

Oracle9iAS Forms Services

利用ガイド

リリース 9.0.2

2002 年 8 月

部品番号 : J05989-01

ORACLE®

Oracle9iAS Forms Services 利用ガイド, リリース 9.0.2

部品番号 : J05989-01

原本名 : Oracle9iAS Forms Services Deployment Guide, Release 9.0.2

原本部品番号 : A92175-01

原本著者 : Orlando Cordero, Tom Pfaeffle, Dean Ho, Duncan Mills, Frank Nimphius, Cathy Godwin, Bryan Roberts

Copyright © 1993, 2002, Oracle Corporation. All rights reserved.

Printed in Japan.

制限付権利の説明

プログラム（ソフトウェアおよびドキュメントを含む）の使用、複製または開示は、オラクル社との契約に記された制約条件に従うものとします。著作権、特許権およびその他の知的財産権に関する法律により保護されています。

当プログラムのリバース・エンジニアリング等は禁止されています。

このドキュメントの情報は、予告なしに変更されることがあります。オラクル社は本ドキュメントの無謬性を保証しません。

* オラクル社とは、Oracle Corporation（米国オラクル）または日本オラクル株式会社（日本オラクル）を指します。

危険な用途への使用について

オラクル社製品は、原子力、航空産業、大量輸送、医療あるいはその他の危険が伴うアプリケーションに用途として開発されておられません。オラクル社製品を上述のようなアプリケーションに使用することについての安全確保は、顧客各位の責任と費用により行ってください。万一かかる用途での使用によりクレームや損害が発生いたしましても、日本オラクル株式会社と開発元である Oracle Corporation（米国オラクル）およびその関連会社は一切責任を負いかねます。当プログラムを米国国防総省の米国政府機関に提供する際には、『Restricted Rights』と共に提供してください。この場合次の Notice が適用されます。

Restricted Rights Notice

Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the Programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

このドキュメントに記載されているその他の会社名および製品名は、あくまでその製品および会社を識別する目的にのみ使用されており、それぞれの所有者の商標または登録商標です。

目次

はじめに	xi
対象読者	xi
このマニュアルの構成	xi
関連文書	xii

1 概要

概要	1-1
Oracle インターネット・プラットフォーム	1-2
Oracle9i Application Server (Oracle9iAS)	1-2
Oracle9i Developer Suite (Oracle9iDS)	1-2
Oracle9i Database	1-2
Oracle9iAS Forms Services	1-3
Oracle9iAS Forms Services の新機能	1-3
Oracle9iAS Forms Services のアーキテクチャ	1-4
Oracle9iAS Forms Services のコンポーネント	1-5
Forms Listener Servlet	1-6
Forms ランタイム・プロセス	1-6
Forms Listener Servlet	1-6

2 Oracle9i Forms アプリケーションの配布の基本

概要	2-1
構成ファイル	2-2
Oracle9i Forms の構成ファイル	2-2
formswb.cfg	2-2
base.htm、basejini.htm、basejpi.htm および baseie.htm	2-3

fttrace.cfg	2-3
Oracle9iAS Containers for J2EE (OC4J) の構成ファイル	2-3
web.xml	2-3
Oracle9i Forms OC4J ファイルのディレクトリ構造	2-4
Oracle HTTP Listener の構成ファイル	2-4
forms90.conf	2-4
標準フォントとアイコン・ファイル	2-5
Registry.dat	2-5
環境変数	2-5
default.env	2-5
アプリケーションの配布	2-5
アプリケーションの配布	2-6
パラメータの指定	2-7
Oracle9iAS Forms Services のアクション	2-9
クライアント・ブラウザのサポート	2-12
Oracle JInitiator	2-12
構成パラメータとベース HTML ファイルをクライアント・ブラウザに連結する方法	2-12

3 Oracle9iAS Forms Services の構成

概要	3-1
構成ファイルのカスタマイズ	3-2
formsweb.cfg	3-2
ファイルのネーミング・パラメータ	3-2
formsweb.cfg で特別な名前を付けた構成の作成	3-2
formsweb.cfg ファイル内のパラメータ	3-3
デフォルトの formsweb.cfg ファイル	3-8
base.htm、basejini.htm、basejpi.htm および baseie.htm	3-12
ベース HTML ファイル内のパラメータと変数	3-14
使用方法	3-14
デフォルトの base.htm ファイル	3-15
デフォルトの basejini.htm ファイル	3-16
デフォルトの basejpi.htm ファイル	3-18
デフォルトの baseie.htm ファイル	3-20
web.xml	3-21
デフォルトの web.xml ファイル	3-21
forms90.conf	3-23
デフォルトの forms90.conf	3-24

Registry.dat	3-25
デフォルトの Registry.dat	3-26
環境変数とレジストリ設定のカスタマイズ	3-28
default.env	3-28
デフォルトの Windows 用 default.env ファイル	3-31
デフォルトの UNIX 用 default.env ファイル	3-33
ユーザー独自のテンプレート HTML ファイルの作成	3-35
Oracle9i Forms アプリケーションへのグラフィックの挿入	3-35
Oracle9iAS Forms Services で使用するアイコンとイメージの配置	3-35
アイコン	3-35
アイコンの Java アーカイブへの格納	3-36
Registry.dat へのアイコン変更の追加	3-36
スプラッシュ画面イメージおよびバックグラウンド・イメージ	3-38
アイコンとイメージを含むカスタム JAR ファイル	3-38
JAR ファイルの作成	3-39
JAR ファイル内でのファイルの使用	3-39
アイコンおよびイメージの検索パス	3-39
DocumentBase	3-40
CodeBase	3-41
Forms Listener Servlet での HTTPS の使用	3-41
サーバー要件	3-41
クライアント要件 : Oracle JInitiator での HTTPS の使用	3-42
ユーザー ID/ パスワード非表示機能の使用	3-42
認証プロキシを使用した Oracle9i Forms アプリケーションの実行	3-42
言語検出の有効化	3-43
言語検出機能の仕組み	3-43
マルチレベルの継承	3-44

4 HTTP Listener と OC4J での Oracle9iAS Forms Services の使用

概要	4-1
OC4J サーバー・プロセス	4-2
パフォーマンス / 拡張性のチューニング	4-3
HTTPD プロセス数の制限	4-3
高い値への maxClient ディレクティブの設定	4-3
OC4J のロード・バランス	4-4
ケース 1: Oracle HTTP Listener と同じホスト上にある複数の OC4J エンジン	4-5
ケース 2: Oracle HTTP Listener と異なるホスト上にある複数の OC4J エンジン	4-5

ケース 3: それぞれ異なるホスト上にある複数の OC4J エンジンと複数の Oracle HTTP Listeners	4-6
ケース 4: 異なるホスト上にある複数の Oracle HTTP Listeners と同じホスト上にある複数の OC4J エンジン	4-7

5 SSO と OID での Oracle9iAS Forms Services の使用

概要	5-1
Single Sign-On (SSO)	5-2
認証フロー	5-3
Single Sign-On と特定のアプリケーション	5-4
2 目目のスタンドアロン・サーバー・インスタンスの作成	5-5
Forms Servlet 別名の追加	5-5
Oracle9i Forms 別名の mod_oc4j への登録	5-5

6 Oracle Enterprise Manager と Oracle9i Forms

7 トレースと診断

Forms Trace	7-2
Forms Trace の構成	7-2
ftrace.cfg	7-2
URL パラメータ・オプション	7-3
トレースの開始	7-5
Forms Trace 出力の表示	7-5
Upload/Translate ユーティリティの実行	7-6
トレース・データのデータベース表の作成	7-6
トレース可能イベントのリスト	7-8
イベントの詳細のリスト	7-11
エラー	7-11
ユーザー・アクション・イベント	7-12
Forms Services のイベント	7-12
イベントの詳細	7-13
3 層イベント	7-13
その他	7-14
Servlet Logging Tools	7-14
ロギングの起動	7-14
URL でのロギングの指定	7-15
formsweb.cfg ファイルでのロギングの指定	7-15
Forms Servlet の起動に使用した URL での完全診断の指定	7-15

ログ・ファイルのディレクトリ	7-16
各レベルのサブレット・ロギングの出力例	7-16
(なし)	7-16
/session	7-16
/sessionperf	7-16
/perf	7-17
/debug	7-17
Oracle Trace	7-18

8 パフォーマンス・チューニングに関する考慮事項

概要	8-1
Forms Services のビルトイン最適化機能	8-2
クライアント・リソース要件の最小化	8-2
Forms Services リソース要件の最小化	8-2
ネットワーク使用量の最小化	8-3
ネットワークを介して送信されるパケットの効率の拡大	8-3
クライアントでのアプリケーション画面の効率的なレンダリング	8-4
Oracle9iAS Forms Services アプリケーションのチューニング	8-4
データ・サーバーに対する Oracle9iAS Forms Services の位置	8-4
アプリケーションの起動時間の最小化	8-5
Java ファイルの使用	8-6
Oracle JInitiator	8-6
IE ネイティブ JVM	8-7
その他のすべての場合（Sun の Java Plug-in など）	8-8
キャッシュの使用	8-8
必須ネットワーク帯域幅の削減	8-8
パフォーマンスを改善するためのその他の方法	8-11

A JInitiator

Oracle JInitiator	A-1
Oracle JInitiator を使用する理由	A-1
Oracle JInitiator の利点	A-2
Oracle JInitiator の使用方法	A-2
サポートされる構成	A-2
Windows 98、NT、2000 および XP:	A-2

システム要件	A-3
Netscape Navigator での Oracle JInitiator の使用方法	A-3
Microsoft Internet Explorer での Oracle JInitiator の使用方法	A-3
Oracle JInitiator プラグインの設定	A-4
Oracle JInitiator マークアップのベース HTML ファイルへの追加	A-4
Oracle JInitiator ダウンロード・ファイルのカスタマイズ	A-4
Oracle JInitiator をダウンロード可能にする	A-4
Oracle JInitiator プラグインの変更	A-5
Oracle JInitiator キャッシュ・サイズの変更	A-5
Oracle JInitiator ヒープ・サイズの変更	A-5
Oracle JInitiator 用のプロキシ・サーバー設定のチェックおよび変更	A-5
Oracle JInitiator 出力の表示	A-6
ベース HTML ファイルの変更	A-6

索引

図リスト

1-1	Oracle9iAS Forms Services アーキテクチャ	1-4
1-2	フォームを実行する 3 層構造	1-5
1-3	Forms Listener Servlet を使用したアーキテクチャ	1-7
8-1	Oracle9iAS Forms Services とデータベース・サーバーの並存	8-5

表リスト

3-1	forms90.conf の仮想パスとサーブレットのマッピング	3-23
3-2	アイコンのディレクトリ・ガイド	3-37
3-3	アイコンおよびイメージの検索パス	3-40
3-4	Oracle9iAS Forms Services で使用されるアイコンとイメージの検索パス	3-41
7-1	Forms Trace のコマンド・ライン・パラメータ	7-3
7-2	Translate ユーティリティのコマンド・ライン・オプション	7-6
7-3	Forms トレース・データ用のデータベース・スキーマ	7-7
7-4	トレース可能イベントのリスト	7-9
7-5	ユーザー・アクション・イベントの詳細	7-12
7-6	Forms Services のイベントの詳細	7-12
7-7	イベントの詳細	7-13
7-8	3 層イベントの詳細	7-13
7-9	その他のイベントの詳細	7-14
7-10	サポートされているロギング機能	7-15

はじめに

対象読者

このマニュアルは、Oracle9i Application Server を使用して Oracle9i Forms アプリケーションを Web に配布するソフトウェア開発者を対象としています。

このマニュアルの構成

このマニュアルには、次の章と付録が含まれています。

- | | |
|-------|---|
| 第 1 章 | 概要
Oracle インターネット・プラットフォームでのアプリケーションの配布、および Forms Services のアーキテクチャおよびコンポーネントの概要について説明します。 |
| 第 2 章 | Oracle9i Forms アプリケーションの配布の基本
Oracle9iAS Forms Services の構成に必要な基本ファイルとアプリケーションの配布手順について説明します。 |
| 第 3 章 | Oracle9iAS Forms Services の構成
Oracle9iAS Forms Services の構成に関する詳細、および Web で利用できる Oracle9i Forms アプリケーションの設計に関するガイドラインとヒントを説明します。 |
| 第 4 章 | HTTP Listener と OC4J での Oracle9iAS Forms Services の使用
Oracle HTTP Listener と OC4J によって Oracle9iAS Forms Services がどのように動作するかを説明します。 |

第 5 章	SSO と OID での Oracle9iAS Forms Services の使用 Oracle9iAS Forms Services で、シングル・サインオンと Oracle Internet Directory を使用する方法について説明します。
第 6 章	Oracle Enterprise Manager と Oracle9i Forms Oracle Enterprise Manager (OEM) システム管理ツールについて説明します。
第 7 章	トレースと診断 Forms に付属のトレース・ツールおよび診断ツールを使用して、Oracle9i Forms アプリケーションの実行時におけるパフォーマンスとリソース消費を分析する方法について説明します。
第 8 章	パフォーマンス・チューニングに関する 考慮事項 Oracle9iAS Forms Services を使用してアプリケーションを配布するときの、チューニングに関する考慮事項を説明します。
付録 A	JInitiator Oracle JInitiator の使用における利点および Oracle JInitiator プラグインの設定方法について説明します。

関連文書

詳細は、次のマニュアルを参照してください。

- 『Oracle9i Forms 日本語リリース・ノート』
- 『Oracle9i Forms Developer and Forms Services Forms アプリケーションの Forms6i からの移行』
- Forms Developer オンライン・ヘルプ (Forms Developer の「ヘルプ」メニューから利用可能)

概要

このマニュアルでは、Oracle9iAS Forms Services を使用したアプリケーションの配布について説明します。アプリケーションをインターネット上に配布することを選択した場合、その実現方法に関して多くのことを決定する必要があります。このマニュアルでは、これらの決定に関する情報を提供します。また、アプリケーションを Web に配布するシステムを構成するための提案およびメソッドを提供します。このマニュアルの各章の構成は、「はじめに」を参照してください。

この章には、次の項が含まれています。

- [Oracle インターネット・プラットフォーム](#)
- [Oracle9iAS Forms Services](#)
- [Oracle9iAS Forms Services のアーキテクチャ](#)
- [Oracle9iAS Forms Services のコンポーネント](#)
- [Forms Listener Servlet](#)

Oracle インターネット・プラットフォーム

Oracle インターネット・プラットフォームは、Oracle9i データベースでデータを管理し、Oracle9i Developer Suite (Oracle9iDS) でアプリケーションを作成して、Oracle9i Application Server (Oracle9iAS) でアプリケーションを実行します。このインターネット・プラットフォームは、あらゆるタイプのアプリケーションを作成し、Web へ配布できる完全なソリューションです。これらの Oracle ツールによって、拡張と高度な利用が可能なインフラストラクチャが提供されるため、お客さまはユーザー数の増加に容易に対応できます。

Oracle のシンプルかつ完全な統合インターネット・プラットフォームは、次の 3 つのコア製品で構成されます。

- [Oracle9i Application Server \(Oracle9iAS\)](#)
- [Oracle9i Developer Suite \(Oracle9iDS\)](#)
- [Oracle9i Database](#)

Oracle9i Application Server (Oracle9iAS)

Oracle9iAS は、スケーラブルでセキュアな中間層のアプリケーション・サーバーです。Oracle9iAS を使用すると、Web コンテンツの配信、Web アプリケーションのホスティング、バックオフィス・アプリケーションへの接続を行うことができます。Forms Services は Oracle9i Application Server の必須要素として、インターネット・コンピューティングの利点を最大限に実現するテクノロジーを提供します。

Oracle9i Developer Suite (Oracle9iDS)

Oracle9iDS は、Oracle Application Development Tools、Oracle Business Intelligence Tools、XML Developer's Kit (XDK)、そして Portlet Development Kit (PDK) をスイートとして統合した製品です。

Oracle9iDS は、J2EE、XML、SOAP、UDDI、UML などのインターネット標準に基づいて、Oracle9iAS と Oracle9i Database 用のアプリケーションを作成する生産性の高い環境を提供します。

Oracle9i Database

Oracle9i Database は、世界で最も一般的な RDBMS の新世代の製品です。Oracle9i Real Application Clusters の無限の拡張性と業界随一の信頼性、スタンバイ・データベース・テクノロジーの利点を採り入れた新しい高可用性テクノロジー (Oracle Data Guard)、ビルトイン OLAP、データ・マイニング機能、そして ETL 機能など数多くの最新機能が盛り込まれています。

Oracle9i Application Server は、Oracle9i Database と、Oracle 開発ツールで作成されたアプリケーションにとって最適なアプリケーション・サーバーです。Oracle9i Application Server

では、共通技術スタックを使用して、データとアプリケーション・ロジックを中間層にキャッシュすることで、Oracle データベースを透過的に拡張できます。

Oracle9iAS Forms Services

Oracle9iAS Forms Services は、Oracle9iAS の一部として、Oracle9i Forms の新規および既存のアプリケーションを World Wide Web に配布できる新世代のツールです。

Oracle9iAS Forms Services は、Oracle9i Forms アプリケーションを複数層環境に配布するために最適化された包括的なアプリケーション・フレームワークです。Oracle9iAS Forms Services では、Web の使用やアクセスが簡単なことを利用し、Web に単なる静的な情報公開メカニズムを超えた、複雑で動的なアプリケーションをサポートできる環境としての機能を与えます。

Oracle9iAS Forms Services の新機能

Forms6i Server に搭載されていた機能の大半は、Oracle9iAS で提供されるコンポーネントに引き継がれています。たとえば、ロード・バランシング、セキュリティ、拡張性、HTTP/S 通信処理、Java サブプレットの配布機能は、すべて Oracle9iAS で提供される Oracle HTTP Server や Oracle9iAS Containers for J2EE (OC4J) などの各種コンポーネントによって実行されます。

Oracle9iAS の Oracle9iAS Forms Services コンポーネントは、Oracle9i Forms アプリケーションに定義されたビジネス・ロジックの実行や Oracle データベースとの接続など、Oracle9i Forms アプリケーション固有のプロセスをすべて処理します。Java アプレットは、クライアントのユーザー・インタフェースを提供します。

Oracle9iAS Forms Services には、次のような新しい特長があります。

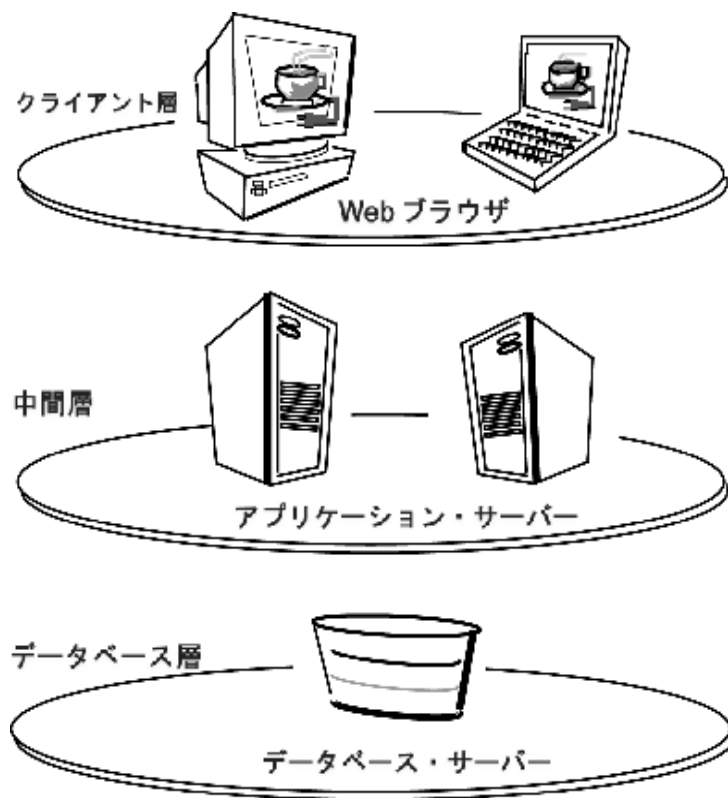
- [Forms Listener Servlet](#) を使用した Web への配布サポート
- OC4J との統合 (第 4 章「[HTTP Listener と OC4J での Oracle9iAS Forms Services の使用](#)」を参照)
- Single Sign-On (第 5 章「[SSO と OID での Oracle9iAS Forms Services の使用](#)」を参照)
- OID との統合 (第 5 章「[SSO と OID での Oracle9iAS Forms Services の使用](#)」を参照)
- 運用管理を容易にする Enterprise Manager との統合 (第 6 章「[Oracle Enterprise Manager と Oracle9i Forms](#)」を参照)
- トレースとロギングの改善 (第 7 章「[トレースと診断](#)」を参照)

Oracle9iAS Forms Services のアーキテクチャ

Oracle9iAS Forms Services では、3 層のアーキテクチャを使用してデータベース・アプリケーションを配布します。図 1-1 は、Oracle9iAS Forms Services アーキテクチャを構成する 3 つの層を示しています。

- クライアント層には、アプリケーションを表示する Web ブラウザが含まれます。
- 中間層は、アプリケーション・ロジックとサーバー・ソフトウェアが格納されるアプリケーション・サーバーです。
- データベース層は、企業データが格納されるデータベース・サーバーです。

図 1-1 Oracle9iAS Forms Services アーキテクチャ



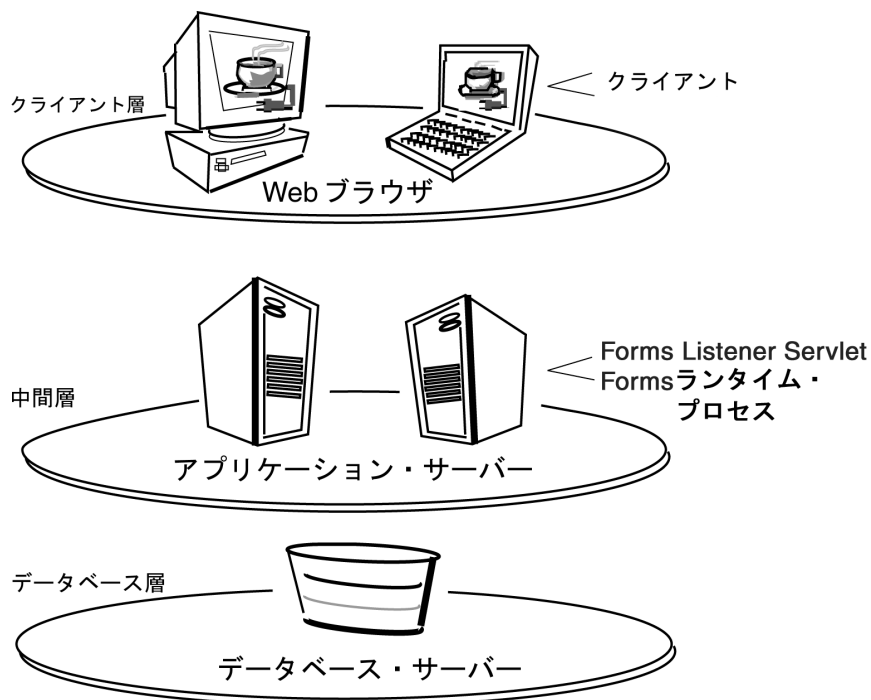
Oracle9iAS Forms Services のコンポーネント

Oracle9iAS Forms Services は、複雑なトランザクション型のフォーム・アプリケーションをインターネット上に配布する中間層のアプリケーション・フレームワークです。開発者は、Oracle9i Forms Developer を使用して新規のアプリケーションを作成し、Oracle9iAS Forms Services を使用してインターネットへ配布できます。また、開発者は従来のクライアント / サーバー型アプリケーションを、そのアプリケーション・コードを変更することなく 3 層のアーキテクチャに移行することもできます。

Oracle9iAS Forms Services は、次の図 1-2 に示す 3 つの主要コンポーネントで構成されます。

- Client (クライアント層に常駐)
- Forms Listener Servlet (中間層に常駐)
- Forms ランタイム・プロセス (中間層に常駐)

図 1-2 フォームを実行する 3 層構造



Forms Listener Servlet

Forms Listener Servlet は、Java クライアントと Forms ランタイム・プロセス間でブローカとしての役割を果たします。Forms Listener Servlet は、Java クライアント・プロセスから接続要求を受け取ると、Java クライアント・プロセスにかかわって Forms ランタイム・プロセスを開始します。

詳細は、「[Forms Listener Servlet](#)」を参照してください。

Forms ランタイム・プロセス

Forms ランタイム・プロセスは、アプリケーション・ロジックと処理を管理します。また Java クライアントのためにデータベース接続を保持します。クライアント / サーバー・モードで実行するのに使用されたのと同じフォーム、メニューおよびライブラリ・ファイルを使用します。

Forms ランタイム・プロセスは、次の 2 つの役割を果たします。クライアント・ブラウザと通信する場合は、クライアント・ブラウザからの要求を処理するサーバーとしての役割を果たし、メタデータをクライアントに送信してユーザー・インタフェースを伝えます。データベース・サーバーと通信する場合は、要求されたデータをデータベースに問い合わせるクライアントとしての役割を果たします。

Forms Listener Servlet

Oracle9iAS Forms Services は Forms Listener Servlet (Java サーブレット) を使用して、Forms ランタイム・プロセスの開始と終了、Forms ランタイム・プロセスとの通信を行います。Forms ランタイムは、特定の Oracle9i Forms アプリケーションに含まれているコードを実行します。Forms Listener Servlet は、クライアントごとに Forms ランタイム・プロセスの生成を管理し、クライアントとそれに対応する Forms ランタイム・プロセス間のネットワーク通信を管理します。Forms Listener Servlet は、Oracle9i Forms より前のリリースで提供されている Forms Listener にかわるものです。

注意： Forms Listener Servlet は、すでに Oracle9iAS のインストール・プロセスで設定されているため、構成する必要はありません。

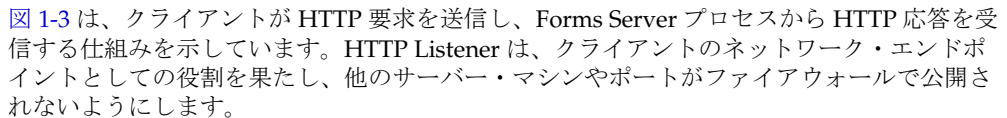
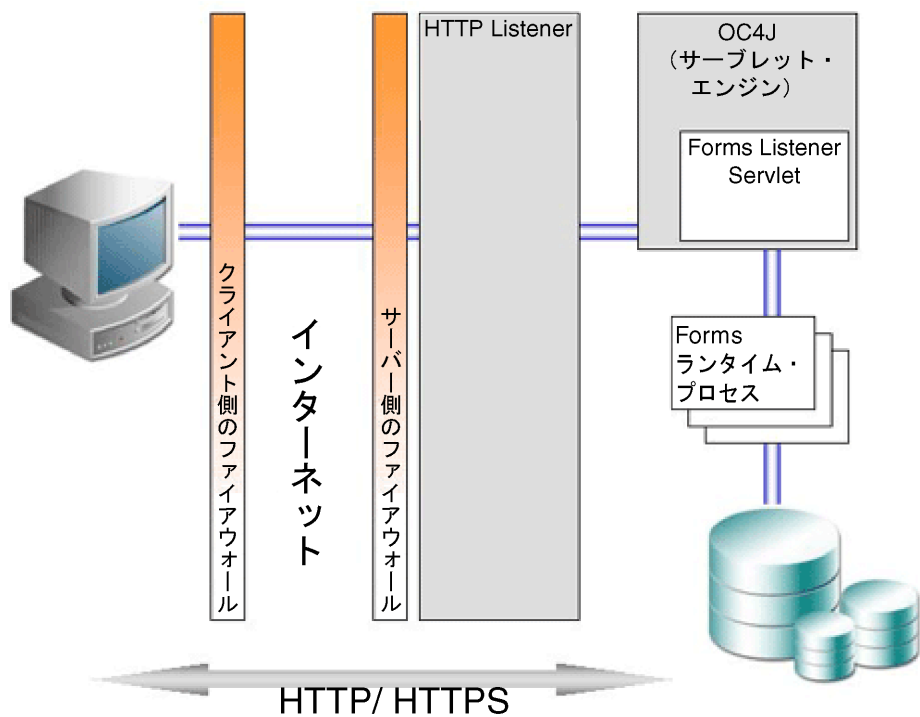
 図 1-3 は、クライアントが HTTP 要求を送信し、Forms Server プロセスから HTTP 応答を受信する仕組みを示しています。HTTP Listener は、クライアントのネットワーク・エンドポイントとしての役割を果たし、他のサーバー・マシンやポートがファイアウォールで公開されないようにします。

図 1-3 Forms Listener Servlet を使用したアーキテクチャ



Oracle9i Forms アプリケーションの配布の基本

概要

この章では、Oracle9iAS Forms Services を構成するのに必要な基本ファイル、Oracle9iAS Forms Services が Oracle9iAS で実行される仕組み、および Oracle9i Forms アプリケーションを配布する際の手順について説明します。インストールの完了後、この章に記載されている情報を使用して、初期構成の変更や必要な変更を行うことができます。

この章には、次の項が含まれています。

- 構成ファイル
- アプリケーションの配布
- Oracle9iAS Forms Services のアクション
- クライアント・ブラウザのサポート

構成ファイル

この項では、Oracle9i Forms アプリケーションを構成するのに必要な基本ファイルについて説明します。構成の詳細は、第 3 章「[Oracle9iAS Forms Services の構成](#)」を参照してください。

この項は、次の小項目に分かれています。

- [Oracle9i Forms の構成ファイル](#)
- [Oracle9iAS Containers for J2EE \(OC4J\) の構成ファイル](#)
- [Oracle HTTP Listener の構成ファイル](#)
- [標準フォントとアイコン・ファイル](#)
- [環境変数](#)

注意： ファイルが存在する場所は、ORACLE_HOME を基準とした相対パスで示しています。Windows では、フォワード・スラッシュを「¥」マークに置き換える必要があります。

Oracle9i Forms の構成ファイル

Oracle9i Forms の構成ファイルを使用すると、フォームのパラメータを指定できます。この項は、次の小項目に分かれています。

- [formsweb.cfg](#)
- [base.htm](#)、[basejini.htm](#)、[basejpi.htm](#) および [baseie.htm](#)
- [ftrace.cfg](#)

formsweb.cfg

場所 : *forms90/server*

これは、Forms Servlet の構成ファイルで、次の内容が含まれています。

- Forms アプレットおよびランタイムのコマンド・ライン・パラメータの値、使用する環境ファイルの名前（envFile 設定）。
- インストール時に設定するサーブレット構成パラメータのほとんどの設定。変更が必要な場合は、これらのパラメータをカスタマイズできます。

ベース HTML ファイル内の変数（%*variablename*%）は、formsweb.cfg ファイルに指定された適切なパラメータ値によって置換されます。URL 要求がある場合は、その問合せパラメータの値によって置換されます。

formsweb.cfg の詳細は、第 3 章の「[formsweb.cfg](#)」を参照してください。

base.htm、basejini.htm、basejpi.htm および baseie.htm

場所 : *forms90/server*

ベース HTML ファイル (base.htm、basejini.htm、basejpi.htm および baseie.htm) は、Oracle9i Forms アプリケーションの起動に使用する HTML ページを生成するとき、Forms Servlet によりテンプレートとして使用されます。

構成の変更は、formsweb.cfg ファイルで行い、ベース HTML ファイルを編集しないようにしてください。ベース HTML ファイルを変更する必要がある場合は、ユーザー自身のバージョンを作成し、適切な設定変更をして formsweb.cfg ファイルから参照します。

ベース HTML ファイルの詳細は、第 3 章の「[base.htm](#)、[basejini.htm](#)、[basejpi.htm](#) および [baseie.htm](#)」を参照してください。

fttrace.cfg

場所 : *forms90/trace*

このファイルを使用すると、Forms Trace を構成できます。Forms Trace は、Oracle 9i Forms より前のリリースで利用できた Forms Runtime Diagnostics (FRD) と Performance Event Collection Services (PECS) の機能にかわるものです。Forms Trace により、フォームの実行パス (たとえば、ユーザーがフォームの使用時に行った手順など) をトレースできます。

fttrace.cfg の詳細は、第 7 章の「[fttrace.cfg](#)」を参照してください。

Oracle9iAS Containers for J2EE (OC4J) の構成ファイル

デフォルトにおける Oracle9i Forms は、forms90app.ear と呼ばれる EAR (Enterprise Archive) ファイルにパッケージした J2EE 準拠アプリケーションとして配布することにより、OC4J 用に構成されています。EAR ファイルは、Oracle9iAS のインストール・プロセス時に配布されます (Oracle9i Forms の構成を選択した場合)。配布時には、EAR ファイルが OC4J インスタンスのアプリケーション・ディレクトリに解凍されます。

この項では、次の小項目について説明します。

- [web.xml](#)
- [Oracle9i Forms OC4J ファイルのディレクトリ構造](#)

web.xml

場所 : *j2ee/OC4J_BI_Forms/applications/forms90app/forms90web/WEB-INF/web.xml*

Oracle9i Forms がインストールされて、構成されると、web.xml ファイルは ORACLE_HOME の下のディレクトリ *j2ee/OC4J_BI_Forms/applications/forms90app/forms90web/WEB-INF* に格納されます。web.xml では、Forms Servlet と Forms Listener Servlet に、別名 "f90servlet" と "l90servlet" が定義されます。

web.xml の詳細は、第 3 章の「[web.xml](#)」を参照してください。

Oracle9i Forms OC4J ファイルのディレクトリ構造

Oracle9iAS のインストールおよび構成時に、Forms EAR ファイル (forms90app.ear) は OC4J_BI_Forms インスタンスに配布されます。その結果、次のようなディレクトリ構造になります。

+ 記号の付いた名前はディレクトリです。

j2ee/OC4J_BI_Forms/applications/forms90app

+META-INF

-application.xml (ear ファイルの構造を定義)

+forms90web

+WEB-INF

-web.xml (サーブレット・パラメータを含む Forms および Listener サブレットの定義)

+lib

-f90srv.jar (Forms Servlet および Listener Servlet のコードを含む)

j2ee/OC4J_BI_Forms/application-deployments/forms90app/forms90web

-orion-web.xml (iDS のみに使用する仮想ディレクトリ・マッピングおよびコンテキスト・パラメータ)

Oracle HTTP Listener の構成ファイル

この項では、Oracle9i Forms 用の Oracle HTTP Listener の構成に使用するファイルを説明します。

forms90.conf

場所: *forms90/server*

forms90.conf は、Oracle9i Forms 用の Oracle HTTP Listener の構成ファイルです。この構成ファイルは oracle_apache.conf に組み込まれた後、httpd.conf (マスター HTTP リスナーの構成ファイル) に組み込まれます。forms90.conf では、仮想ディレクトリ (別名) とサーブレット・マウント・ポイントを定義して、URL 要求を OC4J サブレット・エンジンで動作している Forms Servlets にマップします。

forms90.conf の詳細は、第 3 章の「[forms90.conf](#)」を参照してください。

標準フォントとアイコン・ファイル

この項では、Oracle9i Forms のフォントとアイコンの設定を構成するために使用するファイルについて説明します。

Registry.dat

場所 : *forms90/java/oracle/forms/registry*

このファイルを使用すると、Oracle9iAS Forms Services で使用するデフォルトのフォント、フォント・マッピングおよびアイコンを変更できます。

Registry.dat の詳細は、第 3 章の「[Registry.dat](#)」を参照してください。

環境変数

この項では、Forms ランタイム・プロセスの環境設定を含むデフォルトの環境変数について説明します。

カスタマイズ可能なすべての環境変数の一覧は、第 3 章の「[環境変数とレジストリ設定のカスタマイズ](#)」を参照してください。

default.env

場所 : *forms90/server*

このファイルには Forms ランタイムの環境設定が含まれており、formsweb.cfg ファイルと同じディレクトリに格納されています。UNIX では、default.env に PATH と LD_LIBRARY_PATH を挿入する必要があります。

default.env の詳細は、第 3 章の「[default.env](#)」を参照してください。

アプリケーションの配布

Oracle9i Forms Developer でアプリケーションを作成したら、そのアプリケーションを Web 上に配布できます。Oracle9i Forms は、指定された URL を通して Oracle9iAS のアプリケーションにアクセスします。次に、その URL で HTTP Listener にアクセスし、そこで Listener Servlet と通信します。Listener Servlet は、新しい Oracle9i Forms モジュールごとに ifweb90.exe を起動します。

Oracle9iAS Forms Services の実行状況の詳細は、「[Oracle9iAS Forms Services のアクション](#)」を参照してください。

アプリケーションの配布

インストーラで設定されたデフォルトのパラメータを使用して基本フォームを配布する手順は、次のとおりです。

1. Oracle9i Forms Developer でアプリケーションを作成し、保存します。

.fmb は、Oracle9i Forms Builder 以外ではオープンできない設計時ファイルです。.fmx は .fmb をコンパイルしたときに作成されるランタイム・ファイルで、Web への配布に使用します。

Oracle9i Forms Developer の詳細は、Oracle9i Forms Builder の「ヘルプ」メニューを参照してください。

2. Oracle9i Forms でユーザーのアプリケーション・モジュールにアクセスできるように、formsweb.cfg に構成セクションを作成します。構成セクションの作成は、formsweb.cfg の最後の空きスペースで行います。構成ファイルを作成する構文は、次のとおりです。

```
[my_application]
form=name_of_module
```

これは、次のように記述します。

```
[application]
form=hrapp.fmx
```

上の例では、Oracle9i Forms は "application" というアプリケーションを実行し、ファイル "hrapp.fmx" を呼び出します。

これは、ブラウザの URL に "...?config=application" と入力することにより、Web 上で Oracle9i Forms モジュール hrapp.fmx を呼び出しできることを意味します。

注意： 構成セクションはスペースが含まれていない限り、どのような名前でも付けられます。

3. .fmx ファイルの場所は、forms90_path 環境変数で指定します。たとえば、.fmx ファイルが /export/home/ias/forms90 にある場合、forms90_path に /export/home/ias/forms90 と記述します（複数の場所を記述する場合は、セミコロンで区切ります）。

forms90_path は、formsweb.cfg 内の envFile=<ファイル名> で参照される ASCII ファイルである、Oracle9i Forms 環境ファイルを使用して設定できる環境変数です。使用するデフォルトの環境ファイルは、forms90/Server ディレクトリにある default.env です。

ユーザー自身の環境ファイルを作成するには、default.env をコピーしてから、アプリケーションに必要な変更を行います。formsweb.cfg のアプリケーション構成セクションに、次のパラメータを追加します。

```
[application]
form=hrapp.fmx
envFile=< ファイル名 >.env
```

4. 次の URL に、アプリケーションの名前を入力します。

```
http://mymachine.com:7777/forms90/f90servlet?
```

"mymachine" はマシンの名前、"7777" は HTTP Listener で使用するポートです。

構成ファイルを指定したら、"config=" と構成セクションの名前を追加します。ステップ 2 の例の場合、hrapp.fmx にアクセスする URL は次のようになります。

```
http://mymachine.com:7777/forms90/f90servlet?config=application
```

パラメータの指定

Oracle9i Forms アプリケーションにパラメータ値を事前定義するには、3 つの方法があります。次の方法でパラメータを定義できます。

- **formsweb.cfg のデフォルトのセクションでアプリケーション設定を編集する。**

デフォルトの構成セクションでは、Oracle9i Forms で使用するデフォルト値を表示します。デフォルトの構成セクションには、システム・パラメータとユーザー・パラメータという 2 つの要素があります。既存のデフォルトの構成セクションは編集できますが、新しい構成セクションは作成できません。

たとえば、Microsoft Internet Explorer 5.x で Forms アプレットを実行する方法を指定するシステム・パラメータのデフォルト値は、次のように定義されています。

```
IE=JInitiator
```

Forms アプレットをブラウザのネイティブ JVM で実行する場合は、formsweb.cfg ファイルのパラメータを次のように編集します。

```
IE=native
```

名前を付けたアプリケーションの構成セクションでは、システム・パラメータとユーザー・パラメータの値を上書きできます（「[formsweb.cfg で構成セクションを作成する](#)」を参照）。たとえば、次のように指定します。

```
[myApp]
baseHTML=mybase.htm
baseHTMLjinitiator=mybasejini.htm
baseHTMLjpi=mybasejpi.htm
baseHTMLie=mybaseie.htm
form=myapp.fmx
```

userid=

注意： 名前を付けた構成セクション（formsweb.cfg 内）で指定したパラメータは、システム・パラメータ設定を上書きします。

アプリケーションで基本の HTML テンプレートに変更を行う必要がある場合や、Internet Explorer 仮想マシンに別の値を設定する必要がある場合は、システム・パラメータ設定を上書きします。システム・パラメータ設定の変更は、サーバーで実行するすべてのアプリケーションに変更が必要な場合にのみ行ってください。

注意： システム・パラメータは、ユーザー・パラメータとは異なり URL で上書きできません。

■ formsweb.cfg で構成セクションを作成する。

「[アプリケーションの配布](#)」のステップ 2 で作成した構成セクションで、Oracle9i Forms アプリケーションのパラメータを指定できます。formsweb.cfg にあるすべてのアプリケーションおよびシステム・パラメータを指定できます。

たとえば、次のように、アプリケーションのルック・アンド・フィールを Oracle のルック・アンド・フィールに設定することができます。

```
[Human Resources]
form=hrapp.fmx
envFile=< ファイル名 >.env
userid=scott/tiger@orcl
lookandfeel= oracle
```

また、名前を付けた構成セクションのデフォルトのパラメータ値にも上書きできます。たとえば、**scott/tiger@orcl** へのアプリケーション接続情報を事前定義するには、次のように、名前を付けた構成セクションに **userid** のパラメータ値を設定する必要があります。

```
[Human Resources]
form=hrapp.fmx
envFile=< ファイル名 >.env
userid=scott/tiger@orcl
```

編集可能な他のパラメータは、「[formsweb.cfg ファイル内のパラメータ](#)」を参照してください。

注意： 構成セクションで指定したパラメータは、アプリケーションのデフォルト設定を上書きします。

■ **Oracle9i Forms アプリケーションへのアクセスに使用する URL を編集する。**

Oracle9i Forms アプリケーションにアクセスする URL にパラメータを直接入力できます。前述の例の場合、構成ファイルに pageTitle パラメータを指定するかわりに、URL に次のように入力することもできます。

`http://mymachine.com:7777/forms90/f90servlet?config=hr&pageTitle="My Company"`

アンパサンド (&) を使用すれば、フォームと名前を付けた構成パラメータの組合せを呼び出すことができます。たとえば、次のように指定します。

`http://mymachine.com:7777/forms90/f90servlet?config=ienative&form=hrapp,`

これは、"ienative" で指定したパラメータ設定を使用して、フォーム "hrapp" を呼び出しています。

注意： URL で指定したパラメータは、構成セクションで設定したパラメータを上書きします。

Oracle9iAS Forms Services のアクション

この項では、Forms Servlet を使用して初期の HTML ページを生成することを前提に、Oracle9i Forms アプリケーションが Oracle9iAS で実行される仕組みと、構成ファイルが使用される仕組みを説明します。説明をわかりやすくするために、Web サーバーが "mymachine.com" と呼ばれるマシンのポート 7777 で実行されていると想定します。また、Oracle9iAS のインストール・プロセス時に作成した標準構成には、一切変更を行っていないことを前提にします。

ユーザーが Oracle9i Forms アプリケーションを実行すると、次の一連のイベントが発生します。

1. ユーザーは Web ブラウザを起動し、次のような URL を指定します。

`http://mymachine.com:7777/forms90/f90servlet?config=ienative&form=hrapp`

この場合、実行される（トップ・レベルの）フォーム・モジュールは "hrapp" で、使用する構成セクションは "ienative" です。

2. Oracle HTTP Listener は要求を受信します。Oracle HTTP Listener は、パス `"/forms90/f90servlet"` が `forms90.conf` ファイル (Forms Servlet) の OC4J マウント・ディレクティブの 1 つに一致したため、その要求を OC4J に転送します。
3. OC4J は、その要求を Oracle9i Forms アプリケーション (コンテキスト・ルートは `/forms90`) にマップします。次に、(`web.xml` ファイルで指定した `f90servlet` サブレットのマッピングを使用して) その要求を Forms Servlet にマップします。
4. Forms Servlet (OC4J で実行している) が、その要求を次のように処理します。
 - サブレットの構成ファイル (デフォルトでは `formsweb.cfg`) をオープンします。そのパラメータが設定されていない場合は、デフォルトの構成ファイル (`<ORACLE_HOME>/forms90/server/formsweb.cfg`) が使用されます。
 - `formsweb.cfg` ファイルで使用する構成セクションを決定します。URL には問合せパラメータ `"config=ienative"` が含まれているため、`[ienative]` セクションが使用されます。
 - 使用するベース HTML ファイルを決定します。このときの判断基準は、(a) 要求を発行しているブラウザ、(b) ブラウザを実行しているプラットフォーム、(c) `formsweb.cfg` ファイルの各種パラメータの設定 (`baseHTMLie`、`baseHTMLjinitiator`、`baseHTMLjpi`、`baseHTML` および `IE`) です。
 - ベース HTML ファイルを読み込み、次のように変数を置換した後、コンテンツを HTML ページとしてユーザーの Web ブラウザに送り返します。

Forms Servlet は変数 (`%myParam%` など) を検出すると、一致する URL 問合せパラメータ (`&myParam=xxx` など) を検索し、そのパラメータが見つからないときは `formsweb.cfg` ファイル内で一致するパラメータを検索します。一致するパラメータが見つければ、変数 (`%myParam%`) はそのパラメータの値と置換されます。

たとえば、ベース HTML ファイルにはテキスト `%form%` が含まれています。この場合、このテキストが値 `"hrapp"` と置換されます。

5. Forms Servlet がどのベース HTML ファイルを選択したかに応じて、Web ブラウザに送り返される HTML ページには、Forms アプレット (シン・クライアント) を起動する `Applet`、`Object` または `Embed` タグが含まれます。Forms アプレットは JVM (Web ブラウザのネイティブ JVM か、Oracle JInitiator や Sun の Java プラグインなどのプラグイン JVM のいずれか) で実行されます。
6. 選択したベース HTML ファイルがプラグイン用 (Oracle JInitiator か Sun の JDK Java プラグイン) であり、ユーザーがそのプラグインをまだマシンにインストールしていない場合は、そのプラグインをインストールするように指示されます。JInitiator の場合のダウンロード先は、仮想パス `/forms90/jinitiator` (`forms90.conf` ファイルで定義された仮想パス) です。
7. Forms アプレットを起動するには、その Java コードを最初にロードしておく必要があります。アプレットの場合は、アプレットのコードベースとアーカイブ・パラメータで指定されます。たとえば、ユーザーが Oracle JInitiator で実行している場合、そのアプ

レット・コードはファイル `http://mymachine.com:7777/forms90/java/f90all_jinit.jar` からロードされます。

`"/forms90/java"` の `forms90.conf` ファイルにおける仮想パスの定義により、アプレット・コードは Web サーバーからロードできます。

注意: Forms アプレット・コード (`f90all_jinit.jar` など) がネットワークにロードされるのは、ユーザーが Oracle9i Forms アプリケーションを最初に行ったとき (または、新しいバージョンの Oracle9i Forms が Web サーバーにインストールされたとき) のみです。それ以外は、ローカル・ディスク上の Web ブラウザ (または Java プラグイン) のキャッシュからロードされます。

8. Forms アプレットが実行されると、URL `http://mymachine.com:7777/forms90/l90servlet` にある Forms Listener Servlet に接続して Forms セッションを起動します。
9. Oracle HTTP Listener は要求を受信します。Oracle HTTP Listener は、パス `"/forms90/l90servlet"` が `forms90.conf` ファイル (Forms Listener Servlet) の OC4J マウント・ディレクティブの 1 つに一致したため、その要求を OC4J に転送します。
10. Forms Listener Servlet (`l90servlet`) が Forms セッションのための Forms ランタイム・プロセス (`ifweb90.exe` または `f90webm`) を起動します。
11. Forms アプレット (ユーザーの Web ブラウザで動作している) と Forms ランタイム・プロセス間の通信は、Forms セッションが終了するまで Listener Servlet を介して続きます。
12. コマンド・ライン (実行するフォーム名の指定など) が Forms ランタイム・プロセスに渡されます。ここでは、アプレット・パラメータ `"serverArgs"` として指定されます。ベース HTML ファイルの `serverArgs` 値の一部分は `%form%` であり、`"hrapp"` で置換されています。そのため、実際には、ランタイム・プロセスはファイル `"hrapp.fmx"` のフォームを実行します。

このファイルは、(`formsweb.cfg` ファイルで指定された) `workingDirectory` か、環境ファイル (デフォルトでは `default.env`) で定義された `FORMS90_PATH` 環境設定の指定ディレクトリの 1 つで提示される必要があります。また、`formsweb.cfg` ファイル (`form=myform` など) のディレクトリも指定できます。
13. Forms セッションは、次のいずれかが起きた場合に終了します。
 - トップ・レベルのフォームが終了した場合 (`"exit_form"` 組み込み関数を呼び出す PL/SQL トリガー・コードなどにより)。この場合、保存していない変更があれば、ユーザーはその変更を保存するように指示されます。`"exit_form(no_validate)"` は、プロンプトなしでフォームを終了します。
 - ユーザーが Web ブラウザを終了した場合 (この場合、保存されていない更新は失われます)。

クライアント・ブラウザのサポート

Oracle9i Forms アプリケーションは、Web 上で Oracle JInitiator プラグインを使用して表示できます (Netscape Navigator か Internet Explorer を使用します)。今後のパッチ・リリースでは、その他の仮想マシンもサポートされる予定です。

最新の対応プラットフォームなどのクライアント・ブラウザのサポートの詳細は、Oracle9i Forms Builder の「ヘルプ」メニューで「OTN 上の Forms」を選択してください。

Oracle JInitiator

Oracle JInitiator は、Web ブラウザ内で動作し、Sun の JDK/JRE 1.3 に基づいています。ブラウザの (ネイティブの) デフォルトの Java 仮想マシン (JVM) ではなく、クライアント上の特定の JVM を指定する機能が提供されます。Oracle JInitiator は、ブラウザによって提供されるデフォルトの JVM を置き換えたり、変更したりしません。Netscape Navigator のプラグイン形式や Internet Explorer の ActiveX コンポーネントとして代替の JVM を提供します。

オラクル社からは、2 つの JAR ファイル (f90all.jar および f90all_jinit.jar) が提供されます。これらはともに、ネットワークを介したクライアントへの効率的な配布を目的として、クラスをグループ化して zip 圧縮したものです。f90all_jinit.jar ファイルは高圧縮された JAR ファイルで、ダウンロード時のパフォーマンスを向上させるために Oracle JInitiator でのみ使用できます。これらのファイルがクライアントで使用されると、今後の使用のためにキャッシュされます。

Oracle JInitiator の詳細は、[付録 A 「JInitiator」](#) を参照してください。

構成パラメータとベース HTML ファイルをクライアント・ブラウザに連結する方法

ユーザーが Web 対応のアプリケーションを起動すると (アプリケーションの URL へのリンクをクリックすることで)、Forms Servlet は次の処理を実行します。

1. 使用しているブラウザを検出します。
2. formsweb.cfg ファイルを読み込み、Internet Explorer のパラメータ設定を調べ、ユーザーが Internet Explorer 5.5 以上を使用しているかどうかを判断します。

3. 次の表を使用して適切なベース HTML ファイルを選択します。

検出されたブラウザ	IE パラメータの設定	使用されるベース HTML ファイル
Internet Explorer 5.x または 6*	ネイティブ VM	baseie.htm
Internet Explorer 5.x または 6*	jinitiator	basejini.htm
Netscape Navigator またはバージョン 5 より前の Internet Explorer	適用不可	basejini.htm
その他のすべてのブラウザ	適用不可	base.htm

* Internet Explorer 5.5 からアップグレードした Internet Explorer 6 のみ（IE 6 はベース・リリースの動作確認がされていない）

** Microsoft Native VM により Windows で実行される Internet Explorer

4. ベース HTML ファイル内の変数（%variablename%）が、Forms Servlet .initArgs ファイルや formsweb.cfg ファイルに指定された適切なパラメータ値で置換され、URL 要求がある場合はその問合せパラメータの値で置換されます。
5. HTML ファイルをユーザーのブラウザに送信します。

Oracle9iAS Forms Services の構成

概要

この章には、次の項が含まれています。

- 構成ファイルのカスタマイズ
- 環境変数とレジストリ設定のカスタマイズ
- ユーザー独自のテンプレート HTML ファイルの作成
- Oracle9i Forms アプリケーションへのグラフィックの挿入
- Oracle9iAS Forms Services で使用するアイコンとイメージの配置
- Forms Listener Servlet での HTTPS の使用
- ユーザー ID/ パスワード非表示機能の使用
- 認証プロキシを使用した Oracle9i Forms アプリケーションの実行
- 言語検出の有効化

構成ファイルのカスタマイズ

インストール時に、次の構成ファイルがシステムにインストールされています。

- [formsweb.cfg](#)
- [base.htm](#)、[basejini.htm](#)、[basejpi.htm](#) および [baseie.htm](#)
- [web.xml](#)
- [forms90.conf](#)
- [Registry.dat](#)

ユーザーが最初に Oracle9iAS Forms アプリケーションを起動すると（アプリケーションの URL へのリンクをクリックすることで）、ベース HTML ファイルが Forms Servlet によって読み込まれます。ベース HTML ファイル内の変数（`%variablename%`）は、[formsweb.cfg](#) ファイルに指定された適切なパラメータ値によって置換され、URL 要求がある場合はその問合せパラメータの値によって置換されます。

変更が必要な場合は構成ファイルを変更できます。

formsweb.cfg

Forms Servlet の構成ファイル（[formsweb.cfg](#)）の説明とディレクトリは、第 2 章の「[formsweb.cfg](#)」を参照してください。

ファイルのネーミング・パラメータ

`envFile` パラメータで、使用する環境変数の設定が含まれる環境ファイルへの物理パスが指定されていることを確認します。

4 つのベース HTML パラメータも適切なファイルを示している必要があります。通常は、[formsweb.cfg](#) ファイルの冒頭のデフォルト・セクションに次のような行が表示されます。

```
baseHTML=base.htm
baseHTMLjinitiator=basejini.htm
baseHTMLjpi=basejpi.htm
baseHTMLie=baseie.htm
envFile=default.env
```

これらのすべてのパラメータでファイル名が指定されています。パスが指定されていない場合（例のように）、そのファイルは Forms Servlet の構成ファイル（[formsweb.cfg](#)）と同じディレクトリ（`<ORACLE_HOME>/forms90/server`）にあると想定されます。

formsweb.cfg で特別な名前を付けた構成の作成

名前を付けた固有の構成を [formsweb.cfg](#) ファイル内に作成できます。これらの構成は、フォームの実行に使用するエンド・ユーザーの URL 問合せ文字列から要求できます。

特別な構成を作成するには、大カッコ ([[]]) で囲んだ構成の名前を formsweb.cfg ファイルの最後に追加します。次に、その特別な構成にパラメータ（変更する必要があるもののみ）を指定します。

たとえば、フレームに分かれているブラウザ・ウィンドウで "generic" ルック・アンド・フィールを使用してフォームを実行するための構成を作成するには、formsweb.cfg ファイルに次のコマンドを追加します。

```
[sepwin]
forms=<module>
separateFrame=True
lookandfeel=Generic
```

エンド・ユーザーが "sepwin" 構成を使用するフォームを起動するには、次の URL を入力することになります。

```
http://server:port/forms90/f90servlet?config=sepwin
```

また、Enterprise Manager を使用しても名前を付けた構成セクションを作成できます（第 6 章「Oracle Enterprise Manager と Oracle9i Forms」を参照）。

特別な構成のその他の例は、「デフォルトの formsweb.cfg ファイル」を参照してください。

formsweb.cfg ファイル内のパラメータ

パラメータ	必須 / 任意	パラメータ値
システム・パラメータ これらのパラメータは、Forms Servlet の動作をコントロールします。パラメータはサーブレットの構成ファイル (formsweb.cfg) のみで指定でき、URL の問合せパラメータとしては指定できません。		
baseHTML	任意	APPLET タグを含む HTML ファイルへの物理パス。
baseHTMLjinitiator	必須	JInitiator のタグを含む HTML ファイルへの物理パス。
baseHTMLjpi	任意	Java Plug-in のタグを含む HTML ファイルへの物理パス。クライアント・ブラウザが Windows 上に存在せず、クライアント・ブラウザが Netscape または IE ネイティブ設定のない IE の場合にベース HTML ファイルとして使用されます。
baseHTMLie	任意	Internet Explorer 5 のタグ (CABBASE タグなど) を含む HTML ファイルへの物理パス。デフォルトのパスは <ORACLE_HOME>/forms90/server/baseie.htm です。 Internet Explorer のネイティブ JVM を使用する場合に必要です。

パラメータ	必須 / 任意	パラメータ値
HTML delimiter	必須	変数名のデリミタ。デフォルトで % になります。
workingDirectory	必須	設定されていない場合のデフォルトは <ORACLE_HOME>/forms90 です。
envFile	必須	デフォルトでは default.env に設定されます。
IE	Internet Explorer 5.0 以上のブラウザを使用するユーザーがいる場合に推奨	Microsoft Internet Explorer 5.0 以上で Forms アプレットの実行方法を指定します。クライアントが Internet Explorer 5.0 以上のブラウザを使用している場合は、ネイティブ JVM か JInitiator のいずれかを使用できます。"JInitiator" の設定では basejini.htm ファイルと JInitiator を使用します。"Native" の設定ではブラウザのネイティブ JVM を使用します。
Runform パラメータ (serverArgs パラメータ) 次のパラメータは、すべてベース HTML ファイルの変数 (%parameterName%) と一致します。これらの変数は、URL の問合せ文字列で指定されたパラメータ値か、そうでなければ formsweb.cfg ファイルで指定されたパラメータ値で置換されます。		
form	必須	実行するトップレベルの Forms モジュール (fmx ファイル) の名前を指定します。
userid	任意	ログイン文字列。scott/tiger@ORA8 など。
otherparams	任意	この設定は、form と userid 以外に Forms ランタイム・プロセスに渡すコマンド・ライン・パラメータを指定します。 デフォルトは次のとおりです。 debug=%debug% buffer_records=%buffer% debug_messages=%debug_messages% array=%array% query_only=%query_only% quiet=%quiet% render=%render% host=%host% port=%port% record=%record% tracegroup=%tracegroup% log=%log% term=%term%
HTML ページ・タイトル、BODY タグの属性、フォームの前後に追加する HTML セキュリティ上の理由から、これらは URL の問合せパラメータを使用して設定できません。		
pageTitle	任意	Oracle9iAS Forms Services。
HTMLbodyAttrs	任意	HTML ページの <BODY> タグの属性。
HTMLbeforeForm	任意	Forms アプリケーションが表示される領域の上にページを追加する HTML コンテンツ。
HTMLafterForm	任意	Forms アプリケーションが表示される領域の下にページを追加する HTML コンテンツ。

パラメータ	必須 / 任意	パラメータ値
アプレット・パラメータまたはオブジェクト・パラメータ 次のパラメータは、すべてオブジェクト・パラメータまたはアプレット・パラメータの値としてベース HTML ファイルに指定します。たとえば、次のように指定します。 <code><PARAM NAME="serverURL" VALUE="%serverURL%"></code>		
serverURL	必須	/forms90/190servlet (第1章の「 Forms Listener Servlet 」を参照)。
codebase	必須	物理ディレクトリ <ORACLE_HOME>/forms90/java を示すために定義した仮想ディレクトリ。デフォルトではこのディレクトリからアプレット JAR ファイルがダウンロードされます。 デフォルト値は /forms90/java です。
width	必須	フォーム・アプレットの幅をピクセルで指定します。デフォルトは 650 です。
height	必須	フォーム・アプレットの高さをピクセルで指定します。デフォルトは 500 です。
separateFrame	任意	アプレットを分割ウィンドウ内に表示するかどうかを指定します。 有効な値: TRUE または FALSE。
splashScreen	任意	アプレットが表示される前に表示する .GIF ファイルを指定します。スプラッシュなしの場合は「NO」に設定します。デフォルトのスプラッシュ・イメージを使用する場合は空白のままにします。 パラメータを設定するには、ファイル名 (myfile.gif など) または仮想パスとファイル名 (images/myfile.gif など) を記述します。
background	任意	背景に表示する .GIF ファイルを指定します。背景なしの場合は「NO」に設定します。デフォルトの背景を使用する場合は空白のままにします。
lookAndFeel	任意	アプリケーションのルック・アンド・フィールを指定します。有効な値: Oracle または Generic (Windows のルック・アンド・フィール)
colorScheme	任意	アプリケーションの配色を指定します。有効な値: Teal、Titanium、Red、Khaki、Blue、Olive または Purple 注意: lookAndFeel が Generic に設定されている場合、colorScheme は無視されます。

パラメータ	必須 / 任意	パラメータ値
serverApp	任意	<p>アプリケーションのファイル名がある場合に、デフォルトをその名前に置き換えます。アプリケーション固有のフォント・マッピングの作成およびアイコン・パスの設定には、アプリケーション・クラスを使用します。</p> <p>パラメータを設定するには、ファイル名（ファイルが <Oracle_Home>/forms90/java/oracle/forms/registry にある場合）または仮想パスとファイル名を記述します。</p>
archive	任意	<p>カンマで区切ったアーカイブ・ファイルのリスト。検出されたブラウザがネイティブ JVM を使用する Internet Explorer または JInitiator のいずれでもない場合に、アーカイブ・ファイルが使用されます（デフォルトは f90all.jar）。</p> <p>パラメータを設定するには、ファイル名（ファイルが codebase ディレクトリにある場合）または仮想パスとファイル名を記述します。</p>
archive_jinit	任意	<p>カンマで区切った JAR ファイルのリスト。検出されたブラウザが JInitiator の場合に JAR ファイルが使用されます（デフォルトは f90all_jinit.jar）。</p> <p>パラメータを設定するには、ファイル名（ファイルが codebase ディレクトリにある場合）または仮想パスとファイル名を記述します。</p>
archive_ie	任意	<p>カンマで区切った CAB ファイルのリスト。ネイティブ JVM を使用する Internet Explorer がブラウザとして検出された場合に、CAB ファイルが使用されます（デフォルトは f90all.cab）。</p>
networkRetries	任意	<p>負荷が高い場合やネットワーク障害時に、クライアントが目的のサーブレット・エンジンに要求を送信する試行回数（最大 10）を指定できます。デフォルトの設定は 0 です。この場合、1 回の試行後に Forms セッションが終了します。</p>
mapFonts	任意	<p><PARAM NAME = "mapFonts" VALUE = "yes"> でフォント・マッピングをトリガーします。</p> <p>JDK 1.3 のフォント・レンダリング・コードが一部変更されたため、JDK 1.1 で設定されたフォントが JDK 1.3 では大きくなっています。これにより表示上の問題が生じるため、フォントが JDK 1.1 と同じサイズになるように JDK 1.3 フォントをマップすることができます。</p>

パラメータ	必須 / 任意	パラメータ値
JInitiator のパラメータ		
jinit_download_page	必須 (Netscape のみ)	ユーザー自身のバージョンの Jinitiator ダウンロード・ページを作成する場合、そのページを示すように、このパラメータを設定します。デフォルトは、/forms90/jinitiator/us/jinit_download.htm です。
jinit_classid	必須 (IE のみ)	デフォルトは clsid:CAFECAFE-0013-0001-0008-ABCDEFABCDEF です。
jinit_exename	必須	デフォルトは jinit.exe#Version=1.3.1.8 です。
jinit_mimetype	必須 (Netscape のみ)	デフォルトは application/x-jinit-applet;version=1.3.1.8 です。
Sun の Java Plug-in のパラメータ		
jpi_codebase	必須	Sun の Java Plug-in のコードベース設定。
jpi_classid	必須	Sun の Java Plug-in のクラス ID。
jpi_download_page	必須	Sun の Java Plug-in のダウンロード・ページ。
Enterprise Manager の構成パラメータ		
em_mode	必須	1 (使用可能)。0 (使用不可)。 Enterprise Manager の HTML バージョンのみに適用されます。TRUE は、メトリックやサーブレットのステータスなどの Enterprise Manager 情報がすべて使用可能であることを示します。FALSE は、構成情報のみ使用可能であることを示します。
OID (Oracle Internet Directory) の構成パラメータ		
oid_formsid	必須	Oracle9iAS のインストール時に構成されるため、ユーザーがこのパラメータを変更する必要はありません。
oracle_home	必須	Oracle9iAS のインストール時に構成されるため、ユーザーがこのパラメータを変更する必要はありません。

デフォルトの formsweb.cfg ファイル

デフォルトの formsweb.cfg ファイルには次の内容が含まれます。

```
# formsweb.cfg - Forms Servlet default configuration file
# -----
# This file defines parameter values used by the FormsServlet (f90servlet)

# *****
# DEFAULT CONFIGURATIONS
# *****
#
# These are the default settings. Any of them may be overridden in the
# Named Configurations section. If they are not overridden, then the
# values here will be used.
# System Parameters cannot be overridden in the URL. User Parameters can.
#

#
# SYSTEM PARAMETERS
# -----
# These have fixed names and give information required by the Forms
# Servlet in order to function. They cannot be specified in the URL query
# string. But they can be overridden in a named configuration (see below).
# Some parameters specify file names: if the full path is not given,
# they are assumed to be in the same directory as this file. If a path
# is given, then it should be a physical path, not a URL.
#

baseHTML=base.htm
baseHTMLjinitiator=basejini.htm
baseHTMLjpi=basejpi.htm
baseHTMLie=baseie.htm

HTMLdelimiter=%

#
# WorkingDirectory defaults to <oracle_home>/forms90 if unset.
#
workingDirectory=
envFile=default.env

#
# The next parameter specifies how to execute the Forms applet under
# Microsoft Internet Explorer 5.x. Put IE=native if you want the
# Forms applet to run in the browser's native JVM.
#
IE=JInitiator
```

```
#
# USER PARAMETERS
# -----
# These match variables (e.g. %form%) in the baseHTML file. Their values
# may be overridden by specifying them in the URL query string
# (e.g. "http://myhost.mydomain.com/servlet/f90servlet?form=myform&width=700")
# or by overriding them in a specific, named configuration (see below)
#

#
# 1) Runform arguments:
#
form=test.fmx
userid=

#
# These settings support running and debugging a form from the Builder:
#
otherparams=debug=%debug% buffer_records=%buffer% debug_messages=%debug_messages%
array=%array% query_only=%query_only% quiet=%quiet% render=%render% host=%host%
port=%port% record=%record% tracegroup=%tracegroup% log=%log% term=%term%
debug=no
buffer=no
debug_messages=no
array=no
query_only=no
quiet=yes
render=no
host=
port=
record=
tracegroup=
log=
term=

#
# 2) HTML page title, attributes for the BODY tag, and HTML to add before and
#    after the form:
#
pageTitle=Oracle9iAS Forms Services
HTMLbodyAttrs=
HTMLbeforeForm=
HTMLafterForm=

#
# 3) Values for the Forms applet parameters:
```

```
#
serverURL=/forms90/190servlet
codebase=/forms90/java
imageBase=DocumentBase
width=650
height=500
separateFrame=false
splashScreen=
background=
lookAndFeel=Oracle
colorScheme=teal
logo=
formsMessageListener=
recordFileName=

serverApp=default

#
# The following archive settings are for
#   archive_jini - settings for JInitiator
#   archive_ie   - settings for IE native JVM
#   archive      - settings for all other cases (Java Plugin, Appletviewer, etc)
#
archive_jini=f90all_jinit.jar
archive_ie=f90all.cab
archive=f90all.jar

#
# Number of times client should retry if a network failure occurs. Only
# change after having read the documentation.
#
networkRetries=0

#
# 4) Parameters for JInitiator (used with Windows clients)
#

#
# Page displayed to Netscape users to allow them to download JInitiator.
# If you create your own page, you should set this parameter to point to it.
#
jinit_download_page=/forms90/jinitiator/us/jinit_download.htm

#
# Parameters related to the version of JInitiator.
#
jinit_classid=clsid:CAFECAFE-0013-0001-0008-ABCDEFABCDEF
```

```

jinit_exename=jinit.exe#Version=1,3,1,8
jinit_mimetype=application/x-jinit-applet;version=1.3.1.8

#
# 5) Parameters for the Java Plugin (used with non-Windows clients)
#

#
# Page displayed to users to allow them to download the JPI
# (NOTE: you should check this page and possibly change the settings)
#
jpi_download_page=http://java.sun.com/products/plugin/1.3/plugin-install.html

#
# Parameters related to the version of the Java Plugin
#
jpi_classid=clsid:8AD9C840-044E-11D1-B3E9-00805F499D93
jpi_
codebase=http://java.sun.com/products/plugin/1.3/jinstall-13-win32.cab#Version=1,3,0
,0
jpi_mimetype=application/x-java-applet;version=1.3

#
# 6) EM config parameter
# Set this to "1" to enable Enterprise Manager to track Forms processes
#
em_mode=0

#
# 6) OID Config parameters (for Single Sign-On)
#
oid_formsid=%OID_FORMSID%
oracle_home=%ORACLE_HOME%

# *****
# NAMED CONFIGURATIONS
# *****
#
# You may define your own specific, named configurations (sets of parameters)
# by adding special sections as illustrated in the following examples.
# Note that you need only specify the parameters you want to change. The
# default values (defined above) will be used for all other parameters.
# Use of a specific configuration can be requested by including the text
# "config=<your_config_name>" in the query string of the URL used to run
# a form. For example, to use the sepwin configuration, you could issue
# a URL like "http://myhost.mydomain.com/servlet/f90servlet?config=sepwin".
#

```

```
#
# Example 1: configuration to run forms in a separate browser window with
#           "generic" look and feel (include "config=sepwin" in the URL)
#
[sepwin]
separateFrame=True
lookandfeel=Generic

#
# Example 2: configuration affecting users of MicroSoft Internet Explorer 5.x.
#           Forms applet will run under the browser's native JVM rather than
#           using Oracle JInitiator.
#
[ienative]
IE=native

#
# Example 3: configuration forcing use of the Java Plugin in all cases
#           (even if the client browser is on Windows)
#
[jpi]
baseHTMLJInitiator=basejpi.htm
baseHTMLie=basejpi.htm

#
# Example 4: configuration running the Forms ListenerServlet in debug mode
#           (debug messages will be written to the servlet engine's log file)
#
[debug]
serverURL=/servlet/190servlet/debug
```

base.htm、basejini.htm、basejpi.htm および baseie.htm

base.htm、basejini.htm、basejpi.htm および baseie.htm の概説とディレクトリは、第 2 章の「[base.htm](#)、[basejini.htm](#)、[basejpi.htm](#) および [baseie.htm](#)」を参照してください。

Oracle9iAS のインストールおよび構成時に、Oracle Universal Installer により 4 つのベース HTML ファイルがユーザーのシステムに作成されます。**ほとんどの場合、これらのファイルの変更は必要ありません。**これらのファイルを変更する必要がある場合は、ユーザー自身のバージョンを作成し、これを formsweb.cfg ファイルから参照する必要があります。デフォルトのファイルは、パッチのインストールにより上書きされます。

ユーザーが最初に Oracle9i Forms アプリケーションを起動すると（アプリケーションの URL へのリンクをクリックすること）、ベース HTML ファイルが Forms Servlet によって読み込まれます。

ベース HTML ファイル内の変数 (%*variablename*%) は、「[formsweb.cfg](#)」での説明のように、[formsweb.cfg](#) ファイルに指定された適切なパラメータ値によって置換され、URL 要求がある場合はその問合せパラメータの値によって置換されます。問合せパラメータの値は、[formsweb.cfg](#) ファイルの値を上書きします。

次に、ベース HTML ファイルがユーザーの Web ブラウザにダウンロードされます。

注意： ベース HTML 変数を変更するには、[formsweb.cfg](#) ファイル内の対応するパラメータ値を変更します。

次のベース HTML 初期ファイルは、<ORACLE_HOME>/forms90/server ディレクトリで利用できます。

- **basejini.htm:** これは、Oracle JInitiator を使用する Forms アプレットの実行に必要なタグが含まれるベース HTML ファイルです。オラクル社によってこの方法での動作が確認されたブラウザ（および標準の APPLET タグを使用して動作しないブラウザ）に適しています（Windows プラットフォームのみ）。例は、「[デフォルトの basejini.htm ファイル](#)」を参照してください。
- **basejpi.htm:** これは Java Plug-in 用のベース HTML ファイルです。Forms Servlet がこのファイルを使用するのは、クライアント・ブラウザが Windows 上に存在せず、クライアント・ブラウザが Netscape か IE ネイティブ設定のない IE のどちらかの場合です。
- **base.htm:** これは、AppletViewer または（ネイティブ JVM が Oracle9i Forms で動作することがオラクル社によって確認済の）任意の Web ブラウザで Forms アプレットを実行するために必要な APPLET タグが含まれている、ベース HTML ファイルです。例は、「[デフォルトの base.htm ファイル](#)」を参照してください。
- **baseie.htm:** これは、Internet Explorer 5 でネイティブ JVM を使用するために必要な Internet Explorer 5 のタグが含まれている、ベース HTML ファイルです。例は、「[デフォルトの baseie.htm ファイル](#)」を参照してください。

新しいベース HTML ファイルを作成するには：

1. 新しいベース HTML ファイルを任意のディレクトリ内に置きます。[formsweb.cfg](#) ファイル内にベース HTML ファイルの完全な物理パスが含まれるように、[basejini.htm](#)、[baseie.htm](#)、[basejpi.htm](#) または [base.htm](#) のパラメータを更新します。
2. <ORACLE_HOME>/forms90/server ディレクトリにある、[basejini.htm](#)、[baseie.htm](#)、[basejpi.htm](#) または [base.htm](#) の初期ファイルをコピーします。
3. ファイル名を変更します（たとえば、[order.htm](#)）。
4. ユーザーに表示されるテキストを追加または変更します（たとえば、<TITLE> タグおよび <BODY> タグ内のテキスト）。

5. 必要に応じてパラメータを変更します。ベース HTML ファイルの変数を使用し、「formsweb.cfg」での説明のように、formsweb.cfg ファイルの実際の値を指定してください。

baseHTML と baseHTMLjinitiator のタグも固有の名前の付いた構成セクションで設定し、システムのデフォルト値に上書きできます。これは、個別のカスタムベース HTML テンプレートを使用する必要がある場合に行います。ただし、カスタム・テンプレートをすべてのアプリケーションに使用する場合は、formsweb.cfg ファイルのデフォルトの構成セクションを変更するようにお勧めします。

ベース HTML ファイル内のパラメータと変数

base.htm または basejini.htm ファイルで提供されるパラメータ・タグを使用しない場合は、ファイルから削除してください。

残りのパラメータ値を変数 (%variablename%) としてベース HTML ファイルで指定することをお勧めします。たとえば、次のように指定します。

```
<PARAM NAME="logo" VALUE="%logo%">
```

次に、実際のパラメータ値を「formsweb.cfg ファイル内のパラメータ」で定義される formsweb.cfg ファイルに指定します。すべての変数が実行時に適切なパラメータ値で置き換えられます。

使用方法

- 変数の値は、ベース HTML ファイル内のどこでも使用できます。変数は特別のデリミタで囲まれた名前として指定されます（デフォルトのデリミタは %）。たとえば、HTML ファイルに次の行を置くことができます。

```
ARCHIVE="%Archive%"
```

次に値を formsweb.cfg ファイルまたは URL の問合せ文字列内の %Archive% に割り当てる必要があります。

- すべての変数は実行時に値を受け取る必要があります。変数が値を受け取らないと、Oracle9iAS Forms Services がユーザーの Web ブラウザに渡す適切な HTML ファイルを構築できないため、エラーが発生します。
- パフォーマンスを向上するには、JAR ファイルのダウンロード用のソースとして 1 つの Web サーバーのみを使用してください。同じファイルを異なるサーバーから複数回ダウンロードすることを防げます。

デフォルトの base.htm ファイル

```

<HTML>
<!-- FILE: base.htm (Oracle Forms) -->
<!--
<!-- This is the default baseHTML file for running a form on the -->
<!-- web using a generic APPLET tag to include Forms applet. -->
<!--
<!-- IMPORTANT NOTES: -->
<!-- Default values for all the variables which appear below -->
<!-- (enclosed in percent characters) are defined in the servlet -->
<!-- configuration file (formsweb.cfg). It is preferable to make -->
<!-- changes in that file where possible, rather than this one. -->
<!--
<!-- This file will be REPLACED if you reinstall Oracle Forms, so -->
<!-- you are advised to make your own version if you want to make -->
<!-- want to make any modifications. You should then set the -->
<!-- baseHTML parameter in the Forms Servlet configuration file -->
<!-- (formsweb.cfg) to point to your new file instead of this one. -->

<HEAD><TITLE>%pageTitle%</TITLE></HEAD>

<BODY %HTMLbodyAttrs%
%HTMLbeforeForm%

<!-- Forms applet definition (start) -->
<APPLET CODEBASE="%codebase%"
        CODE="oracle.forms.engine.Main"
        ARCHIVE="%archive%"
        WIDTH="%Width%"
        HEIGHT="%Height%">

<PARAM NAME="serverURL" VALUE="%serverURL%">
<PARAM NAME="networkRetries" VALUE="%networkRetries%">
<PARAM NAME="serverArgs"
        VALUE="module=%form% userid=%userid% sso_userid=%sso_userid% %otherParams%">
<PARAM NAME="separateFrame" VALUE="%separateFrame%">
<PARAM NAME="splashScreen" VALUE="%splashScreen%">
<PARAM NAME="background" VALUE="%background%">
<PARAM NAME="lookAndFeel" VALUE="%lookAndFeel%">
<PARAM NAME="colorScheme" VALUE="%colorScheme%">
<PARAM NAME="serverApp" VALUE="%serverApp%">
<PARAM NAME="logo" VALUE="%logo%">
<PARAM NAME="imageBase" VALUE="%imageBase%">
<PARAM NAME="formsMessageListener" VALUE="%formsMessageListener%">
<PARAM NAME="recordFileName" VALUE="%recordFileName%">

</APPLET>

```

```
<!-- Forms applet definition (end) -->
```

```
%HTMLafterForm%
```

```
</BODY>
```

```
</HTML>
```

デフォルトの basejini.htm ファイル

```
<HTML>
<!-- FILE: basejini.htm (Oracle Forms) -->
<!-- -->
<!-- This is the default baseHTML file for running a form on the -->
<!-- web using JInitiator-style tags to include the Forms applet. -->
<!-- -->
<!-- IMPORTANT NOTES: -->
<!-- Default values for all the variables which appear below -->
<!-- (enclosed in percent characters) are defined in the servlet -->
<!-- configuration file (formsweb.cfg). It is preferable to make -->
<!-- changes in that file where possible, rather than this one. -->
<!-- -->
<!-- This file will be REPLACED if you reinstall Oracle Forms, so -->
<!-- you are advised to make your own version if you want to make -->
<!-- want to make any modifications. You should then set the -->
<!-- baseHTMLJinitiator parameter in the Forms Servlet configuration -->
<!-- file (formsweb.cfg) to point to your new file instead of this. -->

<HEAD><TITLE>%pageTitle%</TITLE></HEAD>

<BODY %HTMLbodyAttrs%>
%HTMLbeforeForm%

<!-- Forms applet definition (start) -->
<OBJECT classid="%jinit_classid%"
        codebase="/forms90/jinitiator/%jinit_exename%"
        WIDTH="%Width%"
        HEIGHT="%Height%"
        HSPACE="0"
        VSPACE="0">
<PARAM NAME="TYPE" VALUE="%jinit_mimetype%">
<PARAM NAME="CODEBASE" VALUE="%codebase%">
<PARAM NAME="CODE" VALUE="oracle.forms.engine.Main" >
<PARAM NAME="ARCHIVE" VALUE="%archive_jini%" >

<PARAM NAME="serverURL" VALUE="%serverURL%">
<PARAM NAME="networkRetries" VALUE="%networkRetries%">
```

```

<PARAM NAME="serverArgs"
      VALUE="module=%form% userid=%userid% sso_userid=%sso_userid% %otherParams%">
<PARAM NAME="separateFrame" VALUE="%separateFrame%">
<PARAM NAME="splashScreen" VALUE="%splashScreen%">
<PARAM NAME="background" VALUE="%background%">
<PARAM NAME="lookAndFeel" VALUE="%lookAndFeel%">
<PARAM NAME="colorScheme" VALUE="%colorScheme%">
<PARAM NAME="serverApp" VALUE="%serverApp%">
<PARAM NAME="logo" VALUE="%logo%">
<PARAM NAME="imageBase" VALUE="%imageBase%">
<PARAM NAME="formsMessageListener" VALUE="%formsMessageListener%">
<PARAM NAME="recordFileName" VALUE="%recordFileName%">
<COMMENT>
<EMBED SRC="" PLUGINSOURCE="%jinit_download_page%"
      TYPE="%jinit_mimetype%"
      java_codebase="%codebase%"
      java_code="oracle.forms.engine.Main"
      java_archive="%archive_jini%"
      WIDTH="%Width%"
      HEIGHT="%Height%"
      HSPACE="0"
      VSPACE="0"

      serverURL="%serverURL%"
      networkRetries="%networkRetries%"
      serverArgs="module=%form% userid=%userid% sso_userid=%sso_userid%
%otherParams%"
      separateFrame="%separateFrame%"
      splashScreen="%splashScreen%"
      background="%background%"
      lookAndFeel="%lookAndFeel%"
      colorScheme="%colorScheme%"
      serverApp="%serverApp%"
      logo="%logo%"
      imageBase="%imageBase%"
      formsMessageListener="%formsMessageListener%"
      recordFileName="%recordFileName%"
>
</EMBED>
</COMMENT>
</NOEMBED></EMBED>
</OBJECT>
<!-- Forms applet definition (end) -->

%HTMLafterForm%

</BODY>

```

```
</HTML>
```

デフォルトの basejpi.htm ファイル

```
<HTML>
<!-- FILE: basejpi.htm (Oracle Forms) -->
<!--
<!-- This is the default baseHTML file for running a form on the -->
<!-- web using the JDK Java Plugin. This is used for example when -->
<!-- running with Netscape on Unix. -->
<!-- -->
<!-- IMPORTANT NOTES: -->
<!-- Default values for all the variables which appear below -->
<!-- (enclosed in percent characters) are defined in the servlet -->
<!-- configuration file (formsweb.cfg). It is preferable to make -->
<!-- changes in that file where possible, rather than this one. -->
<!-- -->
<!-- This file will be REPLACED if you reinstall Oracle Forms, so -->
<!-- you are advised to create your own version if you want to make -->
<!-- any modifications. You should then set the baseHTMLjpi -->
<!-- parameter in the Forms Servlet configuration file (formsweb.cfg) -->
<!-- to point to your new file instead of this one. -->

<HEAD><TITLE>%pageTitle%</TITLE></HEAD>

<BODY %HTMLbodyAttrs%>
%HTMLbeforeForm%

<!-- Forms applet definition (start) -->
<OBJECT classid="%jpi_classid%"
        codebase="%jpi_codebase%"
        WIDTH="%Width%"
        HEIGHT="%Height%"
        HSPACE="0"
        VSPACE="0">
<PARAM NAME="TYPE"          VALUE="%jpi_mimetype%">
<PARAM NAME="CODEBASE"      VALUE="%codebase%">
<PARAM NAME="CODE"          VALUE="oracle.forms.engine.Main" >
<PARAM NAME="ARCHIVE"       VALUE="%archive%" >

<PARAM NAME="serverURL"     VALUE="%serverURL%">
<PARAM NAME="networkRetries" VALUE="%networkRetries%">
<PARAM NAME="serverArgs"
        VALUE="module=%form% userid=%userid% sso_userid=%sso_userid% %otherParams%">
<PARAM NAME="separateFrame" VALUE="%separateFrame%">
<PARAM NAME="splashScreen"  VALUE="%splashScreen%">
<PARAM NAME="background"    VALUE="%background%">
```

```

<PARAM NAME="lookAndFeel" VALUE="%lookAndFeel%">
<PARAM NAME="colorScheme" VALUE="%colorScheme%">
<PARAM NAME="serverApp" VALUE="%serverApp%">
<PARAM NAME="logo" VALUE="%logo%">
<PARAM NAME="imageBase" VALUE="%imageBase%">
<PARAM NAME="formsMessageListener" VALUE="%formsMessageListener%">
<PARAM NAME="recordFileName" VALUE="%recordFileName%">
<COMMENT>
<EMBED SRC="" PLUGINSOURCE="%jpi_download_page%"
        TYPE="%jpi_mimetype%"
        java_codebase="%codebase%"
        java_code="oracle.forms.engine.Main"
        java_archive="%archive%"
        WIDTH="%Width%"
        HEIGHT="%Height%"
        HSPACE="0"
        VSPACE="0"

        serverURL="%serverURL%"
        networkRetries="%networkRetries%"
        serverArgs="module=%form% userid=%userid% sso_userid=%sso_userid%
%otherparams%"
        separateFrame="%separateFrame%"
        splashScreen="%splashScreen%"
        background="%background%"
        lookAndFeel="%lookAndFeel%"
        colorScheme="%colorScheme%"
        serverApp="%serverApp%"
        logo="%logo%"
        imageBase="%imageBase%"
        recordFileName="%recordFileName%"
>
<NOEMBED>
</COMMENT>
</NOEMBED></EMBED>
</OBJECT>
<!-- Forms applet definition (end) -->

%HTMLafterForm%

</BODY>
</HTML>

```

デフォルトの baseie.htm ファイル

```
<HTML>
<!-- FILE: baseie.htm (Oracle Forms) -->
<!-- -->
<!-- This is the default baseHTML file for running a form on the -->
<!-- web with Internet Explorer version 5.0 or above, using that -->
<!-- browser's native JVM. -->
<!-- -->
<!-- IMPORTANT NOTES: -->
<!-- Default values for all the variables which appear below -->
<!-- (enclosed in percent characters) are defined in the servlet -->
<!-- configuration file (formsweb.cfg). It is preferable to make -->
<!-- changes in that file where possible, rather than this one. -->
<!-- -->
<!-- This file will be REPLACED if you reinstall Oracle Forms, so -->
<!-- you are advised to make your own version if you want to make -->
<!-- want to make any modifications. You should then set the -->
<!-- baseHTMLie parameter in the Forms Servlet configuration file -->
<!-- (formsweb.cfg) to point to your new file instead of this one. -->

<HEAD><TITLE>%pageTitle%</TITLE></HEAD>

<BODY %HTMLbodyAttrs%>
%HTMLbeforeForm%

<!-- Forms applet definition (start) -->
<APPLET CODEBASE="%codebase%"
        CODE="oracle.forms.engine.Main"
        WIDTH="%Width%"
        HEIGHT="%Height%">

<PARAM NAME="cabbage" VALUE="%archive_ie%">
<PARAM NAME="serverURL" VALUE="%serverURL%">
<PARAM NAME="networkRetries" VALUE="%networkRetries%">
<PARAM NAME="serverArgs"
        VALUE="module=%form% userid=%userid% sso_userid=%sso_userid% %otherParams%">
<PARAM NAME="separateFrame" VALUE="%separateFrame%">
<PARAM NAME="splashScreen" VALUE="%splashScreen%">
<PARAM NAME="background" VALUE="%background%">
<PARAM NAME="lookAndFeel" VALUE="%lookAndFeel%">
<PARAM NAME="colorScheme" VALUE="%colorScheme%">
<PARAM NAME="serverApp" VALUE="%serverApp%">
<PARAM NAME="logo" VALUE="%logo%">
<PARAM NAME="imageBase" VALUE="%imageBase%">
<PARAM NAME="formsMessageListener" VALUE="%formsMessageListener%">
<PARAM NAME="recordFileName" VALUE="%recordFileName%">
```



```

</APPLET>
<!-- Forms applet definition (end) -->

%HTMLafterForm%

</BODY>
</HTML>

```

web.xml

web.xml の説明とディレクトリは、第 2 章の「[web.xml](#)」を参照してください。

上級ユーザーは、web.xml ファイルを編集して次のことを実行できます。

- 別のテスト・オプションを使用可能にする。
 Oracle9iDS または Oracle9iAS をインストールしても Oracle9i Forms を正しく実行できない場合、通常はセキュリティ上の理由から使用可能にされていないテスト・オプションを使用可能にすると、実行可能になる場合があります。このオプションを使用するには、web.xml ファイルを編集して f90servlet の testMode パラメータを TRUE に設定します。次に Web サーバー（または OC4J）を再起動します。追加のオプションが「Forms サブレット管理」ページ (http://<your_web_server_hostname>:<port>/forms90/f90servlet/admin など) の URL でアクセス可能) に表示されます。
- 標準 (<ORACLE_HOME>/forms90/server/formsweb.cfg) 以外の Forms Servlet 構成ファイルを使用する。
 f90servlet の "configFileName" サブレット・パラメータをアンコメントして変更します。
- (Forms Servlet ではなく) 静的 HTML ページを使用して Oracle9i Forms を実行する。
 Oracle9i Forms アプリケーションを Forms Servlet 以外の方法（静的 HTML ページや JSP など）で実行している場合、formsweb.cfg ファイルのパラメータ設定は使用されません。そのため、Listener Servlet に workingDirectory や envFile などのサブレット・パラメータを定義する必要があります（Forms ランタイム・プロセスの現在の作業ディレクトリや、使用する環境設定が含まれたファイルを指定します）。

デフォルトの web.xml ファイル

```

- <web-app>
  <display-name>Forms 9i Services</display-name>
  <description>Oracle 9iAS: Forms 9i Services</description>
- <welcome-file-list>
  <welcome-file>l90servlet</welcome-file>
</welcome-file-list>
- <!-- Forms 9i page generator servlet
  -->
- <servlet>

```

```
<servlet-name>f90servlet</servlet-name>
<servlet-class>oracle.forms.servlet.FormsServlet</servlet-class>
- <!--
  During product installation the configFileName parameter is
    specified in the orion-web.xml file as a context parameter
    override (in iDS), or as a Java system property (in iAS).
    It is set to <oracle_home>/forms90/server/formsweb.cfg.
    You can override that value here by editing and uncommenting the
    following servlet parameter setting:

-->
- <!--
    <init-param>
      <param-name>configFileName</param-name>
      <param-value><your configuration file name goes here></param-value>
    </init-param>

-->
- <init-param>
- <!--
  Turn on or off sensitive options on the f90servlet/admin page.
    For security reasons this should be set to false for
    production sites.

-->
<param-name>testMode</param-name>
<param-value>false</param-value>
</init-param>
</servlet>
- <!-- Forms 9i listener servlet
-->
- <servlet>
  <servlet-name>l90servlet</servlet-name>
  <servlet-class>oracle.forms.servlet.ListenerServlet</servlet-class>
</servlet>
- <!--
  Forms 9i servlet mappings. Allow these paths to the servlets:
    /forms90/f90servlet or /forms90/f90servlet/*: FormsServlet
    /forms90/l90servlet or /forms90/l90servlet/*: ListenerServlet

-->
- <servlet-mapping>
  <servlet-name>f90servlet</servlet-name>
  <url-pattern>/f90servlet*</url-pattern>
```

```

    </servlet-mapping>
-   <servlet-mapping>
        <servlet-name>l90servlet</servlet-name>
        <url-pattern>/l90servlet*</url-pattern>
    </servlet-mapping>
-   <!--
        The following context parameter is only defined here so it can be
        overridden by the (site-specific) value in the orion-web.xml file.

-->
-   <context-param>
        <param-name>configFileName</param-name>
        <param-value></param-value>
    </context-param>
</web-app>
```

forms90.conf

forms90.conf の説明とディレクトリは、第 2 章の「[forms90.conf](#)」を参照してください。
次の表は、仮想パスとサーブレットのマッピングを示しています。

表 3-1 forms90.conf の仮想パスとサーブレットのマッピング

URL パス	タイプ	マッピング先	説明
/forms90/java	別名	<ORACLE_HOME>/forms90/java	Forms アプレットのコードベース。アプレット・コードをユーザーの Web ブラウザにダウンロードします。
/forms90/html	別名	<ORACLE_HOME>/tools/web90/html	runform.htm へのアクセス (テスト用にフォームを実行します)。
/forms90/jinitiator	別名	<ORACLE_HOME>/jinit	Oracle JInitiator のダウンロード。
/forms90/f90servlet	Servlet マウント・ポイント	Forms Servlet	フォームを実行する HTML ページを作成します。
/forms90/l90servlet	Servlet マウント・ポイント	Forms Listener Servlet	Forms アプレットからのメッセージ通信量を処理します。

デフォルトの forms90.conf

```
# $Id: forms90.conf,v 1.8 2002/03/01 01:11:53 pkuhn Exp $
# Name
#   forms90.conf
# Purpose
#   Apache mod_oc4j and mod_jserv configuration file for Forms 9i Services.
#   This file should be included into the Oracle Apache HTTP Listener
#   configuration file (typically by adding an include statement to the
#   oracle_apache.conf file)
# Remarks
#   If Forms is to be used with JServ, the jserv.properties file needs editing
#   to add the "forms90" servlet zone with properties file forms90.properties
# Notes
#   Virtual paths: We use AliasMatch when defining virtual paths for
#   security reasons (prevents directory browsing).

# Virtual path mapping for Forms Java jar and class files (codebase)
AliasMatch ^/forms90/java/(.*) "%FORMS_ORACLE_HOME%/forms90/java/$1"

# Virtual path for JInitiator downloadable executable and download page
AliasMatch ^/forms90/jinitiator/(.*) "%FORMS_ORACLE_HOME%/jinit/$1"

# Virtual path for runform.htm (used to run a form for testing purposes)
AliasMatch ^/forms90/html/(.*) "%FORMS_ORACLE_HOME%/tools/web90/html/$1"

# Configuration for JServ (if mod_jserv.c is available and not mod_oc4j.c)
<IfModule mod_jserv.c>
# Only configure for JServ if mod_oc4j is NOT available:
<IfModule !mod_oc4j.c>
# Virtual path mapping for FormsServlet and ListenerServlet.
# Purpose: paths to invoke the servlets should be /forms90/f90servlet
# and /forms90/l90servlet respectively.
# We map f90servlet to servlet.if90, and l90servlet to servlet.ifl90.
# The apJServAction directives (below) will then remap those.
AliasMatch ^/forms90/f90servlet(.) "/servlet.if90"
AliasMatch ^/forms90/l90servlet(.) "/servlet.ifl90"

ApJServMount /forms90/servlet /forms90
#
# Let the servlets be called by file extension (e.g /servlet.if90)
#
ApJServAction .if90 /forms90/servlet/f90servlet
ApJServAction .ifl90 /forms90/servlet/l90servlet
# Prevent access to the Forms Servlets by paths other than
# /forms90/f90servlet and /forms90/l90servlet.
# 1. Prevent access via the .if90 and .ifl90 file extensions:
<LocationMatch ^.*¥.if.*90>
```

```

        order deny,allow
        deny from all
    </LocationMatch>
    # 2. Stop access by class (by paths like
    #    /forms90/servlet/oracle.forms.servlet.FormsServlet)
    <LocationMatch ^/forms90/servlet/oracle%.forms.*>
        order deny,allow
        deny from all
    </LocationMatch>
</IfModule>
</IfModule>

# Config. for OC4J
<IfModule mod_oc4j.c>
    Oc4jMount /forms90          ProductGroup2
    Oc4jMount /forms90/f90servlet ProductGroup2
    Oc4jMount /forms90/f90servlet/* ProductGroup2
    Oc4jMount /forms90/l90servlet ProductGroup2
    Oc4jMount /forms90/l90servlet/* ProductGroup2
</IfModule>

```

Registry.dat

Registry.dat の説明とディレクトリは、第 2 章の「[Registry.dat](#)」を参照してください。

このファイルを編集する最大の理由は、アイコン設定を変更することです（「[アイコン](#)」を参照）。また、Registry.dat ファイルで次のセクションを変更すると、デフォルトのフォントとフォント設定も変更できます。

```

default.fontMap.defaultFontname=Dialog
default.fontMap.defaultSize=900
default.fontMap.defaultStyle=PLAIN
default.fontMap.defaultWeight=PLAIN

```

目的のフォント設定を使用するには、前述のいずれかの設定を変更します。たとえば、デフォルトのフォントを Times New Roman に変更する場合は、**Dialog** を **Times New Roman** で置換します。

デフォルトのフォント・マッピングは、次のように変更できます。

```

default.fontMap.appFontnames=Courier
New,Courier,courier,System,Terminal,Fixed,Fixedsys,Times,Times New Roman,MS Sans
Serif,Arial
default.fontMap.javaFontnames=MonoSpaced,MonoSpaced,MonoSpaced,Dialog,MonoSpace
d,Dialog,Dialog,Serif,Serif,Dialog,SansSerif

```

Windows NT には Java に対応しないフォントもあります。そのため、対応しないフォントが見つかったときに表示される Java 対応のフォントを指定する（マップさせる）ことがで

きます。前述の例では、default.fontMap.appFontnames の各フォントは default.fontMap.javaFontnames のフォントに対応しています。

その他の例は、「[デフォルトの Registry.dat](#)」を参照してください。

デフォルトの Registry.dat

```
#

# This is the Registry file.
#
# This file contains the logical [Java] Class name and an associated
# [numerical] identifier that will be used to refer to objects of the
# class in order to reduce the amount of information that needs to be
# repeatedly transmitted to the client.
#
# This file is of the Form understood by java.util.Properties (for now)
#
# The System Level sound file is relative to the CODEBASE
#
#
oracle.classById.1=oracle.forms.engine.Runform
oracle.classById.4=oracle.forms.handler.FormWindow
oracle.classById.5=oracle.forms.handler.AlertDialog
oracle.classById.6=oracle.forms.handler.DisplayList
oracle.classById.7=oracle.forms.handler.LogonDialog
oracle.classById.8=oracle.forms.handler.DisplayErrorDialog
oracle.classById.9=oracle.forms.handler.ListValuesDialog
oracle.classById.10=oracle.forms.handler.EditorDialog
oracle.classById.11=oracle.forms.handler.HelpDialog
oracle.classById.12=oracle.forms.handler.FormStatusBar
oracle.classById.13=oracle.forms.handler.MenuInfo
# oracle.classById.14=UNUSED
oracle.classById.15=oracle.forms.handler.ApplicationTimer
oracle.classById.16=oracle.forms.handler.MenuParametersDialog
oracle.classById.17=oracle.forms.handler.PromptListItem
oracle.classById.18=oracle.forms.handler.CancelQueryDialog
oracle.classById.257=oracle.forms.handler.TextFieldItem
oracle.classById.258=oracle.forms.handler.TextAreaItem
oracle.classById.259=oracle.forms.handler.FormCanvas
oracle.classById.261=oracle.forms.handler.ButtonItem
oracle.classById.262=oracle.forms.handler.CheckboxItem
oracle.classById.263=oracle.forms.handler.PopListItem
oracle.classById.264=oracle.forms.handler.TListItem
oracle.classById.265=oracle.forms.handler.CfmVBX
oracle.classById.266=oracle.forms.handler.CfmOLE
oracle.classById.267=oracle.forms.handler.RadioButtonItem
oracle.classById.268=oracle.forms.handler.ImageItem
```

```
oracle.classById.269=oracle.forms.handler.IconicButtonItem
oracle.classById.270=oracle.forms.handler.BlockScroller
oracle.classById.271=oracle.forms.handler.JavaContainer
oracle.classById.272=oracle.forms.handler.TabControl
oracle.classById.273=oracle.forms.handler.ComboBoxItem
oracle.classById.274=oracle.forms.handler.TreeItem
oracle.classById.281=oracle.forms.handler.PopupHelpItem

#
# Defaults for the Font details, all names are Java Font names. Each of
# these parameters represents the default property to use when none is
# specified.
#
# defaultFontname represents the default Java fontName.
# defaultSize      represents the default fontSize. Note that the size is
#                   multiplied by 100 (e.g. a 10pt font has a size of 1000).
# defaultStyle     represents the default fontStyle, PLAIN or ITALIC.
# defaultWeight    represents the default fontWeight, PLAIN or BOLD.
#
default.fontMap.defaultFontname=Dialog
default.fontMap.defaultSize=900
default.fontMap.defaultStyle=PLAIN
default.fontMap.defaultWeight=PLAIN

#
# Default Font Face mapping.
#
# appFontname represents a comma delimited list of Application Font Names.
# javaFontname represents a comma delimited list of Java Font Names.
#
# The number of entries in the appFontname list should match the number in
# the javaFontname list. The elements of the list are comma separated and
# *all* characters are taken literally, leading and trailing spaces are
# stripped from Face names.
#
# Note that this file uses the Java 1.1 Font names in order to be able to
# handle the NLS Plane (BUG #431051)
#
default.fontMap.appFontnames=Courier
New,Courier,courier,System,Terminal,Fixed,Fixedsys,Times,Times New Roman,MS Sans
Serif,Arial
default.fontMap.javaFontnames=MonoSpaced,MonoSpaced,MonoSpaced,Dialog,MonoSpaced,Dia
log,Dialog,Serif,Serif,Dialog,SansSerif

#
# The Application Level icon files are relative to the DOCUMENTBASE
# example: icons/
```

```
# or an absolute URL.
#   example: http://www.forms.net/~luser/d2k_project/
#
default.icons.iconpath=
default.icons.iconextension=gif
#
# Application level settings to control UI features
#
app.ui.lovButtons=false
app.ui.requiredFieldVA=false
#   The background color is specified as an RGB triple.
app.ui.requiredFieldVABGColor=255,0,0
```

環境変数とレジストリ設定のカスタマイズ

Forms ランタイムの環境変数は、f90webm (UNIX) もしくは ifweb90.exe (Windows) 実行可能ファイルに必要です。Listener Servlet は、実行可能ファイルを呼び出し、それを環境ファイルで提供されている変数値で初期化します。環境ファイルは、デフォルトでは <ORACLE_HOME>/forms90/server/default.env にあります。

default.env

Forms ランタイム・プロセスに必要な環境変数 (PATH、ORACLE_HOME、FORMS90_PATH など) は、環境ファイルで定義できます。環境ファイルに定義されていない環境変数は、サーブレット・エンジン (OC4J) から継承されます。環境ファイルは、formsweb.cfg の envFile パラメータで指定する必要があります。

環境変数をカスタマイズするには、次の事項を考慮してください。

- 環境変数は、Windows のレジストリでも指定できます。環境ファイルの値はレジストリの設定を上書きします。変数が環境ファイルで設定されていない場合は、レジストリ値が使用されます。
- レジストリ値を変更するには、管理者権限が必要です。
- 構成の変更を有効にするために、サーバーを再起動する必要はありません。
- 環境ファイルまたは Windows のレジストリで設定されていない環境変数は、サーブレット・エンジン (OC4J) である親プロセスの環境から継承されます。

default.env で指定されている重要な環境変数

環境変数	有効値	説明
ORACLE_HOME	<ORACLE_HOME> (デフォルト)	Oracle 製品の基本インストール・ディレクトリを示します。
PATH	<ORACLE_HOME>/bin (デフォルト)	Oracle 製品の実行モジュールを含みます。

default.env で指定されている重要な環境変数（続き）

環境変数	有効値	説明
FORM90_PATH	<ORACLE_HOME>/forms90（デフォルト）	<p>Oracle9i Forms が、実行するフォーム、メニューまたはライブラリを検索するときのパスを指定します。</p> <p>Windows の場合、複数のパスはセミコロン (;) で区切ります。</p> <p>UNIX の場合、複数のパスはコロン (:) で区切ります。</p>
NLS_LANG	<p>データベースで指定される言語、地域、キャラクタ・セットのデフォルト</p> <p>NLS_LANG の詳細は、Oracle Server のリファレンス・マニュアルを参照してください。</p>	<p>アプリケーションの言語依存動作を指定します（フランス語、日本語、ドイツ語など）。NLS_LANG には、次の 3 つのコンポーネントがあります。</p> <p><i>language</i> - Oracle のメッセージ、曜日、月などの規則に使用する対応言語を指定します。 <i>language</i> 引数は、<i>territory</i> または <i>charset</i> の引数のデフォルト値を指定するため、<i>territory</i> または <i>charset</i> のどちらかを（または両方とも）省略してもかまいません。</p> <p><i>territory</i> - アプリケーションの地域依存動作を指定します（米国、ブラジル、カナダなど）。</p> <p><i>charset</i> - クライアント・アプリケーションで使用するキャラクタ・セット（US7ASCII、JA16EUC など）を指定します。</p> <p>NLS_LANG の詳細は、Oracle Server のリファレンス・マニュアルを参照してください。</p>
DE_PREFS_TABSIZE	正の整数	PL/SQL エディタで Tab のサイズ（文字数）を定義します。デフォルト値は 2 です。
FORMS90_CATCHTERM	FALSE、TRUE（デフォルト）	<p>通常の Oracle9iAS Forms Services クラッシュ処理をバイパスし、生成するオペレーティング・システム・クラッシュ・ファイル（Unix 上の Core Dump など）を使用可能にします。この変数は通常、サポート担当から指示されない限り設定しません。</p>

default.env で指定されている重要な環境変数（続き）

環境変数	有効値	説明
FORMS90_TRACE_PATH	有効な書込み可能な O/S パス	<p>FORMS90_TRACE_PATH 環境変数は、Oracle9iAS Forms Services の実行可能ファイルのクラッシュの結果として生成されるダンプ・ファイルの位置を指定します。ダンプ・ファイルには、プロセスがクラッシュしたときのイベントに関する診断情報が含まれます。</p> <p>FORMS90_TRACE_PATH が設定されずに、クラッシュが発生した場合、Oracle9iAS Forms Services は実行可能ファイルの呼び出し元のディレクトリ内にダンプ・ファイルを置きます。</p> <p>この環境変数は、jserv.properties ファイルまたは default.env ファイルに設定します。</p>
FORMS90_MMAP	FALSE、TRUE（デフォルト）	<p>FORMS90_MMAP は、Oracle9i Forms アプリケーションの実行中にメモリー・マップド・ファイル i/o を使用するように Oracle9iAS Forms Services を構成します。これにより、複数のユーザーが同じフォームを実行しているときにシステムで必要なメモリー容量が減ります。デフォルトでは、FORMS90_MMAP は ON に設定されています。</p> <p>メモリー・マップド・ファイル i/o は使用中のフォームをロックし、開発者がフォームを新しいバージョンに置換することができないため、開発環境ではメモリー・マッピングを OFF (FALSE) に切り替えておくことをお勧めします。ただし、この切り替えを行うとメモリーの消費量が増加し、パフォーマンスが低下する可能性があります。本番システムでは、FORMS90_MMAP は TRUE に切り替える（または設定する）必要があります。</p>
FORMS90_REJECT_GO_DISABLED_ITEM	FALSE、TRUE（デフォルト）	<p>Oracle9i Forms では、Go_Item() ビルトインは、現在使用不可である項目にカーソルを移動できません。この変数を FALSE に設定すると、これが可能であった以前のバージョンの Oracle9i Forms の動作に戻ります。この変数は、下位互換性を確保する場合のみに使用が必要ありますが、将来のバージョンの製品では削除される可能性があります。</p>

default.env で指定されている重要な環境変数（続き）

環境変数	有効値	説明
FORMS90_SWITCH_ JAVA_EVENTS	FALSE（デフォルト）、TRUE	この変数を使用して、Oracle9i Forms が埋込み JavaBeans からのイベントをキューアップしたり、処理する方法を変更します。この変数は通常、サポート担当から指示されない限り設定しません。
FORMS90_SEPARATE_ DEBUGGER	FALSE（デフォルト）、TRUE	この変数は、Forms デバッガで作成されたウィンドウを Forms Builder に表示する方法に影響を与えます。この変数を TRUE に設定すると、デバッガ・ウィンドウはメインの Forms MDI インタフェースから外部的に作成されます。この設定を使用するのは、デバッガで JAWS などのスクリーン・リーダー・ソフトウェアを使用する必要がある場合です。
FORMS90_CLAF	<NULL>（デフォルト）、TRUE	FORMS90_CLAF を TRUE に設定すると、Forms Builder ユーザー・インタフェースは "Classic" (6i) Forms のルック・アンド・フィールに戻ります。これは、コントラストの高い配色が必要なユーザーに適しています。

注意：Windows NT では、Oracle9i Forms は、環境変数として設定されていない限り、レジストリから Oracle 環境設定を読み込みます。

デフォルトの Windows 用 default.env ファイル

```
# default.env - default Forms environment file, Windows version
#
# This file is used to set the Forms runtime environment parameters.
# If a parameter is not defined here, the value in the Windows registry
# will be used. If no value is found in the registry, the value used will
# be that defined in the environment in which the servlet engine (OC4J
# or JServ) was started.
#
# NOTES
# 1/ The Forms installation process should replace all occurrences of
#    <percent>FORMS_ORACLE_HOME<percent> with the correct ORACLE_HOME
#    setting, and all occurrences of <percent>O_JDK_HOME<percent> with
#    the location of the JDK (usually $ORACLE_HOME/jdk).
#    Please make these changes manually if not.
# 2/ Some of the variables below may need to be changed to suite your needs.
#    Please refer to the Forms documentation for details.
#
ORACLE_HOME=%FORMS_ORACLE_HOME%
```

```
#
# Search path for Forms applications (.fmx files, PL/SQL libraries)
# If you need to include more than one directory, they should be semi-colon
# separated (e.g. /private/dir1;/private/dir2)
#
# FORMS90_PATH=%FORMS_ORACLE_HOME%/forms90

#
# The PATH setting is not required, if the Forms executables and dll's are
# in %ORACLE_HOME%\bin:
#
# PATH=%FORMS_ORACLE_HOME%\bin

#
# Settings for Graphics
# -----
# NOTE: These settings are only needed if Graphics applications
# are called from Forms applications
#

#
# Please uncomment the following and put the correct 9i
# oracle_home value to use Graphics applications.
#
#ORACLE_GRAPHICS6i_HOME=<your Graphics 9i oracle_home here>

#
# Search path for Graphics applications
#
#GRAPHICS60_PATH=

#
# System settings
# -----
# You should not normally need to modify these settings
#

FORMS90=%FORMS_ORACLE_HOME%\forms90
#
# Java class path
# This is required for the Forms debugger
# You can append your own Java code here)
#
CLASSPATH=%FORMS_ORACLE_HOME%\jlib\debugger.jar;%FORMS_ORACLE_
HOME%\jlib\ewt3.jar;%FORMS_ORACLE_HOME%\jlib%\share.jar;%FORMS_ORACLE_
HOME%\jlib\utj90.jar
```

デフォルトの UNIX 用 default.env ファイル

```
# default.env - default Forms environment file, Solaris version
#
# This file is used to set the Forms runtime environment parameters.
# If a parameter is not defined here, the value used will be that defined
# in the environment in which the servlet engine (OC4J or JServ) was started.
#
# NOTES
# 1/ The Forms installation process should replace all occurrences of
#    <percent>FORMS_ORACLE_HOME<percent> with the correct ORACLE_HOME
#    setting, and all occurrences of <percent>O_JDK_HOME<percent> with
#    the location of the JDK (usually $ORACLE_HOME/jdk).
#    Please make these changes manually if not.
# 2/ Some of the variables below may need to be changed to suite your needs.
#    Please refer to the Forms documentation for details.
#
ORACLE_HOME=<FORMS_ORACLE_HOME>

#
# Search path for Forms applications (.fmx files, PL/SQL libraries)
#
FORMS90_PATH=<FORMS_ORACLE_HOME>/forms90

#
# Java class path
#   This is required for the Forms debugger
#   You can append your own Java code here)
#
CLASSPATH=<FORMS_ORACLE_HOME>/jlib/debugger.jar:<FORMS_ORACLE_
HOME>/jlib/utj90.jar:<FORMS_ORACLE_HOME>/jlib/ewt3.jar:<FORMS_ORACLE_
HOME>/jlib/share.jar

#
# The PATH setting is not required if the Forms executables are
# in <ORACLE_HOME>/bin:
#
PATH=<FORMS_ORACLE_HOME>/bin
#
# Settings for Reports
# -----
# NOTE: This setting is only needed if Reports applications
# are called from Forms applications
#
REPORTS_CLASSPATH=<FORMS_ORACLE_HOME>/jlib/zrclient.jar:<FORMS_ORACLE_
HOME>/reports/jlib/rwrun.jar
```

```
#
# Settings for Graphics
# -----
# NOTE: These settings are only needed if Graphics applications
# are called from Forms applications
#

#
# Please uncomment the following and put the correct 6i
# oracle_home value to use Graphics applications.
#
#ORACLE_GRAPHICS6i_HOME=<your Graphics 6i oracle_home here>

#
# Search path for Graphics applications
#
GRAPHICS60_PATH=

#
# Settings for forms9i tracing and logging
# -----
# NOTE: This entry has to be uncommented to enable tracing and
# logging.

#FORMS90_TRACE_PATH=<FORMS_ORACLE_HOME>/forms90/server

#
# System settings
# -----
# You should not normally need to modify these settings
#
#
# Path for shared library objects
#   This is highly platform (if not machine) specific !  At install time
#   <percent>LD_LIBRARY_PATH<percent> should be replaced with the
#   actual value of the LD_LIBRARY_PATH environment variable (at install
#   time).  That should ensure we have the paths for such necessities as
#   the motif and X11 libraries.
# Explanations:
#   - Reports needs the path for libjava.so
#     (/cdm/solaris/o_jdk/1_2_2_0_0/jre/lib/sparc)
#   - Forms needs two paths to the jre, for libjvm.so and libhpi.so
#   - In ojdk 1.3.1 the location of libjvm.so is lib/sparc (there is no
#     classic directory) so we do not include the ../classic directory
#     below.  There are other versions of libjvm.so (in directories server,
#     client and hotspot) but we will use the version in lib/sparc for now.
#
LD_LIBRARY_PATH=%FORMS_ORACLE_HOME%/lib:%O_JDK_HOME%/jre/lib/sparc:%O_JDK_
```

```
HOME%/jre/lib/sparc/native_threads:%LD_LIBRARY_PATH%
```

ユーザー独自のテンプレート HTML ファイルの作成

(Oracle が提供するテンプレートを変更して) ユーザー独自の HTML ファイル・テンプレートを作成することを検討してください。ユーザー独自のテンプレートを作成すると、標準の Forms パラメータおよびパラメータ値をテンプレートに直接指定できます。作成したテンプレートには、標準テキスト、ブラウザ・ウィンドウ・タイトル、またはイメージ（会社のロゴなど）を挿入でき、これらは Web で使用できるフォームを実行するときに参照できる、最初の Web ページに表示されます。標準パラメータ、値、および追加のテキストまたはイメージを追加すると、特定のアプリケーションのテンプレートをカスタマイズするために必要な作業量を減らすことができます。テキスト、イメージ、またはウィンドウ・タイトルを追加するには、テンプレート HTML ファイルに適切なタグを挿入します。

Oracle9i Forms アプリケーションへのグラフィックの挿入

Oracle9i Forms アプリケーションとグラフィック・アプリケーションを統合するには、次のように Forms Servlet 環境にパス定義を設定し、グラフィックを含める必要があります。

```
PATH=<FORMS9I_HOME>/bin;<ORACLE_GRAPHICS6I_HOME>/bin.
```

Forms Servlet 環境のパス定義は、サーブレット・コンテナのパス定義から取得します。パスを定義するファイルまたはディレクトリは、サーブレット・コンテナごとに異なります。

グラフィックの詳細は、『Oracle9i Forms Developer and Forms Services Forms アプリケーションの Forms6i からの移行』を参照してください。

Oracle9iAS Forms Services で使用するアイコンとイメージの配置

この項では、アイコンとイメージのデフォルト・ディレクトリおよび検索パスの指定方法を説明します。

アイコン

Oracle9iAS Forms Services アプリケーションを配布する場合、使用するアイコン・ファイルは JPG や GIF（GIF がデフォルトの形式）などの Web 対応形式にする必要があります。

デフォルトでは、アイコンは DocumentBase ディレクトリを基準とした相対パスにあります。すなわち、DocumentBase はアプリケーションの開始 HTML ファイルのベース・ディレクトリを基準にした相対ディレクトリからイメージを検索します。開始 HTML ファイルは Forms Servlet により動的にレンダリングされるので、forms90 ディレクトリが DocumentBase になります。

たとえば、アプリケーションでボタンのアイコンのディレクトリを myapp/<iconname> に定義している場合、そのアイコンはディレクトリ forms90/myapp 内で検索されます。

デフォルトのディレクトリを変更するには、formsweb.cfg 構成ファイル内の imageBase パラメータを codebase に設定します。あるいは、forms90/java/oracle/forms/registry ディレクトリにある Registry.dat ファイルの default.icons.iconpath 値を変更することもできます。

imageBase パラメータを codebase に設定すると、Oracle9i Forms は forms90/java ディレクトリでアイコン・ファイルを検索します。この設定を使用するのは、ユーザーのイメージが Java アーカイブ・ファイルに格納されている場合です。イメージをアプリケーションや Oracle9i Forms のインストールに依存しない中央のディレクトリに格納する場合は、Registry.dat 構成ファイルでイメージのディレクトリを変更できます。

アイコンの Java アーカイブへの格納

アプリケーションで多くのカスタム・アイコン・イメージを使用する場合、そのアイコンを Java アーカイブ・ファイルに格納し、imageBase 値を codebase に設定しておくようお勧めします。アイコン・ファイルは、Java Development Kit (JDK) の Jar コマンドを使用して Java アーカイブに zip 圧縮できます。

たとえば、コマンド `Jar -cvf myico.jar *.gif` を使用すると、拡張子 gif の付いたファイルはすべて myico.jar という名前のアーカイブ・ファイルに zip 圧縮されます。

Oracle9i Forms がこのアーカイブに格納されたアイコン・ファイルにアクセスするには、そのアーカイブを forms90/java ディレクトリに格納する必要があります。また、アーカイブ・ファイルの名前は formsweb.cfg ファイルのカスタム・アプリケーション・セクションで使用されているアーカイブ・タグの一部（たとえば、archive_jini=f90all_jinit.jar, myico.jar）にする必要があります。これで、初期アプリケーションが起動されたら、アイコン・ファイルはクライアントにダウンロードされ、アーカイブ・ファイルが変更されない限り永久的に保存されます。

注意： Oracle9i Forms のデフォルトのアイコン（デフォルトのスマート・アイコン・バーに表示されるアイコンなど）は、f90all.jsr ファイルの一部であるため、配置する必要はありません。

Registry.dat へのアイコン変更の追加

アイコンの変更を、アプリケーションで使用している Registry.dat ファイルに追加する場合は、既存の Registry.dat ファイルをコピーして、コピーしたファイルを編集するようお勧めします。

Registry.dat ファイルをコピーするには：

1. <ORACLE_HOME>/forms90/java/oracle/forms/registry ディレクトリにある Registry.dat テキスト・ファイルを別のディレクトリにコピーします。このディレクトリは、Web サーバーの仮想ディレクトリ（/appfile など）へマップする必要があります。
2. 新規ファイルをリネームします（myapp.dat など）。

3. アイコンのディレクトリを指定する `iconpath` パラメータを次のとおりに変更します。
- ```
default.icons.iconpath=/mydir or http://myhost.com/mydir
(絶対パスの場合)
または
default.icons.iconpath=mydir
(DocumentBase ディレクトリから始まる相対パスの場合)
```
4. `iconextension` パラメータを次のように変更します。
- ```
default.icons.iconextension=gif
または
default.icons.iconextension=jpg
```

アプリケーション・ファイルを参照するには：

`formsweb.cfg` ファイル内の名前を付けた固有の構成セクションで、`serverApp` パラメータの値を変更し、ユーザーのアプリケーション・ファイルのディレクトリと名前 で値を設定します。

たとえば、次のように指定します。

```
[my_app]
ServerApp=/appfile/myapp
(絶対パスの場合)

または

[my_app]
ServerApp=appfile/myapp
(CodeBase ディレクトリを基準とする相対パスの場合)
```

表 3-2 アイコンのディレクトリ・ガイド

アイコンのディレク トリ	用途	方法
DocumentBase	デフォルト。カスタム・アイコン が少ないかまったくないアプリ ケーション。	アイコンを <code>forms90</code> ディレクトリ か <code>forms90</code> の相対ディレクトリに 格納します。
Java Archives	多くのカスタム・アイコンを使用 するアプリケーション。	<code>ImageBase</code> を "codebase" に設定 し、アイコンの Java アーカイ ブ・ファイルを作成したら、アー カイブ・ファイルを <code>formsweb.cfg</code> のアーカイブ・パ ラメータに追加します。

表 3-2 アイコンのディレクトリ・ガイド（続き）

アイコンのディレクトリ	用途	方法
Registry.dat	Oracle9i Forms のインストールとは異なる場所（別のサーバー）に保存されているカスタム・アイコンがあるアプリケーション。 フォント・マッピングなどのように Registry.dat ファイルに他の変更を行う必要がある場合に有効です。	Registry.dat をコピーし、formsweb.cfg の serverApp パラメータを変更します。

スプラッシュ画面イメージおよびバックグラウンド・イメージ

アプリケーションを配布する場合、（接続中に表示される）スプラッシュ画面イメージおよびバックグラウンド・イメージ・ファイルを指定することができます。

これらのイメージは、次に示すように HTML ファイルまたは formsweb.cfg ファイルで定義します。

HTML の例

```
<PARAM NAME="splashScreen" VALUE="splash.gif">  
  
<PARAM NAME="background" VALUE="back.gif">
```

formsweb.cfg ファイルの例

```
splashScreen=splash.gif  
  
background=back.gif
```

スプラッシュ画面およびバックグラウンド・イメージ・ファイルのデフォルト・ディレクトリは、ベース HTML ファイルが含まれている DocumentBase ディレクトリ内にあります。

アイコンとイメージを含むカスタム JAR ファイル

（スプラッシュ画面またはバックグラウンドの）アイコンまたはイメージを使用するたびに、HTTP 要求が Web サーバーに送信されます。クライアントとサーバー間の HTTP ラウンドトリップ数を減らすには、Java アーカイブ（JAR）ファイルにアイコンおよびイメージを格納するための機能が必要です。この方法を使用すると、JAR ファイルをダウンロードするのに、1 回の HTTP ラウンドトリップのみで済みます。

JAR ファイルの作成

SunSoft JDK には、*jar* と呼ばれる実行可能ファイルが含まれています。このユーティリティを使用すると、Java アーカイブ内にファイルを格納できます。詳細は、www.java.sun.com を参照してください。

例：

```
jar -cvf myjar.jar Splash.gif Back.gif icon1.gif
```

このコマンドにより、*myjar.jar* と呼ばれる単一の JAR ファイルに 3 つのファイル (*Splash.gif*、*Back.gif*、*icon1.gif*) が格納されます。

JAR ファイル内でのファイルの使用

アイコンおよびイメージのデフォルトの検索パスは、*DocumentBase* を基準とした相対パスです。ただし、それらのファイルを格納するために JAR ファイルを使用する場合は、検索パスは、Java アプレットを含むディレクトリ、*CodeBase* ディレクトリを基準とする相対パスにする必要があります。

JAR ファイルを使用してアイコンおよびイメージを格納する場合は、*formsweb.cfg* ファイルか HTML ファイル内の *imageBase* パラメータを使用して、検索パスが *CodeBase* を基準とした相対パスであることを指定する必要があります。

このパラメータは次の 2 つの異なる値が指定可能です。

- **DocumentBase** 検索パスは *DocumentBase* ディレクトリを基準とした相対パスです。これはデフォルトの動作です。
- **CodeBase** 検索パスは JAR ファイルを使用できるようにする *CodeBase* ディレクトリを基準とした相対パスです。

この例では、アイコンを含む JAR ファイルを使用して、検索が *CodeBase* の相対パスになるように指定します。パラメータ "*imageBase*" を設定していない場合は、*DocumentBase* を基準とした相対的な検索となり、アイコンは JAR ファイルからは検索されません。

例 (*formsweb.cfg*)：

```
archive=f90all.jar, icons.jar  
imageBase=codebase
```

アイコンおよびイメージの検索パス

アイコンおよびイメージの検索パスは次の内容によって異なります。

- カスタム・アプリケーション・ファイルで指定した内容（アイコンの場合）
- *formsweb.cfg* ファイルまたは HTML ファイルの *splashScreen* パラメータおよび *background* パラメータで指定した内容（イメージの場合）

- formsweb.cfg ファイルまたは HTML ファイルの imageBase パラメータで指定した内容 (アイコンとイメージの両方の場合)

Forms Services では、指定した内容に応じてアイコンが検索されます。この例では、次のように仮定します。

- host はホスト名。
- documentbase は HTML ファイルを指す URL。
- codebase は、(formsweb.cfg ファイルまたは HTML ファイルで指定した) 開始クラス・ファイルのディレクトリを指す URL。
- mydir は、アイコンまたはイメージ・ディレクトリを指す URL。

DocumentBase

デフォルトの検索パスは、DocumentBase を基準とした相対パスです。この場合、imageBase パラメータを指定する必要はありません。

表 3-3 アイコンおよびイメージの検索パス

	指定ディレクトリ	Forms Services で使用される検索パス
アイコン	デフォルト	http://host/documentbase
	iconpath=mydir (アプリケーション・ファイルで指定)	http://host/documentbase/mydir (相対パス)
	iconpath=/mydir (アプリケーション・ファイルで指定)	http://host/mydir (絶対パス)
イメージ	file.gif (たとえば、formsweb.cfg で splashscreen=file.gif として指定)	http://host/documentbase/file.gif
	mydir/file.gif	http://host/documentbase/mydir/file.gif (相対パス)
	/mydir/file.gif	http://host/mydir/file.gif (絶対パス)

CodeBase

次に示すように、formsweb.cfg ファイルで imageBase=CodeBase パラメータを使用して、JAR ファイル内でのアイコンおよびイメージの検索を可能にします。

表 3-4 Oracle9iAS Forms Services で使用されるアイコンとイメージの検索パス

	指定ディレクトリ	Forms Services で使用される検索パス
アイコン	デフォルト	http://host/codebase または JAR ファイルのルート
	iconpath=mydir (アプリケーション・ファイルで指定)	http://host/codebase/mydir または JAR ファイル内の mydir ディレクトリ (相対パス)
	iconpath=/mydir (アプリケーション・ファイルで指定)	http://host/mydir (絶対パス) JAR ファイルは使用されない。
イメージ	file.gif	http://host/codebase/file.gif または JAR ファイルのルート
	mydir/file.gif (HTML ファイルで指定)	http://host/codebase/mydir/file.gif または JAR ファイル内の mydir ディレクトリ (相対パス)
	/mydir/file.gif (HTML ファイルで指定)	http://host/mydir/file.gif (絶対パス) JAR ファイルは使用されない。

Forms Listener Servlet での HTTPS の使用

Oracle9i Forms に HTTPS を使用するのとは、他の Web ベースのアプリケーションで HTTPS を使用するのと変わりません。

サーバー要件

HTTPS では、デジタル証明書を使用する必要があります。Oracle9iAS Forms Services サーブレットはユーザーの Web サーバーからアクセスできるため、Oracle9i Forms のクライアントとサーバー間の通信に特別な証明書を購入する必要はありません。正式な認証局からユーザーの Web サーバー用の証明書を購入するだけで済みます。

クライアント要件 : Oracle JInitiator での HTTPS の使用

エンド・ユーザーが Oracle JInitiator を Web ブラウザ JVM として実行している場合、ユーザーの Web サイトの SSL 証明書のルート認証局が JInitiator `certdb.txt` ファイルに定義されていることを確認する必要があります。

`certdb.txt` ファイルは通常、JInitiator がインストールされているマシンの `c:\program files\oracle\jinitiator <version>\lib\security` にあります。

注意： テスト目的のために Oracle9iAS から提供されたテスト証明書を使用している場合、JInitiator `certdb.txt` ファイルを編集して、`<ORACLE_HOME>/Apache/Apache/conf/ssl.crt/demoCAcert.txt` にあるデモ・ルート証明書の内容を追加する必要があります。そうしないと、フォームを実行したときに次のようなエラーが表示されます。

```
javax.net.ssl.SSL の例外 : SSL ハンドシェイクに失敗しました
:X509CertChainInvalidErr
```

Oracle JInitiator の詳細は、[付録 A「JInitiator」](#)を参照してください。

ユーザー ID/ パスワード非表示機能の使用

Oracle9iAS Forms Services では、Forms Servlet で生成された HTML に `userid` パラメータ値が含まれていません。

デフォルトでこの機能を使用すると、Oracle9iAS Forms Services では次の処理を行うことができます。

- "userid" (大 / 小文字の区別なし) のパラメータを使用して `user/password@database` を指定できます。Oracle9i Forms のインストール時に提供されるデフォルトのベース HTML ファイルを使用している場合は、この機能をすでに使用しています。これらのファイルには、`"userid=%userid%"` のような構文が含まれています。
- 静的 HTML ファイルではなく Forms Servlet を使用できます。

認証プロキシを使用した Oracle9i Forms アプリケーションの実行

Oracle9iAS のインストール・プロセスで設定されたデフォルトの構成では、認証プロキシがサポートされます。認証プロキシでは、アプリケーションを実行する接続先サーバーにアクセスできるように、ユーザーがユーザー名とパスワードを指定する必要があります。認証プロキシは通常、ユーザーがログオンしているか（または認証されているか）どうかを検出するために Cookie を設定します。Cookie はその後のすべてのネットワーク要求時に送信され、不要なログオン・プロンプトを回避します。

ユーザーが JInitiator で Netscape を実行している場合、プロキシの認証 Cookie がすべての要求とともにサーバーへ送信されるようにするには、特定の構成要件が必要です。基本的要件として、JInitiator がアクセスする必要があるすべての URL は (JAR ファイルと Forms Listener Servlet の場合)、HTML ページのドキュメント・ベースの下に位置する必要があります。これには、Forms Servlet を使用してページを生成し、`https://myserver.com/forms90/f90servlet?config=myApp` などの `/forms90` の下の URL を使用してそのページを呼び出します。

Oracle9iAS のインストール・プロセスで設定されている codebase およびサーバーの URL 値は、`/forms90/java` および `/forms90/l90servlet` です。これらはページのドキュメント・ベース (`/forms90`) の下にあるので、認証プロキシが機能します。

言語検出の有効化

Oracle9i Forms アーキテクチャは、複数の言語の配置をサポートしています。この機能の目的は、ユーザーが使用する言語に一致する適切な構成を自動的に選択することです。この機能を使用すれば、だれでも同じ URL を使用して Oracle9i Forms アプリケーションを実行でき、しかもユーザーが使用する言語でアプリケーションを実行することができます。統合翻訳ツールは提供されていないため、翻訳済のアプリケーション・ソース・ファイルを用意する必要があります。

言語検出を指定する手順は、次のとおりです。

`formsweb.cfg` の各構成セクションに、`<config_name>.<language-code>` などの名前を付けた言語固有のセクションを作成できます。たとえば、構成セクション `"hr"` を作成し、フランス語と中国語を作成する場合は、構成セクションは次のようになります。

```
[hr]
lookAndFeel=oracle
width=600
height=500
envFile=default.env
workingDirectory=/private/apps/hr

[hr.fr]
envFile=french.env
workingDirectory=/private/apps/hr/french

[hr.zh]
envFile=chinese.env
workingDirectory=/private/apps/hr/chinese
```

言語検出機能の仕組み

Forms Servlet が特定の構成 (`http://myserv/servlet/f60servlet?config=hr` など) の要求を受信した場合、要求ヘッダー `"accept-language"` からクライアントの言語設定を取得します。この設定には、優先順位の付いた言語のリストがあります。たとえば、`accept-language: de,`

`fr`, `en_us` は、優先順位がドイツ語、フランス語、米語となります。サーブレットは、最初の言語に一致する言語固有の構成セクションを検索します。1つの構成セクションが見つければ、次の構成セクションへと検索を続けていきます。言語固有の構成が検索できなかった場合は、基本の構成を使用します。

Forms Servlet が、特定の構成が指定されていない要求を受信した場合 ("`config=URL` パラメータがない `http://myserv/servlet/f60servlet` など)、サーブレットはデフォルトのセクションで最初の言語 (`[.fr]` など) に一致する言語固有のセクションを検索します。

マルチレベルの継承

基本構成のあらゆる言語固有の可変の値で共通の値が重複しないように、使いやすさを考慮した結果、言語固有のセクションでは言語固有のパラメータしか定義できないように設計されています。オラクル社では、次の4つのレベルの継承をサポートしています。

- 1) `config=myconfig` のような URL の問合せパラメータを使用した特定の構成が必要な場合、各パラメータの値は、ユーザーのブラウザの言語設定で最も優先順位の高い言語固有の構成セクション (たとえば、セクション `[myconfig.fr]`) で検索されます。
- 2) 検出されなかった場合、その値は基本構成セクション (`[myconfig]`) で検索されません。
- 3) そこでも検出されなかった場合は、言語固有のデフォルト・セクション (`[.fr]` など) で検索されます。
- 4) 最終的には、デフォルト・セクションで検索されます。

一般に、言語間で最も変更されやすいパラメータは "`workingDirectory`" と "`envFile`" です。各言語に異なる `envFile` 設定を使用すると、`NLS_LANG` (異なるキャラクタ・セット、日付、数値書式が可能) と `FORMS90_PATH` (言語固有の `fmx` ファイルの取得) に異なる値を設定できます。異なる `workingDirectory` 設定を使用することにより、言語固有の `fmx` ファイルを別の方法で取得できます。

HTTP Listener と OC4J での Oracle9iAS Forms Services の使用

概要

Oracle9iAS Containers for J2EE (OC4J) は、すべて Java で作成された完全な J2EE (Java 2 Platform Enterprise Edition) サーバーで、標準の Java Runtime Environment (JRE) で実行されます。OC4J は、OC4J Web コンテナなどを含む完全な J2EE 環境を提供します。

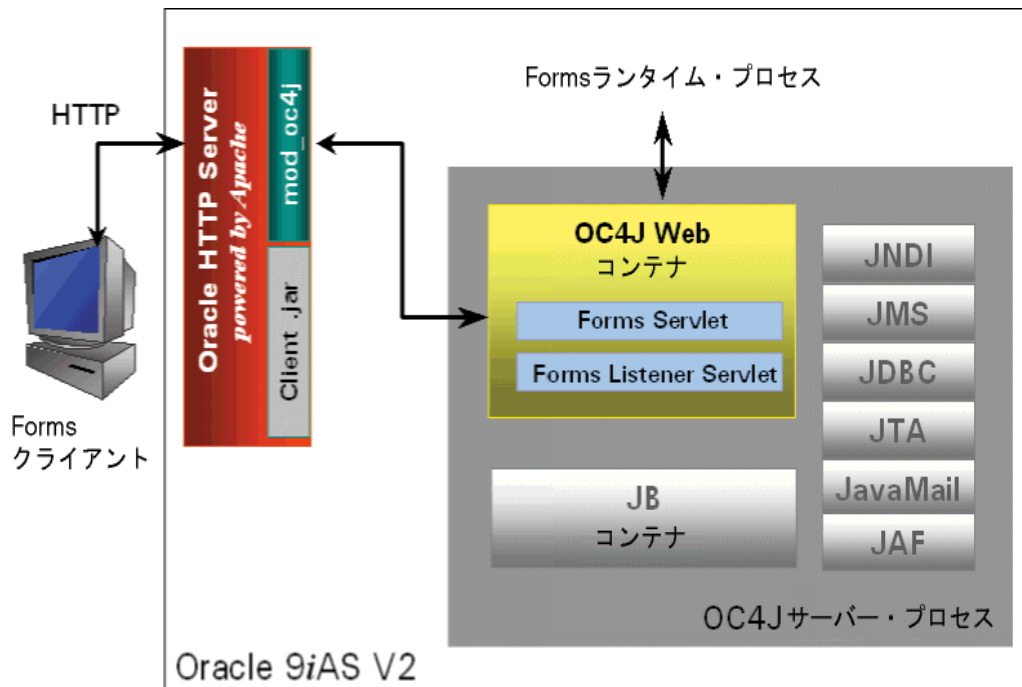
この章には、次の項が含まれています。

- [OC4J サーバー・プロセス](#)
- [パフォーマンス / 拡張性のチューニング](#)
- [OC4J のロード・バランス](#)

OC4J サーバー・プロセス

簡単な使用例では、Forms Servlet は開始 HTML ファイルをレンダリングし、Forms Listener Servlet に関する情報をクライアントに提供します。次に、HTTP 要求が Oracle HTTP Listener で受信され、Listener はこれを OC4J 内で動作している Forms Listener Servlet に渡します。Forms Listener Servlet は Forms Server ランタイム・プロセスを確立し、クライアント・ブラウザとランタイム・プロセス間で継続的な通信を維持します。多くのユーザーが Oracle9i Forms セッションを要求した場合、その要求は HTTP Listener で受信されます。HTTP Listener は再びその要求を Forms Listener Servlet に渡し、Servlet はさらにランタイム・プロセスを確立します。Forms Listener Servlet は、多数の Forms ランタイム・セッションを同時に処理できます。同時ユーザー数を制限する必要がありますが、このアーキテクチャでは高いパフォーマンスを得るためのチューニングや構成を行うことができます（「[パフォーマンス / 拡張性のチューニング](#)」を参照）。

OC4J は次のアーキテクチャで動作します。



パフォーマンス / 拡張性のチューニング

Forms Listener Servlet をチューニングする手順は、スループットの高いサーブレット・アプリケーションをチューニングする手順と同様です。

HTTPD プロセス数の制限

過剰な HTTPD プロセスの生成（多量のメモリー消費）を回避するには、Oracle HTTP Listener 構成ファイル（httpd.conf）に次のディレクティブを設定します。

KeepAlive Off

（別のアプリケーションなどに）KeepAlive On を使用する必要がある場合は、KeepAliveTimeout を低めの数字に設定します（たとえば、デフォルトの 15 秒）。

高い値への maxClient ディレクティブの設定

多数の HTTPD デーモンを作成するタイミングは、HTTP Listener で判断するのが最良の方法です。そのため、構成ファイル（httpd.conf）では MaxClients ディレクティブを高い値に

設定します。ただし、このパラメータを設定する際にはシステム上のメモリー容量を考慮する必要があります。

MaxClients=256 は、同時に要求を処理する HTTPD プロセスをリスナーが最大 256 作成できることを示します。

HTTP 要求が急激に増大し、必要な HTTPD プロセスを開始する時間を短くする必要がある場合は、適切な数のプロセスに対応するように、MinSpareServers と MaxSpareServers (httpd.conf 内) を設定できます。ただし、ほとんどのサイトではそれぞれ 5 と 10 のデフォルト値で十分です。

OC4J のロード・バランス

Forms Listener Servlet のアーキテクチャでは、標準の HTTP ロード・バランス技術を使用して、システムのロード・バランスを実現します。

Oracle HTTP Listener で提供するロード・バランス・メカニズムでは、複数の OC4J インスタンスを HTTP プロセスと同じホストで実行することも、複数の異なるホストやその組合せで実行することもできます。次に、HTTP Listener は HTTP 要求を OC4J インスタンスにルーティングします。

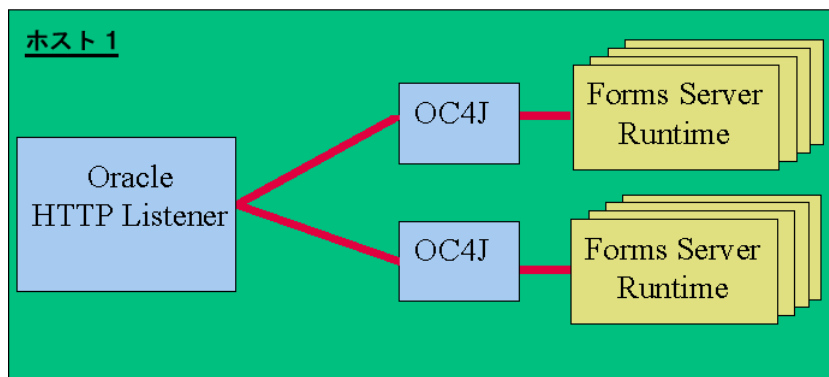
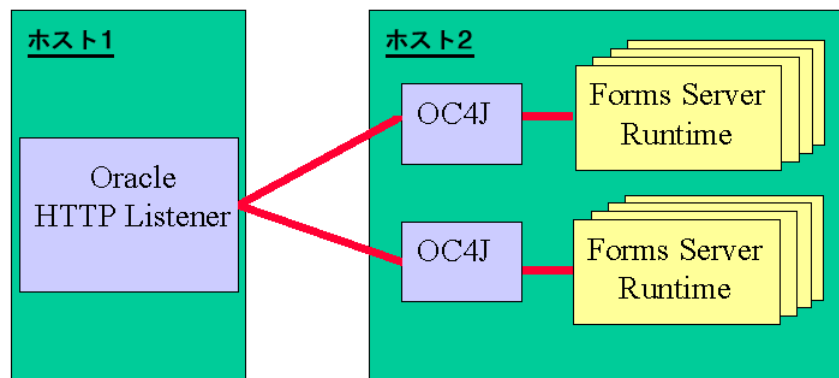
次の使用例では、ホストを組み合わせた使用方法を紹介します。ユーザーのサイトにどのような選択が最も適しているかは、様々な要因により異なります。

この機能の詳細は、『Oracle9i Application Server パフォーマンス・ガイド』の OC4J の章を参照してください (Oracle9iAS Disk 1 CD または OTN (<http://otn.oracle.com/> 英語サイト) または OTN-Japan (<http://otn.oracle.co.jp/> 日本語サイト) で入手可能)。

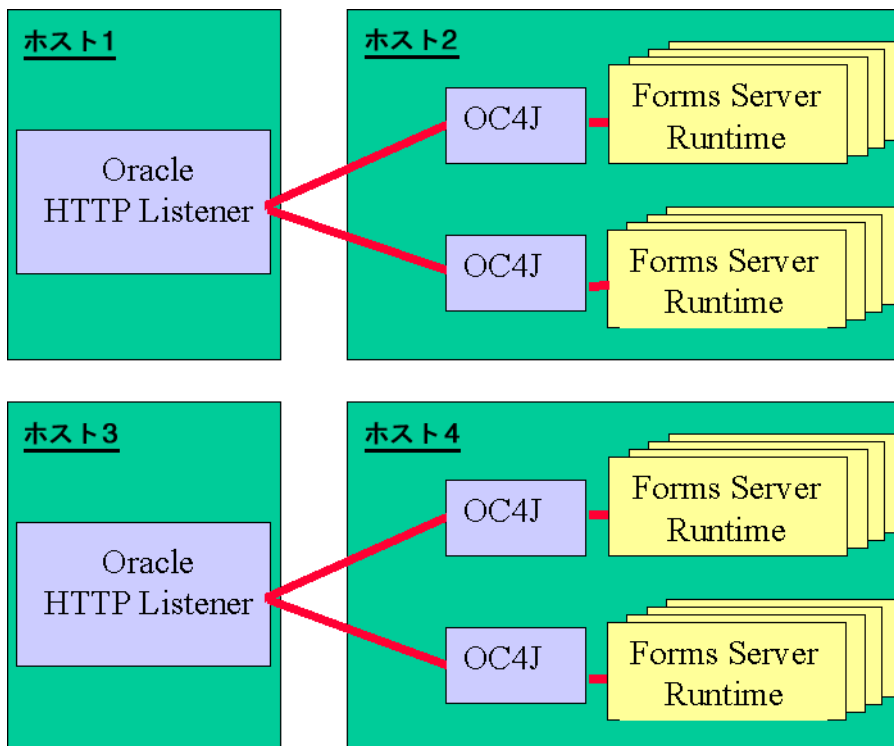
Forms 固有の情報の詳細は、『Oracle9i Forms Developer and Forms 日本語リリース・ガイド』を参照してください。

この項では、4 つの使用例を紹介します。

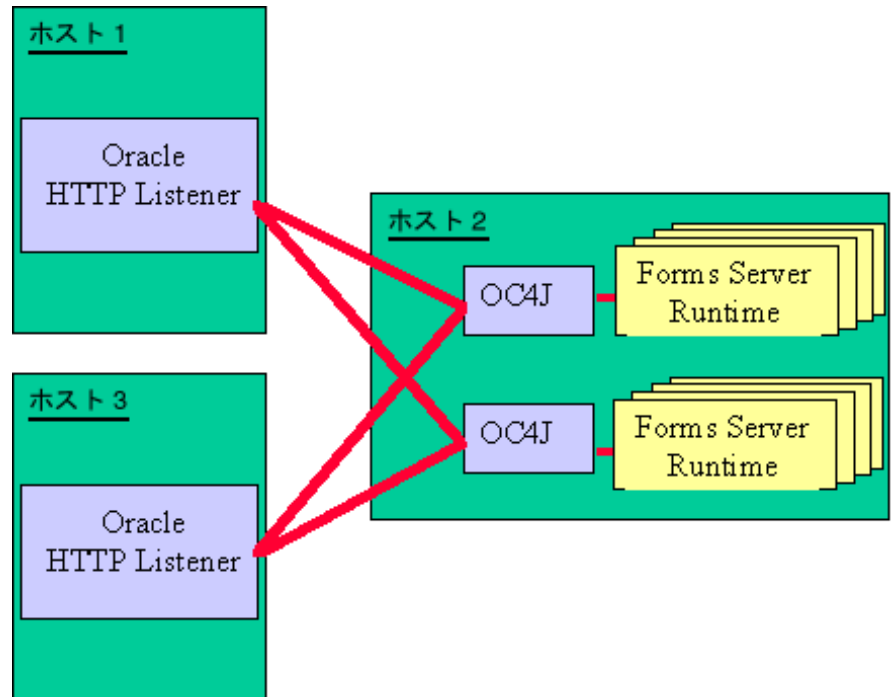
- Oracle HTTP Listener と同じホスト上にある複数の OC4J エンジン間で受信要求を分散する方法
- Oracle HTTP Listener と異なるホスト上にある複数の OC4J エンジン間で受信要求を分散する方法
- Oracle HTTP Listener が複数のホストでそれぞれ動作しており、それとは別の複数の異なるホスト上にある複数の OC4J エンジン間で受信要求を分散する方法
- Oracle HTTP Listener が複数のホストでそれぞれ動作しており、それとは別の 1 つのホスト上にある複数の OC4J エンジン間で受信要求を分散する方法

ケース 1: Oracle HTTP Listener と同じホスト上にある複数の OC4J エンジン**ケース 2: Oracle HTTP Listener と異なるホスト上にある複数の OC4J エンジン**

ケース 3: それぞれ異なるホスト上にある複数の OC4J エンジンと複数の Oracle HTTP Listeners



ケース 4: 異なるホスト上にある複数の Oracle HTTP Listeners と同じホスト上にある複数の OC4J エンジン



SSO と OID での Oracle9iAS Forms Services の使用

概要

Oracle9iAS Forms Services アプリケーションは、Oracle Login Server (Single Sign-On Server) と Oracle Internet Directory (OID) を使用したシングル・サインオン (SSO) 環境で実行し、ユーザー名とパスワード情報を格納できます。SSO は、ポータルを通して複数の Web アプリケーションにアクセスできる Oracle9i Application Server のポータル環境で機能するように設計されています。SSO がない場合、ユーザーはアクセスするアプリケーションごとに ID やパスワードを保持する必要があります。複数のアカウントやパスワードを保持するのは安全とは言えず、しかも効率的ではありません。

この章には、次の項が含まれています。

- [Single Sign-On \(SSO\)](#)
- [認証フロー](#)
- [Single Sign-On と特定のアプリケーション](#)

Single Sign-On (SSO)

SSOの詳細は、『Oracle9i Application Server セキュリティ・ガイド』の第3章「Oracle9iAS Single Sign-On の構成」を参照してください。この項では、Oracle9i Forms の開発に関連した機能について説明します。

シングル・サインオン・サポートを使用する手順は、次のとおりです。

1. URL を保護する必要があります。Forms Listener Servlet に別名を作成すると、選択したアプリケーションで SSO を使用できます。Forms URL を保護するには、次のセクションを `mod_osso.conf` ファイルに追加します。

```
<Location forms90/f90servlet>
    require valid-user
    authType Basic
</Location>
```

デフォルトの Forms URL は、"forms90/f90servlet" です。

2. OID、Delegated Administration Services (DAS) を使用してユーザーを作成し、そのユーザーにリソース情報を割り当てたり作成したりします。リソース情報には DB 接続情報が含まれ、リソース名は、`formswb.cfg` ファイルで定義された SSO 対応アプリケーション用の名前を付けたユーザー構成の名前と一致する必要があります。リソース名は、"config" 問合せパラメータの値として Oracle9iAS Forms Services URL で渡されます。
3. Oracle9i Forms アプリケーションで使用するリソース名に対応した構成セクションを作成します。たとえば、次のように指定します。

```
...
[customers]
form=customers.fmx
...
[orders]
form=orders.fmx
...
```

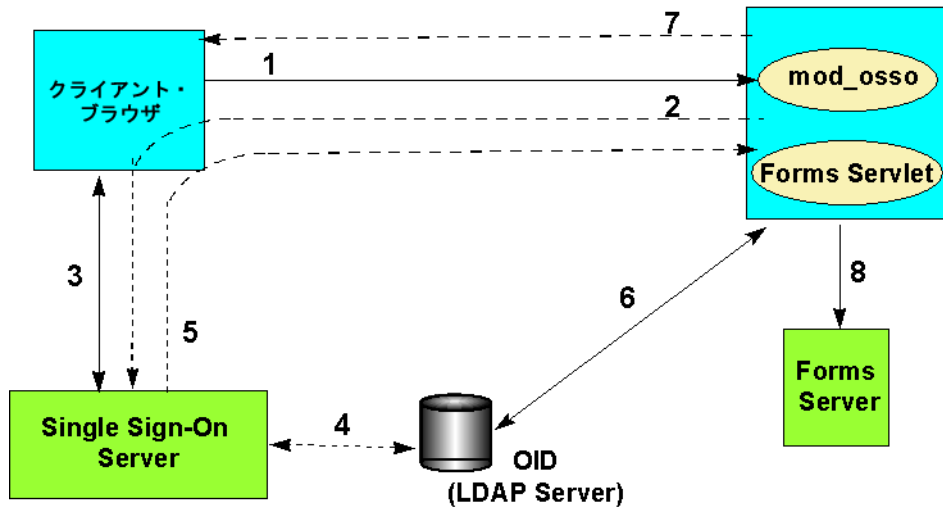
4. リソース名を選択します。たとえば、URL は `http://<server>:<port>/forms90/f90servlet?config=customers` のようになります。これにより、要求は次のように認証されます。

別の構成パラメータ（たとえば、`http://.../forms90/f90servlet?config=orders`）を使用する場合、このリソース名 "orders" を持つユーザーに構成された DB 接続情報は、DB にログインするために Forms Server で使用されます。

5. SSO のログインを追跡するためにセッション Cookie が使用されます。これらは SSO サーバーでのみ利用され、ブラウザ・セッションの終了時には失われます。

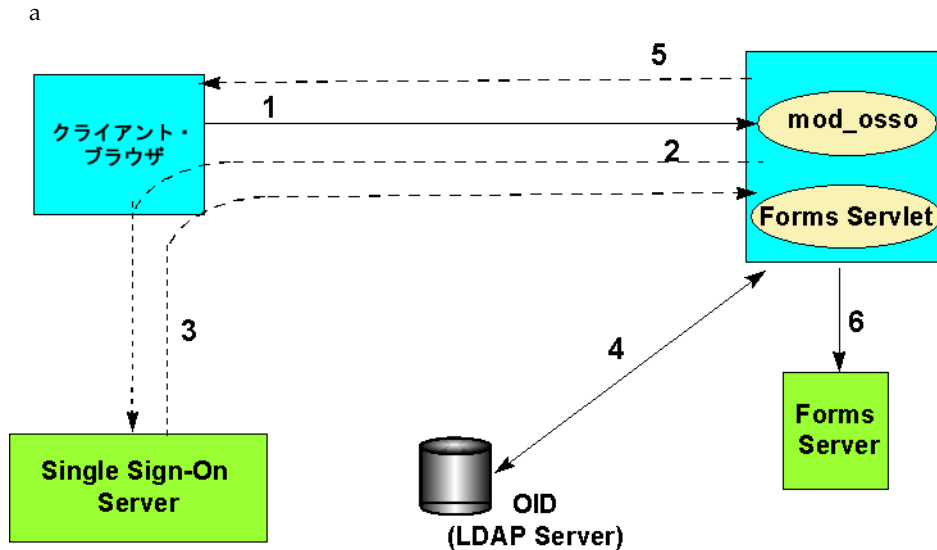
認証フロー

ユーザーが最初に Oracle9iAS Forms Services URL を要求したときの Oracle9i Forms の SSO の認証フローは、次のようになります。



1. ユーザーが、Forms URL を要求します。
2. Forms Servlet が、そのユーザーの要求を SSO サーバーにリダイレクトします。
3. ユーザーが、ログイン・フォームでユーザー名とパスワードを指定します。
4. パスワードが、OID (LDAP Server) で検証されます。
5. ユーザーの要求が、sso_userid 情報とともに URL にリダイレクトされます。
6. Forms Servlet が、OID からデータベース接続情報を取得します。
7. Forms Servlet が、ランフォーム・セッションに userid パラメータを設定し、アプレットが Forms Listener Servlet に接続されます。
8. Forms Servlet が、Forms Server を起動します。

別の Partner Application で認証されたユーザーが Oracle9iAS Forms Services URL を要求した場合、Oracle9i Forms の SSO の認証フローは、次のようになります。



1. ユーザーが、Forms URL を要求します。
2. Forms Servlet が、そのユーザーの要求を SSO サーバーにリダイレクトします。
3. ユーザーの要求が、sso_userid 情報とともに URL にリダイレクトされます。
4. Forms Servlet が、OID からデータベース接続情報を取得します。
5. Forms Servlet が、ランフォーム・セッションに userid パラメータを設定し、アプレットが Forms Listener Servlet に接続されます。
6. Forms Servlet が、Forms Server を起動します。

Single Sign-On と特定のアプリケーション

Oracle9iAS Forms Services では、シングル・サインオンと公開アプリケーションを同時に実行できます。この機能を使用するには、Forms Servlet に 2 つ目の別名を追加する必要があります。2 つ目の Forms Servlet 名は、2 つ目の Forms Services インスタンスを作成するのに必要です。

セキュリティ上の理由から、同じ formsweb.cfg ファイルで 2 つの Forms Servlet 名を共有し、その 1 つの名前を SSO に登録するような使い方は避けてください。たとえば、アプリケーション A がシングル・サインオン認証で保護されており、アプリケーション B が保護されていない場合、アプリケーション A が、SSO を使用しない 2 つ目の Forms Services インスタンスで実行されないようにする必要があります。SSO で保護されているアプリケー

ションと公開アプリケーションを区別する唯一の方法は、別々の構成ファイルを使用することです。そうすれば、アプリケーション A はアプリケーション B と同じ formsweb.cfg ファイルを共有しなくなります。

シングル・サインオンと公開アプリケーションを実行する手順は、次のとおりです。

- 2 目目のスタンドアロン・サーバー・インスタンスの作成
- Forms Servlet 別名の追加
- Oracle9i Forms 別名の mod_oc4j への登録

2 目目のスタンドアロン・サーバー・インスタンスの作成

1. OC4J web.xml ファイルにもう 1 つの Forms Servlet 別名を追加します。
2. 別名を付けた Forms Servlet を mod_oc4j に登録します。

Forms Servlet 別名の追加

Oracle9iAS Forms Services OC4J のインストール・ディレクトリ /applications/forms90app/forms90web/WEB-INF/ に移動します。テキスト・エディタで web.xml 構成ファイルをオープンし、該当する位置に次の行を追加します。

```
<servlet>
  <servlet-name>f90</servlet-name>
  <servlet-class>oracle.forms.servlet.FormsServlet</servlet-class>
  <init-param>
    <param-name>configFileName</param-name>
    <param-value><your configuration file name goes here></param-value>
  </init-param>
</servlet>
<servlet-mapping>
  <servlet-name>f90</servlet-name>
  <url-pattern>/f90*</url-pattern>
</servlet-mapping>
```

この例では、新しい Forms Servlet 名は "f90" ですが、任意の名前を付けることもできます。サーブレットの初期化パラメータ configFileName は、formsweb.cfg ファイルの名前と場所をこのサーブレットに使用します。新しい構成ファイルを作成する最適な方法は、forms90/server ディレクトリから formsweb.cfg ファイルをコピーし、そのファイルをリネームする方法です。コピーした構成ファイルを編集し、SSO がなくてはアクセスできないアプリケーションの構成をすべて削除します。前述の例の "<your configuration file name goes here>" を、コピーして作成した formsweb.cfg の場所と名前置き換えます。

Oracle9i Forms 別名の mod_oc4j への登録

新しい Forms Servlet 名を Oracle9iAS に認識させて OC4J で使用できるようにするには、サーブレットを with mod_oc4j に登録する必要があります。この登録には、Oracle9iAS

Forms Services のインストール・ディレクトリ forms90/server にある forms90.conf ファイルを使用します。

次の f90servlet と l90servlet の別名のエントリ（通常のタイプ）は、forms90.conf の最後にあります。太字の行を追加します。

OC4J の Config

```
<IfModule mod_oc4j.c>
    Oc4jMount /forms90      ProductGroup2
    Oc4jMount /forms90/f90servlet ProductGroup2
    Oc4jMount /forms90/f90servlet/* ProductGroup2
    Oc4jMount /forms90/f90    ProductGroup2
    Oc4jMount /forms90/f90/*  ProductGroup2
    Oc4jMount /forms90/l90servlet ProductGroup2
    Oc4jMount /forms90/l90servlet/* ProductGroup2
</IfModule>
```

Oracle HTTP Server を終了して再起動した後、Forms Servlet は次のパス名を使用してアクセスできます。

```
http://<hostname>:<port>/forms90/f90servlet および
http://<hostname>:<port>/forms90/f90
```

2 つのパスの違いは、forms90/f90servlet のルートはシングル・サインオンで保護され、forms90/f90 パスは異なる Forms 構成ファイル（formsweb.cfg 以外）を使用している点です。

注意：Forms Listener Servlet の "l90servlet" はどちらの構成でも使用できるため、このサーブレットに別名を作成する必要はありません。Forms Listener Servlet は formsweb.cfg ファイルから直接読み込まないため、両方のインスタンスに同じ Listener Servlet を使用してもセキュリティは影響を受けません。

次の URL を入力して、構成をテストできます。

```
http://<hostname>:<port>/forms90/f90/admin （Listener Servlet のテスト・ページが表示されます）
```

Oracle Enterprise Manager と Oracle9i Forms

Oracle9iAS には、次の 2 つの Oracle Enterprise Manager ユーザー・インタフェースが付属しています。

- HTML ベースのツール。デフォルトのブラウザから起動します。
- Java ベースのコンソール。コマンド・ラインから起動します。

注意： Oracle Enterprise Manager の起動方法の詳細は、『Oracle Enterprise Manager 構成ガイド』を参照してください。

Oracle9iAS Forms Services では、HTML ベースの Oracle Enterprise Manager を使用して次の処理を行うことができます。

- Oracle9iAS Forms Services インスタンスのメトリックを監視する。
- ユーザー・セッションのメトリックを監視する。
- ユーザー・セッションを終了する。
- Oracle9iAS Forms Services インスタンスのパラメータを構成する。

Oracle9iAS Forms Services では、Java ベースの OEM コンソールを使用して次の処理を行うことができます。

- 複数の Oracle9iAS Forms Services インスタンスのメトリックにアクセスする。
- Oracle9iAS Forms Services インスタンスの CPU およびメモリーの使用イベントを監視する。

注意： Oracle Enterprise Manager の詳細は、『Oracle9i Application Server 管理者ガイド』を参照してください。

また、Oracle Enterprise Manager を起動するときに、オンライン・ヘルプもご利用になれます。

トレースと診断

Oracle9i Forms アプリケーションを開発して配布したら、そのアプリケーションを最適化するための情報があると便利です。Oracle9i Forms に搭載されたトレース・ツールおよび診断ツールを使用すると、実行時における Oracle9i Forms アプリケーションのパフォーマンスとリソース使用状況を分析できます。トレース出力を利用すると、Oracle9i Forms アプリケーションのパフォーマンスやその他の問題を診断できます。

次のツールを使用して、Oracle9i Forms のトレース情報を収集できます。

- **Forms Trace:** Oracle9i Forms より前のリリースで利用できた Forms Runtime Diagnostics (FRD) と Performance Event Collection Services (PECS) の機能にかわるものです。Forms Trace を使用すると、フォームの使用中にユーザーが実行した手順など、フォームでの実行パスをトレースできます。
- **Servlet Logging Tools:** サイト管理者は、Oracle9i Forms のセッションをすべて記録し、Oracle9i Forms 関連のネットワークの通信量を監視し、サイトの構成問題をデバッグできます。
- **Oracle Trace:** Oracle Database、Oracle9iAS Forms Services および Oracle Trace API をトレース用に実装したその他の製品のパフォーマンスを収集し、分析するツールです。

この章では、トレース・ツールについて説明します。次の項で構成されます。

- **Forms Trace**
- **Servlet Logging Tools**
- **Oracle Trace**

Forms Trace

Forms Trace を使用すると、正確に定義されたフォーム機能の部分やユーザー・アクションに関する情報を記録できます。これを行うには、トレース情報を収集する必要があるイベントを定義します。たとえば、トリガーの実行やマウス・クリック、あるいはその両方に関する情報を記録できます。

この項では、次の項目に関する情報を提供します。

- [Forms Trace の構成](#)
- [トレースの開始](#)
- [Forms Trace 出力の表示](#)
- [トレース可能イベントのリスト](#)
- [イベントの詳細のリスト](#)

Forms Trace の構成

トレースの開始時に、`ftrace.cfg` ファイルまたは URL で、トレースを行う必要があるイベントを定義します。イベントは、ユーザー・アクションの直接または間接的結果として Oracle9i Forms 内に起こる事象を意味します。イベントとそれに対応するイベント番号は、「[トレース可能イベントのリスト](#)」を参照してください。

`ftrace.cfg`

`ftrace.cfg` ファイルは、デフォルトで `<ORACLE_HOME>/forms90/trace` ディレクトリにインストールされ、`FORMS90_TRACE_PATH` 環境変数は `default.env` ファイルに設定されています（`FORMS90_TRACE_PATH` 環境変数では、`ftrace.cfg` ファイルの場所とトレース出力ファイルの場所を指定します）。

テキスト・エディタで `ftrace.cfg` 構成ファイルを編集し、イベント・セットを名前指定します。イベント・セットは、トレース開始ごとの個別のイベント番号ではなく、イベント・セット名を指定するだけでトレースを行うイベントのセットを指定します。

次の例では、2つのイベント・セットを指定した `ftrace.cfg` 構成ファイルを示しています。

```
//  
// example ftrace.cfg file  
// This file is used to specify event groups for use with Forms Trace  
//  
// The file format is  
// name1: event1, event2, ... , event_n  
// name2: event1-event_m  
//  
all: 0-199  
errors: 0-3  
custom1: 32-46, 65, 66, 96, 194
```

次の点に注意してください。

- キーワードのエントリ間には空白行を入れる必要があります。
- キーワードには、空白を含まなければ任意の名前を付けることができます。たとえば、a_b_c というキーワードを使用できます。
- イベント番号の間はカンマで区切る必要があります。

トレースの開始時にコマンド・ラインに `tracegroup = "custom1"` と指定すると、これは `tracegroup = "32-46, 65, 66, 96, 194"` の指定に相当します。

URL パラメータ・オプション

次のコマンド・ライン・パラメータを使用すると、Forms Trace を構成できます。

Record = forms | otrace

Tracegroup =

Log = <filename>

表 7-1 Forms Trace のコマンド・ライン・パラメータ

パラメータ	値	説明
Record	forms	Forms Trace を使用可能にします。
	otrace	Oracle Trace の統合を有効にします。

表 7-1 Forms Trace のコマンド・ライン・パラメータ（続き）

パラメータ	値	説明
Tracegroup		<p>どのイベントを記録およびログ出力するかを特定します。</p> <ul style="list-style-type: none">■ Tracegroup を指定していない場合、エラー・メッセージのみが収集されます。■ Forms Trace をコマンド・ラインで起動しない場合、Tracegroup は無視されます。■ Tracegroup キーワードを使用して名前を付けたイベントのセットを作成できます。たとえば、次のように指定します。 Tracegroup=<keyword>。この<keyword> は、fttrace.cfg で指定します（たとえば、Tracegroup=MyEvents）。 これで、名前を付けたセット SQLInfo のイベントをログ出力できます。■ Tracegroup キーワードを使用して指定した範囲のすべてのイベントをログ出力できます。たとえば、次のように指定します。 Tracegroup = 0-3 これで、0<= イベント <=3 で定義された範囲のすべてのイベントをログ出力できます。■ Tracegroup キーワードを使用して個々のイベントをログ出力できます。たとえば、次のように指定します。 Tracegroup = 34,67■ Tracegroup キーワードを使用してイベント・セットを組み合わせたことができます。たとえば、次のように指定します。 Tracegroup = 0-3,34,67,SQLInfo

表 7-1 Forms Trace のコマンド・ライン・パラメータ (続き)

パラメータ	値	説明
Log		<p>トレース情報の保存場所を指定します。</p> <p>ディレクトリを指定しない場合、ファイルは FORMS90_TRACE_PATH 環境変数で指定されたディレクトリに保存されます (この変数が設定されていない場合、出力は現在作業中のディレクトリに書き込まれます)。</p> <p>ログ・ファイルを指定していない場合、ユーザー・プロセスのプロセス ID (pid) がトレース・ファイルの名前として使用されます (例: forms_<pid>.trc)。</p>

トレースの開始

トレースを開始するには、URL または formsweb.cfg ファイルにトレース・エントリを指定します。エントリには、収集するイベントのグループとトレース・ファイル名を含める必要があります。トレースの収集は、フォームの実行時に開始されます。

次の例は、トレースを開始する URL のサンプルです。

```
http://cx-pc/forms90/f90servlet?form=cxl&record=forms&tracegroup=0-199
http://cx-pc/forms90/f90servlet?form=cxl&record=forms&tracegroup=mysql
http://cx-pc/forms90/f90servlet?form=cxl&record=forms&tracegroup=0-199;log=run1.log
```

Oracle9i Forms の今後のリリースでは、ビルトインを使用してトレースを開始するメソッドが実装されます。最新のドキュメント、ホワイトペーパー、ビューレット・デモンストレーションなどの Oracle9i Forms に関する最新情報は、OTN (<http://otn.oracle.com> 英語サイト) または OTN-Japan (<http://otn.oracle.co.jp> 日本語サイト) を参照してください。

Forms Trace 出力の表示

トレース・データは、*.trc 拡張子の付いたバイナリ形式ファイルで保存されます。トレース・データを表示するには、次のいずれかの方法を実行します。

- Upload/Translate ユーティリティを使用して *.trc ファイルのデータを XML 形式に変換し、XML ビューアを使用してそのデータを表示します。
- Upload/Translate ユーティリティを使用して、*.trc ファイルのデータをデータベース表にアップロードします (表を作成するスクリプトの詳細は、「[トレース・データのデータベース表の作成](#)」を参照してください)。

Upload/Translate ユーティリティの実行

Upload/Translate ユーティリティは、次の機能を実行します。

- トレース・データを XML 形式に変換する。
- トレース・データをデータベース表にアップロードする。

トレース・データを XML 形式に変換するには、コマンド・ラインで次のように入力します。

```
java oracle.forms.diagnostics.Xlate datafile=a.trc xmlfile=myfile.xml
```

これで、myfile.xml が作成されました。

トレース・データをデータベース表にアップロードするには、コマンド・ラインで次のように入力します。

```
java oracle.forms.diagnostics.Xlate datafile=a.trc userid=[password]=pw  
url=[server]:1521:orcl9i
```

これで、トレース・データが表にアップロードされました。

ユーティリティに有効なコマンド・ラインのオプションを次に示します。

表 7-2 Translate ユーティリティのコマンド・ライン・オプション

パラメータ	説明
DataFile	バイナリ・トレース・データ収集ファイルの名前を指定します。
XMLFile	必要に応じて、XML 出力ファイルの名前を指定します。
Url	フォームのデータベース接続を指定します (hostname:port:oracle_sid)。
UserName	データベースのユーザー名を指定します。
Password	データベースのパスワードを指定します。
CollectionId	トレースに一意的識別子を指定します。指定しない場合は、デフォルトの "0" が使用されます。
Regressflag	出力ファイルをテスト用に使用するかどうかを指定します。 TRUE に設定した場合、タイプスタンプ情報が抑制され、実行全体にわたって収集された詳細を比較できます。

トレース・データのデータベース表の作成

トレース・データ用の表を作成するためのスクリプトが提供されています。

バイナリのトレース・ファイルで frmtrc.sql スクリプトを実行し、データベースの表、インデックスおよびビューを作成します。frmtrc.sql スクリプトは、frmtrc.tab スクリプト（表の

作成)、frmtrc.con スクリプト (絞込み条件の作成)、frmtrc.vw スクリプト (ビューの作成) を呼び出します。

これらのスクリプトは、ユーザーのトレース・データ用に次のようなデータベース・スキーマを作成します。

表 7-3 Forms トレース・データ用のデータベース・スキーマ

表	列	属性の説明
Collections	Collection_Id	コレクションを識別する一意の ID。
	Collection_Name	トレース収集ファイルの名前。
	Upload_Date	データがアップロードされる日付。
	Trace_Date	データが収集される日付。
	User_IP_Address	接続が行われる IP アドレス。
	PID	ランフォーム・プロセスのプロセス ID。
User_Actions	Action_Id	コレクション内でアクションを識別するシーケンス番号。
	Module_Module_Id	モジュールに対する外部キーとして使用するモジュール ID。
	Collection_Collection_Id	アクションに対応するコレクション。データを関連付ける外部キーとして使用されます。
	Action_Context	アクションが実行されるコンテキスト。
	Action_Name	ユーザー・アクション。
	TimeStamp	アクションの出現時間。
User_Action_Details	Usr_Action_Id	属性に対応するアクション。データを関連付ける外部キーとして使用されます。
	Collection_Collection_Id	アクションに対応するコレクション。データを関連付ける外部キーとして使用されます。
	Attribute_Name	アクションの詳細な属性名。
	Attribute_Value	アクションの詳細な属性値。
Events	Event_ID	収集イベントのシーケンスを識別するイベント識別子。
	Collection_Collection_Id	イベントに対応するコレクション。
	Usr_Action_Id	User_Actions に対する外部キー。

表 7-3 Forms トレース・データ用のデータベース・スキーマ（続き）

表	列	属性の説明
	Event_Event_Id	関連イベントを識別します。期間イベントに有効です。
	Event_Context	イベントが発生したコンテキスト。
	Event_Name	イベントの名前。
	Event_Number	特定のイベントに割り当てられた番号（たとえば、0-3 がエラーなど）。
	Timestamp	イベントが出現した時を示すタイムスタンプ。
	Duration	イベントが完了するまでに要した時間。期間イベントのみに有効です。
Event_Details	Detail_Id	詳細 ID。
	Collection_Collection_ID	イベントに対応するコレクション。
	Event_Event_Id	詳細に対応するイベント ID。
	Attribute Name	属性の名前。
	AttributeValue	属性の値。
Modules	Module_Id	モジュールの内部 ID。主キー。
	Collection_Collection_Id	コレクションに対する外部キー。
	Module_Name	モジュールの名前（たとえば、my.fmx など）。
	Physical_Path	フォームへの物理パス（たとえば、/temp/my.fmx など）。
	Attached_Libraries	フォームに連結したライブラリの名前。
	Attached_Menus	フォームに連結したメニュー。

トレース可能イベントのリスト

トレースに定義できるイベントを次の表に示します。Forms の今後のリリースでは、さらに多数のイベントがこのリストに追加される予定です。

イベント型には、次のものがあります。

- ポイント・イベント: 識別可能な継続時間がないユーザー・アクションや内部信号の結果として Oracle9i Forms に発生したイベント（ステータス行のエラー・メッセージの表示など）。このイベント型のインスタンスのたびに、ログ・ファイルに 1 つのエントリが作成されます。

- 期間イベント：トリガーのように開始と終了があるイベント。このイベント型のインスタンスのたびに、ログ・ファイルに 1 組のエントリ（開始と終了のイベント）が作成されます。
- ビルトイン・イベント：ビルトインに関連したイベント。このイベント型のインスタンスのたびに、イベントに関する様々な情報（たとえば、引数の値など）が作成されます。

表 7-4 トレース可能イベントのリスト

イベント番号	定義	型
0	異常エラー	ポイント
1	フォーム・オープン時のエラー	ポイント
2	Forms の使用不能エラー	ポイント
3	ステータス・バー上のエラー・メッセージ	ポイント
4-31	予約済	
32	起動	ポイント
33	メニュー	ポイント
34	キー	ポイント
35	クリック	ポイント
36	ダブルクリック	ポイント
37	値	ポイント
38	スクロール	ポイント
39	LOV 選択	ポイント
40	未使用	未使用
41	ウィンドウのクローズ	ポイント
42	ウィンドウのアクティブ化	ポイント
43	ウィンドウの非アクティブ化	ポイント
44	ウィンドウのサイズ変更	ポイント
45	タブ・ページ	ポイント
46	タイマー	ポイント
47	予約済	
48	予約済	

表 7-4 トレース可能イベントのリスト（続き）

イベント番号	定義	型
49-63	予約済	
64	フォーム（開始と終了）	期間
65	プロシージャ（開始と終了）	期間
66	トリガー（開始と終了）	期間
67	LOV（開始と終了）	期間
68	エディタの起動	ポイント
69	キャンバス	ポイント
70	警告	期間
71	ファイルの取得	ポイント
72-95	予約済	
96	ビルトイン（開始と終了）	ビルトイン
97	ユーザー・イグジット（開始と終了）	期間
98	SQL（開始と終了）	期間
99	メニューの作成（開始と終了）	期間
100	PLSQL（開始と終了）	期間
101	問合せの実行	期間
102-127	予約済	
128	クライアントの接続	ポイント
129	クライアントのハンドシェイク	ポイント
130	ハートビート	ポイント
131	HTTP の再接続	ポイント
132	ソケット（開始と終了）	期間
133	HTTP（開始と終了）	期間
134	SSL（開始と終了）	期間
135	DB 処理（開始と終了）	期間
136	DB ログオン（開始と終了）	期間

表 7-4 トレース可能イベントのリスト（続き）

イベント番号	定義	型
137	DB ログオフ（開始と終了）	期間
138-159	予約済	
160-191	予約済	
192*	環境ダンプ	該当せず
193*	状態変化	該当せず
194*	ビルトイン引数	該当せず
195*	ユーザー・イグジット引数	該当せず
196*	プロシージャ引数	該当せず
197*	関数の引数	該当せず
256 以上	ユーザー定義	
1024 以上	予約済	

* これらのイベント番号にイベント型がないのは、実際のイベントではなくイベントの詳細であるためです。たとえば、状態変化は、実際のアクションまたはイベントによって発生したことを確認するために選択できます。

イベントの詳細のリスト

トレースに定義できるイベントの詳細を次の表に示します。

- [エラー](#)
- [ユーザー・アクション・イベント](#)
- [Forms Services のイベント](#)
- [イベントの詳細](#)
- [3 層イベント](#)
- [その他](#)

エラー

エラーのタイプとエラーの詳細をリストします。

ユーザー・アクション・イベント

表 7-5 ユーザー・アクション・イベントの詳細

アクション	詳細	番号
メニュー選択	メニュー名、選択内容	33
キー	キー・プレス、フォーム、ブロック、項目	34
クリック	マウス / キー、フォーム、ブロック、項目	35
ダブルクリック	フォーム、ブロック、項目	36
値	フォーム、ブロック、項目	37
スクロール	フォーム、上、下、ページ、行	38
LOV 選択	LOV 名、選択項目	39
警告	警告名、選択内容	40
タブ	フォーム	45
ウィンドウのアクティブ化、非アクティブ化、クローズ、サイズ変更	ウィンドウ名、フォーム名、サイズ	41,42,43,44

Forms Services のイベント

表 7-6 Forms Services のイベントの詳細

イベント名	詳細	番号
フォーム	フォーム ID、名前、パス、連結ライブラリ、連結メニュー	64
プロシージャ	プロシージャ名、フォーム ID	65
トリガー	トリガー名、フォーム名、ブロック名、項目名、フォーム ID	66
LOV	LOV 名、フォーム ID	67
エディタ	フォーム ID、エディタ名	68
キャンパス	フォーム ID、キャンパス名	69

イベントの詳細

表 7-7 イベントの詳細

イベント名	詳細	番号
ビルトイン	ビルトイン名、フォーム ID	96
ユーザー・イグジット	ユーザー・イグジット名、フォーム ID	97
メニューの作成	メニュー名、フォーム ID	99
PLSQL	PLSQLSTmt、フォーム ID	100
問合せの実行	ブロック名	101

3 層イベント

表 7-8 3 層イベントの詳細

イベント名	詳細	番号
クライアントの接続	タイムスタンプ	128
クライアントのハンドシェイク	タイムスタンプ	129
ハートビート	タイムスタンプ	130
HTTP の再接続		131
ソケット	フォーム ID、パケット、バイト	132
HTTP	フォーム ID、パケット、バイト	133
HTTPS	フォーム ID、パケット、バイト	134
DB 処理	フォーム ID、文	135
DB ログオン	フォーム ID	136
DB ログオフ	フォーム ID	137

その他

表 7-9 その他のイベントの詳細

イベント名	詳細	番号
環境ダンプ	選択した環境情報	192
状態変化	最後のアクション / イベントにより生じた内部状態の変化	193
ビルトイン引数	ビルトインに対する引数の値	194
ユーザー・イグジット引数	ユーザー・イグジットに渡される引数	195
プロシージャ引数	プロシージャに渡される引数（イン / アウト）	196
関数の引数	プロシージャに渡される引数（イン / アウト）	197

Servlet Logging Tools

Oracle9iAS Forms Services で使用できる Servlet Logging Tools では、次のような内容が提供されます。

- セッションの開始時間と終了時間、ユーザーの IP アドレス、ホスト名を含む Oracle9i Forms のすべてのセッションの記録（セッション・レベルのロギング）
- Oracle9i Forms 関連のネットワークの通信量とパフォーマンスの監視（セッション・パフォーマンスと要求パフォーマンス・レベルのロギング）
- サイトの構成問題のデバッグ情報（デバッグ・レベルのロギング）

この項では、次の項目に関する情報を提供します。

- [ロギングの起動](#)
- [ログ・ファイルのディレクトリ](#)
- [各レベルのサーブレット・ロギングの出力例](#)

ロギングの起動

ロギングは、次のようにして起動します。

- [表 7-10「サポートされているロギング機能」](#)の文字列の 1 つを、フォームを開始する URL の serverURL パラメータに追加します。
- [表 7-10「サポートされているロギング機能」](#)の文字列の 1 つを、formsweb.cfg ファイルの serverURL クライアント・パラメータに追加します。

ロギングを起動すると、Listener Servlet がログ・メッセージをサーブレットのログ・ファイルに書き込みます。各ロギング・レベルにおける出力例は、「各レベルのサーブレット・ロギングの出力例」に記載されています。

表 7-10 サポートされているロギング機能

serverURL クライアント・パラメータに追加する文字列	ロギングの説明
(なし)	ログ・メッセージは生成されません。ただし、Forms Servlet の初期化時に、使用する構成ファイルの名前とパスを示すメッセージがログ・ファイルに書き込まれます。
/session	Forms セッションが開始または終了するときには、必ずログ・メッセージが書き込まれます。このメッセージには、クライアント（ユーザーの Web ブラウザが稼動しているマシン）のホスト名と IP アドレス、ランタイム・プロセス ID、および一意の内部セッション ID 番号が含まれます。
/sessionperf	セッション終了メッセージにパフォーマンス・サマリー統計が含まれます。
/perf	クライアントからの要求すべてに、パフォーマンス・メッセージが書き込まれます。
/debug	すべてのデバック・メッセージ。前述のメッセージを含むその他のデバック・メッセージが書き込まれます。このロギング・レベルは情報量が多く、主にデバッグやサポート用に使用します。

URL でのロギングの指定

たとえば、パフォーマンス・レベルのトレースを行うには、次のような URL を使用して Oracle9i Forms アプリケーションを起動します。

```
http://yourserver/forms90/f90servlet?serverURL=/forms90/l90servlet/perf
```

formsweb.cfg ファイルでのロギングの指定

たとえば、すべてのユーザーでセッション・レベルのロギングを開始するには、次のように formsweb.cfg ファイルのデフォルト・セクションで serverURL エントリを変更します。

```
serverURL=/forms90/l90servlet/session
```

Forms Servlet の起動に使用した URL での完全診断の指定

たとえば、完全診断を開始するには、次のような URL を使用して Oracle9i Forms アプリケーションを起動します。ただし、Forms Servlet の起動に使用した URL に /debug を追加する場合、サーブレットはデバック・メッセージをログ・ファイルにも出力します。

```
http://yourserver/forms90/f90servlet/debug?serverURL=/forms90/l90servlet/debug
```

ログ・ファイルのディレクトリ

サーブレットのログ・ファイルは `application.log` です。これは、Forms を配置する OC4J インスタンスの `application-deployments/forms90app` ディレクトリに書き込まれます。

Oracle9iAS では、フル・パス名は次のとおりです。

```
<ORACLE_HOME>/j2ee/OC4J_BI_Forms/application-deployments/forms90app/OC4J_BI_Forms_
default_island/application.log
```

Oracle9iDS では、次のとおりです。

```
<ORACLE_HOME>/j2ee/Oracle9iDS/application-deployments/forms90/application.log
```

各レベルのサーブレット・ロギングの出力例

次のロギング・レベルを使用した場合の各種の出力例を示します。

- (なし)
- `/session`
- `/sessionperf`
- `/perf`
- `/debug`

(なし)

```
FormsServlet init():
configFileName:      d:\orant9i/forms90/server/formsweb.cfg
testMode:
    false
```

`/session`

セッション開始メッセージ (例)

```
Forms session <10> started for test-pc.mycompany.com ( 138.56.98.72 )
Forms session <10> runtime process id = 373
```

セッション終了メッセージ (例)

```
Forms session <10> ended
```

`/sessionperf`

```
Forms session <3> started for test-pc.mycompany.com ( 138.56.98.72 )
Forms session <3> runtime process id = 460
```



```
Forms session <3> ended
  Total duration of network exchanges: 1.041
  Total number of network exchanges: 2 (1 "long" ones over 1.000 sec)
  Average time for one network exchange (excluding long ones): 0.030
  Total bytes: sent 1,110, received 316
```

/perf

```
Forms session <3> started for test-pc.mycompany.com ( 138.56.98.72 )
Forms session <3> runtime process id = 460
Forms session <3>: request processed in 1.011 sec. Received 8 bytes, returned 8
bytes.
Forms session <3>: request processed in 0.030 sec. Received 308 bytes, returned
1,102 bytes.
Forms session <3> ended
  Total duration of network exchanges: 1.041
  Total number of network exchanges: 2 (1 "long" ones over 1.000 sec)
  Average time for one network exchange (excluding long ones): 0.030
  Total bytes: sent 1,110, received 316
```

/debug

<http://test-machine:8888/forms90/f90servlet/debug&config=ienative&serverURL=/forms90/190servlet/debug> の URL を使用した実行例を次に示します。

```
===== FormsServlet =====
GET request received, cmd=debug,
qstring=config=ienative&serverURL=/forms90/190servlet/debug
No current servlet session
File baseie.htm not found, looking in d:\orant9i\forms90\server
The SSO_USERID is: null
===== FormsServlet =====
GET request received, cmd=startsession, qstring=config=ienative&serverURL=
/forms90/190servlet/debug&ifcmd=startsession
No current servlet session
New servlet session started
SSO_USERID in startSession: null
SSO_AuthType in startSession: null
User DN: null
Subscriber DN: null
EM mode in the config file: 0
File default.env not found, looking in d:\orant9i\forms90\server
envFile = d:\orant9i\forms90\server\default.env
serverURL: /forms90/190servlet/debug
rewrittenURL: /forms90/190servlet/debug;jsessionid=27f6412da05c
426ab47db4ae77636113
===== ListenerServlet =====
GET request received, cmd=getinfo,
```

```
qstring=ifcmd=getinfo&ifhost=test-pc.mycompany.com&ifip=130.35.96.71
Existing servlet session, id = 27f6412da05c426ab47db4ae77636113, not from cookie
Creating new Runtime Process using default executable
Starting Forms Server in EM mode
startProcess: executing ifweb90 server webfile=HTTP-0,0,1
Getting stdin, stdout and stderr of child process
Writing working directory to stdin: d:\orant9i\forms90
New server process created
Forms session <4> started for test-pc.mycompany.com ( 138.56.98.72 )
*****
Got POST request, length = 8
HTTP request headers:
    ACCEPT-LANGUAGE: en
    PRAGMA: 1
    CONTENT-TYPE: application/x-www-form-urlencoded
    ACCEPT: text/html, image/gif, image/jpeg, */*; q=.2, */*; q=.2
    USER-AGENT: Mozilla/4.0 (compatible; MSIE 5.0; Win32)
    HOST:test-machine:8888
    CONTENT-LENGTH: 8
    CONNECTION: Keep-Alive
Existing servlet session, id = 27f6412da05c426ab47db4ae77636113, not from cookie
Forms session <4> runtime process id = 474
Port number is 2791
RunformProcess.connect(): connected after 1 attempts
Connected to ifweb process at port 2791
Forms session <4>: request processed in 1.032 sec. Received 8 bytes,
returned 8 bytes.
*****
```

Oracle Trace

Oracle Trace は、Oracle Enterprise Manager の一部です。Oracle Trace は、Oracle9i Database、Oracle9iAS Forms Services、および Oracle Trace API をトレース用に実装したその他の製品のパフォーマンスを収集し、分析するツールです。Oracle Trace を最大限に活用するには、Oracle Enterprise Manager 製品一式に含めて提供されるオプション・パックの1つである、Diagnostics Pack をインストールする必要があります。Diagnostics Pack には、Oracle Trace による収集を GUI インタフェースでリモート管理し、Oracle Trace の出力を効率的に表示するための一連のツールが含まれています。

詳細は、Oracle Enterprise Manager のマニュアルを参照してください。

パフォーマンス・チューニングに関する 考慮事項

概要

この章では、Oracle9iAS Forms Services を使用してアプリケーションを配布する際のチューニングの考慮事項について説明します。この章では、アプリケーション・サーバーのネットワークとリソースについて説明します。次の項で構成されます。

- [Forms Services のビルトイン最適化機能](#)
- [Oracle9iAS Forms Services アプリケーションのチューニング](#)

Oracle9iAS Forms Services とデータベース・サーバー間の接続のチューニングについては、ここでは説明しません。

Forms Services のビルトイン最適化機能

Oracle9iAS Forms Services および Java クライアントには、いくつかの最適化機能が含まれており、大きく次の項目に分類できます。

- クライアント・リソース要件の最小化
- Forms Services リソース要件の最小化
- ネットワーク使用量の最小化
- ネットワークを介して送信されるパケットの効率の拡大
- クライアントでのアプリケーション画面の効率的なレンダリング

クライアント・リソース要件の最小化

Java クライアントは、主にアプリケーション画面のレンダリングを行います。Java クライアントには、埋込みアプリケーションのロジックはありません。Java クライアントをロードすると、複数のフォームを同時に表示できます。Oracle9i Forms のすべてのアプリケーションに汎用 Java クライアントを使用すると、アプリケーションごとにカスタマイズされた Java クライアントよりも、クライアント上のリソースが少なく済みます。

Java クライアントは、多くの Java クラスで構成されています。これらのクラスは、スプラッシュ画面の表示、ネットワーク通信およびロック・アンド・フィールの変更などの、機能サブコンポーネントにグループ化されます。機能サブコンポーネントを使用すると、Oracle9i Forms Developer および Java 仮想マシン (JVM) は、すべての機能クラスを一度にダウンロードしなくても、必要に応じて機能をロードできます。

Forms Services リソース要件の最小化

フォーム定義が FMX ファイルからロードされる時、実行プロセスのプロファイルは次のものに要約できます。

- 暗号化されたプログラム単位
- ボイラープレート・オブジェクト / イメージ
- データ・セグメント

これらの中で、データ・セグメント・セクションのみがアプリケーションの指定したインスタンスに対して一意です。暗号化されたプログラム単位とボイラープレート・オブジェクト / イメージは、どのアプリケーション・ユーザーにも共通しています。Oracle9iAS Forms Services では、共有コンポーネントを物理メモリーにマップし、同じ FMX ファイルにアクセスするすべてのプロセス間でこれを共有します。

指定した FMX ファイルをロードする最初のユーザーは、そのフォームに必要な全メモリー量を使用します。ただし、後続のユーザーの場合は必要なメモリー量が大幅に減らされているので、ローカル・データのエクステンツにのみ依存します。共有コンポーネントをマップ

するこのメソッドを使用すると、指定したアプリケーションに必要な、ユーザーごとの平均メモリー量を減らすことができます。

ネットワーク使用量の最小化

帯域幅は重要なリソースで、インターネット・コンピューティングの一般的な広がりとともに、インフラストラクチャにますます大きな負担を強いるようになっていきます。このため、アプリケーションはネットワークの容量を節約して使用することが重要です。

Oracle9iAS Forms Services は、メタデータ・メッセージを使用して Java クライアントと通信します。メタデータ・メッセージは、実行対象のオブジェクトとその実行方法をクライアントに通知する名前と値のペアのコレクションです。パラメータのみを Java クライアント上の汎用オブジェクトに送信することで、(同じ効果になるよう新規コードを送信した場合と比較して) 通信量を約 90% 減らすことができます。

Oracle9iAS Forms Services では、データ・ストリームを次の 3 つの方法で効率的に圧縮します。

- 同じようなメッセージの集合 (名前と値のペアのコレクション) を送信すると、2 番目以降のメッセージには、前のメッセージとの相違点のみが含まれます。この結果、ネットワーク・トラフィックを大幅に減らすことができます。このプロセスは、差分メッセージングと呼ばれます。
- 同じ文字列がクライアント画面で繰り返されると (たとえば、同じ企業名が記載されている複数行のデータが表示される場合)、Oracle9iAS Forms Services はその文字列を一度のみ送信し、後続のメッセージではその文字列を参照します。参照によって文字列を渡すことで、帯域幅の効率は向上します。
- データ・タイプはその値に必要な最小のバイト数で送信されます。

ネットワークを介して送信されるパケットの効率の拡大

待ち時間は、アプリケーションの応答時間に影響を与える最も重要な要因です。待ち時間の影響をなるべく受けないようにする最もよい方法の 1 つは、Java クライアントと Forms Server 間で、対話中に送信されるネットワーク・パケットの数を最小限にすることです。

Oracle9i Forms Developer モデル内のトリガーを多数使用すると大きな効果がありますが、各トリガーにネットワークの往復が必要なため、待ち時間の影響が大きくなります。トリガーに関連する待ち時間を避ける方法の 1 つは、イベント・バンドルを介してトリガーをグループ化することです。たとえば、ユーザーが項目 A から項目 B にナビゲートする場合 (あるエントリ・フィールドから別のフィールドにタブする場合など)、トリガー実行前後の範囲には、それぞれ Forms Server 上での処理が必要です。

イベント・バンドルは、2 つのオブジェクト間をナビゲートしている間にトリガーされたすべてのイベントを集めて、それらを単一のパケットとして Oracle9iAS Forms Services に配布して処理します。ナビゲートに多くのオブジェクトの間合せが関連している場合 (離れているオブジェクト上でマウスのクリックを行った場合など)、イベント・バンドルは問い合わせ

されたすべてのオブジェクトからすべてのイベントを集めて、そのグループを単一のネットワーク・メッセージとして Oracle9iAS Forms Services に配布します。

クライアントでのアプリケーション画面の効率的なレンダリング

指定したフォーム内のすべてのボイラープレート・オブジェクトは仮想グラフィック・システム (VGS) ツリーの一部です。VGS は、すべての Oracle9i Forms Developer 製品に共通の図形サブコンポーネントです。VGS ツリー・オブジェクトは、座標、カラー、線幅およびフォントなどの属性を使用して記述します。オブジェクトの VGS ツリーを Java クライアントに送信する場合、送信する属性のみが指定したオブジェクト・タイプのデフォルトと異なる属性になります。

イメージは圧縮された JPEG イメージとして送信および格納されます。これにより、ネットワーク・オーバーヘッドとクライアントの必要なメモリー量の両方を減らすことができます。

リソースの最小化には、クライアントおよびサーバー・プロセスのメモリー・オーバーヘッドの最小化も含まれます。ネットワークを最適な状態で使用するには、帯域幅を最小に維持し、ネットワークの待ち時間の影響も含まれるため、クライアントおよび Oracle9iAS Forms Services 間の通信で使用するパケット数を最小化することが必要です。

Oracle9iAS Forms Services アプリケーションのチューニング

アプリケーションの開発者は、Forms Server のビルトイン・アーキテクチャの最適化機能により最大の利益を得ることができます。この章の後半では、多くのアプリケーションに影響を与える主要なパフォーマンスの問題、および開発者がアプリケーションをチューニングしてパフォーマンスを改善し、Forms Server 機能を活用するための方法について説明します。

取り上げる内容は次のとおりです。

- [データ・サーバーに対する Oracle9iAS Forms Services の位置](#)
- [アプリケーションの起動時間の最小化](#)
- [必須ネットワーク帯域幅の削減](#)
- [パフォーマンスを改善するためのその他の方法](#)

データ・サーバーに対する Oracle9iAS Forms Services の位置

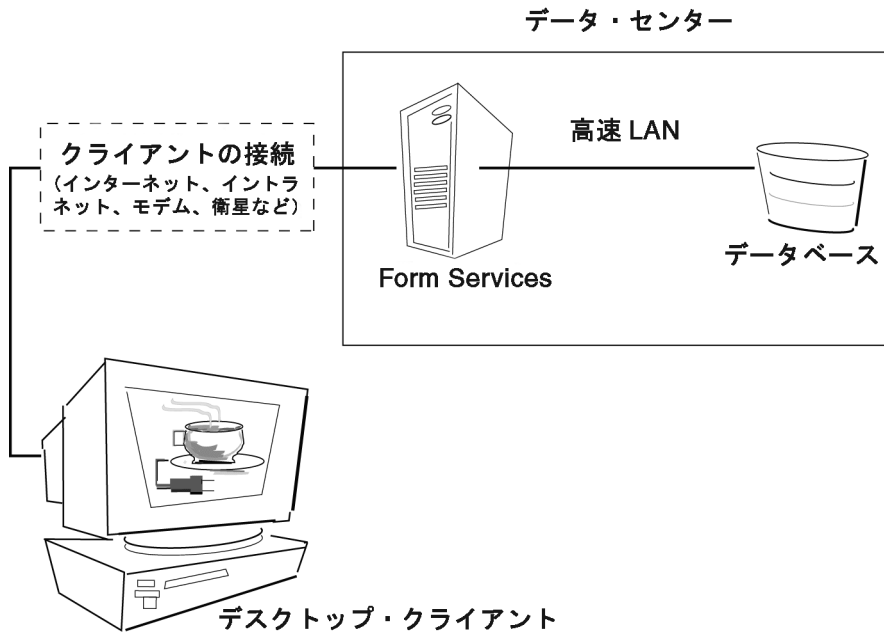
Forms Java クライアントは、GUI オブジェクトを表示する役割のみを担います。Oracle9i Forms のすべてのロジックは、中間層の Oracle9iAS Forms Services で実行されます。これには、データベースへのデータの挿入や更新、データベースからのデータの間合せ、データベースでのストアド・プロシージャの実行などが含まれます。そのため、アプリケーション・サーバーとデータベース・サーバー間の高速接続が重要になります。

このサーバ間の対話は、すべて Forms Java クライアントと通信することなく行われます。画面に変更があった場合のみ、クライアントと Oracle9iAS Forms Services 間で通信が行わ

れます。これにより、Oracle9i Forms アプリケーションは、モデムや衛星を介した低速ネットワークで実行することが可能です。

図 8-1 の構成では、Oracle9iAS Forms Services とデータベース・サーバーがデータ・センター内で並存する仕組みを示しています。

図 8-1 Oracle9iAS Forms Services とデータベース・サーバーの並存



アプリケーションの起動時間の最小化

アプリケーションをロードするためにかかる時間は、第一印象として重要であり、またどのユーザーにとっても主要な基準となります。起動時間は、オーバーヘッドとみなされます。起動時間は、今後のパフォーマンスを期待させるものでもあります。業務でクライアント・テクノロジーを使用する場合、クライアント・コードのロードに必要な追加のオーバーヘッドは、ユーザーにあまりよい影響を与えません。したがって、可能な限りロード時間を最小化することが重要です。

Oracle9i Forms アプリケーションを要求した後、アプリケーションを使用できる状態にするまでに、いくつかの手順を実行する必要があります。

1. Java 仮想マシン (JVM) を起動します。

2. すべての初期 Java クライアント・クラスをロードし、クラスのセキュリティを認証します。
3. スプラッシュ画面を表示します。
4. フォームを次に示す方法で初期化します。
 - a. 必要に応じて、追加の Java クラスをロードします。
 - b. クラスのセキュリティを認証します。
 - c. ボイラープレート・オブジェクトとイメージをレンダリングします。
 - d. 初期画面ですべての要素をレンダリングします。
5. スプラッシュ画面を削除します。
6. フォームを使用できます。

アプリケーション開発者は、JVM を起動するのにかかる時間についてほとんど何も行うことができません。ただし、Java 配置モデルおよび Oracle9i Forms Developer Java クライアントの構造では、開発者はどの Java クラスをどのようにロードするかを決定できます。これにより、Java クラスに必要なロード時間を最小化します。

Java クライアントには、基本機能のクラス（ウィンドウを開くなど）と特定の表示オブジェクトの追加クラス（LOV 項目など）のコア集合が必要です。これらのクラスはサーバーに始めから常駐している必要がありますが、次の方法を使用すると、これらのクラスをクライアントの JVM にロードするのに必要な時間を短縮できます。

- [Java ファイルの使用](#)
- [キャッシュの使用](#)

Java ファイルの使用

Java は Java アーカイブ (JAR) メカニズムを提供して、クライアントにネットワークを介して効率的な配布を行うために、クラスをまとめてグループ化し、圧縮 (Zip 形式) することのできるファイルを作成します。このファイルがクライアントで使用されると、今後の使用のためにキャッシュされます。

Oracle9iAS Forms Services では、通常の配布シナリオをサポートするために、次に示す構成済の JAR ファイルが提供されます。

Oracle JInitiator Oracle JInitiator で使用する次の JAR ファイルが提供されています。

- f90all.jar – 必要なクラスがすべて含まれています。
- f90all_jinit.jar – f90all.jar と同じですが、Oracle JInitiator で使用できるように最適化されています（これはデフォルトです）。

- f90main.jar – f90all.jar よりもクラスの数が減らされています。その他のクラスは、必要に応じて遅延メカニズムを使用してダウンロードできます。そのため、ダウンロード量が少なく、起動時間も短縮されます。

1 つ以上の JAR ファイルを指定するには、Forms 構成ファイル (formsweb.cfg) で指定する構成セクションの archive_jini 設定を使用します。これは、次のように記述します。

[MyApp]

archive_jini=f90all_jinit.jar, icons.jar

archive_jini 設定では、前述の 3 つの JAR ファイルから 1 つのみを使用する必要があります。また、アプリケーションで使用する別のカスタム JAR ファイルも含めることができます (たとえば、前の例に示した icons.jar など)。どのアプリケーションもそれぞれの archive_jini 設定を使用できます。

次に示す JAR ファイルには、f90main.jar にはない遅延クラスが含まれています。これらのファイルは、必要に応じて自動的にダウンロードされるため、archive_jini 設定で参照する必要はありません。また、すでに f90all.jar と f90all_jinit.jar 内には存在するため、f90main.jar を利用する場合のみ使用します。

- 90oracle_laf.jar – Oracle のルック・アンド・フィールのクラス
- 90generic_laf.jar – 汎用 (標準) ルック・アンド・フィールのクラス
- 90resources.jar – 米語以外の言語のリソース・クラス

英語のリソース・クラスは f90all.jar、f90all_jinit.jar および f90main.jar に含まれています。f90resources.jar は米語以外の言語が使用される場合にロードされます。ただし、この JAR ファイルには英語以外のすべての言語のリソースが含まれています。そのため、米語のリソース・クラスか、すべての言語のリソース・クラスのどちらかを持つことになります。

Oracle JInitiator の詳細は、[付録 A「JInitiator」](#)を参照してください。

IE ネイティブ JVM IE は JAR 署名をサポートしないため、CAB ファイルを使用する必要があります。次の CAB ファイルが IE で使用するために提供されています。

- f90all.cab – 必要なクラスがすべて含まれています。

f90all.cab は IE で使用するために提供される唯一のファイルですが、f90all.jar よりかなり小さいファイルです。

1 つ以上の JAR ファイルを指定するには、Forms 構成ファイル (formsweb.cfg) で指定する構成セクションの archive_ie 設定を使用します。これは、次のように記述します。

[MyApp]

archive_ie=f90all.cab

その他のすべての場合（Sun の Java Plug-in など） Jinitiator または IE ネイティブ JVM 以外の Java 仮想マシン（JVM）には、次の JAR ファイルが提供されています。

- f90all.jar - 必要なクラスがすべて含まれています。

1 つ以上の JAR ファイルを指定するには、Forms 構成ファイル（formsweb.cfg）で指定する構成セクションの archive 設定を使用します。これは、次のように記述します。

```
[MyApp]
archive=f90all.jar
```

キャッシュの使用

Oracle9iAS Forms Services でサポートされている JVM（Oracle JInitiator および Oracle JDK）は、どちらも JAR ファイルのキャッシュをサポートします。JVM がクラスを参照する場合、最初にローカルのクライアント・キャッシュにあるキャッシュ済み JAR ファイル内にクラスが存在するかどうかを確認します。クラスがキャッシュ内に存在する場合、JVM はサーバーの JAR ファイルをチェックしてキャッシュ内の JAR ファイルより新しいバージョンが存在するかどうかを確認します。新しいバージョンが存在しない場合は、サーバーからではなくローカルのクライアント・キャッシュからクラスはロードされます。

キャッシュは、効率性を最大化するために適切なサイズにします。キャッシュ・サイズが小さすぎると、有効な JAR ファイルが上書きされてしまう場合があります。その結果、アプリケーションを再度起動すると、別の JAR ファイルのダウンロードが必要になります。デフォルトのキャッシュ・サイズは 20MB です。このサイズは、アプリケーションを正常に実行した後のキャッシュ容量のサイズと比較する必要があります。

JAR ファイルはロード元のホストに関連してキャッシュされます。これには、異なるサーバーからの同一の JAR ファイルがキャッシュを埋めることができるロード・バランス・アーキテクチャという含意があります。JAR ファイルを中央に置き、ロード・バランス構成の各サーバーでそれらを参照することで、開発者は各 JAR ファイルの 1 つのコピーのみがクライアントのキャッシュでメンテナンスされていることを確認できます。この方法を使用すると、JAR ファイル内の特定のクラスに署名して、ロード元のサーバー以外のサーバーに接続を戻せるようにする必要があります。Oracle が提供する JAR ファイルでは、事前にクラスに署名してあります。

必須ネットワーク帯域幅の削減

開発者は、メッセージ間で異なる情報のみを送信する差分メッセージングを使用して、データ・ストリーム圧縮を最大限利用できるようアプリケーションを設計できます。次のステップを実行すると、メッセージ間の相違部分を削減できます。

- **メッセージ送信順序のコントロール。** メッセージの送信順序は次の 2 つの基準で管理されます。
 - 初期画面の場合は、オブジェクト・ナビゲータの表示順
 - 実行中、プログラムの順序は項目プロパティに変更

その結果が有用性に影響を与えない場合は、同じキャンバス上にある似たようなオブジェクトをオブジェクト・ナビゲータ内で並べて配置するようにします。たとえば、ボタンの次にボタン、テキスト項目の次にテキスト項目と配置します。（項目プロパティ「次ナビゲーション項目」を使用する場合、フォーム内の項目でもナビゲーションと同じ順序が使用されます。）オブジェクト・ナビゲータで似たような項目をまとめて順序付けすると、最初のフォームを表示するためにクライアントに送信された項目プロパティには、多くの似たような項目が連続して含まれるので、差分メッセージングアルゴリズムが効率的に機能します。

さらに、トリガーまたは他のロジックを使用して項目プロパティを変更する場合、別の表示タイプの項目プロパティを変更する前に似たような項目のプロパティをまとめてグループ化する必要があります。次に例を示します。

```
set_item_property(text_item1_id, FONT_WEIGHT, FONT_BOLD);
set_item_property(text_item2_id, FONT_WEIGHT, FONT_BOLD);
set_item_property(text_item3_id, FONT_WEIGHT, FONT_BOLD);
set_item_property(button_item1_id, LABEL, 'Exit');
...
```

- **オブジェクト間の類似点を活用。** 似たようなオブジェクトを使用すると、（ユーザーに視覚的によりアピールする上に）差分メッセージングの効率性が向上します。次の手順で、オブジェクト間の一貫性が図られます。
 - プロパティのデフォルト値を受け入れ、オブジェクトに必要な属性のみを変更します。
 - スマート・クラスを使用して、オブジェクトのグループを記述します。
 - ロック・アンド・フィールを少数の可視属性にロックします。
- **ボイラープレート・テキストの使用を削減。** 開発者である場合、可能な限り、ボイラープレート・テキストではなく PROMPT 項目プロパティを使用してください。Forms Developer 6.0 以降には、Associate Prompt 機能が含まれており、この機能を使用して、ボイラープレート・テキストを指定した項目のプロンプトとして再設計できます。
- **ボイラープレート項目（円弧、円、多角形など）の使用を削減。** 指定したフォームのすべてのボイラープレート項目をフォームの初期化時にロードします。ボイラープレート項目をロードしてクライアント上でリソースを使用するには、表示の有無にかかわらず時間がかかります。共通のボイラープレート項目（矩形と線）は最適化されます。このため、アプリケーションをこれらの基本的なボイラープレート項目に制限すると、起動時間を短縮しながらネットワーク帯域幅とクライアント・リソースを削減できます。
- **ナビゲーションを最小に維持。** Event Bundle は、2 つまたはそれ以上のオブジェクトにナビゲーションを拡張する場合でも、ナビゲーション・イベントが完了するたびに送信されます。デフォルト値が受け入れられているときに、フィールド間をナビゲートする必要がないフォームを設計します。フォームは、完了したらユーザーが素早く終了できる

ようにしてください。このようにすると、すべての追加のナビゲーション・イベントは 1 つの Event Bundle として実行されます。

- **初期画面を表示する時間を短縮。** Java クライアントは必須クラスをロードすると、初期画面を表示する前に、表示するすべてのオブジェクトをロードして初期化する必要があります。項目数を最小限に抑えることで、初期画面はより迅速に表示されます。初期画面を表示する時間を短縮する方法は次のとおりです。
 - オブジェクトの集合（タイトル、小さなロゴ、ユーザー名およびパスワードなど）を制限して、アプリケーションのログイン画面を提供します。
 - Form の初期画面では、要素の非表示はただちに必要としません。次のキャンバス・プロパティを使用します。

```
RAISE_ON_ENTRY = YES   (キャンバスのみ)
VISIBLE = NO
```

1 つのシートのみ表示されるいくつかのシートで構成される TAB キャンバスには注意してください。タブ間で応答を切り替える場合、キャンバス上のすべてのシートに対するすべての項目がロードされます。この中には初期タブの後ろに隠れている項目も含まれます。この結果、TAB キャンバスをロードして初期化するためにかかる時間は、初めに可視できるオブジェクトのみではなく、キャンバス上のすべてのオブジェクトに関連します。

ヒント: TAB キャンバスを使用する場合、スタック・キャンバスを使用し、when-tab-page-changed トリガーで正しいキャンバスを表示します。最初の画面に表示されないすべてのキャンバスにプロパティ RAISE_ON_ENTRY = YES と VISIBLE = NO を設定することを忘れないでください。

- **MENU_BUFFERING を使用不可に設定。** デフォルトでは、MENU_BUFFERING は TRUE に設定されます。これは、変更されたメニューが完全な状態で再送信されるときに、メニューに対する変更内容が今後の "同期化" イベントのためにバッファされることを意味します。（ほとんどのアプリケーションは同時に複数の変更を行うことも、まったく行わないこともできます。したがって、クライアント側のメニューを更新する最も効率のよい方法は、すべてのメニューを一度に送信することです。）ただし、指定したアプリケーションはメニューに最小限の変更しか加えない場合があります。この場合、変更するたびに変更内容を送信した方が効率的です。これは、次の文を使用して実行できます。

```
Set_Application_Property (MENU_BUFFERING, 'false');
```

メニューのバッファは、LABEL、ICON、VISIBLE および CHECKED のメニュー・プロパティにのみ適用します。ENABLE/DISABLE イベントは常に送信されますが、メニュー全体を再送信する場合は不要です。

パフォーマンスを改善するためのその他の方法

次に示す方法を使用すると、アプリケーションを実行するのに必要なリソースをさらに削減できます。

- **タイマーを JavaBeans で置換。**多くのタイマーは、時計または動画を提供するために使用されます。タイマーを起動すると、非同期イベントが生成されますが、このイベントと一緒に送出されるキューにたまっている他のイベントが無い場合効率が悪くなります。また、タイマーのサイズはほんの数バイトですが、毎秒実行されるので1分間で60パケットが生成され、通常の営業時間内ではおよそ30,000パケットがネットワークへ送出されます。これらのタイマーを使用するコンポーネントを Forms Services を使用しなくても同じ機能を実現できる自己完結型の JavaBeans で置換します。
- **入力項目の妥当性チェックのローカライズについて考慮。**When-Validate-Item トリガーを使用して項目に対する入力を処理することはよく行われます。トリガー自体は、Forms Services で処理されます。移動可能な Java コンポーネントを使用して、標準のクライアント項目（テキスト・ボックスなど）のデフォルト機能を置換することを考慮する必要があります。次に、日付や最大 / 最小値などの項目の妥当性チェックを項目内に含めます。この方法を使用すると、より複雑な、入力の自動フォーマットなど（XXX-XXXX XXX-XXXX の書式を持つ電話番号など）のアプリケーション固有の妥当性チェックを行うことができます。
- **アプリケーションを、大きな1つのフォームではなく多数の小さいフォームに縮小。**アプリケーションを細かく分けることによって、Forms Services からロードおよび初期化されるオブジェクトをユーザーのナビゲーションで制御できます。大きなフォームの場合、オブジェクトの初期化中にアプリケーションが遅延して、アプリケーションの多くが参照できなくなるという危険性があります。フォームをまとめて連鎖する場合は、ビルトインの OPEN_FORM および NEW_FORM を使用することを検討します。
 - OPEN_FORM を使用すると、コールしているフォームはクライアントとサーバーにオープンされたままの状態になるので、OPEN_FORM による追加のフォームはクライアントとサーバー両方の多くのメモリーを消費します。ただし、フォームが別のユーザーによって使用中である場合、サーバーのメモリー使用量は、データ・セグメントにのみ制限されます。ユーザーが初期フォームに戻ると、フォームはすでにローカル・メモリー内に常駐し再表示の場合でもネットワーク・トラフィックは発生しません。
 - NEW_FORM を使用すると、コールしているフォームはクライアントとサーバー上でクローズされて、すべてのオブジェクト・プロパティが破棄されます。このため、サーバーおよびクライアント上で使用するメモリー量は少なくなります。初期フォームに戻るには、クライアントに再度ダウンロードすることが必要ですが、この場合ネットワーク・リソースが必要で、起動時間が遅延します。初期フォーム（ログイン・フォームなど）を再度コールしない限り、OPEN_FORM を使用して、次のフォームをアプリケーションで表示します。
- **不要なグラフィックとイメージの削除。**アプリケーションに表示されるイメージ項目と背景イメージの数を可能な限り減らします。アプリケーション・ユーザーに対してイメージが表示されるたびに、イメージはアプリケーション・サーバーからユーザーの

Web ブラウザにダウンロードする必要があります。Web アプリケーションで会社のロゴを表示する場合には、アプリケーションの起動時にダウンロードされる HTML ファイルに会社のロゴ・イメージを含めてください。会社のロゴを背景イメージとしてアプリケーションに挿入するかわりにこれを行ってください。会社ロゴを背景イメージとして指定すると、データベースまたはファイルシステムから検索して、ユーザーのマシンにダウンロードするトラフィックが繰り返し発生します。

JInitiator

Oracle JInitiator

この項では、Web ブラウザ・プラグインとして Oracle JInitiator を使用する場合の利点を説明します。Oracle JInitiator を利用すると、Netscape Navigator や Internet Explorer を使用して、Oracle9i Forms アプリケーションを実行できます。ブラウザのデフォルトの JVM ではなく、クライアント上の特定の Java 仮想マシン (JVM) を使用するように指定する機能が提供されます。

Oracle JInitiator は、Netscape Navigator のプラグイン、および Internet Explorer の ActiveX コンポーネントとして実行されます。Oracle JInitiator は、ブラウザによって提供されるデフォルトの JVM を置き換えたり、変更したりしません。かわりに、プラグインのフォームで代替の JVM が提供されます。

オラクル社からは 2 つの JAR ファイル (f90all.jar および f90all_jinit.jar) が提供されます。f90all.jar は標準の JAR ファイルであり、f90all_jinit.jar は Oracle JInitiator でのみ使用できる高圧縮された JAR ファイルです。

Oracle JInitiator を使用する理由

Oracle JInitiator では、Web ブラウザを使用して透過的に起動できる、保証されサポート可能な Java Runtime Environment (JRE) がクライアントのデスクトップに提供されます。

Oracle JInitiator は、JavaSoft の Java Plug-in の Oracle バージョンです。JavaSoft Plug-in は、ブラウザ内から起動できる JavaSoft JRE の配布メカニズムです。また、Oracle JInitiator では、Oracle が保証した JRE の配布メカニズムが提供されます。これにより、Oracle9i Forms アプリケーションを安定しサポートされた方法で、ブラウザ内から実行できます。

Oracle JInitiator では、Oracle9i Forms アプリケーションを実行するために保証されたプラットフォームが提供されるのみではなく、標準 JavaSoft Java Plug-in の他に多くの追加機能も提供されます。これらの機能には、JAR ファイルのキャッシュ書込み、増分の JAR ファイルのロード、およびアプレットのキャッシュ書込みが含まれます (第 8 章の「[アプリケーションの起動時間の最小化](#)」を参照してください)。

Oracle JInitiator の利点

Oracle JInitiator には次の利点があります。

- 旧リリースのブラウザで、最新の Oracle が保証する JVM を実行できます。
- 異なるブラウザ間で JVM の一貫性が保証されます。
- 信頼性が高い実行プラットフォームです。JInitiator は、Oracle9iAS Forms Services での使用が完全にテストされ、保証されています。
- 高性能な実行環境です。JInitiator によって、アプリケーション・クラス・ファイルがキャッシュに自動的に書き込まれます。これにより、アプリケーションを高速起動できます。
- 自己インストールおよび自己メンテナンスが可能な実行環境です。JInitiator はプラグインまたは ActiveX コンポーネントのように、自身を自動的にインストールおよび更新します。ローカルにキャッシュされたアプリケーション・クラス・ファイルは、アプリケーション・サーバーから自動的に更新されます。

Oracle JInitiator の使用方法

クライアントのブラウザが Oracle JInitiator を使用するように指定された HTML ファイルを最初に見つけたとき、Oracle JInitiator はアプリケーション・サーバーからクライアント・マシンに自動的にダウンロードされます。これにより、Windows 98、NT、2000 および XP プラットフォーム上の Netscape Navigator または Internet Explorer 内で、Oracle9i Forms および Graphics アプリケーションを直接実行できます。

Oracle JInitiator は、ブラウザによって提供される標準プラグイン・メカニズムを使用してインストールおよび更新されます。Oracle JInitiator のインストールでは、Oracle9i Forms アプリケーションを信頼されたアプレットとして Oracle JInitiator 環境で実行するために必要なステップが実行されます。

サポートされる構成

Oracle JInitiator では、次の構成がサポートされています。

Windows 98、NT、2000 および XP:

- Netscape Navigator 4.7.3
- Netscape Navigator 4.7.8
- Internet Explorer 5.x

システム要件

Oracle JInitiator の最小システム要件を次に示します。

- Windows 98、NT、2000 および XP
- Pentium 90 MHz 以上のプロセッサ
- 25MB のハード・ディスク空き領域 (30MB を推奨)
- 16MB のシステム RAM (32MB を推奨)

Netscape Navigator での Oracle JInitiator の使用方法

Oracle JInitiator では、QuickTime ムービーまたは Shockwave アニメーション機能など他のプラグインと同様にブラウザ内で実行できるように、Netscape Navigator プラグイン・アーキテクチャが使用されています。Web アプリケーション開発者は Netscape HTML <EMBED> タグを使用して、プラグインを Web ページの一部として実行するように指定できます。これにより、ユーザーの介在を最小にして、Oracle JInitiator を Web ブラウザ内で実行できます。

Netscape Navigator が Oracle JInitiator の使用を指定された HTML ページを最初に見つけたとき、Oracle JInitiator ダウンロード・ページを指示する HTML ページに "Plug-in Not Loaded" ダイアログが表示されます。続いて、各オペレーティング・システム用の Oracle JInitiator バージョンをダウンロードし、インストールできます。

Oracle JInitiator のインストール後、Netscape Navigator をシャットダウンして、再起動します。続いて、元の HTML ページを再び表示します。Oracle JInitiator はアプレットを解放するために、<EMBED> タグ内のパラメータを実行および使用します。Netscape Navigator が Oracle JInitiator の使用を指定された Web ページを次に見つけたとき、ユーザーが介在することなく、Netscape Navigator はプラグインをローカル・ディスクから透過的にロードおよび実行します。

Microsoft Internet Explorer での Oracle JInitiator の使用方法

Oracle JInitiator は Microsoft Internet Explorer 拡張メカニズムを使用して、ActiveX コントロールおよび COM コンポーネントのダウンロードおよびキャッシュ書込みを行います。Web アプリケーション開発者は HTML <OBJECT> タグを使用して、ActiveX コントロールまたは COM コンポーネントを Web ページの一部として実行するように指定できます。このようなコンポーネントには Oracle JInitiator が含まれます。

Internet Explorer は Oracle JInitiator 使用を指定するように変更された HTML ファイルを最初に見つけたとき、オラクル社によって VeriSign デジタル署名が行われた ActiveX コントロールをダウンロードするかどうかをユーザーに確認します。「はい」をクリックすると、Internet Explorer は Oracle JInitiator のダウンロードを開始します。Oracle JInitiator が実行され、アプレットの内容を表現するために <OBJECT> タグ内のパラメータが使用されます。Internet Explorer が Oracle JInitiator をサポートするように変更された Web ページを次に見

つけたとき、ユーザーが介在することなく、Oracle JInitiator をローカル・ディスクから透過的にロードおよび実行します。

Oracle JInitiator プラグインの設定

Oracle JInitiator プラグインを設定する手順は、次のとおりです。

- Oracle JInitiator HTML マークアップをベース HTML ファイルに追加します。
- Oracle JInitiator をサーバーにインストールします（サーバー・ベースのテスト用のみ）。
- Oracle JInitiator ダウンロード・ファイルをカスタマイズします。
- Oracle JInitiator をダウンロード可能にします。

Oracle JInitiator マークアップのベース HTML ファイルへの追加

Oracle JInitiator マークアップをベース HTML ファイルに追加する手順は、次のとおりです。

1. ベース HTML ファイルをテキスト・エディタでオープンします。
2. OBJECT および EMBED タグを追加します。

追加マークアップの例は、第 3 章の「[デフォルトの basejini.htm ファイル](#)」を参照してください。

Oracle JInitiator ダウンロード・ファイルのカスタマイズ

Oracle JInitiator のダウンロード・ファイル (JINIT_DOWNLOAD.HTM) は、ユーザーが Oracle JInitiator ファイルをダウンロードできるテンプレート HTML ファイルです。

Oracle JInitiator ダウンロード・ファイルをカスタマイズする手順は、次のとおりです。

1. JINIT_DOWNLOAD.HTM ファイルを HTML またはテキスト・エディタでオープンします。
2. 必要に応じてテキストを変更します。
3. 変更を保存します。

Oracle JInitiator をダウンロード可能にする

Oracle JInitiator をダウンロード可能にする手順は、次のとおりです。

1. jinit13x.EXE を Web サーバーにコピーします。
jinit13x.EXE は、ベース HTML ファイル内で指定された位置にコピーする必要があります。
2. JINIT_DOWNLOAD.HTM を Web サーバーにコピーします。
JINIT_DOWNLOAD.HTM は、ベース HTML ファイル内で指定された位置にコピーする必要があります。

Oracle JInitiator プラグインの変更

Oracle JInitiator プラグインを変更する手順は、次のとおりです。

- Oracle JInitiator のキャッシュ・サイズを変更します。
- Oracle JInitiator のヒープ・サイズを変更します。
- Oracle JInitiator 用にプロキシ・サーバー設定をチェックおよび変更します。
- Oracle JInitiator 出力を表示します。

Oracle JInitiator キャッシュ・サイズの変更

Oracle JInitiator のキャッシュ・サイズを変更する手順は、次のとおりです。

1. Windows の「スタート」メニューから、「設定」→「コントロール パネル」→「Oracle JInitiator」を選択します。
2. 「基本」タブをクリックします。
3. 「Java 実行時のパラメータ」フィールドで、Dcache サイズを指定します。たとえば、Dcache.size=20000000 と指定すると、キャッシュ・サイズは 20MB に設定されます。

Oracle JInitiator のデフォルト・キャッシュ・サイズは 20000000 です。これは Oracle JInitiator のインストール時に自動設定されます。

Oracle JInitiator ヒープ・サイズの変更

Oracle JInitiator のヒープ・サイズを変更する手順は、次のとおりです。

1. Windows の「スタート」メニューから、「設定」→「コントロール パネル」→「Oracle JInitiator」を選択します。
2. 「基本」タブをクリックします。
3. 「Java 実行時のパラメータ」フィールドで、mx サイズを指定します。たとえば、mx64m と指定すると、最大ヒープ・サイズは 64MB に設定されます。

Oracle JInitiator のデフォルト最大ヒープ・サイズは 64MB です。これは Oracle JInitiator のインストール時に自動設定されます。

Oracle JInitiator 用のプロキシ・サーバー設定のチェックおよび変更

Oracle JInitiator 用にプロキシ・サーバー設定をチェックおよび変更する手順は、次のとおりです。

1. Windows の「スタート」メニューから、「設定」→「コントロール パネル」→「Oracle JInitiator」を選択します。

2. 「プロキシ」タブをクリックします。
3. ブラウザの構成ダイアログ・ボックスの設定を **Oracle JInitiator** で使用できるようにするために、「ブラウザの設定を使用」チェックボックスを選択します。別のプロキシ・サーバー設定を使用する場合は、このボックスにはチェックを付けしないでください。次に、「プロキシのアドレス」フィールドにプロキシ・サーバーのホスト名を入力します。

Oracle JInitiator 出力の表示

Oracle JInitiator 出力を表示する手順は、次のとおりです。

1. Windows の「スタート」メニューから、「設定」→「コントロール パネル」→「Oracle JInitiator」を選択します。
2. 「基本」タブをクリックします。
3. デバッグ出力を使用可能にするために、「Java コンソールの表示」チェックボックスをオンにします。

ベース HTML ファイルの変更

JInitiator を使用して Oracle9i Forms アプリケーションを実行する場合、JInitiator が formsweb.cfg ファイルからパラメータ値を読み込み、その値をベース HTML ファイルに渡します。静的ベース HTML ファイルを作成して常に同じ値が読み出されるようにする場合、手動で値を挿入する必要があります。

Microsoft Internet Explorer および Netscape Navigator の Oracle JInitiator マークアップの例は、第 3 章の「[デフォルトの basejini.htm ファイル](#)」を参照してください。これらのタグをベース HTML ファイルに追加すると、Netscape および Microsoft のブラウザ内でアプリケーションを実行できます。

数字

3 層のアーキテクチャ, 1-4

A

archive_ie パラメータ, 3-6
archive_jinit パラメータ, 3-6
archive パラメータ, 3-6

B

background パラメータ, 3-5
base.htm, 2-3, 3-12
 説明, 3-13
 例, 3-15
baseHTMLIE パラメータ, 3-3
baseHTMLjinitiator パラメータ, 3-3
baseHTMLjpi, 3-3
baseHTML パラメータ, 3-3
baseie.htm, 2-3
 説明, 3-13
 例, 3-20
basejini.htm, 2-3, 3-12
 説明, 3-13
 例, 3-16
basejpi.htm, 2-3

C

CAB ファイル, 8-7
CodeBase, 3-41
codebase パラメータ, 3-5
CollectionId コマンド, 7-6
colorScheme パラメータ, 3-5

D

DataFile コマンド, 7-6
DE_PREFS_TABSIZE, 3-29
default.env, 2-5, 3-28
default.env ファイル, 7-2

E

EAR, 2-3
em_mode, 3-7
envFile, 3-4

F

f90all_jinit.jar
 説明, 2-12
f90all.jar
 説明, 2-12
form, 3-4
FORM90_PATH, 3-29
Forms
 構成ファイル, 2-2
Forms Listener, 1-5, 1-6
Forms Listener Servlet, 1-6
 HTTPS, 3-41
Forms Runtime Diagnostics, 7-1
Forms Services
 アーキテクチャ, 1-4
Forms Servlet, 4-2
Forms Services で使用するアイコンとイメージの配置
 , 3-35
Forms Services の最適化, 8-2
Forms Services リソース要件, 8-2
Forms Trace, 2-3, 7-1, 7-2

FORMS90_CATCHTERM, 3-29
FORMS90_CLAF, 3-31
FORMS90_MMAP, 3-30
FORMS90_REJECT_GO_DISABLED_ITEM, 3-30
FORMS90_SEPARATE_DEBUGGER, 3-31
FORMS90_SWITCH_JAVA_EVENTS, 3-31
FORMS90_TRACE_PATH, 3-30, 7-2
forms90.conf, 2-4, 3-23
formsweb.cfg, 2-2, 3-2
 例, 3-8
Forms ランタイム・プロセス, 1-5, 1-6
FRD, 7-1
ftrace.cfg, 2-3, 7-2

H

height パラメータ, 3-5
HTML delimiter パラメータ, 3-4
HTMLafterForm, 3-4
HTMLbeforeForm, 3-4
HTMLbodyAttrs, 3-4
HTML ベースの Oracle Enterprise Manager, 6-1
HTTP Listener, 4-2
 構成ファイル, 2-4
HTTPD, 4-3
HTTPS
 Forms Listener Servlet, 3-41

I

IE, 3-4
Internet Explorer と JInitiator, A-3

J

J2EE, 4-1
JAR ファイル, 8-6
JAR ファイル、キャッシュ, 8-8
Java plug-in, 8-8
Java クライアント・リソース要件, 8-2
Java ベースの Oracle Enterprise Manager, 6-1
jinit_classid, 3-7
jinit_download_page, 3-7
jinit_exename, 3-7
jinit_mimetype, 3-7
JInitiator, 8-6
 説明, 2-12

JInitiator でサポートされる構成, A-2
JInitiator のキャッシュ・サイズ, A-5
JInitiator の構成, A-4
JInitiator のシステム要件, A-3
JInitiator の説明, A-1
JInitiator のヒープ・サイズ, A-5
JInitiator プロキシ・サーバー, A-5
jpi_classid, 3-7
jpi_codebase, 3-7
jpi_download_page, 3-7

L

Logging Tools, 7-1
lookAndFeel パラメータ, 3-5

M

mapFonts, 3-6
MENU_BUFFERING を使用不可にする, 8-10

N

Netscape Navigator と JInitiator, A-3
networkRetries, 3-6
NLS_LANG, 3-29

O

OC4J, 4-1
 構成ファイル, 2-3
 ロード・バランス, 4-4
OC4J サーバー・プロセス, 4-1
OC4J のロード・バランス, 4-1
OEM (「Oracle Enterprise Manager」を参照), 6-1
OID, 5-1
oid_formsid, 3-7
Oracle Enterprise Manager, 6-1
Oracle HTTP Listener, 4-2
Oracle HTTP Listener の構成ファイル, 2-4
Oracle Internet Directory, 5-1
Oracle JInitiator, 8-6, A-1
Oracle Login Server, 5-1
Oracle Trace, 7-1, 7-18
ORACLE_HOME, 3-28
oracle_home, 3-7
Oracle9i Application Server, 1-2

Oracle9i Database, 1-2
Oracle9i Developer Suite, 1-2
Oracle9i Real Application Clusters, 1-2
Oracle9iAS, 1-2
Oracle9iAS Containers for J2EE (OC4J), 4-1
Oracle9iAS Containers for J2EE (OC4J) の構成ファイル, 2-3
Oracle9iAS Forms Services, 1-3
Oracle9iDS, 1-2
Oracle インターネット・プラットフォーム, 1-2
otherparams, 3-4

P

Password コマンド, 7-6
PATH, 3-28
PECS, 7-1
Performance Event Collection Services, 7-1
Portlet Development Kit, 1-2

R

Registry.dat, 2-5, 3-25
Regressflag コマンド, 7-6
Runform パラメータ, 3-4

S

separateFrame パラメータ, 3-5
serverApp パラメータ, 3-6
serverArgs パラメータ, 3-4
serverURL, 3-5
Servlet Logging Tools, 7-1, 7-14
splashScreen パラメータ, 3-5
SSO, 5-1
Sun の Java Plug-in, 8-8

U

Url コマンド, 7-6
USERID パラメータ, 3-4
UserName コマンド, 7-6

W

web.xml, 2-3, 3-21
Web 上の Forms アプリケーションの機能制限, 3-41

width パラメータ, 3-5
workingDirectory, 3-4

X

XML Developer's Kit, 1-2
XMLFile コマンド, 7-6

Z

VGS ツリー, 8-4

あ

アーキテクチャ
Forms Services, 1-4
アイコン, 3-35
配置, 3-35
アプリケーション
サーバー, 1-4
アプレット
パラメータ, 3-5
暗号化されたプログラム単位, 8-2
イベント、トレース, 7-8
イベントの詳細、トレース, 7-11
イベント・バンドル, 8-3
イメージ, 3-35
スプラッシュ画面, 3-38
バックグラウンド, 3-38

か

仮想グラフィック・システム (VGS) ツリー, 8-4
環境変数, 2-5
関連文書, xii
期間イベント, 7-9
起動時間, 8-5
クライアント層, 1-4
クライアント・リソース要件, 8-2
グラフィック, 3-35
言語検出, 3-43
構成ファイル, 2-2
カスタマイズ, 3-2

さ

サンプル・ファイル

base.htm, 3-15, 3-20
basejinit.htm, 3-16
シングル・サインオン, 5-1
診断ツール, 7-1
スプラッシュ画面, 3-38

た

タイマー、チューニング, 8-11
中間層, 1-4
チューニング, 8-1
JAR ファイルの使用, 8-6
MENU_BUFFERING, 8-10
MENU_BUFFERING を使用不可にする, 8-10
アプリケーションのサイズ, 8-11
初期画面, 8-10
タイマー, 8-11
ナビゲーションの削減, 8-10
ネットワーク帯域幅の削減, 8-8
ボイラープレート・オブジェクトの削減, 8-9
ボイラープレート項目, 8-9
メッセージ順序, 8-8
類似点の活用, 8-9
テンプレート HTML ファイル
作成, 3-35
データ・ストリーム圧縮, 8-8
データ・セグメント, 8-2
データベース層, 1-4
トレース・イベントの詳細, 7-11
トレース可能イベント, 7-8
トレース・ツール, 7-1
トレース・データ用のデータベース・スキーマ, 7-7
トレースの `tracegroup` パラメータ, 7-3
トレースの URL パラメータ・オプション, 7-3
トレースのレコード・パラメータ, 7-3
トレースのログ・パラメータ, 7-4
トレース用 `translate` ユーティリティ, 7-6
トレース用 `upload` ユーティリティ, 7-6

な

ネットワーク
帯域幅の削減, 8-8
ネットワーク使用量, 8-3
ネットワークの待ち時間, 8-3
ネットワーク・パケット, 8-3

は

配布, 2-5
フォームを Web に, 2-1
バックグラウンド, 3-38
パフォーマンス / 拡張性のチューニング, 4-1
パフォーマンス・ツール, 7-1
ビルトイン・イベント, 7-9
ブラウザ, 1-4
ベース HTML, 2-13
ベース HTML ファイル, 2-3
作成, 3-13
ボイラープレート・オブジェクト / イメージ, 8-2
ポイント・イベント, 7-8

ら

リソース、最小化
暗号化されたプログラム単位, 8-2
画面のレンダリング, 8-4
データ・セグメント, 8-2
ネットワーク使用量, 8-3
パケットの送信, 8-3
ボイラープレート・オブジェクト, 8-2
レジストリ設定
カスタマイズ, 3-28
ロギング機能, 7-15
ログ・ファイルのディレクトリ, 7-16