

# Oracle® Application Server Forms Services

利用ガイド

10g (9.0.4)

部品番号 : B12340-02

2004 年 6 月

Oracle Application Server Forms Services 利用ガイド, 10g (9.0.4)

部品番号 : B12340-02

原本名 : Oracle Application Server Forms Services Deployment Guide 10g (9.0.4) for Windows and UNIX

原本部品番号 : B10470-02

原著者 : Orlando Cordero

原協力者 : Emerson deLaubenfels, Art Housinger, Srinil Indla, David Klein, Phil Kuhn, Chris Lewis, Chris Lowes, Duncan Mills, Frank Nimphius, Stephen Noton, Hiromichi Nozaki, Robin Zimmermann.

Copyright © 2004 Oracle Corporation. All rights reserved.

#### 制限付権利の説明

このプログラム（ソフトウェアおよびドキュメントを含む）には、オラクル社およびその関連会社に所有権のある情報が含まれています。このプログラムの使用または開示は、オラクル社およびその関連会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と工業所有権に関する法律により保護されています。

独立して作成された他のソフトウェアとの互換性を得るために必要な場合、もしくは法律によって規定される場合を除き、このプログラムのリバース・エンジニアリング、逆アセンブル、逆コンパイル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場合があります。オラクル社およびその関連会社は、このドキュメントに誤りが無いことの保証は致し兼ねます。これらのプログラムのライセンス契約で許諾されている場合を除き、プログラムを形式、手段（電子的または機械的）、目的に関係なく、複製または転用することはできません。

このプログラムが米国政府機関、もしくは米国政府機関に代わってこのプログラムをライセンスまたは使用する者に提供される場合は、次の注意が適用されます。

#### U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation, and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

このプログラムは、核、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への用途を目的としておりません。このプログラムをかかるとして使用する際、上述のアプリケーションを安全に使用するために、適切な安全装置、バックアップ、冗長性 (redundancy)、その他の対策を講じることが使用者の責任となります。万一かかるプログラムの使用に起因して損害が発生いたしましても、オラクル社およびその関連会社は一切責任を負いかねます。

Oracle は Oracle Corporation およびその関連会社の登録商標です。その他の名称は、Oracle Corporation または各社が所有する商標または登録商標です。

---

---

# 目次

はじめに .....	ix
対象読者 .....	x
このマニュアルの構成 .....	x
関連ドキュメント .....	xi
表記規則 .....	xi
<b>1 概要</b>	
1.1 Oracle インターネット・プラットフォーム .....	1-2
1.1.1 Oracle Application Server (OracleAS) .....	1-2
1.1.2 Oracle Developer Suite (OracleDS) .....	1-2
1.1.3 Oracle Database .....	1-3
1.2 Oracle Application Server Forms Services .....	1-3
1.2.1 Forms Services の新機能 .....	1-3
1.3 OracleAS Forms Services のアーキテクチャ .....	1-5
1.4 OracleAS Forms Services のコンポーネント .....	1-6
1.4.1 Forms Listener Servlet .....	1-7
1.4.2 Forms ランタイム・プロセス .....	1-7
1.5 Forms Listener Servlet .....	1-7
<b>2 Forms Services セキュリティの概要</b>	
2.1 OracleAS Forms Services のセキュリティについて .....	2-2
2.1.1 OracleAS Forms Services のシングル・サインオン .....	2-2
2.1.2 ユーザーのクラスとその権限 .....	2-2
2.1.3 保護されるリソース .....	2-3
2.1.3.1 動的ディレクティブ .....	2-3

2.1.3.2	OID での動的リソースの作成 .....	2-3
2.1.3.3	シングル・サインオン使用時のデータベース・パスワードの期限切れ .....	2-3
2.1.4	権限およびアクセス強制 .....	2-4
2.1.5	Oracle Identity Management Infrastructure の使用 .....	2-4
2.2	OracleAS Forms Services のセキュリティの構成 .....	2-4
2.2.1	Oracle Forms の Oracle Identity Management オプションの構成 .....	2-4
2.2.2	OracleAS Security Framework の Oracle Forms オプションの構成 .....	2-4

### 3 Oracle Forms アプリケーションの配布の基本

3.1	構成ファイル .....	3-2
3.1.1	Oracle Forms 構成ファイル .....	3-2
3.1.1.1	default.env .....	3-2
3.1.1.2	formsweb.cfg .....	3-3
3.1.1.3	base.htm、basejini.htm、basejpi.htm および baseie.htm .....	3-3
3.1.1.4	ftrace.cfg .....	3-3
3.1.2	Oracle Application Server Containers for J2EE (OC4J) 構成ファイル .....	3-4
3.1.2.1	web.xml .....	3-4
3.1.2.2	Oracle Forms OC4J ファイルのディレクトリ構造 .....	3-4
3.1.3	Oracle HTTP Listener の構成ファイル .....	3-5
3.1.3.1	forms90.conf .....	3-5
3.1.4	標準フォントとアイコン・ファイル .....	3-5
3.1.4.1	Registry.dat .....	3-5
3.2	アプリケーションの配布 .....	3-5
3.2.1	アプリケーションの配布 .....	3-6
3.2.2	パラメータの指定 .....	3-7
3.2.3	Oracle Enterprise Manager での構成セクションの作成 .....	3-8
3.2.3.1	Oracle Application Server Forms Services アプリケーションへのアクセスに使用する URL の編集 .....	3-9
3.2.4	ランフォーム・パラメータにおける特殊文字の指定 .....	3-9
3.2.4.1	現行リリースのデフォルト動作 .....	3-10
3.2.4.2	前リリースの動作 .....	3-11
3.2.4.3	現行リリースでの前リリースの動作取得 .....	3-11
3.2.4.4	テンプレート HTML ファイルの考慮事項 .....	3-11
3.2.4.5	静的 HTML ページの考慮事項 .....	3-12
3.3	OracleAS Forms Services のアクション .....	3-13
3.4	クライアント・ブラウザのサポート .....	3-15
3.4.1	Oracle JInitiator .....	3-15

## 4 Oracle Enterprise Manager による Forms Services の構成

4.1	Oracle Application Server Forms Services での Forms アプリケーションの起動方法 .....	4-2
4.2	Oracle Enterprise Manager と Oracle Forms .....	4-2
4.2.1	Forms セッション管理のための Oracle Enterprise Manager の使用方法 .....	4-3
4.2.2	Forms Services の管理に使用する Oracle Enterprise Manager Grid Control の構成 .....	4-4
4.2.3	Application Server Control による Forms Services へのアクセス .....	4-4
4.3	Forms Services の構成 .....	4-5
4.3.1	Application Server Control によるパラメータの構成 .....	4-6
4.3.1.1	ファイルを指定するパラメータ .....	4-6
4.3.2	構成セクションの管理 .....	4-7
4.3.2.1	名前を付けた構成の複製 .....	4-8
4.3.2.2	名前を付けた構成の削除 .....	4-8
4.3.3	パラメータの管理 .....	4-9
4.3.4	デフォルトの Forms 構成パラメータ .....	4-10
4.3.4.1	システムのデフォルトの構成パラメータ .....	4-10
4.3.4.2	ランフォーム・パラメータ (serverArgs パラメータ) .....	4-12
4.3.4.3	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 .....	4-14
4.3.4.4	アプレット・パラメータまたはオブジェクト・パラメータ .....	4-15
4.3.4.5	JInitiator のパラメータ .....	4-17
4.3.4.6	Sun 社の Java Plug-in のパラメータ .....	4-18
4.3.4.7	Oracle Enterprise Manager の構成パラメータ .....	4-18
4.3.4.8	OID (Oracle Internet Directory) の構成パラメータ .....	4-18
4.4	Oracle Enterprise Manager による環境変数の構成 .....	4-19
4.5	ユーザー・セッションの管理 .....	4-22
4.5.1	新規ユーザー・セッションの許可 .....	4-22
4.5.2	新規ユーザー・セッションの禁止 .....	4-22
4.5.3	Forms Services インスタンスでのユーザー・セッションの終了 .....	4-23
4.6	アプリケーションの URL セキュリティの管理 .....	4-23
4.7	ユーザー独自のテンプレート HTML ファイルの作成 .....	4-25
4.8	Oracle Forms アプリケーションへの Graphics の挿入 .....	4-25
4.9	Forms Services で使用するアイコンとイメージの配置 .....	4-26
4.9.1	アイコン .....	4-26
4.9.1.1	アイコンの Java アーカイブへの格納 .....	4-26
4.9.1.2	Registry.dat へのアイコン変更の追加 .....	4-27

4.9.2	スブ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 .....	4-29
4.9.3	アイコンとイメージを含むカスタム JAR ファイル .....	4-29
4.9.3.1	JAR ファイルの作成 .....	4-29
4.9.3.2	JAR ファイル内でのファイルの使用 .....	4-29
4.9.4	アイコンおよびイメージの検索パス .....	4-30
4.9.4.1	DocumentBase .....	4-31
4.9.4.2	CodeBase .....	4-32
4.10	言語検出の有効化 .....	4-33
4.10.1	言語検出の指定 .....	4-33
4.10.2	言語検出機能の仕組み .....	4-33
4.10.2.1	マルチレベルの継承 .....	4-34

## 5 HTTP Listener と OC4J での OracleAS Forms Services の使用方法

5.1	OC4J サーバー・プロセス .....	5-2
5.2	パフォーマンス / スケーラビリティのチューニング .....	5-3
5.3	HTTPD プロセス数の制限 .....	5-3
5.4	高い値の MaxClients ディレクティブの設定 .....	5-3
5.5	OC4J のロード・バランシング .....	5-3
5.6	Forms Listener Servlet での HTTPS の使用方法 .....	5-6
5.7	サーバー要件 .....	5-6
5.8	クライアント要件 : Oracle JInitiator での HTTPS の使用 .....	5-7
5.9	ユーザー ID/ パスワード非表示機能の使用 .....	5-7
5.10	認証プロキシを使用した Oracle Forms アプリケーションの実行 .....	5-8

## 6 Oracle Application Server Single Sign-On と OID での Forms Services の使用方法

6.1	SSO、OID および Forms の新機能 .....	6-2
6.1.1	OID にリソースがない場合の動的リソース作成 .....	6-2
6.1.2	OID でのデフォルト・プリファレンス (Forms リソースの定義) のサポート .....	6-2
6.1.3	Forms SSO での動的ディレクティブのサポート .....	6-3
6.1.4	SSO で実行される Forms のデータベース・パスワード期限切れのサポート .....	6-3
6.2	Forms で使用される SSO コンポーネント .....	6-3
6.3	アプリケーションでの SSO の有効化 .....	6-4
6.3.1	ssoMode .....	6-4
6.3.2	ssoDynamicResourceCreate .....	6-6
6.3.3	ssoErrorURL .....	6-7

6.3.4	ssoCancelUrl .....	6-7
6.3.5	Forms から SSO 情報へのアクセス .....	6-7
6.4	Oracle Forms と Reports の統合について .....	6-8
6.5	認証フロー .....	6-8

## 7 トレースと診断

7.1	Forms Trace .....	7-2
7.1.1	Forms Trace の構成 .....	7-2
7.1.1.1	URL パラメータ・オプションの指定 .....	7-6
7.1.2	トレースの開始 .....	7-7
7.1.3	Forms Trace 出力の表示 .....	7-8
7.1.3.1	Translate ユーティリティの実行 .....	7-8
7.1.4	トレース可能イベントのリスト .....	7-8
7.1.5	イベントの詳細のリスト .....	7-11
7.1.5.1	ユーザー・アクション・イベント .....	7-12
7.1.5.2	Forms Services のイベント .....	7-12
7.1.5.3	イベントの詳細 .....	7-13
7.1.5.4	3 層イベント .....	7-13
7.1.5.5	その他 .....	7-14
7.1.6	Forms Services のトレース・メトリックの監視 .....	7-14
7.2	Servlet Logging Tools .....	7-15
7.2.1	ロギングの有効化 .....	7-15
7.2.1.1	URL でのロギングの指定 .....	7-16
7.2.1.2	Oracle Enterprise Manager でのロギングの指定 .....	7-16
7.2.1.3	Forms Servlet を起動する URL での完全診断の指定 .....	7-16
7.2.2	ログ・ファイルのディレクトリ .....	7-16
7.2.3	各レベルのサーブレット・ロギングの出力例 .....	7-17
7.2.3.1	(なし) .....	7-17
7.2.3.2	/session .....	7-17
7.2.3.3	/sessionperf .....	7-17
7.2.3.4	/perf .....	7-18
7.2.3.5	/debug .....	7-18

## 8 パフォーマンス・チューニングに関する考慮事項

8.1	Forms Services の組込み最適化機能 .....	8-2
-----	--------------------------------	-----

8.1.1	Forms Services の監視 .....	8-2
8.1.1.1	Forms Services インスタンスの監視 .....	8-2
8.1.1.2	Forms イベントの監視 .....	8-3
8.1.1.3	ユーザー・セッションのメトリックの監視 .....	8-3
8.1.1.4	メトリック情報のソート .....	8-4
8.1.1.5	検索 .....	8-4
8.1.2	Forms Services Web ランタイム・プーリング .....	8-4
8.1.2.1	事前起動パラメータの構成 .....	8-4
8.1.2.2	ランタイム・プーリングの起動 .....	8-5
8.1.3	Forms Services のユーティリティ .....	8-6
8.1.3.1	Forms Services のユーティリティを使用するには: .....	8-6
8.1.4	クライアント・リソース要件の最小化 .....	8-6
8.1.5	Forms Services リソース要件の最小化 .....	8-6
8.1.6	ネットワーク使用量の最小化 .....	8-7
8.1.7	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 .....	8-7
8.1.8	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 .....	8-8
8.2	OracleAS Forms Services アプリケーションのチューニング .....	8-8
8.2.1	データ・サーバーに対する Oracle Application Server Forms Services の位置 .....	8-9
8.2.2	アプリケーションの起動時間の最小化 .....	8-10
8.2.2.1	Java ファイルの使用 .....	8-11
8.2.2.1.1	Oracle JInitiator .....	8-11
8.2.2.1.2	IE ネイティブ JVM .....	8-12
8.2.2.1.3	その他のすべての場合 (Sun 社の Java Plug-in など) .....	8-12
8.2.2.2	キャッシュの使用 .....	8-12
8.2.3	必須ネットワーク帯域幅の削減 .....	8-13
8.2.4	パフォーマンスを改善するためのその他の方法 .....	8-15
8.3	Web Cache と Forms の統合 .....	8-17

## A JInitiator

A.1	Oracle JInitiator を使用する理由 .....	A-2
A.2	Oracle JInitiator の利点 .....	A-2
A.3	Oracle JInitiator の使用方法 .....	A-3
A.4	サポートされる構成 .....	A-3
A.4.1	Windows 98、NT、2000 および XP: .....	A-3
A.5	システム要件 .....	A-3
A.6	Netscape Navigator での Oracle JInitiator の使用方法 .....	A-4



A.7	Microsoft Internet Explorer での Oracle JInitiator の使用方法 .....	A-4
A.8	Oracle JInitiator プラグインの設定 .....	A-5
A.8.1	Oracle JInitiator マークアップのベース HTML ファイルへの追加 .....	A-5
A.8.2	Oracle JInitiator ダウンロード・ファイルのカスタマイズ .....	A-5
A.8.3	Oracle JInitiator をダウンロード可能にする .....	A-5
A.9	Oracle JInitiator プラグインの変更 .....	A-6
A.9.1	Oracle JInitiator キャッシュ・サイズの変更 .....	A-6
A.9.2	Oracle JInitiator ヒープ・サイズの変更 .....	A-6
A.9.3	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-7
A.9.4	Oracle JInitiator 出力の表示 .....	A-7
A.10	ベース HTML ファイルの変更 .....	A-7

## B サンプル構成ファイル

B.1	デフォルトの formsweb.cfg ファイル .....	B-2
B.2	プラットフォーム固有の default.env ファイル .....	B-7
B.2.1	デフォルトの Windows 用 default.env ファイル .....	B-7
B.2.2	デフォルトの UNIX 用 default.env ファイル .....	B-9
B.3	base.htm ファイル、basejini.htm ファイル、basejpi.htm ファイルおよび baseie.htm ファイル .....	B-11
B.3.1	ベース HTML ファイル内のパラメータと変数 .....	B-13
B.3.1.1	使用方法 .....	B-13
B.3.2	デフォルトの base.htm ファイル .....	B-13
B.3.3	デフォルトの basejini.htm ファイル .....	B-15
B.3.4	デフォルトの basejpi.htm ファイル .....	B-17
B.3.5	デフォルトの baseie.htm ファイル .....	B-19
B.4	web.xml .....	B-20
B.4.1	デフォルトの web.xml ファイル .....	B-21
B.5	forms90.conf .....	B-23
B.5.1	デフォルトの forms90.conf .....	B-23
B.6	Registry.dat .....	B-25
B.6.1	デフォルトの Registry.dat .....	B-25

## 索引



---

はじめに

## 対象読者

このマニュアルは、Oracle Application Server を使用して Oracle 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「Forms Services セキュリティの概要」

Oracle Application Server Forms Services のセキュリティ機能の使用法について、概要が記載されています。

### 第 3 章「Oracle Forms アプリケーションの配布の基本」

OracleAS 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 4 章「Oracle Enterprise Manager による Forms Services の構成」

OracleAS Forms Services の構成に関する詳細、および Web に配布する Oracle 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 5 章「HTTP Listener と OC4J での OracleAS Forms Services の使用方法」

Oracle HTTP Listener と OC4J によって OracleAS Forms Services がどのように動作するかを説明します。

### 第 6 章「Oracle Application Server Single Sign-On と OID での Forms Services の使用方法」

OracleAS 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 7 章「トレースと診断」

Oracle Enterprise Manager (OEM) システム管理ツール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」

Oracle 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ための、Forms 付属のトレース・ツールおよび診断ツールについて説明します。

## 第 A 章「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。

## 第 B 章「サンプル構成ファイル」

環境、パラメータ、および base.htm、basejini.htm、web.xml、forms90.conf、baseie.htm の各ファイルのサンプル。

# 関連ドキュメント

次のマニュアルを参照してください。

- Oracle Application Server のリリース・ノート
- 『Oracle Developer Suite リリース・ノート』
- 『Oracle Forms for Windows and UNIX Forms6i からの Forms アプリケーションの移行』
- Oracle 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)

# 表記規則

この項では、このマニュアルの本文およびコード例で使用される表記規則について説明します。

規則	意味
. . .	垂直の省略記号は、例に直接関連しない複数の行が省略されていることを示します。
...	水平の省略記号は、例に直接関連しないコードの一部が省略されていることを示します。
<b>太字</b>	太字は、本文中で定義されている用語および用語集に記載されている用語を示します。
<>	山カッコは、カッコ内の項目が、ユーザー指定の名前であることを表します。
[ ]	大カッコは、カッコ内の項目を任意に選択することを表します。大カッコは、入力しないでください。



# 1

## 概要

このマニュアルでは、Oracle Application Server 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
- Oracle Application Server Forms Services
- OracleAS Forms Services のアーキテクチャ
- OracleAS Forms Services のコンポーネント
- Forms Listener Servlet

## 1.1 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、Oracle Database でデータを管理し、Oracle Developer Suite (OracleDS) でアプリケーションを作成して、Oracle Application Server でアプリケーションを実行します。この Oracle 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、ユーザー数の増加に容易に適應できます。

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。

- [Oracle Application Server \(OracleAS\)](#)
- [Oracle Developer Suite \(OracleDS\)](#)
- [Oracle Database](#)

### 1.1.1 Oracle Application Server (OracleAS)

Oracle Application Server 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。OracleAS 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 Oracle 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。

### 1.1.2 Oracle Developer Suite (OracleDS)

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

OracleDS は、J2EE、XML、SOAP、UDDI、UML などのインターネット標準に基づいて、Oracle Application Server と Oracle Database サーバー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。



### 1.1.3 Oracle Database

Oracle Database サーバーは、世界で最も一般的な RDBMS の新世代の製品です。Oracle Real Application Clusters の無限のスケーラビリティ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。

Oracle Application Server は、Oracle Database サーバーと、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。Oracle 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 Database サーバーを透過的に拡張できます。

## 1.2 Oracle Application Server Forms Services

Oracle Application Server Forms Services は、Oracle Application Server の一部として、Forms Services 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。

Forms Services は、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。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。

### 1.2.1 Forms Services の新機能

Forms 6i の Web Server に搭載されていた機能の大半は、Oracle Application Server 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、スケーラビリティ、HTTP/S 通信処理、Java サブレットの配布機能は、すべて OracleAS で提供される Oracle HTTP Server や Oracle Application Server 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。

OracleAS の Forms Services コンポーネントは、Forms Developer 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 Database サーバーとの接続など、Forms Developer 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。

Forms Services には、次のような新しい特長があります。

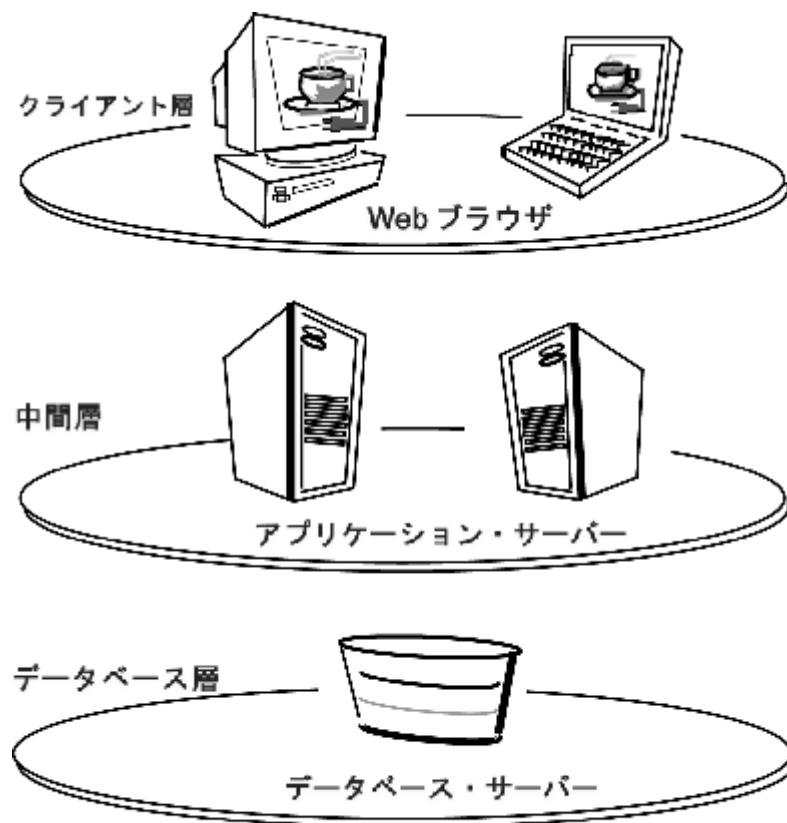
- ランタイム・プーリング (第 8.1.2 項「Forms Services Web ランタイム・プーリング」を参照)
- Oracle Enterprise Manager の向上した Web インタフェース (第 6.3 項「アプリケーションでの SSO の有効化」を参照)
- シングル・サインオン (SSO) の向上 (第 6 章「Oracle Application Server Single Sign-On と OID での Forms Services の使用方法」を参照)
- Forms Listener Servlet を使用した Web への配布サポート
- OC4J との統合 (第 5 章「HTTP Listener と OC4J での OracleAS Forms Services の使用方法」を参照)
- OID との統合 (第 6 章「Oracle Application Server Single Sign-On と OID での Forms Services の使用方法」を参照)
- 運用管理を容易にする Oracle Enterprise Manager との統合 (第 7 章「トレースと診断」を参照)
- トレースとロギングの改善 (第 8 章「パフォーマンス・チューニングに関する考慮事項」を参照)

## 1.3 OracleAS Forms Services のアーキテクチャ

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、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 OracleAS Forms Services アーキテクチャ



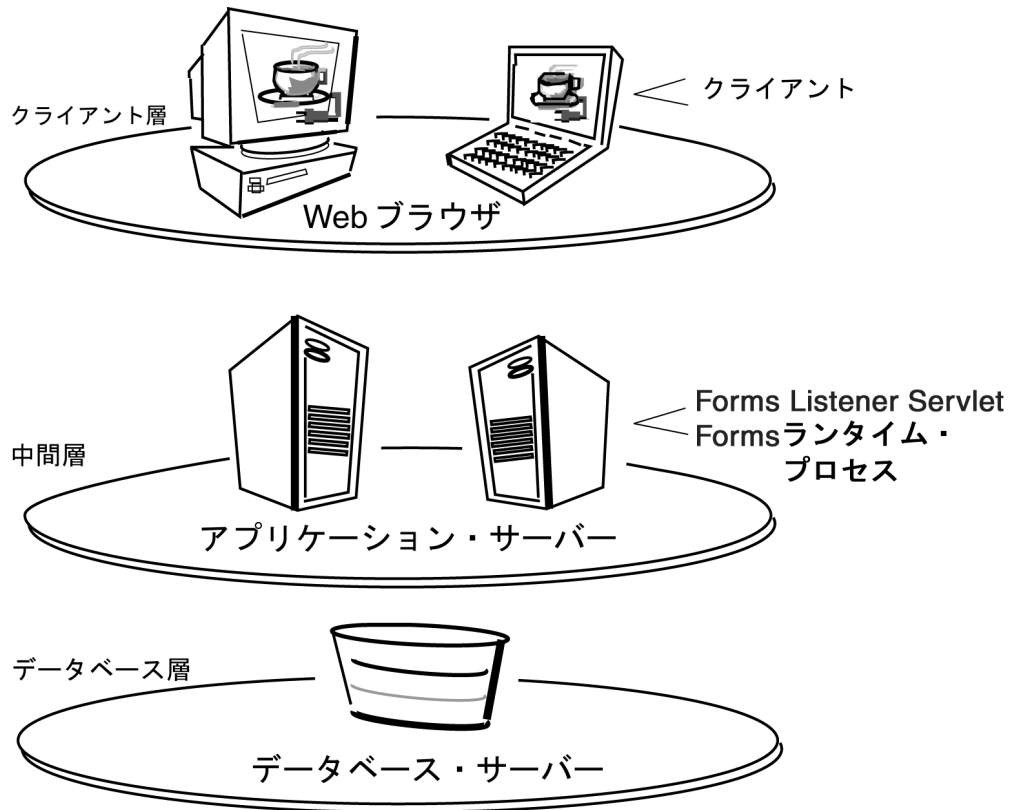
## 1.4 OracleAS Forms Services のコンポーネント

Oracle Application Server Forms Services は、複雑なトランザクション型の Forms 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、Forms Developer を使用して新規のアプリケーションを作成し、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。

OracleAS 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。

- クライアント (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



## 1.4.1 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ーとしての役割を果たします。Forms Listener Servlet は、Java クライアント・プロセスから接続リクエストを受け取ると、Java クライアント・プロセスにかわって Forms ランタイム・プロセスを開始します。

## 1.4.2 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・ブラウザからのリクエスト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、リクエスト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。

## 1.5 Forms Listener Servlet

OracleAS 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 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、Oracle Forms の以前のリリースで提供されている Forms Listener にかわるものです。

---

---

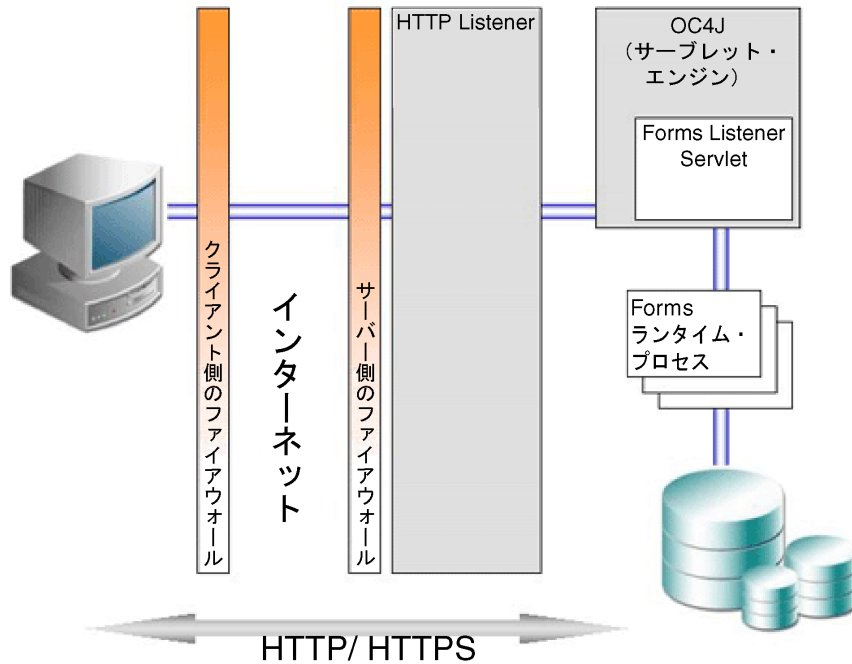
**注意：** Forms Listener Servlet は、すでに OracleAS のインストール・プロセスで設定されているため、構成する必要はありません。

---

---

図 1-3 は、クライアントが HTTP リクエストを送信し、Forms Server プロセスから HTTP レスポンス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



---

## Forms Services セキュリティの概要

Web コンテンツへのユーザー・アクセスを制御し、システムへの侵入者からサイトを保護する機能は非常に重要です。この章では、OracleAS Forms Services におけるセキュリティのアーキテクチャと構成について説明します。

- [OracleAS Forms Services のセキュリティについて](#)
- [OracleAS Forms Services のセキュリティの構成](#)

**関連項目：** セキュリティの詳細は、次のドキュメントを参照してください。

- 『Oracle Application Server 10g セキュリティ・ガイド』には、Oracle Application Server のセキュリティとそのコア機能の概要が記載されています。
- 『Oracle Identity Management 概要および配置プランニング・ガイド』には、Oracle セキュリティ・インフラストラクチャの管理者用の手引きが記載されています。

## 2.1 OracleAS Forms Services のセキュリティについて

この項では、シングル・サインオンを有効化する場合に、Forms アプリケーションの保護に使用できる OracleAS Portal の機能について説明します。

### 2.1.1 OracleAS Forms Services のシングル・サインオン

Oracle Application Server Forms Services のシングル・サインオンは、Oracle HTTP Server の Oracle モジュール `mod_osso` を通じて利用できます。`mod_osso` は、Oracle Application Server Single Sign-On に対してユーザーを認証し、次に Oracle Internet Directory (OID) をユーザー・リポジトリとして使用してから Forms アプリケーションのリクエストを Forms Servlet に転送します。

Forms アプリケーションでは、データベース接続文字列がアプリケーションのリクエストとともに渡されます。渡されない場合は、ログイン・ダイアログが表示されます。シングル・サインオン環境でデータベース接続情報を取得するには、ユーザーのシングル・サインオン名、認証されたユーザー名およびユーザーが起動をリクエストしているアプリケーション名を結合して作成される一意キーの値を、Forms Servlet を使用して OID に問い合わせます。

リソース・アクセス記述子 (RAD) は、各ユーザーおよびアプリケーションに対して定義される、必要なデータベース接続情報を含む OID のエントリです。Forms Servlet は RAD からデータベース接続情報を読み取って、Forms Web アプリケーションを起動するコマンドラインとともに渡します。Forms 認証はまだデータベース中心ですが、`mod_osso` および Forms Servlet は Web ベースのシングル・サインオン環境に統合されています。

### 2.1.2 ユーザーのクラスとその権限

従来、Forms アプリケーションでは、アプリケーション・ユーザーの認証と許可にデータベースが使用されています。シングル・サインオンで Oracle Application Server Forms Services を使用するには、ユーザー・アカウントとその接続情報が Oracle Internet Directory で利用可能である必要があります。Oracle Internet Directory では、PL/SQL、Java または Oracle Delegated Administration Services (DAS) を使用した複数の方法でユーザー・データが供給されます。DAS は、SSO ユーザーおよび委任管理者用の Web ベースのユーザー・インタフェースであり、権限を持つ OID のセルフサービス・データの管理に使用します。

OID でユーザー・アカウントを作成した後は、ユーザーによる Forms アプリケーションの初回リクエスト時に、(このアプリケーションに必要なデータベース接続情報をユーザーが知っていることを前提として) リソース・アクセス記述子 (RAD) のエントリを動的に作成できます。



もうひとつの選択肢は、DAS で作成可能な RAD のエントリーを使用することです。デフォルトの RAD エントリーには、Oracle Application Server Single Sign-On で認証されるすべてのユーザーがアクセスできます。特定の Forms アプリケーションを Web で実行しているときに、すべてのユーザーが同じデータベース接続情報を共有している場合はデフォルトの RAD を使用します。このように、ユーザーはその SSO 接続情報によって個別に認証されますが、デフォルトの RAD エントリーで定義されたアプリケーションでは、すべてのユーザーが共通のデータベース接続を共有します。

## 2.1.3 保護されるリソース

Forms アプリケーションに対してシングル・サインオンを有効化する場合、次の機能で Forms アプリケーションを保護できます。

### 2.1.3.1 動的ディレクティブ

動的な `mod_osso` ディレクティブでは、同一の Oracle Application Server Forms Services インスタンスから、シングル・サインオンで保護された Forms アプリケーションと保護されない Forms アプリケーションの両方が実行され、同一の構成ファイルと Forms Servlet が使用されます。アプリケーションに対するシングル・サインオンは、`forms90/server/formsweb.cfg` 構成ファイルのアプリケーション定義にあるシングル・サインオン・パラメータを使用して有効化します。

### 2.1.3.2 OID での動的リソースの作成

Oracle Application Server Forms Services の以前のリリースでは、特定のアプリケーションおよびユーザーで RAD 定義が見つからない場合は、エラー・メッセージが表示され、認証済にもかかわらずユーザーはその Forms アプリケーションを実行できませんでした。Oracle Application Server Forms Services のこのリリースでは、RAD 定義が存在しない場合は、Oracle Application Server Forms Services を構成して、ユーザーがリアルタイムにこのアプリケーションの RAD を作成できるようになりました。

### 2.1.3.3 シングル・サインオン使用時のデータベース・パスワードの期限切れ

Oracle Application Server Forms Services の以前のリリースでは、データベース・パスワードが期限切れの場合、OID の RAD 情報は更新されませんでした。そのため、ユーザーは、Forms アプリケーションへの接続時にデータベース・パスワードを更新していました。Oracle Application Server Forms Services のこのリリースでは、Forms によるデータベース・パスワードの更新に伴って OID の RAD 情報も自動的に更新されます。Oracle Application Server Forms Services のこの機能を使用するために、追加構成を行う必要はありません。

## 2.1.4 権限およびアクセス強制

ユーザーによる Oracle Application Server Forms Services URL の初回リクエスト時など (パートナー・アプリケーションからのリクエスト時も含む)、Oracle Application Server Forms Services における SSO サポートの認証フローの詳細は、[第 6.5 項「認証フロー」](#) を参照してください。

## 2.1.5 Oracle Identity Management Infrastructure の使用

Oracle Application Server Forms Services では、最小限の構成で Oracle Internet Directory との統合が強化されています。Forms アプリケーションにシングル・サインオンを構成するとき、Oracle Application Server Forms Services では、構成と相互作用の大部分が OID で処理されます。シングル・サインオンと OID の構成の詳細は、[第 6 章「Oracle Application Server Single Sign-On と OID での Forms Services の使用方法」](#) を参照してください。

## 2.2 OracleAS Forms Services のセキュリティの構成

OracleAS Forms Services のセキュリティの構成は、Oracle Enterprise Manager Application Server Control で行われます。画面ごとにオンライン・ヘルプを利用できます。詳細は、[第 4 章「Oracle Enterprise Manager による Forms Services の構成」](#) および [第 6 章「Oracle Application Server Single Sign-On と OID での Forms Services の使用方法」](#) を参照してください。

### 2.2.1 Oracle Forms の Oracle Identity Management オプションの構成

OID でリソースを動的に作成したり、OID リソースを持たないユーザーが共通リソースを使用できるように、OracleAS Forms Services を構成することができます。

詳細は、[第 6 章「Oracle Application Server Single Sign-On と OID での Forms Services の使用方法」](#) を参照してください。

### 2.2.2 OracleAS Security Framework の Oracle Forms オプションの構成

Oracle Forms の構成と保護の詳細は、次の各章を参照してください。

[第 4 章「Oracle Enterprise Manager による Forms Services の構成」](#)

[第 5 章「HTTP Listener と OC4J での OracleAS Forms Services の使用方法」](#)

[第 6 章「Oracle Application Server Single Sign-On と OID での Forms Services の使用方法」](#)

[第 7 章「トレースと診断」](#)

---

---

# Oracle Forms アプリケーションの配布の基本

この章では、Oracle Forms の構成に必要な基本ファイル、Forms Services が Oracle Application Server で実行される仕組み、および 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](#)
- [OracleAS Forms Services のアクション](#)
- [クライアント・ブラウザのサポート](#)

## 3.1 構成ファイル

この項では、Forms アプリケーションの構成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 4 章「Oracle Enterprise Manager による 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。

- Oracle Forms 構成ファイル
- Oracle Application Server Containers for J2EE (OC4J) 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

---

---

**注意：** ファイルが存在する場所は、<ORACLE\_HOME> ディレクトリ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。

---

---

### 3.1.1 Oracle Forms 構成ファイル

Oracle Forms 構成ファイルを使用すると、Forms のパラメータを指定できます。このパラメータは Application Server Control で管理します。この項は、次の小項目に分かれています。

- default.env
- formsweb.cfg
- base.htm、basejini.htm、basejpi.htm および baseie.htm
- ftrace.cfg

---

---

**注意：** 構成ファイルや環境ファイルを手動で編集する場合は、Oracle Enterprise Manager がすべての変更を読み取れるように Oracle Enterprise Manager を再起動する必要があります。Oracle Enterprise Manager を再起動しない場合、これらのファイルに手動で加えた変更は、その後 Oracle Enterprise Manager で加えた変更によって上書きされます。

---

---

#### 3.1.1.1 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、<ORACLE\_HOME>/forms90/server 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。

### 3.1.1.2 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 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 リクエストがある場合は、その問合せパラメータの値によって置換されます。

Oracle Enterprise Manager Application Server Control で formsweb.cfg ファイルを管理します。

formsweb.cfg の詳細は、[第 4.3.1 項「Application Server Control によるパラメータの構成」](#)を参照してください。

### 3.1.1.3 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、Oracle 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。

サンプルのベース HTML ファイルを確認するには、[第 B.3 項「base.htm ファイル、basejini.htm ファイル、basejpi.htm ファイルおよび baseie.htm ファイル」](#)を参照してください。

### 3.1.1.4 ftrace.cfg

場所 : forms90/server

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 Forms の以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。

Oracle Enterprise Manager Application Server Control で Forms Trace を管理します。

ftrace.cfg の詳細は、[第 7 章「トレースと診断」](#)を参照してください。

## 3.1.2 Oracle Application Server Containers for J2EE (OC4J) 構成ファイル

デフォルトでの Forms Services 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。この EAR ファイルは Oracle Application Server のインストール・プロセス時に配布されます (Oracle 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](#)
- [Oracle Forms OC4J ファイルのディレクトリ構造](#)

### 3.1.2.1 web.xml

場所 : j2ee/OC4J\_BI\_FORMS/applications/forms90app/forms90web/WEB-INF/web.xml

Oracle Application Server Forms Services 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。このファイルでは、Forms Servlet と Forms Listener Servlet 用に、別名 f90servlet と l90servlet が定義されます。

web.xml の詳細は、[第 B.4 項「web.xml」](#) を参照してください。

### 3.1.2.2 Oracle Forms OC4J ファイルのディレクトリ構造

Oracle Application Server のインストールおよび構成時に、Forms EAR ファイル (forms90app.ear) が "OC4J\_BI\_FORMS" OC4J 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。

```
<ORACLE_HOME>/j2ee/OC4J_BI_FORMS/applications/forms90app
+META-INF
-application.xml (defines the structure of the ear file)
+forms90web
+WEB-INF
-web.xml (forms & listener servlet definitions, including servlet parameters)
-orion-web.xml (virtual directory mappings and context parameter, only used in iDS)
+lib
    -f90srv.jar (contains the Forms Servlet and Listener Servlet code)
```

### 3.1.3 Oracle HTTP Listener の構成ファイル

この項では、Oracle Application Server Forms Services 用の Oracle HTTP Listener の構成に使用するファイルについて説明します。

#### 3.1.3.1 forms90.conf

場所 : forms90/server

forms90.conf は、Oracle Application Server Forms Services 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 リクエストを OC4J サーブレット・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 B.5 項「forms90.conf」](#) を参照してください。

### 3.1.4 標準フォントとアイコン・ファイル

この項では、Oracle Application Server Forms Services 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。

#### 3.1.4.1 Registry.dat

場所 : forms90/java/oracle/forms/registry

このファイルを使用すると、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 B.6 項「Registry.dat」](#) を参照してください。

## 3.2 アプリケーションの配布

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。Oracle Application Server Forms Services は、指定された URL によって Oracle Application Server 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、Oracle Application Server Forms Services の新規セッションごとに新しい Forms ランタイム・プロセス (Windows では ifweb90.exe、Solaris では f90webm) を起動します。

Forms Services の実行方法の詳細は、「[OracleAS Forms Services のアクション](#)」の項を参照してください。

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

.fmb は、Forms Developer 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。

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

Oracle Application Server Forms Services 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、Oracle Enterprise Manager Application Server Control の「Forms Web 構成」ページに構成セクションを作成します。

次の表は、“form=hrapp.fmx” という Forms モジュールを備えた “application” という名前のアプリケーションに対して構成する要素を示しています。

**表 3-1 新しい構成セクションのパラメータ値の例**

構成セクション名	アプリケーション名	Forms モジュール名の値
my_application	application	hrapp.fmx

構成が完了すると、ブラウザの URL に “...?config=my\_application” (formsweb.cfg の「Forms Web 構成」セクションの名前) と入力することにより、Web 上で Oracle Application Server Forms Services モジュール hrapp.fmx へアクセス可能になります。

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

1. .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 d:¥my\_files¥applications にある場合は、FORMS90\_PATH に d:¥my\_files¥applications 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）。その環境ファイルの「Forms 環境ファイルの編集」ページでこの情報を指定します。

環境ファイルを変更するには、Oracle Enterprise Manager の「環境」ページで環境ファイルを選択し、アプリケーションの必要に応じて環境変数を追加または編集します。たとえば、前述の例では次の環境変数を追加します。



表 3-2 新しい環境変数の値の例

環境変数の名前	環境変数の値
form	hrapp.fmx

新しい環境ファイルでこれらの環境変数を指定した場合は、それぞれの「Forms Web 構成」セクションでその環境ファイルを指定する必要があります。

1. 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。

構成セクションを作成したら、"config=" および構成セクション名を追加します。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=my_
application
```

### 3.2.2 パラメータの指定

Oracle Application Server Forms Services 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。

- Oracle Enterprise Manager Application Server Control の「Forms Web 構成」ページのデフォルト・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には、Oracle Application Server Forms Services で使用するデフォルト値が表示されます。

たとえば、Microsoft Internet Explorer 5.x 以上で 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、「IE」の「値」列のパラメータを次のように編集します。

```
native
```

「適用」をクリックします。

- 名前を付けたアプリケーションの構成セクションでは、他のシステム・パラメータとユーザー・パラメータの値を管理（追加、編集、削除）できます（「[Oracle Enterprise Manager での構成セクションの作成](#)」の項を参照）。たとえば、my\_application に対して作成する構成セクションで、次のパラメータとその値を追加または変更できます。

**表 3-3 構成セクションの例 : my\_application のパラメータ値**

パラメータ名	パラメータ値
baseHTML	mybase.htm
baseHTMLjinitiator	mybasejini.htm
baseHTMLjpi	mybasejpi.htm
baseHTMLie	mybaseie.htm
form	myapp.fmx
userid	

**注意：** Forms Web 構成の名前を付けた構成セクション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。

### 3.2.3 Oracle Enterprise Manager での構成セクションの作成

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、Oracle Application Server Forms Services アプリケーションのパラメータを指定できます。Forms Web 構成のデフォルト・セクションで使用可能なアプリケーションおよびシステム・パラメータを任意で指定できます。

たとえば、lookAndFeel パラメータを oracle の値に設定し、「適用」をクリックすると、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。

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、アプリケーションへの接続情報を scott/tiger@orcl として事前定義するには、userid のパラメータ値を設定する必要があります。これには、名前を付けた構成セクションで userid のパラメータ値を scott/tiger@orcl に変更します。

編集可能な他のパラメータは、「デフォルトの Forms 構成パラメータ」の項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。

---

---

### 3.2.3.1 Oracle Application Server Forms Services アプリケーションへのアクセスに使用する URL の編集

Oracle Application Server Forms Services 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コールすることができます。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" をコールしています。

---

---

**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。詳細は、[第 4.6 項「アプリケーションの URL セキュリティの管理」](#) を参照してください。

---

---

## 3.2.4 ランフォーム・パラメータにおける特殊文字の指定

ランフォーム・パラメータに渡される値に特殊文字が含まれている場合は、特定の考慮事項が適用されます。この項では、こうした考慮事項について説明し、今回のリリースのデフォルト動作と前リリースのデフォルト動作を比較します。

---

---

**注意：** ランフォーム・パラメータは、テンプレート HTML ファイルの `serverArgs` アプレット・パラメータで指定されるパラメータです。テンプレート HTML ファイルの `serverArgs` パラメータで指定された値は、変数の置換後に、コマンドライン・パラメータ文字列と呼ばれることもあります。この値は、空白で区切られた一連の `name=value` のペアで構成されます。この名前には、英数字またはアンダースコアのみが使用可能です。`name=value` ペアの値の部分には、任意の文字列を使用できます。

---

---

### 3.2.4.1 現行リリースのデフォルト動作

ランフォーム・パラメータの値は、次に示す3つの場所のいずれかで指定できます。

1. テンプレート HTML ファイル (base.htm など) の `serverArgs` パラメータの値。
2. (1) の変数参照を (直接または再帰的に) 置換する、構成ファイル (formsweb.cfg など) で指定された変数の値。通常、これらの値は **Application Server Control** を使用して管理します。第 4.3 項「Forms Services の構成」を参照してください。
3. (1) または (2) の変数参照を直接置換する、URL の属性値。

(3) では、URL 構文規則 (ブラウザおよびアプリケーション・サーバーで実行される) により、URL エスケープ・シーケンスとして特定の文字を入力する必要があります (計 3 文字のうち、'%' に続く 2 桁の 16 進数字は文字の ASCII 値を表します)。

この要件には、% 文字自体が含まれます (%25 として入力する必要があります)。さらに、現在 **Oracle Application Server Forms Services** では、ブラウザおよびアプリケーション・サーバーでエスケープなしに引用符を入力できる場合でも、引用符 (") は %22 として入力する必要があります。

URL 構文規則では、(URL エスケープ・シーケンス %20 の代替として) 空白を + として入力することも許可されています。ただし、`otherparams` 構成パラメータの値では、+ は特別に扱われます。+ は、ランフォーム・パラメータの値に埋め込まれた空白を示すのとは対照的に、`name=value` ペアを分割します。

たとえば、ランフォーム・アプリケーションにユーザー・パラメータ `param1` と `param2` が指定されており、これらのパラメータに 'a b' および 'c d' を割り当てる場合は、次の文字列を URL に取り込みます。

```
&otherparams=param1=a%20b+param2=c%20d
```

(1) および (2) のように、構成ファイルのテンプレート HTML ファイルでランフォーム・パラメータを指定する場合、Forms では状況によって URL エスケープ・シーケンスが必要とされる場合や URL エスケープ・シーケンスの使用が許可される場合、また許可されない場合があります。

ランフォーム・パラメータの値の外部では、URL エスケープ・シーケンスは使用できません。たとえば、`name=value` ペアの = は、常にそのまま = として指定する必要があります。隣接する 2 つの `name=value` ペアを区切る空白は、常にそのまま " " (シングル・スペース文字) として指定する必要があります。

ランフォーム・パラメータの値の内部では、空白 (' ') と引用符 (") をそれぞれ URL エスケープ・シーケンス (%20 および %22) として指定する必要があります。HTML デリミタ文字 (構成ファイルで指定) も URL エスケープ・シーケンスとして指定する必要があります。

その他の 7 ビット ASCII 文字も、URL エスケープ・シーケンスとして指定する場合があります。ただし、この指定は必須ではありません (下記に示すとおり、% では必要な場合があります)。% 文字には、特定の追加制限が適用されます。

HTML デリミタが % (デフォルト) である場合、ランフォーム・パラメータの値内の % の出現をエスケープする (%25 として指定する) 必要があります (HTML デリミタ文字のエスケープの必要性については、上記に記載されています)。さらに、7 ビットの ASCII 値を表す 2 桁の 16 進数字で変数名を開始することはできません。

つまり、変数名は最初の文字が 0 ~ 7 の 2 桁の 16 進数字で開始できません。さらに換言すると、変数名は 8 進数字とそれに続く 16 進数字では開始できません。

HTML デリミタが % でなければ、その直後に 8 進数字と 16 進数字が続く場合、% の出現はエスケープする必要があります。%' の他の出現もエスケープすることをお勧めしますが、これは必須ではありません。

(前リリースで作成した既存のテンプレート HTML ファイルまたは構成ファイルを所有し、そのファイルで '%' 以外の HTML デリミタが使用され、ランフォーム・パラメータの値に '%' が含まれている場合は、この推奨事項を無視することもできます)。

### 3.2.4.2 前リリースの動作

前リリースでは、テンプレート HTML ファイルまたは構成ファイルで指定されたランフォーム・パラメータの値に URL エスケープ・シーケンスを使用できませんでした (前述の (1) と (2) の場合)。3 つのすべての場合において、特定の特殊文字、特に空白、引用符、アポストロフィを指定することは困難または不可能でした。さらに、パラメータ値には、値をランフォームに渡す前に特定の変換が適用されました。特に顕著なのは値がアポストロフィで開始および終了する場合で、通常これらのアポストロフィは削除されました。ただし、こうした変換は明確に定義されたものではなく、Web 環境とクライアント / サーバー環境の間には差異がありました。

### 3.2.4.3 現行リリースでの前リリースの動作取得

使用しているアプリケーションが前リリースの動作に依存している場合は、構成ファイルで `escapeparams` 変数の値を `False` に設定すれば、現行リリースで前リリースの動作を取得できます (この作業は Oracle Enterprise Manager で実行します)。

以前の動作の取得を選択したアプリケーションに限定する場合は、異なる構成セクションの `escapeparams` 変数に異なる値を指定できます。以前の動作を必要とするアプリケーションでは、`escapeparams` 変数が `False` に設定されている構成セクションを指定できます。新しい動作を必要とする (または許容する) アプリケーションでは、`escapeparams` 変数が `True` に設定されている構成セクションを指定できます。

### 3.2.4.4 テンプレート HTML ファイルの考慮事項

独自のテンプレート HTML ファイルを作成する場合 (または `base.htm` のような既存のテンプレート HTML ファイルを変更する場合) は、次の点に留意してください。

`escapeparams` 変数 ('%' が HTML デリミタ文字の場合は文字列 `%escapeparams%`) への参照が `serverArgs` アプレット・パラメータ値の最初の部分に表示され、空白が続くように設定することをお勧めします (例については、付属の `base.htm` ファイルを参照してください)。

escapeparams 変数への参照が、テンプレート HTML ファイルの他の部分に表示されないようご注意ください。

escapeparams 変数への参照は serverArgs アプレット・パラメータ値の最初の部分から省略できますが、その場合は、escapeparams 変数の構成ファイルで指定されている値に關係なく、前リリースの動作が常に取得されます。

### 3.2.4.5 静的 HTML ページの考慮事項

静的 HTML を使用してランフォーム・エンジンを呼び出し、新しい動作を取得する場合は、特定の手順に従う必要があります。

基本的な規則として、静的 HTML が Forms Servlet で生成された HTML のように見える必要があります。具体的には、serverArgs アプレット・パラメータの値を文字列 escapeparams=true で開始する必要があります (大 / 小文字の区別なし)。

さらに、serverArgs アプレット・パラメータ値の各 name=value ペアの値部分で、表 3-4 に示す URL エスケープ・シーケンスによって特定の文字を指定する必要があります。

**表 3-4 静的 HTML ページの URL エスケープ・シーケンス**

エスケープが必要な文字	URL エスケープ・シーケンス	
改行	'\n'	%0a
空白	' '	%20
引用符	'\"'	%22
パーセント	'%'	%25
アポストロフィ	'\''	%27
左カッコ	'('	%28
右カッコ	')'	%29

name=value ペアの値部分に使用されている他の 7 ビット ASCII 文字をエスケープすることもできます。

静的 HTML での serverArgs アプレット・パラメータの書式の例を次に示します。この例は、"my form" という名前のフォームに関するものです (引用符は含みません)。ここでは、値 "foo'bar" (引用符は含みません) がユーザー定義パラメータ myparam に渡されます。

```
<PARAM NAME="serverArgs" VALUE="escapeparams=true module=my%20form
userid=scott/tiger@mydb myparam=foo%27bar">
```

### 3.3 OracleAS 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、Forms Services が OracleAS 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、Oracle Application Server 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 Oracle Application Server Forms Services 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 Server Listener はリクエストを受信します。Oracle HTTP Server 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、そのリクエストを OC4J に転送します。
3. OC4J は、そのリクエストを Oracle Application Server Forms Services 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) そのリクエストを Forms Servlet にマップします。
4. Forms Servlet (OC4J で実行している) が、そのリクエスト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) リクエスト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 ファイルを読み取り、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かると、変数 (%myParam%) はそのパラメータの値で置換されます。

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。この例では、このテキストが値 "hrapp" と置換されます。

1. 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 社の Java Plug-in などのプラグ・イン JVM のいずれか) で実行されます。
2. 選択したベース HTML ファイルがプラグイン用 (Oracle JInitiator か Sun 社の JDK Java Plug-in) 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。
3. 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。アプレットの場所は、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](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) がネットワークを介してロードされるのは、ユーザーが Oracle Application Server Forms Services アプリケーションを最初に実行したとき (または、新しいバージョンの Oracle Application Server Forms Services が Web サーバーにインストールされたとき) のみです。それ以外の場合は、ローカル・ディスクの Web ブラウザ (または Java Plug-in) のキャッシュからロードされます。

4. Oracle Application Server Forms Services アプレットが実行されると、URL <http://mymachine.com:7777/forms90/190servlet> にある Forms Listener Servlet に接続して Forms セッションを起動します。
5. Oracle HTTP Server Listener はリクエストを受信します。Oracle HTTP Server Listener は、パス "/forms90/190servlet" 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、そのリクエストを OC4J に転送します。
6. Forms Listener Servlet (190servlet) 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。
7. 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 を介して継続します。



8. 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、(Application Server Control の「Forms Web 構成」ページで指定された) 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 つに格納されている必要があります。「Forms Web 構成」ページではディレクトリも指定できます（例：  
form=c:¥<path>¥myform）。

9. 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" 組み込み関数をコールする 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。

## 3.4 クライアント・ブラウザのサポート

Oracle 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）。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、Forms Developer の「ヘルプ」メニューで「OTN 上の Forms」を選択してください。

### 3.4.1 Oracle JInitiator

Oracle JInitiator は、Web ブラウザ内で動作し、Sun 社の 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。

Oracle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。

### 3.4.2 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。

**表 3-5 Web ブラウザと各ブラウザに対応するベース HTML ファイル**

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*	native**	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

1. ベース HTML ファイル内の変数 (`%variablename%`) が、FormsServlet.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や、URL リクエスト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。
2. HTML ファイルをユーザーのブラウザに送信します。

---

---

# Oracle Enterprise Manager による Forms Services の構成

この章には、次の項が含まれています。

- Oracle Application Server Forms Services での Forms アプリケーションの起動方法
- Oracle Enterprise Manager と Oracle Forms
- Forms Services の構成
- Oracle Enterprise Manager による環境変数の構成
- ユーザー・セッションの管理
- アプリケーションの URL セキュリティの管理
- ユーザー独自のテンプレート HTML ファイルの作成
- Oracle Forms アプリケーションへの Graphics の挿入
- Forms Services で使用するアイコンとイメージの配置
- 言語検出の有効化

## 4.1 Oracle Application Server Forms Services での Forms アプリケーションの起動方法

ユーザーが最初に Oracle 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 によって読み取られます。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 リクエスト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、Oracle Enterprise Manager Application Server Control で容易に構成ファイルを変更できます。

## 4.2 Oracle Enterprise Manager と Oracle Forms

Forms Services に付属している Oracle Enterprise Manager Application Server Control のユーザー・インタフェースは、デフォルトのブラウザから起動する Web ベースのツールです。デフォルトの URL は次のとおりです。

`http://<machine.domain>:1810`

Forms Services では、Web ベースの Oracle Enterprise Manager Application Server Control を使用して、次の操作を実行できます。

- Forms Services インスタンスのメトリックを監視する。詳細は、[第 8.1.1.1 項「Forms Services インスタンスの監視」](#)を参照。
- ユーザー・セッションのメトリックを監視する。詳細は、[第 8.1.1.3 項「ユーザー・セッションのメトリックの監視」](#)を参照。
- 新しいユーザー・セッションを許可または拒否する。詳細は、[第 4.5.1 項「新規ユーザー・セッションの許可」](#)および[第 4.5.2 項「新規ユーザー・セッションの禁止」](#)を参照。
- ユーザー・セッションを終了する。詳細は、[第 4.5.3 項「Forms Services インスタンスでのユーザー・セッションの終了」](#)を参照。
- Forms Services インスタンスのパラメータを構成する。詳細は、[第 4.3.1 項「Application Server Control によるパラメータの構成」](#)を参照。
- Forms Trace を構成してトレースのメトリックを監視する。詳細は、[第 7.1.1 項「Forms Trace の構成」](#)および[第 7.1.6 項「Forms Services のトレース・メトリックの監視」](#)を参照。
- 複数の環境ファイルを構成する。詳細は、[第 4.4 項「Oracle Enterprise Manager による環境変数の構成」](#)を参照。
- 利用可能な Forms Services のユーティリティとランタイム・プーリングを使用する。詳細は、[第 8.1.3 項「Forms Services のユーティリティ」](#)および[第 8.2 項「OracleAS Forms Services アプリケーションのチューニング」](#)を参照。

## 4.2.1 Forms セッション管理のための Oracle Enterprise Manager の使用方法

デフォルトでは、Oracle Enterprise Manager は Forms に関する情報の一部を提供し、構成ファイルを中央で変更できるようにします。ただし、Oracle Enterprise Manager によって Forms に提供される機能をフルに活用するには、次の作業を実行する必要があります。

1. Forms 構成ファイル (formsweb.cfg) のデフォルト・セクションに、次の変数が設定されていることを確認します。この作業は、OEM を使用しても、サーバー上で手動で実行してもかまいません。

```
em_mode=1
```

これで、OEM により、実行中の各 Forms アプリケーションに関するユーザー情報が表示されます。em\_mode を 1 に設定した後で作成されたセッションのみが表示されます。デフォルトでは、この値は 0 (オフ) になっています。

2. Forms 構成ファイル (formsweb.cfg) に次の変数が設定されていることを確認します。この変数は、デフォルト・セクションと特定のアプリケーションのセクションのどちらでも設定できます。ステップ 1 と同様に、この変数は、OEM を使用して設定しても、サーバー上で手動で設定してもかまいません。

```
allow_debug=true
```

この変数によって、OEM 内のトレースのオンとオフを切り替えることができます。

3. **(Windows のみ)** Oracle Application Server をインストールしている中間層ユーザーに対して、「バッチ ジョブとしてログオン」権限を与える必要があります。この権限を持つユーザーまたは管理者権限を持つユーザーとしてログインします。「コントロールパネル」で「管理ツール」を選択します。次に、「ローカルセキュリティ ポリシー」→「ローカル ポリシー」→「ユーザー権利の割り当て」を選択します。Oracle Application Server をインストールしたユーザーのユーザー名を追加します。
4. **(Windows のみ)** Oracle Application Server をインストールしたユーザーまたは管理者権限を持つユーザーとして、「コントロールパネル」にある「サービス」を開きます。Oracle/xxxxxx/ProcessManager サービスを探します。これを右クリックして、「プロパティ」を選択します。「ログオン」タブで、「デスクトップとの対話をサービスに許可」が選択されていることを確認します。
5. **(Windows のみ)** このサービスを再起動する必要があります。サービスの再起動後も、OEM で変更が有効になるまでに数分かかる場合があります。

## 4.2.2 Forms Services の管理に使用する Oracle Enterprise Manager Grid Control の構成

Forms Services のインストール時に、Oracle Enterprise Manager Grid Control の targets.xml ファイルが Oracle Universal Installer によって自動的に編集されます。targets.xml ファイルには、Oracle Enterprise Manager で管理する全サービスのリストが格納されています。

Forms Services を監視するために初めて Oracle Enterprise Manager を使用する際は、監視する Forms Services インスタンスごとに次の手順を実行する必要があります。

Application Server Control を使用してノードの「Enterprise Manager 管理」ページにアクセスする方法は、Oracle Enterprise Manager のマニュアルを参照してください（管理者のユーザー名とパスワードを入力する必要があります）。

**Forms Services の管理に使用する Oracle Enterprise Manager Grid Control を構成するには：**

1. エージェント管理ページでは、監視対象のすべてのサービスが「エージェントの監視中のターゲット」ヘッダーの下に表示されます。
2. Oracle Enterprise Manager で構成する Forms インスタンスの横にあるラジオ・ボタンを選択します。
3. 「編集」をクリックします。
4. Forms インスタンスの ORACLE\_HOME と URL を指定します。
5. 「OK」をクリックします。

---

---

**注意：** このページで実行可能な他の操作の詳細は、Oracle Enterprise Manager のヘルプ・システムを参照してください。

---

---

## 4.2.3 Application Server Control による Forms Services へのアクセス

Application Server Control を使用して Forms Server の大部分の管理タスクを実行するには、最初に Application Server Control で Forms Server の「Forms」ホームページにナビゲートします。

**Application Server Control で Forms Server の「Forms」ホームページにナビゲートするには：**

1. Application Server Control を使用して、管理する Forms Server が含まれるアプリケーション・サーバーのホームページにナビゲートします。

Oracle Enterprise Manager Application Server Control の使用方法の予備知識は、『Oracle Application Server 10g 管理者ガイド』の「管理ツールの概要」を参照してください。

2. アプリケーション・サーバーのホームページの「システム・コンポーネント」セクションで、管理する Forms Server のリンクをクリックします。これにより、Forms Server の「Forms」ホームページが Application Server Control に表示されます。

## 4.3 Forms Services の構成

Forms Services を構成するには、Application Server Control の「構成」ページを使用します。このページでは、formsweb.cfg ファイルの変更がすべて管理されます。

---

---

**注意：** 構成ファイルや環境ファイルを手動で編集する場合は、Oracle Enterprise Manager がすべての変更を読み取れるように、Oracle Enterprise Manager およびすべての Distributed Configuration Management (DCM) プロセスを再起動する必要があります。Oracle Enterprise Manager および DCM プロセスを再起動しない場合、これらのファイルに手動で加えた変更は、その後 Oracle Enterprise Manager で加えた変更によって上書きされます。これらの DCM プロセスは次のとおりです。

- emctl stop em
- dcmctl stop
- opmnctl stopall
- opmnctl startall
- dcmctl start
- emctl start em

---

---

**注意：** formsweb.cfg および default.env ファイルを Oracle Enterprise Manager で編集する前に、これらのファイルのバックアップをとってください。

---

---

### Forms Services を構成するには：

1. Application Server Control を起動します。
2. Application Server Control のメイン・ページで、構成する Oracle Forms Services インスタンスへのリンクを選択します。
3. Forms Services インスタンスで、「構成」タブを選択します。

4. 「表示」プルダウン・リストから「Forms Web 構成」を選択し、「実行」をクリックします。
  - formsweb.cfg ファイルに新しいセクションを作成するには、「新規セクションの作成」をクリックし、このセクションの名前を次のページで入力します。
  - formsweb.cfg ファイルからセクションを削除するには、削除するセクションの横にあるラジオ・ボタンを選択して「削除」をクリックし、次のページで削除を確認します。

---

**注意：** ほとんどの Web アプリケーションでは、ページを移動するだけで、保存されていない変更は簡単に失われてしまいます。Application Server Control で Forms の構成ファイルまたは環境ファイルに変更を加えた場合は、必ず保存してから他のページへ移動してください。

変更の保存に必要な時間は、変更を加えた行数によって異なります。たとえば、50 行のコメントを追加した変更では、単一のエントリを削除した変更よりも保存に時間がかかります。

---

## 4.3.1 Application Server Control によるパラメータの構成

Forms Servlet の構成ファイル (formsweb.cfg) の説明と場所は、[第 3.1.1.2 項「formsweb.cfg」](#) を参照してください。

### 4.3.1.1 ファイルを指定するパラメータ

4 つのベース HTML パラメータは適切なファイルを示している必要があります。通常、次の値とそのパラメータがデフォルトの構成セクションに表示されます。

**表 4-1 ファイルを指定するデフォルトの構成パラメータ**

パラメータ	値
baseHTML	base.htm
baseHTMLie	baseie.htm
baseHTMLJinitiator	basejini.htm
baseHTMLjpi	basejpi.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 (この例のように)、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。



## 4.3.2 構成セクションの管理

Application Server Control の「構成」タブで新しい構成セクションを作成します。これにより、名前を付けた構成が formsweb.cfg ファイルに作成されます。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 問合せ文字列からリクエストできます。

### 新しい構成セクションを作成するには：

1. Oracle Enterprise Manager Application Server Control を起動します。
2. Application Server Control のメイン・ページで、構成する Forms Services インスタンスへのリンクを選択します。
3. Forms Services インスタンスで、「構成」タブを選択します。
4. 「構成」タブの上部にある「新規セクションの作成」をクリックします。  
「Forms 新規セクション名」ページが表示されます。
5. 新しい構成の名前を入力し、「OK」をクリックします。
6. 新しいセクションの説明を入力する場合は、セクションの編集とパラメータの追加を行う前に「適用」をクリックして説明を保存してください。

たとえば、汎用のルック・アンド・フィールドを備えた別のブラウザ・ウィンドウで Forms を実行する構成を作成するには、新しいセクションを作成し、次の表 4-2 のパラメータを追加します。

**表 4-2 新しい構成セクションに追加するサンプル・パラメータ**

パラメータ	値
forms	< モジュール >
separateFrame	TRUE
lookandfeel	Generic

たとえば、ユーザーが "sepwin"（または他の任意の名前）構成を使用するフォームを起動するには、次の URL を入力します。

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

また、Application Server Control 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 7 章「[トレースと診断](#)」を参照）。

特別な構成のその他の例は、[付録 B「サンプル構成ファイル」](#) を参照してください。

#### 4.3.2.1 名前を付けた構成の複製

バックアップ用に、名前を付けた構成のコピーを作成できます。または、複製から新しい構成セクションを作成できます。

**名前を付けた構成を複製するには：**

1. 複製するセクションの横にあるラジオ・ボタンを選択します。
2. 「複製」をクリックします。
3. 次のページで、複製したセクションの一意の名前を新規に入力し、「OK」をクリックします。

複製元のセクションとまったく同じパラメータ、パラメータ値およびコメントを備えた新しいセクションが作成されます。

#### 4.3.2.2 名前を付けた構成の削除

名前を付けた構成を削除すると、構成内の情報はすべて削除されます。特定のパラメータのみを削除する場合は、セクション名の横にあるラジオ・ボタンを選択して「削除」をクリックします。

**名前を付けた構成を削除するには：**

1. Oracle Enterprise Manager Application Server Control を起動します。
2. Application Server Control のメイン・ページで、構成する Forms Services インスタンスへのリンクを選択します。
3. 「構成」タブで、削除する構成セクションの横にあるラジオ・ボタンを選択します。
4. 「削除」をクリックします。  
「確認」ページが表示されます。
5. 「OK」をクリックします。

構成セクションが削除されます。

Application Server Control で「構成」タブが再度表示され、ページの下部に残りの構成が示されます。

### 4.3.3 パラメータの管理

名前を付けた構成内のパラメータを管理するには、Application Server Control を使用します。Application Server Control の「編集セクション」ページで、パラメータの追加、編集または削除を行うことができます。

#### 構成セクションでパラメータを編集するには：

1. Oracle Enterprise Manager Application Server Control の「確認」タブで、パラメータを編集する構成セクションの横にあるラジオ・ボタンを選択します。
2. このページの上部にある「編集」をクリックします。  
選択した構成の「編集セクション」ページが表示されます。
3. 編集するパラメータの横にあるラジオ・ボタンを選択します。
4. テキスト・フィールドの値を変更します。
5. 「適用」をクリックします。  
変更内容が保存されます。

#### 構成にパラメータを追加するには：

1. Application Server Control の「構成」タブで、パラメータを追加する構成セクションの横にあるラジオ・ボタンを選択します。
2. このページの上部にある「編集」をクリックします。  
選択した構成の「編集セクション」ページが表示されます。
3. 新しいパラメータの名前と値を入力し、「新規パラメータの追加」をクリックします。  
「編集セクション」ページが更新され、新しいパラメータが表示されます。
4. 新しいパラメータの説明を追加し、「適用」をクリックします。
5. 「Forms」ページに戻るには、軌跡リンクで「Forms」をクリックします。

#### 構成からパラメータを削除するには：

1. 構成セクションを編集するには、構成セクションの横にあるラジオ・ボタンを選択し、このページの上部にある「編集」をクリックします。  
選択した構成の「編集セクション」ページが表示されます。
2. 削除するパラメータの横にあるラジオ・ボタンを選択します。
3. 「削除」をクリックします。
4. 表示される「確認」ページで削除を確認します。  
構成セクションからパラメータが削除されます。

## 4.3.4 デフォルトの Forms 構成パラメータ

表 4-3 「システムのデフォルトの構成パラメータ」は、formsweb.cfg ファイルにあるデフォルトの Forms 構成パラメータの説明です。シングル・サインオン・パラメータの詳細は、第 6.3 項「アプリケーションでの SSO の有効化」を参照してください。

これらの項には次のものが含まれます。

- システムのデフォルトの構成パラメータ
- ランフォーム・パラメータ (serverArgs パラメータ)
- 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
- アプレット・パラメータまたはオブジェクト・パラメータ
- JInitiator のパラメータ
- Sun 社の Java Plug-in のパラメータ
- Oracle Enterprise Manager の構成パラメータ
- OID (Oracle Internet Directory) の構成パラメータ

### 4.3.4.1 システムのデフォルトの構成パラメータ

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。

表 4-3 システムのデフォルトの構成パラメータ

パラメータ	必須 / 任意	パラメータ値
baseHTML	必須	デフォルトのベース HTML ファイル。
baseHTMLjinitiator	必須	JInitiator のタグを含む HTML ファイルへの物理パス。
connectionDisallowedURL	任意	新しいセッションを起動できない HTML ページに表示される URL です。
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。
HTMLdelimiter	必須	変数名のデリミタ。デフォルトで % になります。

表 4-3 システムのデフォルトの構成パラメータ (続き)

パラメータ	必須 / 任意	パラメータ値
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。
defaultcharset	任意	<p>サーブレットのリクエストとレスポンスで使用するキャラクタ・セットを指定します。デフォルトは ISO-8859-1 (別名 Latin-1) です。サーブレットのリクエストでキャラクタ・セットが (POST のコンテンツタイプ・ヘッダーなどで) 指定されている場合は無視されます。</p> <p>このパラメータの値は、IANA キャラクタ・セット名 (SHIFT_JIS など) または Oracle キャラクタ・セット名 (JA16SJIS など) として指定できます。このキャラクタ・セットは、NLS_LANG 環境変数で指定されているキャラクタ・セットと一致し、さらにブラウザに表示可能である必要があります。また、URL エスケープ・シーケンスとは対照的に、IME 対応など、マルチバイト・キャラクタを URL に直接入力できるブラウザを使用しており、この機能をエンド・ユーザーが利用できるようにする場合、このパラメータの値は、入力文字のバイト・シーケンスへの変換時にブラウザで使用されるキャラクタ・セットと一致している必要があります。</p> <p><b>注意:</b> 7 ビット ASCII 文字以外の文字が名前で使用されている構成セクションが構成ファイルにある場合は、次のルールが適用されます。config パラメータが URL で指定され、または文字セットが指定されていない POST リクエストのボディで指定され、そのパラメータ値に 7 ビット ASCII 文字以外の文字が含まれている場合、このパラメータ値は defaultcharset パラメータの値から名前が導出されるキャラクタ・セットによって解釈されます。ただし、defaultcharset パラメータでは、構成ファイルの言語依存のデフォルト・セクションと言語非依存のデフォルト・セクションのみが検索されます。構成セクションは、名前がまだ判明していないため検索されません。</p>
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。
log	任意	Builder でのフォームの実行およびデバッグをサポートします。デフォルト値は NULL です。

### 4.3.4.2 ランフォーム・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。ランフォーム・パラメータ値に指定された特定の特殊文字をランフォームで処理する方法については、[第 3.2.4 項「ランフォーム・パラメータにおける特殊文字の指定」](#)を参照してください。

表 4-4 ランフォーム・パラメータ (serverArgs パラメータ)

パラメータ	必須 / 任意	パラメータ値
escapeparams	任意	9.0.4 より前のリリースで行っていたように、ランフォーム・パラメータで特殊文字を扱う場合は、このパラメータを <code>false</code> に設定します。
heartBeat	任意	このパラメータは、クライアントが実行中であることを示すために、クライアントがサーバーにパケットを送信する頻度を設定します。分または分の小数部 (例:30 秒は 0.5) でこの整数値を定義します。デフォルトは 2 分です。  heartbeat の値が FORMS90_TIMEOUT の値より少ない場合は、ユーザーがフォームを積極的に使用していない場合でも、ユーザーのセッションは維持されます。
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。例:scott/tiger@ORADB
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。  otherparams=buffer_records=%buffer% debug_messages=%debug_messages% array=%array% obr=%obr% query_only=%query_only% quiet=%quiet% render=%render% record=%record% tracegroup=%tracegroup% log=%log% term=%term%  注意: このパラメータを URL で指定するときは、特別な構文規則が適用されます。複数の name=value ペアを分割するために + が使用される場合があります (詳細は <a href="#">第 3.2.4 項「ランフォーム・パラメータにおける特殊文字の指定」</a> を参照してください)。  本番環境では、エンド・ユーザーが URL で指定可能なランフォーム・パラメータの制御を強化するために、restrictedURLparams パラメータを使用します。
debug	任意	デバッグ・モードでの実行を許可します。  デフォルト値は NO です。

表 4-4 ランフォーム・パラメータ (serverArgs パラメータ) (続き)

パラメータ	必須 / 任意	パラメータ値
buffer	任意	Builder でのフォームの実行およびデバッグをサポートします。 otherparams の第 2 引数。 デフォルト値は NO です。
debug_messages	任意	Builder でのフォームの実行およびデバッグをサポートします。 otherparams の第 2 引数。 デフォルト値は NO です。
allow_debug	任意	TRUE に設定すると、forms90/f90servlet/admin 画面の管理機能がすべてアクティブになります。 forms90/f90servlet/xlate は、指定されたトレース・ファイルで Forms Trace Xlate を実行します。「ユーザー・セッション」画面にトレース・ログを表示できるようにするには、このパラメータを TRUE に設定する必要があります。  デフォルト値は FALSE です。管理機能を試行すると、test.fmx アプリケーションが実行されます。
array	任意	Builder でのフォームの実行およびデバッグをサポートします。 デフォルト値は NO です。
query_only	任意	Builder でのフォームの実行およびデバッグをサポートします。 デフォルト値は NO です。
quiet	任意	Builder でのフォームの実行およびデバッグをサポートします。 デフォルト値は YES です。
render	任意	Builder でのフォームの実行およびデバッグをサポートします。 デフォルト値は NO です。
host	任意	Builder でのフォームの実行およびデバッグをサポートします。 デフォルト値は NULL です。
port	任意	Builder でのフォームの実行およびデバッグをサポートします。 デフォルト値は NULL です。
record	任意	Builder でのフォームの実行およびデバッグをサポートします。 デフォルト値は NULL です。
tracegroup	任意	Builder でのフォームの実行およびデバッグをサポートします。 デフォルト値は NULL です。

表 4-4 ランフォーム・パラメータ (serverArgs パラメータ) (続き)

パラメータ	必須 / 任意	パラメータ値
log	任意	Builder でのフォームの実行およびデバッグをサポートします。 デフォルト値は NULL です。
term	任意	Builder でのフォームの実行およびデバッグをサポートします。 デフォルト値は NULL です。
em_trace	内部使用のみ	

#### 4.3.4.3 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ない場合があります。

表 4-5 HTML ページ・パラメータ

パラメータ	必須 / 任意	パラメータ値
pageTitle	任意	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。
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のページに追加する 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側のページに追加する HTML コンテンツ。



#### 4.3.4.4 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。次に例を示します。<PARAM NAME="serverURL" VALUE="%serverURL%">

表 4-6 アプレット・パラメータまたはオブジェクト・パラメータ

パラメータ	必須 / 任意	パラメータ値
serverURL	必須	/forms90/190servlet (第1章の「 <a href="#">Forms Listener Servlet</a> 」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。
imageBase	任意	アイコン・ファイルの格納場所を示します。次のいずれかを選択します。 <ul style="list-style-type: none"> <li>codeBase。アイコンの検索パスが Java クラスの格納先ディレクトリを基準とした相対パスであることを示します。アイコンを JAR ファイルに格納する場合はこの値を使用します (推奨)。</li> <li>documentBase。デフォルト値です。Forms Server CGI を利用する配置では、カスタム・アプリケーション・ファイルでアイコン・パスを指定する必要があります。</li> </ul>
logo	任意	Forms メニュー・バーに表示する .GIF ファイルを指定します。ロゴなしの場合は NO に設定します。デフォルトの Oracle ロゴを使用する場合は空白のままにします。
restrictedURLparams	任意	ユーザーが特定のパラメータを URL で使用することを制限するために管理者が指定します。パラメータが複数ある場合は、パラメータをカンマで区切る必要があります。restrictedURLparams 自体をこのパラメータ、つまり restrictedURLparams の値にすることはできません。  デフォルト値は、HTMLbodyAttrs,HTMLbeforeForm,pageTitle,HTMLafterForm,log,allow_debug,allowNewConnections です。
formsMessageListener	任意	Forms アプレット・パラメータ。
recordFileName	任意	Forms アプレット・パラメータ。
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。

表 4-6 アプレット・パラメータまたはオブジェクト・パラメータ (続き)

パラメータ	必須 / 任意	パラメータ値
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  <b>注意:</b> lookAndFeel が Generic に設定されている場合、colorScheme は無視されます。
serverApp	任意	必要に応じて、アプリケーション用のファイル名でデフォルト値を置き換えます。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。  パラメータを設定するには、ファイル名を記述するか (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。
archive	任意	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)。  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。
archive_jinit	任意	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)。  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。

表 4-6 アプレット・パラメータまたはオブジェクト・パラメータ (続き)

パラメータ	必須 / 任意	パラメータ値
archive_ie	任意	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 f90a11.cab)。
networkRetries	任意	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・エンジンにリクエスト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。
mapFonts	任意	<PARAM NAME = "mapFonts" VALUE = "yes" > でフォント・マッピングをトリガーします。  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。

#### 4.3.4.5 JInitiator のパラメータ

表 4-7 JInitiator のパラメータ

パラメータ	必須 / 任意	パラメータ値
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/JInitiator/jinit.download.htm です。
jinit_classid	必須 (IE のみ)	デフォルトは clsid:CAFECAFE-0013-0001-0013-ABCDEFABCDEF です。
jinit_exename	必須	デフォルトは jinit.exe#Version=1,3,1,13 です。
jinit_mimetype	必須 (Netscape のみ)	デフォルトは application/x-jinit-applet;version=1.3.1.13 です。
baseHTMLjinitiator	必須	JInitiator のタグを含む HTML ファイルへの物理パス。

### 4.3.4.6 Sun 社の Java Plug-in のパラメータ

表 4-8 Sun 社の Java Plug-in のパラメータ

パラメータ	必須 / 任意	パラメータ値
jpi_codebase	必須	Sun の Java Plug-in のコードベース設定。
jpi_classid	必須	Sun 社の Java Plug-in のクラス ID。
jpi_download_page	必須	Sun 社の Java Plug-in のダウンロード・ページ。

### 4.3.4.7 Oracle Enterprise Manager の構成パラメータ

表 4-9 Oracle Enterprise Manager の構成パラメータ

パラメータ	必須 / 任意	パラメータ値
em_mode	必須	1 (使用可能)。0 (使用不可)。 1 は、メトリックやサーブレットのステータスなどの Oracle Enterprise Manager 情報がすべて使用可能であることを示します。 0 は、構成情報のみが使用可能であることを示します。

### 4.3.4.8 OID (Oracle Internet Directory) の構成パラメータ

表 4-10 OID (Oracle Internet Directory) の構成パラメータ

パラメータ	必須 / 任意	パラメータ値
oid_formsid	必須	OracleAS 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	OracleAS 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。

## 4.4 Oracle Enterprise Manager による環境変数の構成

環境変数を管理するには、「Enterprise Manager Application Server Control」ページの「環境」タブを使用します。このページでは、必要に応じて環境変数の追加、編集または削除を行うことができます。

Forms ランタイム実行可能ファイル（Windows では ifweb90.exe、UNIX では f90webm）用の環境変数（PATH、<ORACLE\_HOME>、FORMS90\_PATH など）は、「環境」タブで定義します。Listener Servlet は、実行可能ファイルをコールして、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。

このページで定義されていない環境変数は、サブレット・エンジン（OC4J）から継承されます。「Forms Web 構成」ページの「デフォルト」セクションの envFile パラメータで環境ファイル名を指定する必要があります。

環境変数をカスタマイズする際には、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。

---

**注意：** 環境ファイルは、Oracle Enterprise Manager Application Server Control で作成または削除できません。環境ファイルは、.env 拡張子を指定して <ORACLE\_HOME>/forms90/server に手動で作成する必要があります。

同様に、環境ファイルは OEM から削除できません。Application Server Control で環境ファイルを新規に取得したり、削除した環境ファイルを非表示にするには、次の Oracle Enterprise Manager プロセスを再起動する必要があります。

- emctl stop em
  - emctl start em
- 

表 4-11 「デフォルトの環境変数」は、default.env で指定されている重要な環境変数の説明です。

表 4-11 デフォルトの環境変数

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 製品の実行可能ファイルを含みます。
FORM90_PATH	<ORACLE_HOME>%forms90 (デフォルト)	<p>実行するフォーム、メニューまたはライブラリを探すときに <b>Oracle Forms</b> で検索するパスを指定します。</p> <p><b>Windows</b> の場合、複数のパスはセミコロン (;) で区切ります。</p> <p><b>UNIX</b> の場合、複数のパスはコロン (:) で区切ります。</p>
FORMS90_TIMEOUT	<p>デフォルト: 15</p> <p>有効値: 3 ~ 1440 (1 日)</p> <p>例:</p> <p>FORMS90_TIMEOUT=1440</p>	<p>このパラメータは、クライアントと <b>Forms Services</b> 間の通信がないときに <b>Forms Services</b> プロセスが終了するまでの経過時間 (分) を指定します。</p> <p>クライアントの通信は、ユーザーが何らかの操作をすることによって生じます。または、ユーザーがフォームを積極的に使用していない場合は、<b>Forms</b> クライアントのハートビートによって生じます。</p>
TNS_ADMIN	<ORACLE_HOME>/network/admin	TNSNAMES.ORA、SQLNET.ORA などの TNS ファイルへのパス名を指定します。
CLASSPATH	<ORACLE_HOME>/jdk/bin/java	Forms デバッガに必要な Java クラス・パスを指定します。

表 4-11 デフォルトの環境変数 (続き)

環境変数	有効値	説明
REPORTS_CLASSPATH	<ORACLE_HOME>/jlib/zrclient.jar:<ORACLE_HOME>/reports/jlib/rwrun.jar	この設定は、Reports アプリケーションが Forms アプリケーションからコールされる場合にのみ必要です。
ORACLE_GRAPHICS6I_HOME	ORACLE_GRAPHICS6I_HOME=<GRAPHICS6I_HOME>	これらの設定は、Graphics アプリケーションが Forms アプリケーションからコールされる場合にのみ必要です。Oracle Enterprise Manager で Graphics アプリケーションを使用するための <ORACLE_HOME> の値を設定します。
LD_LIBRARY_PATH	<p>初回使用時、LD_LIBRARY_PATH 環境変数は次のように設定します。</p> <pre>&lt;ORACLE_HOME&gt;/lib</pre> <p>LD_LIBRARY_PATH をリセットするには、Bourne シェルで次のように入力します。</p> <pre>\$ set LD_LIBRARY_PATH=&lt;ORACLE_HOME&gt;/lib:\${LD_LIBRARY_PATH}</pre> <pre>\$ export LD_LIBRARY_PATH</pre> <p>または、C シェルで次のように入力します。</p> <pre>% setenv LD_LIBRARY_PATH &lt;ORACLE_HOME&gt;/lib:\${LD_LIBRARY_PATH}</pre>	Oracle Forms Developer および Reports Developer 製品では、ダイナミック・ライブラリまたは共有ライブラリが使用されています。そのため、LD_LIBRARY_PATH を設定して、ダイナミック・リンカーでライブラリが検出できるようにする必要があります。

---

**注意：** Windows では、Oracle Application Server Forms Services は、Oracle 環境設定が環境変数として設定されていない限り、Windows レジストリから Oracle 環境設定を読み取ります。

---

## 4.5 ユーザー・セッションの管理

Oracle Application Server Forms Services には、管理者によるユーザー・セッションの管理に役立つ次の機能が搭載されています。

- [新規ユーザー・セッションの許可](#)
- [新規ユーザー・セッションの禁止](#)
- [Forms Services インスタンスでのユーザー・セッションの終了](#)

### 4.5.1 新規ユーザー・セッションの許可

デフォルトでは、ユーザーは Forms セッションを新規に作成できます（青信号で明示）。Forms セッションを使用可能にした後でユーザーに作成を許可することもできます。

**新規 Forms ユーザー・セッションを許可するには：**

- Enterprise Manager Oracle Application Server Forms Services の「概要」ページで、「使用可能」（デフォルト）をクリックします。

信号が青になります。

### 4.5.2 新規ユーザー・セッションの禁止

**新規 Forms ユーザー・セッションを禁止するには：**

- Enterprise Manager Oracle Application Server Forms Services の「概要」ページで、「使用不可」をクリックします。

信号が黄色になります。

「使用不可」を押すと、`formsweb.cfg` ファイルのデフォルト・セクションに新しいパラメータが追加されます。このパラメータの名前は `allowNewConnections` で、「使用不可」を押すとその値が `FALSE` に設定されます。新規ユーザー・セッションが無効の場合、接続の試行は `formsweb.cfg` パラメータ `connectionDisallowedURL`（デフォルト・セクション）で識別された URL へ向けられます。次に、例を示します。

```
connectionDisallowedURL=www.oracle.com
```

```
connectionDisallowedURL=http://www.oracle.com
```

`connectionDisallowedURL` を指定しない場合は、次のメッセージがブラウザに表示されます。

```
The Forms Servlet will not allow new connections. Please contact your System Administrator.
```

ただし、新規ユーザー・セッションの無効化は既存の Forms セッションには適用されず、OC4J インスタンスは起動したままになります。



### 4.5.3 Forms Services インスタンスでのユーザー・セッションの終了

1. Oracle Enterprise Manager Application Server Control を起動します。
2. 終了するユーザー・セッションを含む Forms Services インスタンスへのリンクを選択します。
3. Forms Services インスタンスの「概要」ページで、「セッションの詳細」リンクを選択します。
4. 削除するユーザー・セッションの横にあるラジオ・ボタンをクリックします。
5. 「停止」をクリックします。
6. 表示されるダイアログ・ボックスに接続情報を入力します（必要なユーザー名とパスワードは Forms Services のインストールで使用したものと同じです）。
7. 「OK」をクリックします。

ユーザー・セッションが削除され、ランフォーム・インスタンスが終了します。

## 4.6 アプリケーションの URL セキュリティの管理

Oracle Forms アプリケーションは、ユーザーがブラウザでアクセスする Web 配布のソリューションです。Oracle Forms アーキテクチャでは、Forms 開発者は 2 種類の方法で Forms アプリケーションの実行方法を構成できます。ひとつは、パラメータとその値を URL で設定する方法です。もうひとつは、パラメータとその値を構成ファイル (formsweb.cfg) で設定する方法です。URL で設定したパラメータは、formsweb.cfg で設定したパラメータよりも優先されます。

---

**注意：** restrictedURLparams パラメータは、Oracle Enterprise Manager Application Server Control の「構成」ページで管理します。

---

Forms 管理者はこのデフォルト動作を上書きして、URL で使用可能なパラメータを完全に制御できます。

URL で使用可能にするパラメータを決定する際は、2 つのシナリオを検討します。最初のシナリオは、管理者が、単にデフォルトのログイン・ウィンドウを使用したログインを常にユーザーに強制する URL での USERID パラメータの使用を制限する場合です。2 番目のシナリオは、CONFIG=MyApp のような少数のパラメータを除く、URL でのすべてのパラメータの使用を管理者が禁止する場合です。

パラメータ restrictedURLparams では、Forms 管理者は、formsweb.cfg ファイル内の URL へアクセス可能なパラメータをユーザーの使用が制限されたものとみなすことができます。管理者は、名前を付けた構成セクションでこのパラメータを指定して、デフォルトの構成セクションで指定したパラメータを上書きできます。restrictedURLparams パラメータ自体は URL で設定できません。

図 4-1 は、restrictedURLparams パラメータを [myApp] セクションで定義し、[デフォルト] 構成セクションで設定したパラメータを上書きする例を示しています。

図 4-1 myApp のサンプル構成セクション

### Edit Section: myApp

Refreshed at March 4

Select	Name	Value	Description
<input type="radio"/>	form	<input type="text" value="myapps.fmx"/>	
<input checked="" type="radio"/>	restrictedURLparams	<input type="text" value="userid_debug"/>	

Name  Value

デフォルトでは、このユーザー (scott) は、Forms アプリケーションのデバッグ、Forms Trace の使用または Forms Trace 内のレコードの編集を許可されていません。[myApp] セクションでは、ユーザー scott はアプリケーション・アクセス時にログインのみ要求され、アプリケーションのデバッグは許可されていません。しかし、scott は Forms Trace を操作し、このアプリケーションの URL によってレコードを編集できるようになります。

restrictedURLparams パラメータを使用して、管理者は、このアプリケーションでユーザーの使用が制限 (または許可) されているパラメータ・リストが表示されるエラー・ページへユーザーをリダイレクトできます。

## 4.7 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。

serverArgs アプレット・パラメータのコーディングの詳細は、第 3.2.4 項「ランフォーム・パラメータにおける特殊文字の指定」を参照してください。

## 4.8 Oracle Forms アプリケーションへの Graphics の挿入

Oracle Forms アプリケーションと Graphics 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、Graphics を含める必要があります。

```
PATH=<ORACLE_HOME>/bin;<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。

Graphics の詳細は、『Oracle Forms for Windows and UNIX Forms6i からの Forms アプリケーションの移行』および『Deploying Graphics in Oracle9iAS Forms Services』を参照してください。これらのマニュアルは、Oracle Technology Network (OTN) で入手可能です (<http://otn.oracle.com/products/forms/> または <http://otn.oracle.co.jp/products/forms/>)。

## 4.9 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。詳細は、<http://otn.oracle.co.jp/products/forms/> を参照してください。

### 4.9.1 アイコン

Oracle Forms 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。つまり、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 Document Base 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、Oracle Enterprise Manager Application Server Control の「Forms Web 構成」ページで 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、Oracle 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。アプリケーションや Oracle Forms のインストールに依存しない中央のディレクトリにイメージを格納する場合は、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。

#### 4.9.1.1 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 Software Development Kit (Java SDK) 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・ファイルに圧縮されます。

Oracle 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かぎり永久的に保存されます。

---

---

**注意：** Oracle 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.jar ファイルの一部であるため、配置する必要はありません。

---

---

#### 4.9.1.2 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薦めします。

##### 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）

1. 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を値を設定します。

次に例を示します。

```
[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 4-12 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 forms90 ディレクトリか forms90 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。	ImageBase 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 formsweb.cfg のアーカイブ・パラメータに追加します。
Registry.dat	Oracle 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。

## 4.9.2 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 HTML ファイルで次のように定義します。または、Oracle Enterprise Manager の「Forms Web 構成」ページで定義することもできます。

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

```
<PARAM NAME="background" VALUE="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。

## 4.9.3 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 リクエストが 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を使用します。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。

### 4.9.3.1 JAR ファイルの作成

Java SDK 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、<http://www.javasoft.com> を参照してください。

次に例を示します。

```
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。

### 4.9.3.2 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、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になるよう指定する必要があります。

このパラメータには次の2つの異なる値を指定できます。

- **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
```

## 4.9.4 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)
- デフォルトの Forms Web 構成ファイルまたは 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)
- Oracle Enterprise Manager の「Forms Web 構成」ページ (Forms Web 構成ファイル) 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 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のディレクトリを指す URL。



#### 4.9.4.1 DocumentBase

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。この場合、imageBase パラメータを指定する必要はありません。

**表 4-13 アイコンの検索パス**

指定ディレクトリ	Forms Services で使用する検索パス
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)

**表 4-14 イメージの検索パス**

#### file.gif

(たとえば、formsweb.cfg で splashscreen=file.gif として指定)

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)
file.gif (たとえば、formsweb.cfg で splashscreen=file.gif として指定)	http://host/documentbase/file.gif

#### 4.9.4.2 CodeBase

次に示すように、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 4-15 Forms Services で使用するアイコンの検索パス**

指定ディレクトリ	Forms Services で使用する検索パス
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

**表 4-16 Forms Services で使用するイメージの検索パス**

指定ディレクトリ	Forms Services で使用する検索パス
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

## 4.10 言語検出の有効化

Oracle Forms アーキテクチャは、複数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 Oracle 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。

### 4.10.1 言語検出の指定

「Forms Web 構成」ページ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
```

### 4.10.2 言語検出機能の仕組み

Forms Servlet が特定の構成 (`http://myserv/servlet/f90servlet?config=hr` など) のリクエストを受信した場合、リクエスト・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からなければ、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リクエストを受信した場合 ("config=" URL パラメータがない `http://myserv/servlet/f90servlet` 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。

#### 4.10.2.1 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。次の4つのレベルで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]`）で検索されます。
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。各言語で異なる `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 OracleAS Forms Services の使用方法

Oracle Application Server 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。

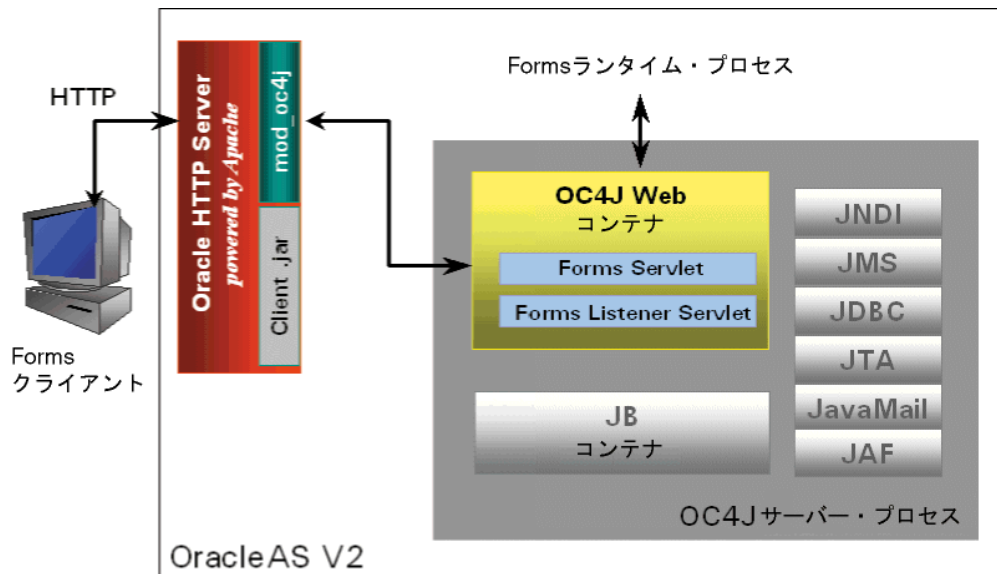
- [OC4J サーバー・プロセス](#)
- [パフォーマンス / スケーラビリティのチューニング](#)
- [OC4J のロード・バランシング](#)
- [Forms Listener Servlet での HTTPS の使用方法](#)

## 5.1 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 リクエストが Oracle HTTP Server 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 Oracle Forms セッションをリクエストした場合、そのリクエストは Oracle HTTP Server Listener で受信されます。HTTP Listener は再びそのリクエストを 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はありますが、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 / スケーラビリティのチューニング」の項を参照）。

OC4J は次のアーキテクチャで動作します。

図 5-1 OC4J アーキテクチャと Forms Services



## 5.2 パフォーマンス / スケーラビリティのチューニング

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。特定の Forms Services の構成に最適なチューニングを行うには、リソース管理とユーザーのニーズを考慮する必要があります。詳細は、『Oracle Application Server パフォーマンス・ガイド』(OTN-J (<http://otn.oracle.co.jp/document/index.html>) で入手可能) を参照してください。

## 5.3 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)。
```

## 5.4 高い値の MaxClients 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 で判断できます。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、同時にリクエストを処理する 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 リクエストが急激に増大し、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。

## 5.5 OC4J のロード・バランシング

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

Oracle HTTP Server Listener で提供するロード・バランシング・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 リクエストを 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、『Oracle Application Server パフォーマンス・ガイド』(OTN-J (<http://otn.oracle.co.jp/document/index.html>) で入手可能) の OC4J の章を参照してください。

Forms 固有の情報の詳細は、『OracleDS Forms Developer and OracleAS Forms Services Release Notes』を参照してください。

次の図では、配置例を 4 つ示します。

- Oracle HTTP Listener と同じホスト上にある複数の OC4J エンジン間で受信リクエストを分散する場合
- Oracle HTTP Listener と異なるホスト上にある複数の OC4J エンジン間で受信リクエストを分散する場合
- 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 エンジン間で受信リクエストを分散する場合
- 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 エンジン間で受信リクエストを分散する場合

図 5-2 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

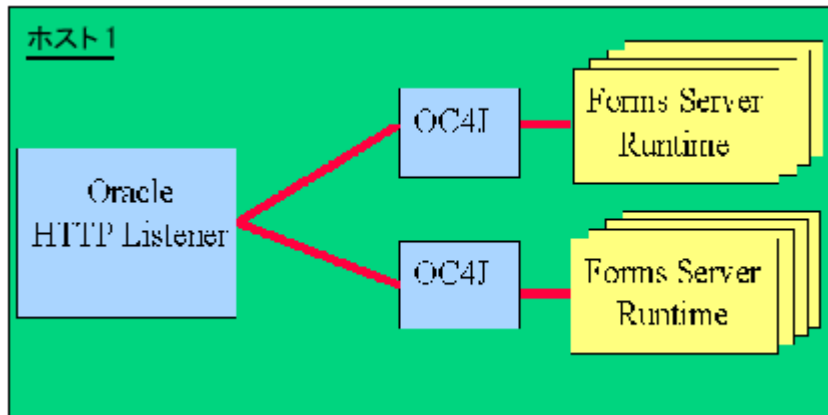




図 5-3 ケース 2: Oracle HTTP Listener と異なるホスト上にある複数の OC4J エンジン

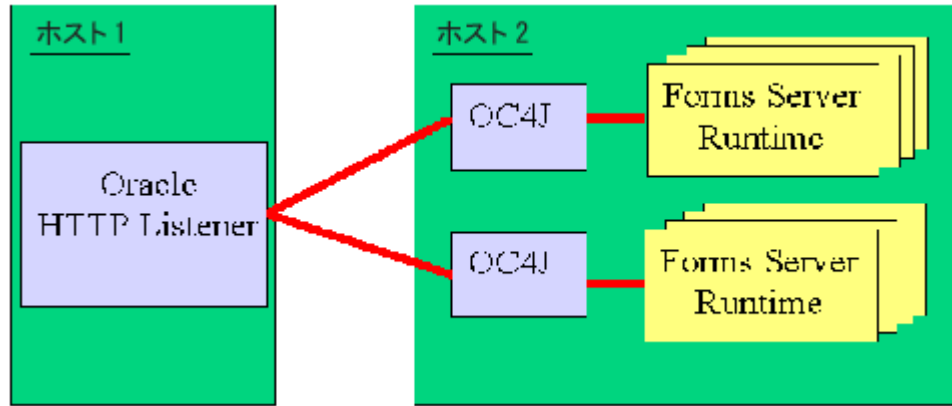


図 5-4 ケース 3: それぞれ異なるホスト上にある複数の OC4J エンジンと複数の Oracle HTTP Listener

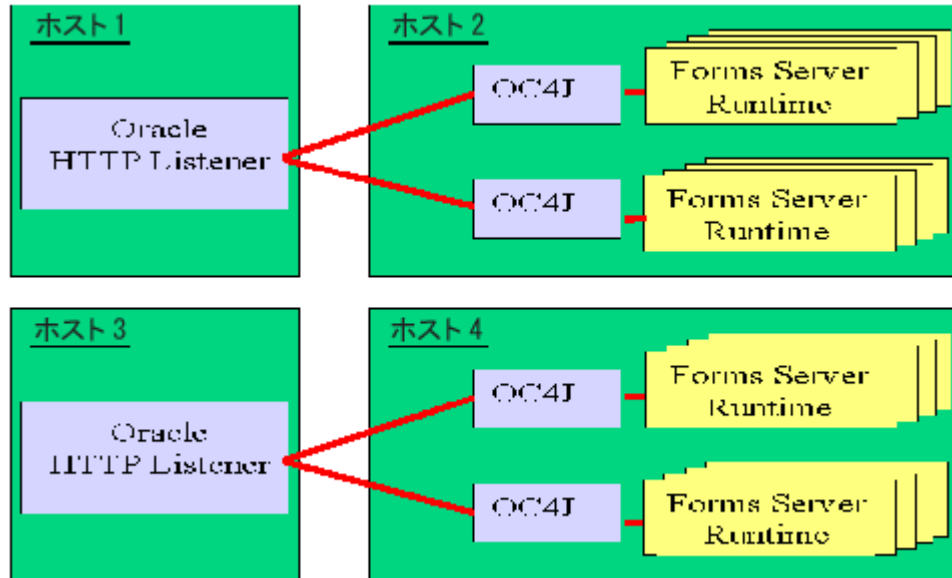
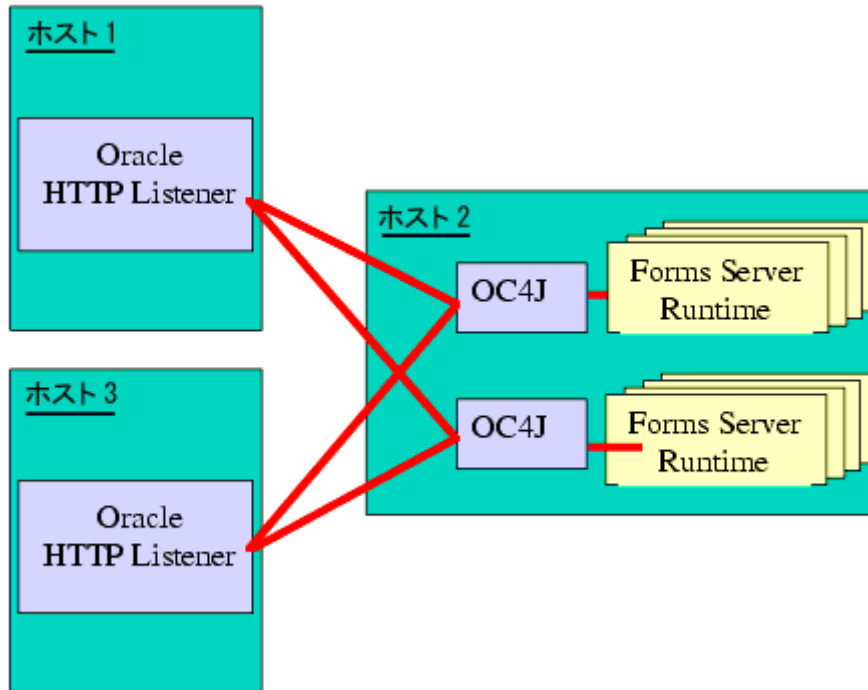


図 5-5 ケース 4: 異なるホスト上にある複数の Oracle HTTP Listener と同じホスト上にある複数の OC4J エンジン



HTTP Listener と OC4J での Forms Services のチューニングと最適化の詳細は、『Oracle Application Server パフォーマンス・ガイド』（OTN-J (<http://otn.oracle.co.jp/document/index.html>) で入手可能) を参照してください。

## 5.6 Forms Listener Servlet での HTTPS の使用方法

Oracle Forms に HTTPS を使用するのには、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。

## 5.7 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。Oracle Application Server Forms Services サ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、Oracle 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。

## 5.8 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 の `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：** Oracle Application Server Web Cache を有効にして実行する場合（通常は有効にします）は、ファイル `<OracleAS_HOME>/webcache/wallets/default/b64certificate.txt` を使用する必要があります。Web Cache で実行しない（Oracle HTTP Server に直接アクセスする）場合は、次の手順でデモ・ルート証明書ファイルを作成する必要があります。

1. Oracle Wallet Manager を起動します。
2. `<ORACLE_HOME>/Apache/Apache/conf/ssl.wlt/default/ewallet.p12` を開きます。
3. 「操作」メニューの「Wallet のエクスポート」を選択します。
4. テキスト・ファイル "demoCertCA.txt" として保存します。

必要な証明書ファイルを用意したら、手順に従って、証明書を使用できるように JInitiator を構成します（証明書を JInitiator の `certdb.txt` ファイルに追加します）。

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

## 5.9 ユーザー ID/パスワード非表示機能の使用法

Oracle Application Server 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。

デフォルトでは、この機能を使用すると Forms Services で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。Oracle 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。

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

Oracle Application Server 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せん。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、ユーザーがログインしているか（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、不要なログイン・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。

Oracle Application Server のインストール・プロセスで設定されている codebase および serverURL 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。

---

# Oracle Application Server Single Sign-On と OID での Forms Services の使用方法

シングル・サインオン (SSO) は、共有認証トークンまたは認証局を使用してユーザーを認証するアプリケーション機能です。たとえば、あるアプリケーションで認証されたユーザーは、同一の認証ドメイン内にある他のすべてのアプリケーションでも自動的に認証されます。

Oracle Application Server Forms Services アプリケーションは、Oracle Single Sign-On Server と Oracle Internet Directory (OID) を使用した 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、複数の Web ベースのアプリケーションにブラウザからアクセスできる Web 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。

Oracle Single Sign-On Server を使用すると、Oracle 製品以外のアプリケーション (カスタム J2EE アプリケーションなど) に対し、SSO を有効にできます。

Oracle Forms アプリケーションは、Oracle Single Sign-On Server と OID に基づいた企業の SSO アーキテクチャに透過的に統合されます。Oracle Application Server のコンポーネントである Oracle Application Server Forms Services では、サーバー・インスタンスで実行されるすべての Forms アプリケーションに対する SSO が、最初からサポート可能な状態になっています。Forms アプリケーションでコーディングを追加する必要はありません。

この章には、次の項が含まれています。

- [SSO、OID および Forms の新機能](#)
- [アプリケーションでの SSO の有効化](#)
- [SSO で実行される Forms のデータベース・パスワード期限切れのサポート](#)
- [認証フロー](#)

## 6.1 SSO、OID および Forms の新機能

このリリースの OracleAS Forms Services では、次の機能および拡張機能が利用可能です。

- [OID](#) にリソースがない場合の動的リソース作成
- [OID](#) でのデフォルト・プリファレンス (Forms リソースの定義) のサポート
- [Forms SSO](#) での動的ディレクティブのサポート
- [SSO](#) で実行される Forms のデータベース・パスワード期限切れのサポート

### 6.1.1 OID にリソースがない場合の動的リソース作成

ユーザーが Forms に接続すると、SSO Server および OID と連動して mod\_osso により認証されます。認証されたユーザーは、Forms Servlet に送られます。Forms Servlet は SSO ユーザー名が含まれたユーザーのリクエスト情報を取得します。このユーザー名とアプリケーション名によって、OID でこのアプリケーションに対するユーザーのリソース情報を識別する一意のペアが作成されます。

認証された Forms ユーザーに、OID で要求されている特定のアプリケーションのリソース情報がない場合、ユーザーはデフォルトで Forms エラー・メッセージ (このアプリケーションで利用可能なリソースがないという内容) を受け取ります。

不足したリソース情報を Forms Services で処理する方法は、アプリケーションまたは Forms Services 管理者によってカスタマイズできます。次のオプションが利用可能です。

- Forms エラー・メッセージの表示 (デフォルト)
- 定義済 URL へのユーザーのリダイレクト
- Forms ログイン画面の表示

システム管理者は、Forms 構成ファイルにリダイレクション URL (絶対 URL または相対 URL) を指定します。

### 6.1.2 OID でのデフォルト・プリファレンス (Forms リソースの定義) のサポート

Forms の以前のリリースでは、Oracle Delegated Administration Services (DAS) を使用して各ユーザー・アカウントに追加されたリソースが使用されました。この実装では、ユーザーが共通リソースを共有している場合でも、10 人、1 万人などのユーザー数に関係なく、各ユーザーにリソースを実装する必要があります。

このリリースの Forms では、Forms 管理者とアプリケーション管理者が OID プリファレンスを使用して、共通に使用されるリソースをデフォルト・リソースとして定義できます。管理者がリソースを作成すると、すべてのユーザー・アカウントでこのリソースが自動的に継承され、Forms 内で使用されます。

### 6.1.3 Forms SSO での動的ディレクティブのサポート

このリリースから、`formsweb.cfg` ファイル内に Forms の SSO が設定されています。また、新しい SSO パラメータである `ssoMode` を使用します。このパラメータにより、カスタム・アプリケーションで SSO 認証を行うかどうか指定されます。

このパラメータを使用すると、Forms Services インスタンスで両方のアプリケーション・タイプ、つまりパブリックの Forms と SSO で保護された Forms を処理できるようになります。`formsweb.cfg` ファイル内で SSO が構成されているため、Oracle Enterprise Manager Application Server Control で SSO パラメータの読み取りと書き込みができます。

詳細は、「`ssoDynamicResourceCreate`」の項を参照してください。

### 6.1.4 SSO で実行される Forms のデータベース・パスワード期限切れのサポート

Oracle Forms の以前のリリースでは、Oracle Forms と Oracle データベース間でパスワード変更が可能でしたが、これらの変更（期限切れも含む）は OID に伝播されませんでした。

このリリースの Oracle Application Server Forms Services では、データベース・パスワードが期限切れになった場合に、SSO モードで実行される OracleAS Forms Services アプリケーションを使用してパスワードを更新すると、ユーザーが入力した新しいパスワードによって、OID でこのアプリケーションのリソース・アクセス記述子 (RAD) が更新されます。この機能により、データベース・パスワードが変更されても、Forms での SSO が継続して機能するようになります。ただし、Forms ではなく SQL\*PLUS でパスワードが変更された場合、データベース接続文字列は OID で更新されません。

## 6.2 Forms で使用される SSO コンポーネント

SSO モードで Forms アプリケーションを実行すると、OracleAS の次のソフトウェア・コンポーネントが使用されます。

- Oracle Application Server Single Sign-On Server - Oracle Application Server の認証サービス。Oracle Internet Directory を使用してユーザー名とパスワードを格納します。
- `mod_osso` - HTTP モジュール `mod_osso` は、Oracle Application Server Single Sign-On Server に対する単一のパートナ・アプリケーションとして認証プロセスを簡素化し、リリース 2 の Oracle Application Server アプリケーションで認証を透過的に行います。Oracle Application Server Forms Services および Oracle Application Server Reports Services は、`mod_osso` を使用して、Oracle Application Server Single Sign-On Server に対するパートナ・アプリケーションとして登録します。
- Oracle Internet Directory - LDAP V3 準拠のディレクトリ・サーバー。ユーザーのログイン情報を格納します。LDAP サーバーは、読み取りアクセス用に最適化された特別なデータベースです。

- Forms Servlet - Oracle Application Server Forms Services コンポーネント。最初のユーザー・リクエストを受け取り Forms アプリケーションを起動します。Forms Servlet は、アプリケーションで SSO が必要かどうかを検出し、リクエストを Oracle Application Server Single Sign-On Server にダイレクトし、Oracle Internet Directory にアクセスしてデータベースの接続情報を取得します。
- formsweb.cfg - Forms 構成ファイル。Forms アプリケーションで SSO を有効にするためのパラメータが含まれます。formsweb.cfg ファイルは、インストールされた Oracle Application Server の forms90/server ディレクトリにあります。

### 6.3 アプリケーションでの SSO の有効化

Oracle Forms アプリケーションは、中央の構成ファイル（forms90/server ディレクトリにある formsweb.cfg ファイル）を使用して構成されます。formsweb.cfg ファイルは、Oracle Enterprise Manager Application Server Control を使用して編集できます。オラクル社ではこの方法をお勧めします。

Oracle Application Server Single Sign-On およびエラー処理は、formsweb.cfg ファイル内の次のパラメータで定義されます。

- ssoMode [TRUE | FALSE]
- ssoDynamicResourceCreate [TRUE | FALSE]
- ssoErrorUrl [ 文字列 URL]
- ssoCancelUrl [ 文字列 URL]

formsweb.cfg ファイル内にあるこれらの Oracle Forms パラメータは、"User Parameter" セクションおよび "Named Configuration" セクションで設定できます。"User Parameter" セクションで設定すると、サーバーで実行されるすべての Forms アプリケーションでこれらの設定がデフォルトの動作になります。"Named Configuration" セクションで設定すると、特定のアプリケーションでのみこれらの設定が有効になります。SSO 定義により、**User Parameter** セクションで設定された同一の定義が上書きされます。

#### 6.3.1 ssoMode

ssoMode パラメータによって、Oracle Application Server Forms Services アプリケーションで SSO が有効になります。デフォルトでは、SSO モードで実行するように Oracle Forms アプリケーションは構成されません。ssoMode パラメータは、formsweb.cfg ファイル内の 2 つの場所で設定できます。値 true を使用して、システム・パラメータとして ssoMode を設定すると、この Forms Services インスタンスによってすべてのアプリケーションが SSO モードで実行されます。Oracle Forms アプリケーションの名前を付けた構成で ssoMode パラメータを設定すると、この特定のアプリケーションでのみ SSO が有効または無効になります。

```
[myApp]
form=myFmx
ssoMode=true
```



**アプリケーションで SSO を有効にするには：**

1. Oracle Enterprise Manager Application Server Control を起動します。
2. 「Forms」を選択します。
3. 「構成」タブを選択します。
4. 目的のアプリケーションの構成セクションの横にあるラジオ・ボタンを選択し、「編集」をクリックします。
5. 「名前」フィールドで、`ssoMode` と入力します。
6. 「値」フィールドで、`true` と入力します。
7. 「新規パラメータの追加」をクリックします。  
選択されたアプリケーションで SSO が有効になります。

**アプリケーションで SSO を無効にするには：**

1. Oracle Enterprise Manager Application Server Control を起動します。
2. 「Forms」を選択します。
3. 「構成」タブを選択します。
4. 目的のアプリケーションの構成セクションの横にあるラジオ・ボタンを選択し、「編集」をクリックします。
5. `ssoMode` パラメータの横にあるラジオ・ボタンを選択します。
6. 「値」列で、`false` と入力します。
7. 「適用」をクリックします。  
選択されたアプリケーションで SSO が無効になります。

`ssoMode` を `false` に設定した場合、オプションとして、`ssoErrorUrl` パラメータに URL を指定し、ユーザーを情報ページにリダイレクトすることもできます。

## 6.3.2 ssoDynamicResourceCreate

ssoDynamicResourceCreate パラメータはデフォルトで true に設定されます。これにより、ユーザーは OID で RAD エントリを作成し、アプリケーションを実行できます（このリソース・エントリが存在しない場合）。表示される Web ページは、DAS で用意された標準フォームです。この Web ページは Oracle Forms 固有のものであるためカスタマイズできません。

動的リソース作成によって、管理者がユーザーの RAD 情報を事前に作成する必要がなくなるため、OID 管理が簡素化されます。ssoDynamicResourceCreate パラメータは、システム・パラメータとして formsweb.cfg ファイルに設定するか、名前を付けた構成のパラメータとして設定できます。デフォルトで true に設定されているため、このパラメータを特定のアプリケーションの名前を付けた構成で使用し、デフォルトとは異なる方法で不足した RAD エントリを処理できます。

ssoDynamicResourceCreate パラメータを false に設定し、SSO を有効にしてアプリケーションを構成し、ssoErrorURL に値を指定しないと、認証されたユーザーとこのアプリケーションに RAD リソースがない場合に Oracle Forms でエラー・メッセージが表示されるので注意してください。

管理者にとって、ユーザーが自分自身のリソースを作成することは望ましくないこともあるため（OID で問題を起こす可能性もあるため）、管理者はこれらのパラメータを使用して、OID のリソース作成を制御できます。デフォルトの動作では、リソースを作成できる HTML フォームにユーザーをダイレクトしますが、管理者はその設定を変更し、ユーザーをカスタム URL にリダイレクトできます。

Forms アプリケーションの構成セクションでは、次のパラメータを設定する必要があります。

```
[myApp]
form=myFmx
ssoMode=true
ssoDynamicResourceCreate=false
```

Oracle Enterprise Manager Application Server Control を使用してこれらのパラメータを設定する方法は、[第 4.3.3 項「パラメータの管理」](#)を参照してください。

### 6.3.3 ssoErrorURL

管理者は `ssoErrorURL` パラメータを使用して、特定のアプリケーションにユーザーの RAD エントリがない場合の処理を行うリダイレクション URL を指定できます。このパラメータは、`ssoDynamicResourceCreate` パラメータが `false` に設定され、動的リソース作成の動作が無効になっている場合にのみ有効になります。`ssoErrorURL` パラメータは、システム・パラメータおよび名前を付けた構成セクションのパラメータとして定義できます。URL には、任意のアプリケーション、静的 HTML ファイル、RAD を作成するカスタム・サーブレット (JSP) アプリケーション (次の例を参照)などを指定できます。

```
[myApp]
form=myFmx
ssoMode=true
ssoDynamicResourceCreate=false
ssoErrorURL=http://myServ.com:7779/servlet/handleCustomRADcreation.jsp
...
```

### 6.3.4 ssoCancelUrl

`ssoCancelURL` パラメータは動的 RAD 作成機能 (`ssoDynamicResourceCreate=true`) とともに使用します。このパラメータでは、ユーザーが HTML フォーム (要求されたアプリケーションの RAD エントリを動的に作成するためのフォーム) で取消しボタンを押した場合にリダイレクトされる URL が定義されます。

### 6.3.5 Forms から SSO 情報へのアクセス

Forms アプリケーションで SSO の認証情報を使用して作業する場合、オプションとして、`GET_APPLICATION_PROPERTY()` ビルトインを使用して、SSO のログイン情報を取得できます。SSO のログイン情報には、Oracle Application Server Single Sign-On のユーザー ID、ユーザーの識別名 (`dn`)、およびサブスクライバの識別名 (サブスクライバ `dn`) があります。

```
authenticated_username := get_application_property('sso_userid') ;
userDistinguishedName := get_application_property('sso_usrdn') ;
subscriberName := get_application_property('sso_subdn') ;
```

## 6.4 Oracle Forms と Reports の統合について

Oracle Application Server Reports Services をインストールすると、SSO が有効になります。

Oracle Forms アプリケーションで統合 Oracle Reports をコールする最適な方法は、Oracle Forms ビルトインの RUN\_REPORT\_OBJECT() を使用することです。

SSO で保護された Oracle Forms アプリケーションからレポートを要求する場合、RUN\_REPORT\_OBJECT() ビルトインへの各コールを使用して、認証されたユーザーの SSO ID が暗黙的に Reports Server へ渡されます。必要に応じてさらに認可チェックを行う場合は、SSO ID を使用して Reports Server へのユーザーを認証します。

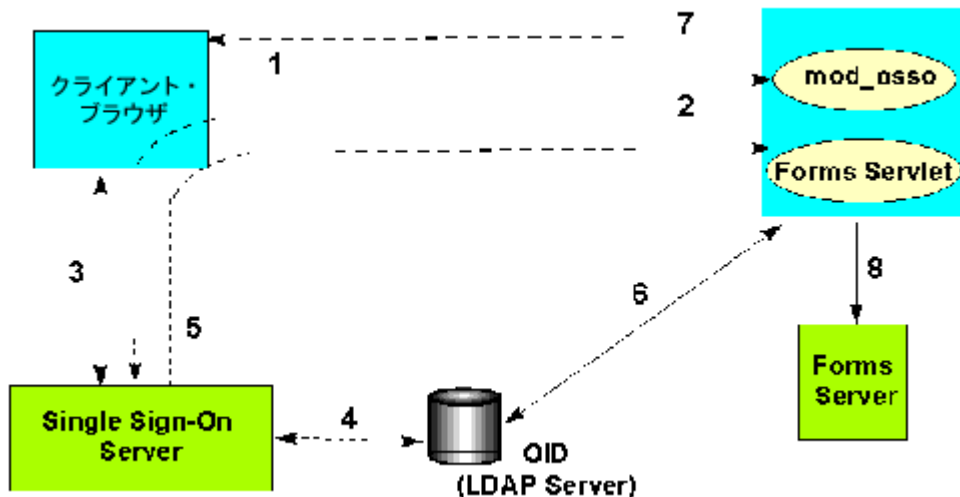
非 SSO モードで実行される Forms アプリケーションでは、SSO で保護された Reports Server のレポートを実行できますが、Reports Server で認可が必要な場合は Forms アプリケーションが失敗します。また、Web に Reports 出力を取り込む場合、ユーザーは SSO 接続情報を指定する必要があります。

Oracle Forms と Oracle Reports の統合の詳細は、Oracle Technology Network Japan (<http://otn.oracle.co.jp/products/forms/>) にあるホワイト・ペーパー『Oracle 9iAS Forms Services と Oracle 9iAS Reports の統合』を参照してください。

## 6.5 認証フロー

ユーザーが最初に Oracle Application Server Single Sign-On で保護されたアプリケーション URL をリクエストした場合、Oracle Forms の SSO の認証フローは次のようになります。

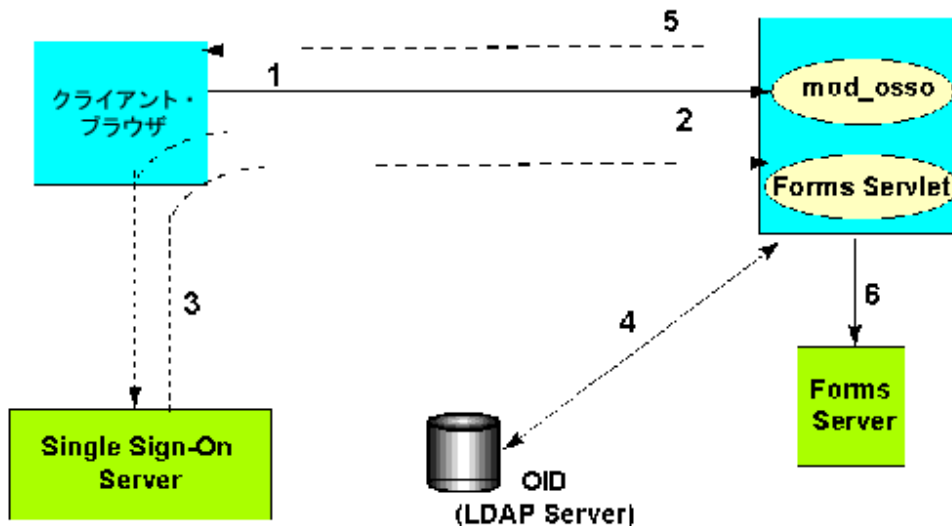
図 6-1 最初のクライアント・リクエストの認証フロー



1. ユーザーは、  
http(s)://<hostname>:<port>/forms90/f90servlet?config=<application>&... のような Forms URL をリクエストします。
2. Forms Servlet が、そのユーザー・リクエストを 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. ユーザーのリクエストが、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・セッションにユーザー ID パラメータを設定し、アプリケーションが Forms Listener Servlet に接続されます。
8. Forms Servlet が、Forms Server を起動します。

別のパートナー・アプリケーションで認証されたユーザーが Oracle Application Server Single Sign-On で保護されたアプリケーションを要求した場合、OracleAS Forms Services の Oracle Application Server Single Sign-On の認証フローは、次のようになります。

図 6-2 後続のクライアント・リクエストの認証フロー



1. ユーザーが、Forms URL をリクエストします。
2. Forms Servlet が、そのユーザー・リクエストを Oracle Application Server Single Sign-On Server にリダイレクトします。
3. ユーザーのリクエストが、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・セッションにユーザー ID パラメータを設定し、アプリケーションが Forms Listener Servlet に接続されます。
6. Forms Servlet が、Forms Server を起動します。

---

## トレースと診断

Oracle 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。Oracle Forms に付属のトレース・ツールおよび診断ツールを使用すると、Oracle 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できます。Oracle Enterprise Manager Application Server Control では、トレース出力を使用して Oracle 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、Oracle Forms のトレース情報を収集できます。

- **Forms Trace:** Oracle Forms の以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、Oracle Forms のセッションをすべて記録し、Oracle Forms 関連のネットワークの通信量を監視して、サイトの構成問題をデバッグできます。

## 7.1 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](#)

### 7.1.1 Forms Trace の構成

イベントとは、ユーザー・アクションの直接または間接的の結果として Oracle Forms 内に起こる事象を意味します。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グループを指定します。

「Oracle Enterprise Manager Application Server Control Forms」ページの「構成」タブで「Forms トレース構成」を選択し、トレースを行うイベントを定義します。このページで、`fttrace.cfg` ファイルの変更がすべて管理されます。

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](#)」の項を参照してください。

---

---

**注意：** ほとんどの Web アプリケーションでは、ページを移動するだけで、保存されていない変更は簡単に失われてしまいます。Oracle Enterprise Manager Application Server Control を使用して Forms 構成ファイル、トレース・ファイル、環境ファイルに変更を加えた場合は、他のページに移動する前に必ず保存するようにしてください。

変更の保存に必要な時間は、変更を加えた行数によって異なります。たとえば、50 行のコメントを追加した変更では、単一のエントリを削除した変更よりも保存に時間がかかります。

---

---



---

---

**注意：** 構成ファイルや環境ファイルを手動で編集する場合は、Oracle Enterprise Manager がすべての変更を読み取れるように、Oracle Enterprise Manager およびすべての Distributed Configuration Management (DCM) プロセスを再起動する必要があります。Oracle Enterprise Manager と DCM プロセスを再起動しない場合は、その後 Oracle Enterprise Manager で加えられた変更によって、これらのファイルに手動で加えられた変更が上書きされます。これらの DCM プロセスは次のとおりです。

- emctl stop agent
  - emctl stop em
  - dcmctl stop
  - opmnctl stopall
  - opmnctl startall
  - dcmctl start
  - emctl start agent
  - emctl start em
- 
- 

---

---

**注意：** formsweb.cfg ファイルと default.env ファイルのバックアップを作成してから、Oracle Enterprise Manager Application Server Control でこれらのファイルを編集する必要があります。

---

---

---

---

**注意：** 最初にトレースを OFF にし、新しい設定でトレースを再度 ON にすると、トレースは新しいトレース・グループで有効になります。

---

---

---

---

**注意：** Windows 上で Forms プロセスをトレースするには、ProcessManager Service の「デスクトップとの対話をサービスに許可」チェック・ボックスが選択されている必要があります。このように設定されていない場合、トレースを有効にしようとすると次のエラーが発生します。

```
oracle.sysman.emSDK.emd.comm.RemoteOperationException.  
ユーザー名とパスワードを確認してください
```

---

---

**Forms Trace を構成するには：**

1. Oracle Enterprise Manager Application Server Control を起動します。
2. Oracle Enterprise Manager Application Server Control のメイン・ページで、構成する Forms Services インスタンスのリンクを選択します。
3. Forms Services インスタンスの「概要」ページで、「構成」リンクを選択します。

**ftrace.cfg ファイルに新しいパラメータを作成するには：**

- この新しいパラメータの名前と値を入力し、ページ下部の「新規パラメータの追加」をクリックします。

**ftrace.cfg ファイルのパラメータを削除するには：**

- 削除するパラメータの横にあるラジオ・ボタンをクリックし、「削除」をクリックします。次のページで削除内容を確認します。

**ftrace.cfg ファイルの既存のパラメータを編集するには：**

- 既存のパラメータの横にあるラジオ・ボタンを選択し、テキスト領域で値を変更します。「適用」をクリックして変更を保存します。

**変更を保存するには：**

- パラメータの横にあるラジオ・ボタンをクリックし、「適用」をクリックします。

次の例では、3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。

図 7-1 Oracle Enterprise Manager でのトレース・イベントの構成

表示

選択	名前	値	説明
<input checked="" type="radio"/>	errors	1-3	
<input type="radio"/>	allevents	1-199	All Events
<input type="radio"/>	windows	41-44	
<input type="radio"/>	sql	96	

名前  値

概要 [ユーザー・セッション](#) **構成** [環境](#) [Formsユーティリティ](#)

ftrace.cfg を手動で編集する場合は、次の事項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。
- イベント・グループには、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。
- 番号の範囲を使用できます。

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 `custom1` と名付けられた `tracegroup` の値 ("32-46, 65, 66, 96, 194" など) の指定に相当します。

### 7.1.1.1 URL パラメータ・オプションの指定

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。

```
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。
Tracegroup	名前、イベント番号、イベントの範囲	<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"> <li>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。</li> <li>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。</li> <li>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=&lt;keyword&gt;。この &lt;keyword&gt; は、ftrace.cfg で指定します (たとえば、Tracegroup=MyEvents)。</li> </ul> <p>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。</p> <ul style="list-style-type: none"> <li>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&lt;= イベント &lt;=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。</li> <li>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</li> <li>Tracegroup キーワードを使用してイベント・セットを組み合せることができます。たとえば、次のように指定します。 Tracegroup = 0-3,34,67,SQLInfo</li> </ul>

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	ディレクトリ	<p>トレース情報の保存場所を指定します。Oracle Enterprise Manager でトレース・ファイルを検出して正しく処理できるように、トレース・ファイルは &lt;ORACLE_HOME&gt;/forms90/trace に保存する必要があります。</p> <p>ディレクトリが指定されていない場合、ファイルは現在の作業ディレクトリに書き込まれます。</p> <p>ログ・ファイルが指定されていない場合、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ず (例: forms_&lt;pid&gt;.trc)。</p>

## 7.1.2 トレースの開始

URL にトレース・エントリを指定するか、Oracle Enterprise Manager Application Server Control を使用して、トレースを開始します。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。

---

**注意：** 表示されたダイアログ・ボックスで接続情報を指定する必要があります (必要なユーザー名とパスワードは、Forms Services のインストール時に使用されたオペレーティング・システム・アカウント用のものです)。

---

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=cx1&record=forms&tracegroup=0-199
http://cx-pc/forms90/f90servlet?form=cx1&record=forms&tracegroup=mysql
http://cx-pc/forms90/f90servlet?form=cx1&record=forms&tracegroup=0-199;log=run1.log
```

Oracle 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 Oracle Forms に関する最新情報は、OTN (<http://otn.oracle.com> / 英語サイト) または OTN-J (<http://otn.oracle.co.jp/> / 日本語サイト) を参照してください。

## 7.1.3 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。Oracle Enterprise Manager Application Server Control を使用しない場合は、Translate ユーティリティを使用する必要があります。

---

---

**注意：** Oracle Enterprise Manager Application Server Control の「ユーザー・セッション」画面でトレース・ログを表示する前に、Forms の Web 構成ファイルのデフォルト・セクションで、allow\_debug パラメータを true に設定する必要があります。

---

---

**トレース・データを表示するには Oracle Enterprise Manager を使用します。**

1. Oracle Enterprise Manager Application Server Control で、「ユーザー・セッション」リンクを選択します。
2. 「トレース・ログの表示」をクリックしてトレース・ログの内容を表示します。

### 7.1.3.1 Translate ユーティリティの実行

Translate ユーティリティによって、トレース・データが XML 形式に変換されます。

**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、次のコマンドを入力します。

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

これにより、myfile.xml が作成されます。

## 7.1.4 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 Oracle 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 7-2 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	該当せず
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	該当せず
48	予約済	該当せず
49-63	予約済	該当せず

表 7-2 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
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	アラート	期間
71	ファイルの取得	ポイント
72-95	予約済	該当せず
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	該当せず
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 ログイン (開始と終了)	期間



表 7-2 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	該当せず
160-191	予約済	該当せず
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	該当せず
1024 以上	予約済	該当せず

\* 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。

## 7.1.5 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。

- [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](#)

### 7.1.5.1 ユーザー・アクション・イベント

**表 7-3 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
アラート	アラート名、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

### 7.1.5.2 Forms Services のイベント

**表 7-4 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
キャンバス	フォーム ID、キャンバス名	69

### 7.1.5.3 イベントの詳細

表 7-5 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

### 7.1.5.4 3 層イベント

表 7-6 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	該当せず	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 ログイン	フォーム ID	136
DB ログオフ	フォーム ID	137

### 7.1.5.5 その他

表 7-7 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

### 7.1.6 Forms Services のトレース・メトリックの監視

この Oracle Enterprise Manager のページを使用して、Forms Services のトレース・メトリックを確認します。

1. Oracle Enterprise Manager Application Server Control を起動します。
2. Oracle Enterprise Manager Application Server Control のメイン・ページで、「ユーザー・セッション」リンクへのリンクを選択します。
3. 「トレース・ログの表示」列でアイコンをクリックします。

## 7.2 Servlet Logging Tools

Oracle Application Server Forms Services で使用できる Servlet Logging Tools では、次の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 Oracle 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）
- Oracle 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・パフォーマンスとリクエスト・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。

- [ロギングの有効化](#)
- [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](#)

### 7.2.1 ロギングの有効化

ロギングは、次のようにして有効にします。

- [表 7-8 「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 7-8 「サポートされているロギング機能」](#) の文字列の 1 つを、Oracle Enterprise Manager Application Server Control の「構成」ページで 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 7-8 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。

**表 7-8 サポートされているロギング機能（続き）**

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
/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リクエストすべてに、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、主にデバックやサポート用に使用します。

### 7.2.1.1 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 Oracle Forms アプリケーションを起動します。

```
http://yourserver/forms90/f90servlet?serverURL=/forms90/190servlet/perf
```

### 7.2.1.2 Oracle Enterprise Manager 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、「Forms Web 構成」ページのデフォルト・セクションで serverURL エントリを次のように変更します。

```
serverURL=/forms90/f90servlet/session
```

### 7.2.1.3 Forms Servlet を起動する 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 Oracle 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=/forms90/190servlet/debug
```

## 7.2.2 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。

Oracle Application Server Forms Services 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_1/application.log
```

Forms Developer では、フルパス名は次のとおりです。

```
<ORACLE_HOME>/j2ee/DevSuite/application-deployments/forms/application.log
```

## 7.2.3 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

### 7.2.3.1 (なし)

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

### 7.2.3.2 /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
```

### 7.2.3.3 /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
```

### 7.2.3.4 /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
```

### 7.2.3.5 /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.
*****
```



---

## パフォーマンス・チューニングに関する 考慮事項

この章では、Oracle Forms アプリケーションを Oracle Application Server Forms Services に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。また、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も説明します。この章には、次の項が含まれています。

- [Forms Services の組み込み最適化機能](#)
- [OracleAS Forms Services アプリケーションのチューニング](#)

Oracle Application Server Forms Services と Oracle Database サーバー間の接続のチューニングについては、この章では扱いません。

## 8.1 Forms Services の組み最適化機能

Oracle Application Server 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。

- [Forms Services Web ランタイム・プーリング](#)
- [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](#)

### 8.1.1 Forms Services の監視

Oracle Enterprise Manager Application Server Control を使用して Oracle Application Server Forms Services を監視し、次のメトリック情報を確認します。

- Forms Services インスタンス
- イベント
- ユーザー・セッション
- Forms Trace

#### 8.1.1.1 Forms Services インスタンスの監視

「概要」 ページを使用して、Forms Services インスタンスのメトリックを監視します。

1. Oracle Enterprise Manager Application Server Control を起動します。
2. Oracle Enterprise Manager Application Server Control のメイン・ページで、監視する Forms Services インスタンスのリンクを選択します。

Forms Services インスタンスのメトリックを監視する「概要」 ページには、次の情報が表示されます。現在の Forms Services インスタンスのステータス（稼動中または停止中）

- 監視対象の Forms Services インスタンスの URL
- 監視対象の Forms Services インスタンスの Oracle ホーム
- この Forms Services インスタンスの全 Forms ランタイム・プロセスによる CPU 使用率 (%)
- この Forms Services インスタンスの全 Forms ランタイム・プロセスによるメモリー使用率 (%)

- ログインしているユーザー数
- Forms Listener Servlet に接続する場合の Forms Services インスタンスの応答時間 (ミリ秒)

次の詳細ページに移動することもできます。

- セッションの詳細
- Forms Services 構成
- 環境
- Forms トレース構成
- Forms ユーティリティ

### 8.1.1.2 Forms イベントの監視

Oracle Enterprise Manager Application Server Control を使用して、すべてのイベントまたは特定のイベントに対するトレースを有効にします。

### 8.1.1.3 ユーザー・セッションのメトリックの監視

1. Oracle Enterprise Manager Application Server Control を起動します。
2. Oracle Enterprise Manager Application Server Control のメイン・ページで、監視する Forms Services インスタンスのリンクを選択します。
3. Forms Services インスタンスの「概要」ページで、「ユーザー・セッション」リンクを選択します。

このページでは、Forms Services インスタンスの各ユーザー・セッションについて次の情報が表示されます。PID: ユーザー・セッションの Forms ランタイム・プロセスのプロセス ID。

- CPU 使用率: ランタイム・プロセスによる CPU 使用率 (%)。
- メモリー使用率: ランタイム・プロセスによるメモリー使用率 (%)。
- クライアント IP アドレス: Forms Services への接続に使用するクライアント・マシンの IP アドレス。
- データベース・ユーザー名: ユーザー・セッションで Forms アプリケーションによって使用されるデータベースのユーザー名。
- 接続時刻: ユーザーが Forms Services に接続した時刻。
- トレース・ステータス: トレースが ON であるか OFF であるかを示します。
- トレース・ログの表示: ユーザーはトレース・ログを表示できます。

- 構成セッション: 特定の Forms セッションで使用される構成セッションの「編集セッション」ページを開きます。

#### 8.1.1.4 メトリック情報のソート

列ヘッダーのリンクをクリックすると、プロセス ID、CPU 使用率、メモリー使用率、IP、ユーザー名、接続時刻を昇順でソートできます。

#### 8.1.1.5 検索

検索を使用して、特定のメトリック情報を見つけます。

セッションの詳細を検索するには:

- プルダウンで「ユーザー名」、「IP アドレス」または「PID」を選択し、次のフィールドに正確な検索文字列（大 / 小文字の区別あり）を入力して、「実行」をクリックします。

検索後に再度セッションの完全な一覧を表示するには:

- 「実行」をクリックします。

## 8.1.2 Forms Services Web ランタイム・プーリング

Forms のランタイム・プーリングを使用すると、構成可能な数のアプリケーション・ランタイム・エンジン事前に起動できます。ランタイム・プーリングによってサーバーのピーク時に迅速に接続できるため、サーバー側アプリケーションの起動時間が短縮されます。ランタイム・プーリングは、サーバー構成のウィンドウが小さく、多くのユーザーが Forms アプリケーションに接続するような状況で役立ちます。事前に起動されたランタイム・エンジンはすべて、同じ環境で実行され、同じアプリケーションで使用されます。

### 8.1.2.1 事前起動パラメータの構成

Oracle Enterprise Manager Application Server Control を使用して、次のパラメータで Forms Services のランタイム・プーリングを構成します。

**表 8-1 Forms のランタイム・プーリング・パラメータ**

パラメータ名	データ型	説明	デフォルト値
prestartRuntimes	ブール	TRUE の場合にのみランタイム・プーリング（事前起動）が有効になります。	FALSE
prestartInit	整数	最初に起動するランタイム実行可能ファイルの数。	1

表 8-1 Forms のランタイム・プーリング・パラメータ (続き)

パラメータ名	データ型	説明	デフォルト値
prestartTimeout	整数	事前に起動された実行可能ファイルがタイムアウトするまでの時間 (分)。	0 (ゼロに設定されるとタイマーは起動しません)
prestartMin	整数	プールに存在するランタイム実行可能ファイルの最小数。	0
prestartIncrement	整数	minRuntimes 未満の場合に作成されるランタイム実行可能ファイルの数。	0

各構成セクションで、これらのパラメータの値を指定できます。prestartRuntimes = true エントリがあっても、関連する事前起動パラメータがない場合はデフォルト値が使用されます。

ロード・バランシングが実現されたシステムに複数の OC4J インスタンスがある場合は、前述のパラメータに指定された様々な値が、アプリケーションの全体ではなく、各 JVM に対応します。

### 8.1.2.2 ランタイム・プーリングの起動

管理者は Oracle Enterprise Manager Application Server Control を使用して、特定のアプリケーションに対して指定された数の実行可能ファイルを事前に起動できます。管理者が必要なアプリケーションを選択すると、Forms Services に通知されます。Web サーバー (OC4J) の起動時に Forms Servlet がロードされます。

Forms Servlet の初期化時に formsweb.cfg ファイルが読み取られ、prestartRuntimes パラメータが有効になったアプリケーションがサーバーで事前に起動されます。

### 8.1.3 Forms Services のユーティリティ

「Forms ユーティリティ」ページの簡単なユーザー・インタフェースを使用して、中間層で一連の操作をコールします。これらの機能は今後拡張される予定です。

現在は、ps（プロセス情報の取得に使用）といくつかの引数のみ使用可能です。

#### 8.1.3.1 Forms Services のユーティリティを使用するには：

- 「パラメータ」テキスト・フィールドで、次の文字列を入力します。

ps  
「送信」をクリックします。

下部のステータス・ウィンドウにプロセス一覧が表示されます。

### 8.1.4 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。すべての Oracle Forms 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・フィールドの変更などの、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、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。

### 8.1.5 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。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。

## 8.1.6 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あり、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。

Oracle Application Server 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。

Oracle Application Server 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)、Oracle Application Server 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。
- データ型はその値に必要な最小のバイト数で送信されます。

## 8.1.7 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

待機時間は、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。待機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。

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、待機時間の影響が大きくなります。トリガーに関連する待機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)、Pre トリガーと Post トリガーの範囲が起動対象となり、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、それらを単一のバケットとして Oracle Application Server Forms Services に配布して処理します。ナビゲートによって複数のオブジェクトを横切る場合（離れているオブジェクトに対してマウスのクリックを行った場合など）、イベント・バンドルは通過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 Oracle Application Server Forms Services に配布します。

## 8.1.8 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 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で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、ネットワークの待機時間の影響も含まれるため、クライアントおよび Oracle Application Server Forms Services 間の通信で使用するバケット数を最小化することが必要です。

## 8.2 OracleAS 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 の組み込み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 Oracle Application Server 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](#)

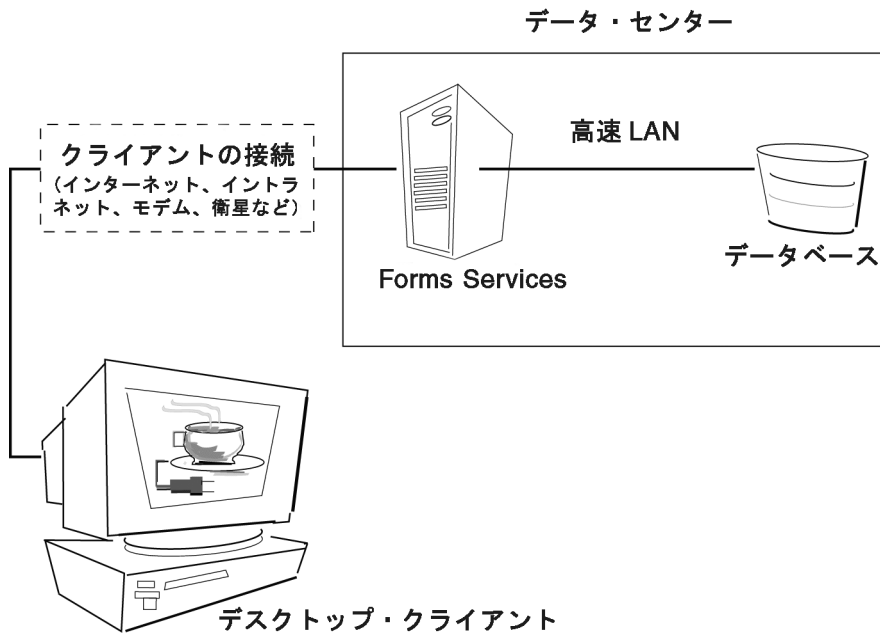
## 8.2.1 データ・サーバーに対する Oracle Application Server Forms Services の位置

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。Oracle Forms のすべてのロジックは、中間層の Oracle Application Server 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問合せ、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・サーバー間的高速接続が重要になります。

このサーバー間の対話はすべて、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 Oracle Application Server Forms Services 間で通信が行われます。これにより、Oracle 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、Oracle Application Server 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 OracleAS Forms Services とデータベース・サーバーの並存



## 8.2.2 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。また起動時間は、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、可能なかぎりロード時間を最小化することが重要です。

Oracle 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は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 の起動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 Oracle 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 にロードするため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](#)

### 8.2.2.1 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) することが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。

Oracle Application Server Forms Services では、通常の配布例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。

#### 8.2.2.1.1 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済み、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、前述の例に示した `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。

- f90oracle\_laf.jar - Oracle のルック・アンド・フィールのクラス
- f90generic\_laf.jar - 汎用 (標準) ルック・アンド・フィールのクラス
- f90resources.jar - 英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` は英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、英語のリソース・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となります。

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

### 8.2.2.1.2 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
```

### 8.2.2.1.3 その他のすべての場合 (Sun 社の 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
```

## 8.2.2.2 キャッシュの使用

Oracle Application Server 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 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。このため、ロード・バランシング・アーキテクチャを使用している場合、異なるサーバーからの同一の JAR ファイルがキャッシュされる可能性があります。JAR ファイルを中央に置き、ロード・バランシング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。

## 8.2.3 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な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。
  - 最初の表示時は、オブジェクト・ナビゲータの表示順序
  - 実行中は、項目プロパティを変更するプログラムの順序

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果が高まります。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。
  - 視覚属性の違いが少なくなるようにロック・アンド・フィールドを整えます。
- **ボイラープレート・テキストの使用を削減。** 開発者として、可能なかぎり、ボイラープレート・テキストではなくプロンプト項目プロパティを使用してください。Forms Developer 6.0 以降には、プロンプトの関連付け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。** イベント・バンドルは、2つまたはそれ以上のオブジェクト間にナビゲーションがおよぶ場合でも、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る場合は、フィールド間をナビゲートする必要がないようにフォームを設計します。フォームは、ユーザーが入力を終了したらスムーズに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つのイベント・バンドル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。
  - アプリケーションのログイン画面を提供して、最初に表示されるオブジェクトを (タイトル、小さなロゴ、ユーザー名およびパスワードなどに) 制限します。
  - Form の初回の表示時は、隠された要素はすぐには必要とされません。次のキャンパス・プロパティを使用します。

エントリでレイズ = はい (キャンパスのみ)

可視 = いいえ



タブ・キャンバスでは、1つのシートのみ表示されますが、いくつかのシートで構成されていることに注意してください。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、タブ・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: タブ・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対し、プロパティ「エントリでレイズ」を「はい」に、「可視」を「いいえ」に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、メニューに対する変更内容が、変更されたメニューの完全な再送信が必要となるような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では、メニューに対し複数の変更を同時に行うか、まったく行わないかのどちらかです。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、メニューに最小限の変更しか加えないアプリケーションもあります。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リングは、LABEL、ICON、VISIBLE および CHECKED のメニュー・プロパティにのみ適用されます。ENABLE/DISABLE イベントは常に送信されますが、メニュー全体の再送信は行われません。

## 8.2.4 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、アプリケーションの実行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。** タイマーを起動すると、非同期イベントが生成されます。キュー内に他のイベントがない場合もあります。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。これらのコンポーネントを、Forms Services やネットワークの介入がなくても同じ効果をもたらす、自己完結型の JavaBeans と置換します。

- **クライアント上での入力項目の妥当性チェックについて考慮。** 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。プラグ可能な 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) 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かぎり、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かぎり減らします。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。

## 8.3 Web Cache と Forms の統合

Oracle Web Cache は Oracle Forms アプリケーションでのロード・バランサとして使用できません。以降の設定手順では、次の状況を想定しています。

1. Oracle Application Server Web Cache インスタンスはホスト A で実行されている。
2. Oracle HTTP Server (OHS) インスタンスと OC4J インスタンスは、Forms アプリケーション D を実行しているホスト B 上にある。
3. OHS インスタンスと OC4J インスタンスは、Oracle Forms アプリケーション D を実行しているホスト C 上にある。

OHS インスタンスと OC4J インスタンスは他にも存在する場合がありますが、ここでは説明を簡略化するために 2 つのインスタンス・ペアのみを取り上げます。Oracle Forms アプリケーションでは OracleAS クラスタリングを利用できないため、OHS インスタンスと OC4J インスタンスはクラスタ化されません。

また、Web Cache 9.0.2.x クラスタも使用できません。Oracle Application Server Web Cache クラスタを使用すると、Oracle Application Server で起動される Oracle Forms のロード・バランシングを実現できません。

Forms アプリケーションはステートフルであるため、セッション・バインド機能を使用して Web Cache を構成し、ステートフルなロード・バランシングを実現する必要があります。

Forms アプリケーションの適切なサイト情報を使用してホスト A 上の Web Cache を構成し、さらにホスト B と C で実行される OHS インスタンスについて、オリジナル・サーバー情報とサイトとサーバー間のマッピング情報を構成します。ホスト B と C についてのオリジナル・サーバー情報を構成する場合、Forms アプリケーション D が実行されているかどうかを検出する ping URL を構成してください（例：/forms90/f90servlet?ifcmd=status）。

### Web Cache でセッション・バインドを構成するには：

1. Web Cache Manager にログインします。
2. ナビゲータ・ペインで「Origin Servers, Sites, and Load Balancing」→「Session Binding」を選択します。
3. 「Session Binding」画面で、「Default Session Binding」を選択して、「Edit Selected」を選択します。
4. 「Edit Session Binding」ダイアログ・ボックスが表示されます。
5. 「Please select a session:」プルダウン・リストで、「Monitoring」→「Health Monitor」を選択します。
6. Oracle Forms アプリケーション D に適切な「Inactivity Timeout」を構成します。
7. 「Submit」をクリックします。
8. 変更を適用し、Oracle Application Server Web Cache を再起動します。

**設定をテストするには：**

1. ブラウザを使用して、Web Cache ホストを指定し、Oracle Forms アプリケーション D にアクセスします。アプリケーションが適切に動作することを確認します。ブラウザ・ウィンドウは開いたままにします。
2. リクエストを処理した OHS と OC4J を識別します。たとえば、これがホスト B であるとする、そのホスト上の OHS と OC4J をシャットダウンします。これで、ホスト C で実行される OHS と OC4J にのみアクセスできるようになります。
3. Oracle Forms クライアントを実行している同じブラウザを使用して、Oracle Forms アプリケーション D に再度アクセスします。リクエストは失敗し、Forms クライアントはセッションを失います。Oracle Forms セッションの状態は、OC4J インスタンス間でレプリケートされません。
4. 次に、ブラウザを使用して、新しい Forms セッションを開始します。Web Cache はホスト C で実行される残りの OHS と OC4J にリクエストをダイレクトします。アプリケーションが適切に動作することを確認します。
5. ホスト B で OHS と OC4J を再度起動します。ブラウザを使用して、Web Cache Manager にログインします。ナビゲータ・ペインで、「Monitoring」→「Health Monitor」を選択します。
6. 「Health Monitor」画面で、ホスト B が「UP」にマークされていることを確認します。

Web Cache の詳細は、『Oracle Application Server Web Cache 管理者ガイド』を参照してください。

---

# 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、Oracle 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 Java 仮想マシン (JVM) ではなく、クライアント上の特定の 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、プラグイン形式で代替の 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。

## A.1 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、Oracle 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、Oracle 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だけでなく、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、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）。

## A.2 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、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。

## A.3 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、Oracle Application Server Forms Services 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、Oracle Forms アプリケーションを信頼できるアプレットとして 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。

## A.4 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。

### A.4.1 Windows 98、NT、2000 および XP:

- Netscape Navigator 4.7.x
- Netscape Navigator 4.7.x
- Internet Explorer 5.x
- Internet Explorer 6.0

## A.5 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)

## A.6 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 はアプレットをレンダリングするために、<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。

## A.7 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、アプレットをレンダリングするために <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。



## A.8 Oracle JInitiator プラグインの設定

Oracle JInitiator プラグインを設定するには：

- 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。

### A.8.1 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には：

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 B.3 項「[base.htm](#) ファイル、[basejini.htm](#) ファイル、[basejpi.htm](#) ファイルおよび [baseie.htm](#) ファイル」を参照してください。

### A.8.2 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には：

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。

### A.8.3 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には：

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。

## A.9 Oracle JInitiator プラグインの変更

**Oracle JInitiator プラグインを変更するには：**

- 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。

### A.9.1 Oracle JInitiator キャッシュ・サイズの変更

**Oracle JInitiator のキャッシュ・サイズを変更するには：**

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. 「実行時のパラメータ」フィールドで、Dcache.size 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。

### A.9.2 Oracle JInitiator ヒープ・サイズの変更

**Oracle JInitiator のヒープ・サイズを変更するには：**

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. 「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。

## A.9.3 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には：

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。

## A.9.4 Oracle JInitiator 出力の表示

Oracle JInitiator 出力を表示するには：

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・ボックスを選択します。

## A.10 ベース HTML ファイルの変更

JInitiator を使用して Oracle 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取り、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取ら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 B.3 項「[base.htm ファイル](#)、[basejini.htm ファイル](#)、[basejpi.htm ファイル](#)および [baseie.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。



---

---

## サンプル構成ファイル

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。

- デフォルトの formsweb.cfg ファイル
- プラットフォーム固有の default.env ファイル
- base.htm ファイル、basejini.htm ファイル、basejpi.htm ファイルおよび baseie.htm ファイル
- web.xml
- forms90.conf
- Registry.dat

## B.1 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 defines parameter values used by the FormsServlet (f90servlet)
# This section defines the Default settings. Any of them may be overridden in
# the following Named Configuration sections. If they are not overridden, then
# the values here will be used.
# The default settings comprise two types of parameters: System parameters,
# which cannot be overridden in the URL, and User Parameters, which can.
# Parameters which are not marked as System parameters are User parameters.
# 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.
# 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)
[default]
# System parameter: default base HTML file
baseHTML=base.htm
# System parameter: base HTML file for use with JInitiator client
baseHTMLjinitiator=basejini.htm
# System parameter: base HTML file for use with Sun's Java Plug-In
baseHTMLjpi=basejpi.htm
# System parameter: base HTML file for use with Microsoft Internet Explorer
# (when using the native JVM)
baseHTMLie=baseie.htm
# System parameter: delimiter for parameters in the base HTML files
HTMLdelimiter=%
# System parameter: working directory for Forms runtime processes
# WorkingDirectory defaults to <oracle_home>/forms90 if unset.
workingDirectory=
# System parameter: file setting environment variables for the Forms runtime
# processes
envFile=default.env
# System parameter: JVM option for Microsoft Internet Explorer.
# This parameter specifies how to execute the Forms applet under
# Microsoft Internet Explorer 5.x or above. Put IE=native if you want
# the Forms applet to run in the browser's native JVM.
IE=JInitiator
```

```
# Forms runtime argument: whether to escape certain special characters
# in values extracted from the URL for other runtime arguments
escapeparams=true
# Forms runtime argument: which form module to run
form=test.fmx
# Forms runtime argument: database connection details
userid=
# Forms runtime argument: whether to run in debug mode
debug=no
# Forms runtime argument: host for debugging
host=
# Forms runtime argument: port for debugging
port=
# Other Forms runtime arguments: grouped together as one parameter.
# These settings support running and debugging a form from the Builder:
otherparams=buffer_records=%buffer% debug_messages=%debug_messages% array=%array%
obr=%obr% query_only=%query_only% quiet=%quiet% render=%render% record=%record%
tracegroup=%tracegroup% log=%log% term=%term%
# Sub argument for otherparams
buffer=no
# Sub argument for otherparams
debug_messages=no
# Sub argument for otherparams
array=no
# Sub argument for otherparams
obr=no
# Sub argument for otherparams
query_only=no
# Sub argument for otherparams
quiet=yes
# Sub argument for otherparams
render=no
# Sub argument for otherparams
record=
# Sub argument for otherparams
tracegroup=
# Sub argument for otherparams
log=
# Sub argument for otherparams
term=
# HTML page title
pageTitle=Oracle Application Server Forms Services
# HTML attributes for the BODY tag
HTMLbodyAttrs=
# HTML to add before the form
HTMLbeforeForm=
# HTML to add after the form
```

```
HTMLafterForm=
# Forms applet parameter: URL path to Forms ListenerServlet
serverURL=/forms90/190servlet
# Forms applet parameter
codebase=/forms90/java
# Forms applet parameter
imageBase=DocumentBase
# Forms applet parameter
width=750
# Forms applet parameter
height=600
# Forms applet parameter
separateFrame=false
# Forms applet parameter
splashScreen=
# Forms applet parameter
background=
# Forms applet parameter
lookAndFeel=Oracle
# Forms applet parameter
colorScheme=teal
# Forms applet parameter
logo=
# Forms applet parameter
restrictedURLparams=HTMLbodyAttrs,HTMLbeforeForm,pageTitle,HTMLafterForm,log,allow_
debug,allowNewConnections
# Forms applet parameter
formsMessageListener=
# Forms applet parameter
recordFileName=
# Forms applet parameter
serverApp=default
# Forms applet archive setting for JInitiator
archive_jini=f90all_jinit.jar
# Forms applet archive setting for Microsoft Internet Explorer native JVM
archive_ie=f90all.cab
# Forms applet archive setting for other clients (Sun Java Plugin, Appletviewer,
# etc)
archive=f90all.jar
# Number of times client should retry if a network failure occurs. You should
# only change this after reading the documentation.
networkRetries=0
# Page displayed to Netscape users to allow them to download Oracle JInitiator.
# Oracle JInitiator is used with Windows clients.
# If you create your own page, you should set this parameter to point to it.
jinit_download_page=/forms90/jinitiator/us/jinit_download.htm
# Parameter related to the version of JInitiator
```



```
jinit_classid=clsid:CAFECAFE-0013-0001-0017-ABCDEFABCDEF
# Parameter related to the version of JInitiator
jinit_exename=jinit.exe#Version=1,3,1,17
# Parameter related to the version of JInitiator
jinit_mimetype=application/x-jinit-applet;version=1.3.1.17
# Page displayed to users to allow them to download Sun's Java Plugin.
# Sun's Java Plugin is typically used for non-Windows clients.
# (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
# Parameter related to the version of the Java Plugin
jpi_classid=clsid:8AD9C840-044E-11D1-B3E9-00805F499D93
# Parameter related to the version of the Java Plugin
jpi_
codebase=http://java.sun.com/products/plugin/1.3/jinstall-13-win32.cab#Version=1,3,0
,0
# Parameter related to the version of the Java Plugin
jpi_mimetype=application/x-java-applet;version=1.3

# EM config parameter
# Set this to "1" to enable Enterprise Manager to track Forms processes
em_mode=0
# Single Sign-On OID configuration parameter
oid_formsid=formsApp_adtqa_ui7.us.oracle.com_6A5E0A34DCD44048B706A0ECE46EC3A6
# Single Sign-On OID configuration parameter
oracle_home=D:\AS10g_M30_bif
# Single Sign-On OID configuration parameter
formsid_group_dn=cn=Logical Application Group, orclApplicationCommonName=formsApp_
adtqa_ui7.us.oracle.com_6A5E0A34DCD44048B706A0ECE46EC3A6, cn=forms, cn=Products,
cn=OracleContext
# Single Sign-On OID configuration parameter: indicates whether we allow
# dynamic resource creation if the resource is not yet created in the OID.
ssoDynamicResourceCreate=true
# Single Sign-On parameter: URL to redirect to if ssoDynamicResourceCreate=false
ssoErrorUrl=
# Single Sign-On parameter: Cancel URL for the dynamic resource creation DAS page.
ssoCancelUrl=
# Single Sign-On parameter: indicates whether the url is protected in which
# case mod_osso will be given control for authentication or continue in
# the FormsServlet if not. It is false by default. Set it to true in an
# application-specific section to enable Single Sign-On for that application.
ssoMode=false
# The parameter allow_debug determines whether debugging is permitted.
# Administrators should set allow_debug to "true" if servlet
# debugging is required, or to provide access to the Forms Trace Xlate utility.
# Otherwise these activities will not be allowed (for security reasons).
allow_debug=false
# Parameter which determines whether new Forms sessions are allowed.
```

```
# This is also read by the Forms EM Overview page to show the
# current Forms status.
allowNewConnections=true
# Example Named Configuration Section
# Example 1: configuration to run forms in a separate browser window with
# "generic" look and feel (include "config=sepwin" in the URL)
# 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".
[sepwin]
separateFrame=True
lookandfeel=Generic
# Example Named Configuration Section
# 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 Named Configuration Section
# 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 Named Configuration Section
# Example 4: configuration running the Forms ListenerServlet in debug mode
# (debug messages will be written to the servlet engine's log file).
[debug]
serverURL=/forms90/l90servlet/debug
```

## B.2 プラットフォーム固有の default.env ファイル

default.env には、プラットフォーム固有の 2 つのバージョンがあります。

- デフォルトの Windows 用 default.env ファイル
- デフォルトの UNIX 用 default.env ファイル

### B.2.1 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=<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=<ORACLE_HOME>/forms90

#
# The PATH setting is required in order to pick up the JVM (jvm.dll).
# The Forms runtime executable and dll's are assumed to be in
# <ORACLE_HOME>%bin if they are not in the PATH.
# In addition, if you are running Graphics applications, you will need
# to append the following to the path (where <Graphics Oracle Home> should
# be replaced with the actual location of your Graphics 6i <ORACLE_HOME>):
#
# ;<Graphics Oracle Home>%bin;<Graphics Oracle Home>%jdk%bin
#
```

```
PATH=<ORACLE_HOME>%bin;<ORACLE_HOME>%jdk%jre%bin%client

#
# Settings for Graphics
# -----
# NOTE: These settings are only needed if Graphics applications
# are called from Forms applications. In addition, you will need to
# modify the PATH variable above as described above.
#

#
# 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 forms tracing and logging
# -----
# Note: This entry has to be uncommented to enable tracing and
# logging.

#FORMS90_TRACE_PATH=<ORACLE_HOME>%forms90%server

#
# System settings
# -----
# You should not normally need to modify these settings
#
FORMS90=<ORACLE_HOME>%forms90

#
# Java class path
# This is required for the Forms debugger
# You can append your own Java code here)
# f90srv.jar, repository.jar and ldapjclnt9.jar are required for
# the password expiry feature to work(#2213140).
#
CLASSPATH=<ORACLE_HOME>%j2ee%OC4J_BI_
Forms%applications%forms90app%forms90web%WEB-INF%lib%f90srv.jar;<ORACLE_
HOME>%jlib%repository.jar;<ORACLE_HOME>%jlib%ldapjclnt9.jar;<ORACLE_
```

```
HOME>%jlib%debugger.jar;<ORACLE_HOME>%jlib%ewt3.jar;<ORACLE_
HOME>%jlib%share.jar;<ORACLE_HOME>%jlib%utj90.jar;<ORACLE_
HOME>%jlib%zrclient.jar;<ORACLE_HOME>%reports%jlib%rwrun.jar
```

## B.2.2 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=<ORACLE_HOME>

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

# Java class path
# This is required for the Forms debugger
# You can append your own Java code here)
# f90srv.jar, repository.jar and ldapjclnt9.jar are required for
# the password expiry feature to work(#2213140).
#
CLASSPATH=<ORACLE_HOME>%j2ee%OC4J_BI_
Forms%applications%forms90app%forms90web%WEB-INF%lib%f90srv.jar;<ORACLE_
HOME>%jlib%repository.jar;<ORACLE_HOME>%jlib%ldapjclnt9.jar;<ORACLE_
HOME>%jlib%debugger.jar;<ORACLE_HOME>%jlib%ewt3.jar;<ORACLE_
HOME>%jlib%share.jar;<ORACLE_HOME>%jlib%utj90.jar;<ORACLE_
HOME>%jlib%zrclient.jar;<ORACLE_HOME>%reports%jlib%rwrun.jar

#
# The PATH setting is not required for Forms if the Forms executables are
# in <ORACLE_HOME>/bin. However, it is required if Graphics applications
# are called from Forms applications.
#
PATH=<ORACLE_HOME>/bin
```

```
#
# Settings for Reports
# -----
# NOTE: This setting is only needed if Reports applications
# are called from Forms applications
# However, because of bug 2336698 where a report is started from
# a forms debugger session with an already running JVM, then
# the report's class path should also be included in the forms
# class path.
REPORTS_CLASSPATH=<ORACLE_HOME>/jlib/zrclient.jar:<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 forms tracing and logging
# -----
# Note: This entry has to be uncommented to enable tracing and
# logging.

#FORMS90_TRACE_PATH=<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=<ORACLE_HOME>/lib:%O_JDK_HOME%/jre/lib/sparc:%O_JDK_
HOME%/jre/lib/sparc/native_threads:%LD_LIBRARY_PATH%
```

## B.3 base.htm ファイル、basejini.htm ファイル、basejpi.htm ファイルおよび baseie.htm ファイル

base.htm、basejini.htm、basejpi.htm および baseie.htm の概説とディレクトリは、[第 3.1.1.3 項「base.htm、basejini.htm、basejpi.htm および baseie.htm」](#) を参照してください。

OracleAS 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 Oracle 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 によって読み取られます。

ベース HTML ファイル内の変数 (`%variablename%`) は、[第 4.3.1 項「Application Server Control によるパラメータの構成」](#)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 リクエスト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、[第 4.3.1 項「Application Server Control によるパラメータの構成」](#) 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](#)」の項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 Oracle 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](#)」の項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 のタグを含んだベース HTML ファイルです。例は、「[デフォルトの baseie.htm ファイル](#)」の項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、[第 3.1.1.2 項「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。



## B.3.1 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、formsweb.cfg ファイルに実際のパラメータ値を指定します。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。

### B.3.1.1 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、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。

## B.3.2 デフォルトの base.htm ファイル

```
<HTML>
<!-- FILE: base.htm (Oracle Forms) -->
<!-- -->
<!-- This is the default base HTML 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="%escapeParams% module=%form% userid=%userid% sso_userid=%sso_userid%
sso_formsid=%sso_formsid% sso_subDN=%sso_subDN% sso_usrDN=%sso_usrDN% debug=%debug%
host=%host% port=%port% %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>
```

### B.3.3 デフォルトの basejini.htm ファイル

```

<HTML>
<!-- FILE: basejini.htm (Oracle Forms) -->
<!--
-->
<!-- This is the default base HTML 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="/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="%escapeParams% module=%form% userid=%userid% sso_userid=%sso_userid%
sso_formsid=%sso_formsid% sso_subDN=%sso_subDN% sso_usrDN=%sso_usrDN% debug=%debug%
host=%host% port=%port% %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="" PLUGINSPAGE="%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="%escapeParams% module=%form% userid=%userid% sso_userid=%sso_
userid% sso_formsid=%sso_formsid% sso_subDN=%sso_subDN% sso_usrDN=%sso_usrDN%
debug=%debug% host=%host% port=%port% %otherparams%"
  separateFrame="%separateFrame%"
  splashScreen="%splashScreen%"
  background="%background%"
  lookAndFeel="%lookAndFeel%"
  colorScheme="%colorScheme%"
  serverApp="%serverApp%"
  logo="%logo%"
  imageBase="%imageBase%"
  formsMessageListener="%formsMessageListener%"
  recordFileName="%recordFileName%"
>
<NOEMBED>
</COMMENT>
</NOEMBED></EMBED>
</OBJECT>
<!-- Forms applet definition (end) -->

%HTMLafterForm%

</BODY>
</HTML>
```

## B.3.4 デフォルトの basejpi.htm ファイル

```

<HTML>
<!-- FILE: basejpi.htm (Oracle Forms) -->
<!--
<!-- This is the default base HTML 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="%escapeParams% module=%form% userid=%userid% sso_userid=%sso_userid%
sso_formsid=%sso_formsid% sso_subDN=%sso_subDN% sso_usrDN=%sso_usrDN% debug=%debug%
host=%host% port=%port% %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="" PLUGINSPAGE="%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="%escapeParams% module=%form% userid=%userid% sso_userid=%sso_
userid% sso_formsid=%sso_formsid% sso_subDN=%sso_subDN% sso_usrDN=%sso_usrDN%
debug=%debug% host=%host% port=%port% %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>

```

## B.3.5 デフォルトの baseie.htm ファイル

```

<HTML>
<!-- FILE: baseie.htm (Oracle Forms) -->
<!--
<!-- This is the default base HTML 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="cabbase" VALUE="%archive_ie%">
<PARAM NAME="serverURL" VALUE="%serverURL%">
<PARAM NAME="networkRetries" VALUE="%networkRetries%">
<PARAM NAME="serverArgs"
        VALUE="%escapeParams% module=%form% userid=%userid% sso_userid=%sso_userid%
sso_formsid=%sso_formsid% sso_subDN=%sso_subDN% sso_usrDN=%sso_usrDN% debug=%debug%
host=%host% port=%port% %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>
```

## B.4 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。

Oracle Developer Suite (OracleDS) または OracleAS をインストールしても Oracle 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ことが役立つ場合があります。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「サーブレット管理」ページ ([http://<your\\_web\\_server\\_hostname>:<port>/forms90/f90servlet/admin](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" サーブレット・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 Oracle Forms を実行する。

Oracle 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 などのサーブレット・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）。



## B.4.1 デフォルトの web.xml ファイル

```

- <web-app>
  <display-name>Forms Services</display-name>
  <description>OracleAS: Forms Services</description>
- <welcome-file-list>
  <welcome-file>l90servlet</welcome-file>
</welcome-file-list>
- <!-- Forms 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/formswb.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 listener servlet
  -->
- <servlet>
  <servlet-name>l90servlet</servlet-name>
  <servlet-class>oracle.forms.servlet.ListenerServlet</servlet-class>

```

```
    </servlet>
- <!--
  Forms 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 />
</context-param>
</web-app>
```

## B.5 forms90.conf

forms90.conf の説明とディレクトリは、[第 3.1.3.1 項「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 B-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。

### B.5.1 デフォルトの forms90.conf

```
# Name
# forms90.conf
# Purpose
# Apache mod_oc4j and mod_jserv configuration file for Forms 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/(.*) "<ORACLE_HOME>/forms90/java/$1"

# Virtual path for JInitiator downloadable executable and download page
AliasMatch ^/jinitiator/(.*) "<ORACLE_HOME>/jinit/$1"
```

```

    # Virtual path for runform.htm (used to run a form for testing purposes)
AliasMatch ^/forms90/html/(.*) "<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 OC4J_BI_Forms
    Oc4jMount /forms90/f90servlet OC4J_BI_Forms
    Oc4jMount /forms90/f90servlet/* OC4J_BI_Forms
    Oc4jMount /forms90/l90servlet OC4J_BI_Forms
    Oc4jMount /forms90/l90servlet/* OC4J_BI_Forms
</IfModule>

```

## B.6 Registry.dat

Registry.dat の説明とディレクトリは、[第 3.1.4.1 項「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 ([第 4.9.1 項「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, MonoSpaced, Dia
log, 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、[第 B.6.1 項「デフォルトの Registry.dat」](#) を参照してください。

### B.6.1 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
```





## 数字

3 層のアーキテクチャ, 1-5

## A

allow\_debug, トレース・ログの表示, 4-13

archive\_ie パラメータ, 4-17

archive\_jinit パラメータ, 4-16

archive パラメータ, 4-16

## B

background パラメータ, 4-16

base.htm, 3-3, B-11

説明, B-12

例, B-13

baseie.htm, 3-3

説明, B-12

例, B-19

basejini.htm, 3-3, B-11

説明, B-12

例, B-15

basejpi.htm, 3-3

説明, B-12

basejpi.htm ファイル

サンプル・デフォルト, B-17

## C

CAB ファイル, 8-12

CodeBase, 4-32

codebase パラメータ, 4-15

colorScheme パラメータ, 4-16

## D

DCM プロセス

再起動, 7-3

default.env

UNIX のサンプル, B-9

Windows のサンプル・デフォルト, B-7

## E

EAR, 3-4

em\_mode, 4-18

## F

f90all\_jinit.jar

説明, 3-16

f90all.jar

説明, 3-16

FORM90\_PATH, 4-20

Forms Listener, 1-6, 1-7

Forms Listener Servlet, 1-7

HTTPS, 5-6

クライアント要件, 5-7

サーバー要件, 5-6

Forms Runtime Diagnostics, 7-1

Forms Services

Web ランタイム・プーリング, 8-4

イベントの監視, 8-3

インスタンスの監視, 8-2

メトリック情報の検索, 8-4

メトリック情報のソート, 8-4

ユーザー・セッションの監視, 8-3

Forms Servlet, 5-2

Forms Services で使用するアイコンとイメージの配置  
, 4-26

Forms Services の最適化, 8-2  
Forms Services のメトリック  
監視, 7-14  
Forms Services リソース要件, 8-6  
Forms Trace, 3-3, 7-1, 7-2  
forms90.conf, B-23  
説明, 3-5  
デフォルト・サンプル, B-23  
formsMessageListener, 4-15  
FormsServlet.initArgs, 4-6  
formswb.cfg, 3-3  
例, B-2  
Forms の統合  
Web Cache, 8-17  
Forms ランタイム・エンジン, 1-6, 1-7  
Forms ランタイム・プロセス, 1-7  
Forms リソース  
OID のデフォルト・プリファレンスでの定義, 6-2  
FRD, 7-1  
ftrace.cfg, 3-3

## G

---

Graphics, 4-25

## H

---

height パラメータ, 4-15  
HTML ベースの Oracle Enterprise Manager, 4-2  
HTTP Listener, 5-2  
構成ファイル, 3-5  
HTTPD, 5-3  
HTTPS  
Forms Listener Servlet, 5-6

## I

---

imageBase, 4-15  
Internet Explorer と JInitiator, A-4

## J

---

J2EE, 5-1  
JAR ファイル, 8-11  
JAR ファイル, キャッシュ, 8-12  
Java Plug-in, 8-12  
Java クライアント・リソース要件, 8-6

jinit\_classid, 4-17  
jinit\_download\_page, 4-17  
jinit\_exename, 4-17  
jinit\_mimetype, 4-17  
JInitiator, 8-11  
説明, 3-15  
JInitiator のキャッシュ・サイズ, A-6  
JInitiator の説明, A-2  
JInitiator のヒープ・サイズ, A-6  
JInitiator プロキシ・サーバー, A-7  
jpi\_classid, 4-18  
jpi\_codebase, 4-18  
jpi\_download\_page, 4-18

## L

---

Logging Tool, 7-1  
logo, 4-15  
lookAndFeel パラメータ, 4-16

## M

---

mapFonts, 4-17  
MENU\_BUFFERING を使用不可にする, 8-15

## N

---

networkRetries, 4-17

## O

---

OC4J, 5-1  
構成ファイル, 3-4  
ロード・バランシング, 5-3  
OC4J サーバー・プロセス, 5-1  
OC4J のロード・バランシング, 5-1  
OEM (Oracle Enterprise Manager), 4-2  
OID, 2-2, 6-1  
Forms リソースを定義するデフォルト・プリファレンス, 6-2  
構成のオプション, 2-4  
動的リソースの作成, 2-3  
oid\_formsids, 4-18  
Oracle Database, 1-3  
Oracle HTTP Listener の構成ファイル, 3-5  
Oracle Identity Management Infrastructure, 2-4  
Oracle Internet Directory, 6-1

Oracle JInitiator, 8-11, A-2  
  Internet Explorer での使用方法, A-4  
  Netscape Navigator での使用方法, A-4  
  概要, 3-15  
  キャッシュ・サイズの変更, A-6  
  サポートされる構成, A-3  
  システム要件, A-3  
  出力の表示, A-7  
  ヒープ・サイズの変更, A-6  
  プラグインの設定, A-5  
  利点, A-2  
Oracle Real Application Clusters, 1-3  
Oracle Single Sing-On Server, 6-1  
ORACLE\_GRAPHICS6I\_HOME, 4-21  
ORACLE\_HOME, 4-20  
oracle\_home, 4-18  
OracleAS, 1-2  
OracleAS Forms Services アーキテクチャ, イメージ  
  , 1-5  
OracleDS, 1-2  
Oracle インターネット・プラットフォーム, 1-2

## P

---

PATH, 4-20  
PECS, 7-1  
Performance Event Collection Services, 7-1

## R

---

RAD エントリ, 2-2  
recordFileName, 4-15  
Registry.dat  
  説明, 3-5  
registry.dat, 3-5, B-25  
  サンプル・デフォルト, B-25  
REPORTS\_CLASSPATH, 4-21  
restrictedURLparams, 4-15

## S

---

separateFrame パラメータ, 4-16  
serverApp パラメータ, 4-16  
serverArgs パラメータ, 4-12  
serverURL, 4-15  
Servlet Logging Tools, 7-1, 7-15  
splashScreen パラメータ, 4-16

SSO, 6-1  
  Forms からのアクセス, 6-7  
  アプリケーションでの有効化, 6-3  
  データベース・パスワード期限切れ, 2-3, 6-3  
  動的ディレクティブ, 6-3  
  認証フロー, 6-8  
ssoCancelUrl, 6-7  
ssoDynamicResourceCreate  
  概要, 6-6  
ssoErrorURL, 6-7  
ssoMode  
  概要, 6-4  
ssoMode パラメータ  
  特定のアプリケーションでの有効化の例, 6-4  
Sun 社の Java Plug-in, 8-12

## U

---

Upload/Translate ユーティリティ  
  起動, 7-8  
URL エスケープ・シーケンス, 3-10

## V

---

VGS ツリー, 8-8

## W

---

Web Cache  
  Forms の統合, 8-17  
  セッション・バインドの構成, 8-17  
  設定のテスト, 8-18  
web.xml, 3-4, B-20  
web.xml ファイル  
  デフォルト・サンプル, B-21  
Web 上の Forms アプリケーションの機能制限, 4-33  
width パラメータ, 4-15

## あ

---

アイコン, 4-26  
  配置, 4-26  
アプリケーション  
  サーバー, 1-5  
アプリケーションの配布  
  概要, 3-5  
  手順, 3-6

アプレット

パラメータ, 4-15

暗号化されたプログラム単位, 8-6

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, 8-7

イメージ, 4-26

スプラッシュ画面, 4-29

バックグラウンド, 4-29

上書き, 3-8

## か

---

仮想グラフィック・システム (VGS) ツリー, 8-8

期間イベント, 7-8

起動時間, 8-10

クライアント層, 1-5

クライアント・ブラウザのサポート

概要, 3-15

クライアント・リソース要件, 8-6

権限

ユーザーのクラス, 2-2

権限およびアクセス強制, 2-4

言語検出, 4-33

構成パラメータ

ベース HTML ファイルとクライアント・ブラウザ  
, 3-16

構成ファイル, 3-2

## さ

---

サーブレットのログ・ファイル

サンプル出力, 7-17

ディレクトリ, 7-16

サーブレットのログ・ファイルのディレクトリ, 7-16

サンプル値, 3-8

サンプル・ファイル

base.htm, B-13, B-19

basejinit.htm, B-15

指定, 3-7

使用, 2-4

シングル・サインオン, 6-1

診断ツール, 7-1

スプラッシュ画面, 4-29

## た

---

タイマー, チューニング, 8-15

中間層, 1-5

チューニング, 8-1

JAR ファイルの使用, 8-11

MENU\_BUFFERING, 8-15

MENU\_BUFFERING を使用不可にする, 8-15

アプリケーションのサイズ, 8-16

初期画面, 8-14

タイマー, 8-15

ナビゲーションの削減, 8-14

ネットワーク帯域幅の削減, 8-13

ボイラープレート・オブジェクトの削減, 8-14

ボイラープレート項目, 8-14

メッセージ順序, 8-13

類似点の活用, 8-14

テンプレート HTML

考慮事項, 静的, 3-12

テンプレート HTML ファイル

考慮事項, 3-11

作成, 4-25

データ・ストリーム圧縮, 8-13

データ・セグメント, 8-6

データベース層

説明, 1-5

デフォルト動作, 3-9

デフォルトの formsweb.cfg ファイル

サンプル, B-2

デフォルトの構成パラメータ

allow\_debug, 4-13

array, 4-13

baseHTML, 4-10

baseHTMLjinitiator, 4-10, 4-17

baseHTMLjpi, 4-10

buffer, 4-13

connectionDisallowedURL, 4-10

debug, 4-12

debug\_messages, 4-13

defaultcharset, 4-11

em\_trace, 4-14

envFile, 4-11

escapeparams, 4-12

form, 4-12

heartBeat, 4-12

host, 4-13

HTML delimiter, 4-10

- HTMLafterForm, 4-14
- HTMLbeforeForm, 4-14
- HTMLbodyAttrs, 4-14
- ie50, 4-11
- log, 4-11, 4-14
- otherparams, 4-12
- pageTitle, 4-14
- port, 4-13
- query\_only, 4-13
- quiet, 4-13
- record, 4-13
- render, 4-13
- term, 4-14
- tracegroup, 4-13
- USERID, 4-12
- workingDirectory, 4-11
- 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
  - XML に変換, 7-8
- トレースの tracegroup パラメータ, 7-6
- トレースの URL パラメータ・オプション, 7-6
- トレースのログ・パラメータ, 7-6
- トレース用 translate ユーティリティ, 7-8

## な

---

- ネットワーク
  - 帯域幅の削減, 8-13
- ネットワーク使用量, 8-7
- ネットワークの待機時間, 8-7
- ネットワーク・パケット, 8-7

## は

---

- 配布
  - フォームと Web, 3-1
- バックグラウンド, 4-29
- パフォーマンス / スケーラビリティのチューニング, 5-1
- パフォーマンス・ツール, 7-1
- パラメータ, 3-7, 3-8
- パラメータ・オプション
  - URL での指定, 7-6
- ビルトイン・イベント, 7-9
- ベース HTML ファイル

- 作成, B-11, B-12
- 選択, 3-16
- パラメータと変数, B-13
- 変更, A-7
- 変数の変更, B-11
  - リスト, 3-3
- 保護, 2-3
- ボイラープレート・オブジェクト / イメージ, 8-6
- ポイント・イベント, 7-8

## ま

---

- メトリックのロギング
  - URL での指定, 7-16
  - 有効化, 7-15

## や

---

- ユーザー ID / パスワード機能
  - 設定, 5-7

## ら

---

- ランタイム・プーリング, 1-4
  - 事前起動パラメータの構成, 8-4
- ランフォーム・パラメータ, 3-9, 3-10, 4-12
  - 定義, 3-9
  - デフォルト動作, 3-10
  - デフォルト動作, 前リリース, 3-11
  - 特殊文字の値, 3-9
- リソース, 2-3
  - 動的ディレクティブ, 2-3
- リソース, 最小化
  - 暗号化されたプログラム単位, 8-6
  - 画面のレンダリング, 8-8
  - データ・セグメント, 8-6
  - ネットワーク使用量, 8-7
  - パケットの送信, 8-7
  - ボイラープレート・オブジェクト, 8-6
- ロギング機能, 7-15

