Oracleホワイト・ペーパー
2011年6月

# Oracle Application Development Framework概要

ORACLE

## 概要

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）方式を利用した複合アプリケーションへの構築ニーズが高まったことで、開発者はきわめて俊敏性に優れたアプリケーション（アジャイル・アプリケーション）を作成する必要に迫られています。これらのベスト・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、大量のインフラストラクチャ・コードを記述する必要があるため、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もう1つの障害となっています。

Oracle Application Development Frameworkはパフォーマンスと保守性に優れた堅牢なアプリケーションを提供するだけでなく、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のベスト・オブ・ブリードのインフラストラクチャ・コードを提供します。これによって、組織の"独自開発"に必要な労力が解消されるとともに、開発チームはインフラストラクチャ構築ではなく付加価値の提供にすぐに取り組むことができます。

## Oracle ADF - Java EE開発を簡単に

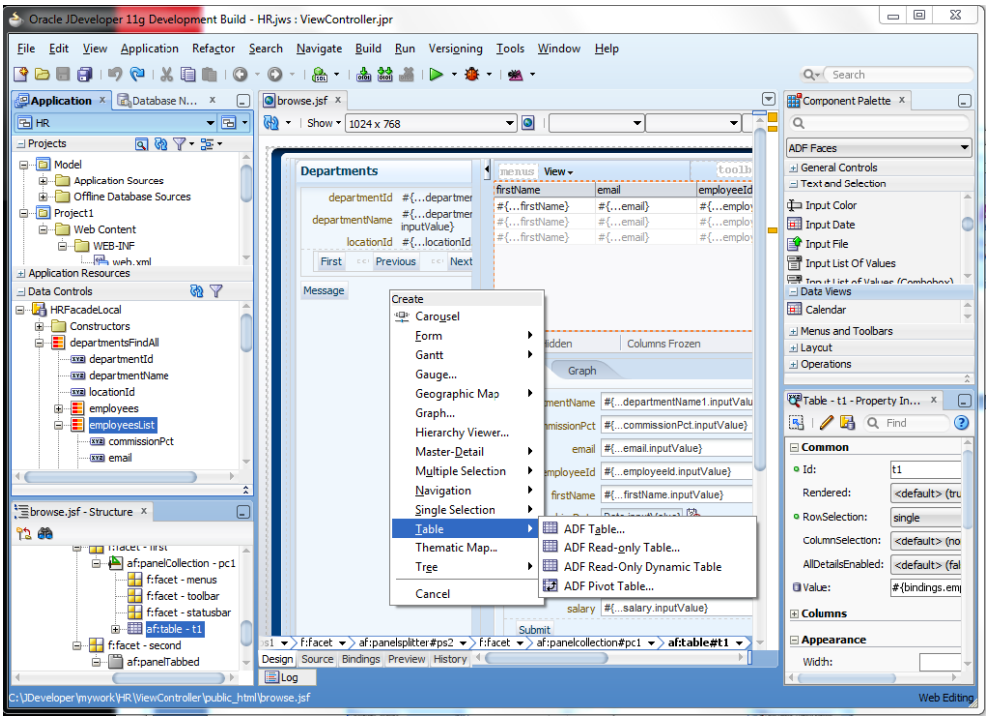
Oracle Application Development Frameworkは、オラクルが提供する、革新的でありながら成熟したJava EE開発フレームワークです。このフレームワークは、受賞暦のあるOracle JDeveloper 11g開発環境でサポートされており、この開発環境から直接使用できます。

Oracle ADFを使用すると、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記述する必要が最小限に抑えられるため、Java EE開発が簡単になり、開発者は実際のアプリケーション機能に焦点を合わせることができます。Oracle ADFはこれらのインフラストラクチャ実装をフレームワークの一部として提供します。Oracle ADFは一連のランタイム・サービスを認識するだけでなく、開発エクスペリエンスも重視しており、Oracle JDeveloper 11g開発ツールを介して、ビジュアル型と宣言型の両方のJava EE開発アプローチを提供しています。

Oracle ADFはModel-View-Controller（MVC）設計パターンを実装しており、このアーキテクチャのすべてのレイヤーを網羅した統合ソリューションを提供します。提供するソリューションは、オブジェクト/リレーショナル・マッピング、データ永続性、再利用可能なコントローラ・レイヤー、リッチWebのユーザー・インタフェース・フレームワーク、UIへのデータ・バインディング、セキュリティ、カスタマイズなどの領域を対象としています。Oracle ADFは中心となるWebベースのMVCアプローチの枠を越えてOracle SOAおよびOracle WebCenter Portalフレームワークを統合することで、複合アプリケーション全体の構築を簡素化します。

たとえば、Oracle ADFでは、組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でメタデータを介して実行されます。このメタデータ主導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、目的のビジネス・サービスをビジュアル・ページ・デザイナにドラッグ・アンド・ドロップし、データを表現するコンポーネント・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。残りは、Oracle ADFが処理してくれます。




Oracle ADFでは、これらのサービス実装の詳細情報をOracle ADFモデル・レイヤーのメタデータ内に格納します。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。

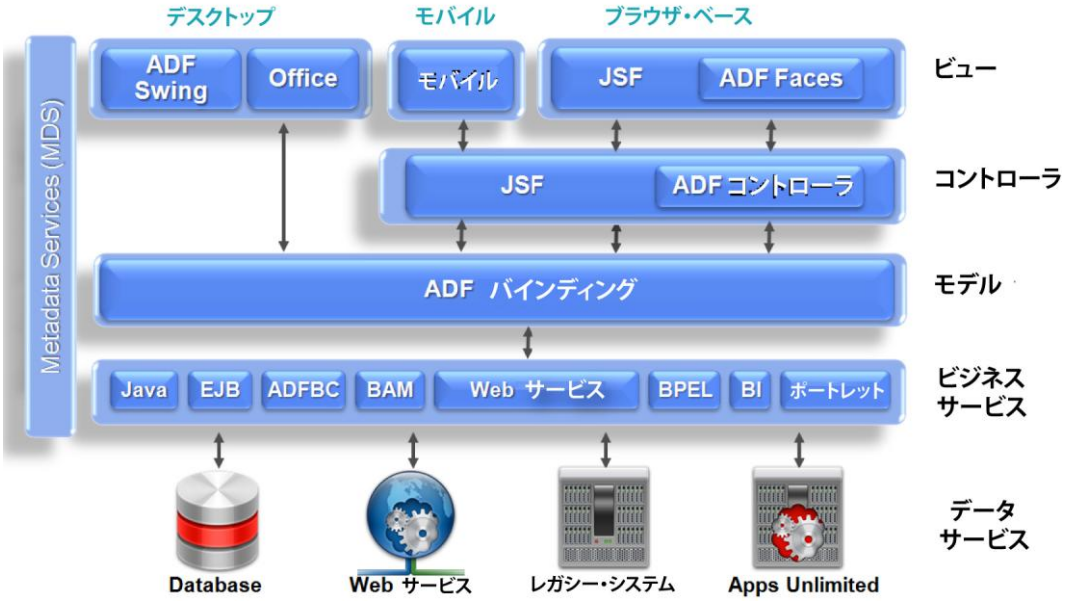
## Oracle ADFのアーキテクチャ

Oracle ADFは、Model-View-Controller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。各レイヤーは他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、JSF、デスクトップのSwingアプリケーション、およびMS Officeのフロントエンドを使用して実装されたWebベースのインタフェースに加えて、モバイル・デバイス向けのインタフェースが含まれます。

## 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、Webサービス、JPAオブジェクト、Oracle ADF Business Componentのいずれかのオプションを使用して実装できます。また、データはファイル（XMLまたはCSV）やRESTから直接消費できます。

## 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。

JDeveloperのWebベース・アプリケーションには、標準のJSFコントローラとOracle ADFコントローラという2つのコントローラ・オプションがあります。Oracle ADFコントローラは、JSFコントローラの機能を拡張したものです。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・コールや決定ポイント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。

Oracle ADFはビジネス・サービスへのマルチチャネル・アクセスをサポートしているため、ビジネス・サービスを再利用することで、Webクライアント、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、またはスマートフォンなどのモバイル・デバイスからサービスにアクセスできます。

Webベースのインタフェースに対して、Oracle ADFは150を上回る豊富なAjax対応のJSFコンポーネントを提供しているため、魅力的な動的ユーザー・インタフェースを簡単に作成できます。

## 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。

## 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では、使用する動作とメソッドは同じものになります。実装テクノロジーを選択できるだけではなく、開発スタイル（宣言型、ビジュアル、またはコーディング）を選択できるため、スキルや作業方法の異なる開発者が同じプロジェクトで協業できます。

## 宣言型のカスタマイズとパーソナライズ

Oracle ADF独自の特徴として、構築されたアプリケーションを特定のユーザー要件に合わせてカスタマイズできる機能があります。MDS（MetaData Services）と呼ばれるレイヤーを使用すると、開発者はOracle ADFが利用するXML定義をカスタマイズして、ユーザーごとに異なるバージョンを使用できます。

最初のカスタマイズ・レベルは、実行時にエンドユーザーに対して有効化され、ユーザーはアプリケーションのインタフェースをパーソナライズできます。たとえば、表に含まれるフィールドの順序を変更したり、特定のアコーディオンを展開したりすると、MDSによってこれらの変更が維持されます

2番目のカスタマイズ・レベルはシード済みカスタマイゼーションと呼ばれており、開発者はこれを利用することで、Oracle ADFによってXML形式で維持される、ビジネス・ルール、ページ・フロー、ページ・レイアウトなどの項目に変更を加えることができます。これらの変更は、アプリケーションにログインしたユーザーに応じて、実行時に適用されます。このように階層化されたカスタマイズ・アプローチを利用すると、1つのベース・アプリケーションを使用してさまざまな顧客要件を満たすことができます。

## 統合セキュリティ

Oracle ADFには、レイヤー間にまたがるセキュリティ実装が組み込まれています。開発者はユーザーとロールを定義して、各種の認可を割り当てることができます。また、権限はビジネス・サービス・レイヤー、コントローラ・レイヤー、またはUIレイヤーで割り当てることができます。レイヤー間にわたるこの統合セキュリティ・フレームワークを使用することで、レイヤーごとにセキュリティ・メカニズムを複製する必要がなくなります。

たとえば、ビジネス・サービス内のフィールドが一部のユーザーに対して更新不可能と定義されている場合、開発者がこのチェックをUIレイヤーにコーディングしなくても、UIレイヤーではすべての画面で自動的にこのフィールドの表示が読取り専用モードに切り替えられます。

# Oracle ADFの利点

## ビジュアル型と宣言型の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して、これらの表に対して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。また、右クリックするだけで、これらのインタフェースをWebサービス（SDOベースのWebサービスを含む）として公開できます。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。

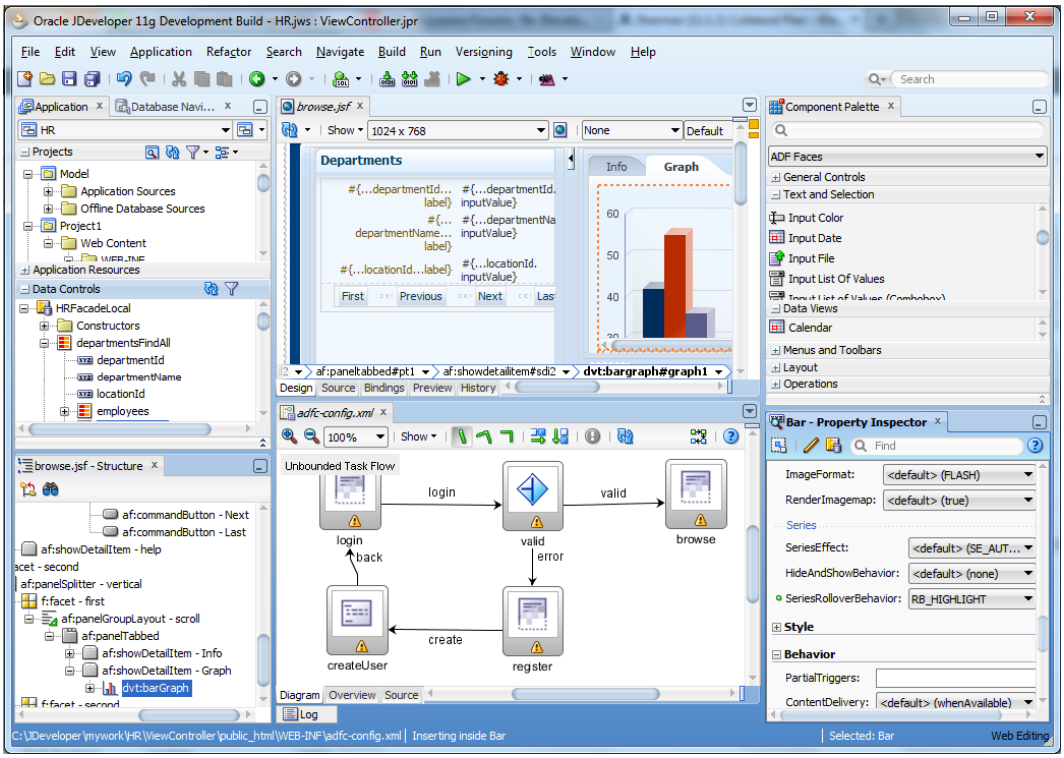
Oracle ADF Business Componentは、オブジェクトの作成に焦点を当てたフレームワークです。このオブジェクトは、より宣言的な方法で、データソースの上にビジネス・サービス・レイヤーを実装します。トランザクション管理、リソース・プーリング、ロック、宣言型検証規則、変換、オブジェクト・リレーショナル・マッピングなどのサービスが標準で提供されます。Oracle ADF BCは、SQLに基づくJavaオブジェクト定義、検証規則の宣言型定義、ビジネス・サービスのライフ・サイクルにコードを注入できる事前定義済みイベントなどの機能を提供しているため、4GLを使用した宣言型データベース主導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 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150以上の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。
- Oracle ADFコントローラ用のページ・フロー・モデラーと標準JSFフレームワークのページ・フローは、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を使用すると、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。




**視覚的なJSF開発**

視覚的開発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、革新的なバインディング・レイヤー・アプローチを使用することで、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する方法を提供します。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行うだけで良いのです。

## その他のフレームワークに対するOracle ADFの優位性

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、開発ツールによる強力な統合サポート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、フレームワーク・コードを記述する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言えます。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できます。Eclipseをはじめとするその他のIDEを使用する必要がある場合、Oracle Enterprise Pack for Eclipseパッケージで提供される組込み機能を利用し、Java EE標準に対するOracle ADFサポートを利用することで、これを実現できます。

**プラットフォームの独立性** - 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、各種の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・セットであるOracle Fusion Applicationsに対して選ばれたテクノロジーであり、社内開発のために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。

**メタデータ主導型** - 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。

**宣言型のカスタマイズ** - Oracle ADF独自のソリューションを使用することで、組織は1つのベース・アプリケーションを使用し、さまざまなユーザーの要件に合わせてこのアプリケーションをカスタマイズできます。Oracle ADFは、2種類の実装レイヤーを介してアプリケーションのカスタマイズを実現するMetaData Servicesレイヤーと連携して動作します。最初のレイヤーはアプリケーション全体のカスタマイゼーションを指す"シード済みカスタマイゼーション"であり、特定のグループに属するユーザーがアプリケーションにアクセスする際に有効になります。2番目のレイヤーは"パーソナライゼーション"と呼ばれることの多い"ユーザー・カスタマイゼーション"であり、エンドユーザーはこれを使用して、個人的なエクスペリエンスに対するカスタマイズを指定できます。指定した内容はMDSリポジトリに維持されます。

**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、再利用可能なタスク・フロー、タスク・フロー・テンプレート、再利用可能なビジネス・サービス、Oracle ADFライブラリ、JSF断片ベースのリージョンなどが含まれます。

**ソースの提供** - オラクルでは、サポート・ライセンスを持つ顧客に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Instrucor 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み合わせは、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について、詳しくはOracle Technology Network（OTN）のWebサイト（http://www.oracle.com/technetwork/jp/developer-tools/jdev/overview/index.html）を参照してください。

Oracle Application Development
Framework概要
2011年6月
著者：Shaun O’Brien、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



Hardware and Software, Engineered to Work Together