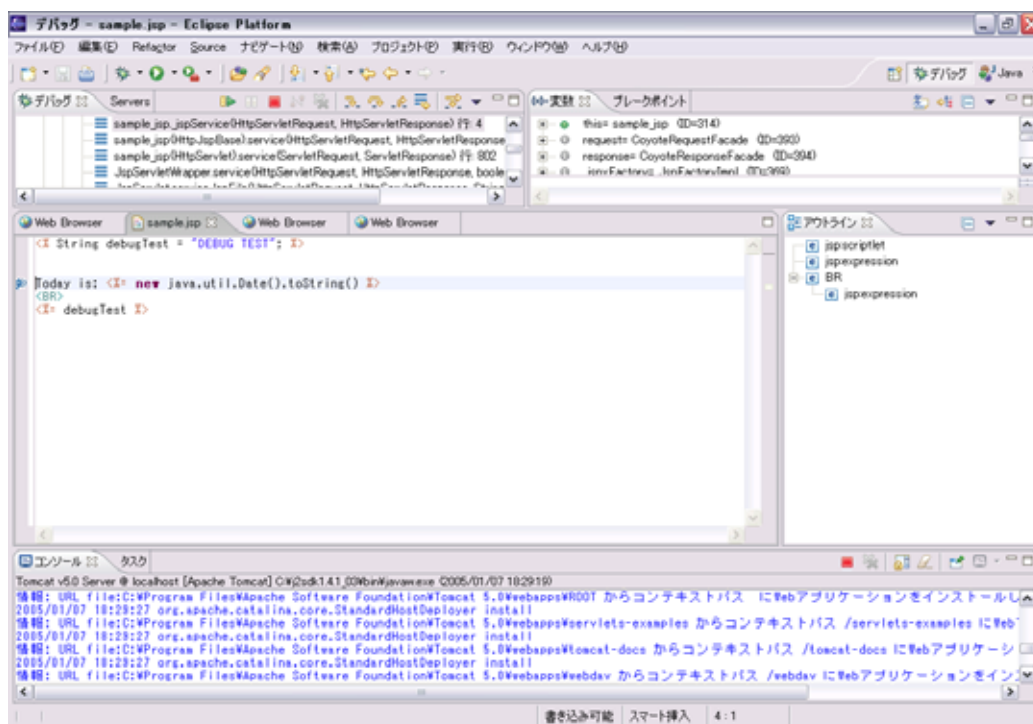
作成した"sample.jsp"を右クリックし、[デバッグ]-[Debug on Server]を選択します。



登録したサーバを選択し、[終了]を押下します。



サーバが起動しデバッグパースペクティブに切り替わります。



## (2) TCP/IP モニタ

メニューから[ウインドウ]-[設定]を選択します。

[Internet]-[TCP/IP Monitor]を選択し、[Add...]を押下します。



[New Monitor]画面で以下の様に入力し、OKを押します。

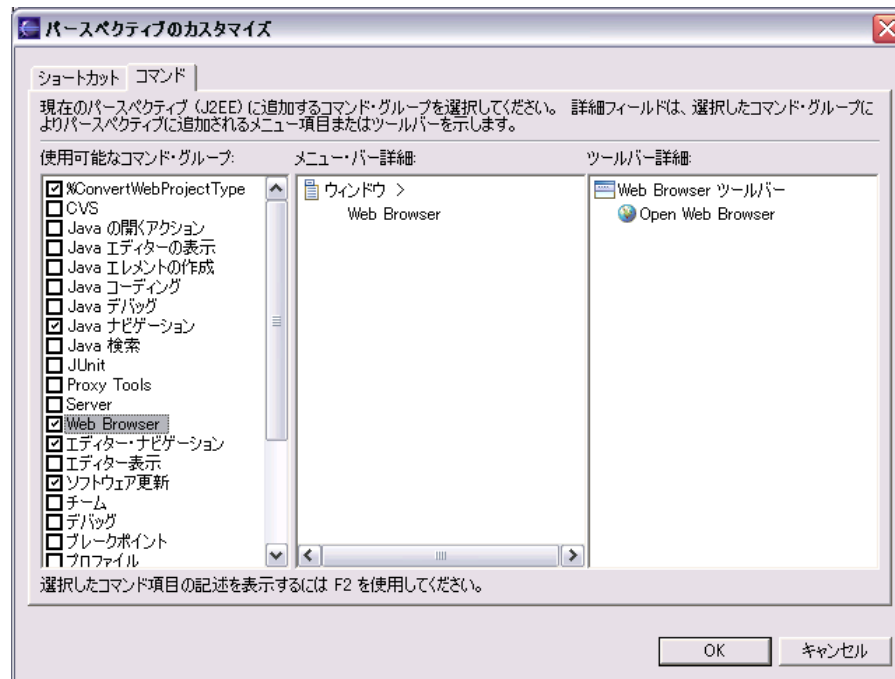


[Start]を押し、[Status]が”started”となったことを確認します。

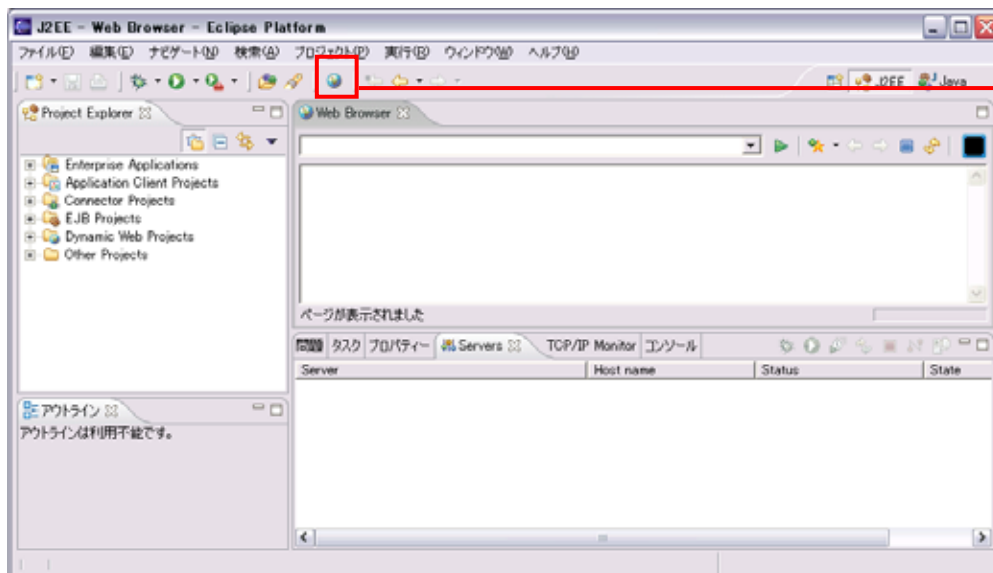


今回は Eclipse の内部ブラウザを使用しますので、コマンドグループに[Open Web Browser]を追加します。

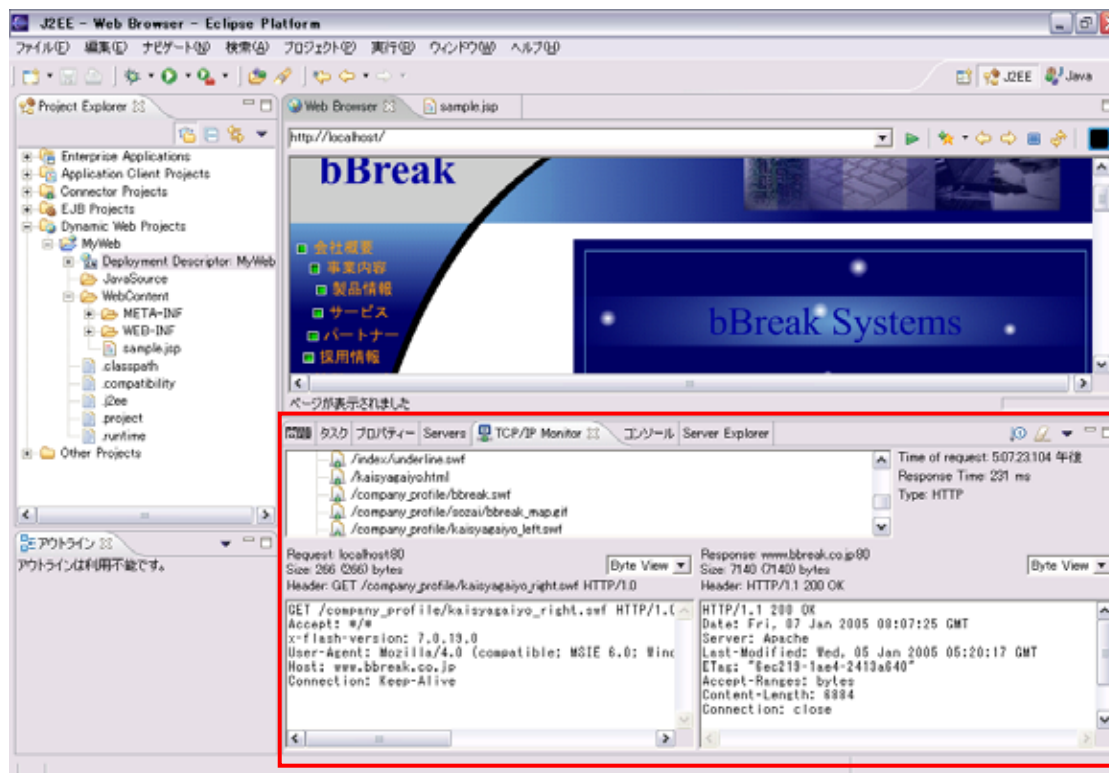
メニューから[ウィンドウ]-[パースペクティブのカスタマイズ]を選択し、以下の様に[Web Brower]にチェックを入れます。



[Open Web Brower(下図中 )]を押し、内部ブラウザを開きます。



URL に"<http://localhost>"と入力すると、モニタリングされた結果が[TCP/IP Monitor]ビューに表示されます。



## 2.4. 構造化エディタ(SSE)

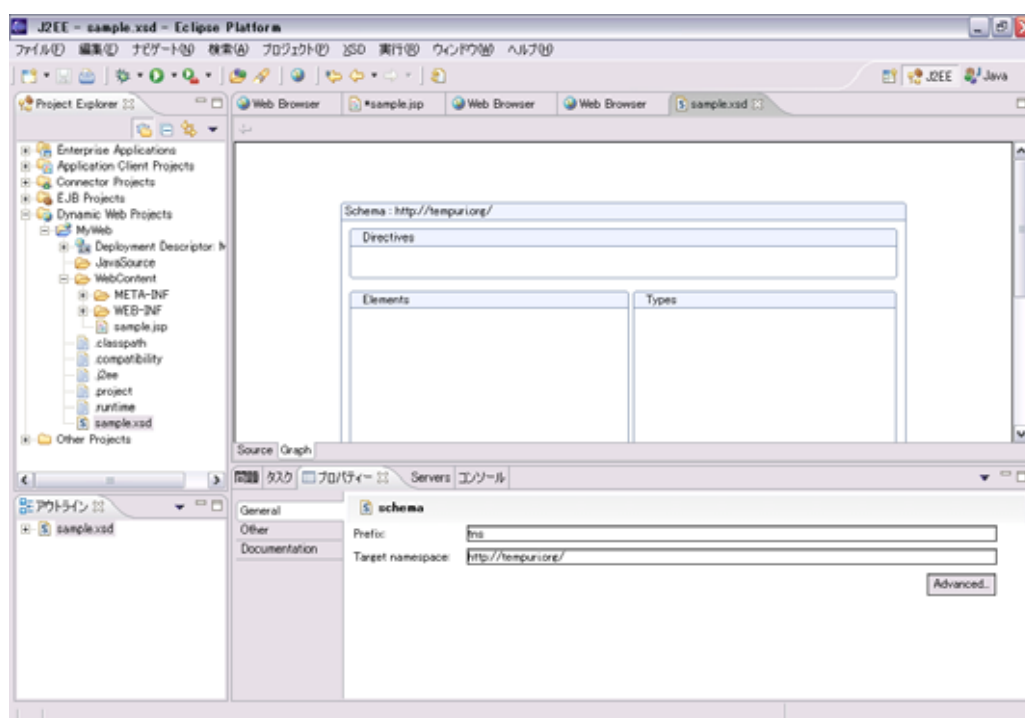
SSE(Structured Source Editors)はXML、HTML、XHTML、CSS、DTD、JavaScript、JSPのエディタとして使用でき、コードアシスト、ハイライト表示、タイプチェック、JSPのインデント修正などの機能を持ちます。現バージョンではいくつかの不具合も見られるようです。

## 2.5. XML・Webサービスツール

XML関連のエディタやウィザードは充実しており、スキーマをグラフィカルに表示するなどの機能があります。

### (1) XMLスキーマエディタ

エディタの[Graph]タブを選択するとスキーマ情報をグラフィカルに表示することが出来ます。





### 3. 参考資料

- EclipseWTP プロジェクトページ(<http://www.eclipse.org/webtools/>)
- EclipseWTP チュートリアル  
(<http://www.eclipse.org/webtools/initial-contribution/IBM/Getting%20Started.html>)
- Eclipse3.0 と WebToolsPlatform プロジェクトによる Web アプリケーション開発  
(<http://www-6.ibm.com/jp/software/swcoc/events/pdf/ed041029-01.pdf>)

## 4. 最後に

実際の開発に使えるかどうかはプラグインやツールの充実度次第かと思っていましたが、デフォルト構成でJ2EEに必要なセットが揃っていますのでそのままでも十分に使用できます。特にエディタとXML関連のコンポーネントはかなり充実しているという印象を受けました。まだ開発中のためエディタやTCP/IP モニタなどでいくつか不具合が見られますが致命的なものではないので、今後標準となる可能性のある環境にいち早く触れてみてはどうでしょうか。

なお、今回触れていないコンポーネントについても今後随時記述を追加していく予定です。

記述内容に何かお気づきの点、質問等ありましたら下記までご連絡ください。

開発部 横井 朗 yokoi@bbreak.co.jp
-----------------------------------