ORACLE
FUSION MIDDLEWARE

12c における Oracle WebCenter Portal のタスク・フローのカスタマイズ

Oracle ホワイト・ペーパー | 2016 年 4 月

ORACLE

免責事項

下記事項は、弊社の一般的な製品の方向性に関する概要を説明するものです。また、情報提供を唯一の目的とするものであり、いかなる契約にも組み込むことはできません。以下の事項は、マテリアルやコード、機能を提供することをコミットメント（確約）するものではないため、購買決定を行う際の判断材料になさらないでください。オラクルの製品に関して記載されている機能の開発、リリース、および時期については、弊社の裁量により決定されます。

目次

はじめに

Oracle WebCenter Portal では、タスク・フロー（ADF ライブラリ）を使って共通機能を公開します。Oracle JDeveloper Customization Developer ロールを使用すると、Oracle WebCenter Portal のタスク・フローをカスタマイズして、標準のルック・アンド・フィールと機能を拡張または変更することができます。JDeveloper Customization Developer ロールの長所は、Oracle WebCenter Portal に標準搭載されているタスク・フローのルック・アンド・フィールを、実際にコードを作成しなくても変更できることです。このドキュメントでは、JDeveloper Customization Developer ロールを使って、標準搭載されているタスク・フローのルック・アンド・フィールを変更する手順について説明します。

Oracle WebCenterタスク・フローのカスタマイズ・プロセスについて

Oracle WebCenter Portal にタスク・フローのカスタマイズを適用すると、タスク・フローの全インスタンスにそのカスタマイズが適用されます。したがって、タスク・フローを使用する各ページにこれらのカスタマイズを繰り返す必要がありません。

Oracle WebCenter Portal タスク・フローのカスタマイズでは、タスク・フロー用のシード済みカスタマイズを Oracle ADF Fusion テクノロジー・アプリケーションで作成し、JDeveloper を使って、タスク・フロー・カスタマイズを JAR アーカイブにパッケージ化します。Oracle WebCenter Portal インスタンスに接続された MDS リポジトリにシード済みカスタマイズを後でインポートします。

カスタマイズ可能なOracle WebCenter Portalアプリケーションの準備

図 1 に示すように、デフォルト・ロールで Oracle ADF Fusion アプリケーションを作成します。

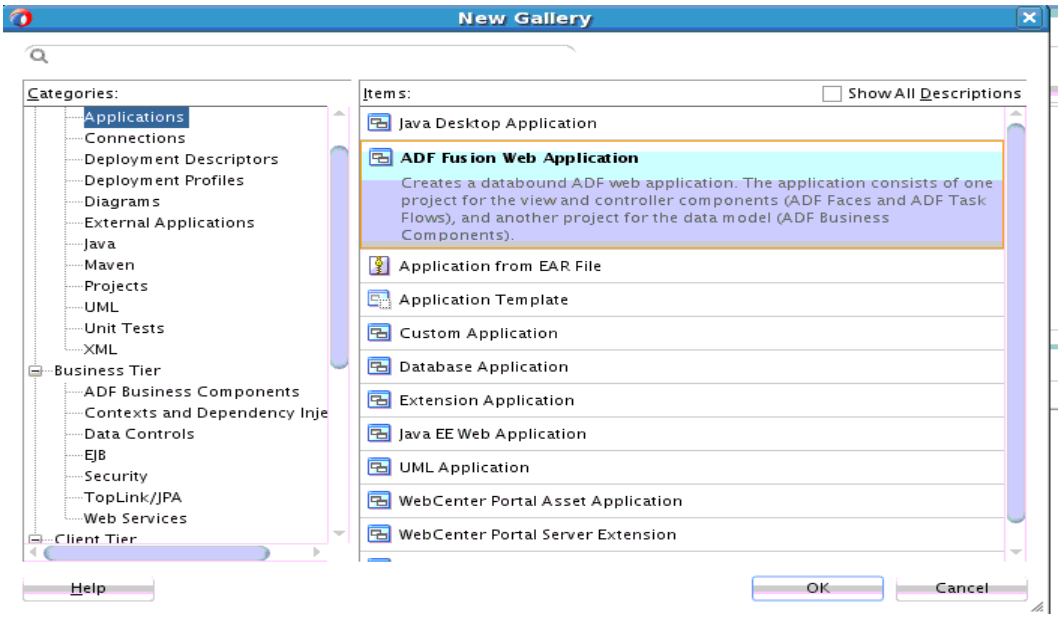

図1：Oracle ADF Fusionアプリケーションの作成

カスタマイズ可能なクラスの作成

カスタマイズ・クラスは、基本定義メタデータにどのカスタマイズを適用するかを定義する際にMDS によって使用されるインタフェースです。各カスタマイズ・クラスはカスタマイズ・レイヤー（`site` や `user` など）を定義し、複数のレイヤー値を含めることができます。タスク・フローのカスタマイズの場合、 `SiteCC` レイヤーを使用します。アプリケーションで使用されるカスタマイズ・クラスは、アプリケーションのカスタマイズ時に JDeveloper で使用可能になります。

このホワイト・ペーパーでは、サンプル・アプリケーションで CC クラスを作成するアプローチを利用します。後で、CC クラスを JAR にパッケージ化して、JDeveloper のクラスパス下に配置します。

カスタマイズ・クラスを作成するには、次の手順を実行します。

1. 新たに作成した Oracle ADF Fusion アプリケーションのアプリケーション ViewController プロジェクトで、`WCSiteCC` という名前の新しい java クラスを追加して、"`oracle.mds.cust.CustomizationClass`"を拡張します（図 2 を参照）。

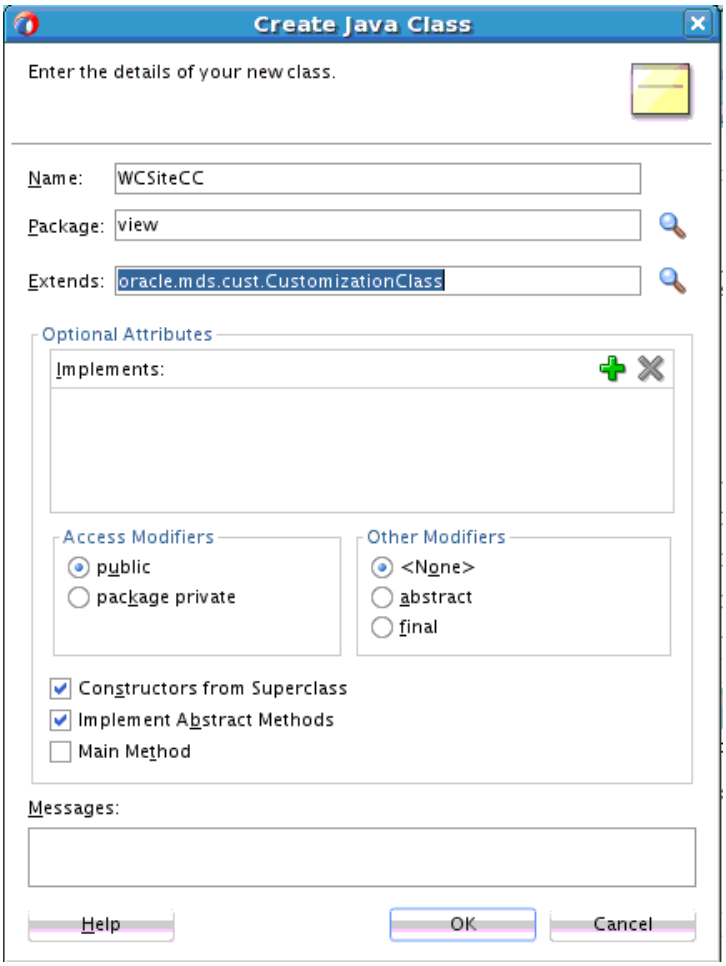

図2：カスタマイズ可能なクラスの作成

2. WCSiteCC.java にコードを追加します。以下は、WCSiteCC クラスのサンプル・コードです。

```
package view;

import java.io.IOException;
import java.io.Input
Stream;

import java.util.Properties;

import oracle.mds.core.MetadataObject;
import oracle.mds.core.RestrictedSession;
import oracle.mds.cust.CacheHint;
import oracle.mds.cust.CustomizationClass;

public class WCSiteCC extends CustomizationClass {
	public WCSiteCC() {
super();
	}

	@Override
	public CacheHint getCacheHint() {
// TODO Implement this method
return null;
	}

	@Override
	public String getName() {
// TODO Implement this method
return "site";
	}


	@Override
	public String[] getValue(RestrictedSession mdsSession, MetadataObject mo) {
// TODO Implement this method
return new String[]{"webcenter"};
	}
}
```

3. ViewController プロジェクトをリビルドします。

カスタマイズ・クラスをJDeveloperで使用可能にする

カスタマイズ・クラスを作成したら JDeveloper で使用可能にして、カスタマイズの実装時に使えるようにする必要があります。Customization Developer ロールで作業を行う際には、カスタマイズ・クラスがプロジェクトのクラスパス上で使用可能になっている必要があります。

カスタマイズ・クラスを使用可能にするには、次の手順を実行します。

1. JDeveloper の Application ウィンドウにある「ViewController」プロジェクトを右クリックして、「**Project Properties**」を選択します。
2. Project Properties ダイアログで「**Deployment**」を選択し、「**New Profile**」をクリックします。
3. Create Deployment Profile ダイアログ（図 3）の **Profile Type** ドロップダウン・リストから「JAR File」を選択します。
4. Deployment Profile Name に、`myCC` などデプロイメント・プロファイル名を入力し、「**OK**」をクリックします。

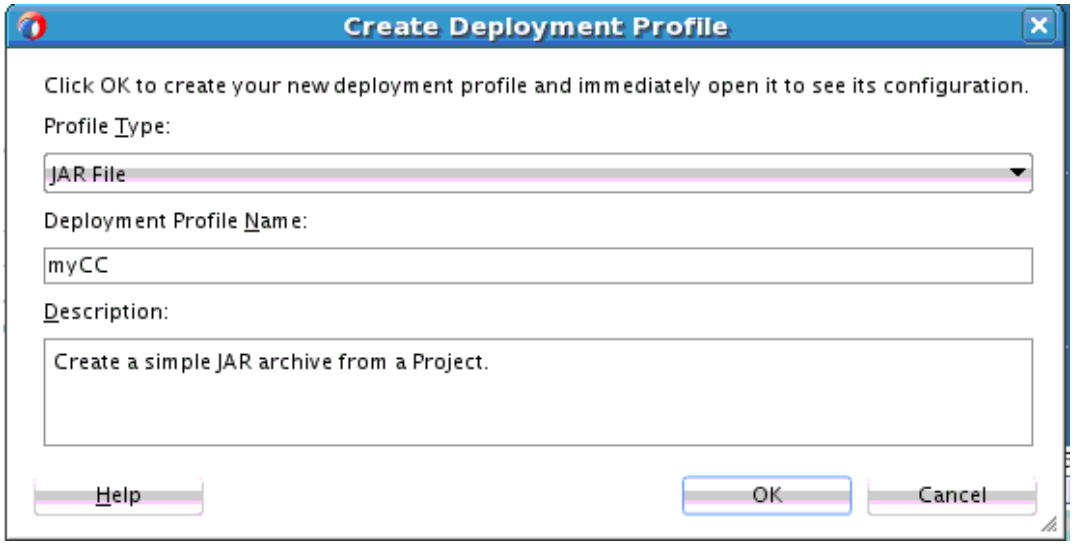

図3：デプロイメント・プロファイルの作成

5. デプロイメント・プロファイルを編集して `WCSiteCC` クラスを追加します（図 4）。

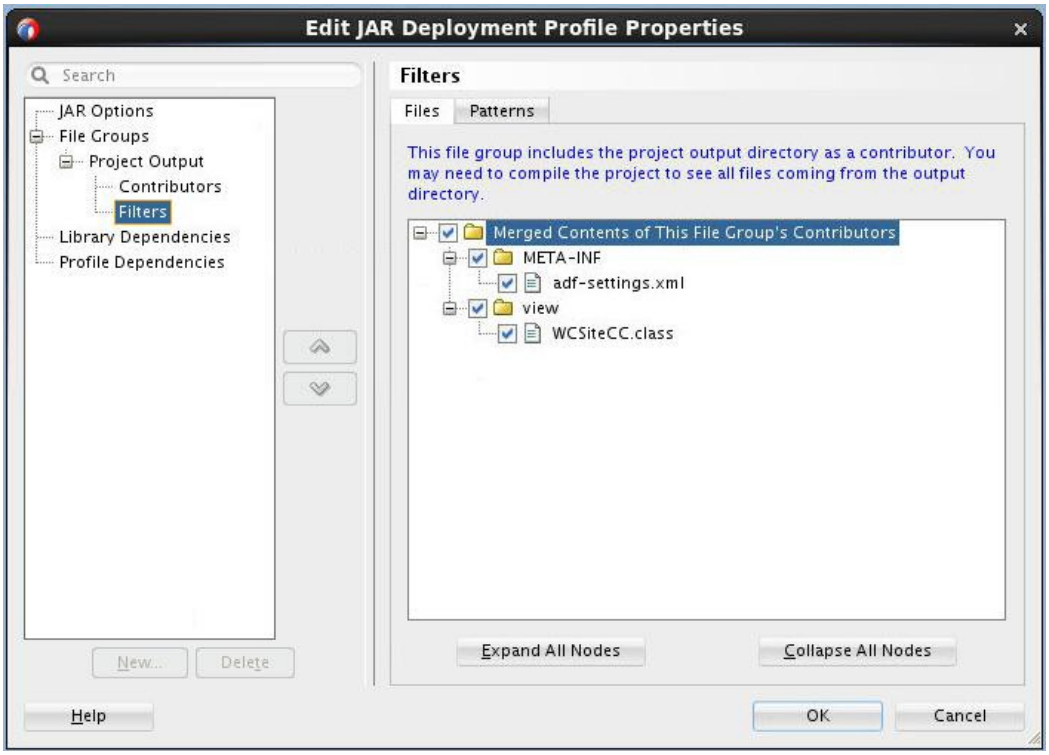

図4：JARデプロイメント・プロファイルのプロパティの編集

6. 「**OK**」をクリックして、プロファイルを保存します。
7. 「**OK**」をクリックして、プロジェクト・プロパティを閉じます。
8. File メニューで「**Save All**」をクリックします。
9. JDeveloper の Application ウィンドウにある「ViewController」プロジェクトを右クリックして、「**Deploy**」を選択し、`myCC` など新しく作成したデプロイメント・プロファイルを選択します。
10. Deploy ダイアログで「Deploy to JAR File」を選択して、「**Finish**」をクリックします。

    これにより、jar プロファイルが
    `<Application>/ViewController/deploy/myCC.jar` 下に展開されます。
    `<Application>`はアプリケーションの名前です。
11. JDeveloper の Application ウィンドウにある「ViewController」プロジェクトを右クリックして、「**Project Properties**」を選択します。
12. Project Properties ダイアログで「**Libraries and Classpath**」→「**Add Jar/Directory**」の順に選択し、`myCC.jar` を追加して、コントローラのクラスパスを表示します。

Viewプロジェクトのシード済みカスタマイズの有効化

View プロジェクトのシード済みカスタマイズを有効にするには、次の手順を実行します。

1. JDeveloper の Application ウィンドウにある「ViewController」プロジェクトを右クリックして、「Project Properties」を選択します。
2. Project Properties ダイアログで「ADF View」を選択します。
3. 「Enable Seeded Customizations」チェック・ボックスを選択します（図 5）。

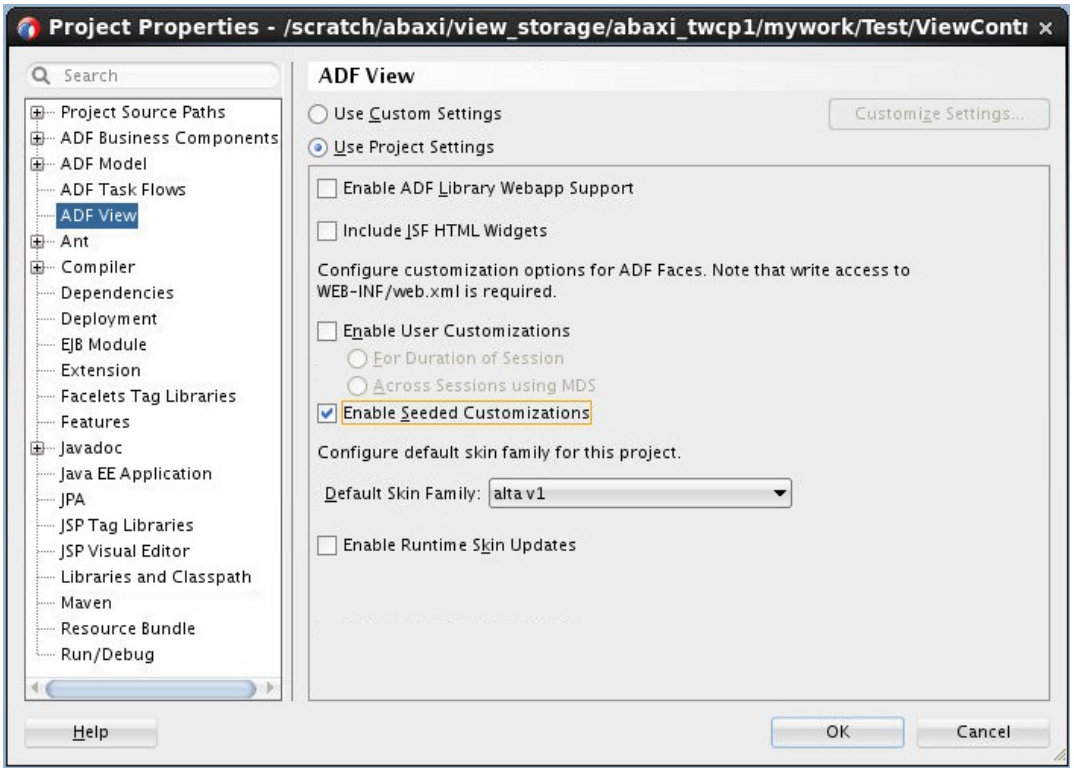

図5：Viewプロジェクトのシード済みカスタマイズの有効化

4. 「OK」をクリックします。
5. File メニューで「Save All」をクリックします。

JDeveloper用の設計時カスタマイズ・レイヤーの構成

Oracle WebCenter Portal のタスク・フローをカスタマイズするには、JDeveloper Customization ロールで使用可能な CC レイヤー値を構成します。CC レイヤー値は JDEV_HOME/jdev/CustomizationLayerValues.xml で構成できます。

WC タスク・フローのカスタマイズでは、`Site` レイヤーを値 `WebCenter` とともに使用しています。たとえば、次のようになります。

```
<cust-layers xmlns="http://xmlns.oracle.com/mds/dt">
    <cust-layer name="site" id-prefix="s">
        <cust-layer-value value="webcenter" display-name="WebCenter"/>
    </cust-layer>
</cust-layers>
```

上記の設定を基に、JDeveloper Customization ロールはレイヤー値を レイヤーと見なします。

注：サイト・レイヤー値は大/小文字が区別されます。

adf-config.xmlファイルの構成

アプリケーションの `adf-config.xml` ファイルの `mds-config` セクションには、適切な `cust-config` 要素が指定されている必要があります。`cust-config` 要素により、カスタマイズ・クラスを順番に並べた名前付きリストを定義できます。`adf-config.xml` ファイルに対して概要エディタを使って、カスタマイズ・クラスを追加できます。

カスタマイズ・クラスを追加するには、次の手順を実行します。

1. アプリケーションの adf-config.xml ファイルを概要エディタで開きます。
2. 概要エディタで「MDS」を選択します。
3. WCSiteCC クラスを追加して、MDS カスタマイズ構成を生成します（図 6 を参照）。

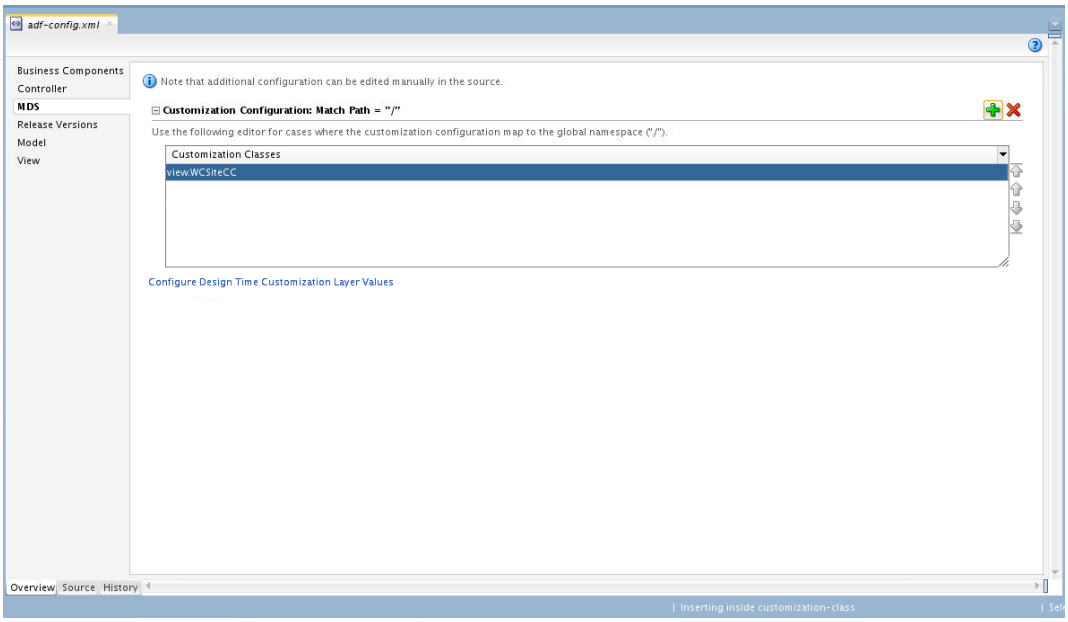

図6：カスタマイズ・クラスの追加

4. File メニューで「**Save All**」をクリックします。

以下は、adf-config.xml ファイルのカスタマイズ・クラスの順序を示す例です。

```
<adf-config xmlns="http://xmlns.oracle.com/adf/config">
  <adf-mds-config xmlns="http://xmlns.oracle.com/adf/mds/config">
    <mds-config xmlns="http://xmlns.oracle.com/mds/config"
version="11.1.1.000">
      <cust-config>
        <match path="/">
          <customization-class name="view.WCSiteCC"/>
        </match>
      </cust-config>
    </mds-config>
  </adf-mds-config>
</adf-config>
```

Oracle WebCenter Portalのタスク・フローのカスタマイズ

WebCenter Portal のタスク・フローにはすべて、明確に定義された UI と動作、つまり、ビジネス・ロジックが標準で備わっています。お客様の展開環境によっては、タスク・フローに標準で備わっているルック・アンド・フィールや動作が合わないことがあります。ポータル・サーバーのルック・アンド・フィールを変更したい場合や、既存の Oracle WebCenter Portal のタスク・フローのいずれかに独自のタスク・フローを追加したい場合、または、ツールバーの一部のボタンを排除するか独自のボタンを追加して、Oracle WebCenter Portal のタスク・フローの UI を変更したい場合があります。これらの変更には、Oracle WebCenter Portal のタスク・フローの JAR を開く必要も、Oracle WebCenter Portal の標準コードを変更する必要もありません。

JDeveloper Customization ロールは強力なメカニズムで、これを使うと、基本ライブラリ JAR のコードを変更しなくても、ADF Library をカスタマイズできます。Oracle WebCenter Portal と ADF レイヤーは MDS 上に構築されているため、Oracle WebCenter Portal のタスク・フローを拡張する場合は、JDeveloper Customization ロールを利用できます。Oracle WebCenter Portal のタスク・フローはすべて ADF Library としパッケージ化されるため、JDeveloper 設計時にタスク・フローをカスタマイズできます。

セットアップが完了したら（「JDeveloper 用の設計時カスタマイズ・レイヤーの構成」を参照）、Oracle WebCenter Portal のタスク・フローのカスタマイズを開始できます。

Oracle WebCenter Portalのタスク・フローのセットアップ

このセットアップを JDeveloper で 1 回実行して、Oracle WebCenter Portal のタスク・フローのカスタマイズを可能にするワークスペースをセットアップします。上記にあるタスク・フローのユースケースのいずれかに取り組む前に、この手順を必ず実行してください。

Oracle WebCenter Portal のタスク・フロー用のワークスペースをセットアップするには、次の手順を実行します。

1. Oracle WebCenter Portal のタスク・フローをカスタマイズするには、Oracle WebCenter Portal のタスク・フロー・ビュー、JAR をプロジェクトに含める必要があります。JDeveloper Studio を"Default Role"で実行します。
2. アプリケーション・ウィンドウの Projects パレットで「**Libraries & Classpath**」を選択します。
3. 「**Add Jar / Directory**」をクリックします。
4. Oracle WebCenter Portal のタスク・フロー・ディレクトリにナビゲートして、タスク・フロー・ビュー、JAR を選択します。

   Oracle WebCenter Portal のタスク・フローのビュー、JAR は、インストール済みの JDeveloper の次の場所に格納されています。
   `JDEV_HOME/jdeveloper/webcenter/modules/oracle.webcenter.framework`
5. プロジェクト・プロパティで「**Select**」→「**OK**」の順にクリックします。
6. **File** メニューで「**Save All**」をクリックします。
7. アプリケーションの Projects パレットで「**Navigate Display Options**」アイコンを選択し、「**Show Libraries**」（図 7）をクリックします。

   ナビゲータには、ViewController プロジェクトに含まれているすべてのライブラリが表示されます。

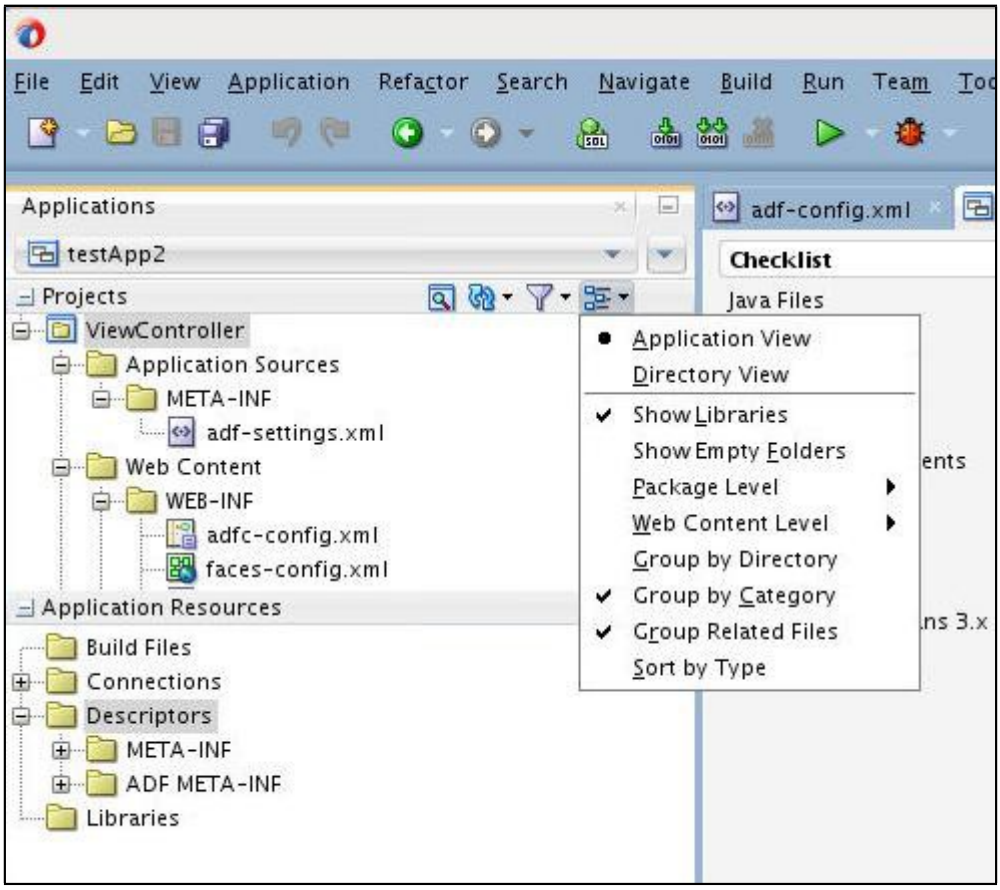

図7：アプリケーションのProjectsパレット

8. 「**Tools**」→「**Switch Roles**」→「**Customization Developer**」の順に選択して、JDeveloper を Customization ロールで起動します。

9. JDeveloper Customization ロールで、`site` レイヤー がレイヤー値として `WebCenter` とともに Customization Context ウィンドウ内で選択されていることを確認します（図 8）。

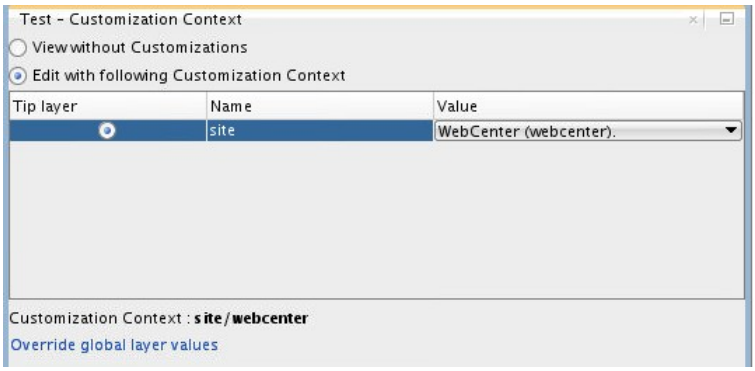

図8：Customization Context

10. Oracle WebCenter Portal のタスク・フローから `jsff` または `jspx` を選択して、カスタマイズを開始します。図 9 に、Discussion Forum Service で `createTopic.jeff` をカスタマイズする例を示します。

    カスタマイズはすべて、以下のディレクトリに保存されます。
    `Application Directory/ViewController/libraryCustomizations`

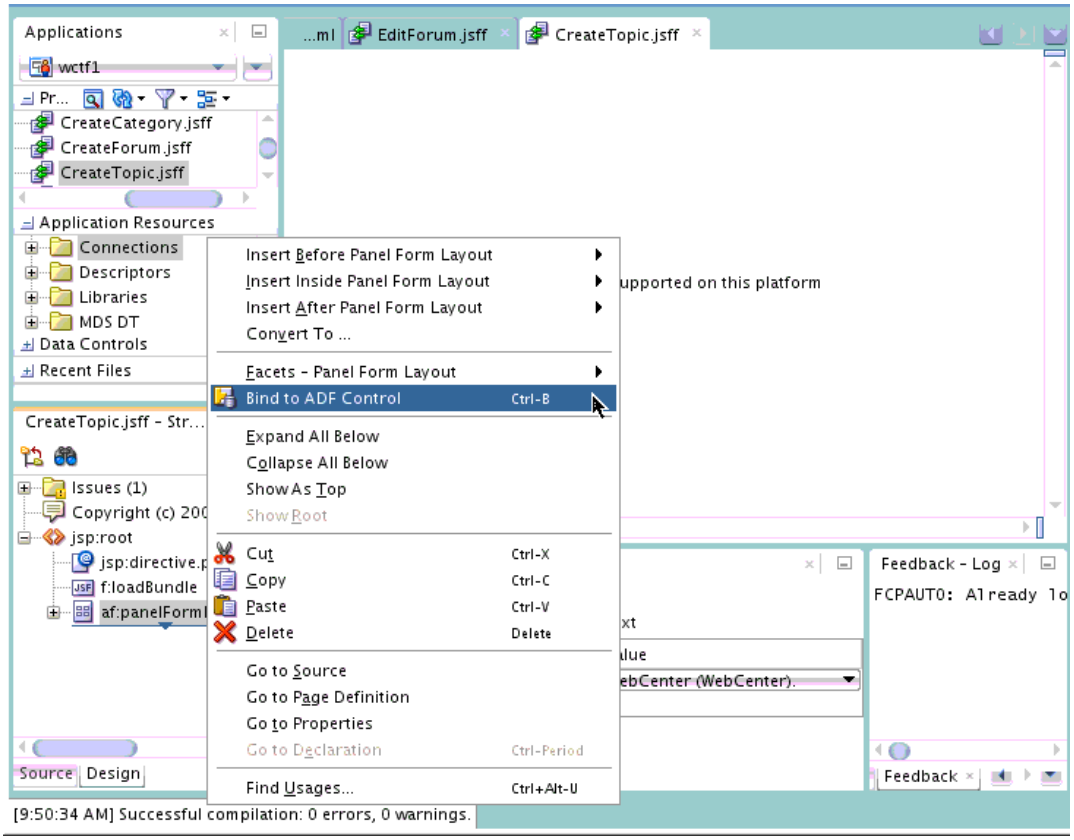

図9：Discussion Forum ServiceでのcreateTopic.jsffのカスタマイズ例

カスタマイズの適用

タスク・フローのカスタマイズの出力は、MDS カスタマイズとして生成されます。カスタマイズは`.xml.xml`、`.jspx.xml`、または `jsff.xml` ファイルとしてプロジェクト内に表示されます。これらのカスタマイズ・ドキュメントは基本的に、付属の基本ドキュメント（実行時のカスタマイズ動作を示す）の上で MDS にデルタを適用させるための命令です。

展開済みのOracle WebCenter Portalアプリケーションへのカスタマイズの適用

「Oracle WebCenter Portal のタスク・フローのカスタマイズ」で行ったカスタマイズを表示するには、これらのカスタマイズを展開済みの Oracle WebCenter Portal インスタンスの MDS リポジトリに転送する必要があります。この手順では、実行時の Oracle WebCenter Portal メタデータ・リポジトリが更新されるため、この手順を実行する前に MDS スキーマをバックアップすることをお勧めします。カスタマイズに不備があって、タスク・フローの動作に問題が生じる場合は、元の状態にリストアできます。

展開済みのポータル・アプリケーションにカスタマイズを適用するには、次の手順を実行します。

1. デフォルト・ユーザーとして、JDeveloper でアプリケーションをリビルドします。
2. JAR プロファイルを作成し、ライブラリのカスタマイズを JAR アーカイブの ViewController プロジェクト下にパッケージ化します。カスタマイズを Jar に入れるには、図 10 に示すように、`ViewController/libraryCustomizations` のディレクトリを Edit JAR Deployment Profile ダイアログの Contributors に追加します。

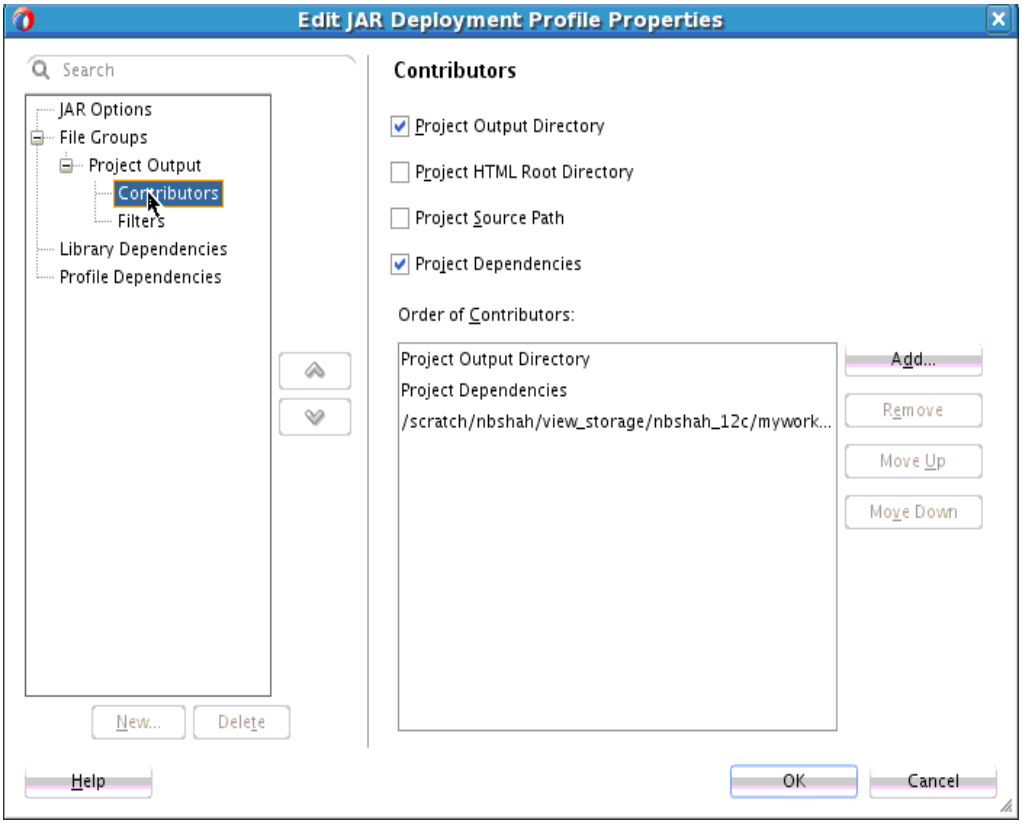

図10：Edit JAR Deployment Profile Properties

3. 新しいデプロイメント・プロファイルを実行して、カスタマイズを含んだ生成済みの JAR を Oracle WebCenter Portal 展開先マシンに配置します。Jar のコンテンツを、`/tmp/wc-cust` などマシン上の一時ディレクトリに展開します。
4. `importMetadata` WLST コマンドを使って、これらのタスク・フローのカスタマイズを Oracle WebCenter Portal アプリケーションの MDS リポジトリにインポートします。

以下は、カスタマイズをインポートするためのコマンド例です。

```
wls:/weblogic/serverConfig>importMetadata(application='webcenter',
server='WC_Portal',
fromLocation='/tmp/wc-cust', docs="/**")
```

`importMetadata` WLST コマンドの詳細については、『WLST Command Reference for Infrastructure Components』を参照してください。

カスタマイズの削除

「カスタマイズの適用」で適用したカスタマイズは、必要に応じて削除できます。いったん削除すると、Oracle WebCenter Portal のタスク・フローの動作やルック・アンド・フィールは元の出荷時の製品に戻ります。

`deleteMetadata` WLST コマンドを使って、適用済みのカスタマイズを削除します。

注：`deleteMetadata WLST` コマンドの実行には注意してください。このコマンドを誤って使用すると、メタデータ・ドキュメントが失われる可能性があります。

以下は、`deleteMetadata` WLST コマンドの例です。

```
deleteMetadata(application, server, docs, [restrictCustTo], [excludeAllCust],
[excludeBaseDocs], [excludeExtendedMetadata], [cancelOnException],
[applicationVersion])
```

`deleteMetadata` WLST コマンドの詳細については、『WLST Command Reference for Infrastructure Components』を参照してください。

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

海外からのお問い合わせ窓口
電話：+1.650.506.7000
ファクシミリ：+1.650.506.7200

CONNECT WITH US

blogs.oracle.com/oracle
facebook.com/oracle
twitter.com/oracle
oracle.com

Hardware and Software, Engineered to Work Together

ホワイト・ペーパー・タイトル 2016 年 4 月
著者：Nitin Shah
共著者：Amit Baxi