

Oracle Forms Services と Oracle Forms Developer 12c の技術概要

Oracle ホワイト・ペーパー | 2016 年 1 月





免責事項


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



ORACLE®

目次

概要	1
アプリケーションの迅速な開発とデプロイメント	2
Oracle Forms Developer	2
Oracle Fusion Middleware Forms Services	2
完全なアプリケーション・フレームワーク	3
Oracle Forms Developer を使用した機能豊富なアプリケーションの構築	3
Oracle Forms Developer による RAD 開発	3
データ・ブロック・ウィザード	3
レイアウト・ウィザード	3
プロパティ・パレット	4
統合 PL/SQL エディタ	4
オブジェクト・ライブラリ	4
組込みパッケージ	4
Unicode のサポート	4
機能豊富で拡張可能なユーザー・インタフェースの構築	5
Builder からブラウザへの直接実行	6
N-TN 層の“リモート”デバッグ	6
Forms API および Java	6
XML 表現	7
アクセシビリティ	7
Oracle Fusion Middleware Forms Services – アーキテクチャ	7
Forms クライアント	7
Forms ランタイム・プロセス	8
Forms リスナー・サーブレット	8
接続プロセスの概要	8
接続プロセスの詳細	9
Oracle Forms Services の利点	11
幅広いファイアウォールとプロキシのサポート	11
プロトコルの制限がない (HTTP/1.1 または HTTP/1.0)	11
SSL デプロイメントのための証明書の購入は不要	11
標準ロードバランシングのサポート	11
Forms Java クライアント用の Java 7 や Java 8 をサポート	11



中間層での Java 8 のサポート	11
グローバルなデプロイメント	12
あらゆるネットワークで実行可能	12
1 度だけのログイン	12
あらゆるタイムゾーンでアプリケーションの実行が可能	12
ブラウザ言語の検出	13
Oracle データベースとの緊密な統合	13
Oracle データベースの国際化のサポート	14
パフォーマンスとスケーラビリティ	14
ロードバランシング	14
結論	15



概要

このホワイト・ペーパーでは、Oracle Forms Developer の機能と利点の概要、Oracle Fusion Middleware Application Server のコンポーネントである Oracle Fusion Middleware Forms Services アーキテクチャの要点、および Oracle Forms アプリケーションのデプロイ・プロセスについて説明します。

アプリケーションの迅速な開発とデプロイメント

E-Commerce とインターネットの急速な発展は、ユーザー・コミュニティの拡大に拍車をかけています。このため、アプリケーションのパフォーマンス、信頼性、可用性に対する要求がさらに高まっています。ビジネス環境はかつてないほど流動的になり、ビジネスでは変化する要件に迅速に対応し、アプリケーションを適合させることが必須になっています。変化するビジネス要件に迅速に対応し、アプリケーションを短期間でシームレスにデプロイするツールが、これまで以上に求められています。

Oracle Forms Developer

Oracle Forms Developer はこれらのニーズに応えるツールです。Oracle Forms Developer は、銀行および金融、株式および債券、航空宇宙、通信、製造、小売、健康、法律、政府、教育などの業界で利用されるアプリケーションの開発レベルを引き上げる、非常に優れた製品です。ビジネス・アプリケーションの開発者は宣言型の RAD 環境で Oracle Forms Developer を使用し、インターネット向けに最適化された包括的な Java クライアント・アプリケーションを構築できます。Java コードの記述が不要なため構築期間が短縮され、専門的なユーザー・コミュニティの要件に対応します。このような Java クライアント・アプリケーションは機能が豊富で、Web で展開されます。このため、大量のデータや複雑な計算、分析、およびトランザクションを短時間で処理するといった、ニーズに応じた使用ができます。

Oracle Forms Developer の統合されたツール・セットを使用すれば、高度なデータベース・フォームとビジネス・ロジックを最小限の作業で簡単かつ迅速に構築できます。この開発環境には、ウィザード、組込み、ドラッグ・アンド・ドロップなど、わずかな時間と最小限のコーディングでデータベースの定義から完全に機能するアプリケーションを作成する宣言型の強力な機能が用意されています。また、Oracle Forms Developer は、オープンで拡張可能なユーザー・インタフェース・モデルも備えているため、Java を使ったアプリケーションの高度なカスタマイズと拡張が可能です。

Oracle Fusion Middleware Forms Services

Oracle Fusion Middleware Forms Services は、Web 上での Oracle Forms アプリケーションのデプロイ用に最適化されたアプリケーション・サーバーであり、そのサービスの結合でもあります。Oracle Fusion Middleware Forms Services には、すぐに使用できる機能とシステム固有のサービスが含まれています。これらを使用すると、任意のネットワークを介して、Oracle Forms アプリケーションを自動的に拡張および実行できます。また、Web 用に最適化された、機能豊富で拡張可能な Java クライアントを実現できます。

Oracle Forms の組込みサービスには、トランザクション管理、レコード・キャッシング、レコード・ロッキング、および例外処理が含まれます。Oracle Fusion Middleware Forms Services の組込みサービスには重要なインフラストラクチャが含まれており、これにより、アプリケーション全体で、開発者が何度も手作業でコーディングしたり再コーディングしたりする必要がなくなります。

完全なアプリケーション・フレームワーク

Oracle Forms Developer と Oracle Forms Services は、インターネットと企業ネットワークに Oracle Forms アプリケーションを最適にデプロイメントするための、完全なアプリケーション・フレームワークを備えています。また、高速アプリケーション開発（RAD）環境とアプリケーション・インフラストラクチャも提供しているため、ネットワーク上でのインターネット・アプリケーションの拡張と実行が可能となります。このフレームワークはオープンで拡張可能であり、さらに改善を続けています。これには、次のようなメリットがあります。

- » 新しいテクノロジーに対するアプリケーションのシームレスな適用
- » プラガブル Java コンポーネントを使用しているため、ネイティブ Java を介したユーザー・インタフェースの簡単な拡張
- » コードベースの統合による、Java や XML などのテクノロジーの利用

Oracle Forms Developer と Oracle Fusion Middleware Forms Services によってアプリケーション・フレームワーク・インフラストラクチャが提供されますが、最新テクノロジーをご使用のアプリケーションに活用できる柔軟性も持ちます。このため、ユーザーはアプリケーションのインフラストラクチャに煩わされることなく、より重要な要件であるアプリケーションのビジネス・ロジックや機能に集中できます。

Oracle Forms Developerを使用した機能豊富なアプリケーションの構築

Oracle Forms クライアントは、Java を使って記述およびデプロイされています。Oracle Forms Developer を使用すれば、ビジネス開発者は Java コードを記述しなくてもインターネット向けに最適化された Java アプリケーションを構築できます。

Oracle Forms DeveloperによるRAD開発

Oracle Forms Developer の宣言型の RAD 環境は、開発初心者でも熟練開発者でも使用できます。このツール・セットには、アプリケーション開発をスピーディに行うための次のような多くのウィザードやユーティリティが含まれています。

データ・ブロック・ウィザード

データ・ブロック・ウィザードを使用すると、アプリケーションで使用するデータ・ブロックを簡単に作成または変更できます。データ・ブロック・ウィザードでデータ・ブロックが作成してある場合は、ウィザードを使用してそのデータ・ブロックを変更できます。また、このウィザード以外で作成した既存のデータ・ブロックもウィザードで変更できます。

レイアウト・ウィザード

レイアウト・ウィザードを使用すると、データ・ブロックの項目を迅速にレイアウトできます。このウィザードでは、キャンバス上のフレーム内に項目が表示され、特定のレイアウト・スタイルに従ってレイアウトされます。このレイアウト・スタイルは、独自の仕様に合わせて手動で変更できます。レイアウト・ウィザードでフレームが作成してある場合は、ウィザードを使用して変更できます。また、このレイアウト・ウィザード以外で作成した既存のフレームもウィザードで変更できます。

プロパティ・パレット

プロパティ・パレットを使用すると、フォームおよびメニュー・モジュールで作成するオブジェクトのプロパティを設定できます。エディタまたはオブジェクト・ナビゲータでオブジェクトを選択すると、プロパティ・パレットが更新されてそのオブジェクトのプロパティが表示されます。必要に応じて別のプロパティ・パレットを起動して、別なオブジェクトのプロパティを比較することもできます。

統合PL/SQLエディタ

統合 PL/SQL エディタを使用すると、Forms Builder から PL/SQL コードを記述できます。このエディタでは、アプリケーションの編集およびデバッグ用のインターフェースが用意されています。

オブジェクト・ライブラリ

オブジェクト・ライブラリを使用すると、オブジェクトを簡単に再利用できます。これにより、開発組織全体にわたって、プログラミングとルック・アンド・フィールが標準化されます。

組み込みパッケージ

Oracle Forms Developer にはいくつかの組み込みパッケージがあります。これらのパッケージには、アプリケーションの構築やアプリケーション・コードのデバッグの際に参照できる PL/SQL 構文が多数含まれています。

Unicodeのサポート

Oracle Forms Developer と Oracle Fusion Middleware Forms Services では、ユーザーのネイティブ言語でアプリケーションを使用できる機能を提供します。Unicode グローバル・キャラクタ・セットのサポートにより、多国籍の企業が多言語に対応した 1 つのアプリケーションを開発して、世界中のユーザーにデプロイできます。

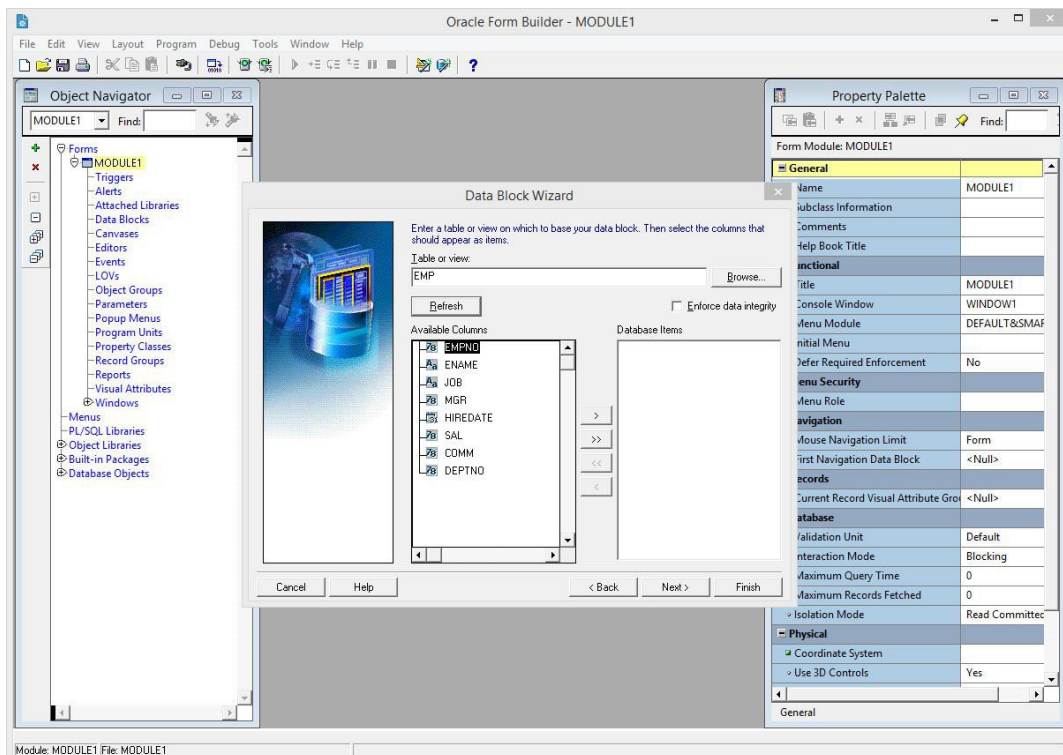


図1 : Forms Builderのデータ・ブロック・ウィザード

機能豊富で拡張可能なユーザー・インタフェースの構築

Oracle Forms Developer には、最小限のコーディングで優れた Java UI を短時間で開発するための固有のウィジェットとデフォルト・コントロールが多数用意されています。固有のウィジェットとデフォルト・コントロールには、階層ツリー・コントロール、タブページ、チェック・ボックス、ポップアップ・リスト、ツールチップ、サマリー合計、および計算結果のフィールドが含まれます。

Oracle Forms Developer には、プラグブル Java コンポーネント (PJC) を使った拡張可能な UI があります。独自のカスタム構築した Java コンポーネントを Forms アプリケーションに統合することで、Java ユーザー・インタフェースをデフォルトの機能以上に簡単に拡張およびカスタマイズできます。Oracle Technology Network (<http://www.oracle.com/technetwork/developer-tools/forms>) の Oracle Forms Download 領域から、JavaBeans と PJC の多数のサンプルをダウンロードできます。これらのサンプルには、Web ロールオーバー・ボタン、ハイパーリンク、およびクライアントからのアップロード機能が含まれます。Oracle JDeveloper や好みの 3GL Java 開発環境で作成した Java の拡張機能を、Oracle Forms アプリケーションにシームレスに統合できます。

これは Oracle JDeveloper と Oracle Forms Developer 間における統合の好例で、オラクルの統合ツール製品の利点をよく表しています。優れたパフォーマンスを実現する Oracle Forms Developer の RAD 環境と Oracle Fusion Middleware Forms Services のスケーラビリティを使用できるうえ、Java とのオープンな統合により、アプリケーションのカスタマイズと拡張が簡単であるという利点もあります。

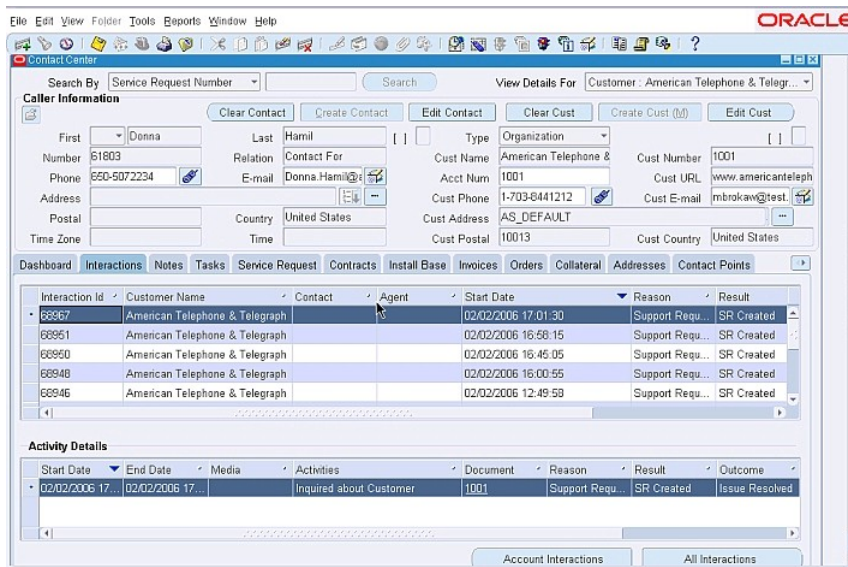


図2 : Oracle Formsによる機能豊富なUIの例

Builderからブラウザへの直接実行

Oracle Forms では、Form Builder からアプリケーションを起動した場合、そのアプリケーションを任意のブラウザ、または Builder のプリファレンスに構成するブラウザから起動できます。これで、PJC などのコンポーネントが完全に機能した状態で、デプロイ後の外観が正確に表現されます。

N-TN層の“リモート”デバッグ

フォームは、Builder のデバッグ・モードで直接実行したり、より効果的に、ネットワーク経由でリモート・ランタイム・セッションに動的に割り当てたりすることもできます。つまり、エンドユーザーと協力して作業し、そのユーザーによって開発上のバグが誘発される操作がアプリケーションで実行されることを監視し、コードやスタックを確認できます。

Forms APIおよびJava

Forms Application Program Interface (API) は、Forms 5.0 で導入されました。このため開発者は、Forms Builder で Forms モジュールを開かずに、問合せや操作をバッチ処理できます。この API は C 言語ベースであるため、“パワー・ユーザー”向けのツールと考えられますが、オラクルは、現在の Oracle Forms でも引き続き C API を採用し、Java バージョンの Java Development API (JDAPI) も導入しました。Java は誰もが習得を目指す言語であり、実際に使用すると API が簡潔になり、理解しやすく使用が容易になります。

XML表現

Forms API は、依存性の分析、差分の取得、大幅な変更などに役立つツールです。市場には、この API を使用して同様のサービスを提供するサード・パーティ製のツールもありますが、オラクルでは、Oracle Forms でこれらの操作をより簡単かつ利用しやすくするために、XML コンバータを Forms に提供しています。このユーティリティは、Forms モジュールをドキュメント化された XML 形式に変換します。この形式は、スタイル・シートを使って、カスタム出力やレポートへ編集または変換できます。

XML ユーティリティは双方向の操作です。Forms モジュールを XML 表現から再作成できるため、同様に提供されている FMT ファイル形式の代替ストレージ形式として最適です。

アクセシビリティ

Oracle Forms では、ビルダーおよびランタイムを、Job Access With Speech (JAWS) スクリーン・リーダーと一緒に使用できます。また、マウスを使用せずに優れた操作性を実現したり、高コントラストのカラー・スキーマに対応したりなど、その他のアクセシビリティ要件にも対応しています。

Oracle Fusion Middleware Forms Services – アーキテクチャ

Oracle Fusion Middleware Forms Services は 3 つのコンポーネントで構成されています。エンドユーザーのブラウザに自動的にダウンロードおよびキャッシュされる Forms クライアント、Forms リスナー・サーブレット、および Forms ランタイムです。いずれも中間層に常駐します。

Formsクライアント

ユーザーが Forms セッションを実行すると、完全に thin Java アプレットである Forms クライアントが、Oracle Fusion Middleware Application Server から動的にダウンロードされます。この汎用 Java アプレットは、中間層での関連する Forms ランタイム・プロセスにユーザー・インタフェースを提供したり、項目間のナビゲーションやチェック・ボックスをオンにする操作などで生成されるユーザーの操作や視覚的なフィードバックを処理したりします。すべての Forms アプリケーションでは同じ Java アプレットが使用されるため、Java アプレットを 1 度ダウンロードしてクライアントにキャッシュすると、以降の Forms アプリケーションで繰り返し使用できます。

Java アプレットをブラウザで実行するには、Java プラグインのインストールが必要です。このプラグインはクライアント側にインストールされ、プラットフォームに依存します。オラクルでは、この Oracle Forms Services リリース向けに、Java プラグイン 8U51 以降を推奨および認定しています。

Oracle Forms では、Java Web Start や Forms Standalone Launcher を使った、ブラウザを使用しない実行もサポートされています。クライアント・マシンへの Java のインストールは必須ですが、プラグインのインストールはオプションです。代わりに、Java Development Kit (JDK) のインストールを使用できます。

Formsランタイム・プロセス

Forms ランタイム・プロセスは、Forms クライアントに代わりデータベースとの接続を管理するプロセスです。このプロセスは、Forms アプリケーションが含まれるページにユーザーがアクセスしたときに作成されます。Forms アプリケーションを閉じたり、ブラウザ・ウィンドウを終了したりすると、プロセスはすぐに自動的に停止します。

Formsリスナー・サーブレット

Forms リスナー・サーブレットで管理される内容は次のとおりです。

- » ユーザーが Forms アプリケーションの実行を要求したときの、各クライアントに対する Forms ランタイム・プロセスの作成
- » ユーザーが Forms アプリケーションを閉じたりブラウザ・ウィンドウを終了したりした場合のランタイム・プロセスの停止
- » クライアント、およびクライアントに関連する Forms ランタイム・プロセス間のネットワーク通信

接続プロセスの概要

Oracle Fusion Middleware Forms Services アーキテクチャには、クライアントと HTTP リスナー間に Web ベースのアプリケーションに似た接続が 1 つ存在します。HTTP リスナーは、リクエストを Forms リスナー・サーブレットヘルパーティングし、Forms リスナー・サーブレットは、Forms クライアントから Forms ランタイムヘルクエストをルーティングします。

Forms クライアントと Forms ランタイム間の通信は常に HTTP リスナーを経由して行われるため、アプリケーションからインターネットに対して開いているポートは 1 つだけです。

このシナリオでは、クライアントが HTTP リクエストを送信して、HTTP リスナー・プロセスから HTTP レスポンスを受信します。HTTP リスナーはクライアントのネットワーク・エンドポイントとして機能するため、次の図のように、他のサーバー・マシンとポートはファイアウォールに保護されます。

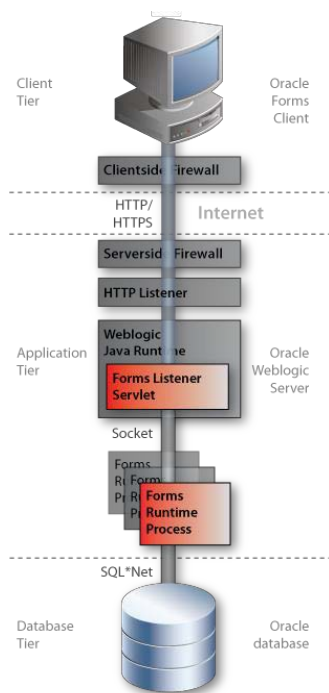


図3：アーキテクチャの概要

接続プロセスの詳細

1. ユーザーは Web ページからハイパーリンクを選択するか、ブラウザに URL を直接入力します。
2. HTTP リスナーは、渡された URL を解釈し、Forms Java クライアントを記述する <OBJECT>タグ（構成によっては JNLP タグ）が含まれる HTML ページをクライアントのブラウザに表示します。渡された URL によって、Forms サブレットが呼び出され、アプリケーション・サーバーにある html テンプレート・ファイルに基づいて、HTML ページが動的に作成されます。
3. クライアントは、HTTP リスナーから提供された HTML ファイルを受け取ります。この HTML ファイル内のタグには、Forms Java クライアントを構成する Java クラス・ファイルの特定に必要な情報が含まれています。HTML ファイル内のタグ内には、実行するフォーム、および Forms セッションに渡すその他のパラメータ（ログイン情報など）に関する情報も含まれています。タグ定義には、実行する Forms Services の指示、（ルック・アンド・フィールド、カラー・スキーマなど）Java クライアントの要素をカスタマイズする際に役立つ多数のパラメータも含まれます。また、この HTML ファイルには、他の HTML 属性（クライアント上の特定のバージョンの JRE を使用して、ブラウザに特定のアプレットの実行を通知するなど）が含まれる場合もあります。
4. クライアントに戻される html ページには、JRE（または Web Start。構成によって異なる）の起動に必要な情報も含まれます。クライアント側の JVM が起動すると、ページ内のいくつかのパラメータによって必要なアーカイブ（jar）ファイルがリクエストされ、それらのファイル内の起動 Java クラス・ファイルが特定されます。このアーカイブの拡張子は JAR で、アプレットで必要な個々の CLASS ファイルがすべて含まれた ZIP ファイルであると考

えることができます。JAR ファイルを使用すると、Java クライアントのダウンロードが高速になり、クライアントをキャッシュして、以降のコールで使用できます。使用するJAR ファイルを ARCHIVE パラメータで定義します。クライアント JVM (Java プラグインと Java Web Start エンジンのコンポーネント) によって、Forms クライアントのキャッシュされた Java コードのタイムスタンプ (キャッシュされたバージョンが存在する場合) とサーバー・バージョンの比較という追加手順が実行されます。新しい jar がダウンロードされるのは、プラグイン・キャッシュに保存されたタイムスタンプ付き jar よりサーバーのほうが新しい場合のみです。

5. JAR ファイルがダウンロードされていない場合は、これらをダウンロードし、Java アプレットを開始します。
6. Java クライアント・アプレットは、HTTP リスナーと Forms リスナー・サーブレットを経由して、Forms セッションを開始するリクエストを送信します。Forms リスナー・サーブレットは、HTML ファイルの serverURL パラメータによって定義されます。
7. Java クライアントから接続リクエストを受信すると、Forms リスナー・サーブレットは、このクライアントに対して新しい Forms ランタイム・プロセスを開始します。構成ファイル内で、サーブレットの初期化 envFile パラメータ (formsweb.cfg) を特定の環境ファイルに設定すると、ランタイム・プロセスごとにユーザー固有の環境を定義できます。次の図では、異なる言語をサポートする 3 つのファイル (French.env、Swedish.env、English.env) が作成されています。デフォルト・ファイルは default.env で、このファイルには、たとえば、Forms ORACLE_HOME および FORMS_PATH が含まれています。
8. このクライアントに割り当てられた Forms ランタイム・プロセスは、HTML ファイルで指定されたモジュールと、このフォームに必要なすべてのライブラリとメニューをロードします。Forms クライアントと Forms ランタイム・プロセス間のすべての通信は、Forms リスナー・サーブレット経由で渡されます。
9. ユーザーがデータベースのログイン情報を入力していない場合は、その情報を入力した後、データベース・サーバーへの接続が確立されます。
10. これで、ユーザーが作業できる状態になります。

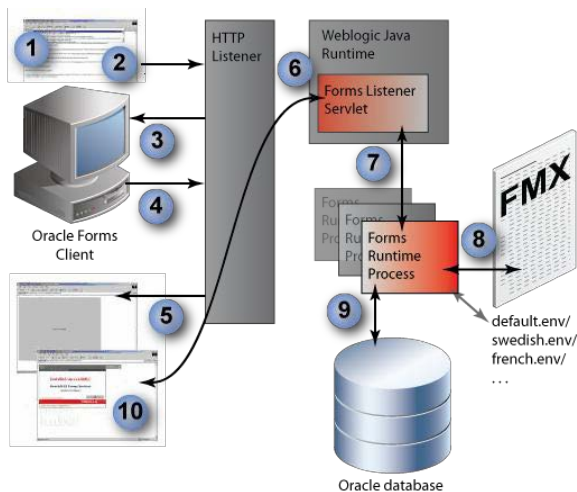


図4：接続プロセスの概要

Oracle Forms Servicesの利点

Forms リスナー・サーブレットは、インターネット上で Forms アプリケーションを堅牢かつ標準的にデプロイできるように設計されています。Forms リスナー・サーブレットには、次の利点があります。

幅広いファイアウォールとプロキシのサポート

クライアント・ブラウザは、HTTP または HTTPS を使って HTTP リスナーと常に通信しています（クライアントと Forms ランタイム・プロセス間には直接的な接続は存在しません）。そのため、このアーキテクチャは、サーブレット・セッションを使用した標準サーブレットで機能するすべてのファイアウォールまたはプロキシをサポートします。

プロトコルの制限がない（HTTP/1.1またはHTTP/1.0）

標準の通信が使用されるため、HTTP/1.0 と HTTP/1.1 の両方がサポートされています。パフォーマンスは HTTP/1.1 の方が優れていますが、サーバーのリソース消費量が増加します。

SSLデプロイメントのための証明書の購入は不要

SSL (Secure Sockets Layer) を使用したデプロイメントには、クライアント・ブラウザと HTTP リスナー間で HTTPS 接続が発生します。このため、Forms サーバー・レベルでは、特別なセキュリティの設定要件はありません。

標準ロードバランシングのサポート

このアーキテクチャでは、ハードウェアベースのロードバランシング、リバース・プロキシ、標準サーブレット・ロードバランシングなど、標準的なロードバランシング技術を使用できます（詳細は後述します）。

Forms Javaクライアント用のJava 7やJava 8をサポート

12c (12.2.1) リリースでは、Forms Java クライアントで Java 7 や Java 8 を使用できます（正確なバージョンのリストについては、製品サーティフィケーション・マトリックスを参照してください）。プラグブル Java コンポーネントを記述する開発者や、Forms Java クライアントを他の Java アプレットや JavaBeans と統合する開発者は、Java 7 や Java 8 の API の機能をフルに活用できます。

中間層でのJava 8のサポート

Java Importer の機能は Forms の統合戦略の主要部分であり、PL/SQL から Java を簡単に統合できます。また、Java Importer は Java 8 でも使用できるため、Forms アプリケーションを最新の外部プログラムやサービスと統合し、中間層で Java クラスを呼び出すことができます。

グローバルなデプロイメント

さまざまなネットワークやシステムを介して、さらに多くのユーザーに対してアプリケーションをデプロイしている企業の例は増え続けています。ネットワークやサーバーの構成の複雑さが問題になる場合もあります。今回の Oracle Forms は、Web での使用を前提としたリリースであり、アプリケーションのグローバルなデプロイメントという課題に対応できます。

あらゆるネットワークで実行可能

Forms リスナー・サーブレットのアーキテクチャを使用すれば、堅牢かつ標準的な方法で Oracle Forms アプリケーションをデプロイできます。Java サーブレットを使用すれば、インターネット、イントラネット、エクストラネットなどのあらゆるネットワークでアプリケーションを実行できます。ファイアウォールでは、標準ポートのみをオープンする必要があります。また、認証プロキシがサポートされています。

Forms Services では HTTP リスナーの証明書を使用します。セキュアな接続を実現するために SSL 証明書を別途用意する必要はありません。また、Forms リスナー・サーブレットのアーキテクチャでは、Forms リスナーを管理するための管理オーバーヘッドが発生せず、HTTP リスナーまたはハードウェアの標準的なロードバランシング機能全体を利用できます。

1度みのログイン

企業の情報インフラストラクチャのオンライン化は、ますます進んでいます。そのため、アクセスするアプリケーションごとに別々のユーザー名とパスワードを管理するユーザーが増えてきました。複数のアカウントとパスワードの管理にはコストがかかり、安全な方法ではないうえに実用的ではありません。

Oracle Forms では、Oracle Access Manager および Web Gate による認証サービス (SSO) を使用できます。Oracle Access Manager および Web Gate は Oracle Fusion Middleware のコンポーネントで、さまざまな LDAP サーバーと統合されています (サポートされる LDAP サーバーについては、Oracle Access Manager のドキュメントを参照してください)。シングル・サインオンを使用すると、ユーザーは 1 回のログインで済むため、複数のパスワードを覚えておく必要がありません。また、Forms アプリケーション・モジュールでの追加のコーディングも不要です。

あらゆるタイムゾーンでアプリケーションの実行が可能

アプリケーションをデプロイする際に見過ごされがちな重要なポイントとして、時間という根本的な概念があります。ユーザーがアプリケーションに時刻を記録する場合、それは正確には何を意味するのでしょうか。そのユーザーがいる場所の時刻でしょうか。それとも Forms Services が実行されている場所の時刻でしょうか。またはデータベース・サーバー上の時刻でしょうか。

次のような状況を想像してみてください。英国に本拠地をおくコール・センター窓口で、ハワイで休暇中の顧客のクレジット・カードの盗難時刻を現地 (ハワイ) 時間で聞き取り、そのデータがニューヨークのデータベースに保存されます。さらにこのデータを、シンガポールの Web サイトで発生したインターネット経由のカードの不正使用と照合する必要があるとしたら、全世界に共通の時間概念がいかに重要であるかが理解できると思います。

Oracle Forms はアプリケーションのすべてのレベルでタイムゾーンを定義する機能をサポートしているため、正確な時刻がデータベースに保存されます。

ブラウザ言語の検出

Oracle Forms には、クライアントのブラウザの言語設定を自動的に検出する機能があります。この機能を使用すると、ユーザーは同じ URL を介してアプリケーションにアクセスでき、それぞれの言語またはブラウザで設定された言語を使用してアプリケーションへ自動的にリダイレクトされます。

Oracle データベースとの緊密な統合

オラクルは、あらゆる業界におけるデータベース・サーバーの世界的なリーダーであり、グローバル市場で 60% のシェアを占めています。オラクル製品は管理性、可用性、使いやすさに優れています。オラクル製品は、組織内のあらゆるインターネット・コンテンツに対して、統合されたデータ管理を提供します。Oracle Forms Developer は、特にオラクルのトランザクション・データベース・アプリケーションを構築するために設計および最適化され、Oracle データベース用に設計されています。Oracle Forms Developer では、次のサービスをネイティブに提供しているため、手動によるコーディングが不要です。

- » Oracle データベースへの接続と、その接続の管理
- » 大量のレコードの必要に応じた問合せと処理
- » データベース・レコードの必要に応じたロック
- » マルチユーザーをロックするシナリオを自動的にサポートするコードの生成
- » 挿入、更新、削除を自動的に管理
- » 開発者がレコード・セットをプログラムによって操作
- » データベースとの間でトランザクションをアトミックかつ効率的にやり取りすることで、トランザクションを完全に利用するか、まったく利用しない
- » データベースのアドバンスト・キューイング・キューとの通信の自動処理
- » デプロイメントにデータベース・プロキシ・ユーザーが使用されている場合にログインを自動的に処理

データ・ブロック・ウィザードを使用すると、データベース内の表にアプリケーションを自動的にリンクさせ、複雑なマスター/ディテール Web フォームを簡単に構築し、複数のデータセットを自動的に同期化できます。データの問合せ、挿入、更新、削除は、フォームを使用し、その変更内容を Web ですぐにプレビューできます。このように Oracle Forms Developer と Oracle データベースをシームレスに統合することで、リソース管理、アドバンスト・キューイング、サブスクリプション、分散問合せ、パーティショニング、パラレル・サーバーを使用し、アプリケーション間の効率的なリソース共有を可能にするインターネット・アプリケーションを構築し、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を改善できます。最適化された分散問合せの計画と、その他のレプリケーション機能の強化によって、複数の同種および異種の環境にわたる問合せのパフォーマンスを改善します。オラクルのセキュリティ・ポリシー管理機能によって、詳細なレベルの集中アクセス制御および監査を実装します。

Oracleデータベースの国際化のサポート

Forms は、アプリケーションのデプロイ先の言語に合わせて、多言語アプリケーションで使用する文字変数の長さを調整できる、Oracle データベースの機能をサポートしています。次の宣言を例に説明します。

```
my_string VARCHAR2(100);
```

この変数は 100 バイトを保持しています。英語または西ヨーロッパの標準的なキャラクタ・セット (US7ASCII や WE8ISO8869P1 など) では、この変数の長さは 100 文字です。ただし、マルチバイトのキャラクタ・セット (JA16SJIS や UTF8 など) を使用している場合は、この変数が保持できるのは 50 文字以下です。新しいキャラクタ・セマンティクスの機能は、ベースとなるキャラクタ・セットに関係なく、必要な文字の絶対数によって変数を宣言することによって、この問題を解決します。

パフォーマンスとスケーラビリティ

Web デプロイメントの管理オーバーヘッドも大幅に低減し、クライアント/サーバーの実装により期待どおりのパフォーマンスを提供するため、Oracle Fusion Middleware Forms Services ではあらゆる改善を行ってきました。

ロードバランシング

Oracle Forms のアーキテクチャは標準ベースのサーブレット・アーキテクチャを採用しているため、Forms 固有のロードバランシング・ユーティリティは不要です。Oracle Fusion Middleware Forms Services は、ロードバランシング機能を持つ Oracle WebLogic Server インフラストラクチャと Oracle HTTP Server を利用します。

また、このアーキテクチャでは、ハードウェアベースのロードバランシング、管理対象サーバーのクラスタリング、リバース・プロキシなど、標準的なロードバランシング技術を使用できます。ロードバランシングの設定方法については、Oracle Fusion Middleware のドキュメントを参照してください。

結論

多くの企業の情報システムにおいては、変化するビジネス要件や新しいテクノロジーに迅速に対応することが求められています。開発チームは、常に、アプリケーションの機能の拡張、ユーザー・インタフェースの改善、より複雑で高パフォーマンスなデプロイメント構成に対する要望に直面しています。Oracle Forms Developer および Oracle Fusion Middleware Forms Services のスケーラブルで柔軟なアーキテクチャによって、高パフォーマンスなエンタープライズ・クラスのアプリケーションをあらゆるユーザーに自動的に提供できます。オラクルのお客様は、Oracle Fusion Middleware Forms Services を使用しています。

世界中の何千人ものユーザーをサポートし、何千人ものユーザーが同時に実行した場合でも優れたパフォーマンスを発揮することは、ベンチマークで証明されています。Oracle Internet Platform が持つオープン性と本来のスケーラビリティの特性を活用し、Oracle Forms Developer と Oracle Fusion Middleware Forms Services は、自動的に拡張および実行するインターネット・アプリケーションの統合デリバリー環境を提供します。



Oracle Corporation, World Headquarters

500 Oracle Parkway
Redwood Shores, CA 94065, USA

海外からのお問い合わせ窓口

電話：+1.650.506.7000
ファクシミリ：+1.650.506.7200

CONNECT WITH US

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

Integrated Cloud Applications & Platform Services

Copyright © 2016, 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、記載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 および Java は Oracle およびその子会社、関連会社の登録商標です。その他の名称はそれぞれの会社の商標です。

Intel および Intel Xeon は Intel Corporation の商標または登録商標です。すべての SPARC 商標はライセンスに基づいて使用される SPARC International, Inc. の商標または登録商標です。AMD、Opteron、AMD ロゴおよび AMD Opteron ロゴは、Advanced Micro Devices の商標または登録商標です。UNIX は、The Open Group の登録商標です。0615

Oracle Forms Services と Oracle Forms Developer 12c の技術概要 2016 年 1 月

著者：Michael Ferrante

共著者：Oracle Product Management



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