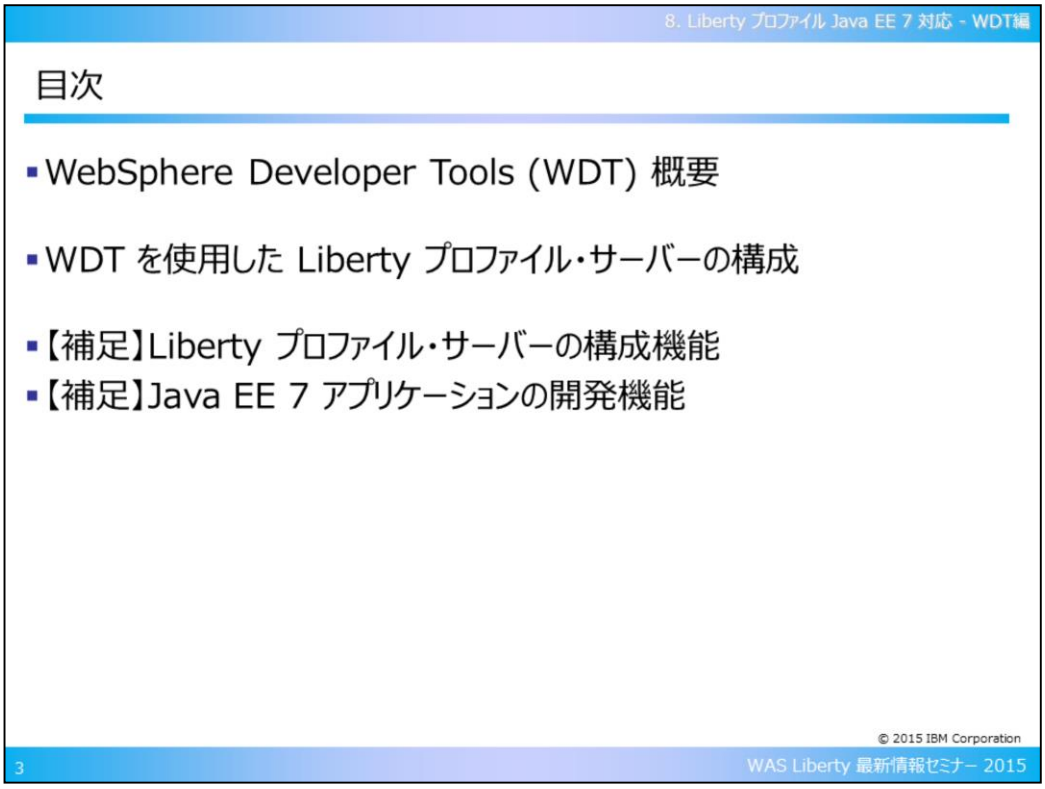

WAS Liberty 最新情報セミナー 2015
IBM
～ 次世代ランタイム Libertyで変わる Java EE システム～
8. Libertyプロファイル Java EE 7 対応
WDT 編
© 2015 IBM Corporation
1

Disclaimer

- この資料は日本アイ・ビー・エム株式会社ならびに日本アイ・ビー・エム システムズ・エンジニアリング株式会社の正式なレビューを受けておりません。
- 当資料は、資料内で説明されている製品の仕様を保証するものではありません。
- 資料の内容には正確を期するよう注意しておりますが、この資料の内容は2015年7月現在の情報であり、製品の新しいリリース、PTFなどによって動作、仕様が変わる可能性があるのでご注意下さい。
- 今後国内で提供されるリリース情報は、対応する発表レターなどでご確認ください。
- ＩＢＭ、ＩＢＭロゴおよびibm.comは、世界の多くの国で登録されたInternational Business Machines Corporationの商標です。他の製品名およびサービス名等は、それぞれＩＢＭまたは各社の商標である場合があります。現時点でのＩＢＭの商標リストについては、www.ibm.com/legal/copytrade.shtmlをご覧ください。
- 当資料をコピー等で複製することは、日本アイ・ビー・エム株式会社ならびに日本アイ・ビー・エム システムズ・エンジニアリング株式会社の承諾なしではできません。
- 当資料に記載された製品名または会社名はそれぞれの各社の商標または登録商標です。
- JavaおよびすべてのJava関連の商標およびロゴはOracleやその関連会社の米国およびその他の国における商標または登録商標です。
- Microsoft, Windows および Windowsロゴは、Microsoft Corporationの米国およびその他の国における商標です。
- Linuxは、Linus Torvaldsの米国およびその他の国における登録商標です。
- UNIXはThe Open Groupの米国およびその他の国における登録商標です。

© 2015 IBM Corporation



この資料では、WebSphere Developer Tools (WDT) の概要と、WDT を使用した Liberty プロファイル・サーバーの構成について説明します。

Liberty プロファイル・サーバーの構成機能に関しては、補足の部分にも記載がありますので合わせて参考にしてください。

補足資料には、WDT が提供する Java EE 7 アプリケーションの開発機能も載せてあります。

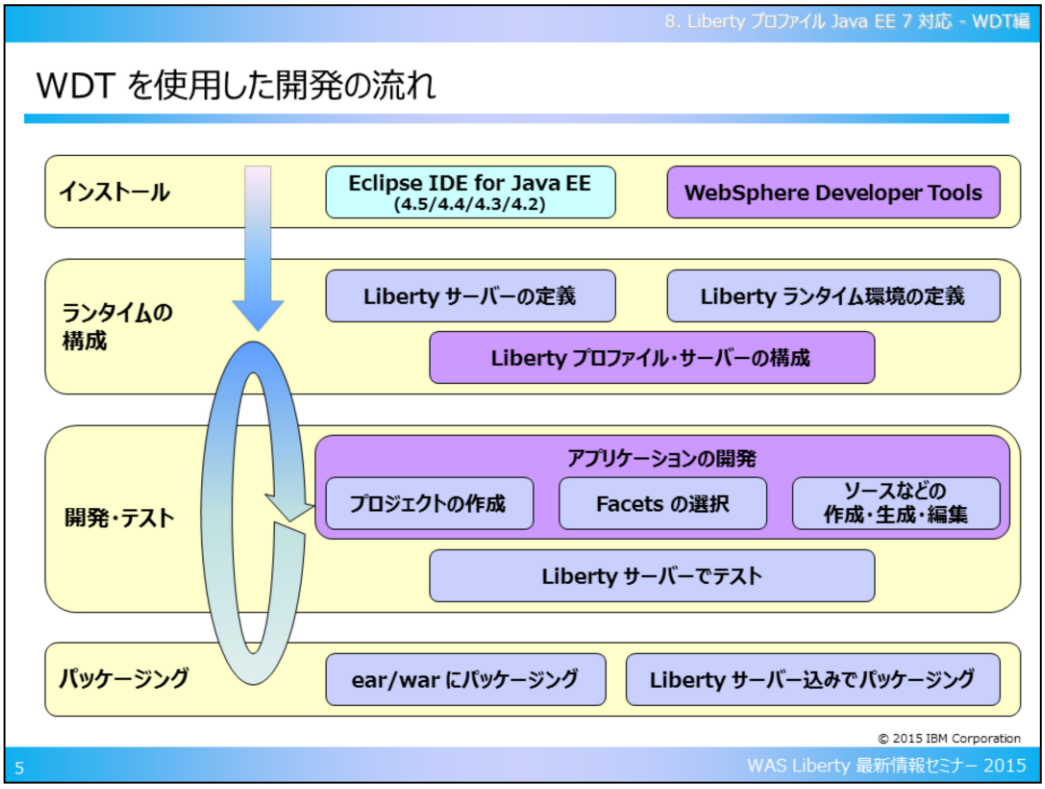


WDT は、WAS 向けの Java EE アプリケーションを開発・アセンブルするための軽量のツール・セットで、Eclipse のアドオン・ツールです。WDT が提供する機能は大きく２つに分類できます。

１つめは、Liberty プロファイル・サーバーの構成機能です。たとえば、WDT が提供する Liberty Profile Configuration Editor を使用すると、Liberty サーバーの構成ファイルである server.xml ファイルをビジュアルに編集できます。bootstrap.properties, jvm.options, server.env などのファイルも作成・編集できます。また、アプリケーションを含めた状態で、Liberty プロファイル・サーバーを zip ファイルなどにパッケージングすることもできます。

２つめは、 Java EE 7 アプリケーションの開発機能です。アプリケーションを構成する様々なファイルを作成・編集するために、各種のエディター、ウィザード、ツールなどが提供されており、Java EE アプリケーションの開発をスムーズに進めることができます。

尚、WDT が提供する Liberty プロファイル・サーバーの構成機能は、アプリケーションの開発だけではなく、Liberty プロファイル・サーバーの汎用的な構成ツールとして使用することができます。

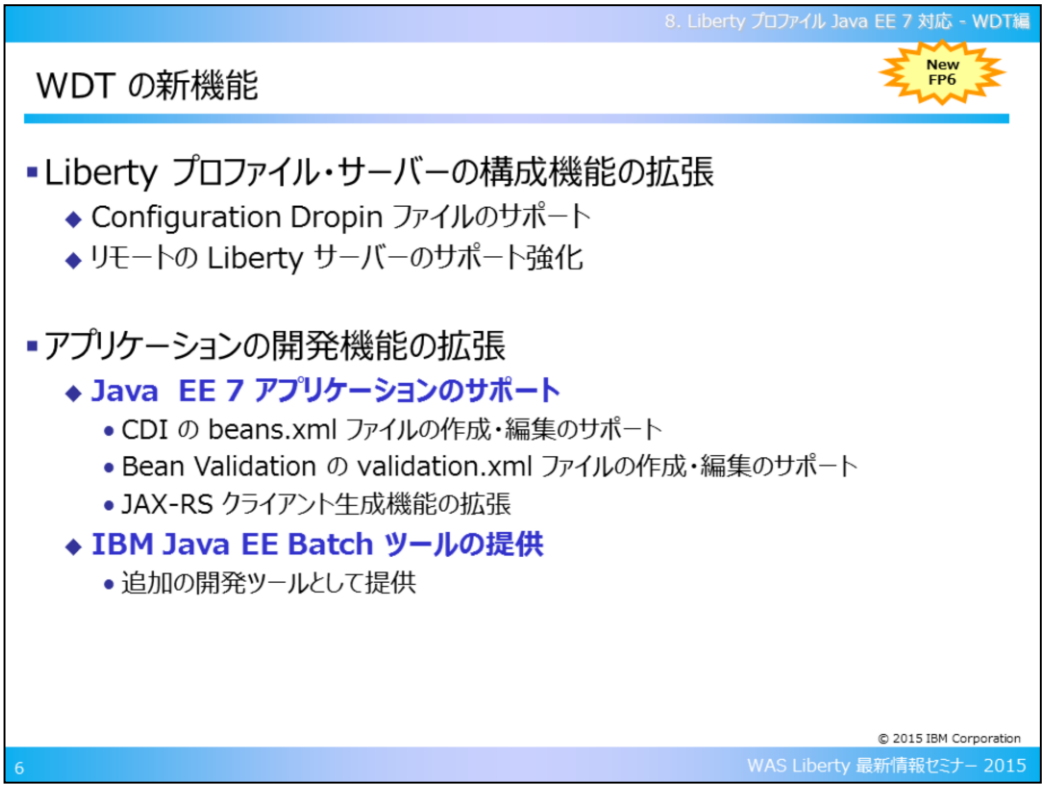


この図は、WDT を使用した開発の流れを表したものです。

最初のステップは、Eclipse と WDT のインストールで、次のステップがランタイムの構成です。ランタイムの構成では、開発・テストに使用する Liberty サーバーのサーバー定義を Eclipse に追加します。この時、Liberty プロファイルのインストール先や使用する JRE を指定し、Liberty サーバーのランタイム環境を定義し、Liberty サーバーを作成します。ランタイムの構成では、Liberty プロファイル・サーバーの構成も行います。

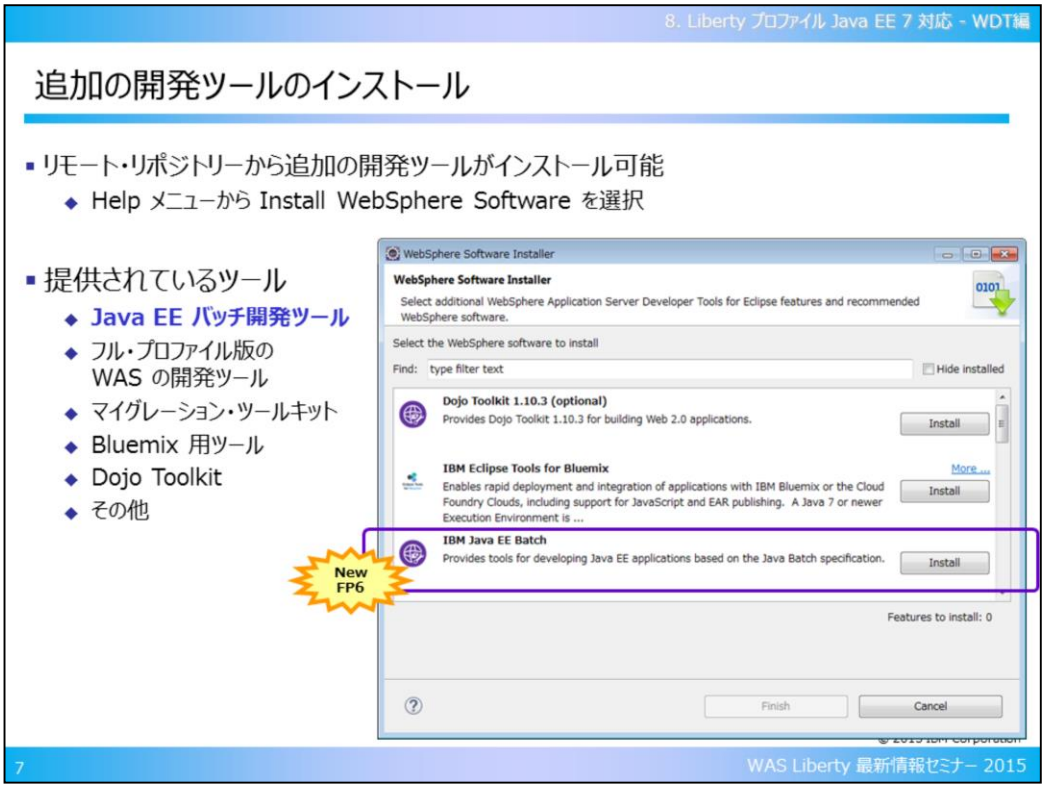
次のステップは、アプリケーションの開発・テストです。WDT が提供する各種エディター、ウィザード、ツールを使用してアプリケーションを開発し、前のステップで作成した Liberty サーバーにデプロイしてテストを行います。

開発・テストが完了したアプリケーションは、WDT の機能を使用して、ear ファイルや war ファイルにエクスポートしたり、Liberty サーバーのランタイムを含めた状態で zip ファイルなどにパッケージングすることができます。



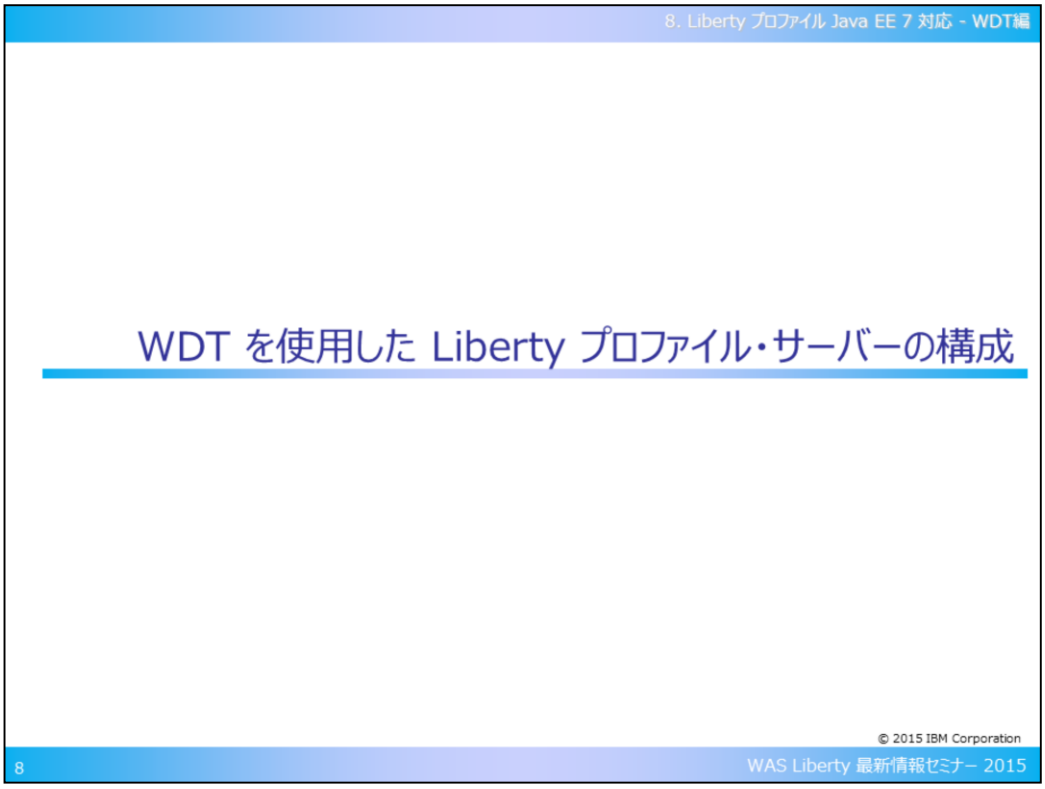
FixPack 6 から提供された主な新規は上記の通りです。

アプリケーションの開発機能の拡張としては、Java EE 7 アプリケーションのサポートと IBM Java EE Batch ツールの提供が大きな拡張点です。

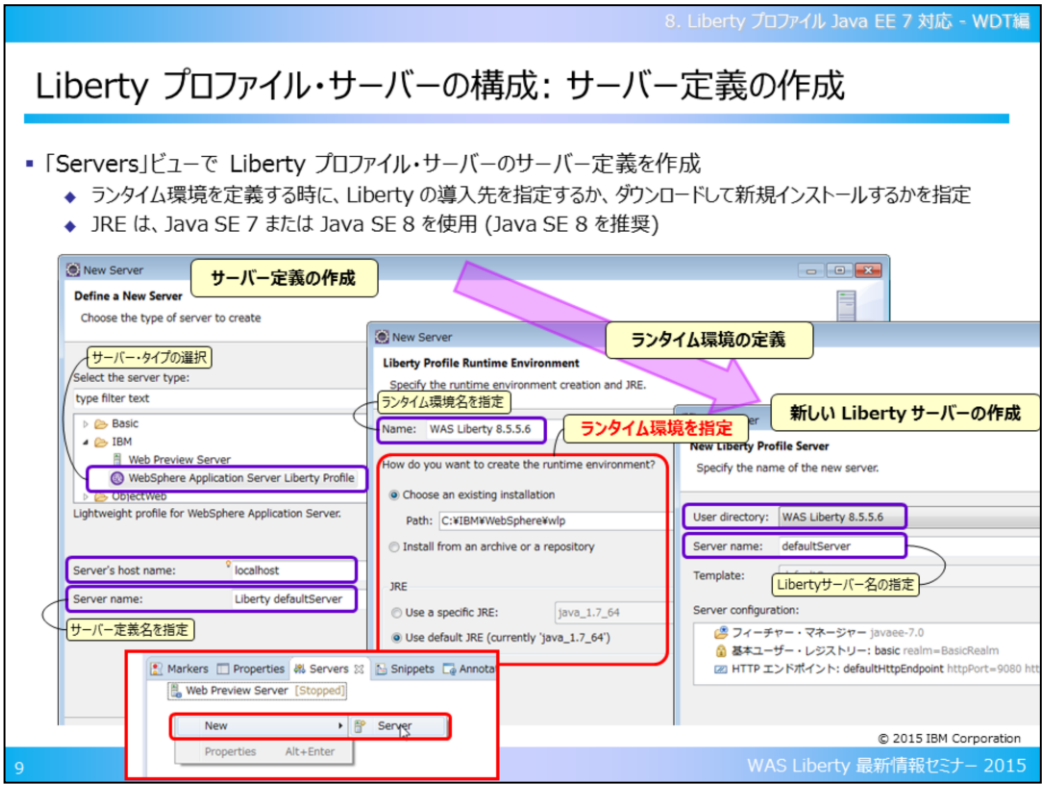


Help メニューから Install WebSphere Software を選択し、WebSphere Software Installer を起動して、追加の開発ツールをインストールすることができます。

提供されている主な追加の開発ツールは上記の通りです。IBM Java EE Batch ツールも、追加の開発ツールとして提供されています。



以降のページでは、WDT を使用した Liberty プロファイル・サーバーの構成に関して説明します。

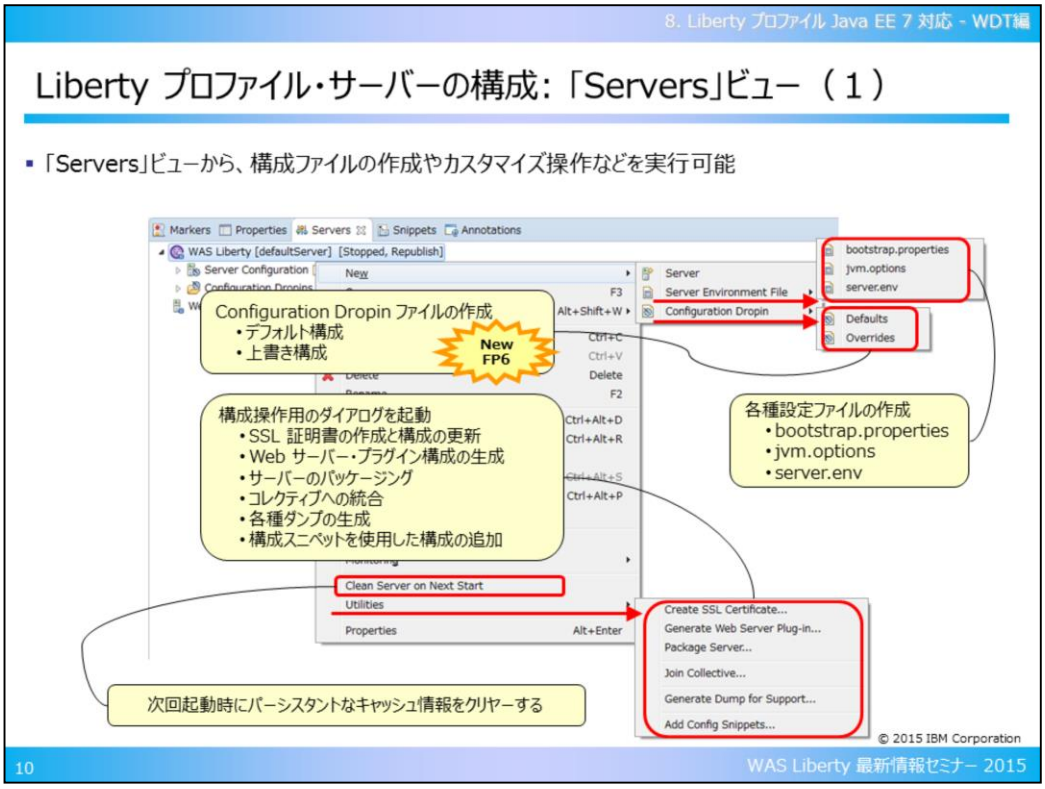


最初に説明するのは、サーバー定義の作成です。「Servers」ビューのポップアップ・メニューから New -> Server を選択して開始します。

初めのウイザード画面は、Eclipse へのサーバー定義の追加です。サーバー・タイプとして Liberty Profile を選択し、ホスト名とサーバー定義の名前を指定します。

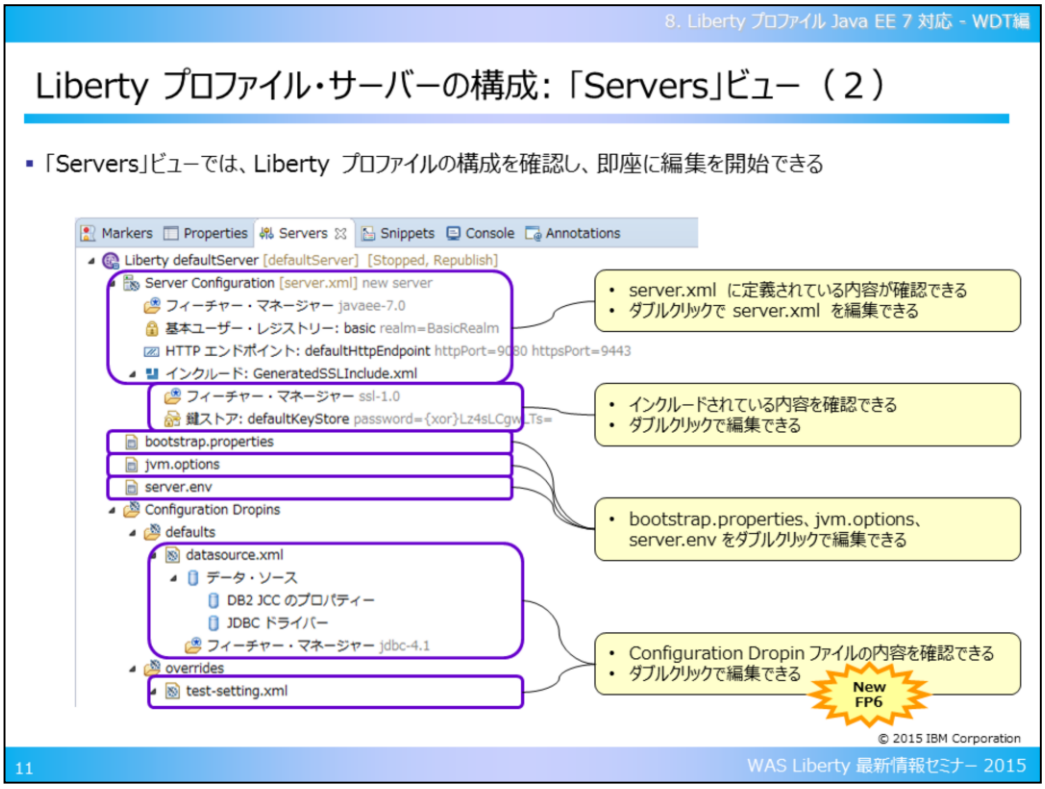
次のウイザード画面は、ランタイム環境の定義です。Liberty Profile がインストールされているパスと JRE を指定します。この画面から、Liberty Profile をダウンロードしてインストールすることもできます。

最後のウイザード画面は、Liberty サーバーの作成です。Liberty サーバーが既に存在する場合は、この画面で既存の Liberty プロファイル・サーバーを選択することもできます。

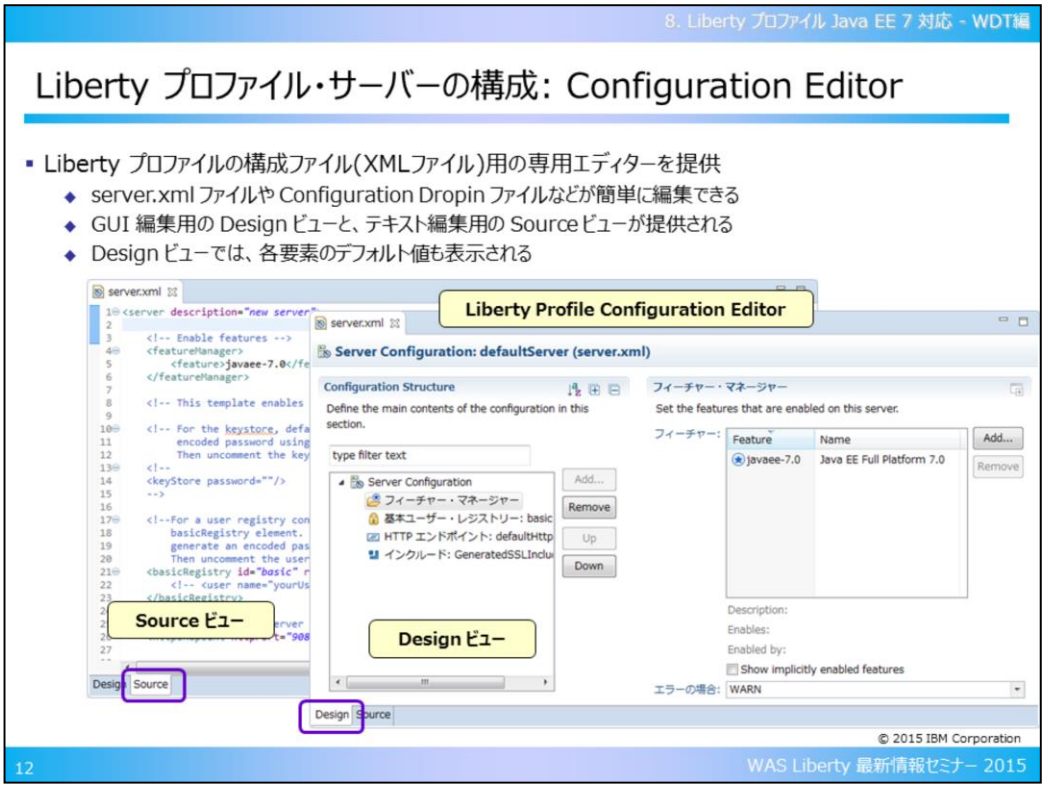


Liberty プロファイル・サーバーのサーバー定義を作成すると、「Servers」ビューから該当の Liberty サーバーの構成をカスタマイズできるようになります。

上の図は、サーバー定義のポップアップ・メニューから起動できる、カスタマイズ操作を記載したものです。



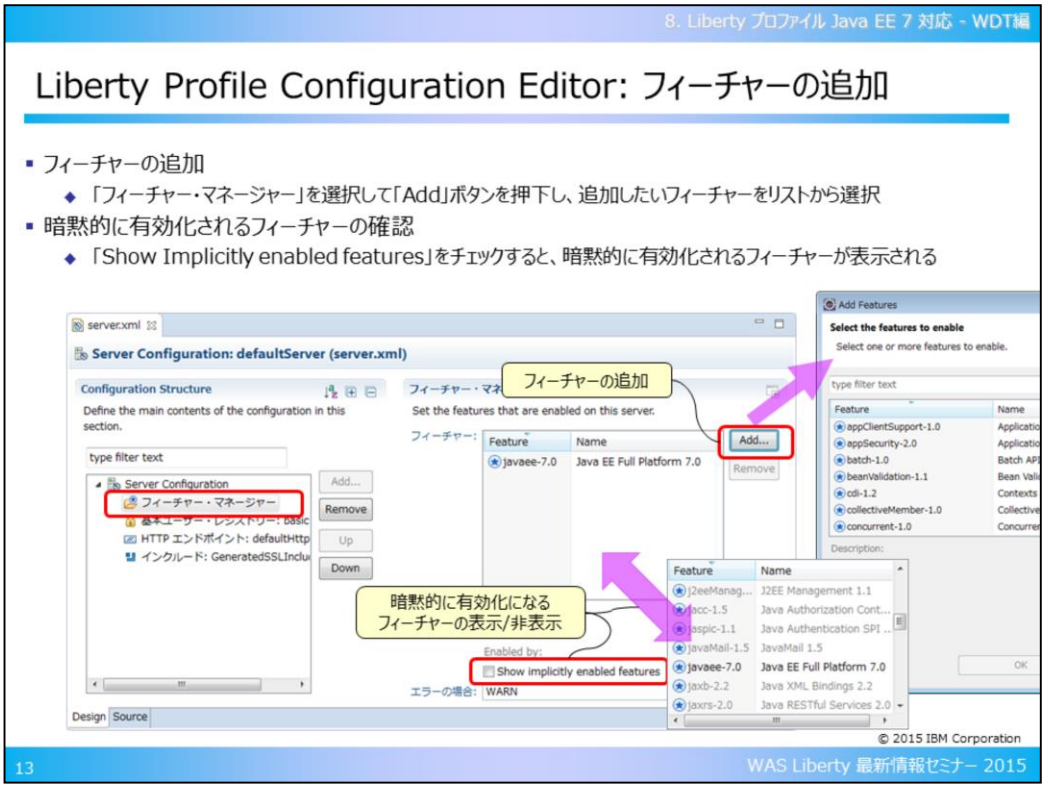
「Servers」ビューの表示を展開すると、Liberty プロファイル・サーバーの構成を確認することができます。
また、各項目をダブル・クリックすることで、該当のファイルをエディターで開き、編集することできます。



上の図は、Liberty Profile Configuration Editor の画面です。

Liberty Profile Configuration Editor は、Liberty プロファイルの構成ファイル(XMLファイル)用の専用エディターで、「Servers」ビューで Server Configuration (server.xml) などをダブル・クリックすると表示されます。

GUI 編集用の Design ビューと、テキスト編集用の Source ビューが提供されており、server.xml ファイルや Configuration Dropin ファイルなどの編集に使用できます。Design ビューでは、各要素のデフォルト値も表示されます。



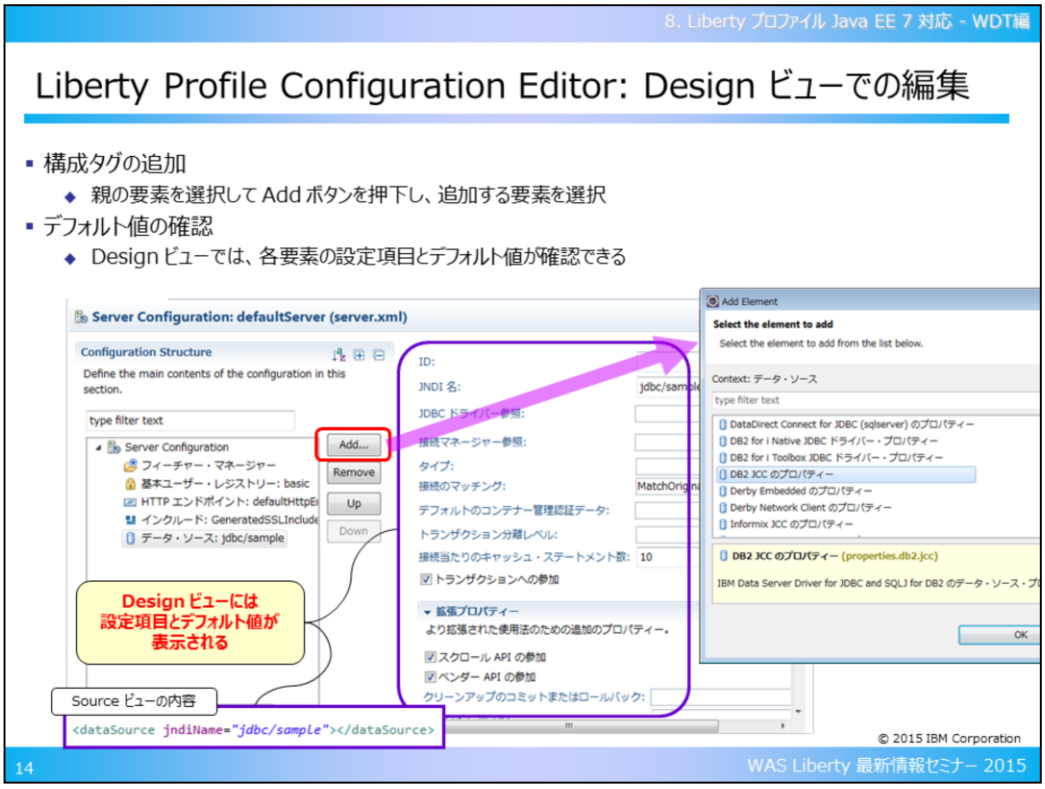
上の図は、Liberty Profile Configuration Editor を使用して、Liberty サーバーにフィーチャーを追加する流れを示したものです。

フィーチャーを追加する場合は、まず、エディターの左側のリストで「フィーチャー・マネージャー」を選択し、画面の右側に現状の定義内容を表示させます。

画面の右側にある Add ボタンを押し、フィーチャーの一覧を表示し、一覧から追加したいフィーチャーを選択します。

この操作により、フィーチャーが追加されます。

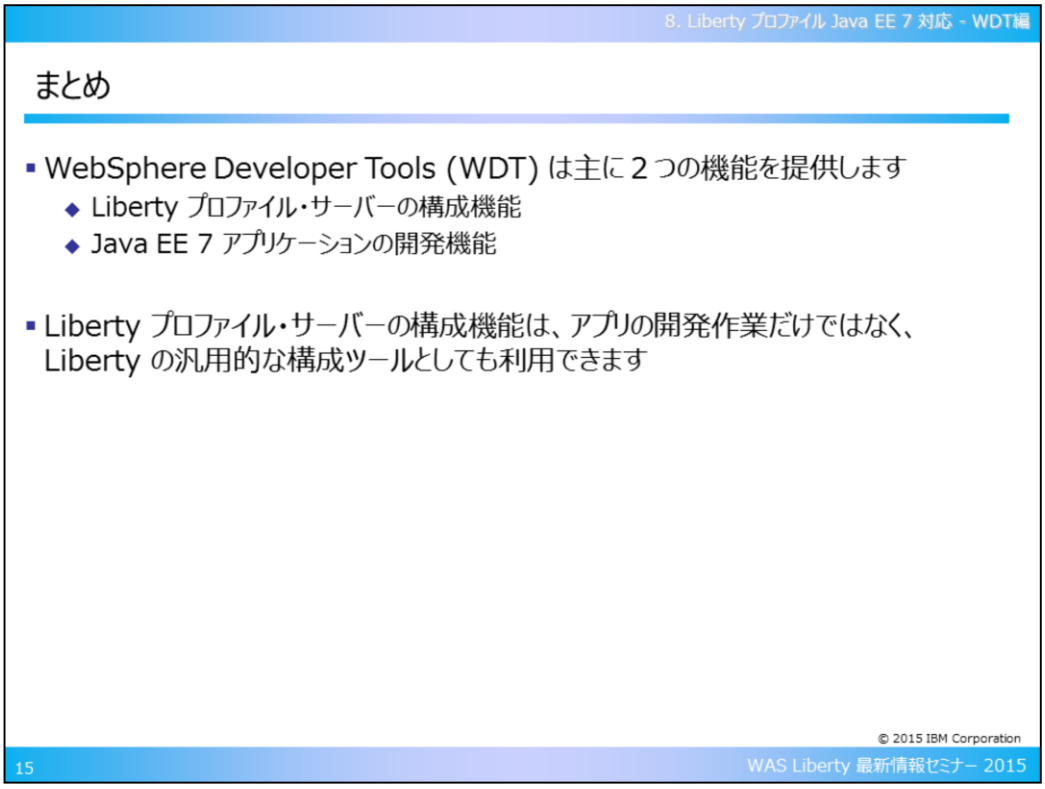
画面の右下に表示されている "Show implicitly enabled features" をチェックすると、暗黙的に有効化されるフィーチャーを表示することできます。



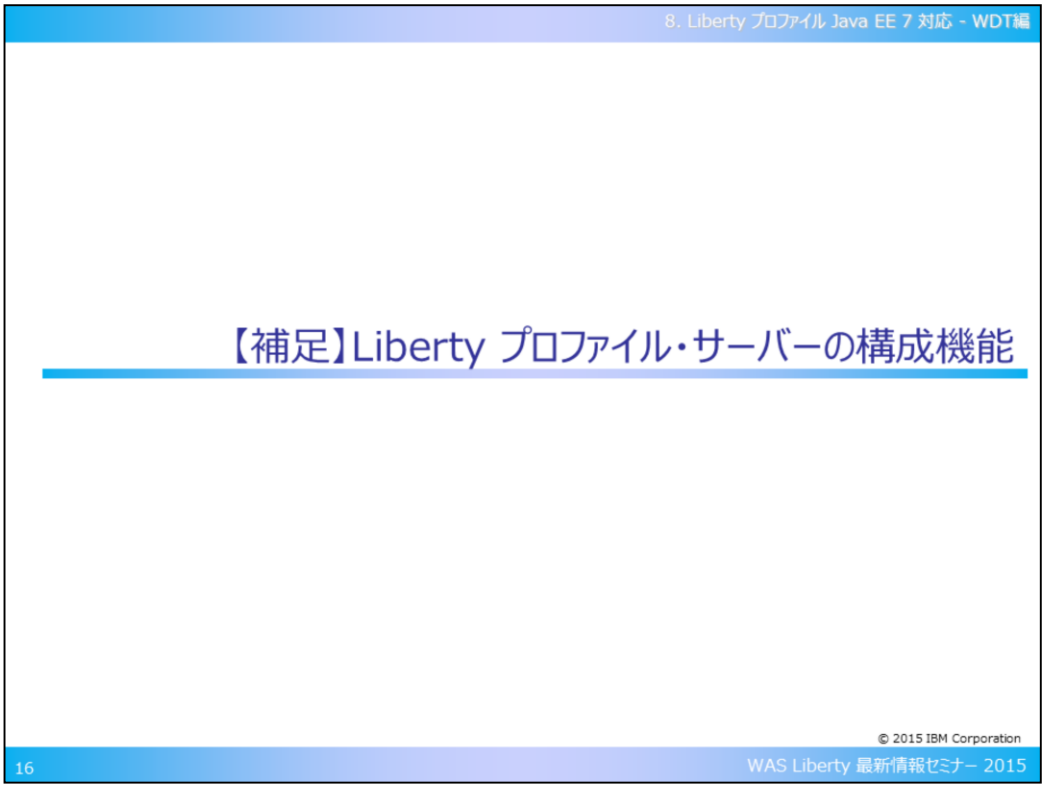
Liberty Profile Configuration Editor の左側のリストで構成要素を選択すると、その要素の定義内容が画面の右側に表示されます。

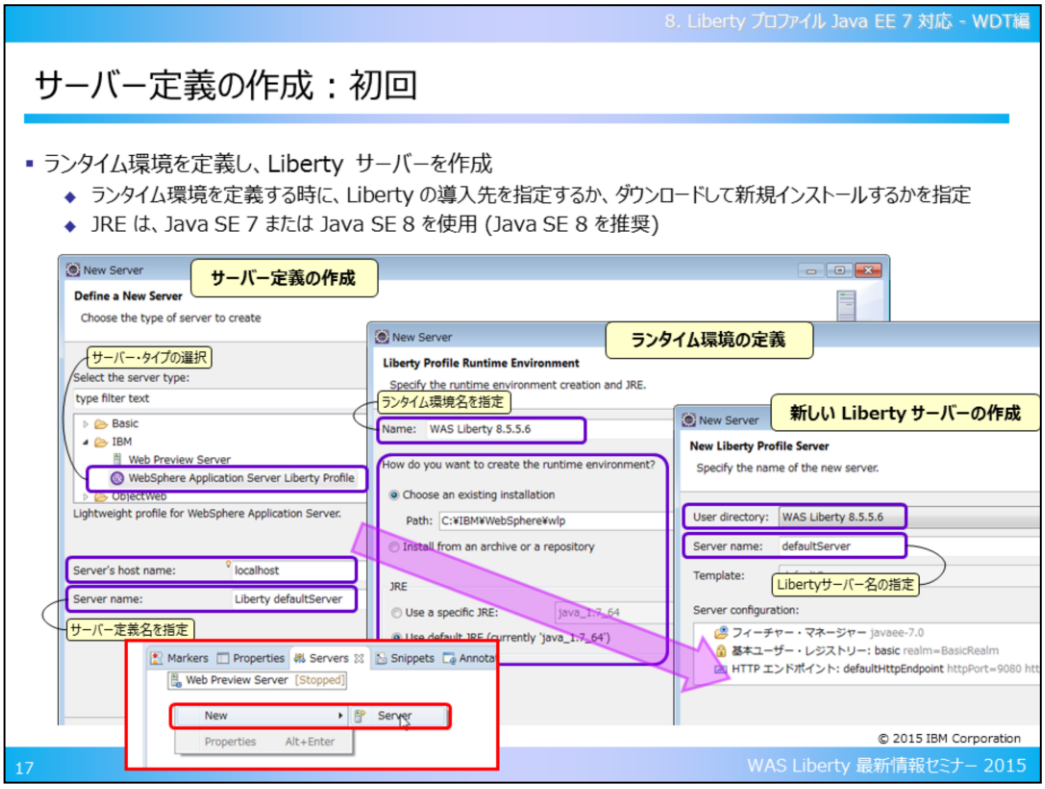
Liberty Profile Configuration Editor の Design ビューでは、各要素の設定項目とデフォルト値が確認できます。

構成要素を追加する場合は、エディターの左側のリストで親の要素を選択し、Add ボタンを押下します。追加可能な子要素の一覧が表示されるので、目的の要素を選択して追加します。



このセッションでは、上記の内容をご紹介しました。


8. Liberty プロファイル Java EE 7 対応 - WDT編
【補足】Liberty プロファイル・サーバーの構成機能
© 2015 IBM Corporation
16
WAS Liberty 最新情報セミナー 2015



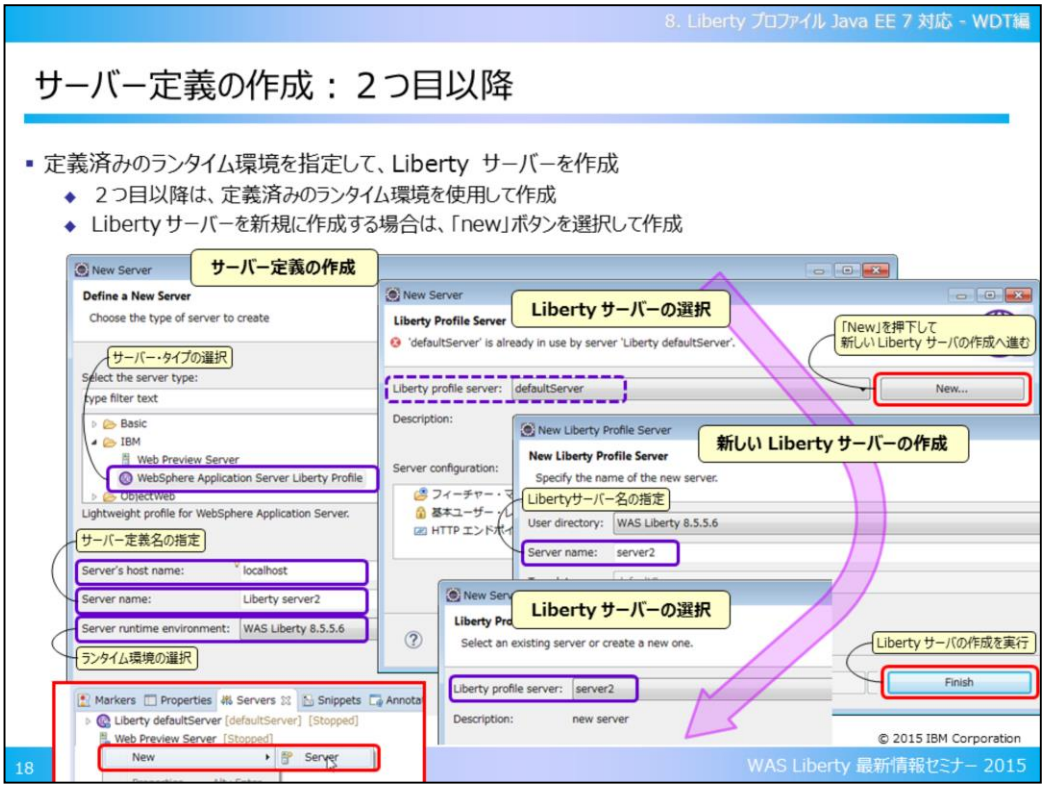
上の図は、初回のサーバー定義の作成の流れを示したものです。

サーバー定義の作成は、「Servers」ビューのポップアップ・メニューから New -> Server を選択して開始します。

初めのウイザード画面は、Eclipse へのサーバー定義の追加です。サーバー・タイプとして Liberty Profile を選択し、ホスト名とサーバー定義の名前を指定します。

次のウイザード画面は、ランタイム環境の定義です。Liberty Profile がインストールされているパスと JRE を指定します。この画面から、Liberty Profile をダウンロードしてインストールすることもできます。

最後のウイザード画面は、Liberty サーバーの作成です。Liberty サーバーが既に存在する場合は、この画面で既存の Liberty プロファイル・サーバーを選択することもできます。

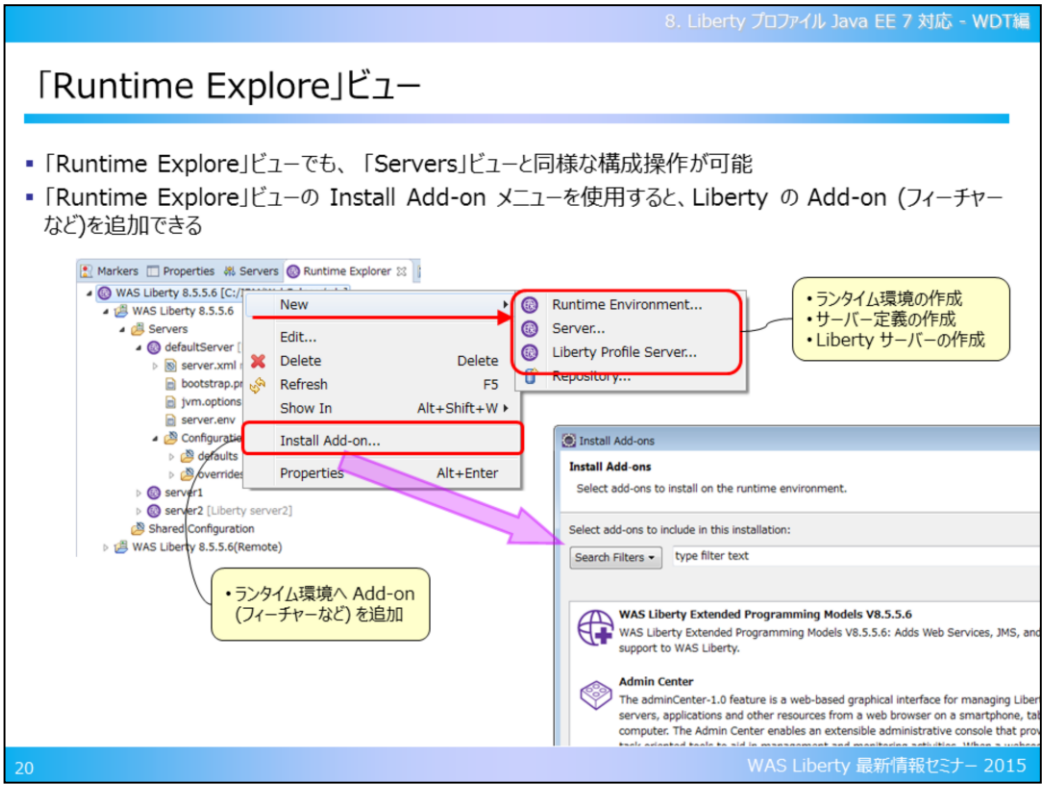


２つ目以降は、すでに登録してあるランタイム環境を指定して作成します。

Liberty サーバーを選択する画面で New ボタンを押すと、新しい Liberty サーバーを作成する画面に進むことできます。

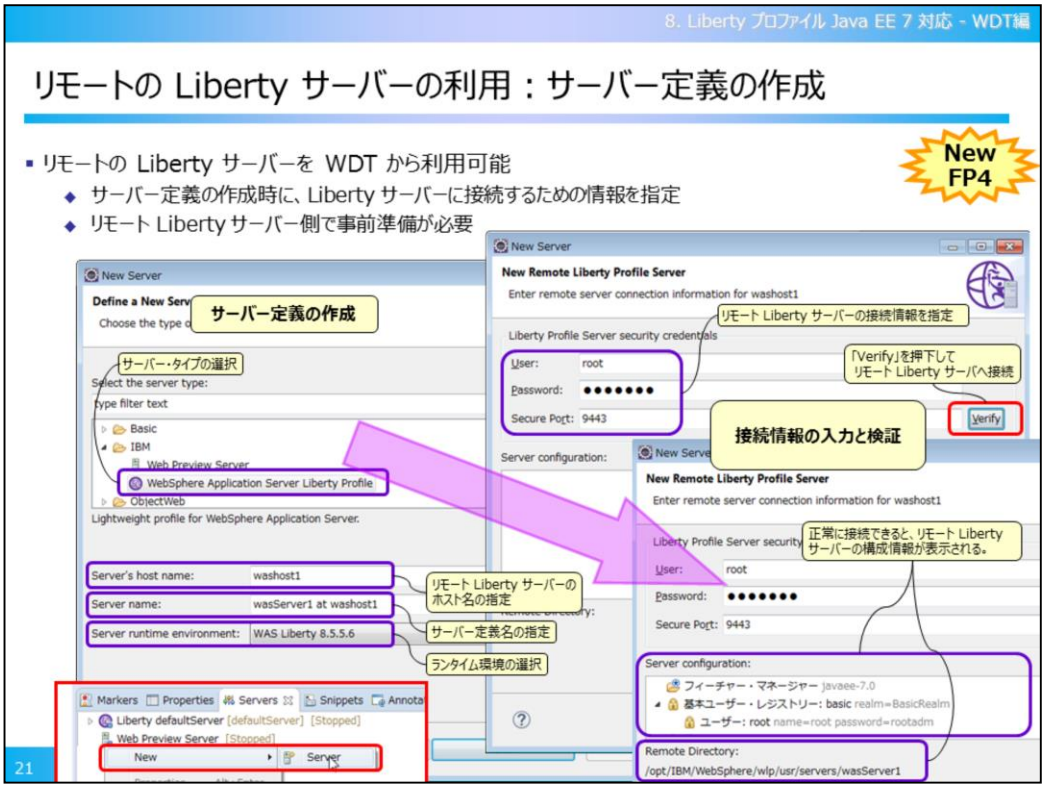


「Enterprise Explore」ビューからも Liberty プロファイル・サーバーの構成をカスタマイズすることできます。
このビューからは、shared フォルダー内のファイルも編集できます。



「Runtime Explore」ビューでも「Servers」ビューと同様な構成操作が行えますが、こちらはランタイムのカスタマイズに向いたビューになります。

Install Add-on メニューを使用すると、Liberty の Add-on (フィーチャーなど)を追加できます。



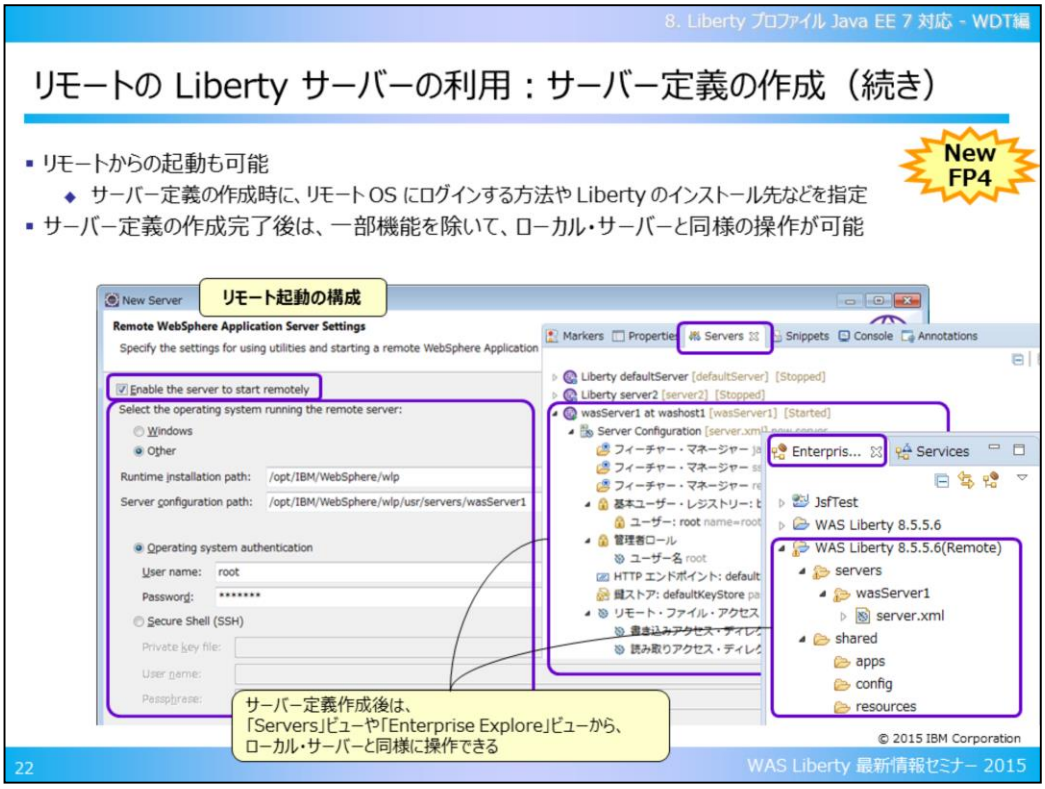
上の図は、リモートの Liberty プロファイル・サーバー用のサーバー定義を eclipse に追加する流れを示したものです。

ローカルの場合と同様に、「Servers」ビューのポップアップ・メニューから New -> Server を選択して、サーバー定義の追加を開始します。

最初の画面は、ホスト名、サーバー定義の名前、ランタイム環境などの指定で、ローカルの場合と同様です。

次の画面では、リモートの Liberty プロファイル・サーバーに接続するための情報を指定します。Verify ボタンを押し、正常に接続されると、リモートの Liberty プロファイル・サーバーの定義内容が表示されます。

[続きの操作は次のスライドに示します]

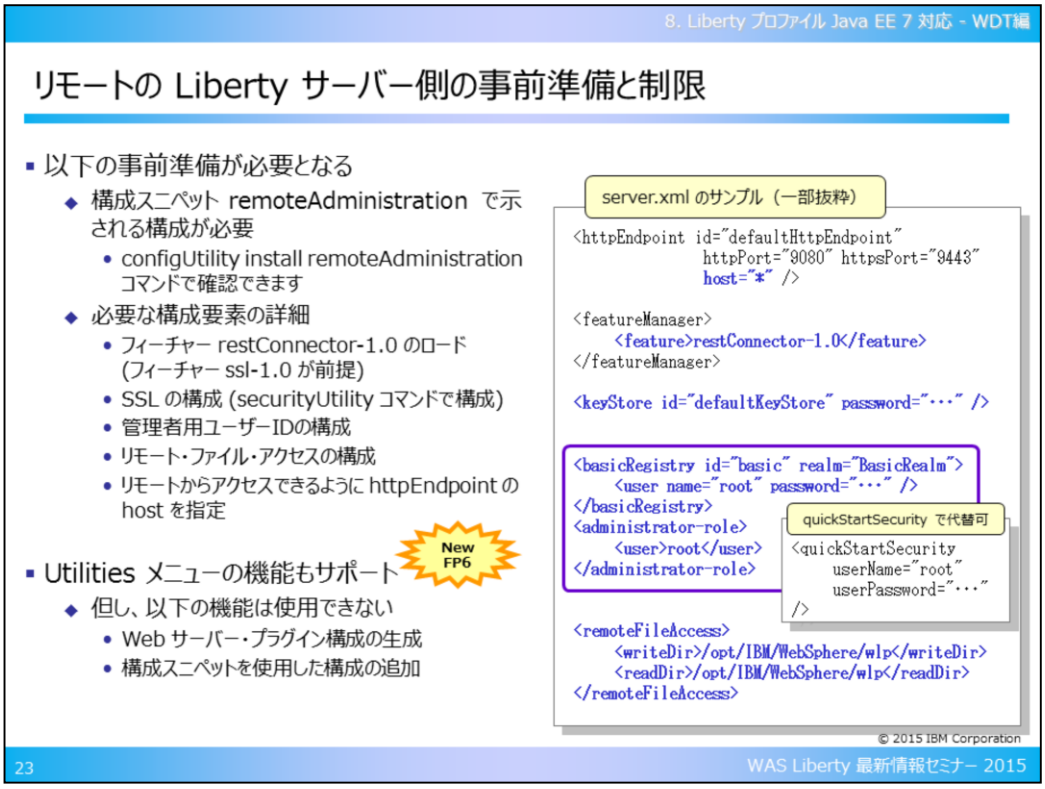


リモートの Liberty プロファイル・サーバー用のサーバー定義の続きです。

リモートの Liberty プロファイル・サーバーの起動・停止を WDT から実行したい場合は、上の図の画面で、リモート OS にログインする方法や Liberty のインストール先などを指定

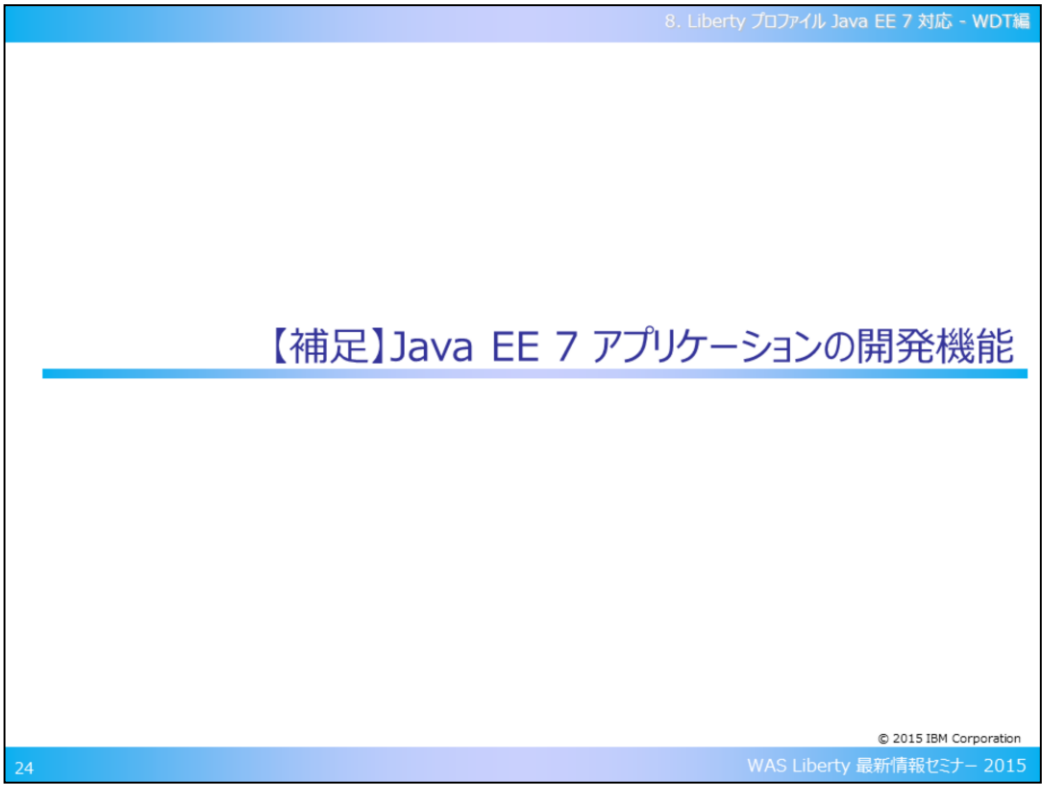
します。

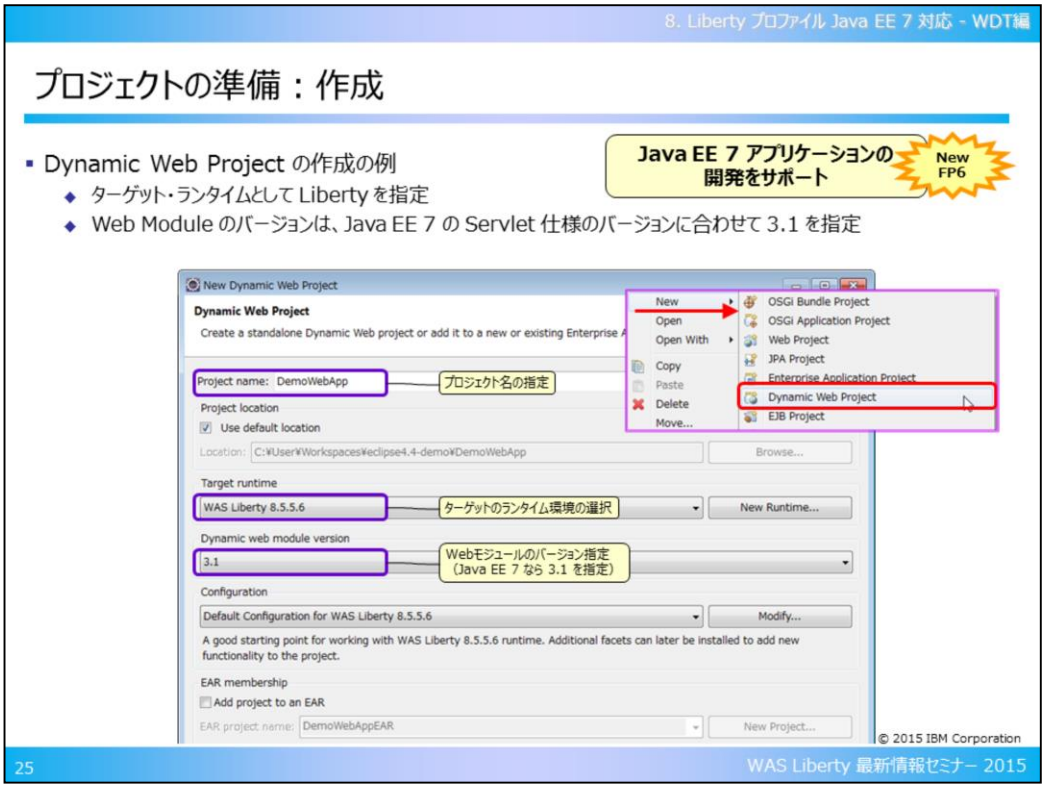
サーバー定義作成後は、「Servers」ビューや「Enterprise Explore」ビューなどから、ローカル・サーバーと同様に操作できます。



リモートの Liberty プロファイル・サーバーを WDT から使用するには、リモートの Liberty サーバーで事前準備が必要になります。

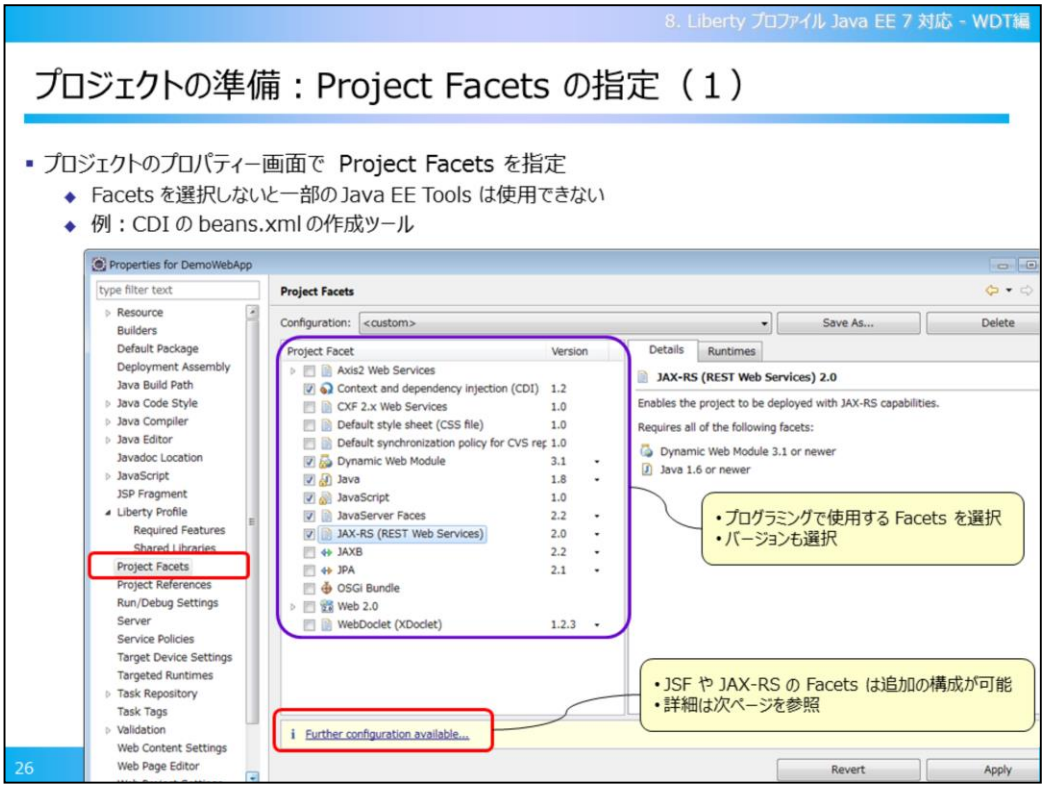
必要な事前準備の内容は上記のとおりです。


8. Liberty プロファイル Java EE 7 対応 - WDT編
【補足】Java EE 7 アプリケーションの開発機能
© 2015 IBM Corporation
24
WAS Liberty 最新情報セミナー 2015

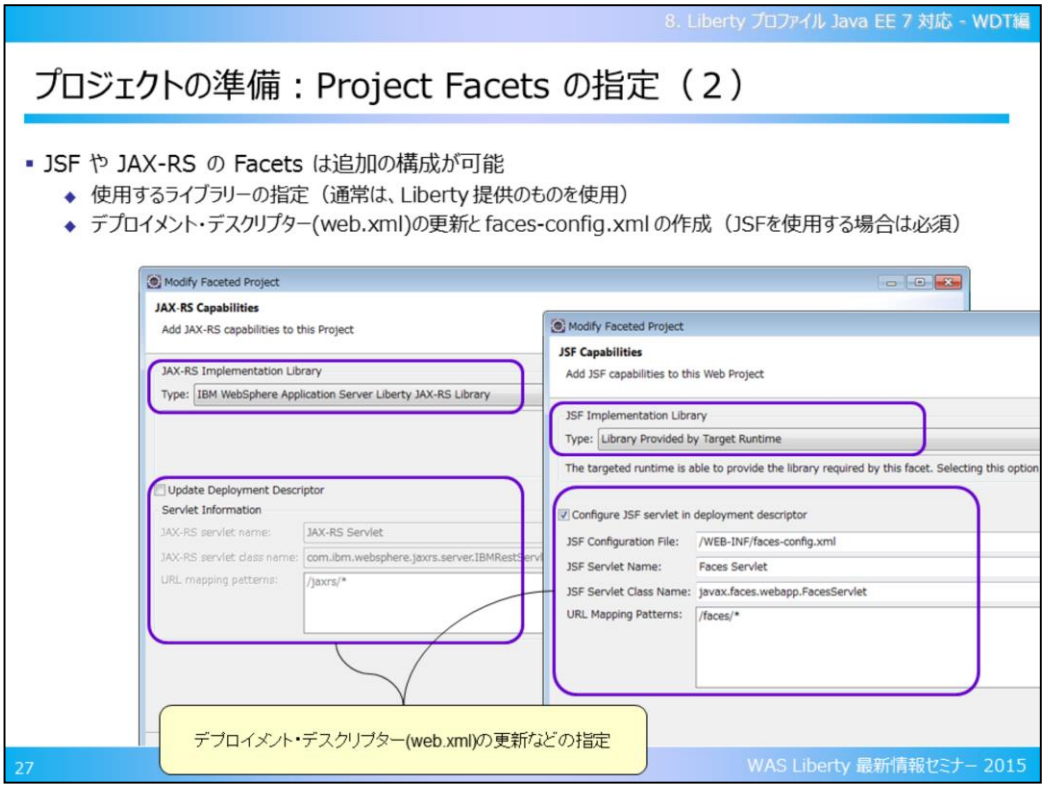


Java EE 7 用のプロジェクトの作成と設定を Dynamic Web Project を例として説明します。

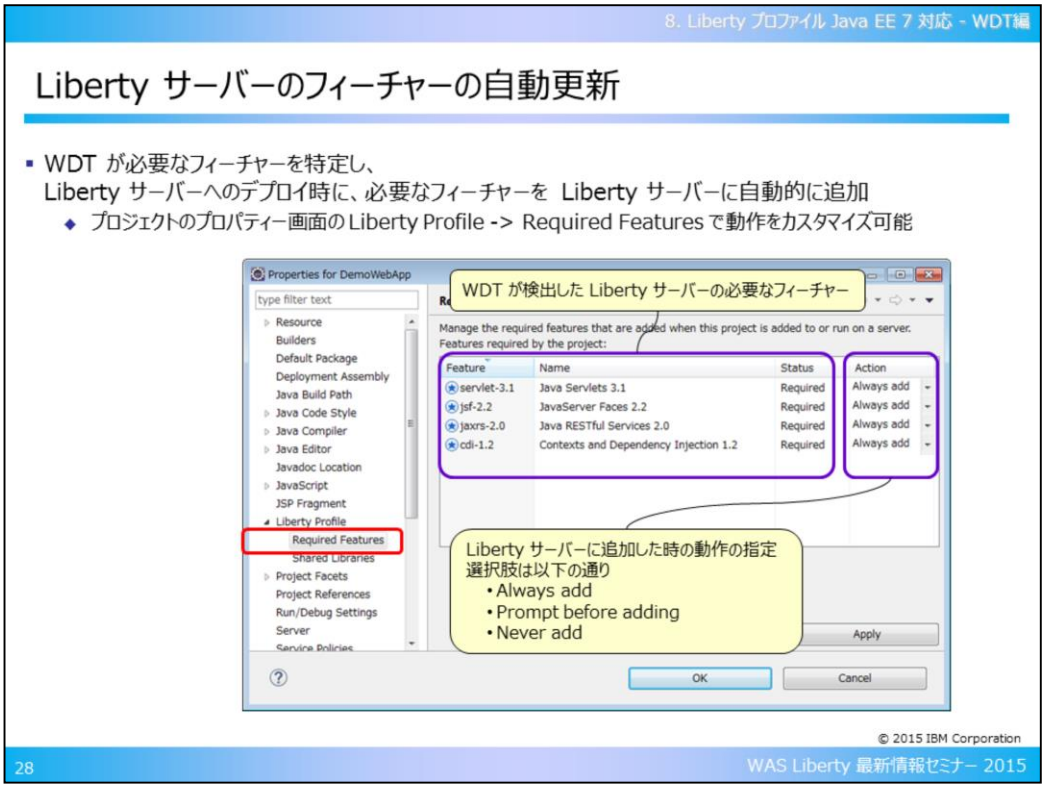
Dynamic Web Project の作成では、Liberty プロファイルのランタイム環境を選択し、Dynamic Web Module のバージョンとして 3.1 を指定します。



プロジェクトを作成したら、プロジェクトのプロパティー画面で Project Facets を指定します。
JSF や JAX-RS の Facets は追加の構成が行えます。

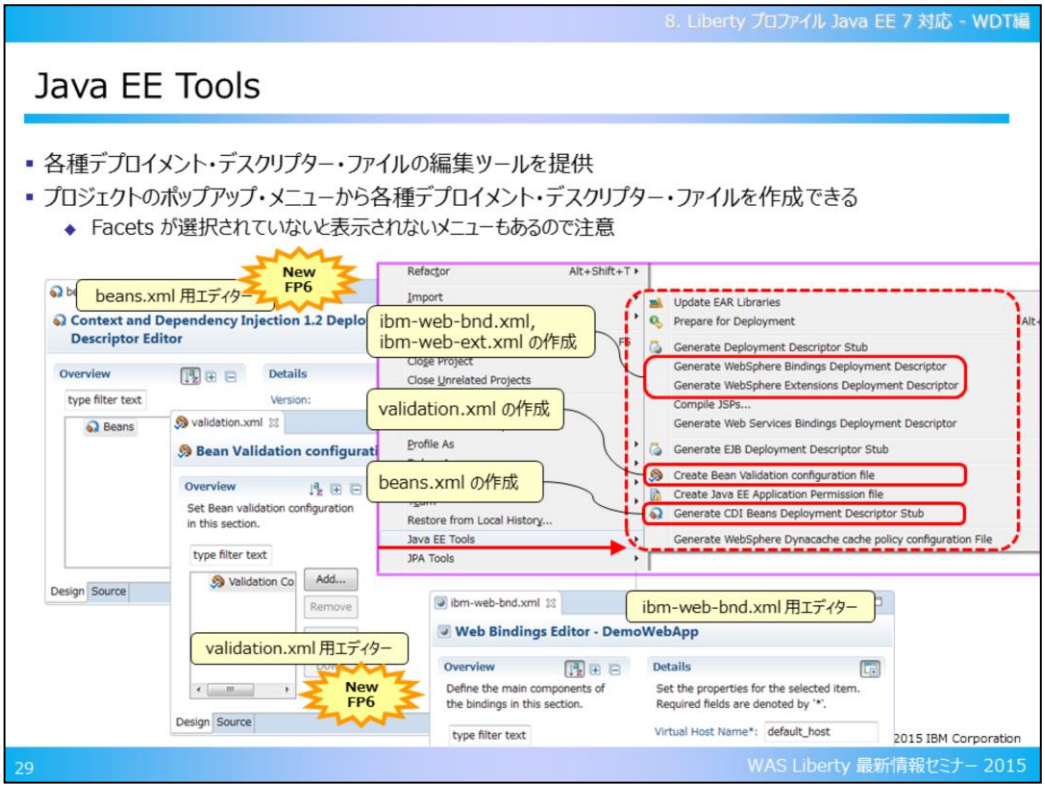


JSF や JAX-RS の Facets は追加の構成が行えます。上の図は、その内容です。



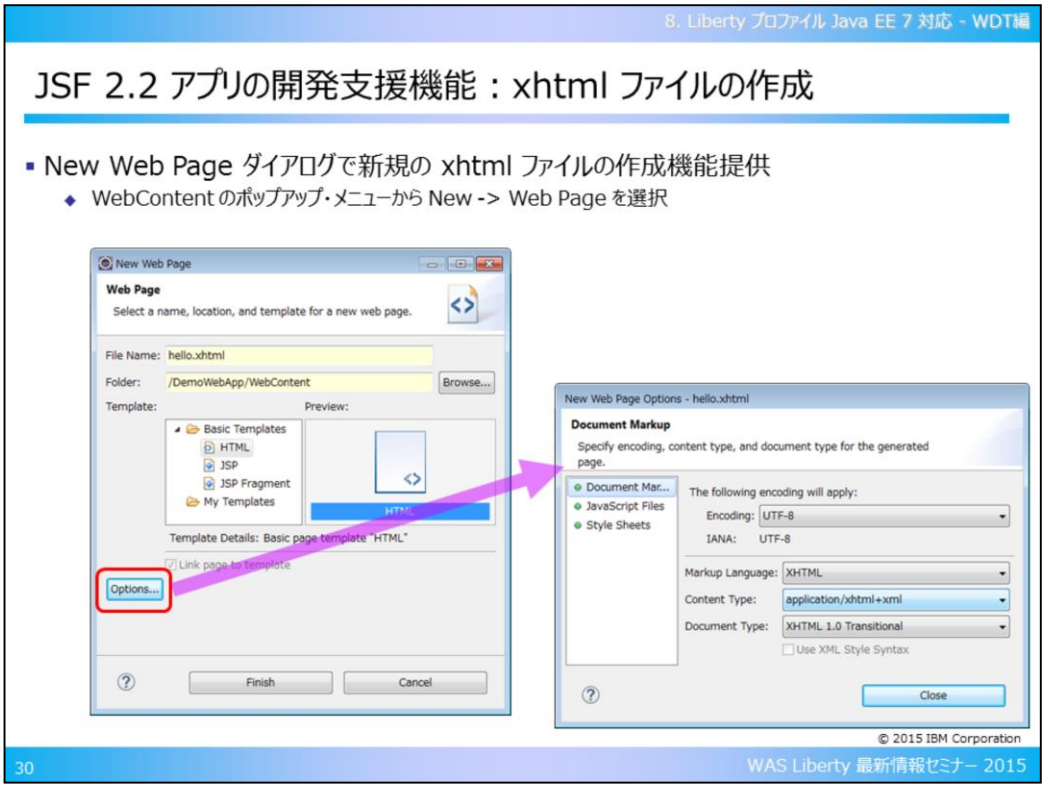
WDT は、Liberty サーバーへのデプロイ時に、必要なフィーチャーを Liberty サーバーに自動的に追加する機能を提供しています。

プロジェクトのプロパティー画面で Liberty Profile -> Required Features を選択すると、この動作をカスタマイズすることできます。

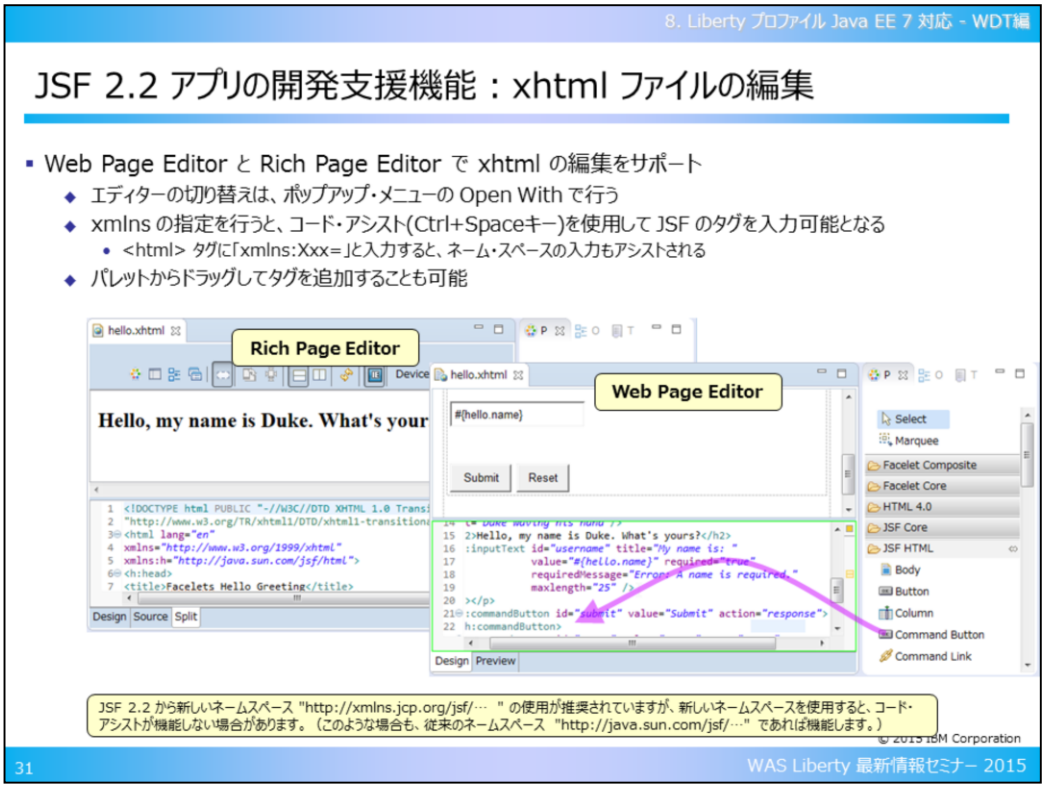


プロジェクトのポップアップ・メニューの Java EE Tools の中には、様々デプロイメント・デスクリプター・ファイルの作成機能が含まれています。

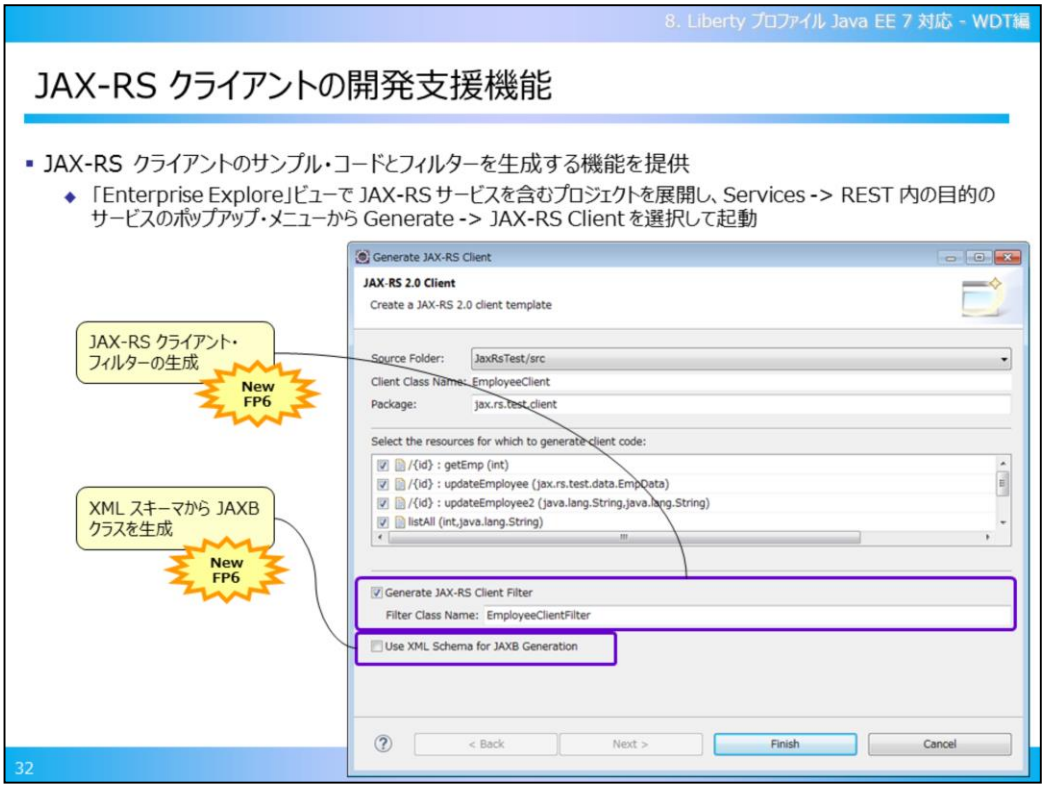
また、WDT は、CDI の beans.xml ファイルや、Bean Validation の validation.xml ファイルを編集するためのエディターも提供しています。



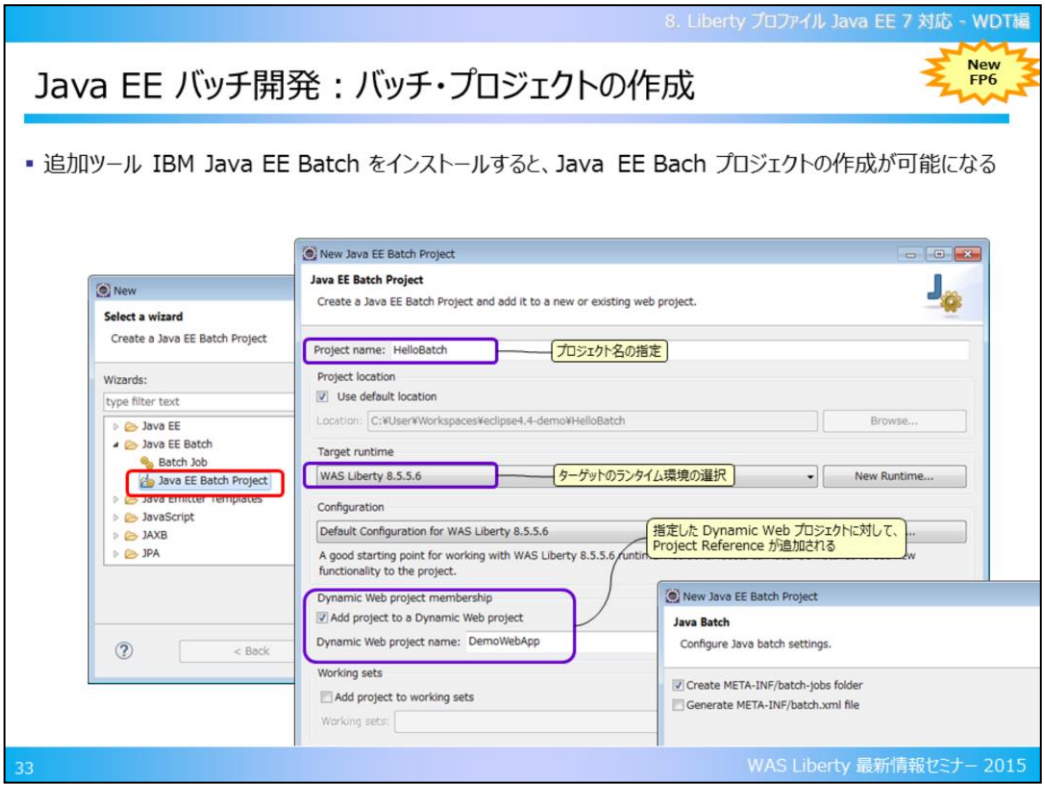
様々なタイプの Web ページを作成できるウイザードが提供されています。このウイザードを使用すると xhtml ファイルも作成できます。



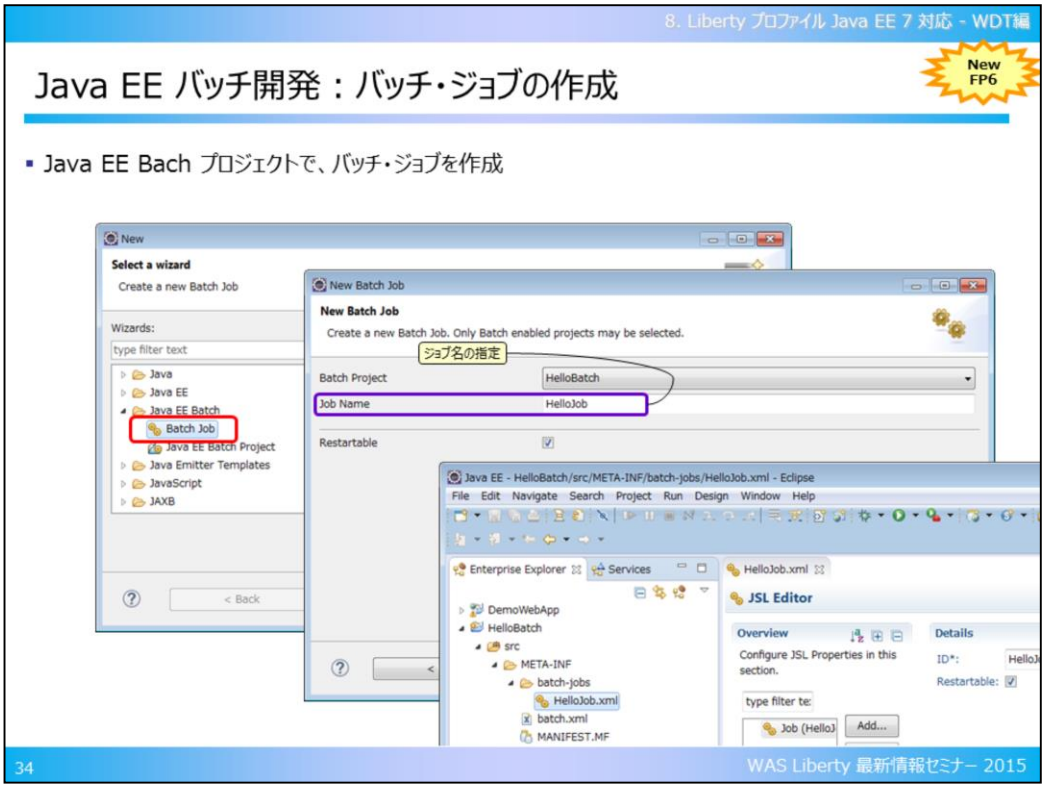
xhtml ファイルのエディターも提供されています。



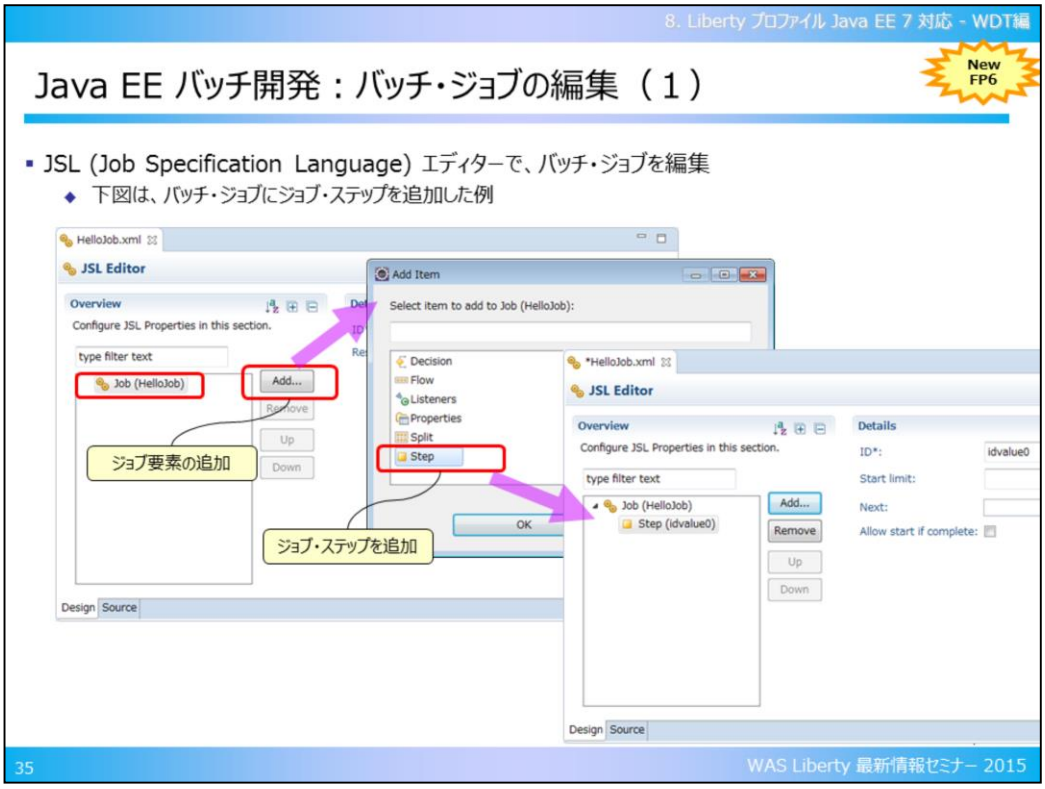
FixPack 6 では、JAX-RS クライアントのサンプル・コードとフィルターを生成する機能が拡張されました。



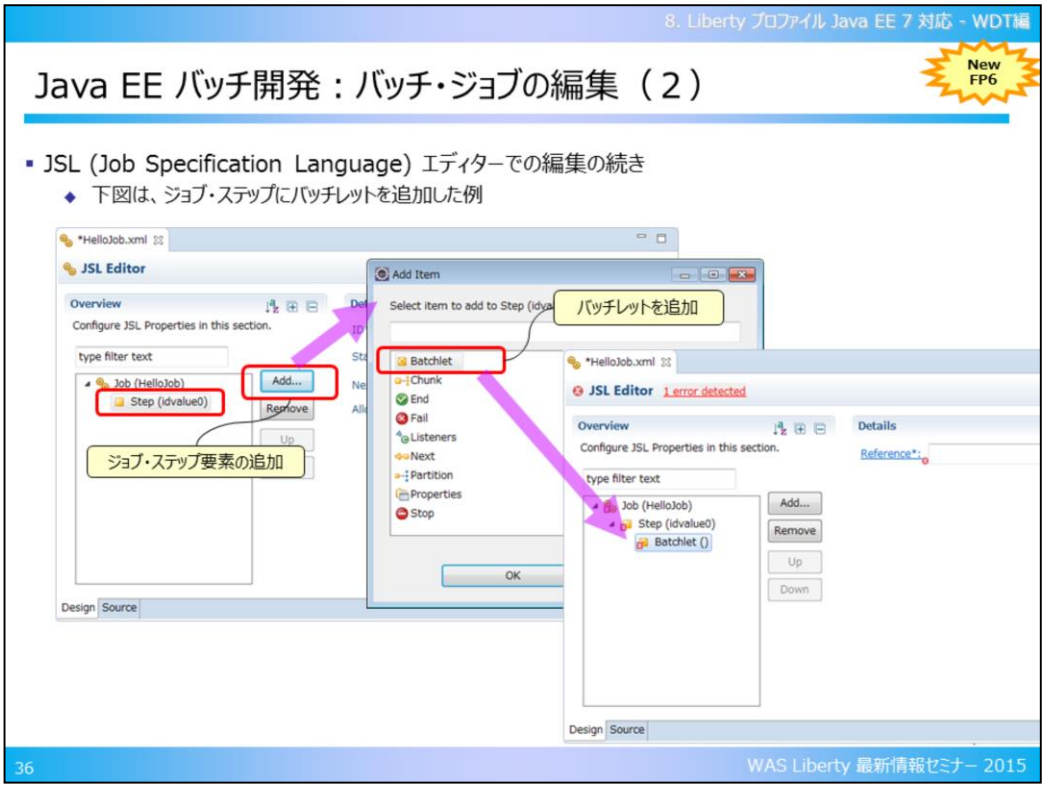
以降では、WDT を使用して Java EE バッチを開発する流れを簡単に紹介します。
最初は、Java EE Batch プロジェクトの作成です。



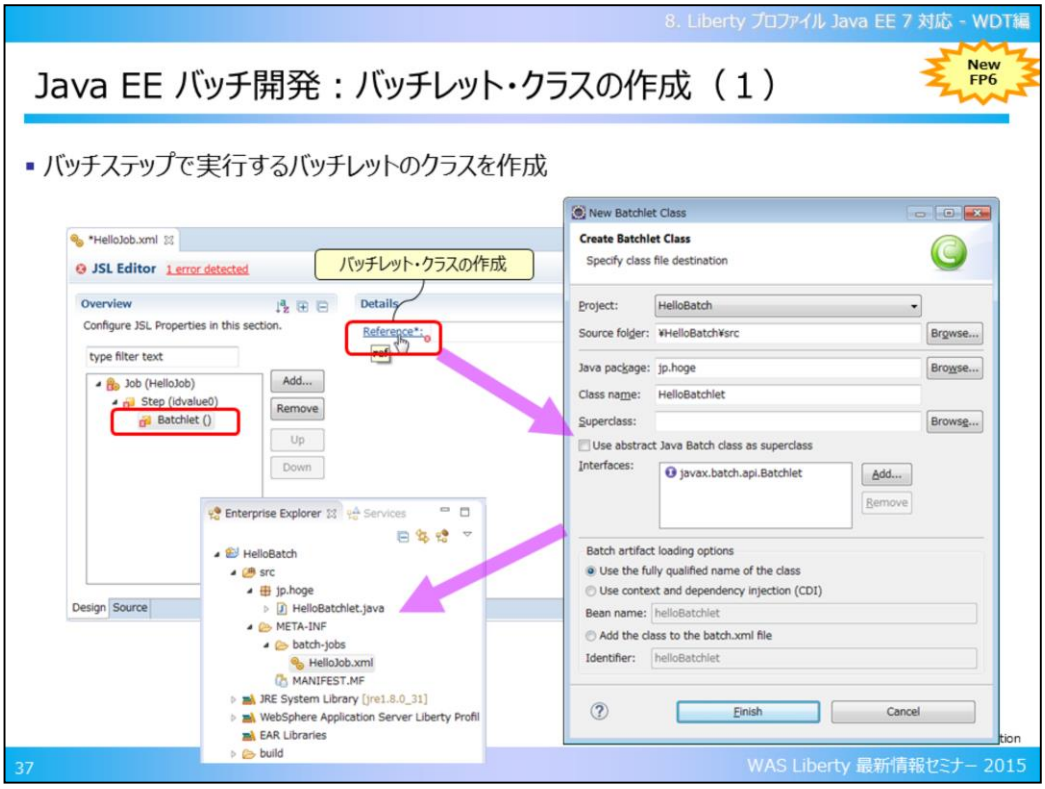
次に、バッチ・ジョブを作成します。



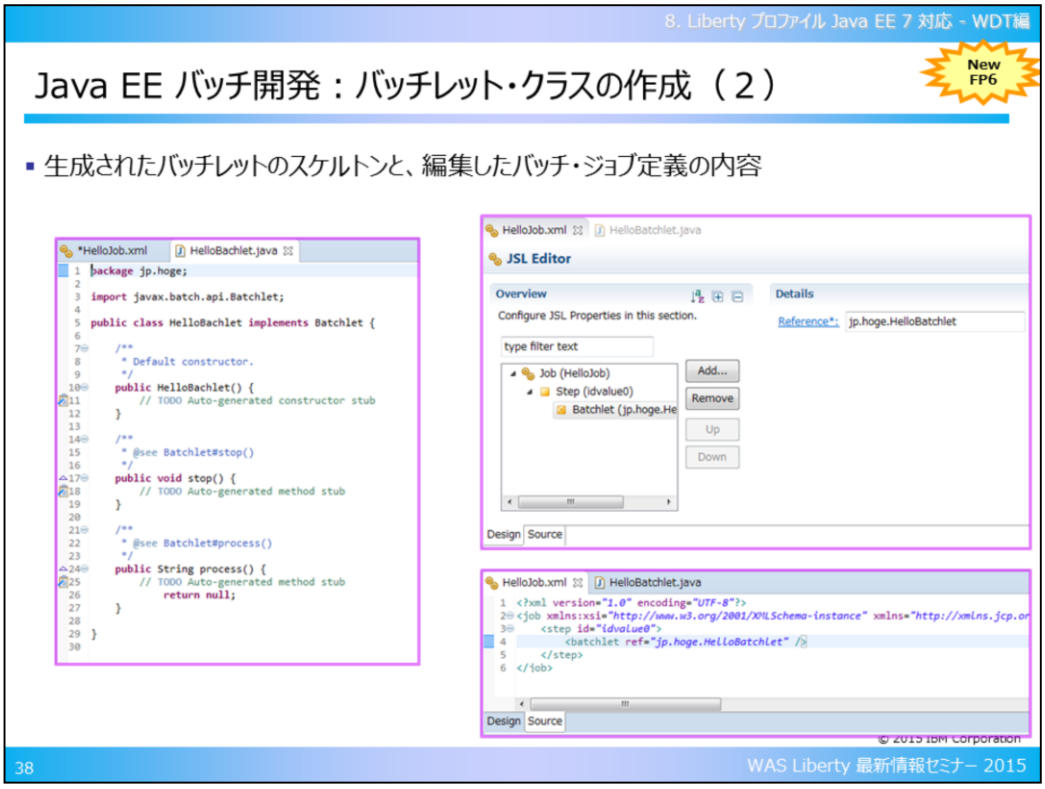
バッチ・ジョブは、JSL (Job Specification Language) エディターで編集できます。
上の図は、バッチ・ジョブにジョブ・ステップを追加する流れです。



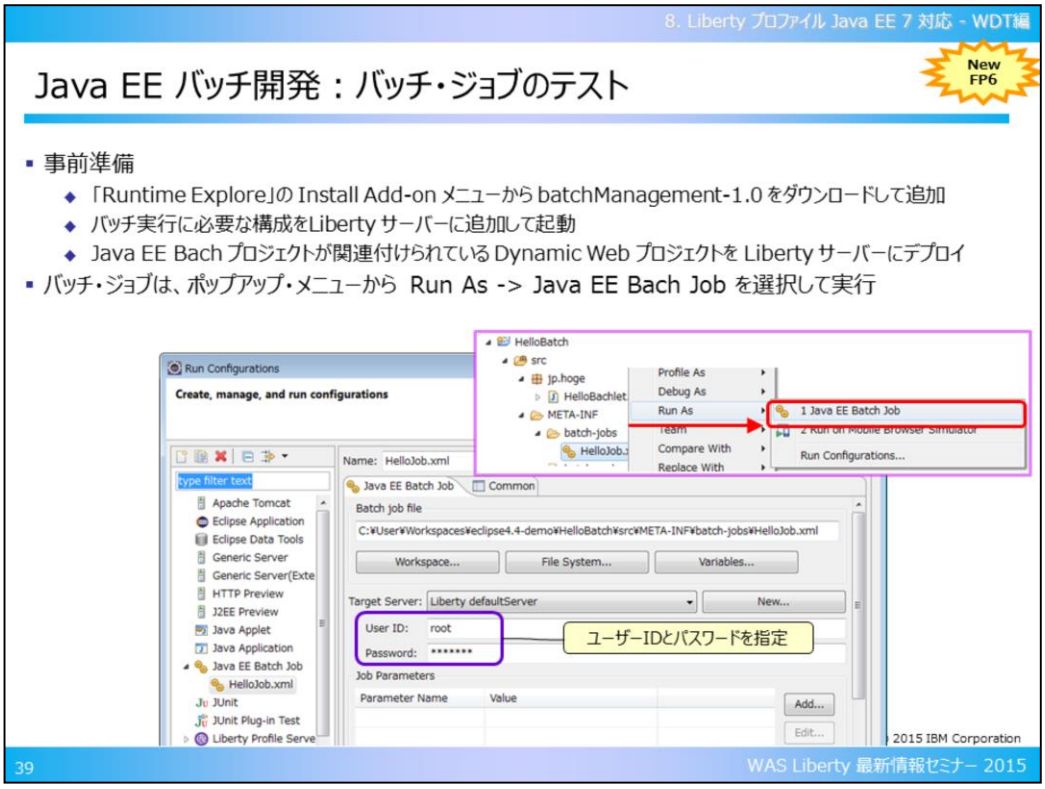
バッチ・ステップにバッチレットを追加します。



追加したバッチレットを選択し、リンクをクリックして、バッチレットのクラスを作成します。



以上の流れで作成した、バッチ・ジョブとバッチレットの内容です。この状態で、空のバッチ・ジョブとして実行できます。



バッチを実行するには、上記の準備が必要です。

バッチジョブの実行は、バッチ・ジョブのポップアップ・メニューから Run As -> Java EE Bach Job を選択して行います。表示される Run Configuration の画面では、バッチを実行するユーザーとパスワードの指定が必須です。


8. Liberty プロファイル Java EE 7 対応 - WDT編
THANK
YOU
© 2015 IBM Corporation
40
WAS Liberty 最新情報セミナー 2015