

Oracle ホワイト・ペーパー

2009年7月

Oracle Application Development Framework 概要

はじめに.....	1
Oracle ADF - Java EE 開発を簡単に	2
Oracle ADF のアーキテクチャリオ.....	3
ビジネス・サービス・レイヤー	4
コントローラ・レイヤー	4
ビュー・レイヤー	4
モデル・レイヤー	4
生産性と選択肢の両立.....	5
Oracle ADF の利点	5
視覚的で宣言型の Java EE 開発	5
その他のフレームワークを寄せ付けない優位性	7
結論.....	8

はじめに

Java EE は、強固かつスケーラブルでセキュアな標準プラットフォームであり、現在のエンタープライズ・アプリケーションの多くがこれに基づいて作成されています。Java EE は、Java™ 言語を使用して多層アプリケーションを構築するための仕様セットを提供します。以前は、アプリケーションの強さと、それを実現するために必要な複雑さの間に直接的な相関関係がありました。しかし、Oracle Application Development Framework (Oracle ADF) の登場により、標準的なパターンやプラクティスに従ってさえいれば、大幅に少ない労力で、きわめて高度な Java EE アプリケーションを実装できるようになりました。

また、サービス指向アーキテクチャ (SOA) 方式を利用する複合アプリケーションへの構築ニーズが高まったため、開発者は、変化に素早く対応できるアプリケーション (アジャイル・アプリケーション) を開発しなければならなくなりました。このベスト・プラクティスをアジャイル・アプリケーションに実装するには、通常、膨大な量のインフラストラクチャ・コードの記述が必要であり、初めて Java EE アプリケーションを構築する開発者にとってはこれがもう 1 つの障壁となります。

Oracle Application Development Framework を利用すると、パフォーマンスと保守性に優れた強固なアプリケーションを構築できるだけでなく、SOA ベースのアジャイル・アプリケーションを実装するために最善のインフラストラクチャ・コードも提供されます。これにより、組織の"自力開発"に伴う労力が削減できるとともに、インフラストラクチャ構築に対する付加価値の実現に向けてすぐに取り組むことが可能です。

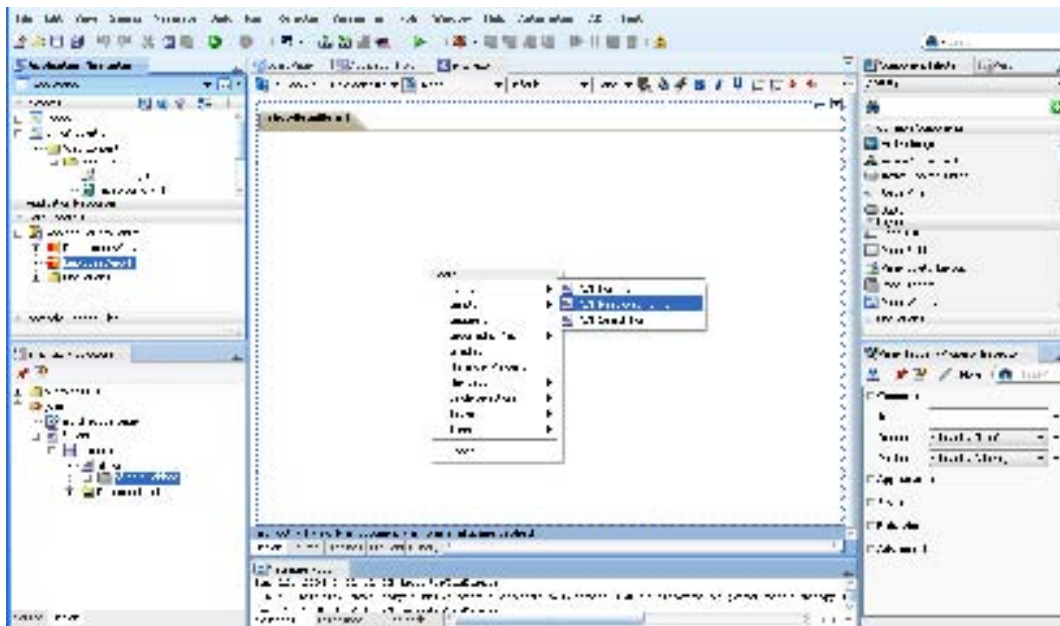
Oracle ADF - Java EE 開発を簡単に

Oracle Application Development Framework は、革新的でありながら完成度の高いオラクルの Java EE 開発フレームワークです。ほかの多くのフレームワークとは異なり、直接サポートを受けられるほか、受賞歴のある開発環境である Oracle JDeveloper 11g とともに使用できます。

Oracle ADF では、アプリケーション・インフラストラクチャの実装コードを作成する必要が最低限に抑えられるため、Java EE 開発が簡単になり、ユーザーは実際のアプリケーション機能に重点を置くことができます。Oracle ADF はフレームワークの一部としてこれらのインフラストラクチャを提供します。一連のランタイム・サービスが不十分であることを認識するために、Oracle ADF は、Oracle JDeveloper 11g 開発ツールを使用した視覚的で宣言的な Java EE 開発方法を提供する開発エクスペリエンスも目的の 1 つとしています。

たとえば、Oracle ADF では、組み込み済みのビジネス・サービスに対してサービス・インタフェースをつなげるだけで、データをサービスとして公開するアジャイル・アプリケーションを簡単に開発できます。ビジネス・サービスの実装を分離する仕組みは、Oracle ADF でメタデータを使用することによって実現しています。このメタデータ駆動型のアーキテクチャにより、アプリケーション開発者は、サービスの細かいアクセス方法に煩わされることなく、ビジネス・ロジックやユーザー・エクスペリエンスに重点を置くことができます。

ユーザー・エクスペリエンスの作成は、目的のデータ・コントロールをドラッグしてページ・デザイン上にドロップし、データを表すコンポーネント・タイプを指定するだけで実行できます。次に示す例では、あるデータベース表をビジネス・サービスとして公開し、データを表形式でレンダリングするよう JDeveloper に指定しています。実際の作業は、ページ上にコントロールをドラッグ・アンド・ドロップし、自動的に表示されるポップアップで、レンダリング・コンポーネントとして表を選択するだけで完了です。あとは、Oracle ADF が処理してくれます。



Oracle ADF では、サービスの実装情報はメタデータとして Oracle ADF モデル・レイヤーに格納されます。これにより、ユーザー・インタフェースを変更しなくても別のサービスに交換できるため、アプリケーションの機敏性がきわめて高くなります。また、ユーザー・インタフェースを作成する開発者が、ビジネス・サービスの細かいアクセスを気にする必要はありません。かわりに、アプリケーション・インタフェースとインタラクション・ロジックにのみ集中できます。

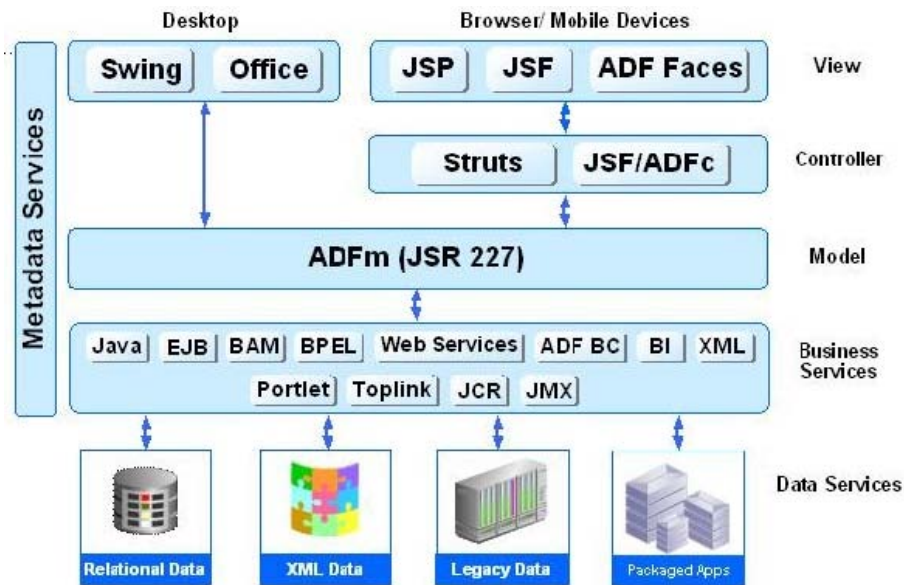
Oracle ADF のアーキテクチャリオ

Oracle ADF は、Model-View-Controller (MVC) 設計パターンをベースとしています。MVC アプリケーションは、次の 3 つのコンポーネントから構成されます。1) データソースとのやり取りを処理し、ビジネス・ロジックを実行するモデル・レイヤー、2) アプリケーションのユーザー・インタフェースを担当するビュー・レイヤー、3) アプリケーション・フローを管理し、モデル・レイヤーとビュー・レイヤー間のインタフェースとして機能するコントローラ。

アプリケーションをこれら 3 つのレイヤーに分割することにより、アプリケーション全体で、簡単にコンポーネントをメンテナンスおよび再利用できます。各レイヤーはほかのレイヤーから独立しているため、疎結合のサービス指向アーキテクチャ (SOA) が実現します。

Oracle ADF では、アプリケーションのサービス指向開発を可能にするため、MVC を実装した上に、さらにモデル・レイヤーからビジネス・サービスを分離しています。Oracle ADF アーキテクチャは、次の 4 つのレイヤーで構成されています。

- ビジネス・サービス・レイヤー - 各種ソースに格納されたデータへのアクセスを提供し、ビジネス・ロジックを処理します。
- モデル・レイヤー - ビジネス・サービス・レイヤーの上に抽象化レイヤーを提供することで、ビュー・レイヤーとコントローラ・レイヤーが一貫した方法でさまざまなビジネス・サービス実装を処理できるようにします。
- コントローラ・レイヤー - Web アプリケーション・フローを制御する仕組みを提供します。
- ビュー・レイヤー - アプリケーションのユーザー・インタフェースを提供します。



Oracle ADF のアーキテクチャ

Oracle ADF では、開発者が各レイヤーの実装時に使用するテクノロジーを選択できます。上の図では、Oracle ADF アプリケーションを構築する際に使用できるさまざまなオプションが示されています。Java EE アプリケーションの各種コンポーネントを統合し、開発を非常に柔軟にする接着剤の役割を果たしているのが、Oracle ADF モデル・レイヤーです。EJB、Web サービス、JavaBeans、JPA/EclipseLink/TopLink などのオブジェクトはすべて、Oracle ADF モデルからビジネス・サービスとして使用されます。ビュー・レイヤーには、Swing アプリケーションや MS Office 統合から、JSP、Java Server Faces (JSF)、Oracle ADF Faces を使用した HTML インタフェースまで含めることができます。

ビジネス・サービス・レイヤー

ビジネス・サービス・レイヤーは、データ永続性レイヤーとのやり取りを管理します。提供されるサービスには、データ永続性、オブジェクト・マッピング、リレーショナル・マッピング、トランザクション管理、ビジネス・ロジック実行などがあります。

Oracle ADF のビジネス・サービス・レイヤーは、単純な Java クラス、EJB 2.1 または 3.0、Web サービス、JPA オブジェクト、Oracle ADF Business Component といった実装方法で使用できます。

コントローラ・レイヤー

コントローラ・レイヤーは、アプリケーション・フローを管理するとともに、ユーザー入力を処理します。たとえば、あるページの検索ボタンをクリックすると、コントローラにより実行するアクション（検索の実行）とナビゲート先（結果ページ）が決定されます。

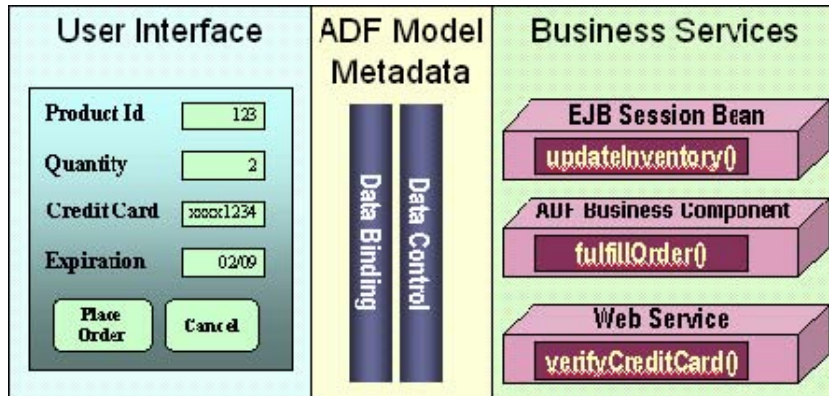
Oracle JDeveloper の Web ベース・アプリケーションでは、次の 3 つのコントローラ・オプションがあります。もっとも多くの機能を提供している Oracle ADF コントローラのほかに、標準 JSF コントローラ、Apache Struts があります。どのコントローラを使用している場合も、通常、ページやナビゲーション・ルールをダイアグラムに配置してアプリケーション・フローを設計します。Oracle ADF コントローラを使用すると、アプリケーション・フローを再利用可能な小さいタスク・フローに分割し、メソッド・コールなどの非ビジュアル・コンポーネントをフローに組み込んで、1 つのページに含まれる領域内で実行される "ページ断片" フローを作成することができます。この方法を利用すると、ユーザー・インタフェースの断片の再利用性が最大化されるとともに、ポータルやマッシュアップ・アプリケーションへの統合が簡素化されます。

ビュー・レイヤー

ビュー・レイヤーは、アプリケーションのユーザー・インタフェースの役割を果たします。ビュー・レイヤーは、HTML、JSP、JavaServer Faces (JSF)、リッチ Java コンポーネント、または XML とその変種に基づいて、ユーザー・インタフェースのレンダリングをおこないます。ビュー・レイヤーは Web クライアントである場合もあれば、クライアント/サーバー型の Swing デスクトップ・アプリケーション、Microsoft Excel スプレッドシート、または電話などのモバイル機器をワイヤレス実装したものである場合もあります。

モデル・レイヤー

モデル・レイヤーは、ビジネス・サービスと、このビジネス・サービスを使用する別のレイヤー内のオブジェクトを結びつけます。Oracle ADF では、ビジネス・サービスの上にモデル・レイヤーを実装することで、1 つのインタフェースからあらゆるビジネス・サービスへアクセスできるようにしています。モデル・レイヤーは、データ・コントロールとデータ・バインディングの 2 つのコンポーネントで構成されており、インタフェースを定義するメタデータ・ファイルを利用します。データ・コントロールは、ビジネス・サービスの詳細実装をクライアントから切り離します。データ・バインディングは、データ・コントロールのメソッドと属性を UI コンポーネントに公開することで、ビュー・レイヤーとモデル・レイヤーの明確な区別を実現しています。モデル・レイヤーのメタデータ・アーキテクチャにより、ビュー・レイヤーやコントローラ・レイヤーにどのビジネス・サービス・レイヤー実装をバインドしても、同じ方法で開発できます。また、Oracle ADF のモデル・レイヤーは、JSR-227 "A Standard Data Binding & Data Access Facility for J2EE" の基盤の役割を果たします。この JSR の目的は、Java EE アプリケーションのデータ・バインディングを標準化することにあります。



生産性と選択肢の両立

Oracle ADF では、各レイヤーを実装する際に、それぞれ異なるテクノロジーを選ぶことができますが、何を選んでも一貫して生産性の高い開発エクスペリエンスを得られます。たとえば、Oracle ADF Business Component (Oracle ADF BC) を使用した Oracle ADF Swing アプリケーションを作成する場合と Enterprise JavaBeans を使用した Oracle ADF Faces アプリケーションを作成する場合は、使用する動作とメソッドは同じものになります。さらに、実装テクノロジーを選ぶことができるだけでなく、開発スタイル (宣言型、ビジュアル、またはコーディング) や開発環境 (JDeveloper や、Eclipse といったそのほかの IDE)、そして配置環境 (すべての Java EE 準拠サーバー) をも選ぶことができます。

Oracle ADF の利点

視覚的で宣言型の Java EE 開発

便利な開発フレームワークに欠かせない特徴は、このフレームワークを使用することでアプリケーション開発が簡単になるような開発ツールを備えていることです。

Oracle ADF では、それぞれのレイヤーに対して、ビジュアル・ツールと宣言型ツールの両方が提供されています。JDeveloper IDE に組み込まれているこれらのツールにより、Oracle ADF のランタイム機能を使用しない Java 開発者であっても、メリットが得られます。

ビジネス・サービスの開発

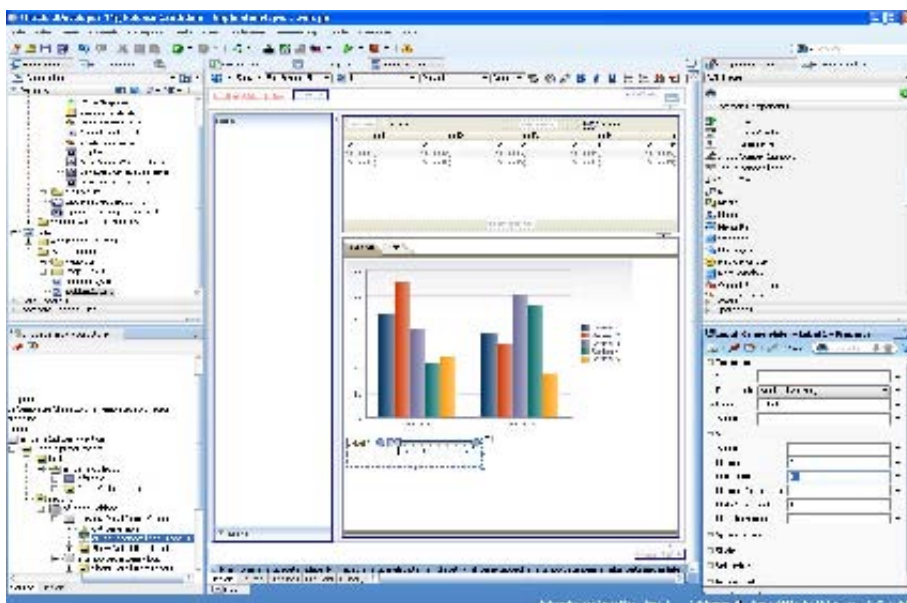
Oracle JDeveloper では、EJB/JPA、Web サービス、シンプル Java オブジェクト、Oracle ADF BC など、各種の方法を使用してビジネス・サービスを構築できます。"選択肢と生産性の両立"がこのアプローチの基本です。ビジネス・サービスを生成する際、ウィザード主導のアプローチを使用すると、表に対する Java インタフェースを提供するビジネス・サービスを生成できます。これらのインタフェースを右クリックするだけで、SDO ベース Web サービスなどの Web サービスとして公開することができます。また、視覚的で宣言型の開発というテーマに従うことから、ビジュアル・モデリングを使用してこれらのインタフェースを生成することもできます。

Oracle ADF Business Component は、オブジェクトの作成に焦点を当てたフレームワークです。このオブジェクトは、より宣言的な方法で、データソースの上にビジネス・サービス・レイヤーを実装します。トランザクション管理、リソース・プーリング、ロック、宣言型検証規則、変換、オブジェクト・リレーショナル・マッピングなど、面倒な設定なくすぐに使用できるサービスが標準で提供されます。Oracle ADF BC は、SQL に基づく Java オブジェクト定義、検証規則の宣言型定義、ビジネス・サービスのライフ・サイクルにコードを注入できる事前定義済みイベントなどの機能を提供しているため、4GL を使用した宣言型データベース主導開発の経験がある開発者にとって、なじみやすいものです。Oracle ADF BC 開発には、宣言型ダイアログとプロパティ・インスペクタを使用します。一般的な Java EE 設計パターンの実装がフレームワーク内に組み込まれているため、アプリケーションのパフォーマンスとスケーラビリティが保証されます。

ユーザー・インタフェースの開発

アプリケーションのビュー・レイヤーおよびコントローラ・レイヤーにおける視覚的で宣言型の開発機能は、Oracle JDeveloper でも数多く提供されています。

- Oracle ADF コントローラ用のページ・フロー・モデラー、基本的な JSF フレームワークによるページ・フロー・コントローラ、およびオープンソースの Apache Struts フレームワークは、ダイアグラム上へコンポーネントをドラッグ・アンド・ドロップするだけで実行できる、視覚的なページ・フロー・モデリングを提供します。
- JSP、JSF、HTML、Swing、およびワイヤレス・ベースのユーザー・インタフェース向けのビジュアル・エディタでは、あらゆるコンポーネントに対して WYSIWYG 開発を実行できます。
- ユーザー・インタフェースにコンポーネントを追加する宣言型開発ツールは、宣言型コンポーネント、プロパティ・インスペクタ、拡張可能なコンポーネント・パレット、およびデータ・コントロール・パレットを作成できます。
- 再利用性機能には、タスク・フロー、Oracle ADF ライブラリ、宣言型コンポーネントの作成などの再利用性を最大化する機能が含まれます。
- Oracle ADF Faces は、部分ページ・レンダリングや Ajax などの最新テクノロジーを利用する標準 JSF API の上に構築された多数の UI コンポーネントのセットであり、高機能でインタラクティブなユーザー・インタフェースを提供します。



JSP と JSF の視覚的な編集

視覚的开发ツールと宣言型開発ツールは Oracle JDeveloper IDE 内で同期化されているため、ビジュアル・エディタ、プロパティ・インスペクタ、モデラーは、常にソース・コードとの同期が取られます。このため、開発者は、ドラッグ・アンド・ドロップを使用するのか、宣言的にプロパティを定義するのか、またはソース・コードを直接編集するのか、好みの開発スタイルを選ぶことができます。

ビジネス・サービス・コンポーネントからユーザー・インタフェースへのバインド

Oracle JDeveloper では、JSR-227 データ・コントロールの革新的な実装により、ビジネス・サービス・レイヤーからコントローラ・レイヤーとビュー・レイヤーに対して、きわめて簡単にコンポーネントをバインドできます。データ・コントロール・パレットを使ってビジネス・サービス・レイヤーを確認したら、データ・オブジェクトをドラッグ・アンド・ドロップして、各自のユーザー・インタフェース実装にバインドするだけで完了です。同じ仕組みにより、コントローラ・アクションからビジネス・サービス・レイヤーに定義されたメソッドへのバインドが簡単に実行できます。すべては、視覚的で宣言型の操作をおこなうだけでよいのです。

その他のフレームワークを寄せ付けない優位性

ここでは、数ある Java EE フレームワークの中で Oracle ADF を際立たせている特徴について説明します。

エンド・ツー・エンドのソリューション - Oracle ADF は、Java EE アーキテクチャの 1 つのレイヤーだけに焦点を合わせているわけではありません。Oracle ADF は、ビュー・レイヤーやデータ・バインディング (JSR-227) からビジネス・サービスおよびデータ・アクセスにいたるまで、すべての Java EE レイヤーに対して統一された完全なソリューションを提供します。また、開始からサポートまでのすべての開発ライフ・サイクル・フェーズに対応しています。

開発環境 - 多くの Java EE フレームワークには、十分な開発ツールが備わっていません。視覚的ツールと宣言型アプローチを提供することでコーディングの必要性を最小化した Oracle JDeveloper は、Oracle ADF ベースのアプリケーション構築における最高のツールであるといえます。また、この宣言型開発アプローチのおかげで、4GL 方式のツールに慣れた開発者でも簡単に習得することができます。Oracle ADF は標準化されているため、Eclipse などの別の IDE を使用することもできます。

プラットフォーム独立性 - 多くのフレームワークでは、特定のソフトウェア・ベンダーへの束縛を余儀なくされます。しかし、Oracle ADF を使用すれば、任意の Java EE 準拠アプリケーション・サーバーにランタイムをインストールでき、任意の SQL-92 準拠データベースにビジネス・サービスを接続できます。

テクノロジーの選択 - 開発者には、アプリケーションのレイヤーごとに好みの実装方法があります。Oracle ADF は、アプリケーションの各レイヤーに対して複数のテクノロジーをサポートしているため、特定のテクノロジーや開発スタイルが開発者に押しつけられることはありません。

テクノロジーへのコミットメント - Oracle ADF は、オラクルの次世代エンタープライズ・アプリケーション・セットにおいて選ばれたテクノロジーであり、社内の開発で継続的に使用されているという事実に注目してください。この製品はポータル・アプリケーション、ワイヤレス・アプリケーション、そして Web アプリケーションの開発に使用されているため、一貫性のあるテクノロジー・スタックが提供され、サポートされることが約束されています。

メタデータ駆動型 - Oracle ADF フレームワークのすべてのレイヤーで、XML メタデータを使用した宣言型の開発オプションが提供されるとともに、必要に応じてカスタム・コーディングを使用することもできます。構築アプリケーションにおいて、フレームワークのすべてを使用するか、一部を使用するかを選択することで、アプリケーション・コンポーネントの再利用性と柔軟性を大幅に向上できます。また、メタデータを使用することで、データがバインドされたフィールドに対するルールをモデル・レイヤーで指定できます。メタデータでは、Oracle ADF データ・バインディングのラベル・プロパティ、検証プロパティ、ツールチップ・プロパティを指定できます。これらのプロパティは、ユーザー・インタフェース実装から独立して利用されます。さらに、MDS レイヤーは 2 つの異なる実装レイヤーを介してユーザー・エクスペリエンスのカスタマイゼーションに対応しています。1 つ目は"シード済カスタマイゼーション"で、特定のグループのアプリケーションにアクセスするユーザーを対象とするアプリケーション全体のカスタマイゼーションです。2 つ目は、通常"パーソナライゼーション"とも呼ばれる"ユーザー・カスタマイゼーション"です。この場合は、エンドユーザーがパーソナル・エクスペリエンスへのカスタマイゼーションを指定後、その内容は MDS リポジトリで保持されます。

再利用性の強化 - JDeveloper と Oracle ADF の組合せにより、優れた再利用性機能がサポートされます。これらの機能には、JSF テンプレート、タスク・フローとタスク・フロー・テンプレート、Oracle ADF ライブラリ、JSF 断片ベースのリージョンとタスク・フローなどがあります。

ソースの提供 - オラクルでは、サポート・ライセンスをもつ顧客に Oracle ADF フレームワークのソース・コードを提供しています。ソースを使用できることで、開発者はフレームワークの根底にあるメカニズムを理解できるため、アプリケーションの問題をデバッグすることが容易になります。

サポート - Oracle ADF はオラクルの正式な製品であり、Oracle Support によるサポート・サービスが提供されます。このサービスにより、定評のあるサポートが 24 時間体制で受けられます。

トレーニング - Oracle University では、Oracle ADF と Oracle JDeveloper に関する Instructor Lead Training (ILT) コースが提供されています。

結論

Oracle ADF では、すぐに実装できる設計パターンとインフラストラクチャ・コードを標準で提供することで、容易な Java EE 開発を実現します。Oracle ADF では、開発アプローチ、使用するテクノロジー、そして配置プラットフォームを選択できます。Oracle ADF の先進アーキテクチャと Oracle JDeveloper 11g の視覚的開発環境の組合せは、初心者であっても上級者であっても、Java アプリケーション開発の生産性向上を目指す開発者にとって最高のソリューションです。

Oracle ADF および Oracle JDeveloper に関して、詳しくは Oracle Technology Network (OTN : <http://www.oracle.com/technology/global/jp/products/jdev/index.html>) のWebサイトを参照してください。



Oracle Application Development
Framework の概要

2009 年 7 月

著者 : Shaun O'Brien

共著者 : Shay Shmeltzer

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

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
www.oracle.com



Oracle is committed to developing practices and products that help protect the environment

Copyright © 2009, Oracle and/or its affiliates. All rights reserved.

本文書は情報提供のみを目的として提供されており、ここに記載される内容は予告なく変更されることがあります。本文書は一切間違いがないことを保証するものではなく、さらに、口述による明示または法律による黙示を問わず、特定の目的に対する商品性もしくは適合性についての黙示的な保証を含み、いかなる他の保証や条件も提供するものではありません。オラクルは本文書に関するいかなる法的責任も明確に否認し、本文書によって直接的または間接的に確立される契約義務はないものとします。本文書はオラクル社の書面による許可を前もって得ることなく、いかなる目的のためにも、電子または印刷を含むいかなる形式や手段によっても再作成または送信することはできません。

Oracle は米国 Oracle Corporation およびその子会社、関連会社の登録商標です。そのほかの名称はそれぞれの会社の商標です。