BEA WebLogic Portal

移行ガイド

リリース 7.0 サービス パック 1
マニュアルの日付 : 2002 年 9 月

著作権

限定的権利条項

商標または登録商標

移行ガイド

パート番号	マニュアルの日付	ソフトウェアのバージョン
なし	2002 年 9 月	7.0 サービス パック 1

目次

付録 A Migration Viewer ツールによるコードへの変更の参照

付録 B 移行ファイル

索引

まえがき

『移行ガイド』へようこそ。このガイドは、リリース 4.0 から 7.0 への移行ステップを説明します。

以下のリソースも活用されることをお勧めします。

オンライン マニュアルの参照　BEA 製品マニュアルは、BEA 社の Web サイトで公開しています。BEA Home ページで [製品のドキュメント] リンクをクリックするか、「e-docs」製品ドキュメント ページ (**http://edocs.beasys.co.jp/e-docs/**) に直接アクセスしてください。

マニュアルについてのフィードバック　BEA WebLogic Portal マニュアルについてのフィードバックをお寄せください。ご質問やコメントがあれば、電子メールで **docsupport-jp@bea.com** までお送りください。なお、お送りいただく電子メールには、BEA WebLogic Portal **7.0** リリースのマニュアルをお使いであることを明記してください。

BEA WebSUPPORT への連絡　このバージョンの BEA WebLogic Portal について質問がある場合、または BEA WebLogic Portal のインストールや実行に問題がある場合には、BEA WebSUPPORT (**http://support.bea.com/welcome.jsp**) を通じて BEA カスタマ サポートにご連絡ください。製品パッケージに同梱のカスタマ サポート カードに記載されている連絡先にお問い合わせいただいても結構です。

第 1 章 WebLogic Portal 7.0 から WebLogic Portal 7.0 サービス パック 1 への移行

WebLogic Platform 7.0 の新機能である Configuration Wizard を使用すると、新しいドメインをすばやく、簡単に作成することができます。WebLogic Platform 7.0 で Configuration Wizard を使用して作成したドメインを WebLogic Platform 7.0 サービス パック 1 で使用するためには、それらのドメインを移行する必要があります。

ほとんどのドメインについては、WebLogic Portal 7.0 から WebLogic Portal 7.0 サービス パック 1 への移行は 3 つのステップからなる手順で行います。

- ステップ 1: 製品の JAR ファイルのアップグレード。この作業はドメイン ディレクトリ内で行います。この作業のための移行スクリプトが用意されています。

  注意 : ドメインを移行前の状態に戻すこともできます。

- ステップ 2: サービス パック 1 での変更に合わせたドメインの更新。ドメインの生成に使用したドメイン テンプレートの種類によっては、既存のスクリプトまたはファイルの追加または修正が必要になる場合があります。

- ステップ 3: 新しい BEA_HOME ディレクトリの位置に合わせた起動スクリプトとコンフィグレーション ファイルの更新 (アップグレードでない場合のみ)。このステップは、WebLogic Platform 7.0 サービス パック 1 を新規のディレクトリにインストールする場合にのみ実行します。

  注意 : 既にインストールされていた WebLogic Platform 7.0 を *アップグレード*している場合には、このステップは不要です。

以下の各節では、これらのステップについて詳しく説明します。この手順は、移行する *個別の*ドメインに対して繰り返し実行する必要があります。

注意： この章では、WebLogic Portal に固有のドメインを移行する方法について説明しています。その他の WebLogic Platform ドメインの移行については、次の URL にある「*WebLogic Platform 7.0 サービス パック 1 リリース ノート*」内の「Configuration Wizard を使って作成したドメインの移行」を参照してください。

http://edocs.beasys.co.jp/e-docs/wlp/docs70/relnotes/relnotes.htm#migration

ステップ 1: 製品の JAR ファイルのアップグレード

Configuration Wizard を使用して生成したドメインを対象として、製品の JAR ファイルを サービス パック 1 にアップグレードするには、*BEA_HOME*\weblogic700\server\bin ディレクトリに移動して次のコマンドを実行します。

Windows の場合：`migrate.cmd` *`domain mode`*
UNIX の場合：`migrate.sh` *`domain mode`*

注意： コマンド実行後、表示されたメッセージに従って何かキーを押すと処理が開始します。

次の表にコマンドライン引数の定義を示します。

引数	説明
domain	ドメイン ディレクトリの絶対パス名

引数	説明
mode	移行モード。モードは次のいずれかの値に設定可能 ♦ upgrade: ドメイン ディレクトリ内の製品 JAR ファイルを必要に応じてアップグレードする。元の製品 JAR ファイルは *.jar.orig という名前で保存される。既存の製品 JAR ファイルのタイムスタンプが、SP1 でインストールされる同じ製品 JAR ファイルのタイムスタンプよりも新しい場合、そのファイルは上書きされない。これはデフォルトのモード。 ♦ revert: 以前に移行済みのドメインを、移行時に生成されたバックアップ ファイル (*.jar.orig) を使用して元に戻す。一連の *.jar.orig ファイルが存在しない場合、コマンドは無視される。

たとえば次のコマンドを実行すると、デフォルトのユーザ プロジェクト ディレクトリ (*BEA_HOME*\user_projects) に位置する mydomain という名前のドメインがアップグレードされます。

Windows の場合 : `migrate.cmd c:\bea\user_projects\mydomain upgrade`
UNIX の場合 : `migrate.sh c:/bea/user_projects/mydomain upgrade`

次のコマンドでは、移行の間に mydomain に対して行われた変更を元に戻します。

Windows の場合 : `migrate.cmd c:\bea\user_projects\mydomain revert`
UNIX の場合 : `migrate.sh c:/bea/user_projects/mydomain revert`

ステップ 2: サービス パック 1 での変更に合わせたドメインの更新

WLP Domain テンプレートを基に作成されたドメインを、WebLogic Platform 7.0 サービス パック 1 に対応するように更新するには、次の手順に従います。

注意 : ファイルの追加または修正を行う前に、元のファイルをバックアップしておくことをお勧めします。

1. (デフォルトで *BEA_HOME*\user_projects*domain*\beaApps\portalApp\tools\WEB-INF ディレクトリに位置する) tools Web アプリケーションの web.xml ファイル内で、セキュリティ制約「Customer Profile and Order Pages」の箇所を探し、セキュリティ制約が適用されるリソースを <url-pattern> 要素を使用して定義します。

   次のサンプルは web.xml ファイルの一部です。必要な更新は**太字**で示されています。

```
<security-constraint>
  <!-- リソース コレクションを定義する -->
  <web-resource-collection>
    <web-resource-name>
      Customer Profile and Order Pages
    </web-resource-name>
    <description>
      Customer Profile and Order Pages
    </description>
  <!-- リソース コレクションの URL パターン -->
    <url-pattern>/tools/*</url-pattern>
    <url-pattern>/repository/*</url-pattern>
    <url-pattern>/security/*</url-pattern>
    <http-method>GET</http-method>
    <http-method>POST</http-method>
  </web-resource-collection>
</security-constraint>
```

   注意 : WebLogic Server は、security-constraint 要素内部の個別の web-resource-collection 要素について、少なくとも 1 つ以上の URL パターンが含まれているかどうかを検証します。対象ドメイン内に他にも Web アプリケーションがある場合、すべての web-resource-collection 要素に、セキュリティ制約に適合する URL パターンが少なくとも 1 つ以上含まれていることを確認してください。

2. *BEA_HOME*\weblogic700\common\templates\domains\shared\bea\portal\webapps\tools\tools ディレクトリから、tools Web アプリケーションの tools ディレクトリに次のファイルをコピーします。カスタマイズされているファイルがある場合、そのファイルを上書きしないように注意してください。

```
catalog\category_add_remove_items.jsp
catalog\item_property_edit.jsp
catalog\item_property_edit_mr.jsp
catalog\item_property_edit_mu.jsp
```

```
catalog\item_property_edit_sr.jsp
catalog\item_property_edit_su.jsp
catalog\item_search.jsp
usermgmt\groupuser_property_edit_mr.jsp
usermgmt\groupuser_property_edit_mu.jsp
usermgmt\groupuser_property_edit_sr.jsp
usermgmt\groupuser_property_edit_su.jsp
usermgmt\group_add_remove_users.jsp
usermgmt\group_edit.jsp
usermgmt\group_scope_property.jsp
usermgmt\user_create.jsp
usermgmt\user_edit_info.jsp
usermgmt\user_scope_property.jsp
```

3. ドメイン内にポータル Web アプリケーションが作成されている場合、それぞれのアプリケーションについて次の手順を実行する必要があります。

   a. *BEA_HOME*\weblogic700\common\templates\webapps\portal\baseportal\j2ee\framework ディレクトリから、ポータル Web アプリケーションの framework ディレクトリに次のファイルをコピーします。これらと重複したファイル名を使用して作成されているファイルがある場合、そのファイルを上書きしないように注意してください。

   ```
   edit_titlebar.properties
   error\configurationerror.properties
   error\footer.inc
   error\header.inc
   error\header.properties
   error\missingformfield.properties
   error\parameters.properties
   error\pipeline.properties
   error\request.properties
   error\runtimeerror.properties
   hnav_bar.properties
   maximize_titlebar.properties
   minimize_titlebar.properties
   normal_titlebar.properties
   security\help.properties
   security\meta.inc
   vnav_bar.properties
   ```

   b. ポータル Web アプリケーションの framework ディレクトリ内の、次の JSP ファイルを更新します。このとき、*BEA_HOME*\weblogic700\common\templates\webapps\portal\baseportal\j2ee\framework ディレクトリ内の同名のファイルを比較のための参考として使用します。ポータルをカスタマイズする目的で JSP ファイル

に修正が加えられている場合があるため、該当のファイルがカスタマイズされていないことが確実でない場合は、既存のファイルを上書きしないことをお勧めします。

```
edit_titlebar.inc
error\configurationerror.jsp
error\error.jsp
error\missingformfield.jsp
error\parameters.jsp
error\pipeline.jsp
error\request.jsp
error\runtimeerror.jsp
error\sessiontimeout.jsp
error\sessiontimeout.properties
floated_portlet.jsp
hnav_bar.jsp
maximize_titlebar.inc
minimize_titlebar.inc
normal_titlebar.inc
security\help.jsp
security\login_header.inc
security\need_group.jsp
security\new_user.jsp
security\set_password.jsp
tools\header.jsp
tools\header.properties
tools\portal_prefs.jsp
vnav_bar.jsp
```

c. *BEA_HOME*\weblogic700\common\templates\webapps\portal\baseportal\j2ee\WEB-INF\lib ディレクトリから、ポータル Web アプリケーションの WEB-INF\lib ディレクトリに次のファイルをコピーします。カスタマイズされているファイルがある場合、そのファイルを上書きしないように注意してください。

```
ent_taglib.jar
es_taglib.jar
i18n_taglib.jar
lic_taglib.jar
p13n_servlet.jar
portal_servlet.jar
portal_taglib.jar
portlet_taglib.jar
ren_taglib.jar
res_taglib.jar
um_taglib.jar
util_taglib.jar
visitor_taglib.jar
```

```
webflow_servlet.jar
webflow_taglib.jar
weblogic-tags.jar
```

ステップ 3: 新しい BEA_HOME ディレクトリの位置に合わせた起動スクリプトとコンフィグレーション ファイルの更新 (アップグレードでない場合のみ)

注意 : このステップは、インストール済みの WebLogic Platform 7.0 とは別の新しいディレクトリに、WebLogic Platform 7.0 サービス パック 1 をインストールする場合にのみ必要です。既にインストールされていた WebLogic Platform 7.0 の *アップグレード* を行っている場合には、このステップは実行不要です。

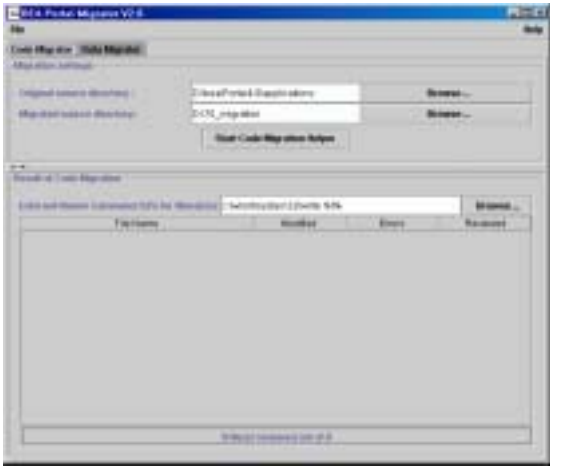
`startWebLogic` などのドメイン起動スクリプトと、`config.xml` などのコンフィグレーション ファイルでは、*`BEA_HOME`* ディレクトリ内部のファイルの絶対パス名を定義します。これらの絶対パス名を検索し、新しい *`BEA_HOME`* ディレクトリの位置に合わせて更新する必要があります。また、ビルド スクリプトなどのカスタム スクリプトで、*`BEA_HOME`* ディレクトリ内部のファイルの絶対パス名を定義しているものがあれば、新しい *`BEA_HOME`* の位置に合わせてそれらのスクリプトも更新しなければなりません。

注意 : 多くの起動スクリプトでは、*`BEA_HOME`* ディレクトリを参照する変数を含め、作業中のシェルにおける環境変数を設定します。スクリプト ファイル内で *`BEA_HOME`* への参照を更新した後で、最新の環境設定が確実に反映されるように新しいシェルを開くことをお勧めします。

第 2 章 WebLogic Portal 4.0 から WebLogic Portal 7.0 への移行

この章では、WebLogic Portal リリース 4.0 から 7.0 への移行に関する情報を示します。移行プロセスをスムーズに進めるために、図 2-1 に示すような移行ツールを使用します。

図 2-1 移行ツールの [Code Migrator] ウィンドウ



移行プロセスでは、まずいくつかの移行準備ステップを完了し、次にコードおよびデータを移行し、移行されたファイルをリリース 7.0 のコンフィグレーションでアセンブルします。この章では、以下の内容について説明します。

- ステップ 1: 移行前の環境およびファイルの準備
- ステップ 2: 4.0 から 7.0 へのデータの移行
- ステップ 3: 4.0 から 7.0 へのコードの移行
- ステップ 4: その他の WebLogic 製品の移行

- ステップ 5: 移行されたファイルのアセンブルと追加のコンフィグレーション タスクの実行

注意： ステップ 5 には、移行したリリース 7.0 実装を稼働させるために必ず実行しなければならないタスクが含まれます。補助的な移行タスクについては、*「トラブルシューティング」*の索引エントリを確認してください。また、移行に関するその他のヒントについては、カスタマサポート (http://websupport.bea.com/welcome.jsp) および BEA dev2dev (http://developer.bea.com/index.jsp) の各 Web サイトを参照してください。

- ステップ 6: 4.0 から 7.0 への移行の検証
- ステップ 7: 次に行うこと
- Migration Viewer ツールによるコードへの変更の参照
- 移行ファイル

ステップ 1: 移行前の環境およびファイルの準備

この節では、以下の内容について説明します。

- 最新のマニュアルおよび関連ファイルの入手
- 移行関連の問題への対処
- 環境の準備
- migrator.bat および migration_install.properties ファイルの編集

最新のマニュアルおよび関連ファイルの入手

以下の場所から、最新のマニュアルおよび関連ファイルを入手してください。

- メイン リリースおよびサービス パックのリリース ノート

すべてのバージョンのサービス パックに付属するリリース ノートには、ソフトウェアの変更点に関する重要な情報が記載されています。移行を完了する前に、リリース ノートの内容をもう一度確認してください。サービス パックは次の場所から入手できます。

- Evaluation Center – http://www.beasys.co.jp/evaluation/index.html
- カスタマサポート（アップグレード インストールのみ対象）– http://websupport.bea.com/welcome.jsp（サポート サイトにアクセスするには、正規のサポート契約を締結している必要があります。）

■ BEA 製品のドキュメント (edocs) および BEA dev2dev Web サイト

- 製品のドキュメント – http://edocs.beasys.co.jp/e-docs/
- BEA dev2dev – http://developer.bea.com/index.jsp

インストール済みパッチのチェック

サービス パックをインストールする前に、現在インストールされているパッチの種類を確認します。これには、WebLogic Server など、使用しているすべての WebLogic 製品のパッチが該当します。

WebLogic Portal の移行は、WebLogic Portal 4.0 のインストールと、サービス パック 2 に含まれていた一連のパッチがベースとなります。ただし、一部のパッチについては、WebLogic Portal およびその他の WebLogic 製品のサービス パックとは別途に配布されているため、移行を行う前にすべての WebLogic パッチを検証する必要があります。たとえば、サービス パックとは別にインストールしたパッチに、移行時に使用するものとバージョンが異なる JAR ファイルが含まれているような場合があります。BEA カスタマサポート (http://www.bea.com/support/programs.shtml) に問い合わせて、インストール済みのすべてのパッチがサービス パックに含まれていたものかどうかを調べ、そうでない場合は、該当するパッチについて移行関連の問題が報告されていないかどうかを確認してください。

移行関連の問題への対処

移行プロセスの開始前またはその間、以下の各項目の内容をよく確認し、実装環境、またそれぞれの問題が実装にどのように影響するかに応じて適切な対処を行ってください。

- 移行時の WebLogic Integration との相互作用
- データベースの移行と Sybase に関する重要情報
- トリガと WebLogic Portal 4.0 サービス パック 1 に関する注意事項
- E-Business Control Center プロジェクト移行の詳細

移行時の WebLogic Integration との相互作用

既に Web Logic Portal または WebLogic Integration を使用しており、これから WebLogic 7.0 プラットフォームへの移行を行う場合、WebLogic Portal と WebLogic Integration の両方を移行しなければなりません。これら 2 つの製品は WebLogic Portal RDBMS レルムを共有します。コンフィグレーション済みの RDBMS には、WebLogic Integration の事前定義済みデータが含まれています。WebLogic Integration からインストールされた fileRealm には、WebLogic Integration のシステム ユーザだけが含まれ、WebLogic Integration からのその他の事前定義済みデータは含まれません。

これらのコンフィグレーション関連の問題は、Configuration Wizard と同様に移行による影響を受けます。詳細については、WebLogic Integration *Platform のセキュリティ ページ*(http://edocs.beasys.co.jp/e-docs/wls/docs70/security.html) を参照してください。

データベースの移行と Sybase に関する重要情報

Sybase ではカラム サイズに関する制限が緩和されており、WebLogic Portal でもその緩和された制限が適用されます。

Sybase を使用している場合、以下の手順を順序どおりに実行する必要があります。

現在 Sybase 11.9 または 12.0 を使用している場合、リリース 4.0 から 7.0 に移行します。Sybase 12.5 に更新する場合、BEA dev2dev (http://developer.bea.com/index.jsp) から Sybase スキーマ更新スクリプトをダウンロードして実行してから、12.5 への更新を行います。

現在 Sybase 12.5 を使用している場合、リリース 4.0 から 7.0 に移行します。次に、BEA dev2dev Web サイトから Sybase スキーマ更新スクリプトをダウンロードし、実行します。

トリガと WebLogic Portal 4.0 サービス パック 1 に関する注意事項

サービス パック 1 では、実装に悪影響を及ぼす可能性があるトリガが削除されています。まだサービス パック 1 を使用している場合、移行の前にサービス パック 2 をインストールしてください。

サービス パックは次の場所から入手できます。

- Evaluation Center: http://commerce.bea.com/downloads/products.jsp
- サポート サイト（アップグレード インストールのみ対象）
  http://support.bea.com/welcome.jsp

E-Business Control Center プロジェクトの移行について

E-Business Control Center プロジェクトの移行に関して、通常は特に問題となる点はありません。ただし、プロジェクトの内容または最新バージョンへの準拠状況によっては、予期しない結果が生じる可能性があります。プロジェクトが完全でない場合、その移行時に内容の有効性はチェックされないため、破損したファイルもすべて移行されます。これを避けるには、移行の前に E-Business Control Center プロジェクトに対して必要なすべての作業を行ってください。移行した後は、プロジェクトが想定どおりに機能することを確認し、管理ツールを使用して必要な変更を行ってください。

プロジェクトの移行手順の詳細については、B-18 ページの「E-Business Control Center プロジェクト移行の詳細」を参照してください。

ポータルのデフォルト コンフィグレーション設定の移行

各ポータルには、匿名ユーザがポータルにアクセスしたときに使用されるデフォルト コンフィグレーションがあります。デフォルト コンフィグレーションは、ポータルに関するその他のデータとともにこのリリースに移行されます。

デフォルト コンフィグレーションが確実に移行されるようにするには、移行を開始する前にポータル コンフィグレーションをサーバと同期します。同期を行わないか、何か他の理由によって移行後のポータルにデフォルト コンフィグレーション セットが存在しない場合、ポータルにはデフォルト コンフィグレーション「*default*」が割り当てられます。デフォルト コンフィグレーションを変更するには、WebLogic Portal Administration Tools を使用します。

環境の準備

この節では、以下の内容について説明します。

- サポート対象プラットフォームをチェックする
- 最新のサービス パックをインストールする
- BEA WebLogic 7.0 をインストールしてビルド環境をセットアップする
- コード、データ、データベースの完全バックアップを作成する
- 移行環境をチェックする
- E-Business Control Center プロジェクトをサーバと同期する

サポート対象プラットフォームをチェックする

WebLogic Portal 7.0 のサポート対象プラットフォーム ドキュメントに記載された仕様にシステムが準拠していることを確認します。このドキュメントは http://edocs.beasys.co.jp/e-docs/platform/docs70/support/index.html から入手できます。

最新のサービス パックをインストールする

最新のサービス パックがインストール済みであることを確認してください。このマニュアルの説明は、サービス パック 2 (SP2) からの移行を前提として記述されています。

BEA WebLogic 7.0 をインストールしてビルド環境をセットアップする

『WebLogic Platform のインストール』(http://edocs.beasys.co.jp/e-docs/platform/docs70/install/index.html) と、このマニュアル中で示しているその他のドキュメントの指示に従って、リリース 7.0 をシステムに正しくインストールします。次の手順に移る前に、インストール環境およびソフトウェアが正しくインストールおよびセットアップされていることを確認してください。

変換されたコード ファイルをその後の移行プロセスで修正、テスト、およびデプロイできるように、`.jar` ファイルやその他のファイルの作成など、リリース 7.0 のビルド環境システムのセットアップを行います。これは、ユーザが自分自身で作成して構築する独自のビルド環境です。

警告 : 4.0 形式のデータベースに対して create_all スクリプトを実行しないでください。このスクリプトは、空のデータベースを作成し、そのデータベースを指すようにサーバを設定し、リリース 7.0 が正しくインストールされて稼働していることを移行開始前に確認する目的で実行できます。

注意 : リリース 7.0 への WebLogic Portal の移行を終了するまでは、リリース 4.0 のデータベースを指すように設定されていた WebLogic Server 7.0 を実行することはできません。

コード、データ、データベースの完全バックアップを作成する

データベース スキーマを含めて、WebLogic Portal 4.0 システム全体を少なくとも 1 回完全にバックアップします。移行プロセスが途中で失敗した場合、バックアップした内容を復元して再度移行を行う必要があります。

移行環境をチェックする

WebLogic Server がインストールされているのと同じコンピュータ上に、移行用の各種ファイルがインストールされていることを確認します。移行時には、BEA ディレクトリ内の `weblogic.jar` ファイルおよびライセンス ファイルにアクセスできることが必要です。

E-Business Control Center プロジェクトをサーバと同期する

デフォルト コンフィグレーションが取得されるように、必ずすべてのポータル ファイルの最新バージョンをサーバと同期します。

DataMigratorBundle.properties または MigratorBundle.properties の修正有無の判断

これらのファイルは、移行時の重要な処理を制御します。(任意の順序でタスクを実施できるように) Data Migrator でのタスクをスキップする必要がある場合や、移行ツールを国際化する必要がある場合の手順については、B-1 ページの「移行ファイル」を参照してください。

警告： 一般的な移行では、これらのファイルを修正する必要はありません。これらのファイルの修正は、熟練した開発者の手で、移行ツールまたは移行プロセスに対して行わなければならない変更がある場合にのみ行うようにしてください。

migrator.bat および migration_install.properties ファイルの編集

以下のようにコンフィグレーション設定を入力します。

1. <PORTAL_HOME>\migration\bin\migrator.bat（または migrator.sh）ファイルを開きます。コード リスト 2-1 に示すように、使用しているデータベースを示している行をコメントアウトします。

コード リスト 2-1　migrator.bat ファイル

```
echo off
REM -------------------------------------------------------#
REM  Migrator Starter Script on Windows                     #
REM -------------------------------------------------------#
SETLOCAL

set DATABASE=ORACLE_THIN
REM set DATABASE=MSSQL
REM set DATABASE=SYBASE_JCONNECT
REM set DATABASE=DB2_TYPE2
if "%DATABASE%" == "" ( echo "The DATABASE variable must be uncommented in
migrator.bat"
    exit 1 )

CALL ..\..\bin\win32\set-environment.bat

REM -------------------------------------------------------#
REM         VARIABLES TO SET
REM -------------------------------------------------------#

REM Due to database specifics, you may have to set your
REM path to include dll

REM -------------------------------------------------------#
REM          The mini script
REM -------------------------------------------------------#
set MIGRATION_DIR=%WL_COMMERCE_HOME%\migration
set MIGRATION_LIB=%MIGRATION_DIR%\lib
set
MIG_CLASSPATH=%BEA_HOME%\lib\tools.jar;%MIGRATION_LIB%\migration.jar;%MIGRATION
_LIB%\apache\xerces-1_4_3\xerces.jar;%MIGRATION_LIB%\apache\xalan-j_2_0_1\xalan
.jar;%MIGRATION_LIB%\p13n_system.jar;%WEBLOGIC_HOME%\lib\weblogic.jar;%BEA_HOME
%

REM -------------------------------------------------------#
REM For testing the setup only
REM -------------------------------------------------------#
REM echo BEA_HOME=%BEA_HOME%
REM echo WEBLOGIC_HOME=%WEBLOGIC_HOME%
REM echo WL_COMMERCE_HOME=%WL_COMMERCE_HOME%
```

```
REM echo MIGRATION_DIR=%MIGRATION_DIR%
REM echo MIGRATION_LIB=%MIGRATION_LIB%
REM echo MIG_CLASSPATH=%MIG_CLASSPATH%


%JDK_HOME%\bin\java -cp %MIG_CLASSPATH%  -DMIGRATION_DIR=%migration_dir%
com.bea.commerce.migration.tools.Migrator

echo on
```

2. <PORTAL_HOME>\migration\migration_install.properties ファイルを開きます。コード リスト 2-2 にファイルの内容を示します。編集する必要のある項目は太字で示しています。
3. 「migration_start」の行で、値を true に設定します。
4. 使用しているデータベース用の設定ブロックで、各行の先頭のコメント記号を外します。また、その他のデータベースの設定ブロックで、すべての行がコメントアウトされていることを確認します。
5. コード リスト 2-2 に示すように、データベース接続プロパティを変更します。

注意： Oracle については、database.version の値は「817」だけに設定できます。Oracle 9i を使用する場合も、この設定は「817」のままにします。

SQL Server については、該当するバージョンの行をコメント解除することによって、SQL Server 7 または 2000 からバージョンを選択できます。

server=**YOURSERVER** の箇所で、データベース サーバの名前を入力します。たとえば Oracle の場合、システムのコンフィグレーションに応じて、Oracle の SID または netservice 名のどちらかを入力します。

コード リスト 2-2　migration_install.properties ファイル

```
#######################################################################
#
# PROPERTIES TO BE SET BY THE USER AT 'INSTALL' TIME
#
#######################################################################
```

```
# ------------------------------------------------------------
#       When you have set the properties you need, set the
#       following flag to 'true' so the migrator tool will start.
#       Otherwise, the Migrator will assume you have NOT set
#       any property and will refuse to run.
# ------------------------------------------------------------
start_migrator=false

# ------------------------------------------------------------
#Database Properties  // uncomment the database you are using
                      // and enter the appropriate connection values
# ------------------------------------------------------------


# Database connection properties
#
#------Oracle Thin Driver----------------------#
#
# For oracle, replace the following:
#    @USER@, @PASSWORD@, @SERVER@, @PORT@, and @SID@
#    e.g. jdbc:oracle:thin:@localhost:1521:ORCL
#
database.connection.driver = oracle.jdbc.driver.OracleDriver
database.connection.url = jdbc:oracle:thin:@@SERVER@:@PORT@:@SID@
database.connection.props = user=@USER@;password=@PASSWORD@


#------MS SQL Server ----------------------#
#
# For SQL Server, replace the following:
#    @SERVERPORTNUMBER@, @USER@, @PASSWORD@, and @SERVER@ (in two different
locations)
#
#database.connection.driver = weblogic.jdbc.mssqlserver4.Driver
#database.connection.url =
jdbc:weblogic:mssqlserver4:@SERVER@:@SERVERPORTNUMBER@
#database.connection.props =
user=@USER@;password=@PASSWORD@;server=@SERVER@;weblogic.t3.waitForConnection=t
rue;weblogic.t3.waitSecondsForConnection=999999999999;weblogic.jts.waitSecondsF
orConnectionSecs=999999999999


#------Sybase jConnect 5.2 ----------------------#
#
# For Sybase, replace the following:
#    @SERVER@, @PORTNUMBER@, @USER@, and @PASSWORD@
#
#database.connection.driver=com.sybase.jdbc2.jdbc.SybDriver
#database.connection.url=jdbc:sybase:Tds:@SERVER@:@PORTNUMBER@
```

```
#database.connection.props =
user=@USER@;password=@PASSWORD@;server=@SERVER@;weblogic.t3.waitForConnection=t
rue;weblogic.t3.waitSecondsForConnection=999999999999;weblogic.jts.waitSecondsF
orConnectionSecs=999999999999




#------IBMS's DB2 --------------------------------------#
#
# For DB2, replace the following:
#    @DB2_DATABASE@, @USER@, and @PASSWORD@
#
#database.connection.driver=COM.ibm.db2.jdbc.app.DB2Driver
#database.connection.url=jdbc:db2:@DB2_DATABASE@
#database.connection.props =
user=@USER@;password=@PASSWORD@;server=@SERVER@;weblogic.t3.waitForConnection=t
rue;weblogic.t3.waitSecondsForConnection=999999999999;weblogic.jts.waitSecondsF
orConnectionSecs=999999999999


#
# Database and version
#
# for Oracle users use database.version=817 for either Oracle 8.1.7 or 9i
database.name=oracle
database.version=817
#database.name=sql_server
#database.version=2000
#database.version=7
#database.name=sybase
#database.version=12
#database.name=db2
#database.version=7
```

ステップ 2: 4.0 から 7.0 へのデータの移行

この節では、データを 7.0 リリースに移行する方法について説明します。説明する内容は以下のとおりです。

- ツールとプロセスを理解する

ツールとプロセスを理解する

この節では、以下のトピックの概要を示します。

移行ツールの機能

移行ツールは、WebLogic Portal バージョン 4.0 から 7.0 へのコードとデータの移行を支援します。移行ツールは補助ユーティリティであり、コードとデータを移行してコメントを挿入しますが、それによって移行がすべて完了するわけではありません。移行作業の多くの部分は、担当するチームが人手で行う必要があります。

移行ツールは、稼働中のサーバとのやり取りを必要としないスタンドアロンツールです。

データ移行プロセスを再確認する

移行プロセスでは、一部のファイルを未変更のまま残す、ファイルの名前を変更する、新しいファイルを作成するなどのさまざまな処理が行われます。データベースの移行では、リリース 4.0 のテーブルそのものを対象に処理が行われます。リリース 4.0 のテーブルに手を付けずに、新しいリリース 7.0 形式のテーブルの集合を作成するのではありません。

移行の影響を受けるテーブルはバックアップされ、`tablename_temp40` のような名前が付けられます。移行が完了し、システムがリリース 7.0 で正しく機能していることを確認した後で、それらのテーブルを削除することができます。

データ移行ツールのタスクを再確認する

この節では、ツールによって行われるタスクについて説明します。タスクの実行順は、最後の 2 つを除いてそれほど重要ではありません。（以下に示す中の）「RDBMS データ同期」タスクが完了していないと、サーバは起動しません。すべてのタスクが完了するまでは、サーバは正しく機能しません。

E-Business Control Center プロジェクトの移行は個別に行う必要があります。言い換えると、すべてのプロジェクトに対して E-Business Control Center データの移行を一度ずつ実行する必要があります。

以下の手順と説明は、移行ツールを使用してデータ移行を行うときに画面上にも表示されます。

- **E-Business Control Center データの移行** – E-Business Control Center プロジェクトのデータをプロジェクト単位で移行します。[Execute] をクリックすると、移行対象のプロジェクトを指定するためのウィンドウが表示されます。プロジェクトの移行元ディレクトリに移動します（指定するディレクトリには、`application-sync` という名前の子ディレクトリが存在している必要があります）。移行後のプロジェクトが書き込まれる出力ディレクトリに移動します。これは、既存のデータを上書きしない限りはどのようなディレクトリでもかまいません。エンタープライズ アプリケーション ディレクトリが分かっている場合、ここでそのディレクトリを入力します。分からない場合、後から E-Business Control Center を使用してディレクトリを入力できます。移行ツールは、プロジェクトの名前と構造を保ちながら、このディレクトリに移行後のプロジェクトを書き込みます。

- **同期データベース (Data Sync RDBMS) の移行** – DATA_SYNC_ITEM、DATA_SYNC_APPLICATION、および DATA_SYNC_SCHEMA_URI の各テーブルのデータをクリアし、DATA_SYNC_APPLICATION、DATA_SYNC_SCHEMA_URI、および DATA_SYNC_VERSION の各テーブルの参照されていない行を削除するトリガを作成します。このステップは反復可能です。

注意： 以前にデータベース テーブルの変更を行った場合、以降のタスクに影響が及ぶ場合があります。ここまでの 2 つのタスクは影響を受けません。

- **ポータル データベース (Portal RDBMS) の移行** – PORTAL_PAGE_P13N テーブルのデータを PORTAL_PAGE_P13N_TMP40 テーブルに保存します。PORTAL_PAGE_P13N への外部キー制約を削除してから、テーブルを削除します。7.0 形式の PORTAL_PAGE_P13N テーブルを作成し、保存したデー

タをこのテーブルに再び挿入します。PORTAL_PAGE_P13N を参照する外部キー、チェック制約、およびトリガを追加します。このステップは反復可能です。

- **資格ルールセット データベース (ENTITLEMENT_RULESET RDBMS) の移行** – ENTITLEMENT_RULESET_TEMP40 テーブルを作成し、ENTITLEMENT_RULESET テーブルのデータを挿入します。作成されたテーブルは、RULESET_DOCUMENT カラムに入っていたドキュメントを変換するためのデータ ソースとして使用されます。資格ルールセット データベース（ENTITLEMENT_RULESET RDBMS）の移行は、資格ルールセット データの移行よりも先に実行しなければなりません。

警告 : このステップは、次のステップである「資格ルールセット データ（Entitlement Ruleset Data）の移行」の実行後に繰り返すことは*できません*。

- **資格ルールセット データ (Entitlement Ruleset Data) の移行** – 資格ルールセット データを移行します。ENTITLEMENT_RULESET テーブルの内容が、変換された RESULTSET_DOCUMENT（結果セット ドキュメント）によって更新されます。

変更が行われたデータベースの手動移行

リリース 4.0 のデータベース テーブルの定義を変更した場合、それらのテーブルの移行を手動で行う必要があります。

注意 : これは複雑な手順になる可能性があります。手動での移行を開始する前に、移行タスク、タスク間の依存関係、関係する SQL の内容について十分に理解しておいてください。まず、リリース 4.0 のデータのコピーを作り、そのコピーに対して移行を試してみることをお勧めします。

移行ツールを使用して、データベース定義への変更の影響を受けない移行タスクを完了します。次に、SQL ツールを使用して、変更の影響を受ける一連の移行 SQL 文を実行します。2-14 ページの「データ移行ツールのタスクを再確認する」に示されている順序の変更を行います。

定義が変更されたデータベースを移行するには、次の手順に従います。

1. データベースに対して行った変更を一覧にします。

2. 2-14 ページの「データ移行ツールのタスクを再確認する」で、データ移行作業の説明および前提関係に目を通し、データベースに行った変更の影響を受けるタスクを特定します。
3. 使用しているデータベース用の .sql ファイルを探します。ファイルは <PORTAL_HOME>\migration ディレクトリに収められています。.sql ファイルには、処理終了文を除いて、データ移行の各ステップを実行するために必要なすべての SQL 文が入っています。
4. .sql ファイルから、手順 2 で特定したタスクの箇所を探します。タスク別の SQL 文のブロックは、次に示すように、タスク名を示すコメント行で始まります。oracle-817-v4_0to7_0.sql ファイルの一部をコード リスト 2-3 に示します。

コード リスト 2-3　oracle-817-v4_0to7_0.sql ファイルの一部

```
-- ==============================

-- ENTITLEMENT_RULESET RDBMS Migration

CREATE TABLE ENTITLEMENT_RULESET_TEMP40 ( APPLICATION_NAME VARCHAR2(100) NOT
NULL, RULESET_URI VARCHAR2(254) NOT NULL, RULESET_DOCUMENT CLOB NULL,
CREATION_DATE DATE DEFAULT SYSDATE NOT NULL, MODIFIED_DATE DATE DEFAULT SYSDATE
NOT NULL)

INSERT INTO ENTITLEMENT_RULESET_TEMP40 ( APPLICATION_NAME, RULESET_URI,
RULESET_DOCUMENT, CREATION_DATE, MODIFIED_DATE ) SELECT APPLICATION_NAME,
RULESET_URI, RULESET_DOCUMENT, CREATION_DATE, MODIFIED_DATE FROM
ENTITLEMENT_RULESET
```

5. 特定したそれぞれのタスクについて、データベース定義の変更内容に合わせて新しい SQL 文を記述するとともに、既存の SQL 文を編集します。SQL 文を修正する必要のある移行タスク別に、SQL 文のファイルを 1 つずつ作成します。
6. SQL*Plus などの任意の SQL ツールを使用して、データベースに対して SQL 文を実行します。

移行ツールを使用したデータの移行

移行ツールによるデータ移行ステップを完了するには、この節の手順に従います。

警告： PORTAL_HOME\migration ディレクトリ内の migration_install.properties ファイルの編集が済んでいることを確認します。適切な情報が入力されていないと、移行ツールを実行しようとしたときにエラー メッセージが表示されます。

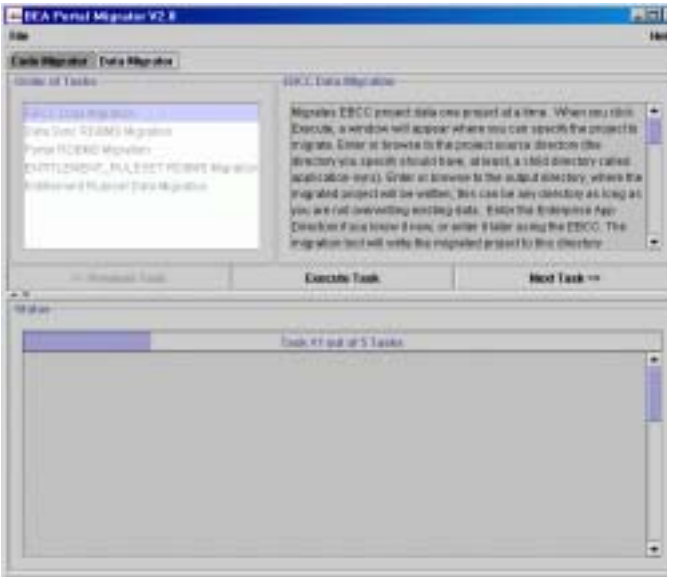
データベースを参照および操作するためのツールを使用して、データ移行プロセスの進捗を確認します。データベースに影響を及ぼすそれぞれのタスクが終了した後で、リリース 4.0 のデータベースを参照し、発生した変化を確認することができます。

1. WebLogic Portal サーバを停止します。
2. まだ作成していない場合、データベースとプロパティ ファイルの両方を含めて、データの完全なバックアップを *2 組*作成します。使用している場合、行動追跡データベースとイベント データベースもバックアップします。
3. ダブルクリックするかコマンド ラインを使用して、<PORTAL_HOME>\migration\bin ディレクトリ内の migrator.bat または migrator.sh ファイルを実行し、移行ツールを起動します。

   注意： 問題が発生した場合、migrator.bat ファイルを開き、すべての設定が正しいことを確認します。

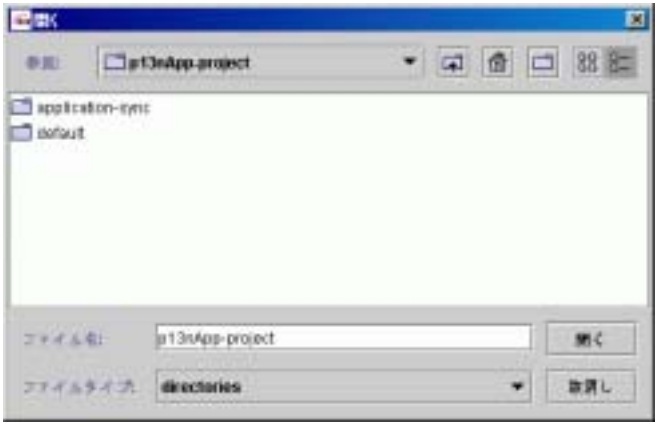
4. まだ前面に表示されていない場合、[Data Migrator] タブをクリックして 図 2-2 のような状態にします。

図 2-2　データ移行用ウィンドウ



5. リストの先頭のタスクを選択して [**Execute Task**] をクリックします。
6. 最初のタスクである「E-Business Control Center Data Migration」についてのみ、別のウィンドウが表示されます。図 2-3 に示すように、最初の E-Business Control Center プロジェクトについて移行元ディレクトリを入力します。これは通常プロジェクト ディレクトリ、つまり、`application-sync` ディレクトリの親ディレクトリです。

図 2-3 E-Business Control Center プロジェクトの移行元ディレクトリの選択



7. [E-Business Control Center Project Migration] ウィンドウが表示されたら、3 つのフィールドのそれぞれに適切な値を入力します。
   - E-Business Control Center Project Source Directory: プロジェクト ディレクトリ（例：`D:\BEA\Portal4.0\applications\p13nApp-project`）を指定します。
   - E-Business Control Center Project Destination Directory: 任意のディレクトリを指定できます。既存のファイルを上書きしないようにしてください。たとえば、`D:\70_migration` のように指定します。

注意： まだ存在していないプロジェクト移行先ディレクトリも指定でき、その場合新しくディレクトリが作成されます。

   - [Enterprise Application Root Directory]: プロジェクトがその内部で機能するエンタープライズ アプリケーション ディレクトリを指定します。まだディレクトリが分からない場合、空白のままにしておいて後から E-Business Control Center を使用して入力できます。詳細については、『E-Business Control Center オンライン ヘルプ』を参照してください。

   移行先ディレクトリ内に、移行元と同じ名前と構造のプロジェクト ディレクトリが作成されます。

8. 表示されるステータス メッセージで、移行の進捗状況を確認します。

   データ移行ウィンドウ内のメッセージについて メッセージの多くは非常に詳細で、価値のある情報を含んでおり、移行されたファイルまたは新規ファ

イルが作成されたディレクトリが示されることもあります。また、移行タスクによって影響を受けるテーブルが列挙されます。タスクごとのコメント集合の最後に、移行の成否を示すメッセージが表示されます。

`migration.log` ファイルには、より完全な情報が記録されます。このファイルには、ウィンドウに表示されたすべてのメッセージだけでなく、スタックトレースなどの補足情報も含まれます。

移行上の問題が発生したことを知らせるその他のメッセージが表示された場合、画面上のエラー メッセージから原因を判断して問題を解決し、タスクを再度実行します。一部のタスクについては、実行前にデータベースをバックアップから復元する必要があります。そのときは、問題を解決してからタスクを再実行してください。そのような状況では、ウィンドウのメッセージにもそのことが通知されます。データベースに対して行った変更が原因で問題が発生した場合の対処については、2-15 ページの「変更が行われたデータベースの手動移行」を参照してください。

9. その他すべての E-Business Control Center プロジェクトを移行します。
10. [**Next Task**] をクリックし、次の移行タスクに進みます。
11. タスクが問題なく完了したら、[**Next Task**] をクリックします。以後、すべてのタスクが完了するまでこの手順を繰り返します（B-5 ページの「MigratorBundle.properties および DataMigratorBundle.properties ファイル」で説明されているように、`DataMigratorBundle.properties` を編集することによってタスクを省略した場合、リスト内のそれらのタスクをスキップできます。それ以外の場合は、すべてのタスクを順番に実行しなければなりません）。

最新の XSLT に合わせたイベントの手動更新

WebLogic Portal 4.0 でイベントを使用していない場合、この節を読む必要はありません。

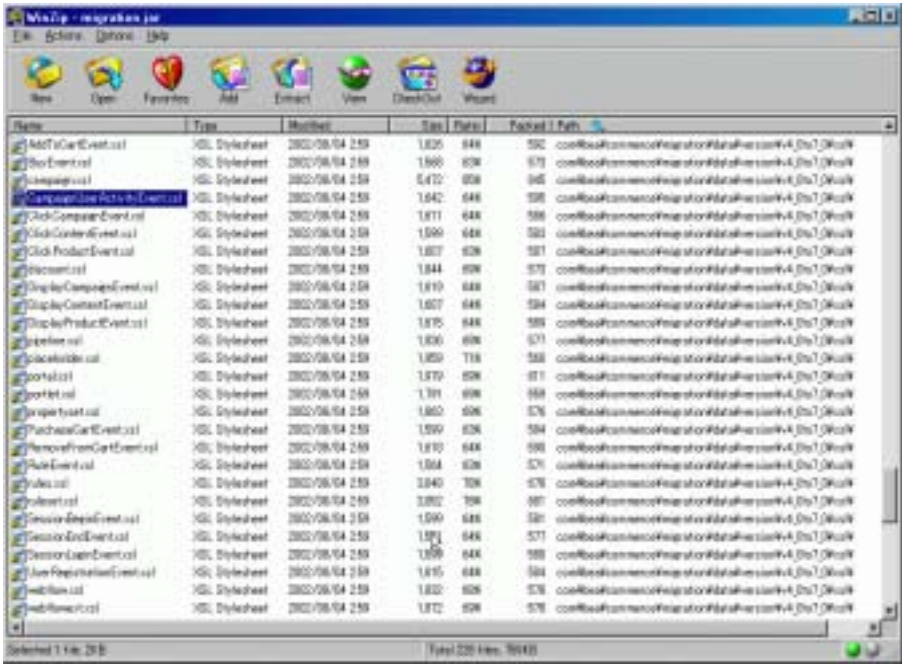
イベントはサイト内の特定の選択肢のクリックなど、ユーザが起こすアクションのことです。必要に応じて、データベース内でイベントを追跡し、Broadbase などの任意のツールを使用してイベントを分析できます （WebLogic Portal にはイベント用の分析ツールは付属しません）。

7.0 形式で新しいイベント データベースを作成し、そのデータベースでイベントの記録を開始することをお勧めします。イベントはサード パーティ製のツールで分析されるため、移行は必要ありません。

WebLogic Portal 7.0 でのイベントの記録には、最新の XSLT が使用されます。Xylon または同様のツールを使用して WebLogic Portal 4.0 のイベント データを最新の形式に更新する場合、付属の `.XSL` ファイルを使用できます。

図 2-4 に示すように、ファイルは `migration.jar` ファイルに収められています。イベントと関係するのは `*Event.xsl` という名前のファイルです。新しい XSLT を 4.0 形式のイベント データに適用するには、Xylon またはその他の任意のツールと、`*Event.xsl` ファイル群を使用します。

図 2-4 イベント データ更新用の XSL ファイル



システム移行後の 4.0 形式テーブルの削除

移行プロセスでは、移行中に作成されるバックアップ テーブル（tablename_temp40 のような名前）は削除されません。移行後のサイトが正常に稼働することをある程度の期間確認したら、これらの 4.0 形式のテーブルを削除してください。

ステップ 3: 4.0 から 7.0 へのコードの移行

コードの移行に際しては、この節の情報を参考にしてください。この節では、以下の内容について説明します。

- コード移行プロセスの再確認
- Migration Viewer の使用
- コードの移行
- JSP の移行

注意： 移行プロセスを迅速化するために、複数の人物が同時に移行を実行することができます。たとえば、ソース コード管理システムを使用する場合、1 人の開発者が、メインのリリース 4.0 ディレクトリの内部のメイン サブディレクトリのそれぞれをチェックアウトし、移行することができます。

警告： ある人物の変更によって別の人物の変更が上書きされることを防ぐには、別の人物が移行を行っているコードのサブディレクトリについて、誰も移行を行っていないことを確認します。

コード移行プロセスの再確認

コード移行ツールは .java ファイルを分析し、必要な変更を行うか、またはコードは更新せずに、変更が必要な箇所に目印となるコメントを挿入します。変更の種類に関係なく、ファイルにはコメントが挿入されます。コード移行ツールは移行プロセスを支援しますが、移行を完了する前に、すべてのコード ファイ

ルを人手で見直し、必要であれば変更を加える必要があります。変更およびコメントの挿入は、ファイルの新しいコピーに対して行われます。このコピーは、移行を開始する前に指定した位置に作成されます。

ツールによる移行が可能なコードの量は、実装によって大きく異なります。

JSP をプリコンパイルし、コード移行ツールを実行し、JSP、タグ、およびその他の関連するコードを適切に更新することによって、標準の `.java` ファイルおよび JSP の両方を移行することができます。

コード移行ツールは、以下の要素を重点的に分析します。

- import 文（パッケージおよび完全修飾クラス）
- 変数宣言
- オブジェクトのインスタンス化
- 変数参照
- メソッド呼び出し
- フィールド参照
- キャスト
- ステートメントのインスタンス
- `extends` 文と `implements` 文
- `return` 文

Migration Viewer の使用

リリース 7.0 でのコードへの変更点は、`<PORTAL_HOME>\migration\doc` ディレクトリ内の `migrinfo.html` ファイルにまとめられています。情報を検索およびソートするには、Migration Viewer ツールを使用します。詳細については、A-1 ページの「Migration Viewer ツールによるコードへの変更の参照」を参照してください。

これは非常に有益な情報であり、このリリースでどのクラス、パッケージ、メソッド、およびその他の API 要素が変更されているかが詳細にまとめられています。Migration Viewer は、コードおよび JSP を移行する過程で移行ツールと同

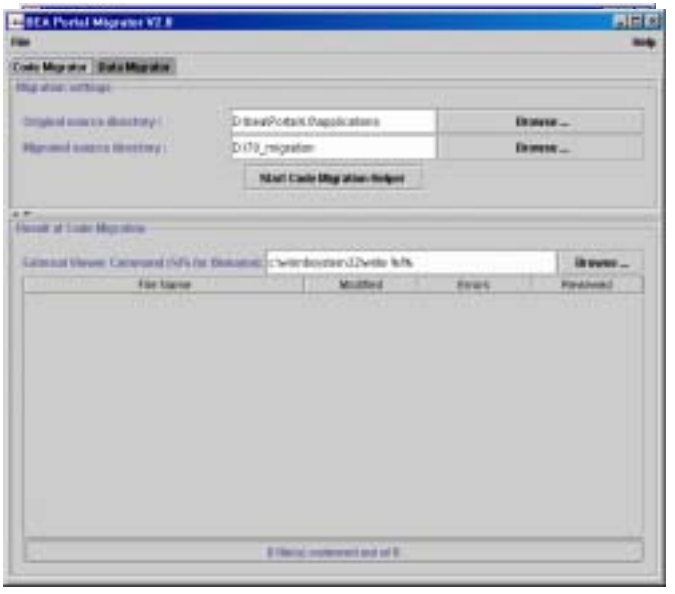
時に使用します。移行された個々のファイルの内容と移行に関するコメントを確認する過程で、API への変更にどのように対応しなければならないかを調べるために Migration Viewer を使用します。

コードの移行

この節では、ユーティリティを使用してコードを WebLogic Portal 7.0 に移行する手順を説明します。

1. 移行ユーティリティをまだ実行していない場合、`\migration\bin` ディレクトリにある `migrator.bat` または `migrator.sh` ファイルをダブルクリックして起動します。
2. まだ前面に表示されていない場合、[**Code Migrator**] タブをクリックして図 2-5 のような状態にします。

図 2-5　[Code Migrator] タブ



3. 移行元ディレクトリと移行先ディレクトリを指定します。

警告： 複数の人物が同時にコードの移行を行っている場合、ある人物が既に移行を行っているサブディレクトリに対して別の人物がさらに移行を行うことのないように注意してください。

4. 図 2-6 に示すように、コード ファイルの表示および編集に使用するエディタ プログラムのパスを入力します。エディタを登録すると、移行を行った後で、コード移行ツール ウィンドウの下部にあるリスト内でファイルを選択することによってコード ファイルを開けるようになります。

図 2-6 外部ビューアのパスの入力

注意： パスの終端のスペースと「`%f%`」はそのまま残す必要があります。これは、編集対象のコード ファイル名のプレースホルダです。

5. 移行が完了すると、移行されたファイルの一覧がウィンドウの下半分に表示されます。ウィンドウ内でファイル名をダブルクリックすると、登録しておいたエディタでファイルを開くことができます。

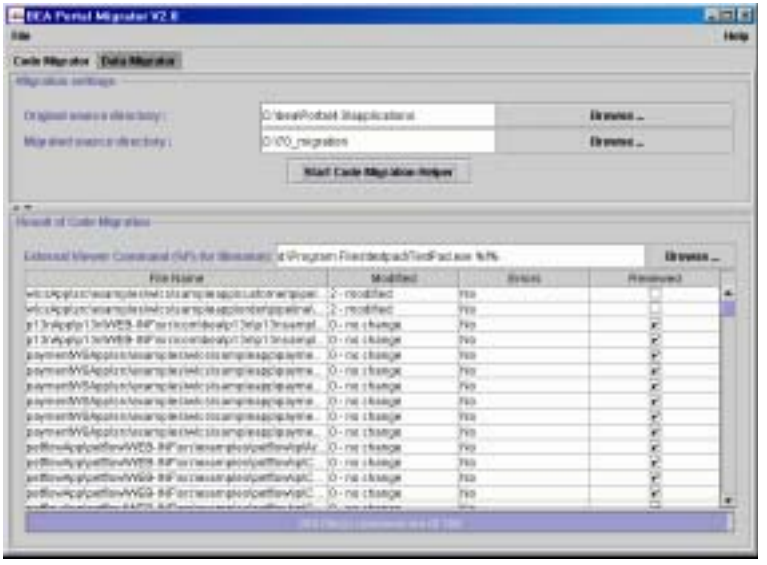
6. 移行プロセスでは、移行元ディレクトリ内のすべてのコードが 1 回のプロセスで移行されます。ファイルを 1 つずつ移行する必要はありません。プロセスの進行中は、ステータス メッセージと移行処理中のファイルの名前が画面に表示されます。

   移行元ディレクトリ全体を移行する準備ができたら、[**Start Code Migration Helper**] をクリックします。

   エラーが発生した場合、`PORTAL_HOME\migration` ディレクトリ内の `migration.log` ファイル、画面上のステータス メッセージ、および、移行後のコード ファイルに挿入された移行に関するコメントを参考にして問題を解決します。

7. 移行プロセスが完了すると、移行処理中に発生した問題、ファイル変更の有無などの情報とともにファイルの一覧が表示されます。図 2-7 に例を示します。

図 2-7　移行されたファイルの一覧とステータス情報



各ファイルの移行結果は、0 ～ 3 のレベルで示されます。レベル 3 は、人手による確認および追加の変更の必要性が最も高いことを示します。以下、それぞれのレベルについての説明です。

- **0 - no change:** ファイルへの変更は行われませんでした。これらのファイルについては、ウィンドウ内で [Reviewed] カラムにチェック マークが入ります。
- **1 - comments:** 移行に関するコメントが挿入されています。ファイルを確認して、さらに手動での変更を行う必要があるかどうかを判断します。
- **2 - modified:** ファイルの内容が変更されています。ファイルに行われた変更が適切かどうかを確かめるとともに、挿入された移行に関するコメントを確認してください。
- **3 - danger:** ファイルの内容を大幅に変更する必要があります。たとえば、存在しなくなったクラスを extend するコードがファイルに含まれている場合などが該当します。

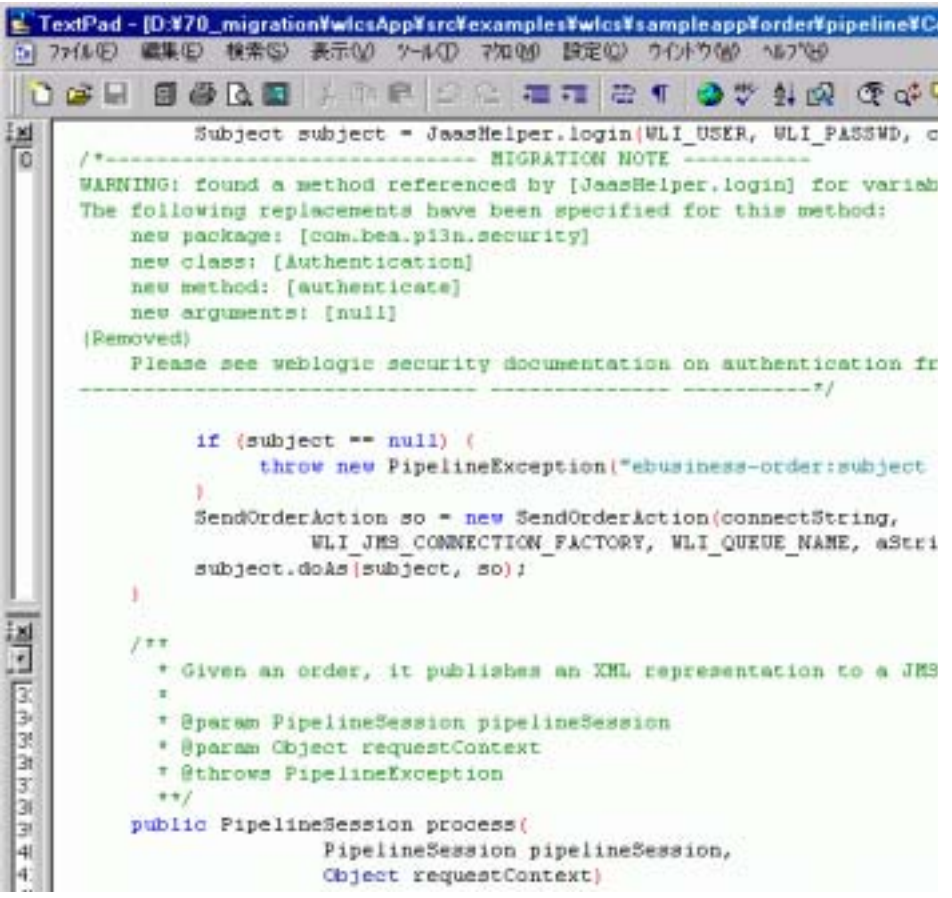
注意： 空の文（多くの場合、静的な初期化ブロックの直後）があるとエラーが報告されます。`migration.log` ファイルでエラーの詳細を確認してください。ログを分析すると、移行中に発生した問題の原因が明らかになります。空の文が原因でエラーが発生していた場合、空の文を取り除いてからコード移行をもう一度実行します。

8. 移行先ディレクトリで、移行されたそれぞれのファイルについて挿入されたコメントを確認し、必要であればコメントに従ってさらに変更を行います。

注意： Migration Viewer を使用すると、ファイルの移行状況および移行後の各ファイルに挿入されたコメントに基づいて、どのような変更をどのようにして行う必要があるかを判断するのに役立ちます。Migration Viewer の使用方法の詳細については、A-1 ページの「Migration Viewer ツールによるコードへの変更の参照」を参照してください。

移行されたコード ファイルと挿入されたコメントの例を図 2-8 に示します。

図 2-8　コメントが挿入された移行後のコード ファイル



9. 各ファイルの移行が完了したら、移行ユーティリティのウィンドウで、そのファイルの [**Reviewed**] カラムがチェックされていることを確認します。

JSP の移行

この節では、リリース 7.0 での JSP (JavaServer Pages) への変更点と、それらの変更に対応して行う必要のある対処について説明します。

注意： 移行が完了した後で、2-34 ページの「Webflow と JSP をコンフィグレーションする」の情報を基に、JSP に対してどのようにして変更を行うかを検討する必要があります。

1. それぞれの JSP に対して .java ファイルを作成します（2 通りある作成方法については、2-29 ページの「JSP の .java ファイルの生成」を参照してください）。

   アプリケーションの WEB-INF ディレクトリで始まるパスに .java ファイルが作成されます。たとえば、JSP の位置が mywebapp\myJSPs*JSPfilenames*.JSP である場合、生成される .java ファイルは WEB-INF_tmp*\jsp_compiled_myJSPs**JSPfilename*.java になります。

2. 2-24 ページの「コードの移行」の手順に従って、生成された .java ファイルに対してコード移行ツールを実行します。移行先ディレクトリとして、PORTAL_HOME ディレクトリの外部のディレクトリを選択します。

   移行された JSP の確認と変更を行う前に、PORTAL_HOME ディレクトリ内の適切なディレクトリに JSP を移動する必要があります。変更内容が有効であることを検証するには、JSP をリリース 7.0 の環境に配置する必要があります。通常、この配置先は <PORTAL_HOME>\applications\wlcsApp-project*webappdir*\ です。

3. 移行されたそれぞれの .java ファイルについて、挿入されたコメントと行われた変更を確認します。.java ファイルに挿入された移行に関する注記（コメント）には、JSP 内で対応する箇所の行番号が示されます。各 .java ファイルの先頭には、対応する JSP ファイルの名前とパスが付加されます。

4. JSP に必要な変更を行います。

コード移行ツールで変換または警告が発生しなくなるまで、上記の手順を繰り返します。

JSP の .java ファイルの生成

2 通りの方法で行うことができます。

- web.xml ファイルに次の内容を追加してから、それぞれの JSP を開きます。

```
<!-- JSP configuration -->
  <jsp-descriptor>
```

```
 <jsp-param>
   <param-name>keepgenerated</param-name>
   <param-value>true</param-value>
 </jsp-param>
</jsp-descriptor>
```

- **手早い方法** jspcPrepare.bat ファイルを使用して、この処理を自動で行うことができます。このファイルは <PORTAL_HOME>\migration\bin ディレクトリに収められています。ファイルはこのディレクトリ内に示されます。ファイル内で適切なパスを編集して jspsPrepare.bat を実行するか、または、適切な値を指定してコマンドラインから各コマンドを個別に実行することができます。

ステップ 4: その他の WebLogic 製品の移行

前のバージョンの WebLogic Server、WebLogic Integration、またはその他の WebLogic 製品を使用している場合、ここでそれらの製品を移行することをお勧めします。詳細については、http://edocs.beasys.co.jp/e-docs/platform/docs70/index.html にある、リリース 7.0 への移行に関するマニュアルを参照してください。

WebLogic Platform 7.0 マニュアルの参照

WebLogic Platform 7.0 のマニュアルには、ドメイン別に組織される新しいディレクトリ構造など、WebLogic Server、WebLogic Integration、および WebLogic Workshop に関する、重要かつ有益な情報が収められています。詳細については、http://edocs.beasys.co.jp/e-docs/platform/docs70/index.html を参照してください。

ステップ 5: 移行されたファイルのアセンブルと追加のコンフィグレーション タスクの実行

すべてのファイルが移行された後で、WebLogic Portal 7.0 環境で機能するように、以下の手順でファイルを配置し、コンフィグレーションする必要があります。

1. すべてのファイルとデータベースが移行されたことを確認する
2. 新しい 7.0 プロジェクト ディレクトリを作成する
3. 新しいドメインにファイルを移動する
4. Webflow と JSP をコンフィグレーションする
5. 各ドメインの起動スクリプトでデータベース接続変数の値を設定する
6. mem_args 変数の値を設定する
7. 起動スクリプトが WebLogic Server を呼び出す方式を変更する
8. config.xml ファイルを編集して 2 フェーズ Mbean デプロイメントを設定する
9. データベース情報を config.xml に追加する
10. コンフィグレーション ファイル内のハードコード化されたパスを更新する
11. EJB への参照をコンフィグレーション ファイルに追加する
12. weblogic-application ファイルをすべてのアプリケーションに追加する
13. 移行後の Web アプリケーション ディレクトリ構造に新しい JAR ファイルをコピーする
14. WebLogic Server アップグレード（移行）ガイドの指示に従う
15. Java ソースと EJB のビルドを完了する
16. Xerces を使用するためにクラスパスをコンフィグレーションする
17. 移行に関する補足情報をチェックする

すべてのファイルとデータベースが移行されたことを確認する

この節でのタスクを完了する前に、必要な移行がすべて完了していることを確認します。

- 2-2 ページの「ステップ 1: 移行前の環境およびファイルの準備」から 2-30 ページの「ステップ 4: その他の WebLogic 製品の移行」までの、コード移行およびデータ移行のすべてのタスクが完了していることを確認します。
- 2-4 ページの「移行関連の問題への対処」で説明されているすべての問題点に対処済みであることを確認します。
- コードの移行時に Migration Viewer を使用しなかった場合、ここで Migration Viewer を使用して、リリース 7.0 での API の変更点についてすべて確認し、実装環境で必要な対処を行ってください。詳細については、A-1 ページの「Migration Viewer ツールによるコードへの変更の参照」を参照してください。
- 必ず、WebLogic Platform 7.0 のマニュアル、特に、移行とこのリリースでの変更点および新機能に関する情報にもう一度目を通すようにしてください。

新しい 7.0 プロジェクト ディレクトリを作成する

リリース 7.0 ではディレクトリ構造が大きく変化しているため、以下で説明する手順に従って、新しいビルド環境を作成します。

1. すべてのビルド スクリプトを含む新しいプロジェクト ディレクトリを作成します。
2. 作成したプロジェクト ディレクトリで Configuration Wizard を実行し、プロジェクトに新しいドメインを作成します。作成するドメインは必ず WebLogic Portal ポータル ドメインとします。ウィザードを使用すると、さまざまな種類のドメインを作成できます。

   Configuration Wizard の詳細については、『コンフィグレーション ウィザードの使い方』(http://edocs.beasys.co.jp/e-docs/platform/docs70/confgwiz/index.html) を参照してください。

注意： 既存のディレクトリとファイルを再配置するのではなく、新しいドメインを作成*しなければなりません*。ユーザの作成や同期などのタスクを行うための Web アプリケーション ツールには大きな変更が加えられており、4.0 バージョンのツールはリリース 7.0 では使用できません。

新しいドメインにファイルを移動する

1. 移行されたすべての Java ソース ファイルを、新しいプロジェクトの所定のディレクトリにコピーします。
2. 移行を行っている Web アプリケーションのディレクトリを、Configuration Wizard で作成したエンタープライズ アプリケーション ディレクトリにコピーします。たとえば、Configuration Wizard で `DOMAIN_DIR/beaApps/portalApp` というディレクトリを作成したとします。この場合、Web アプリケーション ディレクトリを `portalApp` ディレクトリにコピーします。
3. ポータル Web アプリケーションでのみ使用する E-Business Control Center プロジェクト ファイルを、移行後の E-Business Control Center データ ファイル ディレクトリから、Configuration Wizard によって作成された E-Business Control Center プロジェクト ディレクトリに手動でコピーします。コピー先のディレクトリは `DOMAIN_DIR/beaApps/portalApp-project` です。対象の Web アプリケーションのファイルだけをコピーします。Web アプリケーションがポータルである場合にのみ存在する `tools` Web アプリケーションなど、その他の Web アプリケーションのファイルはコピーしません。次に示したのは、その内部の一部またはすべてのファイルをコピーすることが推奨されるディレクトリの例です。

   ```
   application-sync\pipelines
   application-sync\entitlements
   application-sync\entitlements\GlobalEntitlements
   application-sync\Portlets
   application-sync\userprofiles
   application-sync\webapps\your_web_app
   ```

ディレクトリ構造の詳細については、『*WebLogic Server 管理者ガイド*』(http://edocs.beasys.co.jp/e-docs/wls/docs70/adminguide/overview.html) を参照してください。ドメインの詳細については、『*WebLogic Server ドメイン管理*』(http://edocs.beasys.co.jp/e-docs/wls/docs70/admin_domain/index.html) を参照してください。

図 2-9 および図 2-10 に、ディレクトリ構造の変更の要旨を示します。これを参考にして、移行後のファイルを適切なディレクトリに配置し直してください。

図 2-9　WebLogic Portal 4.0 のディレクトリ構造における WebLogic Portal アプリケーション

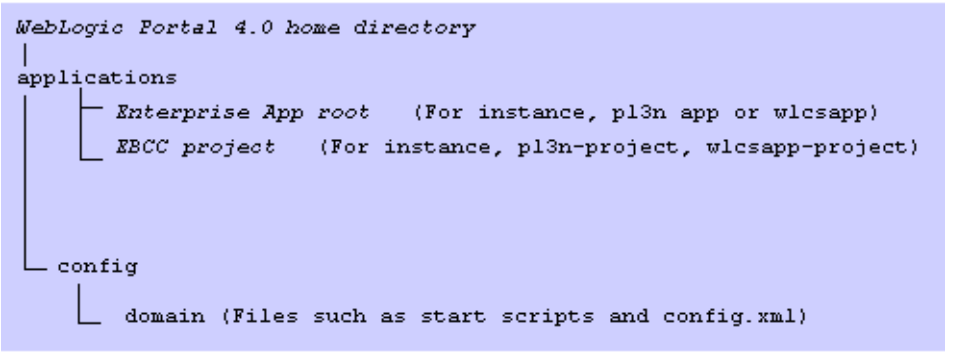
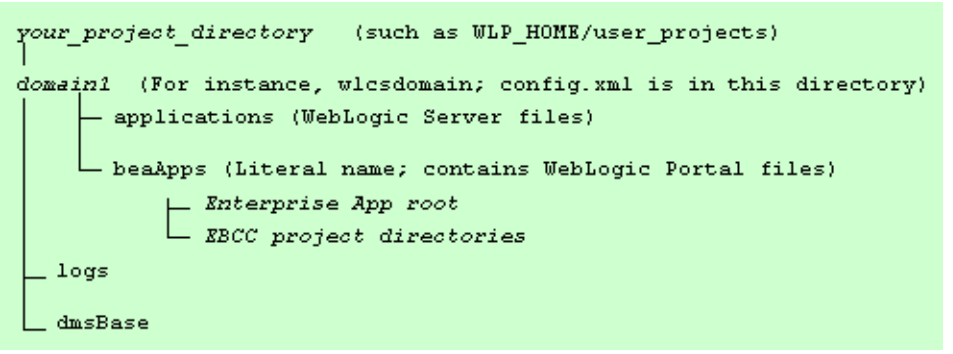


図 2-10　WebLogic Platform 7.0 のディレクトリ構造における WebLogic Portal アプリケーション



Webflow と JSP をコンフィグレーションする

この節で示す情報は、標準のポータル Webflow または JSP に対して行った変更との互換性を保ちながら、移行後のポータルに最新の WebLogic Portal 7.0 のポータル機能を取り入れるために役立ちます。これは手作業でのプロセスであり、その難易度は行った変更の内容によって異なります。

下の各節の指示に従って、適切なコンフィグレーションを行ってください。

コンフィグレーションの必要性の判断

Web アプリケーションでポータルを使用していなかった場合は、この節を読む必要はありません。Web アプリケーションにポータルが存在するかどうか分からない場合、`application-sync/webapps/yourWebAppName` ディレクトリに次のファイルがあるかどうかを確認してください。

- `portal.wf`
- `security.wf`
- `tools.wf`
- `user_account.wf`

これらのファイルが存在しない場合は、この節を読む必要はありません。存在する場合は、次の手順に進んでください。

はじめる前に

WebLogic Portal1 の必須ベース バージョンここでの説明は、WebLogic Portal 4.0 サービス パック 2 からの移行にのみ当てはまります。サービス パック 1 以前を使用している場合、まず SP2 にアップグレードする必要*があります*。SP2 とそれ以前のバージョンの間には、この節で説明する手順に影響する相違点があります。

この節で使用しているディレクトリ変数 この節では、`<BEA_HOME>/weblogic700/samples/portal/sampleportalDomain/beaApps` ディレクトリを指す目的で `<SAMPLE_DIR>` 変数を使用しています。

Webflow ファイルのコンフィグレーション

ポータルでは、WebLogic Portal 7.0 への移行時に更新しなければならない 3 つの Webflow ファイルを使用します。Webflow の名前は `security`、`tools`、および `user_account` です。Portal Web アプリケーションでは `portal` Webflow も使用されますが、このファイルについてはリリース 4.0 からの変更はありません)。

それぞれの Webflow は、拡張子が `.wf` の同名のファイルによって表されます。

以前にファイルを変更していない場合　何も行う必要はありません。新しいドメインと作成済みの Web アプリケーションで、新しいバージョンをそのまま使用できます。

以前にファイルを変更した場合　変更されたファイルに対して可能な処置が 2 通りあります。どちらの処置を行うかは、変更の度合いに応じて判断します。

- 必要に応じて、以前の変更内容を新しいファイルに適用できます。これは推奨される方針です。
- もう 1 つの方法として、リリース 7.0 での新しい変更点を手作業でファイルに適用し、作成した新しい Web アプリケーションにそれらのファイルをコピーすることができます。この方法を選んだ場合は、B-20 ページの「ポータル Webflow の変更点」の記述を参考にしてください。各ファイルに対して行う必要のある変更の一覧が示されています。

JSP ファイルのコンフィグレーション

ポータル フレームワークを構成する JSP ファイルの一部が、WebLogic Portal 4.0 から WebLogic Portal 7.0 で変更されています。Web アプリケーションを正しく機能させるためには、これらの変更を Web アプリケーションに適用しなければなりません。移行後の Web アプリケーションに追加しなければならない新しいファイルもいくつか存在します。

- 新しいディレクトリ
- 新しいファイル
- 修正されたファイル

新しいディレクトリ

作成した新しいドメイン内で、Web アプリケーションに 2 つの新しいディレクトリが追加されます。

```
<WEBAPP>\registration
<WEBAPP>\util
```

新しいファイル

新しいファイルが追加されています。これらのファイルに関して特に必要な作業はありません。

```
<WEBAPP>\framework\tools\change_name.jsp
<WEBAPP>\framework\tools\change_name.properties
```

修正されたファイル

次のリストは、WebLogic Portal 7.0 で更新されたファイルの一覧です。以下の対応が可能です。

- ユーザ独自のファイルの代わりにこれらのファイルを使用する。その場合、独自のファイルを新しいドメインおよび Web アプリケーションにコピーしません。
- これらのファイルに対して独自に変更を行った場合、これらのファイルをユーザ独自のファイルと統合する。

ファイルを統合する必要がある場合、コピーする前に個々のファイルを検証し、変更内容をそのまま保持する必要があるかどうかを判断します。ファイルに何らかの変更を行った場合、7.0 バージョンのファイルとの違いを解決するのはユーザ自身が行う作業となります。リリース 7.0 の新しいファイルに独自の変更を適用し直すのと、既存のファイルに 7.0 での変更を反映してから変更を行うのとでどちらが簡単かを判断します。

```
<WEBAPP>\framework\edit_titlebar.inc
<WEBAPP>\framework\header_links.inc
<WEBAPP>\framework\hnav_bar.jsp
<WEBAPP>\framework\maximize_titlebar.inc
<WEBAPP>\framework\minimize_titlebar.inc
<WEBAPP>\framework\normal_titlebar.inc
```

```
<WEBAPP>\framework\resourceURL.inc
<WEBAPP>\framework\titlebar.jsp
<WEBAPP>\framework\vnav_bar.jsp
<WEBAPP>\framework\security\badlogin.properties
<WEBAPP>\framework\security\login.properties
<WEBAPP>\framework\tools\portal_prefs.jsp
<WEBAPP>\framework\tools\portal_prefs.properties
<WEBAPP>\framework\tools\select_portal_pages.jsp
<WEBAPP>\framework\tools\select_portal_pages.properties
<WEBAPP>\framework\tools\select_portlets.jsp
<WEBAPP>\framework\tools\select_portlets.properties
<WEBAPP>\framework\tools\select_skins.properties
```

注意： 移行されたプロジェクトを開こうとして問題が発生した場合、`eaprj` をチェックします。「EnterpriseAppRoot」要素が大文字の E で始まっている場合、小文字に変更します。

各ドメインの起動スクリプトでデータベース接続変数の値を設定する

データベース接続情報を含むデータベース情報は、ドメインごとに固有です。WebLogic Platform 7.0 では、起動スクリプト内でデータベース情報が定義され、`set-environment.bat` または `set-environment.sh` が使用する変数に割り当てられるように変更されています。

したがって、WebLogic Server とその他の項目を呼び出す、`start-portal.bat` などの各起動スクリプトに対して、`set-environment` スクリプト内で呼び出される変数の定義を設定します。これは、Domain Wizard によって作成される `dbsettings.properties` ファイル内で、次のようにしてデータベース接続パラメータの値を編集することによって行います。

1. 使用しているデータベース用の情報を定義している行をコメント解除します。

2. データベース接続変数の正しい値を入力します（*@variablename@* の代わりに正しい値を入力します）。

mem_args 変数の値を設定する

mem_args 変数はこのリリースで新しく追加され、すべての起動スクリプトに出現します。この変数の値を設定する必要があります。値を設定しないと移行後のシステムが動作せず、サーバがハングしているように見えます。この変数は、WebLogic Server によるメモリの処理方法を設定します。

1. 適切な量のメモリを割り当てます。次の例は、メモリを 128MB RAM に設定する方法を示しています。

   ```
   MEM_ARGS="-Xms128m -Xmx128m -XX:MaxPermSize=128m"
   ```

2. アプリケーションごとに適した値を選択します。Hotspot VM エラーでサーバがロックアップまたはクラッシュする場合、この数値を大きくする必要があります。この現象は Hotspot VM の問題が原因で発生します。

起動スクリプトが WebLogic Server を呼び出す方式を変更する

これまでは、各起動スクリプトが WebLogic Server への Java 呼び出しを直接行っていました。WebLogic Platform 7.0 リリースでは、この方式が変更されました。個々の起動スクリプトが、次のようにしてサーバを呼び出す必要があります。

```
call %WLP_HOME%\bin\win32\startWebLogic.cmd
```

起動スクリプトのサーバ呼び出しを上の行で置き換えます。WLP_HOME 変数の値は、WebLogic Portal 7.0 のインストール時に設定されます。

config.xml ファイルを編集して 2 フェーズ Mbean デプロイメントを設定する

このリリースでの新機能に、2 フェーズ Mbean デプロイメントがあります。この機能については、このリリースの WebLogic Server のマニュアル (http://edocs.beasys.co.jp/e-docs/wls/docs70/index.html) で説明しています。

config.xml ファイルを編集して、この機能を取り入れるように WebLogic Portal をセットアップする必要があります。

コード リスト 2-4 に示すように、config.xml ファイルの Application タグに属性を追加します。

注意： 値は true または false に設定できますが、すべてのアプリケーションに対してこの属性が存在しなければなりません。

コード リスト 2-4　config.xml に追加する属性

```
<Application
  Deployed="true" TwoPhase="false"
  Name="p13nConsoleApp"
  Path="D:/bea/weblogic700/samples/portal/p13nDomain/beaApps/p13nConsoleApp">
  <WebAppComponent
    Name="p13nConsole"
    ServletReloadCheckSecs="300"
    Targets="p13nServer"
    URI="p13nConsole"
  />
</Application>
```

データベース情報を config.xml に追加する

適切なデータベース ドライバを使用するように、DOMAIN_DIR/config.xml の内容を修正します。手順については、『管理者ガイド』 (http://edocs.beasys.co.jp/e-docs/wlp/docs70/index.htm) を参照してください。

デフォルトの commercePool とは別にデータベース接続プールまたはデータ ソースが必要な場合、DOMAIN_DIR/config.xml に追加します。作成するプールごとに、DOMAIN_DIR/fileRealm.properties に適切な形式の行を必ず追加するようにします。たとえば、「oraclePool」という名前の接続プールを追加する場合、次のような行を fileRealm.properties に追加します。

```
acl.reserve.weblogic.jdbc.connectionPool.oraclePool=everyone
```

コンフィグレーション ファイル内のハードコード化されたパスを更新する

以下のすべてのファイルを検索し、新しいディレクトリ構造に符合するようにパスを更新します。

- config.xml
- application.xml
- application-config.xml
- weblogic-application.xml（このリリースでの新規ファイル）

EJB への参照をコンフィグレーション ファイルに追加する

EJB（存在する場合）と Web アプリケーションのエントリを次のファイルに追加します。

```
<DOMAIN_DIR>/beaApps/portalApp/application.xml
<DOMAIN_DIR>/config.xml
```

weblogic-application ファイルをすべてのアプリケーションに追加する

新しい weblogic-application.xml ファイルでは、WebLogic Portal アプリケーションで使用する WebLogic Server の機能を細かく設定することができます。

アプリケーションが動作するためには、このファイルが存在*していなければなりません*。このファイルをすべてのアプリケーションの META-INF ディレクトリにコピーします。コピー先の例を次に示します。

```
weblogic700\samples\portal\p13nDomain\beaApps\p13nApp\META-INF
```

このファイルを使用して設定を調整する方法については、WebLogic Server のマニュアル (http://edocs.beasys.co.jp/e-docs/wls/docs70/index.html) を参照してください。

移行後の Web アプリケーション ディレクトリ構造に新しい JAR ファイルをコピーする

アプリケーションに固有でないすべての JAR ファイルを <migrated_webapp>/WEB-INF/lib ディレクトリから削除し、次のディレクトリにあるすべての JAR ファイルで置き換えます。

```
<WLS_HOME>/weblogic700/samples/portal/sampleportalDomain/beaApps/sampleportal/sampleportal/WEB-INF/lib
```

WebLogic Server アップグレード（移行）ガイドの指示に従う

http://e-docs.beasys.co.jp/e-docs/wls/docs70/upgrade/index.html を参照して、WebLogic Server のアップグレードに関する指示に従います。CMP エンティティ Bean の記述子内の EJB-QL の変更点に特に注意します。

Java ソースと EJB のビルドを完了する

Java ソースと EJB のビルドを完了します。ビルドを実行するためにはビルド スクリプトを使用しなければなりません。

Xerces を使用するためにクラスパスをコンフィグレーションする

ソース ファイルで Xerces パーサを使用する場合、JAR ファイル `<WLS_HOME>/weblogic700/lib/xerces.jar` をサーバの起動クラスパスに追加します。

移行に関する補足情報をチェックする

移行に関する補足情報は、カスタマサポート (http://websupport.bea.com/welcome.jsp) および BEA dev2dev (http://developer.bea.com/index.jsp) の各 Web サイトで公開されます。

ステップ 6: 4.0 から 7.0 への移行の検証

サーバを起動します。E-Business Control Center から、または EBCC プロジェクト ディレクトリにある `sync.bat` ファイルを使用して、datasync を実行します。

ポータル Web アプリケーションが正しく同期されることを確認します。ただしこの後も、グループ ポータルや、ポートレットの可視性およびレイアウトなどのその他の設定を、JSP 管理ツールを使用して再度コンフィグレーションする必要があります。

正しく移行されていれば、この時点でポータルにログインすることができます。

ステップ 7: 次に行うこと

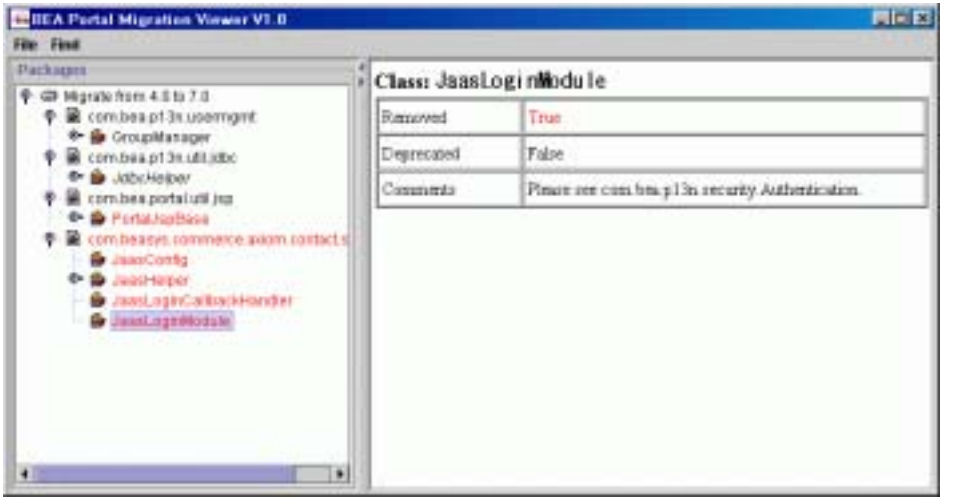
以上で、一通りの移行作業が完了しました。次のステップに進み、移行の細部をチェックして WebLogic Portal 7.0 の使用を開始してください。

オンライン ドキュメント サイト (http://edocs.beasys.co.jp/e-docs/wlp/docs70/index.htm) にアクセスし、製品をより詳しく理解するために必要なマニュアルを入手してください。

付録 A　Migration Viewer ツールによるコードへの変更の参照

このリリースに付属する Migration Viewer は、WebLogic Portal 7.0 での API の変更に関する情報をすばやく、容易に参照するためのツールです。このビューアを使用すると、変更のうちどの程度が実装に影響するか、また、移行に際してどの程度の作業量が必要かを容易に判断することができます。図 A-1 にビューアの画面例を示します。

図 A-1　Migration Viewer ツール



注意： Migration Viewer から参照できるこの情報は、HTML ファイル（`migrinfo.html`）にも収められています。

Migration Viewer の主な機能には、次のものがあります。

- 変更点を容易に調べることのできる GUI 環境
- 変更の種類を示す視覚効果（非推奨は斜体、削除は赤色）

- パッケージ、クラス、フィールド、およびメソッドを示すアイコン
- 名前およびコメントを対象とした、パッケージ、クラス、フィールド、およびメソッドのキーワード検索機能

Migration Viewer の構成ファイル

zip ファイルには以下のファイルが収められています。

- `<PORTAL_HOME>/migration/lib/migration_viewer.jar`: 必要なクラス ファイルと画像
- `<PORTAL_HOME>/migration/bin/viewer.bat`: ビューアを起動するための Windows バッチ スクリプト
- `<PORTAL_HOME>/migration/bin/viewer.sh`: ビューアを起動するための UNIX シェル スクリプト

Migration Viewer の使用方法

ビューア内では、主に 2 種類のタスクを実行できます。

- メインの GUI ウィンドウを使用して、一覧表示されたパッケージ、クラス、フィールド、メソッドについての情報を参照する
- [Find] ウィンドウを使用して、特定のキーワードを検索する

メイン ウィンドウでの情報の参照

メイン ウィンドウで情報を参照するには、次の手順に従います。

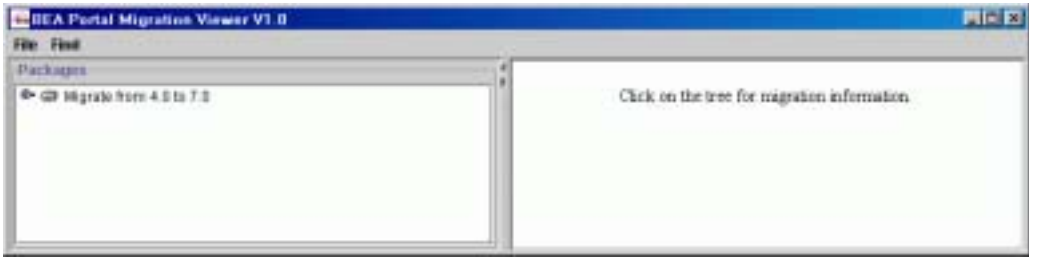
1. 次のファイルを実行して、Migration Viewer を起動します。

   Windows: `<PORTAL_HOME>\migration\bin\viewer.bat`

   UNIX: `<PORTAL_HOME>/migration/bin/viewer.sh`

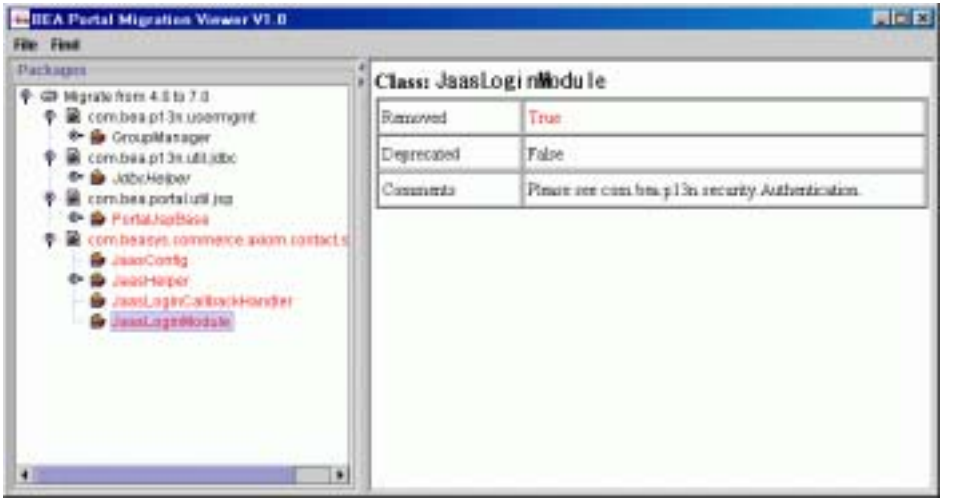
図 A-2 のように Migration Viewer のウィンドウが表示されます。

図 A-2　Migration Viewer ツール — 初期画面



2. 左パネルで、[Migrate from 4.0 to 7.0] の隣にある鍵のアイコンをクリックします。この移行で使用されるパッケージが左パネルに、選択された項目についての情報が右パネルにそれぞれ表示されます。
3. 図 A-3 に示すように、アイコンを展開して、情報を参照したいパッケージ、クラス、フィールド、およびメソッドを選択します。

図 A-3　Migration Viewer ツール — 項目選択後の画面



左パネルの各視覚効果の意味は次のとおりです。

- 赤色のテキストは削除された項目を示します。

- 斜体のテキストは非推奨になった項目を示します。
- 鍵のアイコンはパッケージを示します。
- 箱のアイコンはクラスを示します。
- 「**M**」アイコンはメソッドを示します。
- 「**f**」アイコンはフィールドを示します。

[Find] ウィンドウでのキーワード検索

API の変更に関する説明の中から特定のキーワードを検索する場合、検索機能を使用します。

1. Migration Viewer の [Find] メニューから [**Find**] を選択します。

   図 A-4 に示すように [Find] ウィンドウが表示されます。

図 A-4　Migration Viewer ツール — 検索のステップ 1

2. [**Find**] フィールドにキーワードを入力します。

3. キーワードを検索しないオブジェクト タイプまたは検索先のチェック マークを外します。
4. [**Search**] をクリックします。コード リスト A-5 に示すように、検索結果が [Results] ウィンドウに表示されます。

図 A-5　Migration Viewer ツール — 検索のステップ 2

5. [Results] ウィンドウ内の項目をダブルクリックすると、その項目についての情報が Migration Viewer のメイン ウィンドウに表示されます。

付録 B 移行ファイル

migration ディレクトリには、移行プロセスで使われるプログラムおよび補助ファイルが収められています。この節では、E-Business Control Center プロジェクトの移行に関する情報を示します。

- migration_install.properties ファイル
- migrator.bat および migrator.sh ファイル
- MigratorBundle.properties および DataMigratorBundle.properties ファイル
- 移行ログ ファイル
- カスタマイズされたデータベースを変換する .sql ファイル
- API の変更点を Migration Viewer と migrinfo.html で確認する
- E-Business Control Center プロジェクト移行の詳細
- ポータル Webflow の変更点

migration_install.properties ファイル

ルートの migration ディレクトリ内の `migration_install.properties` ファイルでは、データベースの種類やデータベース接続に関するデータなど、移行ツールを実行するために必要な情報を指定します。コード リスト B-1 にこのファイルの例を示します。

コード リスト B-1　MigratorInstall.properties

```
######################################################################
## PROPERTIES TO BE SET BY THE USER AT INSTALL TIME
######################################################################
# ----------------------------------------------------------
```

```
##       When you have set the properties you need, set the
#       following flag to 'true' so the migrator tool will start.
#       Otherwise, the Migrator will assume you have NOT set
#       any property and will refuse to run.
## ---------------------------------------------------------------
start_migrator=false
# ---------------------------------------------------------------
##Database Properties
## ---------------------------------------------------------------
# Database connection properties
#------Oracle Thin Driver----------------------#
#
# For oracle, replace the following:
#    @USER@, @PASSWORD@, @SERVER@, @PORT@, and @SID@
#    e.g. jdbc:oracle:thin:@localhost:1521:ORCL
#
database.connection.driver = oracle.jdbc.driver.OracleDriver
database.connection.url = jdbc:oracle:thin:@@SERVER@:@PORT@:@SID@
database.connection.props = user=@USER@;password=@PASSWORD@
#------MS SQL Server ----------------------#
#
# For SQL Server, replace the following:
#    @SERVERPORTNUMBER@, @USER@, @PASSWORD@, and @SERVER@ (in two different
locations)
#
#database.connection.driver = weblogic.jdbc.mssqlserver4.Driver
#database.connection.url =
jdbc:weblogic:mssqlserver4:@SERVER@:@SERVERPORTNUMBER@
#database.connection.props =
user=@USER@;password=@PASSWORD@;server=@SERVER@;weblogic.t3.waitForConnection=t
rue;weblogic.t3.waitSecondsForConnection=9999999999999;weblogic.jts.waitSecondsF
orConnectionSecs=999999999999
#------Sybase jConnect 5.2 ----------------------#
#
# For Sybase, replace the following:
#    @SERVER@, @PORTNUMBER@, @USER@, and @PASSWORD@
#
#database.connection.driver=com.sybase.jdbc2.jdbc.SybDriver
#database.connection.url=jdbc:sybase:Tds:@SERVER@:@PORTNUMBER@
#database.connection.props =
user=@USER@;password=@PASSWORD@;server=@SERVER@;weblogic.t3.waitForConnection=t
rue;weblogic.t3.waitSecondsForConnection=9999999999999;weblogic.jts.waitSecondsF
orConnectionSecs=999999999999
#------IBMS's DB2 ----------------------------------------#
#
# For DB2, replace the following:
#    @DB2_DATABASE@, @USER@, and @PASSWORD@
#
#database.connection.driver=COM.ibm.db2.jdbc.app.DB2Driver
```

```
#database.connection.url=jdbc:db2:@DB2_DATABASE@
#database.connection.props =
user=@USER@;password=@PASSWORD@;server=@SERVER@;weblogic.t3.waitForConnection=t
rue;weblogic.t3.waitSecondsForConnection=999999999999;weblogic.jts.waitSecondsF
orConnectionSecs=999999999999
# Database and version
# for Oracle users use database.version=817 for either Oracle 8.1.7 or 9i
database.name=oracle
database.version=817
#database.name=sql_server
#database.version=2000
#database.version=7
#database.name=sybase
#database.version=12
#database.name=db2
#database.version=7
# This is used to continue a task even if required statements fail.
# Use it to get a longer list of failed actions
database.continue.despite.failure=false
```

migrator.bat および migrator.sh ファイル

migrator.bat（Windows）および migrator.sh（UNIX）の各スクリプトは移行ツールを起動します。

コード リスト B-2 は migrator.bat ファイルの例です。

コード リスト B-2　migrator.bat ファイルの例

```
echo off
REM -------------------------------------------------------#
REM  Migrator Starter Script on Windows                     #
REM -------------------------------------------------------#
SETLOCAL
set DATABASE=ORACLE_THIN
REM set DATABASE=MSSQL
REM set DATABASE=SYBASE_JCONNECT
REM set DATABASE=DB2_TYPE2
if "%DATABASE%" == "" ( echo "The DATABASE variable must be uncommented in
migrator.bat"
    exit 1 )
```

```
CALL ..\..\bin\win32\set-environment.bat
REM ------------------------------------------------------#
REM         VARIABLES TO SET
REM ------------------------------------------------------#
REM Due to database specifics, you may have to set your
REM path to include dll
REM ------------------------------------------------------#
REM          The mini script
REM ------------------------------------------------------#
set MIGRATION_DIR=%WL_COMMERCE_HOME%\migration
set MIGRATION_LIB=%MIGRATION_DIR%\lib
set
MIG_CLASSPATH=%BEA_HOME%\lib\tools.jar;%MIGRATION_LIB%\migration.jar;%MIGRATION
_LIB%\apache\xerces-1_4_3\xerces.jar;%MIGRATION_LIB%\apache\xalan-j_2_0_1\xalan
.jar;%MIGRATION_LIB%\p13n_system.jar;%WEBLOGIC_HOME%\lib\weblogic.jar;%BEA_HOME
%
REM ------------------------------------------------------#
REM For testing the setup only
REM ------------------------------------------------------#
REM echo BEA_HOME=%BEA_HOME%
REM echo WEBLOGIC_HOME=%WEBLOGIC_HOME%
REM echo WL_COMMERCE_HOME=%WL_COMMERCE_HOME%
REM echo MIGRATION_DIR=%MIGRATION_DIR%
REM echo MIGRATION_LIB=%MIGRATION_LIB%
REM echo MIG_CLASSPATH=%MIG_CLASSPATH%
%JDK_HOME%\bin\java -cp %MIG_CLASSPATH%  -DMIGRATION_DIR=%migration_dir%
com.bea.commerce.migration.tools.Migrator
echo on
```

MigratorBundle.properties および DataMigratorBundle.properties ファイル

移行プロセスの設定の多くは、MigratorBundle.properties および DataMigratorBundle.properties ファイルに格納されます。これらのファイルは、<PORTAL_HOME>\migration\lib ディレクトリ内の migration.jar ファイルに収められており、移行の間に次の場所に展開（unjar）されます。

```
<MIGRATION_HOME>\com\bea\commerce\migration\tools\
```

- DataMigratorBundle.properties には、データ移行タスクの設定の記述が含まれます。このファイルには、タスクがスキップ可能かどうか、つまり、そのタスクを完了せずに別にタスクに進むことができるかどうかの設定が含まれます。タスクをスキップ可能にする以外の理由で、このファイルを編集することはほとんどありません。
- MigratorBundle.properties には、コード移行プロセスの設定が含まれます。このファイルには、移行ツールの GUI 文字列やエラー メッセージなど、データおよびコード移行ツールの設定が含まれます。国際化を行う必要がある場合を除いて、このファイルを編集する必要は通常ありません。

注意： これらのファイルを展開または修正した場合、ファイルを再度 JAR ファイルにアーカイブ化する必要があります。

コード リスト B-3 とコード リスト B-4 にこれらのファイルの例を示します。

コード リスト B-3　MigratorBundle.properties

```
#########################################################################
#  MIGRATOR's Resources
#########################################################################
#########################################################################
# Code Migration Mapper: Versions available
#
#  The structure points to XML; however, this
#  is all internal here. If we decided to go
#  for an actual product, let's put this in
#  a 'migratorconfig.xml' file.
#  Structure: versions give you all the roots, separated by commas: ","
#             for each root, there are two resources:
```

```
#              1. menuname, appears in the version combo-box
#              2. resource, is used to initialize the Mapper for Code Migration
#
#########################################################################

# example with many versions:
mapper_resource=com.bea.commerce.migration.code.version.v4_0to7_0.MapperBundle


#########################################################################
#
#  PROPERTIES FOR THE GUI
#
#########################################################################
title=  BEA Portal Migrator V2.0
migratorWelcome= Welcome to the BEA Portal Migrator, V2.0
migratorIcon=/com/bea/commerce/migration/tools/images/migratorIcon.gif

#
# For the Migrator Menu bar
#
helpURL=http://www.bea.com/
fileMenu=File
fileAccess=To exit this tool.
exitMenu=Exit
exitAccess=To exit this tool.
HelpMainMenu=Help
HelpMainAccess=To get help on this tool

overviewHelp=Overview and Strategy
overviewAccess=What is migration and how to use it

codeMigratorHelp=Code Migrator Help
codeMigratorAccess=How to use the code migrator
codeMigratorHelpURL=http://www.bea.com/

dataMigratorHelp=Data Migrator Help
dataMigratorAccess=How to use the data migrator
dataMigratorHelpURL=http://www.sun.com/

# SHOULD BE FOUND UNDER '<migrator_dir>\doc'
overviewHelpURL=mainhelp.html

#
# The log message on starting a session
#
startLogMessage= ------------------ New Migration Session ---------------
```

```
#########################################################################


#
# Code Migration GUI
#
#########################################################################

#
# JTable Columns/yes-no
#
fileNameColumn=File Name
modifiedColumn=Modified
errorColumn= Errors
reviewedColumn=Reviewed

yes=Yes
no=No

#
# Code Migration: directory selection
#
codeMigratorTitle = Code Migrator

codeMigratorDirTitle = Migration settings
originalDir    = Original source directory :
destinationDir = Migrated source directory :
chooseDir = Browse ...
chosenDir = Using directory:
noChosenDir = No directory selected
dirFilter = directories
startMigration = Start Code Migration Helper

#
#  External Viewer's command
#
external_viewer_command=c:\\winnt\\system32\\write %f%

#
# Code Migration: Analyzed file results
#
codeMigrationResultTitle = Result of Code Migration
analysisProgress = Progress of Files Analysis
externalViewerLabel=External Viewer Command (%f% for filename):
reviewed= {0} file(s) reviewed out of {1}
analyzed= {0} file(s) analyzed out of {1}

#
```

```
# Code Migration: Recommendation Strings
#
migratorChangeHeader=// WARNING: THE MIGRATOR HELPER MODIFIED THIS FILE, MAKE SURE
TO REVIEW THE CHANGES


migratorNote   =/*----------------------------- MIGRATION NOTE ----------
migratorNoteEnd=------------------------------ -------------- ----------*/\n
originalLines=-> original line(s):
importReplaced=In the ''import'' statement, replaced [{0}] with [{1}]
importWarning=WARNING: In the ''import'' statement, the package or class [{0}]
castReplaced=In the ''cast'' statement, replaced [{0}] with [{1}]
castWarning=WARNING: In the ''cast'' statement, the class [{0}]
instanceofReplaced=In the ''instance of'' statement, replaced [{0}] with [{1}]
instanceofWarning=WARNING: In the ''instance of'' statement, the class [{0}]
newObjectReplaced=In the ''instantiation'' (new) statement, replaced [{0}] with
[{1}]
newSameObject=In the ''instantiation'' (new) statement, the class [{0}]
varDeclarationReplaced=In the ''variable declaration'' statement, replaced [{0}]
with [{1}]
varDeclarationSameObject=In the ''variable declaration'' statement, the class
[{0}]
returnReplaced=In the ''return statement'', replaced [{0}] with [{1}]
returnWarning=WARNING: In the ''return statement'', , the class [{0}]
objectReferenceWarning=WARNING: the variable reference by [{0}] is of type [{1}]
fieldReplaced=In the ''field reference'' statement [{0}], variable of type [{1}],
replaced with [{2}]
fieldReferenceWarning=WARNING: found a field referenced by [{0}] for variable type
[{1}]
methodReferenceWarning=WARNING: found a method referenced by [{0}] for variable
type [{1}]
methodReplaceMessaqe=The following replacements have been specified for this
method:\n\tnew package: [{0}]\n\tnew class: [{1}]\n\tnew method: [{2}]\n\tnew
arguments: [{3}]
implementsDanger=DANGER: this class implements [{0}] now moved to: [{1}]
implementsWarning=WARNING: this class implements [{0}] see notes below
extendsDanger=DANGER: this class extends [{0}] now moved to: [{1}]
extendsWarning=WARNING: this class extends [{0}] see notes below
removedStr=(Removed)
deprecatedStr=(Deprecated)
noAdditionalDoc=[There was no additional documentation available regarding this
change]

#
# Code Migration: Result Strings
#
level0= 0 - no change
level1= 1 - comments
level2= 2 - modified
```

```
level3= 3 - danger

#########################################################################
#
# Data Migration GUI
#


#########################################################################

dataMigratorTitle= Data Migrator

taskProgress= Task #{0} out of {1} Tasks

apply=Execute Task
back=<= Previous Task
next=Next Task =>

taskPanelTitle= Task
description=Description
status=Status

taskListTitle= Order of Tasks

startTask= \n -------------- Start of {0} ------------------ \n
endTask  = \n --------------  End of {0}  ------------------ \n


#########################################################################
#
# Data Migration - EBCCDataMigration 4.0 to 7.0 specific
#
#########################################################################

# String for the EBCCDataMigDialog
ebcc_dialog_title=EBCC Project Migration
ebcc_proj_scr_dir=EBCC Project Source Directory:
ebcc_proj_dest_dir=EBCC Project Destination Directory:
ejb_app_root_dir=Enterprise Application Root Directory:
ebcc_dialog_ok_button_text=OK
ebcc_dialog_cancel_button_text=Cancel

# String for the EBCCDataMigration task
app_sync_dir_name=application-sync
failed_verify_dir_name=failed-verification
ebcc_proj_file_ext=eaprj
proj_name_token=PROJ_NAME_TOKEN
app_root_token=APP_ROOT_TOKEN
encoding_token=ENCODING_TOKEN
```

```
ebcc_proj_file_template=<?xml version="1.0"
encoding="ENCODING_TOKEN"?>\n<project name="PROJ_NAME_TOKEN" version="1.0">\n\t
\

<server>\n\t\t<EnterpriseAppRoot>APP_ROOT_TOKEN</EnterpriseAppRoot>\n\t</server
>\n</project>

#########################################################################


#
# Data Migration Strings
#
#########################################################################

xml_schema_namespace=http://www.w3.org/2001/XMLSchema-instance
properties_to_XML_schema=properties-to-xml-1_0_1.xsd


# specify the document encoding to be used by the data migration tool when
# writing XML file, if not set here the default is UTF-8
xml_document_encoding=UTF-8

#
# For resources as Stream, will find xsl in .jar file at
# right location
#
xsl_resource = xsl/

#
# the relative location of the XML schemas
#
schema_location = /lib/schema/7_0

#
# Whenever a task needs an 'OK' flag, here it is
#
taskOK= : The task completed successfully.
taskError= : *** The task compeleted with errors (check migration.log file) or
was aborted ***

#
# For WebFlowMigrate and PipelineMigrate Caller.
#

genericConversion=Convert "{0}" to a generic properties XML format.
fileNotFound=Could not find: "{0}".
genericOK=The generic XML "{0}" was written out without error.
validation=Validate the file: "{0}".
```

```
validationError=Error: validating the file:"{0}".
validationOK=OK: validating the file:"{0}".
specificConversion=Convert the generic file, "{0}",  to a specific file, "{1}".
storing=Store result to: "{0}".

webflowEndMessage=Copy the file "{0}" to the proper WebApp location, and use \n \
   the Webflow Editor to modify and set up the Webflow for that WebApp. \n
  Once the Webflow is properly set up, use data sync to push it to the appropriate
server.



pipelineEndMessage=Copy the file "{0}" to the proper WebApp location, and use \n \
   the Webflow Editor to modify and set up the pipelines for that WebApp. \n \
  You will also want to move the tracking pipelines to a ''tracking'' namespace,
in \
   a file called "tracking.pln". \
   Once the pipelines are properly set up, use data sync to push it to the
appropriate server.


migrationError=There was an error with this migration task. \
            Please refer to the log file (tmp\migration.bat) for more information.

#
# Schema/Property Set migration strings
#
schemaError=Error while retrieving data from WLCS_SCHEMA.\n Cannot continue.
schemaOK=Retrieved data from WLCS_SCHEMA: OK.
schemaMigrationComplete=\nThis task completed satisfactorily.\n Please review the
messages and take any eventual corrective actions.
schemaBadType=Check for WLCS_PROP_MD properties of type "5" or "6", which are not
migrated by this tool. \n
schemaUnmigratedProps=\nA number of schemas were not migrated due to their
SCHEMA_GROUP_NAME.\n \
                      You will need to deal with them on a case by case basis,
checking in \n  \
      the WLCS_SCHEMA table, using the SCHEMA_GROUP_NAME. \n \
      The following list gives you an overview:
schemaUnmigrated=\t{1} Schemas not migrated, group name = "{0}" .

######################################################################
#
# Logger Levels (prepended to log messages based on log 'type'
#
######################################################################
logLevel1=Message:
logLevel2=Warning:
logLevel3=Error  :
```

```
#####################################################################
#
# Error Messages
#
#####################################################################
errorTitle=Error

editInstallFileError=Before running the Migrator, \n "{0}" must be edited
correctly.

URLerror=Could not open browser properly. \nCheck

http://edocs.bea.com/index.html.

mapperError= The Code Migrator''s configuration file (mapper) could not be
initialized!
sameDirError=Source and destination CANNOT be the same directory.
isNotDirError=: is NOT a directory.

viewerError=Could NOT spawn viewer on file with:
readError=Cannot read the file \n
createError=Cannot create the file \n
writeError=Cannot write to the file \n

storeConfig=store configuration settings
restoreConfig=restore configuration settings

restoreConfigError=Could not restore Code Migration settings from file:
storeConfigError=Could not store Code Migration settings to file:

storeAnalysisError=Could not persist the result of the analysis to file:
restoreAnalysisError=Could not read the result of the analysis from file:

restoreSettingsError=Could not read settings from file:
storeSettingsError  =Could not persist settings to file:

fileError= {0} could not be handled by the Code Migrator.\n Check the original
file.

dataMigratorResourceError=Could not read configuration for task {0}.
dataMigratorInitError = Could not initialize the Data Migrator:

error=* Error * :

#############################################################################
###
#
# Status and error messages for Data Migration
```

```
#
################################################################################
###

#
# for com.bea.commerce.migration.data.CallerBase
#
data.callerbase.validate.doc.name=\nValidating transformed version of {0}
data.callerbase.validate.failed=Validation of {0} failed\n
data.callerbase.validate.success=Validation of {0} succeeded\n
data.callerbase.validate.IOException=An IOException occurred during
validation...
data.callerbase.validate.ValidationException=A ValidationException occurred

during validation...
data.callerbase.validate.start=Performing validation

#
# for com.bea.commerce.migration.data.version.XMLcaller
#
XMLcaller.retrieveXML.DBConnectionException=A DBConnectionException occurred
while attempting to retrieve document(s)...
XMLcaller.retrieveXML.SQLException=A SQLException occurred while attempting to
retrieve document(s)...
XMLcaller.retrieveXML.IOException=An IOException occurred while attempting to
retrieve document(s)...
XMLcaller.retrieveXML.ParserConfigurationException=A
ParserConfigurationException occurred while attempting to retrieve document(s)...
XMLcaller.retrieveXML.SAXException=A SAXException occurred while attempting to
retrieve document(s)...
XMLcaller.retrieveXML.UnsupportedArgumentTypeException=One or more invalid
arguments where used while attempting to retrieve document(s)...
XMLcaller.transform.doc.InvalidTransformerStateException=An
InvalidTransformerStateException occurred while attempting to transform
document(s)...
XMLcaller.transform.doc.TransformerFailureException=An
InvalidTransformerStateException occurred while attempting to transform
document(s)...
XMLcaller.transform.docs.InvalidTransformerStateException=An
InvalidTransformerStateException occurred while attempting to transform
document(s)...
XMLcaller.transform.docs.TransformerFailureException=An
InvalidTransformerStateException occurred while attempting to transform
document(s)...


#
# for
com.bea.commerce.migration.data.version.v4_0to7_0.EntitlementRulesetMigration
```

```
#
entitlement_ruleset.exe.retrieve=Retrieving ENTITLEMENT_RULESET entries...
entitlement_ruleset.exe.found=There are {0} ENTITLEMENT_RULESET entries to
migrate.
entitlement_ruleset.exe.transform.begin=Beginning transformation of
ENTITLEMENT_RULESET entries...
entitlement_ruleset.exe.transform.end=Ending transformation of
ENTITLEMENT_RULESET entries...
entitlement_ruleset.exe.validate.begin=Beginning validation of
ENTITLEMENT_RULESET documents...
entitlement_ruleset.exe.validate.success=All ENTITLEMENT_RULESET documents
validated...
entitlement_ruleset.exe.validate.failed=Some ENTITLEMENT_RULESET documents
failed validation. Check log file for errors.


entitlement_ruleset.exe.saving=Persisting transformed ENTITLEMENT_RULESET
documents.
entitlement_ruleset.exe.rule_set.not.found=No ENTITLEMENT_RULESET documents
found.
entitlement_ruleset.exe.abort=Terminating Entitlement Ruleset Migration due to
failure.
entitlement_ruleset.update.missing.clob=The ENTITLEMENT_RULESET row specified by
APPLICATION_NAME [{0}] and RULESET_URI [{1}] had no existing RULESET_DOCUMENT and
could not be updated.  There may be a data integrity problem in your orginal
ENTITLEMENT_RULESET data.

#
# for com.bea.commerce.migration.data.version.v4_0to7_0.EBCCDataMigration
#
EBCCDataMigration.exe.abort=Aborting migration task...
EBCCDataMigration.exe.failed=Migration task failed...
EBCCDataMigration.processAuto.create.projectfile=Creating EBCC project file
EBCCDataMigration.processAuto.starting.mig=Beginning migration of EBCC project
named: {0}\n
EBCCDataMigration.processAuto.end.mig=Finished migration of  EBCC project named:
{0}
EBCCDataMigration.processFile.copyfile=Copying file with unknown type named: {0}
EBCCDataMigration.processFile.processfile=Processing file named {0}
EBCCDataMigration.processFile.validation.success=Validation success for
transformed doc named: {0} \n
EBCCDataMigration.processFile.validation.failed=Validation failed for
transformed doc named: {0} \n
EBCCDataMigration.processFile.skip.validation=Skipping validation for the
document named: {0}.  The document may not be complete.\n
EBCCDataMigration.main.success=Successful completion
EBCCDataMigration.main.failed=Task failed
EBCCDataMigration.processCommandLine.invalid.num.args=Invalid number of
arguments...
```

```
EBCCDataMigration.processCommandLine.invalid.arg.value=The command line argument
{0} had an null or zero length value
EBCCDataMigration.processCommandLine.invalid.arg=The following invalid command
line argument was encountered: {0}
EBCCDataMigration.printUsage.usage=Usage :
```

コード リスト B-4 DataMigratorBundle.properties

```
######################################################################
#
#  Data Migrator's  Task Specification
#
######################################################################

#
# To create a new task:
#  1. add it to the list of tasks (tasks=...) in the right order
#  2. copy another task and replace the root (as in the task list)
#     and set the right values for those elements
#  3. in the DESCRIPTION of the task, state whether it is required
#     task and indicate alternatives if that task is skippable.
# The list of all tasks: the ORDER is quite important, as
# it determines the dependency of tasks
#
tasks=EBCCDataMigration, rdbms-data-sync, rdbms-portal,
rdbms-entitlement-ruleset, EntitlementRulesetMigration
#
# The EBCCDataMigration task
#
EBCCDataMigration_title=EBCC Data Migration

EBCCDataMigration_description=Migrates EBCC project data one project at a time.
Indicate the project source directory (an EBCC project directory should have at
minimum a child directory called application-sync)in the Source dialog box.
Indicate the output directory, where the migrated project will be written, in the
Destination dialog box.  The migration tool will write the migrated project to
this directory preserving the project name and structure.

EBCCDataMigration_classname=com.bea.commerce.migration.data.version.v4_0to7_0.E
BCCDataMigration
EBCCDataMigration_gif=null
EBCCDataMigration_skippable=true
#
# The rdbms-data-sync task
#
```

```
rdbms-data-sync_title=Data Sync RDBMS Migration
rdbms-data-sync_description=Resets Data Sync tables prior to post-migration data
sync.
rdbms-data-sync_classname=com.bea.commerce.migration.data.version.v4_0to7_0.Dat
aSyncDB
rdbms-data-sync_gif=null
rdbms-data-sync_skippable=true

#
# The rdbms-portal task
#
rdbms-portal_title=Portal RDBMS Migration
rdbms-portal_description=Migrates Portal tables.
rdbms-portal_classname=com.bea.commerce.migration.data.version.v4_0to7_0.Portal
DB
rdbms-portal_gif=null


rdbms-portal_skippable=true

#
# The rdbms-entitlement-ruleset task
#
rdbms-entitlement-ruleset_title=ENTITLEMENT_RULESET RDBMS Migration
rdbms-entitlement-ruleset_description=Creates new 7.0 ENTITLEMENT_RULESET table.
rdbms-entitlement-ruleset_classname=com.bea.commerce.migration.data.version.v4_
0to7_0.EntitlementRulesetDB
rdbms-entitlement-ruleset_gif=null
rdbms-entitlement-ruleset_skippable=true
#
# The EntitlementRulesetMigration task
#
EntitlementRulesetMigration_title=Entitlement Ruleset Data Migration
EntitlementRulesetMigration_description=Migrates Entitlement Ruleset Data.
EntitlementRulesetMigration_classname=com.bea.commerce.migration.data.version.v
4_0to7_0.EntitlementRulesetMigration
EntitlementRulesetMigration_gif=null
EntitlementRulesetMigration_skippable=true
```

移行ログ ファイル

Migration Viewer は、migration ディレクトリ内の migration.log ファイルにアクションを記録します。このファイルは移行を実行したときに作成されます。

ログ ファイルにはツールのバージョン、インストールされているサービス パック、その他の移行環境情報などのシステム プロパティ情報も記録されます。移行ツールの使用中に問題または疑問が生じた場合、このファイルの情報が解決の手掛かりとなります。ファイルには画面上のメッセージよりも詳細な情報が記録され、移行プロセスのスタック トレース全体も記録されます。

カスタマイズされたデータベースを変換する .sql ファイル

リリース 4.0 形式のデータベースを 7.0 形式のデータベースに移行するために使われるすべての SQL スクリプトは、データベースの種類およびバージョン別に `migration` ディレクトリに収められています。`oracle-817-v4_0to7_0.sql` のような名前が示すように、各種データベースのバージョンごとに 1 つずつのファイルが用意されています。データベースに手動で修正を行った場合、これらのファイルを使用して適切な修正済み SQL 文を作成し、実行する必要があります。手順については、2-15 ページの「変更が行われたデータベースの手動移行」を参照してください。

API の変更点を Migration Viewer と migrinfo.html で確認する

`migrinfo.html` ファイルには、このリリースでの API の変更点に関する情報が収められています。コードの移行作業を行うときは、Migration Viewer で情報を参照すると役に立ちます。詳細については、付録 A の「Migration Viewer ツールによるコードへの変更の参照」を参照してください。

E-Business Control Center プロジェクト移行の詳細

E-Business Control Center プロジェクトの移行に関して、通常は特に問題となる点はありません。ただし、プロジェクトの内容または最新バージョンへの準拠状況によっては、予期しない結果が生じる可能性があります。作業を続ける前に、この節の情報を再確認してください。

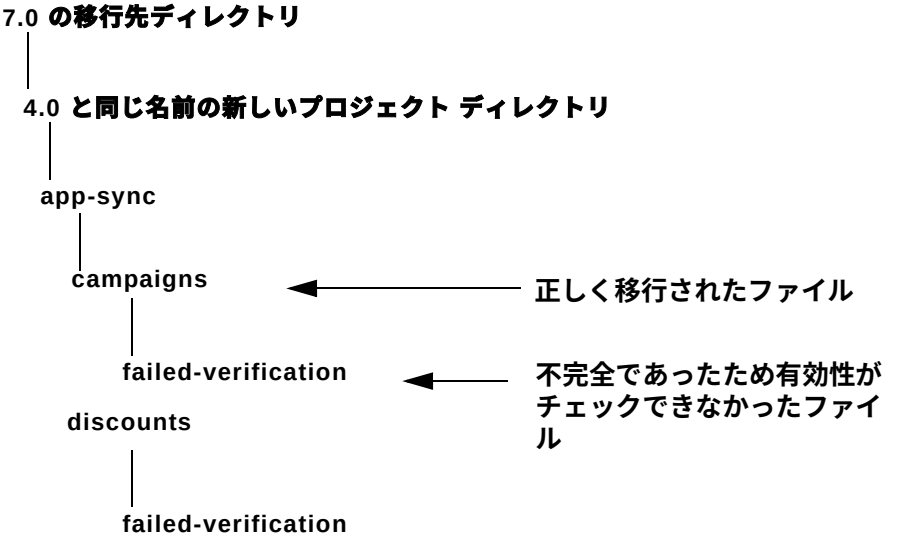
完全性と有効性が評価される 移行の間、すべてのプロジェクトは異なった形式に変換されます。移行プロセスでは、プロジェクト ファイル内のデータの有効性も検証されます。ただし、プロジェクトが不完全な場合、有効性のチェックは実行できません。

- ファイルが完全であっても、破損しているファイルや無効なファイルについては、ユーザが定義する名前の別の移行フォルダに移行されます。フォルダのデフォルト名は `failed-verification` です。このフォルダの名前は変更できます。2-8 ページの「migrator.bat および migration_install.properties ファイルの編集」を参照してください。
- ファイルが不完全な場合、有効性はチェックされません。ファイルは標準の移行先ディレクトリにコピーされます。2-8 ページの「migrator.bat および migration_install.properties ファイルの編集」の指示に従うことによって、このディレクトリも設定できます。

不完全なファイルの有効性はチェックされないため、移行を行う前にファイルが完全であることを確認するようにします。

図 B-1 にディレクトリの構造を示します。

図 B-1 移行先ディレクトリ



完全性の判定 ファイルが不完全かどうかの判定は、ファイルの `iscomplete` 属性によって設定されますが、次の例外があります。

- ポータルおよびポートレット ドキュメントでは、ファイルの先頭のルート タグ内の `iscomplete` 属性の代わりに、独立した `iscomplete` タグが存在します。
- `.wfx` ファイルはテンプレートであり、移行プロセスは常にテンプレートの検証を試みます。
- `.pln` ファイルには `incomplete` 属性があります。値が `false` の場合、ファイルは不完全です。属性が出現しない場合、ファイルは完全です。

不完全または無効なファイルが移行またはコピーされる場合 結果としてコピーまたは移行されたファイルに関しての問題を修正するには、E-Business Control Center を使用して、正しい情報または完全な情報を入力します。破損が著しい無効ファイルについては、E-Business Control Center で開くことができない場合があります。不完全なファイルは問題なく開くことができます。

ポータル Webflow の変更点

注意： この節では、2-34 ページの「Webflow と JSP をコンフィグレーションする」の補足情報を示します。

この節では、Portal Web アプリケーションに必要な 3 つのポータル Webflow への変更内容を示します。Web アプリケーションがポータルでない場合、これらの変更は必要ありません。

各 Webflow について変更または追加されたノードの一覧を示していますが、それぞれの新規ノードまたは変更されたノードの構造を詳しく確認するには、Webflow の 7.0 バージョンのコピーを実際に開いてみることをお勧めします。

Tools Webflow

プレゼンテーション ノード

- `portal-prefs.jsp` – ノード `change_name.jsp` への発信イベント `link.change_name` が追加されています。
- `change-name.jsp` – これはリリース 7.0 の新しいノードです。このノードとすべての関連イベントを追加します。

入力プロセッサ

- `displayNameSpecialCharacterProcessor` – これはリリース 7.0 の新しいノードです。このノードとすべての関連イベントを追加します。

ワイルドカード プロセッサ ノード

- `com.bea.p13n.appflow.webflow.forms.MissingFormFieldException` – これはリリース 7.0 の新しい例外です。
- `com.bea.p13n.appflow.exception.InvalidArgumentException` – これはリリース 7.0 の新しい例外です。

セキュリティ Webflow

入力プロセッサ

- loginProcessor – groupProcessor への発信イベント success と、badlogin.jsp への発信イベント failure が追加されています。
- specialCharacterProcessor – これはリリース 7.0 の新しいノードです。このノードとすべての関連イベントを追加します。

user_account Webflow

プレゼンテーション ノード

- new_user.jsp –（userProcessor への）発信イベント button.go が削除され、createAccountFormProcessor への発信イベント button.create_user_account が追加されています。
- account_info.jsp – これはリリース 7.0 の新しいノードです。このノードとすべての関連イベントを追加します。

入力プロセッサ

- postLoginProcessor – portal Webflow 内の portalRefreshProcessor への発信イベント success が追加され、発信イベント user.create の送り先が dispatchUserRegEventProcessor から createUserProfileProcessor に変更されています。
- getAccountProcessor – これはリリース 7.0 の新しいノードです。このノードとすべての関連イベントを追加します。
- createAccountFormProcessor – これはリリース 7.0 の新しいノードです。このノードとすべての関連イベントを追加します。
- updateAccountFormProcessor – これはリリース 7.0 の新しいノードです。このノードとすべての関連イベントを追加します。

- updateUserProfileProcessor – これはリリース 7.0 の新しいノードです。このノードとすべての関連イベントを追加します。
- createUserProfileProcessor – これはリリース 7.0 の新しいノードです。このノードとすべての関連イベントを追加します。

ワイルドカード プレゼンテーション ノード

- link.account_info – これはリリース 7.0 の新しいワイルドカードです。
- link.new_user – これはリリース 7.0 の新しい例外です。

索引

D

E

J

M

O

R

S

W