

Oracle® Application Server Forms Services

利用ガイド

10g リリース 2 (10.1.2)

部品番号 : B25065-02

2006 年 8 月

Oracle Application Server Forms Services 利用ガイド, 10g リリース 2 (10.1.2)

部品番号 : B25065-02

原本名 : Oracle Application Server Forms Services Deployment Guide, 10g Release 2 (10.1.2)

原本部品番号 : B14032-03

原本著者 : Orlando Cordero

原本協力者 : Suvarna Balachandra, Nishad Desai, Pam Gamer, Art Housinger, Phil Kuhn, Chris Lewis, Hiro Nozaki, Gururaja Padakandla, Ganesh Puram, Slava Podokshik, Grant Ronald, Ananth Satyanarayana, Naseer Syed, Sudarshan Upadhyaya, Robin Zimmermann

Copyright © 2006, Oracle. 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は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、JD Edwards、PeopleSoft、Retek は米国 Oracle Corporation およびその子会社、関連会社の登録商標です。その他の名称は、他社の商標の可能性ががあります。

このプログラムは、第三者の Web サイトへリンクし、第三者のコンテンツ、製品、サービスへアクセスすることがあります。オラクル社およびその関連会社は第三者の Web サイトで提供されるコンテンツについては、一切の責任を負いかねます。当該コンテンツの利用は、お客様の責任になります。第三者の製品またはサービスを購入する場合は、第三者と直接の取引となります。オラクル社およびその関連会社は、第三者の製品およびサービスの品質、契約の履行（製品またはサービスの提供、保証義務を含む）に関しては責任を負いかねます。また、第三者との取引により損失や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の責任を負いかねます。

目次

はじめに	xi
対象読者	xii
ドキュメントのアクセシビリティについて	xii
関連ドキュメント	xii
表記規則	xii
サポートおよびサービス	xiii
1 概要	
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	1-2
1.1.2 Oracle Developer Suite	1-2
1.1.3 Oracle Database 10g	1-2
1.2 Oracle Application Server Forms Services	1-2
1.2.1 Forms Services の新機能	1-3
1.3 OracleAS Forms Services のアーキテクチャ	1-3
1.4 OracleAS Forms Services のコンポーネント	1-4
1.4.1 Forms Listener Servlet	1-4
1.4.2 Forms ランタイム・プロセス	1-4
1.5 Forms Listener Servlet	1-5
2 Forms Services セキュリティの概要	
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	2-2
2.1.3.1 動的ディレクティブ	2-3
2.1.3.2 Oracle Internet Directory における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	2-3
2.1.5 Oracle Identity Management Infrastructure の使用	2-3
2.2 OracleAS Forms Services のセキュリティの構成	2-3
2.2.1 Oracle Forms の Oracle Identity Management オプションの構成	2-3
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	OracleAS Forms Services のアクション	3-2
3.2	構成ファイル	3-3
3.2.1	Oracle Forms 構成ファイル	3-4
3.2.1.1	default.env	3-4
3.2.1.2	formsweb.cfg	3-4
3.2.1.3	base.htm、basejini.htm、および basejpi.htm	3-4
3.2.1.4	ftrace.cfg	3-5
3.2.2	Oracle Application Server Containers for J2EE (OC4J) 構成ファイル	3-5
3.2.2.1	web.xml	3-5
3.2.2.2	Oracle Forms OC4J ファイルのディレクトリ構造	3-5
3.2.3	Oracle HTTP Listener の構成ファイル	3-6
3.2.3.1	forms.conf	3-6
3.2.4	標準フォントとアイコン・ファイル	3-6
3.2.4.1	Registry.dat	3-6
3.2.5	WebUtil 構成ファイル	3-6
3.2.5.1	デフォルトの webutil.cfg	3-6
3.2.5.2	デフォルトの webutilbase.htm	3-7
3.2.5.3	デフォルトの webutiljini.htm	3-7
3.2.5.4	デフォルトの webutiljpi.htm	3-7
3.3	アプリケーションの配布	3-7
3.3.1	アプリケーションの配布	3-7
3.3.2	パラメータの指定	3-8
3.3.3	Oracle Enterprise Manager での構成セクションの作成	3-9
3.3.3.1	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	3-10
3.3.4	ランフォーム・パラメータにおける特殊文字の指定	3-10
3.3.4.1	現行リリースのデフォルト動作	3-10
3.3.4.2	前リリースの動作	3-11
3.3.4.3	現行リリースでの前リリースの動作取得	3-11
3.3.4.4	テンプレート HTML ファイルの考慮事項	3-12
3.3.4.5	静的 HTML ページの考慮事項	3-12
3.4	クライアント・ブラウザのサポート	3-12
3.4.1	Oracle JInitiator	3-13
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	3-13

4 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 Enterprise Manager Application Server Control 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	4-3
4.2.3	Application Server Control コンソールによる Forms Services へのアクセス	4-4
4.3	Forms Services の構成	4-4
4.3.1	Application Server Control コンソールによるパラメータの構成	4-5
4.3.1.1	ファイルを指定するパラメータ	4-5
4.3.2	構成セクションの管理	4-5
4.3.2.1	名前を付けた構成の複製	4-6

4.3.2.2	名前を付けた構成の削除	4-6
4.3.3	パラメータの管理	4-7
4.3.4	デフォルトの Forms 構成パラメータ	4-8
4.3.4.1	システムのデフォルトの構成パラメータ	4-8
4.3.4.2	ランフォーム・パラメータ (serverArgs パラメータ)	4-10
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	4-12
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	4-12
4.3.4.5	JInitiator のパラメータ	4-14
4.3.4.6	Sun 社の Java Plug-in のパラメータ	4-14
4.3.4.7	Oracle Enterprise Manager の構成パラメータ	4-14
4.3.4.8	Oracle Internet Directory の構成パラメータ	4-15
4.4	Oracle Enterprise Manager による環境変数の構成	4-15
4.5	ユーザー・セッションの管理	4-17
4.5.1	新規ユーザー・セッションの許可	4-17
4.5.2	新規ユーザー・セッションの禁止	4-17
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	4-18
4.6	アプリケーションの URL セキュリティの管理	4-18
4.6.1	Oracle Forms テスト・フォームの保護	4-19
4.7	ユーザー独自のテンプレート HTML ファイルの作成	4-21
4.8	Oracle Forms アプリケーションへの Graphics の挿入	4-21
4.8.1	Oracle Graphics 6i および Oracle Database 9.0.1.4.0 (64 ビット)	4-21
4.8.2	Reports Server による Graphics 6i の構成	4-21
4.9	Forms Services で使用するアイコンとイメージの配置	4-22
4.9.1	Application Server Control を使った Registry.dat の管理	4-22
4.9.2	アプリケーション・アイコンの配置	4-22
4.9.2.1	アイコンの Java アーカイブ・ファイルへの格納	4-23
4.9.2.2	Registry.dat へのアイコン変更の追加	4-23
4.9.3	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	4-24
4.9.4	アイコンとイメージを含むカスタム JAR ファイル	4-25
4.9.4.1	イメージの JAR ファイルの作成	4-25
4.9.4.2	JAR ファイル内でのファイルの使用	4-25
4.9.5	アイコンおよびイメージの検索パス	4-25
4.9.5.1	DocumentBase	4-26
4.9.5.2	CodeBase	4-26
4.10	言語検出の有効化	4-27
4.10.1	言語検出の指定	4-27
4.10.2	インライン IME サポート	4-27
4.10.3	言語検出機能の仕組み	4-28
4.10.3.1	マルチレベルの継承	4-28
4.11	キー・マッピングの有効化	4-28
4.11.1	fmrweb.res のカスタマイズ	4-29
4.11.1.1	変更の例: 入力マッピングと実行マッピングのスイッチング	4-29
4.11.1.2	例外または特殊なキー・マッピング	4-29
4.11.1.2.1	[F2] のマッピング	4-30
4.11.1.2.2	KEY-ENTER-TRIGGER を起動する [Enter] のマッピング	4-30
4.11.1.2.3	ナンバー・キーのマッピング	4-30
4.11.1.2.4	Web フォームを終了する [Esc] キーのマッピング	4-31

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	5-2
5.3	HTTPD プロセス数の制限	5-2
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	5-5
5.7	サーバー要件	5-5
5.8	クライアント要件 : Oracle JInitiator での HTTPS の使用	5-5
5.9	ユーザー ID/ パスワード非表示機能の使用	5-6
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	5-6
5.11	Oracle Forms Services と SSL	5-7
5.11.1	SSL を使用する Oracle HTTP Server の構成	5-7
5.11.2	SSL を使用する Oracle Web Cache の構成	5-8
5.11.3	SSL を使用するフォームの実行	5-9
5.11.4	ロード・バランシング・ルーターを使用した SSL の構成	5-10

6 Oracle Application Server Single Sign-On での Forms Services の使用

6.1	概要	6-2
6.2	OracleAS Single Sign-On、Oracle Internet Directory、および Forms で使用可能な機能	6-2
6.2.1	Oracle Internet Directory にリソースがない場合のリソースの動的な作成	6-2
6.2.2	Oracle Internet Directory 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	6-3
6.2.3	Forms と OracleAS Single Sign-On での動的ディレクティブのサポート	6-3
6.2.4	OracleAS Single Sign-On 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
6.3	Oracle Forms で使用される OracleAS Single Sign-On コンポーネント	6-4
6.4	アプリケーションでの OracleAS Single Sign-On の有効化	6-4
6.4.1	ssoMode	6-5
6.4.2	ssoDynamicResourceCreate	6-5
6.4.3	ssoErrorURL	6-6
6.4.4	ssoCancelUrl	6-6
6.4.5	Forms から SSO 情報へのアクセス	6-6
6.5	Oracle Forms と Reports の統合	6-7
6.6	認証フロー	6-7

7 JVM プーリング

7.1	概要	7-2
7.2	JVM プーリングの例	7-2
7.3	設計時の考慮事項	7-3
7.3.1	以前のバージョンの Java Importer について	7-3
7.3.2	Java コードの再インポート	7-3
7.3.3	複数の JVM 間での静的変数の共有について	7-4
7.4	JVM コントローラについて	7-4
7.5	JVM プーリングの管理	7-6
7.5.1	Enterprise Manager Application Server Control による JVM コントローラの管理について	7-6
7.5.2	コマンドラインによる JVM コントローラの管理について	7-7

7.5.3	新規 JVM コントローラの作成	7-7
7.5.4	JVM コントローラの削除	7-8
7.5.5	Enterprise Manager Application Server Control による JVM コントローラの プロパティの編集	7-9
7.5.6	デフォルト JVM コントローラのプロパティの指定	7-9
7.5.7	Enterprise Manager Application Server Control による JVM コントローラの 起動と停止	7-10
7.5.7.1	JVM コントローラの起動または再起動	7-10
7.5.8	JVM コントローラの使用コマンド	7-11
7.5.8.1	コマンドの制限	7-11
7.5.8.2	コマンドラインによる JVM コントローラの起動	7-11
7.5.8.3	JVM コントローラの停止	7-12
7.5.9	JVM コントローラの構成ファイル	7-12
7.5.9.1	起動オプションの優先順位	7-13
7.5.10	JVM コントローラのコマンドの例	7-13
7.5.11	Forms 構成ファイル設定	7-14
7.5.12	起動の例	7-15
7.5.13	複数の JVM コントローラについて	7-16
7.5.14	子 JVM について	7-16
7.5.14.1	子 JVM の例	7-16
7.6	JVM コントローラのロギング管理	7-17
7.6.1	ロギングの有効化と無効化	7-17
7.6.1.1	デフォルトのロギング・プロパティの指定	7-17
7.6.1.2	ログ・ファイルのディレクトリの指定	7-18
7.6.1.3	ログ・ファイルへのアクセス	7-18
7.6.1.4	JVM コントローラのログ・ファイルの削除	7-19
7.7	JVM プーリングのエラー・メッセージ	7-19

8 トレースと診断

8.1	Forms Trace について	8-2
8.2	Forms Trace の構成	8-2
8.2.1	URL パラメータ・オプションの指定	8-4
8.3	Forms Trace の開始	8-5
8.4	Forms Trace 出力の表示	8-5
8.4.1	Translate ユーティリティの実行	8-6
8.5	トレース可能イベントのリスト	8-6
8.5.1	イベントの詳細のリスト	8-8
8.5.1.1	ユーザー・アクション・イベント	8-8
8.5.1.2	Forms Services のイベント	8-9
8.5.1.3	イベントの詳細	8-9
8.5.1.4	3層イベント	8-9
8.5.1.5	その他のイベント	8-10
8.6	Forms Services のトレース・メトリックの監視	8-10
8.7	Servlet Logging Tools	8-10
8.7.1	ロギングの有効化	8-11
8.7.1.1	URL でのロギングの指定	8-11
8.7.1.2	Oracle Enterprise Manager でのロギングの指定	8-11
8.7.1.3	Forms Servlet を起動する URL での完全診断の指定	8-12

8.7.2	ログ・ファイルのディレクトリ	8-12
8.7.3	各レベルのサブレット・ロギングの出力例	8-12
8.7.3.1	(なし)	8-12
8.7.3.2	/session	8-12
8.7.3.3	/sessionperf	8-12
8.7.3.4	/perf	8-13
8.7.3.5	/debug	8-13

9 エンド・ユーザー監視の構成

9.1	エンド・ユーザー監視について	9-2
9.2	エンド・ユーザー監視の構成	9-2
9.2.1	エンド・ユーザー監視の使用要件	9-2
9.2.2	エンド・ユーザー監視を使用する Web Cache の構成	9-3
9.2.3	Enterprise Manager Grid Control で監視する Web Cache インスタンスの指定	9-3
9.2.4	デフォルトの最小ヒット数のしきい値の変更	9-4
9.2.5	不適切なレスポンス時間を排除する設定の変更	9-4
9.3	エンド・ユーザー監視の有効化	9-4
9.3.1	formsweb.cfg の変更	9-4
9.4	その他の情報	9-5

10 パフォーマンス・チューニングに関する考慮事項

10.1	Forms Services の組み込み最適化機能	10-2
10.1.1	Forms Services の監視	10-2
10.1.1.1	Forms Services インスタンスの監視	10-2
10.1.1.2	Forms イベントの監視	10-3
10.1.1.3	ユーザー・セッションのメトリックの監視	10-3
10.1.1.4	メトリック情報のソート	10-3
10.1.1.5	検索	10-3
10.1.2	Forms Services Web ランタイム・プーリング	10-3
10.1.2.1	事前起動パラメータの構成	10-4
10.1.2.2	ランタイム・プーリングの起動	10-4
10.1.3	Forms Services のユーティリティ	10-5
10.1.3.1	Forms Services のユーティリティを使用するには:	10-5
10.1.4	クライアント・リソース要件の最小化	10-5
10.1.5	Forms Services リソース要件の最小化	10-5
10.1.6	ネットワーク使用量の最小化	10-6
10.1.7	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	10-6
10.1.8	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	10-6
10.2	OracleAS Forms Services アプリケーションのチューニング	10-7
10.2.1	データ・サーバーに対する Oracle Application Server Forms Services の位置	10-7
10.2.2	アプリケーションの起動時間の最小化	10-8
10.2.2.1	Java ファイルの使用	10-8
10.2.2.1.1	Oracle JInitiator	10-8
10.2.2.1.2	その他のすべての場合 (Sun 社の Java Plug-in など)	10-9
10.2.2.2	キャッシュの使用	10-9
10.2.3	必須ネットワーク帯域幅の削減	10-10
10.2.4	パフォーマンスを改善するためのその他の方法	10-11
10.3	Web Cache と Forms の統合	10-12

11 OracleAS Forms Services へのアップグレード

11.1	OracleAS Forms Services アップグレード項目	11-2
11.2	OracleAS Forms Services に関連するコンポーネント	11-2
11.3	OracleAS Forms Services アップグレード・タスク	11-3
11.3.1	アップグレードに関する推奨事項およびトラブルシューティングのヒント	11-3
11.3.2	OracleAS Forms Services アプリケーション・モジュールのアップグレード	11-4
11.3.3	Common Gateway Interface (CGI) から Oracle Forms Servlet へのアップグレード	11-4
11.3.4	静的 HTML 開始ファイルから汎用アプリケーション HTML 開始ファイルへの アップグレード	11-5
11.3.4.1	OracleAS Forms Services での静的 HTML ファイルの使用	11-6
11.3.5	Forms 6i Listener から Forms Listener Servlet へのアップグレード	11-7
11.3.6	Forms Listener Servlet アーキテクチャから OracleAS Forms Services への アップグレード	11-9
11.3.7	ロード・バランシングのアップグレード	11-10
11.3.8	使用方法	11-10
11.3.8.1	Forms Servlet によるアイコン・イメージの配置	11-10
11.3.8.2	Oracle Reports に対する統合されたコールから Oracle Reports の使用への アップグレード	11-11
11.3.8.3	OC4J での Forms Listener Servlet 別名の作成	11-11
11.3.8.4	Listener Servlet 管理ページへのアクセス	11-12
11.4	OracleAS Forms Services のアップグレードの検証	11-12

A Oracle Forms Services のトラブルシューティング

A.1	インストールの確認	A-2
A.1.1	Web Form Tester の使用	A-2
A.1.2	ポート情報の確認	A-2
A.2	FRM-XXXXX エラーの診断	A-3
A.2.1	Oracle Forms アプレット	A-3
A.2.2	JInitiator Java コンソール	A-3
A.2.3	FRM-92XXX エラー・メッセージ	A-3
A.2.3.1	FRM-92010	A-5
A.2.3.2	FRM-92050	A-5
A.2.3.3	FRM-92100	A-6
A.2.3.4	FRM-92101	A-7
A.2.3.5	FRM-92102	A-7
A.2.3.6	FRM-92120	A-8
A.2.3.7	FRM-92150/FRM-92160	A-8
A.3	スタック・トレースを使用したサーバーのクラッシュの診断	A-8
A.3.1	スタック・トレースについて	A-8
A.3.2	スタック・トレースの構成と使用	A-9
A.3.2.1	環境の確認	A-9
A.3.2.2	Solaris のスタック・トレースについて	A-9
A.3.2.3	Windows のスタック・トレースについて	A-9
A.4	クライアント・クラッシュの診断	A-9
A.4.1	クライアント・クラッシュの診断について	A-9
A.4.2	ハングしているアプリケーションの診断	A-10
A.4.2.1	アプリケーションがハングする原因	A-10
A.4.2.2	Java スレッド・ダンプの作成	A-10

A.5	Forms Trace と Servlet Logging Tools	A-10
A.6	メモリーの問題の解決	A-11
A.6.1	Java のメモリーの使用方法	A-11
A.6.2	初期 Java ヒープの設定	A-11
A.6.3	メモリー・リークについて	A-11
A.6.3.1	Java でのメモリー・リーク	A-12
A.6.3.2	メモリー・リークの特長	A-12
A.6.4	キャッシングによるパフォーマンスの向上	A-12
A.6.5	OJDK でのパフォーマンス向上	A-13
A.7	トラブルシューティングのヒント	A-13
A.8	さらに調査が必要な場合	A-14

B JInitiator

B.1	Oracle JInitiator を使用する理由	B-2
B.2	Oracle JInitiator の利点	B-2
B.3	Oracle JInitiator の使用方法	B-2
B.4	サポートされる構成	B-2
B.4.1	Windows 98、NT、2000 および XP	B-2
B.5	システム要件	B-3
B.6	Netscape Navigator での Oracle JInitiator の使用方法	B-3
B.7	Microsoft Internet Explorer での Oracle JInitiator の使用方法	B-3
B.8	Oracle JInitiator プラグインの設定	B-4
B.8.1	Oracle JInitiator マークアップのベース HTML ファイルへの追加	B-4
B.8.2	Oracle JInitiator ダウンロード・ファイルのカスタマイズ	B-4
B.8.3	Oracle JInitiator をダウンロード可能にする	B-4
B.9	Oracle JInitiator プラグインの変更	B-4
B.9.1	Oracle JInitiator キャッシュ・サイズの変更	B-5
B.9.2	Oracle JInitiator ヒープ・サイズの変更	B-5
B.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	B-5
B.9.4	Oracle JInitiator 出力の表示	B-5
B.10	ベース HTML ファイルの変更	B-6

C サンプル構成ファイル

C.1	デフォルトの formsweb.cfg ファイル	C-2
C.2	プラットフォーム固有の default.env ファイル	C-5
C.2.1	デフォルトの Windows 用 default.env ファイル	C-5
C.2.2	デフォルトの Solaris 用 default.env ファイル	C-7
C.3	base.htm ファイル、basejini.htm ファイル、および basejpi.htm ファイル	C-8
C.3.1	ベース HTML ファイル内のパラメータと変数	C-9
C.3.1.1	使用方法	C-10
C.3.2	デフォルトの base.htm ファイル	C-10
C.3.3	デフォルトの basejini.htm ファイル	C-11
C.3.4	デフォルトの basejpi.htm ファイル	C-12
C.4	web.xml	C-14
C.4.1	デフォルトの web.xml ファイル	C-14
C.5	forms.conf	C-16
C.5.1	デフォルトの forms.conf	C-16
C.6	Registry.dat	C-17

C.6.1	デフォルトの Registry.dat	C-18
C.7	デフォルトの jvmcontroller.cfg	C-19
C.8	デフォルトの webutil.cfg	C-20
C.9	デフォルトの webutilbase.htm	C-21
C.10	デフォルトの webutiljini.htm	C-23
C.11	デフォルトの webutiljpi.htm	C-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。

ドキュメントのアクセシビリティについて

オラクル社は、障害のあるお客様にもオラクル社の製品、サービスおよびサポート・ドキュメントを簡単にご利用いただけることを目標としています。オラクル社のドキュメントには、ユーザーが障害支援技術を使用して情報を利用できる機能が組み込まれています。HTML 形式のドキュメントで用意されており、障害のあるお客様が簡単にアクセスできるようにマークアップされています。標準規格は改善されつつあります。オラクル社はドキュメントをすべてのお客様がご利用できるように、市場をリードする他の技術ベンダーと積極的に連携して技術的な問題に対応しています。オラクル社のアクセシビリティについての詳細情報は、Oracle Accessibility Program の Web サイト <http://www.oracle.com/accessibility/> を参照してください。

ドキュメント内のサンプル・コードのアクセシビリティについて

スクリーン・リーダーは、ドキュメント内のサンプル・コードを正確に読めない場合があります。コード表記規則では閉じ括弧だけを行に記述する必要があります。しかし JAWS は括弧だけの行を読まない場合があります。

外部 Web サイトのドキュメントのアクセシビリティについて

このドキュメントにはオラクル社およびその関連会社が所有または管理しない Web サイトへのリンクが含まれている場合があります。オラクル社およびその関連会社は、それらの Web サイトのアクセシビリティに関しての評価や言及は行っておりません。

関連ドキュメント

詳細は、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 のリリース・ノート
- 『Oracle Forms Forms 6i 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

本文では、次の表記規則を使用します。

表記規則	意味
太字	太字は、操作に関連するグラフィカル・ユーザー・インタフェース要素、また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。
イタリック	イタリックは、特定の値を指定するプレースホルダ変数を示します。
固定幅フォント	固定幅フォントは、パラグラフ内のコマンド、URL、例に記載されているコード、画面に表示されるテキスト、または入力するテキストを示します。

サポートおよびサービス

次の各項に、各サービスに接続するための URL を記載します。

オラクル社カスタマ・サポート・センター

オラクル製品サポートの購入方法、およびオラクル社カスタマ・サポート・センターへの連絡方法の詳細は、次の URL を参照してください。

<http://www.oracle.co.jp/support/>

製品マニュアル

製品のマニュアルは、次の URL にあります。

<http://otn.oracle.co.jp/document/>

研修およびトレーニング

研修に関する情報とスケジュールは、次の URL で入手できます。

<http://www.oracle.co.jp/education/>

その他の情報

オラクル製品やサービスに関するその他の情報については、次の URL から参照してください。

<http://www.oracle.co.jp>

<http://otn.oracle.co.jp>

注意： ドキュメント内に記載されている URL や参照ドキュメントには、Oracle Corporation が提供する英語の情報も含まれています。日本語版の情報については、前述の URL を参照してください。

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。

この章には、次の項が含まれています。

- 第 1.1 項 「Oracle インターネット・プラットフォーム」
- 第 1.2 項 「Oracle Application Server Forms Services」
- 第 1.3 項 「OracleAS Forms Services のアーキテクチャ」
- 第 1.4 項 「OracleAS Forms Services のコンポーネント」
- 第 1.5 項 「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 10g でデータを管理し、Oracle Developer Suite 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応できます。

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

- 第 1.1.1 項「Oracle Application Server」
- 第 1.1.2 項「Oracle Developer Suite」
- 第 1.1.3 項「Oracle Database 10g」

1.1.1 Oracle Application Server

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

Oracle Developer Suite は、Oracle Application Development Tools、Oracle Business Intelligence Tools、Oracle XML Developer's 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。

Oracle Developer Suite 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 10g 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 10g

Oracle Database 10g は、世界で最も一般的な RDBMS の新世代の製品です。Oracle Real Application Clusters 10g 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 10g 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 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 サープ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 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。

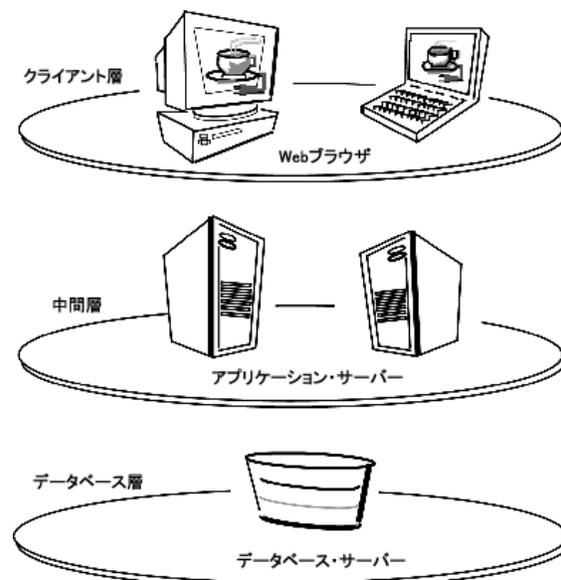
- ランタイム・プーリング (第 10.1.2 項「Forms Services Web ランタイム・プーリング」を参照)
- Oracle Enterprise Manager の向上した Web インタフェース (第 6.4 項「アプリケーションでの OracleAS Single Sign-On の有効化」を参照)
- OracleAS Single Sign-On の向上 (第 6 章「Oracle Application Server Single Sign-On での Forms Services の使用」を参照)
- 運用管理を容易にする Oracle Enterprise Manager との統合の向上 (第 8 章「トレースと診断」および第 9 章「エンド・ユーザー監視の構成」を参照)
- トレースとロギングの改善 (第 10 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)
- Java 仮想マシン (JVM) プーリング (第 7 章「JVM プーリング」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



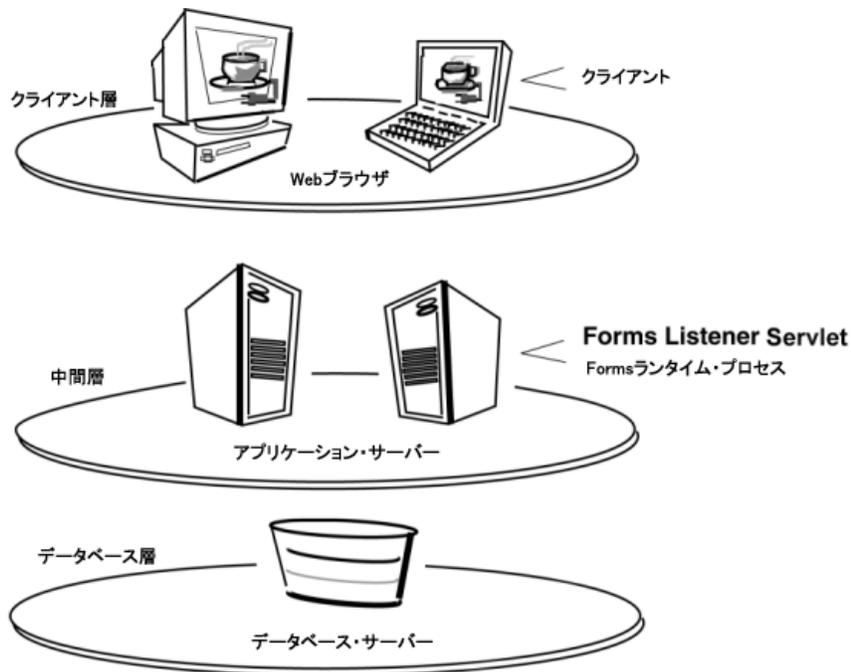
1.4 OracleAS Forms Services のコンポーネント

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・プロセスに代わって 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・モードでの実行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。

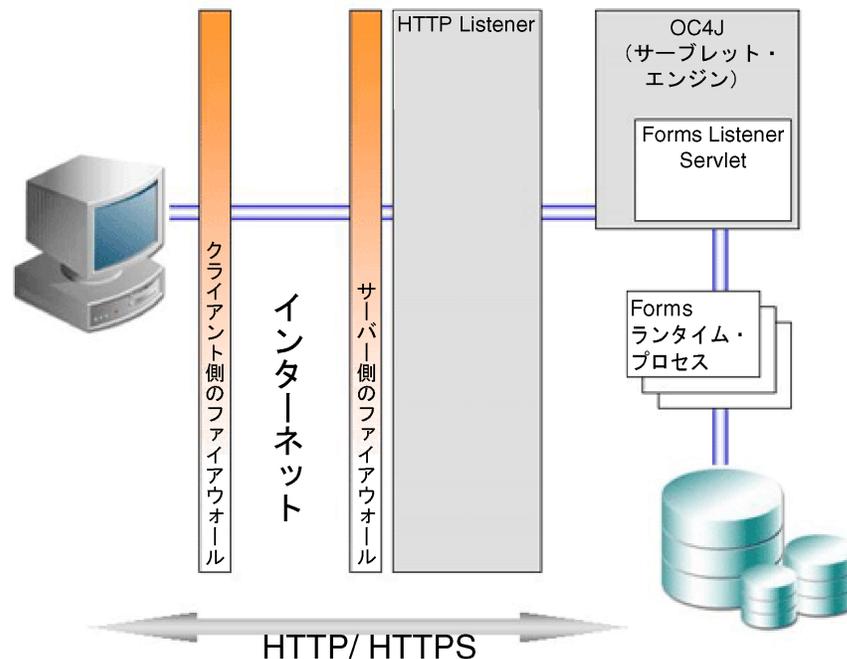
1.5 Forms Listener Servlet

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 に代わるものです。

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。

- [第 2.1 項「OracleAS Forms Services のセキュリティについて」](#)
- [第 2.2 項「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 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 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。OracleAS Single Sign-On 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 OracleAS Single Sign-On 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 Oracle Internet Directory 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 Oracle Internet Directory 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 OracleAS Single Sign-On 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。OracleAS Single Sign-On 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 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。Oracle Delegated Administration Services は、OracleAS Single Sign-On 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 Oracle Internet Directory のセルフサービス・データの管理に使用します。

Oracle Internet Directory 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、Oracle Delegated Administration Services で作成可能な RAD のエントリを使用することです。デフォルトの RAD エントリには、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 OracleAS Single Sign-On 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 OracleAS Single Sign-On 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、OracleAS Single Sign-On で保護された Forms アプリケーションと OracleAS Single Sign-On で保護されない 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、`forms/server/formsweb.cfg` 構成ファイルのアプリケーション定義にある OracleAS Single Sign-On パラメータを使用して有効化します。

2.1.3.2 Oracle Internet Directory における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、Oracle Internet Directory 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 Oracle Internet Directory 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 OracleAS Single Sign-On サポートの認証フローの詳細は、[第 6.6 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 OracleAS Single Sign-On 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 Oracle Internet Directory で処理されます。OracleAS Single Sign-On および Oracle Internet Directory の構成の詳細は、[第 6 章「Oracle Application Server Single Sign-On での Forms Services の使用」](#)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 10g Application Server Control コンソールで行われます。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「Forms Services の構成」](#)および[第 6 章「Oracle Application Server Single Sign-On での Forms Services の使用」](#)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

Oracle Internet Directory でリソースを動的に作成したり、Oracle Internet Directory 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 での Forms Services の使用」](#)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「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 での Forms Services の使用」](#)
- [第8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。

- [第 3.1 項「OracleAS Forms Services のアクション」](#)
- [第 3.2 項「構成ファイル」](#)
- [第 3.3 項「アプリケーションの配布」](#)
- [第 3.4 項「クライアント・ブラウザのサポート」](#)

3.1 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 `mycomputer.com` と呼ばれるコンピュータのポート `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://mycomputer.com:7777/forms/frmservlet?config=myapp&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 `myapp` 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 `/forms/frmservlet` が `forms.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 `/forms`) にマップします。次に、(`web.xml` ファイルで指定した `frmservlet` 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。
 - サー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/forms/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=myapp` が含まれているため、`[myapp]` 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ったら、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コンピュータ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 `/forms/jinitiator` (`forms.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://mycomputer.com:7777/forms/java/frmall_jinit.jar` からロードされます。

/forms/java の formsweb.cfg 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 (frmall_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://mycomputer.com:7777/forms/lservlet` 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 /forms/lservlet が forms.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 (lservlet) は、Forms セッションに Forms ランタイム・プロセス (frmweb.exe または frmweb) 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 を介して続きます。
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 コンソールの「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 FORMS_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。

3.2 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「Forms Services の構成」を参照してください。

この項は、次の項目に分かれています。

- 第 3.2.1 項「Oracle Forms 構成ファイル」
- 第 3.2.2 項「Oracle Application Server Containers for J2EE (OC4J) 構成ファイル」
- 第 3.2.3 項「Oracle HTTP Listener の構成ファイル」
- 第 3.2.4 項「標準フォントとアイコン・ファイル」
- 第 3.2.5 項「WebUtil 構成ファイル」

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。

3.2.1 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 コンソールで管理します。これらの構成ファイルには次のものが含まれます。

- [default.env](#)
- [formsweb.cfg](#)
- [base.htm](#)、[basejini.htm](#)、および [basejpi.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。

3.2.1.1 default.env

場所: forms/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/forms/server ディレクトリに格納されています。Solaris では、default.env に PATH と LD_LIBRARY_PATH を挿入する必要があります。

3.2.1.2 formsweb.cfg

場所: forms/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 コンソールで 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 コンソールによるパラメータの構成」](#)を参照してください。

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

場所: forms/server

ベース HTML ファイル (base.htm、basejini.htm および basejpi.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 C.3 項「base.htm ファイル、basejini.htm ファイル、および basejpi.htm ファイル」](#) を参照してください。

3.2.1.4 ftrace.cfg

場所: forms/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) の機能に代わるものです。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 コンソールで Forms Trace を管理します。

ftrace.cfg の詳細は、[第 8 章「トレースと診断」](#) を参照してください。

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

デフォルトでの Forms Services は、formsapp.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](#)

3.2.2.1 web.xml

場所: j2ee/OC4J_BI_FORMS/applications/formsapp/formsweb/WEB-INF/web.xml

Once 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 のディレクトリ j2ee/OC4J_BI_FORMS/applications/formsapp/formsweb/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 frmservlet と lservlet が定義されます。

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

3.2.2.2 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 (formsapp.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/formsapp
+META-INF
-application.xml (defines the structure of the ear file)
+formsweb
+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
  -frmsrv.jar (contains the Forms Servlet and Listener Servlet code)
```

3.2.3 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。

3.2.3.1 forms.conf

場所: forms/server

これは、Oracle Application Server Forms Services 用の Oracle HTTP Listener の構成ファイルです。このファイルは 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。forms.conf では、仮想ディレクトリ (別名) とサブレット・マウント・ポイントを定義して、URL リクエストを OC4J サブレット・エンジンで動作している Forms Servlet にマップします。

forms.conf の詳細は、[第 C.5 項「forms.conf」](#) を参照してください。

3.2.4 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。

3.2.4.1 Registry.dat

場所: forms/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 C.6 項「Registry.dat」](#) を参照してください。

3.2.5 WebUtil 構成ファイル

この項では、ランタイムに WebUtil 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。設計時の WebUtil の使用については、[Oracle Forms オンライン・ヘルプ](#) を参照してください。

これらのファイルは、Oracle Developer Suite の Oracle_Home/forms/server ディレクトリでのみ利用できます。これらのファイルは、Oracle Application Server、Oracle Application Server Forms Services および Reports Services では利用できません。

- [デフォルトの webutil.cfg](#)
- [デフォルトの webutilbase.htm](#)
- [デフォルトの webutiljini.htm](#)
- [デフォルトの webutiljpi.htm](#)

3.2.5.1 デフォルトの webutil.cfg

場所: forms/server

このファイルには、次のような WebUtil の構成設定がすべて示されます。

- [ログイン・オプション](#)
- [インストール・オプション](#)
- [ファイルのアップロードおよびダウンロードのオプション](#)
- [ログイン・エラーやログ・メッセージに対するサーバー・サイドのログイン・オプション](#)

詳細は、[第 C.8 項「デフォルトの webutil.cfg」](#) を参照してください。

3.2.5.2 デフォルトの webutilbase.htm

場所: forms/server

これは、Web でフォームを実行するデフォルトのベース HTML ファイルです。汎用の APPLET タグを使用して、WebUtil に証明書を登録する Forms アプレットを含めます。

詳細は、[第 C.9 項「デフォルトの webutilbase.htm」](#) を参照してください。

3.2.5.3 デフォルトの webutiljini.htm

場所: forms/server

これは、Web でフォームを実行する HTML のテンプレート・ファイルです。JInitiator 型のタグを使用して、WebUtil に証明書を登録する Forms アプレットを含めます。

詳細は、[第 C.10 項「デフォルトの webutiljini.htm」](#) を参照してください。

3.2.5.4 デフォルトの webutiljpi.htm

場所: forms/server

これは JDK Java Plug-in を使用して、Web のフォームを実行するデフォルトのベース HTML ファイルです。これは、たとえば、Solaris 上で Netscape を使って Web のフォームを実行する際、および WebUtil の証明書の登録を実行する際に使用されます。

詳細は、[第 C.11 項「デフォルトの webutiljpi.htm」](#) を参照してください。

3.3 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、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 frmweb.exe、Solaris では rmweb.sh）を起動します。

Forms Services の実行方法の詳細は、[第 3.1 項「OracleAS Forms Services のアクション」](#) を参照してください。

3.3.1 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。

2. 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 10g Application Server Control コンソールの「Forms Web 構成」ページに構成セクションを作成します。

[表 3-1](#) 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。

- .fmx ファイルの場所は、FORMS_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、FORMS_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、表 3-2 に示すとおり、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。

- 次の URL に、アプリケーションの名前を入力します。

```
http://mycomputer.com:7777/forms/frmservlet?
```

mycomputer はコンピュータの名前、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://mycomputer.com:7777/forms/frmservlet?config=
application
```

3.3.2 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 コンソールの「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します。

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（第 3.3.3 項「Oracle Enterprise Manager での構成セクションの作成」を参照）。たとえば、myApp に対して作成する構成セクションで、表 3-3 に示す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：myApp 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
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。

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

第 3.3.1 項「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、第 4.3.4 項「デフォルトの Forms 構成パラメータ」を参照してください。

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

3.3.3.1 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://mycomputer.com:7777/forms/frmservlet?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://mycomputer.com:7777/forms/frmservlet?config=myapp&form=hrapp
```

これは、`myapp` 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。

3.3.4 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。

3.3.4.1 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** コンソール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比的に、`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「ab」および「cd」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 (次に示すとおり、`%` 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、前述のとおりです)。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)。

3.3.4.2 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。

3.3.4.3 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。

3.3.4.4 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）。

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注意してください。

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関係なく、前リリースの動作が常に取得されます。

3.3.4.5 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（大文字と小文字の区別なし）。

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。ここでは、値 foo'bar 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">
```

3.4 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。

オラクル社からは、2つの JAR ファイル (frmall.jar および frmall_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。frmall_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 でのみ使用できません。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 B 「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. [表 3-5](#) に示すように適切なベース 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*	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)

4. 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。
5. HTML ファイルをユーザーのブラウザに送信します。

Forms Services の構成

この章には、次の項が含まれています。

- 第 4.1 項 「Oracle Application Server Forms Services での Forms アプリケーションの起動方法」
- 第 4.2 項 「Oracle Enterprise Manager と Oracle Forms」
- 第 4.3 項 「Forms Services の構成」
- 第 4.4 項 「Oracle Enterprise Manager による環境変数の構成」
- 第 4.5 項 「ユーザー・セッションの管理」
- 第 4.6 項 「アプリケーションの URL セキュリティの管理」
- 第 4.7 項 「ユーザー独自のテンプレート HTML ファイルの作成」
- 第 4.8 項 「Oracle Forms アプリケーションへの Graphics の挿入」
- 第 4.9 項 「Forms Services で使用するアイコンとイメージの配置」
- 第 4.10 項 「言語検出の有効化」
- 第 4.11 項 「キー・マッピングの有効化」

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 10g Application Server Control コンソール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 コンソールのユーザー・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://<computer.domain>:1156`

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

Forms Services では、Web ベースの Oracle Enterprise Manager Application Server Control コンソールを使用して、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 10.1.1.1 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 10.1.1.3 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 コンソール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 8.2 項「Forms Trace の構成」](#)および[第 8.6 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 10.1.3 項「Forms Services のユーティリティ」](#)および[第 10.2 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 Enterprise Manager Application Server Control の使用方法

デフォルトでは、Enterprise Manager Application Server Control 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。

```
em_mode=1
```

これで、Application Server Control 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 に設定した後に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 で、Application Server Control を使用してこの変数を設定できます。

```
allow_debug=true
```

この変数によって、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、Application Server Control 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 コンソール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 コンソールによる Forms Services へのアクセス

Application Server Control コンソールを使用して 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 コンソールで Forms Server の「Forms」ホームページにナビゲートします。

Application Server Control コンソールで Forms Server の「Forms」ホームページにナビゲートするには：

1. Application Server Control コンソールを使用して、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 コンソールの使用方法の予備知識は、『Oracle Application Server 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 コンソールに表示されます。

4.3 Forms Services の構成

Forms Services を構成するには、Application Server Control コンソールの「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 10g 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 コンソールを起動します。
2. Application Server Control コンソールのメイン・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 コンソールで 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 コンソール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 3.2.1.2 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 baseHTML 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。表 4-1「ファイル指定するデフォルトの構成パラメータ」で示されるとおり、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
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/forms/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 コンソールの「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 コンソールを起動します。
2. Application Server Control コンソールのメイン・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。

- Forms Services インスタンスで、「構成」タブを選択します。
- 「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。
- 新しい構成の名前を入力し、「OK」をクリックします。
- 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 「新しい構成セクションに追加するサンプル・パラメータ」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/forms/frmservlet?config=sepwin
```

特別な構成のその他の例は、第 C.1 項「デフォルトの `formsweb.cfg` 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：

- 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。
- 「複製」をクリックします。
- 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、第 4.3.3 項「パラメータの管理」を参照してください。

名前を付けた構成を削除するには：

- Oracle Enterprise Manager Application Server Control コンソールを起動します。
- Application Server Control コンソールのメイン・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。
- 「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。
- 「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 コンソールは「構成」タブに戻り、残りの構成を表示します。

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 コンソールを使用します。Application Server Control コンソールの「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. Enterprise Manager Application Server Control コンソールの「構成」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 コンソールの「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

この項は、次の項目に分かれています。

- 第 4.3.4.1 項「システムのデフォルトの構成パラメータ」
- 第 4.3.4.2 項「ランフォーム・パラメータ (serverArgs パラメータ)」
- 第 4.3.4.3 項「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」
- 第 4.3.4.4 項「アプレット・パラメータまたはオブジェクト・パラメータ」
- 第 4.3.4.5 項「JInitiator のパラメータ」
- 第 4.3.4.6 項「Sun 社の Java Plug-in のパラメータ」
- 第 4.3.4.7 項「Oracle Enterprise Manager の構成パラメータ」
- 第 4.3.4.8 項「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。これらのパラメータは表 4-3 「システムのデフォルトの構成パラメータ」で説明します。

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	パラメータ値と説明
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。
HTMLdelimiter	必須	変数名のデリミタ。デフォルトで % になります。
workingDirectory	必須	設定されていない場合のデフォルトは ORACLE_HOME/forms です。
envFile	必須	デフォルトでは default.env に設定されます。

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	パラメータ値と説明
defaultcharset	任意	<p>サ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>注意: 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せん。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	<p>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。</p>
log	任意	<p>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。</p> <p>デフォルト値は NULL です。</p>
jvmcontroller	任意	<p>有効値の詳細は、第 7.5.8.2 項「コマンドラインによる JVM コントローラの起動」を参照してください。また、JVM の指定は省略できます。</p> <p>デフォルト値はありません。</p> <p>このパラメータはデフォルト・セクションでグローバルに設定することも、アプリケーション・セクションを優先することもできます。これによって、Forms ランタイム・プロセスで使用する JVM コントローラが決まります。dejvm 実行可能ファイルの jvmcontroller パラメータに対応します。</p> <p>jvmcontroller に値が指定されていない場合は、Oracle Forms ランタイム・プロセスは自身のインプロセス JVM を起動し、Java Importer は 10g より前のリリースの動作をします。</p>

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 3.3.4 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。これらのランフォーム・パラメータの説明は、[表 4-4「ランフォーム・パラメータ \(serverArgs パラメータ\)」](#)にあります。

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	パラメータ値と説明
clientDPI	任意	インチ当たりドット数 (DPI) を指定し、JVM で返される DPI 設定を上書きします。これにより、プラットフォームごとに異なる DPI 設定を管理できます。たとえば、Win32 プラットフォームで開発されるフォームは、DPI 値が異なるため、UNIX プラットフォームでは正しく表示されないことがあります。clientDPI 値には、任意の正の整数を指定できます。<param name="clientDPI" value="200"> など、50 から 200 までの整数から値を選択することをお勧めします。
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 false 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 FORMS_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 第 3.3.4 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)。 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。
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。

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	パラメータ値と説明
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、forms/frmservlet/admin 画面の管理機能がすべてアクティブになります。forms/frmservlet/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。 デフォルトの値は、false です。allow_debug=true の場合のみ、トレースが表示可能であることを伝えるインライン・メッセージが表示されます。
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。
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

表 4-5 「HTML ページ・パラメータ」の説明のように、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	パラメータ値と説明
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

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」にある次のパラメータは、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。例: <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	パラメータ値と説明
serverURL	必須	/forms/lervlet (第 1.5 項「Forms Listener Servlet」を参照。)
codebase	必須	物理ディレクトリ ORACLE_HOME/forms/java を示すために定義する仮想ディレクトリ。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 /forms/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"> 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)。 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。
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 で使用するの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。

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	パラメータ値と説明
height	必須	フォーム・アプレットの高さをピクセルで指定します。デフォルトは 500 です。
separateFrame	任意	アプレットを分割ウィンドウ内に表示するかどうかを指定します。有効な値: True または False。
splashScreen	任意	アプレットが表示される前に表示する .GIF ファイルを指定します。スプラッシュなしの場合は NO に設定します。デフォルトのスプラッシュ・イメージを使用する場合は空白のままにします。 パラメータを設定するには、ファイル名 (myfile.gif など) または仮想パスとファイル名 (images/myfile.gif など) を記述します。
background	任意	背景に表示する .GIF ファイルを指定します。背景なしの場合は NO に設定します。デフォルトの背景を使用する場合は空白のままにします。
lookAndFeel	任意	アプリケーションのルック・アンド・フィールを指定します。有効な値: Oracle または Generic (Windows のルック・アンド・フィール)。
colorScheme	任意	アプリケーションの配色を指定します。有効な値: Teal、Titanium、Red、Khaki、Blue、Olive または Purple。 注意: lookAndFeel が Generic に設定されている場合、colorScheme は無視されます。
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/forms/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 frmall.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 frmall_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。
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 frmall.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」の説明のように、次のパラメータは JInitiator に固有です。

表 4-7 JInitiator のパラメータ

パラメータ	必須 / 任意	パラメータ値と説明
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、 /forms/jinitiator/us/JInitiator/jinit.download.htm です。
jinit_classid	必須 (IE のみ)	デフォルトは clsid:CAFECAFE-0013-0001-0017-ABCDEFABCDEF です。
jinit_exename	必須	デフォルトは jinit.exe#Version=1.3.1.17 です。
jinit_mimetype	必須 (Netscape のみ)	デフォルトは application/x-jinit-applet;version=1.3.1.17 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」の説明のように、次のパラメータは Sun 社の Java Plug-in で使用されます。

表 4-8 Sun 社の Java Plug-in のパラメータ

パラメータ	必須 / 任意	パラメータ値と説明
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」の説明のように、次のパラメータは Enterprise Manager の構成に使用されます。

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

パラメータ	必須 / 任意	パラメータ値と説明
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。
EndUserMonitoringEnabled	任意	エンド・ユーザー監視の統合が有効かどうかを示します。
EndUserMonitoringURL	任意	エンド・ユーザー監視のデータを保存する場所を示します。

4.3.4.8 Oracle Internet Directory の構成パラメータ

表 4-10 「Oracle Internet Directory の構成パラメータ」の説明のように、次のパラメータは Oracle Internet Directory の構成に使用されます。OracleAS Infrastructure インスタンスを使用している場合は、これらのパラメータのみ構成できます。

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

パラメータ	必須 / 任意	パラメータ値と説明
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 コンソール」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 `frmweb.exe`、Solaris では `frmweb`）用の環境変数（`PATH`、`ORACLE_HOME`、`FORMS_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/forms/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、サーブレット・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、サーブレット・エンジン（OC4J）である親プロセスの環境から継承されます。

注意： 環境ファイルは、Oracle Enterprise Manager Application Server Control コンソール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/forms/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 Application Server Control から削除できません。Application Server Control コンソール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。
FORM_PATH	ORACLE_HOME¥forms (デフォルト)	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 Oracle Forms で検索するパスを指定します。 Windows の場合、複数のパスはセミコロン (;) で区切ります。 Solaris の場合、複数のパスはコロン (:) で区切ります。
FORMS_TIMEOUT	デフォルト :15 有効値 :3 ~ 1440 (1 日) 例: FORMS_TIMEOUT=1440	このパラメータは、クライアントと Forms Services 間の通信がないときに Forms Services プロセスが終了するまでの経過時間 (分) を指定します。 クライアントの通信は、ユーザーがなん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、Forms クライアントのハートビートによって生じます。
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。
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。
REPORTS_SERVERMAP	cluster:repserver 各変数の意味は次のとおりです。 cluster は Reports Server のクラスタ名です。 repserver は Reports Server の名前です。	この設定は、Forms アプリケーションが Reports サーバーではなく、Reports クラスタから Reports アプリケーションをコールしている場合に必要です。この設定は、Forms アプリケーションが web.show_document を使って、Reports アプリケーションをコールするときにも必要です。Reports 構成の詳細は、『Oracle Application Server Reports Services レポート Web 公開ガイド』を参照してください。
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。

表 4-11 デフォルトの環境変数 (続き)

環境変数	有効値	説明
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>ORACLE_HOME/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=ORACLE_HOME/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 ORACLE_HOME/lib:\${LD_LIBRARY_PATH}</pre>	<p>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。</p>

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かぎり、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。

- 第 4.5.1 項「新規ユーザー・セッションの許可」
- 第 4.5.2 項「新規ユーザー・セッションの禁止」
- 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」

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 セッションを使用不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 の「Oracle Application Server Forms Services 概要」ページで、「有効化」(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 の「Oracle Application Server Forms Services 概要」ページで、「無効化」をクリックします。
- 信号が黄色になります。

「無効化」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、「無効化」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せん。

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 10g Application Server Control コンソールを起動します。
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. 「**確認**」ページが表示されます。
7. 「**はい**」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 コンソールの「**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見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 「restrictedURLparams パラメータの定義」 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 restrictedURLparams パラメータの定義

<input type="radio"/>	colorScheme	teal	
<input type="radio"/>	logo		Forms applet parameter
<input checked="" type="radio"/>	restrictedURLparams	userid.debug	Forms applet parameter

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。

4.6.1 Oracle Forms テスト・フォームの保護

実行するアプリケーションを指定しないで Oracle Forms URL にアクセスすると、テスト・フォームが実行されます。たとえば、通常は次の構文を使って Oracle Forms アプリケーションをコールします。

```
http://<host>:<port>/forms/frmservlet?config=myApp
```

Forms Servlet は、formsweb.cfg ファイルから [myApp] を探し出し、これを起動します。アプリケーションが指定されないときは、次のようになります。

```
http://<host>:<port>/forms/frmservlet
```

Forms Servlet は、formsweb.cfg ファイルのデフォルトのセクションの設定を使用します。これらの設定は、Forms 構成ファイルの [default] にあります（アプリケーションがこれらの設定を上書きしない場合は、デフォルトが使用されます）。デフォルトのセクションには、次の設定が含まれます。

```
form=test.fmx
```

これは、Oracle Forms Services のインストールと構成をテストできるテスト・フォームです。このため、アプリケーションを指定しない場合は、Forms は test.fmx ファイルを起動します。これを次のように変更できます。

```
form=
```

このようにすると、フォームは実行されません。ただし、これは最適ではありません。Forms Servlet は動的に生成された HTML ファイルをクライアントに送信するため、ユーザーが情報を勝手に取得することができます。保護を最適にするソリューションは、クライアントに表示される情報 HTML ページにリクエストをリダイレクトすることです。このためには、formsweb.cfg ファイルのパラメータの一部を変更する必要があります。

Oracle Forms Services のインストール時に変更するパラメータを、デフォルトの値とともに次に示します。

```
# 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
```

これらのパラメータは、クライアントに送信される HTML 情報のテンプレートです。情報 HTML ページを作成して、これらの変数をかわりにポイントします。たとえば、ORACLE_HOME/forms/server ディレクトリに、次のコンテンツを含む forbidden.html という名前の簡単な HTML ページを作成します。

```
<html>
  <head>
    <title>Forbidden</title>
  </head>
  <body>
    <h1>Forbidden!</h1>
    <h2>You may not access this Forms application.</h2>
  </body>
</html>
```

注意： クライアント情報のリダイレクトおよびメッセージ・ページの代替表示は、リクエストしたコンテンツにアクセス権の制限がある場合に Web サーバーが返す Web ページとは異なります。

次に、元のパラメータをコメント・アウトまたは変更して、formsweb.cfg パラメータを変更します。

```
# System parameter: default base HTML file
#baseHTML=base.htm
baseHTML=forbidden.html
# System parameter: base HTML file for use with JInitiator client
#baseHTMLjinitiator=basejini.htm
baseHTMLjinitiator=forbidden.html
# System parameter: base HTML file for use with Sun's Java Plug-In
#baseHTMLjpi=basejpi.htm
baseHTMLjpi=forbidden.html
# System parameter: base HTML file for use with Microsoft Internet Explorer
# (when using the native JVM)
#baseHTMLie=baseie.htm
baseHTMLie=forbidden.html
```

ユーザーが次の URL を入力すると、カスタマイズされた Web ページが表示されます。

http://<host>:<port>/forms/frmservlet

formsweb.cfg ファイルで該当するパラメータを変更すれば、forbidden.html のコンテンツ、ファイル名、および場所などもカスタマイズできます。この情報 Web ページには、警告、エラー、タイムスタンプ、IP ロギングまたは問合せ先などの情報を、サーバー構成への影響を最小限にして格納できます。

注意： formsweb.cfg のデフォルト・セクションの baseHTML テンプレートのエントリを上書きするには、アプリケーション固有の名前が付いた構成に元の値（または他の有効な HTML ファイル）を示す同じエントリを追加する必要があります。

```
[myApp]
form=myApplication.fmx
lookandfeel=oracle
baseHTML=base.htm
baseHTMLjinitiator=basejini.htm
baseHTMLjpi=basejpi.htm
baseHTMLie=baseie.htm
```

これらの baseHTML の値を指定しないでユーザーがアプリケーションを実行する場合は、アプリケーション固有の構成セクションがデフォルトの値を上書きしていないので、forbidden.html ページが表示されます。

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 3.3.4 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://www.oracle.com/technology/products/forms/>)。

4.8.1 Oracle Graphics 6i および Oracle Database 9.0.1.4.0 (64 ビット)

RSF 8.0.6 の制限により、Windows の Oracle Graphics 6i は、64 ビットのデータベースに接続できません。このため、Oracle Forms 10g (9.0.4) 以降を使用して 64 ビットのデータベースに接続中で、Oracle Graphics と統合する必要がある場合は、Oracle 6i Home (Graphics がインストールされている場所) をアップグレードして、Bug 3088708 を修正する RSF バージョンを含める必要があります。この RSF の可用性については、オラクル社カスタマ・サポート・センターに問い合せてください。

4.8.2 Reports Server による Graphics 6i の構成

次の手順を実行して、Forms/Reports/Graphics の統合に Reports/Graphics を正しく設定します。

1. graphicsrun.sh スクリプトに次を入力します。

```
ORACLE_GRAPHICS6I_HOME=<location forms6i>
export ORACLE_GRAPHICS6I_HOME
TK_PRINTER=<real printer>
```

2. reports.sh スクリプトに次を入力します。

```
ORACLE_GRAPHICS6I_HOME=<location forms6i>; export ORACLE_GRAPHICS6I_HOME
REPORTS_DEFAULT_DISPLAY=NO; export REPORTS_DEFAULT_DISPLAY
DISPLAY=<computer name>:0.0; export DISPLAY
```

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、Registry.dat のアイコンとイメージのデフォルトのディレクトリおよび検索パスの指定方法を説明します。

4.9.1 Application Server Control を使った Registry.dat の管理

Application Server Control を使って、Registry.dat のパラメータを変更、追加または削除します。

Registry.dat パラメータの値を変更するには：

1. Enterprise Manager の「構成」ページを選択します。
2. 「表示」ドロップダウン・リストから「Forms のフォントおよびアイコンのマッピング (Registry.dat)」を選択します。
3. 「値」テキスト・フィールドでパラメータの横のラジオ・ボタンを選択し、値を変更します。
4. 「適用」をクリックします。
変更内容が保存されます。

Registry.dat パラメータとその値を追加するには：

1. Enterprise Manager の「構成」ページを選択します。
2. 「表示」ドロップダウン・リストから「Forms のフォントおよびアイコンのマッピング (Registry.dat)」を選択します。
3. 「Registry.dat」ページの最後で、「名前」テキスト・フィールドにパラメータの名前を入力します。
4. 「値」テキスト・フィールドに新しいパラメータの値を入力します。
5. 「新規パラメータの追加」をクリックします。
変更内容が保存されます。

Registry.dat パラメータとその値を削除するには：

1. Enterprise Manager の「構成」ページを選択します。
2. 「表示」ドロップダウン・リストから、「Forms のフォントおよびアイコンのマッピング (Registry.dat)」を選択します。
3. 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。
4. 「確認」ページが表示されます。「はい」をクリックします。
5. パラメータが削除され、「構成」ページが再び表示されます。

4.9.2 アプリケーション・アイコンの配置

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、forms ディレクトリが DocumentBase になります。

たとえば、アプリケーションでボタンのアイコンのディレクトリを myapp/<iconname> に定義している場合、そのアイコンはディレクトリ forms/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 コンソールの「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、`forms/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 `forms/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。

4.9.2.1 アイコンの Java アーカイブ・ファイルへの格納

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薦めします。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・ファイルにパッケージされます。

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 `forms/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=frmall_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、`frmall.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。

4.9.2.2 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/forms/java/oracle/forms/registry` ディレクトリにある `Registry.dat` テキスト・ファイルを別のディレクトリにコピーします。このディレクトリは、Web サーバーの仮想ディレクトリ（`/appfile` など）へマップする必要があります。
2. 新規ファイルをリネームします（`myapp.dat` など）。
3. アイコンのディレクトリを指定する `iconpath` パラメータを次のとおりに変更します。

```
default.icons.iconpath=/mydir or http://myhost.com/mydir
(絶対パスの場合)
```

または

```
default.icons.iconpath=mydir
```

(DocumentBase ディレクトリから始まる相対パスの場合)

4. `iconextension` パラメータを次のように変更します。

```
default.icons.iconextension=gif
```

または

```
default.icons.iconextension=jpg
```

アプリケーション・ファイルを参照するには：

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

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」は、アプリケーション・アイコンを配置する正しい場所を示したものです。

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 forms ディレクトリか forms の相対ディレクトリに格納します。
Java アーカイブ	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。

4.9.3 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。

4.9.4 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。

4.9.4.1 イメージの 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://java.sun.com/> を参照してください。

次に例を示します。

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

このコマンドにより、myico.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。

4.9.4.2 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=firmall.jar, icons.jar
imageBase=codebase
```

4.9.5 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)
- Application Server Control 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* はコンピュータ名
- *DocumentBase* は HTML ファイルを指す URL
- *CodeBase* は、(formsweb.cfg ファイルまたは HTML ファイルで指定した) 開始クラス・ファイルのディレクトリを指す URL
- *mydir* は、アイコンまたはイメージのディレクトリを指す URL

4.9.5.1 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

指定ディレクトリ	Forms Services で使用する検索パス
file.gif (たとえば、formsweb.cfg で splashscreen=file.gif として指定)	http://host/documentbase/file.gif
mydir/file.gif	http://host/documentbase/mydir/file.gif (相対パス)
/mydir/file.gif	http://host/mydir/file.gif (絶対パス)

4.9.5.2 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 (表 4-15) およびイメージ (表 4-16) 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。Oracle Forms Services には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 インライン IME サポート

キーを一度打つと挿入カーソルのわきにその文字が直接表示されること（いわゆるインラインまたはオンザスポット入力）が（アジア言語の文字のように）不可能な場合であっても、インライン IME サポートにより、Forms Web アプリケーションでは作成するテキストを正しく表示できます。この機能はデフォルトで有効になっています。これを無効にするには、ベース HTML ファイルのアプリレット・パラメータ inlineIME を false に設定します。

```
<HTML>
<!-- FILE: basejini.htm (Oracle Forms) -->
<BODY>
...
<OBJECT classid=...
>
<PARAM NAME="inlineIME" VALUE="false">
<EMBED SRC="" ...
inlineIME="false"
```

```
>
...
.
</BODY>
</HTML>
.
```

インライン IME サポートを利用するには、クライアントに Jinitiator 1.3.1 または Plug-in 1.4.1 以上が必要です。

baseHTML の使用の詳細は、第 C.3 項「[base.htm ファイル](#)、[basejini.htm ファイル](#)、および [basejpi.htm ファイル](#)」を参照してください。

4.10.3 言語検出機能の仕組み

Forms Servlet が特定の構成 (<http://myserv/servlet/frmservlet?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/frmservlet> 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。

4.10.3.1 マルチレベルの継承

基本構成のあらゆる言語固有の変数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 `FORMS_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。

4.11 キー・マッピングの有効化

キー・バインディングは、アプリケーション機能にキーを接続します。キーを機能にバインドすると、プログラムはそのキーストロークが入力されるとバインドした機能を実行します。たとえば `ORACLE_HOME/forms/admin/resource/US` など、UNIX の `ORACLE_HOME/admin/resource/<言語ディレクトリ>` ディレクトリにある `fmrweb.res` ファイルにキー・バインディングを定義します。Windows の場合、ディレクトリは `ORACLE_HOME\Forms` になります。

キー・バインディングを定義することで、様々なキーボードを統合して、アプリケーションの使い勝手に類似性を持たせることができます。

プラットフォームによっては、再マップできないキーもあります。たとえば、Microsoft Windows では、キーは Windows のキーボード・デバイス・ドライバで定義されるので、キーの一部を再マップできません。[Alt]+[F4]（ウィンドウを閉じる）や [F1]（ヘルプ）など、Windows に必須のキーの組み合わせは、再マップできません。一般に、拡張キーボードの一部であるキーも再マップできません。これらのキーには、テンキー、矢印および編集キー、[Print Screen]、[Scroll Lock] および [Pause] などがあります。

注意：異なる NLS_LANG 設定で異なるリソース・ファイルを実行する場合は、NLS_LANG=GERMAN_GERMANY=WE8ISO8859P1 fmrwebd.res などのファイルが使用されます。

サポートされる各言語には、リソース・ファイルがあります。これを上書きするには、パラメータ term=fullpath¥filename.res を Oracle Forms Runtime プロセスに渡します。

このパラメータは URL 内で直接渡すことができます。次に例を示します。

```
http://hostname/forms/f90servlet?Form=test.fmx&term=fullpath¥filename.res
```

formsweb.cfg ファイルでこのパラメータを設定することもできます。次に例を示します。

```
otherParams=term=fullpath¥filename.res
```

4.11.1 fmrweb.res のカスタマイズ

fmrweb.res は、UNIX の vi あるいは Windows のメモ帳やワードパッドなどのテキスト・エディタを使って編集できるテキスト・ファイルです。Oracle 6i Forms とは異なり、Oracle Terminal エディタは不要です。テキスト・ファイルは自己ドキュメント化されます。

注意： 特に文字モード・フォームと比較した場合、カスタマイズには制限があります。Oracle Enterprise Manager Application Server Control では fmrweb.res を編集することはできません。

4.11.1.1 変更の例：入力マッピングと実行マッピングのスイッチング

USER-READABLE STRINGS とマークされたセクションで、以下のエントリを検索します。

```
122 : 0 : "F11" : 76 : "Enter Query"
122 : 2 : "Ctrl+F11" : 77 : "Execute Query"
```

これを次に変更します。

```
122 : 2 : "Ctrl+F11" : 76 : "Enter Query"
122 : 0 : "F11" : 77 : "Execute Query"
```

注意： デフォルトでは、fmrweb.res において、Microsoft Windows のクライアント / サーバーのキーボード・マッピングは反映されません。UNIX X Window/Motif でクライアント / サーバーを実行している場合は、キー・マッピングが反映されます。

Microsoft Windows のクライアント / サーバーのキーボード・マッピングを実行する fmrpcweb.res と呼ばれるファイルも提供されます。このファイルを使用するには、fmrpcweb.res を fmrweb_orig.res などの名前に変更して、fmrpcweb.res を fmrweb.res にコピーします。または、前述のとおり term パラメータを使用します。

4.11.1.2 例外または特殊なキー・マッピング

次に特殊なキー・マッピングの例を示します。

- 第 4.11.1.2.1 項「[F2] のマッピング」
- 第 4.11.1.2.2 項「KEY-ENTER-TRIGGER を起動する [Enter] のマッピング」
- 第 4.11.1.2.3 項「ナンバー・キーのマッピング」
- 第 4.11.1.2.4 項「Web フォームを終了する [Esc] キーのマッピング」

4.11.1.2.1 [F2] のマッピング

[F2] をマップするには、[F2] のデフォルトのエントリ、"List Tab Pages" を別のキーに変更します。次にデフォルトのエントリの例を示します。

```
113: 0 : "F2" : 95 : "List Tab Pages"
```

次に示すように、これを別のキー・マッピングに明示的に変更する必要があります。

```
113: 8 : "F2" : 95 : "List Tab Pages"
```

F2 機能を [F2] キーにマップするには、113:0 および 113:8 で始まる行を # 記号でコメント・アウトして、次の行をリソース・ファイルの最後に追加します。

```
113: 0 : "F2" : 84 : "Function 2"
```

```
113: 8 : " " : 95 : " "
```

デフォルトで [F2] を使用する新しい機能が追加されたので、[F2] キーをマップするにはこの新しい機能を別のものに明示的にマップする必要があります。この機能の追加により、タブ・キャンバス・ページ間のキーボード・ナビゲーションが可能になりました。このデフォルトは [F2] です。この機能がコメント・アウトされており、[F2] に割り当てられていなくても、この機能 Forms Function Number 95 が別のキーにマップされないかぎり [F2] キーはマップされません。

4.11.1.2.2 KEY-ENTER-TRIGGER を起動する [Enter] のマッピング

デフォルトでは、クライアント / サーバーまたは Web 上のいずれの場合も [Enter] キーを押して配置すると、カーソルがブロックの次のナビゲート先のアイテムに移動します。このデフォルトの動作を上書きするには、フォームのリソース・ファイルを変更して、キーのマッピングの詳細を変更する必要があります。

fmrweb.res を変更して、[Return] キーに対する Forms Function Number (FFN) を 27 から 75 に変更します。行は次のように変更します。

```
10 : 0 : "Return" : 75 : "Return"
```

デフォルトでは、この行の FFN は次のように 27 と表示されています。

```
10 : 0 : "Return" : 27 : "Return"
```

[Return] キーまたは [Enter] キーは実際に FFN の 27 に示される Return 機能を返しているのですが、この行では Key-Enter トリガを起動しません。FFN の 75 は Enter 機能を示しており、Key-Enter トリガを起動します。

4.11.1.2.3 ナンバー・キーのマッピング

fmrweb.res で [Ctrl]+<番号> キーを 0 から 9 の数字にマップすることができます。fmrweb.res には数字で表される Java 関数キーは含まれていません。例で実行する手順は、[CTRL]+[1] を「次レコード」にマップするために必要な手順を示しています。

1. fmrweb.res ファイルでキー・マッピングに実装する Java 関数キーの番号をリストします。次に例を示します。

```
public static final int VK_1 = 0x31;
```

2. 16 進値は、fmrweb.res で使用する前に、10 進の等価に変換する必要があります。

ステップ 1 では、0x31 は 16 進値なので 10 進の等価に変換する必要があります (Note:1019580.6)。次に例を示します。

```
SQL> select hextoDEC('31') from dual;
```

```
HEXTODEC('31')
```

```
-----
```

```
49
```

3. この 10 進値を fmrweb.res の数字キー [1] のマッピングに使用します。たとえば、[Ctrl]+[1] は次のとおり「次レコード」にマップされます。

```
49 : 2 : "CTRL+1" : 67 : "Next Record"
```

4.11.1.2.4 Web フォームを終了する [Esc] キーのマッピング

1. fmrweb.res のバックアップ・コピーを作成します。
2. パス ORACLE_HOME/FORMS で示される fmrweb.res ファイルを開いて、次のエントリを追加します。

```
27 : 0 : "Esc" : 32 : "Exit"
```

3. 古いエントリをコメント・アウトするか削除します。

```
#115 : 0 : "F4" : 32 : "Exit"
```

最初の数字（115）は、別のバージョンまたはプラットフォームでは異なる可能性があります。Web Form を実行して、[Esc] キーを押すと、Form は終了します。

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。

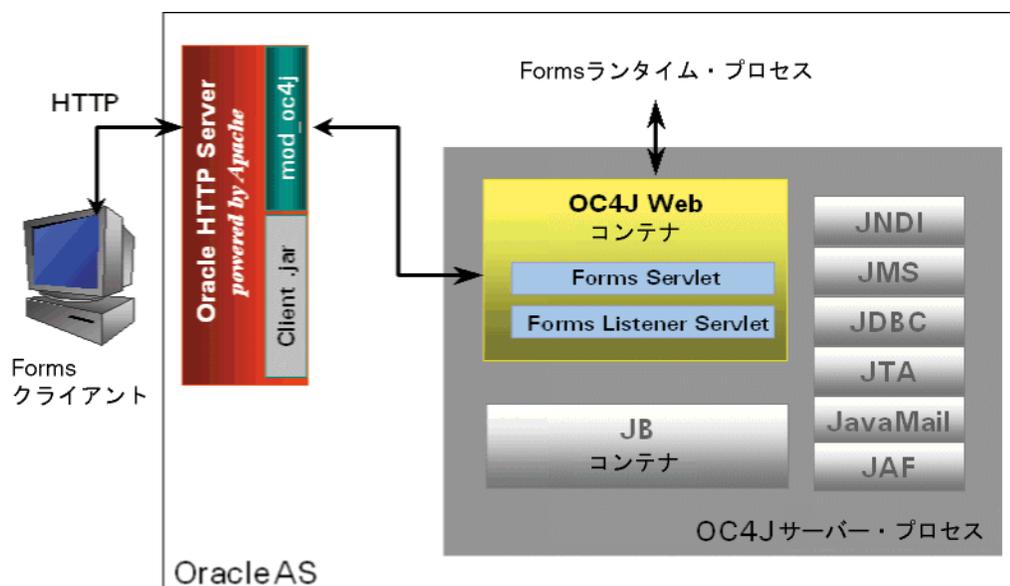
- 第 5.1 項 「OC4J サーバー・プロセス」
- 第 5.2 項 「パフォーマンス / スケーラビリティのチューニング」
- 第 5.3 項 「HTTPD プロセス数の制限」
- 第 5.4 項 「高い値の MaxClients ディレクティブの設定」
- 第 5.5 項 「OC4J のロード・バランシング」
- 第 5.6 項 「Forms Listener Servlet での HTTPS の使用方法」
- 第 5.7 項 「サーバー要件」
- 第 5.8 項 「クライアント要件 : Oracle JInitiator での HTTPS の使用」
- 第 5.9 項 「ユーザー ID / パスワード非表示機能の使用法」
- 第 5.10 項 「認証プロキシを使用した Oracle Forms アプリケーションの実行」
- 第 5.11 項 「Oracle Forms Services と SSL」

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せんが、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（第 5.2 項「パフォーマンス / スケーラビリティのチューニング」を参照）。

図 5-1 「OC4J アーキテクチャと Forms Services」は、OC4J アーキテクチャがどのように Forms Services によって使用されるかを示しています。

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、Start Here CD-ROM または OTN

(<http://www.oracle.com/technology/documentation/>) から入手可能な『Oracle Application Server パフォーマンス・ガイド』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 をデフォルトの 15 秒など低めの数字に設定します。

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』(Start Here CD-ROM または OTN (<http://www.oracle.com/technology/products/ias/>) から入手可能) の OC4J に関する章を参照してください。

Forms 固有の情報の詳細は、Oracle Developer Suite と Oracle Application Server のリリース・ノートを参照してください。

次の図では、配置例を 4 つ示します。

- [図 5-2](#): 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
- [図 5-3](#): 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
- [図 5-4](#): 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
- [図 5-5](#): 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 Oracle HTTP Listener と同じホスト上にある複数の OC4J エンジン

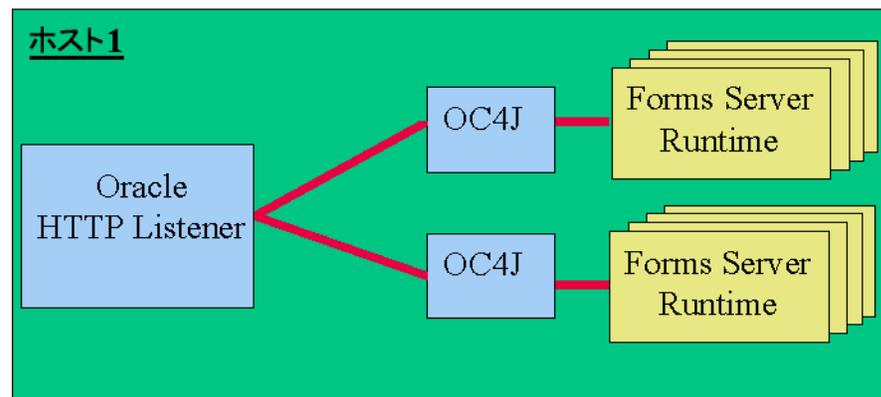


図 5-3 Oracle HTTP Listener と異なるホスト上にある複数の OC4J エンジン

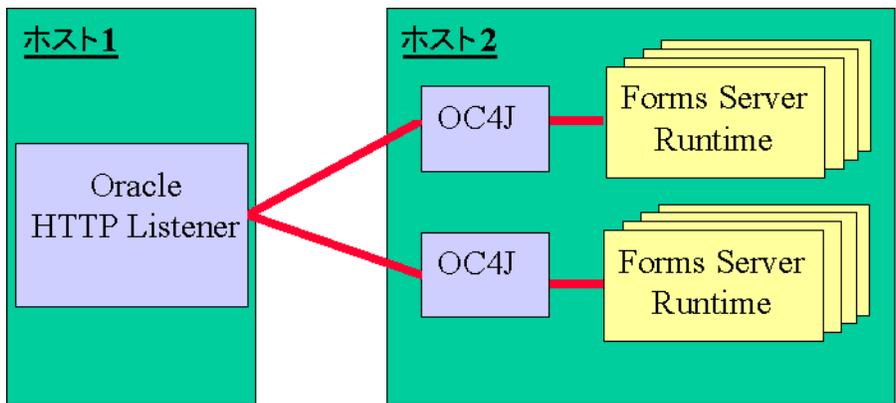


図 5-4 それぞれ異なるホスト上にある複数の OC4J エンジンと複数の Oracle HTTP Listener

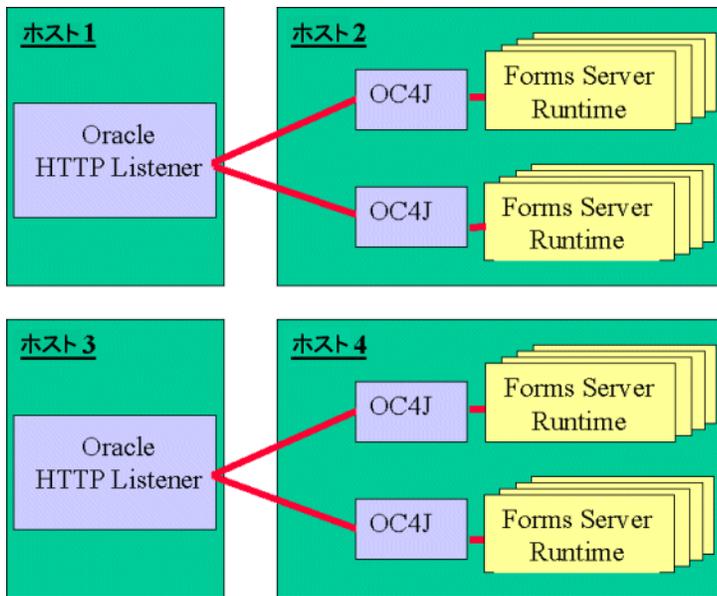
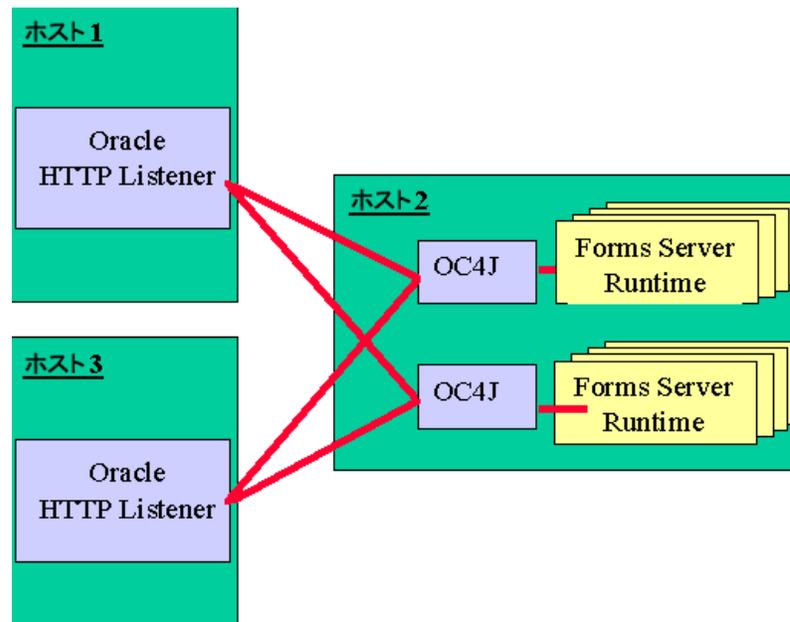


図 5-5 異なるホスト上にある複数の Oracle HTTP Listener と同じホスト上にある複数の OC4J エンジン



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 の Start Here CD-ROM および Oracle Technology Network (OTN) (<http://www.oracle.com/technology/products/ias/>) から入手可能な『Oracle Application Server パフォーマンス・ガイド』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、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コンピュータの `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。

ユーザーの Web サイトのルート認証局を定義するには：

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「ウォレットのエクスポート」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 B 「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方法

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（大文字と小文字の区別なし）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す。認証プロキシでは、アプリケーションを実行する接続先サーバーにアクセスできるように、ユーザーがユーザー名とパスワードを指定する必要があります。認証プロキシは通常、ユーザーがログインしているか（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/forms/frmservlet?config=myApp` などの /forms 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、/forms/java および /forms/lervlet です。これらはページのドキュメント・ベース（/forms）の下にあるので、認証プロキシが機能します。

5.11 Oracle Forms Services と SSL

Oracle Application Server 10g のインストール時、HTTPS ポートはデフォルトで無効です。SSL を有効にするには、まず Oracle HTTP Server の HTTPS ポートを有効にします。次に Web Cache を有効にして、Oracle HTTP Server からの HTTPS 接続を受け入れます。

注意：（実行しているアプリケーションの構成時にユーザー / パスワードが指定されておらず）ログイン・ダイアログが表示されるように Forms アプリケーションがコーディングされている場合、SSL/HTTPS でアプリケーションを実行しないと、セキュリティ上の危険があることに注意してください。ログオン・ダイアログに入力するパスワードは、ネットワーク上に送信されます。

5.11.1 SSL を使用する Oracle HTTP Server の構成

SSL を使用するよう Oracle HTTP Server を構成するとき、Oracle Process Manager and Notification Server (OPMN) の構成ファイルの一部を変更します。このファイルを変更および保存したら、OPMN プロセスを再起動する必要があります。

SSL を使用する Oracle HTTP Server を構成するには：

1. テキスト・エディタで ORACLE_HOME/opmn/conf/opmn.xml を開き、次のコード・ブロックを検索します。

```
<ias-component id="HTTP_Server">
  <process-type id="HTTP_Server" module-id="OHS">
    <module-data>
      <category id="start-parameters">
        <data id="start-mode" value="ssl-disabled"/>
      </category>
    </module-data>
    <process-set id="HTTP_Server" numprocs="1"/>
  </process-type>
</ias-component>
```

2. start-mode パラメータの値を ssl-enabled に変更します。

```
<ias-component id="HTTP_Server">
  <process-type id="HTTP_Server" module-id="OHS">
    <module-data>
      <category id="start-parameters">
        <data id="start-mode" value="ssl-enabled"/>
      </category>
    </module-data>
    <process-set id="HTTP_Server" numprocs="1"/>
  </process-type>
</ias-component>
```

3. 変更された opmn.xml 構成ファイルを OPMN が強制的にリロードするようにします。

```
opmnctl reload
```

5.11.2 SSL を使用する Oracle Web Cache の構成

「Web Cache Admin」 ページを使用して、Oracle HTTP Server からの HTTPS の接続を有効にします。

SSL を使用する Oracle Web Cache を構成するには：

- 「Web Cache Manager」 ページを開きます。インストール時に OracleAS Web Cache を構成している場合は、`http://hostname.domain:port` からこれにアクセスできます。デフォルトのポートは 4000 です。また、次のファイルに Web Cache の HTTP Listen ポート番号がリストされています。
 - Solaris の場合：`ORACLE_HOME/install/portlist.ini`
 - Windows の場合：`ORACLE_HOME\install\portlist.ini`
- アプリケーション・サーバーの管理者としてログインします。
- ナビゲータ・フレームで「Port」に移動して、「Listen Ports」をクリックします。
- 「Add」をクリックします。
- 「For Cache」ドロップダウン・リストから、対象となる Web Cache を選択します。
- 表 5-1 に示す情報を入力します。

表 5-1 HTTPS ポートの構成情報

設定	説明
IP Address	有効な任意の IP アドレス
Port Number	443
Protocol	HTTPS
Require Client-Side Certification	<p>クライアント側の証明書を有効または無効にします。</p> <p>「Require Client-Side Certificate」を選択して、OracleAS Web Cache がブラウザに SSL 証明書を要求するようにします。それには <code>webcache/wallet/default</code> から <code>ewallet.p12</code> ファイルをインポートする必要があります。</p> <p>クライアント側の証明書は、クライアントの識別情報を検証する方法の 1 つです。証明書は、クライアント・ユーザーの情報をユーザーの公開鍵にバインドするもので、信頼できる認証局によってデジタル署名されている必要があります。</p>
Wallet	<p>Wallet のディレクトリ上の場所を入力します。このディレクトリには Wallet がすでに含まれている必要があります。この Wallet は、OracleAS Web Cache によってホストされるサイトの HTTPS リクエストを管理、無効化および統計監視するために使用されます。</p> <p>このデフォルトを使用している場合でも、Wallet の場所を入力することをお勧めします。デフォルトのパスは、<code>ORACLE_HOME/webcache/wallets/default</code> です。</p>

注意： クライアント側の証明書を選択および使用するときは、Sun Java Plug-in 1.4.2 以上を使用してください。詳細は、<http://java.sun.com/> を参照してください。

7. 「Apply Changes」をクリックします。
8. Web Cache を再起動します。

5.11.3 SSL を使用するフォームの実行

HTTPS ポートを使用する Forms アプリケーションを実行するには、証明書をインポートする必要があります。ポート 4443 から Web Cache にアクセスする場合、Web Cache 証明書をインポートする必要があります。ポート 4444 から Oracle HTTP Server にアクセスする場合、Oracle HTTP Server の証明書をインポートする必要があります。

Web Cache 証明書をインポートするには：

1. Wallet Manager を起動します。
 - Windows の場合：Wallet Manager の launch.exe を起動します。
 - Solaris の場合：owm
2. ORACLE_HOME¥WebCache¥wallets¥default を開きます。
3. パスワードとして welcome を入力します。
4. 「自動ログイン」を選択します。
5. 「テスト用のみ」を選択します。
6. 「操作」→「信頼できる証明書のエクスポート」を選択し、名前を入力します。
7. このファイルをテキスト・エディタで開き、内容をすべて選択して C:¥Program Files¥Oracle¥JInitiator 1.3.1.21¥lib¥security¥certdb.txt に追加します。
8. Form Servlet を https://computer.mycompany.com:4443/forms/frmservlet として実行します。
9. JInitiator ログ・ウィンドウが HTTPS プロトコルを表示することを確認します。

Oracle HTTP Server の証明書をインポートするには：

1. Wallet Manager を起動します。
 - Windows の場合：Wallet Manager の launch.exe を起動します。
 - Solaris の場合：owm
2. ORACLE_HOME¥Apache¥Apache¥conf¥ssl.wlt¥default を開きます。
3. パスワードとして welcome を入力します。
4. 「自動ログイン」を選択します。
5. 「テスト用のみ」を選択します。
6. 「操作」→「信頼できる証明書のエクスポート」を選択し、名前を入力します。
7. このファイルをテキスト・エディタで開き、内容をすべて選択して C:¥Program Files¥Oracle¥JInitiator 1.3.1.21¥lib¥security¥certdb.txt に追加します。
8. Form Servlet を https://computer.mycompany.com:4444/forms/frmservlet として実行します。
9. JInitiator ログ・ウィンドウが HTTPS プロトコルを表示することを確認します。

5.11.4 ロード・バランシング・ルーターを使用した SSL の構成

HTTPS ポートを使用する Forms アプリケーションを実行するには、証明書をインポートする必要があります。Forms サーバーがロード・バランシング・ルーターの後ろにあり、Forms サーバーで SSL が終了する場合、証明書をロード・バランシング・ルーターからインポートする必要があります。

ロード・バランシング・ルーター上の Forms アプリケーションで SSL を有効にするには、次の手順に従います。

1. Web ブラウザで `https://mycomputer.us.oracle.com:443/forms/frmservlet` を開き、「**セキュリティの警告**」ダイアログを開きます。
2. 「**証明書の表示**」をクリックします。
3. 「証明書」ダイアログで、「**詳細**」タブをクリックします。
4. 「**ファイルにコピー**」をクリックします。
5. 証明書のエクスポート・ウィザードの開始ページで「**次へ**」をクリックします。
6. 「エクスポート ファイルの形式」ページで、「**Base-64 encoded X.509 (.CER)**」を選択し、「**次へ**」をクリックします。
7. `c:¥temp¥forms` などのファイル名を入力して、「**次へ**」をクリックします。
8. 「**完了**」をクリックします。

エクスポートが正常に終了したことを知らせるメッセージが表示されます。

9. 「**OK**」をクリックします。
10. 「**セキュリティの警告**」ダイアログを開いたまま証明書のエクスポート・ウィザードを終了します。
11. テキスト・エディタで `c:¥temp¥forms.cer` を開きます。
12. ファイルの内容を `lib¥security` にある `JInitiator` の `certdb.txt` ファイルへコピーします。
13. `certdb.txt` を保存します。
14. 「**セキュリティの警告**」ダイアログをもう一度開いて、「**はい**」をクリックします。

Oracle Application Server Single Sign-On での Forms Services の使用

この章には、次の項が含まれています。

- 第 6.1 項「概要」
- 第 6.2 項「OracleAS Single Sign-On、Oracle Internet Directory、および Forms で使用可能な機能」
- 第 6.3 項「Oracle Forms で使用される OracleAS Single Sign-On コンポーネント」
- 第 6.4 項「アプリケーションでの OracleAS Single Sign-On の有効化」
- 第 6.5 項「Oracle Forms と Reports の統合」
- 第 6.6 項「認証フロー」

6.1 概要

Oracle Application Server Single Sign-On は、共有される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 を使用して、ユーザー名とパスワード情報が格納されるシングル・サインオン環境で実行できます。OracleAS Single Sign-On 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する Web 環境で機能するように設計されています。OracleAS Single Sign-On 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。

注意： シングル・サインオンは Oracle Application Server Forms and Reports Services のインストール・タイプには含まれていません。OracleAS Infrastructure の使用方法の詳細は、Oracle Application Server Forms and Reports Services のインストレーション・ガイドを参照してください。

OracleAS 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）にも OracleAS Single Sign-On を有効にできます。

Oracle Forms アプリケーションは、OracleAS Single Sign-On と Oracle Internet Directory に基づく企業の OracleAS Single Sign-On アーキテクチャに透過的に統合できます。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シングル・サインオンが、最初からサポートされています。Forms アプリケーションでさらにコーディングを行う必要はありません。

6.2 OracleAS Single Sign-On、Oracle Internet Directory、および Forms で使用可能な機能

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。

- 第 6.2.1 項「Oracle Internet Directory にリソースがない場合のリソースの動的な作成」
- 第 6.2.2 項「Oracle Internet Directory 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」
- 第 6.2.3 項「Forms と OracleAS Single Sign-On での動的ディレクティブのサポート」
- 第 6.2.4 項「OracleAS Single Sign-On 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」

6.2.1 Oracle Internet Directory にリソースがない場合のリソースの動的な作成

ユーザーが Forms に接続すると、OracleAS Single Sign-On Server および Oracle Internet Directory 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 OracleAS Single Sign-On 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、Oracle Internet Directory 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 ユーザーが、リクエストしている特定のアプリケーションのリソース、または Oracle Internet Directory のデフォルトのリソースのいずれも持たない場合、Oracle Internet Directory/DAS ページにリダイレクトされリソースが動的に作成されます。このリソースの作成後、ユーザーは元の Forms のリクエスト URL に再びリダイレクトされます。

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。

- リソースの動的な作成の許可（デフォルト）
- ssoErrorUrl パラメータに指定されている事前定義された URL へのユーザーのリダイレクト
- Forms のエラー・メッセージの表示

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。

6.2.2 Oracle Internet Directory 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 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ていました。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 Oracle Internet Directory 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。

6.2.3 Forms と OracleAS Single Sign-On 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 OracleAS Single Sign-On が設定されています。また、新しい OracleAS Single Sign-On 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 OracleAS Single Sign-On 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 OracleAS Single Sign-On 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 OracleAS Single Sign-On が構成されているため、Oracle Enterprise Manager Application Server Control コンソールで OracleAS Single Sign-On パラメータの読取りと書込みができます。

6.2.4 OracleAS Single Sign-On 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 Oracle Internet Directory に伝播されませんでした。

このリリースの OracleAS 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、OracleAS Single Sign-On モードで実行される 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、Oracle Internet Directory 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 OracleAS Single Sign-On 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 Oracle Internet Directory で更新されません。

6.3 Oracle Forms で使用される OracleAS Single Sign-On コンポーネント

OracleAS Single Sign-On 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、Oracle Application Server アプリケーションで認証を透過的に行います。OracleAS Forms Services および OracleAS 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 : OracleAS 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 OracleAS Single Sign-On が必要かどうかを検出し、リクエストを OracleAS 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 OracleAS Single Sign-On 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 forms/server ディレクトリにあります。

6.4 アプリケーションでの OracleAS Single Sign-On の有効化

Oracle Forms アプリケーションは、中央の構成ファイル (forms/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 コンソールを使用して編集することをお勧めします。

OracleAS 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。OracleAS Single Sign-On 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。

6.4.1 ssoMode

ssoMode パラメータによって、Forms Services アプリケーションで OracleAS Single Sign-On が有効になります。デフォルトでは、OracleAS Single Sign-On 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 OracleAS Single Sign-On 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 OracleAS Single Sign-On 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 OracleAS Single Sign-On を有効にするには：

1. Oracle Enterprise Manager Application Server Control コンソールを起動します。
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。
8. 「適用」をクリックして、formsweb.cfg ファイルを更新します。
選択されたアプリケーションで SSO が有効になります。

アプリケーションの OracleAS Single Sign-On を無効にするには：

1. Oracle Enterprise Manager Application Server Control コンソールを起動します。
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。

6.4.2 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 Oracle Internet Directory でリソース・アクセス記述子 (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、Oracle Delegated Administration Services 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。

リソースが動的に作成されることによって、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、Oracle Internet Directory の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、OracleAS Single Sign-On を有効にしてアプリケーションを構成し、ssoErrorURL に値を指定しないでおくと、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 (Oracle Internet Directory 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、Oracle Internet Directory 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、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 コンソール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。

6.4.3 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
...
```

6.4.4 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。

6.4.5 Forms から SSO 情報へのアクセス

Forms アプリケーションで OracleAS Single Sign-On 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、OracleAS Single Sign-On のログイン情報を取得できます。SSO のログイン情報には、OracleAS 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') ;
formsAppEntity := get_application_property('sso_formsid') ;
config := get_application_property('config').
```

注意: config は非シングル・サインオン・モードでも取得できます。

6.5 Oracle Forms と Reports の統合

Oracle Reports は OracleAS Single Sign-On を有効にしてインストールします。

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。

OracleAS Single Sign-On 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 OracleAS Single Sign-On 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、OracleAS Single Sign-On ID を使用して Reports Server へのユーザーを認証します。

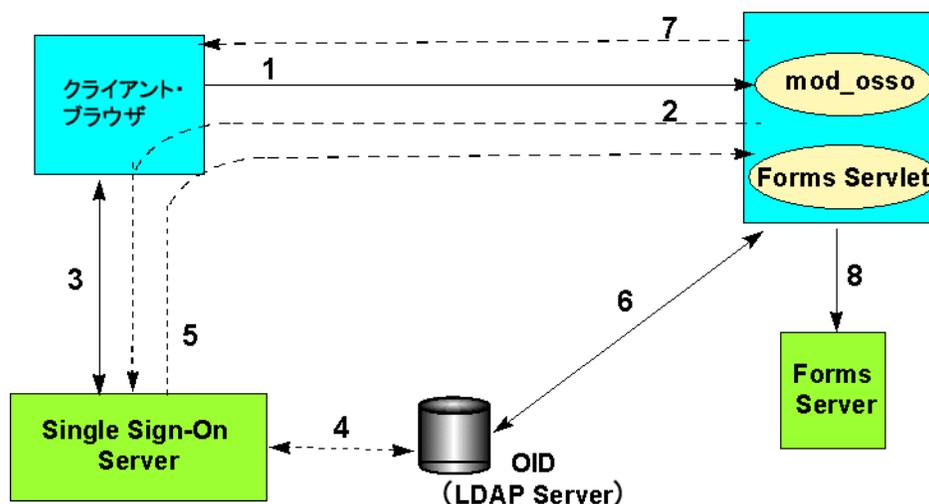
非 OracleAS Single Sign-On モードで実行される Forms アプリケーションでは、シングル・サインオンで保護された Reports Server のレポートを実行できますが、Reports Server で認可が必要な場合は Forms アプリケーションが失敗します。また、Web に Reports 出力を取り込む場合、ユーザーは OracleAS Single Sign-On 接続情報を指定する必要があります。

Oracle Forms と Oracle Reports の統合の詳細は、Oracle Technology Network (<http://www.oracle.com/technology/products/forms/>) にあるホワイト・ペーパー『Integrating Oracle Forms 10g and Oracle Reports 10g』を参照してください。

6.6 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 OracleAS Single Sign-On サポートの認証フローは図 6-1 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

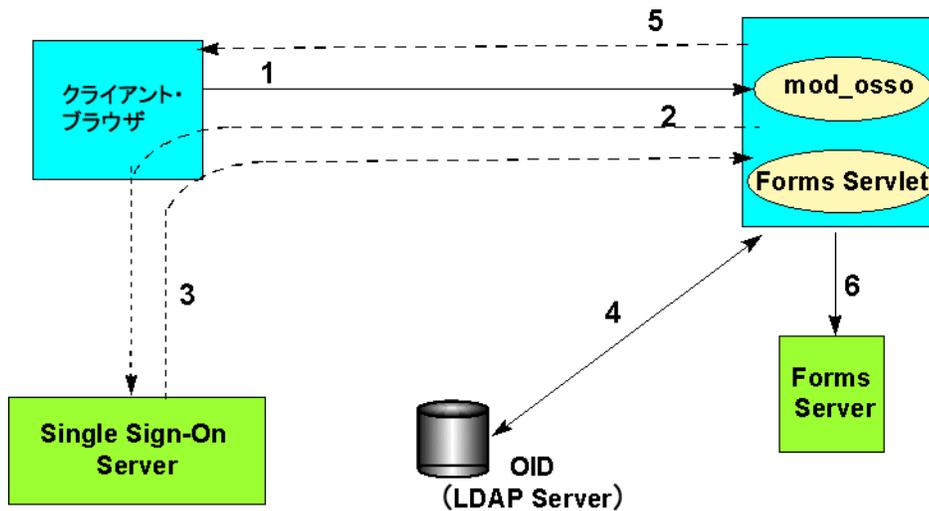


1. ユーザーは、
`http(s)://<hostname>:<port>/forms/frmservlet?config=<application>&..` のような Forms URL をリクエストします。
2. Forms Servlet が、そのユーザー・リクエストを OracleAS Single Sign-On サーバーにリダイレクトします。
3. ユーザーが、ログイン・フォームでユーザー名とパスワードを指定します。
4. パスワードが、Oracle Internet Directory (LDAP Server) で検証されます。
5. ユーザーのリクエストが、sso_userid 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、アプレットが Forms Listener Servlet に接続されます。
8. Forms Servlet が、Forms Server を起動します。

別のパートナ・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 サポートの認証フローは、[図 6-2](#) 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 OracleAS 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、アプレットが Forms Listener Servlet に接続されます。
6. Forms Servlet が、Forms Server を起動します。

JVM プーリング

この章には、次の項が含まれています。

- 第 7.1 項「概要」
- 第 7.2 項「JVM プーリングの例」
- 第 7.3 項「設計時の考慮事項」
- 第 7.4 項「JVM コントローラについて」
- 第 7.5 項「JVM プーリングの管理」
- 第 7.6 項「JVM コントローラのログ管理」
- 第 7.7 項「JVM プーリングのエラー・メッセージ」

7.1 概要

JVM プーリングを使用すると、管理者は使用される JVM の数を整理して、複数の Forms セッションで JVM を共有し、Forms セッションごとに個別の JVM インスタンスを持たないようにすることができます。JVM プーリングでは、メモリー消費が大幅に削減されるため、サーバーのより多くのリソースが解放されます。

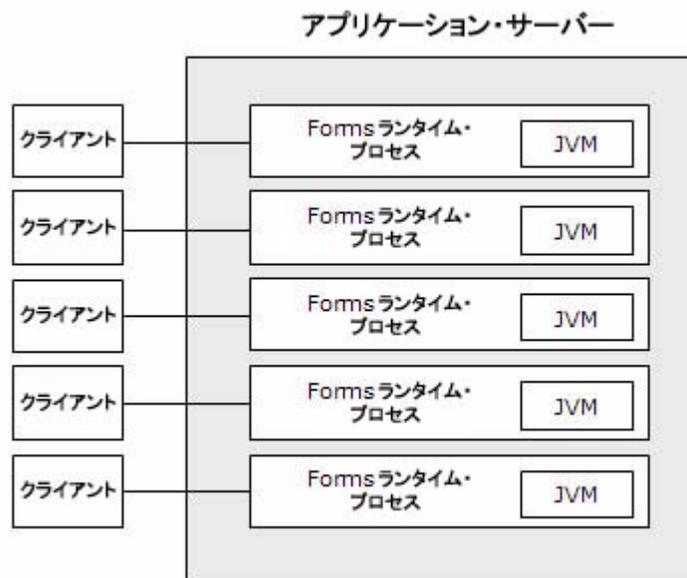
JVM プーリングは、Java Importer との組合せで機能します。Java Importer を使用すると、開発者は設計時に Forms Builder 内の PL/SQL から Java クラスを参照できます。実行時に、Forms は Java 仮想マシン (JVM) を使用して Java コードを実行します。JVM プーリングにより、Oracle Forms ではネイティブで実行できない操作が可能になったため、Forms アプリケーションの機能は拡張されます。Java Importer の詳細は、Oracle Forms Developer のオンライン・ヘルプを参照してください。

7.2 JVM プーリングの例

ここでは、ユーザー・インタフェースのボタンを持つ Oracle Forms アプリケーションを例に取り上げます。ボタンが押されると、Oracle Forms は画面上のフィールドから値を取得して Java に渡し (Java Importer 機能を使用)、PL/SQL では実行できないなんらかの複雑な計算を実行します。その後、結果が返され、フォーム内のフィールドに表示されます。

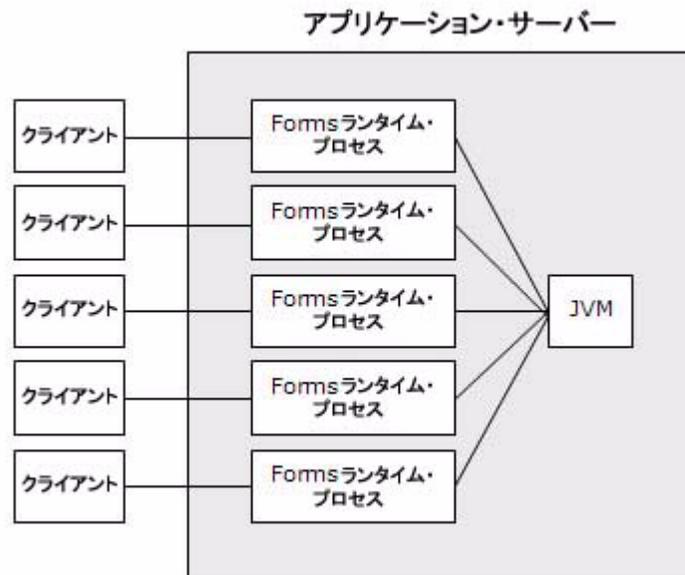
JVM プーリングが有効になっていない場合、Forms ランタイム・プロセスは図 7-1 のように動作します。この場合は、各 Oracle Forms セッションがインプロセス JVM という個別の JVM を持ちます。

図 7-1 JVM プーリングを使用しない場合の Forms ランタイム



JVM プーリングが有効な場合は、図 7-2 のように、理論上 1 つの JVM を複数の Forms プロセスで使用できます。

図 7-2 JVM プーリングが有効な場合の Forms ランタイム



この例では5つのJVMインスタンスが1つに減少し、メモリー使用量が削減されます。

7.3 設計時の考慮事項

この項は、次の項目に分かれています。

- 第 7.3.1 項「以前のバージョンの Java Importer について」
- 第 7.3.2 項「Java コードの再インポート」
- 第 7.3.3 項「複数の JVM 間での静的変数の共有について」

7.3.1 以前のバージョンの Java Importer について

Oracle Forms に Java Importer が追加されたときには、Java Importer を使用する各 Forms セッションに Java コードを実行するための個別の JVM インスタンスがありました。各 JVM がサーバー上のメモリーを消費するため、同時に多数のユーザーが存在すると、複数の JVM プロセスによって消費されるメモリーの量は膨大なものになる可能性があります。

7.3.2 Java コードの再インポート

JVM プーリングより先に Oracle Forms の Java Importer 機能を使用していた場合は、JVM プーリングを使用する前に Java クラスを再インポートする必要があります。Java クラスを初めてインポートしたときには、その Java クラスの PL/SQL ラッパーが生成されています。これは、フォームで作成されたプログラム・ユニット内で確認することができます。ただし、Java Importer で必要とされる PL/SQL ラッパーはこれとは異なります。

Oracle Application Server Forms Services 10g 以降では、Java Importer は新しい PL/SQL ラッパーを生成します。Java Importer を使用し、JVM プーリングを利用しない場合は、インプロセス JVM は新しい PL/SQL ラッパーで正常に機能します。また、旧スタイルの PL/SQL ラッパーでもこれまでどおりに機能します。

アプリケーションで Java を使用しても、JVM プーリングを使用しない場合は、Java クラスを再インポートする必要はありません。インプロセス JVM は、以前に生成された PL/SQL ラッパーでも新規に生成される PL/SQL ラッパーでも機能します。

注意： インプロセス JVM では、JVM は Oracle Forms ランタイム・プロセスの一部であり、別プロセスではありません。JVM プーリングを使用すると、JVM は Forms ランタイム・プロセスから独立した外部プロセスになります。

7.3.3 複数の JVM 間での静的変数の共有について

JVM プーリングなしで Java Importer 機能を使用していた場合、各 Forms ランタイム・プロセスは他の JVM とは別に独自のインプロセス JVM を持っています。JVM プーリングを使用すると、複数の Forms ランタイム・プロセスが同じ JVM 内で Java コードを実行するようにできます。このメリットの 1 つに、同一クラスの複数のインスタンス間で静的変数を使用してデータを共有する機能があります。ただし、静的変数は 1 つの JVM 内の同一クラスのインスタンス間で共有され、複数の JVM 間では共有されません。この条件に従って計画を立案する必要があります。

たとえば、ローン・クラスのすべてのインスタンスで同じ金利を使用して計算を実行するという理由で、`interestRate` という静的変数が用意されているとします。JVM を 1 つのみ使用する場合は、ローン・クラスの 1 つのインスタンスで `interestRate` が変更されると、他のすべてのインスタンスが影響を受けます（これは意図した動作です）。

ただし、JVM コントローラに 1 つ以上の子 JVM が存在する場合は、2 つ以上の JVM が存在する場合があります。1 つの JVM で `interestRate` が変更されても、他の JVM にあるローン・インスタンスでは新しい値が認識されません。子 JVM の管理の詳細は、[第 7.5.14 項「子 JVM について」](#)を参照してください。

JVM プーリングを使用しておらず、`interestRate` を変更した場合は、各 Oracle Forms Services のランタイム・プロセスが個別にインプロセス JVM を持つため、その変更により他のインスタンスは影響を受けません。

静的変数を使用してクラスの複数のインスタンス間で情報を共有する場合は、`maxsessions` を 65535 に設定して、子 JVM が生成されないようにしてください。

7.4 JVM コントローラについて

アプリケーション・サーバーには、Oracle Forms セッションごとに 1 つの Oracle Forms ランタイム・プロセスが存在します。Oracle Forms は、実際にはこのプロセスで実行されます。また、このプロセスは、データベース接続の管理、データの間合せと更新、フォーム内の任意の PL/SQL の実行、トリガーの実行なども行います。

また、Oracle Forms ランタイム・プロセスには、アプリケーション内で Java を実行するための JVM も含まれています。最適化のために、JVM は Oracle Forms アプリケーションで Java Importer が使用される場合のみ起動されます。Java Importer を使用しないアプリケーションでは、JVM 用にメモリーが消費されることはありません。

JVM プーリングは、JVM コントローラを含む新しいプロセスです。JVM プーリングでは、JVM は Oracle Forms ランタイム・プロセスの外部で実行されます。また、JVM を複数の Oracle Forms ランタイム・プロセスで共有することもできます。JVM コントローラ・プロセスは JVM そのものではなく、JVM を含むコンテナです。これは、Oracle Forms ランタイム・プロセスがインプロセス JVM を含む仕組みと同様です。

注意： JVM プーリングの使用はオプションです。管理者は、JVM プーリングを使用せず、JVM が Oracle Forms ランタイム・プロセスに含まれるようにすることもできます。

Oracle Forms ランタイム・プロセスは、Java の実行が必要になると、JVM コントローラに含まれる JVM にメッセージを送信します。JVM は、その Oracle Forms ランタイム・プロセス用に新しいスレッドを作成します。その後、JVM は別の Oracle Forms ランタイム・プロセスからの次の新しいリクエストのリスニングを続行します。一方、リクエストは新しく作成されたスレッドによって処理され、結果が Oracle Forms ランタイム・プロセスに返されます。該当の Oracle Forms セッションが有効である間、Oracle Forms ランタイム・プロセスはそのスレッドと直接通信します。

各 Oracle Forms ランタイム・プロセスが JVM 内に個別にスレッドを持つため、同時実行性が実現します。指定された同時実行リクエスト数に JVM が達すると、子 JVM が生成され、負荷が分散されます。さらに、複数の JVM コントローラを持ち、それぞれに子 JVM を持つことも可能です。

たとえば、複数の Oracle Forms アプリケーションで、オプションやクラスパスが異なる JVM を使用する場合があります。特定の Oracle Forms アプリケーションで使用する JVM コントローラは、Oracle Forms 構成ファイル (formsweb.cfg) の名前を付けたセクションで指定できます。詳細は、第 7.5.14 項「子 JVM について」を参照してください。

図 7-3 は、JVM プーリングを使用した場合の環境の例を示しています。2 つの JVM コントローラがあり、1 つはインプロセス JVM のみを使用し、もう 1 つは 3 つの JVM を使用しています。

図 7-3 子プロセスを持つ複数の JVM コントローラ

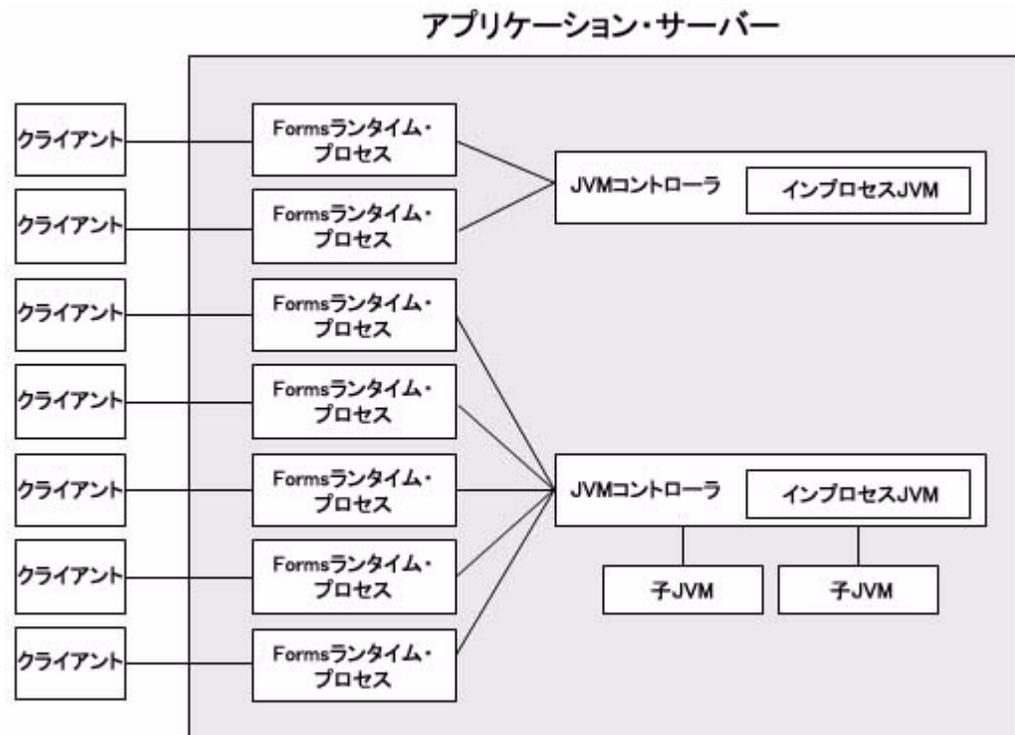


図 7-3 には示されていませんが、各 JVM コントローラには、起動、停止、または Oracle Forms 構成ファイル内で参照に使用される一意な名前があります。

図 7-3 は、複数の Oracle Forms アプリケーションで異なる JVM コントローラを使用していることを概念的に示しているだけです。ただし、Oracle Forms ランタイム・プロセスは、JVM コントローラと通信せず、使用可能な JVM のいずれかと直接通信します。したがって、図の最初の 2 つのクライアントはインプロセス JVM を使用できるのですが、残りのクライアントには処理に使用できる JVM が 3 つあります。

7.5 JVM プーリングの管理

Oracle Forms ランタイム・プロセスが JVM と直接対話している間、JVM コントローラは管理者からのコマンド（JVM の停止やロギングの有効化 / 無効化など）を受け付けます。たとえば、管理者が停止コマンドを発行すると、そのコマンドによってすべての子 JVM も停止します。

JVM コントローラは、次の 2 つの方法で管理できます。

- Enterprise Manager Application Server Control による管理（推奨）
- コマンドラインによる管理

7.5.1 Enterprise Manager Application Server Control による JVM コントローラの管理について

Enterprise Manager Application Server Control では、使用可能なすべての JVM プーリング・オプションを管理するための Web ベースの環境が用意されています。Enterprise Manager Application Server Control が JVM コントローラと対話することで、管理者は JVM プーリングを管理することができます。Enterprise Manager Application Server Control は、JVM コントローラのすべての JVM を内部的に管理する集中管理のポイントとなります。また、Enterprise Manager Application Server Control では、環境内のすべての JVM コントローラが一覧表示され、(リモートで) これらを管理することができます。たとえば、JVM コントローラの起動や停止、新規コントローラの追加、既存のコントローラの再構成などができます。さらに、Enterprise Manager Application Server Control では、JVM コントローラによって消費されるリソース（メモリーや CPU）などのメトリック情報も取得することができます。

Application Server Control の「JVM」ページは、次のような JVM プーリング・タスクの管理に使用します。

- [第 7.5.3 項「新規 JVM コントローラの作成」](#)
- [第 7.5.5 項「Enterprise Manager Application Server Control による JVM コントローラのプロパティの編集」](#)
- [第 7.5.6 項「デフォルト JVM コントローラのプロパティの指定」](#)
- [第 7.5.4 項「JVM コントローラの削除」](#)
- [第 7.5.7.1 項「JVM コントローラの起動または再起動」](#)
- [第 7.5.8.3 項「JVM コントローラの停止」](#)
- [第 7.6 項「JVM コントローラのロギング管理」](#)

表 7-1 で、「JVM 概要」ページに表示される情報について説明します。

表 7-1 Application Server Control の JVM プーリングの概要情報

項目	説明
合計メモリー使用率 (%)	プライベート、共有および合計のメモリー使用量の統計を示します。 例: プライベート 0.8519 共有 3.4226 合計 4.2745
すべての詳細を表示	開くと、すべての JVM インスタンスのクラスパスの情報が表示されます。
JVM オプション	この JVM について有効になっているオプションを表示します。
ログ・ディレクトリ	ログ・ファイルへの完全なパスを表示します。
コメント	JVM コントローラに関するコメントを表示します。例: start time: Fri Aug 20 03:58:57 2004 PDT
すべての詳細を非表示	すべての JVM インスタンスについて、表示されている情報をすべて閉じます。

表 7-1 Application Server Control の JVM プーリングの概要情報 (続き)

項目	説明
選択	このラジオ・ボタンを使用して管理ターゲットの JVM を選択します。
詳細	「表示 (+)」または「非表示 (-)」をクリックして、選択した JVM インスタンスの情報を開くか、または閉じます。
名前	該当の JVM コントローラが作成されたときの名前を表示します。
ステータス	JVM コントローラが実行中かどうかを示します。
CPU 使用率 (%)	CPU 使用率の統計をパーセンテージで表示します。
プライベート・メモリー使用率 (%)	プライベート・メモリーの使用率をパーセンテージで表示します。
JVM	ターゲット JVM の実行中のインスタンス数を表示します。
現行セッション数	ターゲット JVM に接続されているセッション数を表示します。
JVM 当たりの最大セッション数	1 つの JVM に接続可能なセッション数に関して指定された最大数を表示します。
ロギング	該当の JVM についてロギングが有効になっているかどうかを示します。
ログ・ファイル	ロギングが有効になっている場合、このアイコンがログ・ファイル表示用のリンクになります。

7.5.2 コマンドラインによる JVM コントローラの管理について

コマンドラインで JVM コントローラを管理する場合は、JVM を起動および停止するオプションと環境を指定するオプションについての知識が必要です。アクセスできる JVM コントローラは、同一コンピュータから実行されているもののみです。

注意： この章で説明する JVM コントローラの制御手順は、ほとんどがコマンドラインに関係するものです。Enterprise Manager Application Server Control の使いやすい画面やオンライン・ヘルプを使用する方が簡単です。

ただし、様々なフィールドとオプションの意味や JVM コントローラの動作の仕組みを理解するために、Enterprise Manager Application Server Control のユーザーも次の情報を一読することををお勧めします。

7.5.3 新規 JVM コントローラの作成

新規 JVM コントローラの作成後は、[第 7.5.7.1 項「JVM コントローラの起動または再起動」](#)の説明に従って JVM を起動する必要があります。

新規の JVM コントローラを作成するには：

- 「JVM コントローラを作成」をクリックします。
「JVM コントローラを新規作成」ページが表示されます。
- 新規の JVM について、[表 7-2「新規 JVM コントローラの作成オプション」](#)の説明に従って情報を入力します。

表 7-2 新規 JVM コントローラの作成オプション

オプション	説明
名前	JVM の名前を入力します。この名前は、文字で始まり、英数字、 <code>_</code> 、 <code>\$</code> 、または <code>#</code> で構成される有効な Oracle 識別子にする必要があります。Oracle 識別子の長さは 30 バイトです。 ヒント: 名前は、その JVM にアクセスするアプリケーションの名前に基づいたものを入力することをお勧めします。この JVM コントローラの名前は後から変更できません。
JVM 当たりの最大セッション数	新規 JVM が生成される前に該当の JVM が処理する最大同時 Oracle Forms セッション数を指定します。この値によって、デフォルト JVM コントローラ用のセットが上書きされます。
クラスパス	クラスパスを指定すると、環境で指定されたクラスパスまたはシステムのクラスパス、またはデフォルト JVM コントローラ用のクラスパス・セットが上書きされます。
JVM オプション	JVM に渡すために有効なオプションを入力します。この値によって、デフォルト JVM コントローラ用のセットが上書きされます。有効な JVM 起動パラメータの詳細は、Sun 社の Java ドキュメントを参照してください。
ログ・ディレクトリ	デフォルト JVM コントローラでログ・ディレクトリを使用する場合は、「ログ・ディレクトリ」を空白のままにしておきます。他のディレクトリが設定されている場合は、ログ・ファイルは Enterprise Manager で表示できません。
ロギング	「オン」、「オフ」または「デフォルト」です。「オン」または「オフ」に設定すると、デフォルトの JVM ロギング動作よりも優先されます。「デフォルト」に設定すると、新規 JVM はデフォルトのロギング・プロパティをデフォルトの JVM から継承します。
コメント	このテキスト領域に、該当の JVM に関するコメントを追加します。

3. 「適用」をクリックして、JVM をこれらの設定で作成します。
Forms JVM コントローラ・ページが再表示されます。
4. JVM を再起動します。詳細は、[第 7.5.7.1 項「JVM コントローラの起動または再起動」](#)を参照してください。

7.5.4 JVM コントローラの削除

JVM コントローラを削除する前に、コントローラを停止することをお勧めします。JVM コントローラを停止せずに削除すると、JVM コントローラのページから JVM が削除されません。

JVM コントローラを削除するには：

1. ターゲットの JVM コントローラの左にあるラジオ・ボタンをクリックします。
2. 「編集」をクリックします。
「JVM コントローラを編集」ページが表示されます。
3. 「削除」をクリックします。
「確認」ページが表示されます。
4. 「はい」をクリックして削除します。
削除された JVM コントローラのない Forms JVM コントローラ・ページが再表示されます。

7.5.5 Enterprise Manager Application Server Control による JVM コントローラのプロパティの編集

JVM コントローラのプロパティは、HTML ベースのグラフィカル・ユーザー・インタフェースを備えた Enterprise Manager Application Server Control で編集します。

JVM コントローラのプロパティを編集するには：

1. ターゲットの JVM コントローラの左にあるラジオ・ボタンをクリックします。
2. 「編集」をクリックします。
「JVM コントローラを編集」ページが表示されます。
3. 新規の JVM について、表 7-3 「JVM コントローラのプロパティの設定」の説明に従って情報を編集します。

表 7-3 JVM コントローラのプロパティの設定

設定	説明
JVM 当たりの最大セッション数	新規 JVM が生成される前に該当の JVM が処理する最大同時 Oracle Forms セッション数を指定します。この値によって、デフォルト JVM コントローラ用に設定されている他の値が上書きされます。
クラスパス	クラスパスを指定すると、環境で指定されたクラスパスまたはシステムのクラスパス、またはデフォルト JVM コントローラ用のクラスパス・セットが上書きされます。
JVM オプション	JVM に渡すために有効なオプションを入力します。この値によって、デフォルト JVM コントローラ用のセットが上書きされます。有効な JVM 起動パラメータの詳細は、Sun 社の Java ドキュメントを参照してください。
ログ・ディレクトリ	デフォルト JVM コントローラでデフォルトのログ・ディレクトリを使用する場合は、「ログ・ディレクトリ」を空白のままにしておきます。他のディレクトリが設定されている場合は、ログ・ファイルは Enterprise Manager で表示できません。
ロギング	「オン」、「オフ」または「デフォルト」です。「オン」または「オフ」に設定すると、デフォルトの JVM ロギング動作よりも優先されます。「デフォルト」に設定すると、新規 JVM はデフォルトのロギング・プロパティをデフォルトの JVM から継承します。
コメント	このテキスト領域に、該当の JVM に関するコメントを追加します。

4. 「適用」をクリックして、JVM のプロパティ設定をコミットします。
Forms JVM コントローラ・ページが再表示されます。

7.5.6 デフォルト JVM コントローラのプロパティの指定

デフォルト JVM コントローラを使用して、事前定義されたプロパティを新規の JVM コントローラが継承するようにできます。

デフォルト JVM コントローラのプロパティを編集するには：

1. デフォルト JVM コントローラの左にあるラジオ・ボタンをクリックします。
「JVM コントローラを編集」ページが表示されます。
2. デフォルト JVM の次の情報を編集します。

表 7-4 デフォルト JVM コントローラのオプション

オプション	説明
JVM 当たりの最大セッション数	新規 JVM が生成される前にデフォルトの JVM が処理する最大同時 Oracle Forms セッション数を指定します。
クラスパス	クラスパスを指定すると、環境で指定されたクラスパスまたはシステムのクラスパスが上書きされます。
JVM オプション	JVM に渡すために有効なオプションを入力します。有効な JVM 起動パラメータの詳細は、Sun 社の Java ドキュメントを参照してください。
ログ・ディレクトリ	デフォルト JVM コントローラでログ・ディレクトリを使用する場合は、「ログ・ディレクトリ」を空白のままにしておきます。他のディレクトリが設定されている場合は、ログ・ファイルは Enterprise Manager で表示できません。
ロギング	「オン」または「オフ」です。
コメント	このテキスト領域に、該当のデフォルト JVM に関するコメントを追加します。

3. 「適用」をクリックして、デフォルト JVM のプロパティ設定を変更します。

Forms JVM コントローラ・ページが再表示されます。

7.5.7 Enterprise Manager Application Server Control による JVM コントローラの起動と停止

Enterprise Manager Application Server Control は、JVM コントローラの起動、停止、再起動など、Oracle Forms Services を管理するツールです。Oracle Forms Services の管理には、このツールの使用をお勧めします。

7.5.7.1 JVM コントローラの起動または再起動

JVM コントローラが停止した場合は、起動することができます。JVM コントローラがすでに実行中の場合は、手動で停止せずに再起動することができます。停止の手順は Enterprise Manager Application Server Control によって行われます。

実行中でない JVM コントローラを起動するには：

1. ターゲットの JVM コントローラの左にあるラジオ・ボタンをクリックします。
2. 「起動」をクリックします。

JVM コントローラが起動すると、Forms JVM コントローラ・ページ上部に確認のメッセージが表示されます。

実行中の JVM コントローラを再起動するには：

1. ターゲットの JVM コントローラの左にあるラジオ・ボタンをクリックします。
2. 「再起動」をクリックします。
3. 「確認」ページで「はい」をクリックします。

Forms JVM コントローラ・ページが再表示されます。

JVM コントローラが再起動すると、Forms JVM コントローラ・ページ上部に確認のメッセージが表示されます。

JVM コントローラを停止するには：

1. ターゲットの JVM コントローラの左にあるラジオ・ボタンをクリックします。
2. 「停止」をクリックします。
3. 「確認」 ページで「はい」をクリックします。

JVM コントローラが停止すると、Forms JVM コントローラ・ページ上部に確認のメッセージが表示されます。

7.5.8 JVM コントローラの使用コマンド

JVM コントローラには、コマンドと様々なオプションがあります。管理する JVM コントローラの名前を引数として指定する必要があります。

JVM コントローラの実行可能な名前は、`dejvm` です。この名前は、JVM コントローラの起動と停止、およびその実行の管理に使用されます。

コマンドラインの形式は次のとおりです。

```
dejvm -<command> jvmcontroller=<controllerName> [options]
```

ここで、

- *command* は、JVM コントローラに対して発行するコマンドです。
- *controllerName* は、参照する JVM コントローラの名前です。
- *options* は選択したコマンドで使用可能なオプションで、省略できます。

使用可能なコマンドとオプションの詳細は、[第 7.5.10 項「JVM コントローラのコマンドの例」](#)を参照してください。

7.5.8.1 コマンドの制限

コマンドの次の制限に注意してください。

- コマンドには大文字と小文字の区別があります。
- JVM コントローラに対して発行できるコマンドは、一度に 1 つのみです。
- コマンドは、一度に 1 つの JVM コントローラに対してのみ発行できます。

JVM コントローラ (`dejvm` プロセス) に対して使用できるコマンドを次に示します。

Enterprise Manager Application Server Control を使用する場合は、これらのコマンドを発行するためのインタフェースを備えた画面が用意されています。

7.5.8.2 コマンドラインによる JVM コントローラの起動

JVM コントローラを起動するには、`-start` コマンドと、[表 7-5 「start コマンドのパラメータ」](#)に示すパラメータを指定します。

表 7-5 start コマンドのパラメータ

パラメータ	説明
<code>jvmcontroller</code>	<p>コマンド発行対象となる JVM コントローラの名前。また、Forms ランタイム・プロセスは、この名前によってリクエスト送信先の JVM コントローラを識別します。この名前はコンピュータ内で一意である必要がありますが、他のコンピュータ上の別の JVM コントローラでは同じ名前を使用してもかまいません。</p> <p>このパラメータの形式にはファイル名と同じ制限が適用されます。たとえば、「/」、「:」、「*」などの特殊文字を含めることはできません。</p> <p>このパラメータは必須です。</p>

表 7-5 start コマンドのパラメータ (続き)

パラメータ	説明
maxsessions	JVM が子 JVM を作成する前に処理できる Forms ランタイム・プロセスの最大数。maxsessions の値を超えると、JVM コントローラはその JVM コントローラと同じ設定を持つ新しい子 JVM プロセスを自動的に生成します。maxsessions が 65535 の場合は接続数が無制限であることを意味するため、子 JVM は生成されません。 この設定は、特定の JVM が一定数の Forms ランタイム・プロセスを超えて処理を行うとパフォーマンスが低下することが経験や調査によって判明している場合に便利です。 このパラメータはオプションです。デフォルトは 65535 です。
logdir	ログ・ファイルのディレクトリ。ログ・ファイル名は自動的に生成され、<jvm controller>.log になります。<jvm controller> は JVM コントローラの名前です。この値を上書きすると、Enterprise Manager でログを表示できなくなります。デフォルトのログ・ディレクトリを使用する場合は (推奨)、Enterprise Manager でログを表示できます。 このパラメータはオプションです。デフォルトは、ORACLE_HOME/tools/jvm/log です。
classpath	JVM のクラスパス。クラスパスを指定すると、システムのクラスパスは無視され、指定したクラスパスのみが使用されます。 このパラメータはオプションです。デフォルトは、システムのクラスパスまたは現在の環境のクラスパスです。
jvmoptions	指定する JVM オプション。有効な JVM 起動パラメータの詳細は、Sun 社の Java ドキュメントを参照してください。 このパラメータはオプションです。デフォルト値はありません。 このパラメータをコマンドラインで指定する場合は、スペースを含む値は引用符で囲みます。この値を jvmcontrollers.cfg で指定する場合は、値に空白が含まれていても引用符を使用しないでください。
logging	ロギングを ON または OFF で指定します。デフォルトは ON です。

7.5.8.3 JVM コントローラの停止

JVM コントローラを停止するには、-stop コマンドを使用します。停止する JVM コントローラの名前を引数として指定する必要があります。指定した名前の JVM コントローラが実行中ではない場合はエラーが発生します。コマンドに追加するオプションはありません。詳細は、[第 7.5.10 項「JVM コントローラのコマンドの例」](#)を参照してください。

7.5.9 JVM コントローラの構成ファイル

JVM コントローラの構成ファイルは Enterprise Manager によって使用されます。また、管理者によってコマンドラインで利用される場合もあります。構成ファイルの名前とディレクトリは、次のとおりです。

ORACLE_HOME/tools/jvm/jvmcontrollers.cfg

注意： JVM コントローラの構成ファイルのディレクトリや名前を変更することはできません。

この構成ファイルには、名前と値のペア、デフォルト・セクションおよび名前を付けたセクションが含まれ、Forms 構成ファイル (formsweb.cfg) と同様の機能を果たします。jvmcontrollers.cfg に含まれるパラメータは、JVM コントローラの起動パラメータに相当します。

JVM コントローラを起動するときには、設定をコマンドラインから入力するのではなく、構成ファイルから取得するようにできます。このファイルでは、デフォルト・セクションと名前を付けたセクションの両方で、オプションをまったく指定しないことも、一部またはすべてのオプションを指定することもできます。

jvmcontrollers.cfg ファイルの例は次のようになります。

```
#
# This is the default section. These parameters will
# apply unless overridden in a named section (lower down)
# or on the command line.
#
[default]
jvmoptions=-Xms512m -Xmx1024m
maxsessions=50

#
# Below are the named sections.
#
[hrJVM]
jvmoptions=-Xms256m -Xmx512m
classpath=/myJava/hrClasses
```

注意： JVM コントローラで jvmcontrollers.cfg ファイルが使用されるのは、-start コマンドが使用される時のみです。その他のすべてのコマンドでは、jvmcontrollers.cfg ファイルは使用されません。

7.5.9.1 起動オプションの優先順位

この項では、起動オプションが適用される優先順位について説明します。JVM を起動するときには、jvmcontroller パラメータを指定する必要があります。JVM コントローラにより、次の手順が実行されます。

1. JVM コントローラは jvmcontrollers.cfg のデフォルト・セクションを参照し、そこに指定されているすべてのオプションを適用します。
2. JVM コントローラは jvmcontrollers.cfg を参照し、jvmcontroller パラメータに相当する名前の付いたセクションがあるかどうかを確認します。該当のセクションが存在する場合は、指定されているすべてのオプションを取得し、ステップ 1 で見つかっているものがある場合はそれよりも優先します。
3. 次に JVM コントローラはコマンドライン引数を確認します。ここで指定されているオプションは、ステップ 1 およびステップ 2 のオプションより優先されます。

この手順からわかるように、優先順位はコマンドライン・パラメータが最も高く、次いで JVM コントローラ構成ファイルにおける名前を付けたセクション、デフォルト・セクション、デフォルト値またはシステム設定（クラスパスなど）の順になります。

前述の手順で指定されていないコマンドでは、デフォルト値が使用されます。

7.5.10 JVM コントローラのコマンドの例

次に、コマンドラインの例をいくつか示します。jvmcontrollers.cfg ファイルは前述の例と同様のものとします。

- `dejvm -start jvmcontroller=hrJVM`

ID が hrJVM である JVM コントローラを起動します。コントローラ名 hrJVM は、構成ファイルにおける名前を付けたセクションとして定義されています。したがって、JVM オプションとクラスパスのパラメータは構成ファイルから取得されます。maxsessions は、デフォルト・セクションの定義に従って 50 になり、それ以外のパラメータではデフォルト値が使用されます。

- `dejava -start jvmcontroller=myJVM`

IDが `myJVM` である JVM コントローラを起動します。オプションが指定されておらず、名前を付けたセクションが `jvmcontrollers.cfg` に存在しないため、JVM オプションのパラメータはデフォルト・セクションの設定に従って `"-Xms512m -Xmx1024m"` と `maxsessions=50` になります。その他のパラメータではデフォルト値が使用されます。たとえば、`CLASSPATH` の値はシステムの `CLASSPATH` になります。
- `dejava -start jvmcontroller=hrJVM jvmoptions="-Xms128m -Xmx256m" maxsessions=75`

名前を付けたセクションの定義に従って、クラスパスを `/myJava/hrClasses` に設定します。コマンドラインは `jvmcontrollers.cfg` ファイルよりも優先されるため、JVM オプションは `"-Xms128m -Xmx256m"` になります。同様に、`maxsessions` は 75 になります。その他のすべてのパラメータではデフォルト値が使用されます。
- `dejava -start jvmcontroller=myJVM maxsessions=100 classpath=/myJava/myClasses;/moreJava/moreClasses`

コントローラは、`jvmcontrollers.cfg` のデフォルト・セクションの定義に従って `jvmoptions="-Xms512m -Xmx1024m"` を使用します。`maxsessions` はデフォルト・セクションより優先されて 100 となり、クラスパスは `/myJava/myClasses;/moreJava/moreClasses` になります。その他のすべてのパラメータではデフォルト値が使用されます。
- `dejava -stop jvmcontroller=hrJVM`

`hrJVM` コントローラを停止します。このコマンドを正常に実行するには、コントローラがすでに起動されている必要があります。

7.5.11 Forms 構成ファイル設定

この項では、Forms 構成ファイル (`formswb.cfg`) で使用される JVM プーリング・パラメータについて説明します。パラメータ名には大文字と小文字の区別はありません。Forms 構成ファイルは Enterprise Manager を使用して管理できます。表 7-6 「Oracle Forms の JVM コントローラ起動パラメータ」は、`formswb.cfg` で指定できる起動オプションを示しています。

表 7-6 Oracle Forms の JVM コントローラ起動パラメータ

パラメータ	説明
<code>jvmcontroller</code>	<p>有効値の詳細は、第 7.5.8.2 項「コマンドラインによる JVM コントローラの起動」を参照してください。また、JVM の指定は省略できます。</p> <p>デフォルト値はありません。</p> <p>このパラメータはデフォルト・セクションでグローバルに設定することも、アプリケーション・セクションを優先することもできます。これによって、Forms ランタイム・プロセスで使用する JVM コントローラが決まります。<code>dejava</code> 実行可能ファイルの <code>jvmcontroller</code> パラメータに対応します。</p> <p><code>jvmcontroller</code> に値が指定されていない場合は、Oracle Forms ランタイム・プロセスは自身のインプロセス JVM を起動し、Java Importer は 10g より前のリリースの動作をします。</p>

7.5.12 起動の例

次のコードは formsweb.cfg ファイルからの抜粋で、起動のフローを示しています。

```
# System settings
[default]
jvmcontroller=commonJVM

[ordersApp]
form=orders.fmx
userid=orders/orderspw@orcl

[hrApp]
form=hr.fmx
userid=hr/hrpw@orcl
jvmcontroller=hrJVM

[salesApp]
form=sales.fmx
userid=sales/salespw@orcl
```

ユーザーが ordersApp アプリケーションを起動し、アプリケーションが Java コードを実行すると、Oracle Forms ランタイム・プロセスはそのリクエストを commonJVM という名前の JVM コントローラにルーティングします。[ordersApp] アプリケーション・セクションでは使用する JVM コントローラが指定されていないので、Oracle Forms ランタイム・プロセスはグローバルな JVM コントローラを使用します。JVM コントローラが起動していない場合は、動的に起動します。別のユーザーが同じアプリケーションを起動すると、同様に commonJVM に接続します。

ユーザーが hrApp アプリケーションを起動し、アプリケーションが Java コードを実行すると、Oracle Forms ランタイム・プロセスはそのリクエストを hrJVM という名前の JVM コントローラに送信します。これは、[hrApp] アプリケーション・セクションがグローバルな設定より優先されるためです。JVM コントローラが起動していない場合は、動的に起動します。別のユーザーが同じアプリケーションを起動すると、同様に hrJVM に接続されます。

ユーザーが salesApp アプリケーションを起動し、アプリケーションが Java コードを実行すると、Oracle Forms ランタイム・プロセスはインプロセス JVM を起動します。これは、Java Importer が JVM プーリングを使用せずに動作する場合と同じです。別のユーザーが同じアプリケーションを起動すると、アプリケーションは自身のインプロセス JVM を起動するため、[図 7-4](#) に示すようにメモリーの消費量が多くなります。

図 7-4 複数のアプリケーションによる複数 JVM

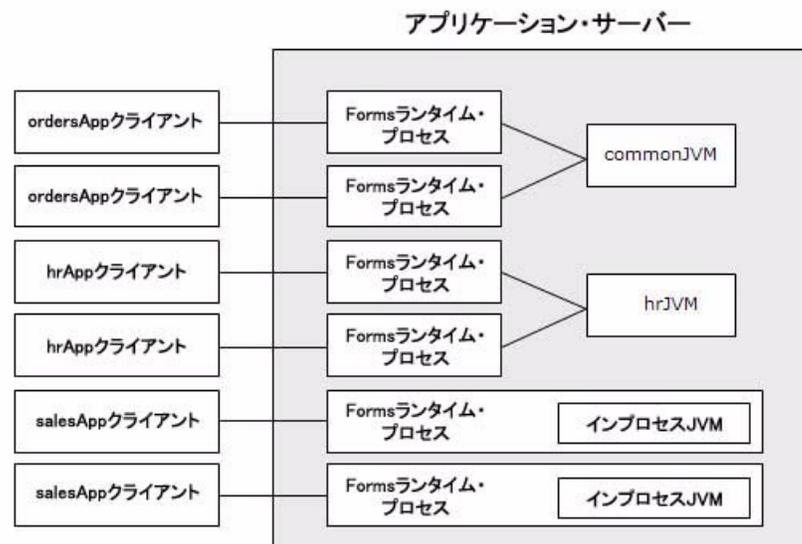


図 7-4 では、commonJVM コントローラとそのインプロセス JVM、およびそのすべての子 JVM が 1 つの枠に示されています。hrJVMcontroller の場合も同様です。

7.5.13 複数の JVM コントローラについて

JVM プーリングのアーキテクチャでは、複数の JVM コントローラを持ち、それぞれが子 JVM を持つことができます。複数の JVM コントローラを使用するのは、次のような場合です。

- アプリケーションごとに JVM コントローラを持ち、他のコントローラとは個別に起動および停止できるようにする場合。
- アプリケーションごとに異なる設定が必要な場合。たとえば、複数のコントローラでクラスパスや JVM の設定を分ける場合です。
- Enterprise Manager から JVM コントローラのリソース使用率を監視する場合。複数のアプリケーションまたはユーザー・グループで異なる JVM コントローラが使用されている場合は、Java Importer コードによってどのようにリソースが消費されているかを確認できます。
- 同一コンピュータ上に複数の開発環境、テスト環境または本番環境がある場合。
- 複数の異なるアプリケーションで静的データを共有しない場合。

7.5.14 子 JVM について

JVM のパフォーマンスが大幅に低下した場合は、その JVM が処理するリクエストが多すぎる可能性があります。その場合は、その JVM コントローラで複数の子 JVM を持つようにできます。子 JVM は必要に応じて動的に作成されます。

JVM パラメータの `maxsessions` は、新しい子 JVM が作成されるまでに 1 つの JVM に対して接続できる Oracle Forms ランタイム・プロセスの数を指定します。子 JVM が起動すると、その子 JVM は JVM コントローラと同じパラメータを継承します。

JVM が `maxsessions` で指定された接続数に達すると、その JVM は新しい Oracle Forms ランタイム・プロセスからのリクエストを受け付けなくなります。新しい Oracle Forms ランタイム・プロセスが初めて Java コードを実行しようとする、そのプロセスは使用可能な JVM (`maxsessions` より少ない接続数のもの) に接続されます。JVM の選択方法は完全に任意で、ロード・バランシングやラウンドロビン・アルゴリズムはありません。

JVM が `maxsessions` で指定された接続数に達しても、別の JVM がそれに達していなければ、新しい JVM は作成されません。すべての JVM が同時に `maxsessions` で指定された接続数に達した場合は、別の子 JVM が作成されます。

子 JVM は負荷が低下しても自動的に削除されません。そのため、一部の子 JVM を削除する場合は、JVM コントローラを停止する必要があります。これにより、すべての子 JVM も停止します。その後、JVM コントローラを再起動できます。

子 JVM の有効範囲は、JVM コントローラのネームスペースのコンテキスト内に限定されます。たとえば、2 つの JVM コントローラ、`ordersJVM` および `hrJVM` がある場合、`ordersJVM` とその子 JVM は、`hrJVM` またはその子 JVM に影響を与えることも、これらから影響を受けることもありません。

7.5.14.1 子 JVM の例

`ordersJVM` という JVM コントローラが `maxsessions=50` に設定されているとします。実行中の各オーダー・アプリケーションは、`ordersJVM` にリクエストを送信します。新しい Oracle Forms ランタイム・プロセスが `ordersJVM` にリクエストを送信すると、そのたびに Oracle Forms ランタイム・プロセスと通信する新しいスレッドが作成されます。その後、JVM コントローラは新しいリクエストのリスニングに戻ります。ユーザーがセッションを終了すると、JVM 内のスレッドも終了します。

ordersJVM コントローラが 50 番目の同時リクエストを受信すると（一部のユーザーが終了した後で開始するユーザーも存在するため、最初から 50 番目のユーザーとは限りません）、コントローラは子 JVM を生成します。子 JVM は親の設定を継承するため、この子 JVM の maxsessions も 50 になります。この段階では、JVM コントローラには 50 の接続があり、子 JVM には接続はありません。

新しいユーザーがこの Oracle Forms アプリケーションを起動し、Java コードを実行すると、Oracle Forms ランタイム・プロセスはその JVM コントローラのネームスペース内でリスニングを実行している JVM に接続します。この JVM コントローラには 50 の接続があり、使用できないため、子 JVM がこのリクエストを受信します。その後、親の JVM コントローラで一部のユーザーがアプリケーションを終了し、接続数が減少すると、コントローラは maxsessions で指定された接続数に達しないかぎり新しいリクエストを受信することができます。

このような処理が行われる間、hrJVM はこれとは別に動作しています。ordersJVM で接続が超過しても hrJVM には接続されず、ordersJVM の子 JVM に対してのみ接続が行われます。

7.6 JVM コントローラのロギング管理

ロギングを有効にすると、JVM コントローラは次のような情報をログ・ファイルに記録します。

- JVM パラメータ（maxsessions、classpath など）の値。
- JVM コントローラが起動および終了した日時。
- 子 JVM が生成された日時。
- Oracle Forms ランタイム・プロセスが新規の接続を開始した日時とそのプロセス ID。
この情報は、診断または管理の目的でどの Oracle Forms ランタイム・プロセスがどの JVM コントローラに接続されているかを調べるうえで役立ちます。
- Oracle Forms ランタイム・プロセスのセッションが終了し JVM との接続が解除された日時。

7.6.1 ロギングの有効化と無効化

この項は、次の項目に分かれています。

- [第 7.6.1.1 項「デフォルトのロギング・プロパティの指定」](#)
- [第 7.6.1.2 項「ログ・ファイルのディレクトリの指定」](#)
- [第 7.6.1.3 項「ログ・ファイルへのアクセス」](#)
- [第 7.6.1.4 項「JVM コントローラのログ・ファイルの削除」](#)

7.6.1.1 デフォルトのロギング・プロパティの指定

1. デフォルトの JVM コントローラを選択します。
2. 「**編集**」をクリックします。
3. 「JVM コントローラを編集」ページが表示されます。
4. 「ロギング」ドロップダウン・ボックスで「オン」または「オフ」を選択します。
5. 「**適用**」をクリックします。

Forms JVM コントローラ・ページが再表示されます。

7.6.1.2 ログ・ファイルのディレクトリの指定

ログ・ファイルのディレクトリは、JVM コントローラを作成または編集するときに指定できます。また、他の JVM コントローラが使用するデフォルト JVM コントローラのログ・ファイル・ディレクトリも指定できます。

JVM コントローラを作成または編集するときにログ・ファイルのディレクトリを指定するには：

1. 新規のコントローラを作成する場合は「**JVM コントローラを作成**」をクリックします。それ以外は、ターゲット JVM コントローラの横にあるラジオ・ボタンを選択します。

「JVM コントローラを新規作成」ページまたは「JVM コントローラを編集」ページが表示されます。

2. JVM のログ・ファイルのディレクトリに関する次の情報を入力します。

ログ・ディレクトリ

デフォルト JVM コントローラでデフォルトのログ・ディレクトリを使用する場合は、「**ログ・ディレクトリ**」を空白のままにしておきます。他のディレクトリが設定されている場合は、ログ・ファイルは Enterprise Manager で表示できません。

3. 「**適用**」をクリックして、JVM をこれらの設定で作成します。新しい JVM が自動的に起動します。

Forms JVM コントローラ・ページが再表示されます。

7.6.1.3 ログ・ファイルへのアクセス

JVM コントローラのロギングが有効になっている場合は、「ログ・ファイル」列をクリック可能なアイコンが表示されます。ロギングが無効な場合は、その JVM コントローラのログ・ファイルのエントリは空になります。詳細は、[第 7.6.1 項「ロギングの有効化と無効化」](#)を参照してください。

ログ・ファイルにアクセスするには：

- 該当の JVM コントローラで使用可能な「ログ・ファイル」列の「ログ・ファイル」アイコンをクリックします。

「ログ・ファイル」ページに、[表 7-7](#) に示す情報が表示されます。

表 7-7 「ログ・ファイル」 ページの情報

項目	説明
ログ・ファイル	選択した JVM コントローラの名前が入っています。
コンポーネント名	JVM コントローラの名前が表示されます。
コンポーネント・タイプ	アプリケーション・タイプが表示されます。
変更済	ログ・ファイルが最後に更新された時刻が表示されます。
サイズ	ログ・ファイルのサイズがバイト単位で表示されます。
ログ・ファイルの内容	最新のログ・エントリを含む表です。最大 2000 行が取得されます。
ログ・テキスト	ログ・ファイルの最新のエントリが表示されます。

7.6.1.4 JVM コントローラのログ・ファイルの削除

JVM コントローラのログ・ファイルを削除するには：

1. 「JVM コントローラ」 ページで、ターゲット JVM の横にあるラジオ・ボタンを選択します。
2. 「ログファイルを削除」 をクリックします。
削除の確認ページが表示されます。
3. 「はい」 をクリックします。
「JVM コントローラ」 ページが再表示されます。

7.7 JVM プーリングのエラー・メッセージ

PDE-JM001: JVM コントローラ : <jvm_name> と通信できません。

原因: JVM コントローラの起動または既存の JVM コントローラへの接続に失敗しました。

処置: 管理者に連絡してください。

8

トレースと診断

この章には、次の項が含まれています。

- 第 8.1 項 「Forms Trace について」
- 第 8.2 項 「Forms Trace の構成」
- 第 8.3 項 「Forms Trace の開始」
- 第 8.4 項 「Forms Trace 出力の表示」
- 第 8.5 項 「トレース可能イベントのリスト」
- 第 8.6 項 「Forms Services のトレース・メトリックの監視」
- 第 8.7 項 「Servlet Logging Tools」

8.1 Forms Trace について

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。Oracle Enterprise Manager Application Server Control コンソールから、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。

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) の機能に代わるものです。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。

8.2 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 10g Application Server Control コンソール 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、`ftrace.cfg` ファイルの変更がすべて管理されます。

Forms Trace を使用する際には、次のことに注意してください。

- 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。
- Windows 上で Forms プロセスをトレースするには、Process Manager 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、トレースを有効にすると「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。」というエラーが発生します。
- `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 10g Application Server Control コンソールでこれらのファイルを編集する必要があります。
- 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 10g Application Server Control コンソールを使用して 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。
- 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

イベントとそれに対応するイベント番号は、第 8.5 項「トレース可能イベントのリスト」を参照してください。

Forms Trace を構成するには：

1. Oracle Enterprise Manager Application Server Control コンソールを起動します。
2. Oracle Enterprise Manager Application Server Control コンソールのメイン・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。

図 8-1 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 8-1 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。

8.2.1 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 8-1 では、パラメータの値を説明しています。

表 8-1 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"> ■ <code>Tracegroup</code> を指定していない場合、エラー・メッセージのみが収集されます。 ■ Forms Trace をコマンドラインで起動しない場合、<code>Tracegroup</code> は無視されます。 ■ <code>Tracegroup</code> 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。 <code>Tracegroup=<keyword></code>。この <code><keyword></code> は、<code>ftrace.cfg</code> で指定します (たとえば、<code>Tracegroup=MyEvents</code>)。 <p>これで、名前を付けたセット <code>SQLInfo</code> のイベントをログ出力できます。</p> <ul style="list-style-type: none"> ■ <code>Tracegroup</code> 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。 <code>Tracegroup = 0-3</code> これで、<code>0<= イベント <=3</code> 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。 ■ <code>Tracegroup</code> 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。 <code>Tracegroup = 34,67</code> ■ <code>Tracegroup</code> キーワードを使用してイベント・セットを組み合わせることができます。たとえば、次のように指定します。 <code>Tracegroup = 0-3,34,67,SQLInfo</code>

表 8-1 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。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 ORACLE_HOME/forms/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ます (例: forms_<pid>.trc)。</p>

8.3 Forms Trace 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 コンソールを使用して、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/forms/frmservlet?form=cxl&record=forms&tracegroup=0-199
http://cx-pc/forms/frmservlet?form=cxl&record=forms&tracegroup=mysql
http://cx-pc/forms/frmservlet?form=cxl&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る予定です。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による Oracle Forms の最新情報は、OTN (<http://www.oracle.com/technology/products/forms/>) から入手できます。

8.4 Forms Trace 出力の表示

管理者権限を持つユーザーのみが、トレース・ログ・ファイルを参照できます。ユーザーは、管理者グループのユーザーとしてログインする必要があります。デフォルトでは、管理者グループのデフォルトのユーザーであるユーザー `admin` を使用します。その後、このグループにユーザーを作成および追加して、それらのユーザーがトレース・ログ・ファイルにログを記述したり表示できるようにします。

ユーザーは一度ログインすると、異なる formsweb セッションのトレース・ログ・ファイルを表示するために同じブラウザ・セッションに再度ログインする必要がなくなります。

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 コンソール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： Enterprise Manager Application Server Control コンソールの「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。

Oracle Enterprise Manager を使用して、トレース・データを表示するには：

1. Oracle Enterprise Manager Application Server Control コンソールで、「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。

8.4.1 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 形式または HTML 形式に変換されます。そのためには、WriteOut または WriteOutHTML の 2 つの有効値のいずれかを指定して、パラメータ OutputClass を追加する必要があります。WriteOut を使用すると、出力ファイルは XML 形式になります。WriteOutHTML を使用すると、出力ファイルは HTML 形式になります。

これらの 2 つの値 (WriteOut および WriteOutHTML) では、大文字と小文字は区別されます。

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
outputfile=myfile.xml outputclass=WriteOut
```

これにより、myfile.xml が作成されます。

トレース・データを HTML 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
outputfile=myfile.html outputclass=WriteOutHTML
```

8.5 トレース可能イベントのリスト

表 8-2 「トレース可能イベントのリスト」は、トレースに定義できるイベントを示しています。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 8-2 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

表 8-2 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
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
64	フォーム (開始と終了)	期間
65	プロシージャ (開始と終了)。今後のリリースで 使用可能になる予定。	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)。今後のリリースで使用 可能になる予定。	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

表 8-2 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
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
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。今後のリリースで使用可能になる予定。	該当せず
197*	関数の引数。今後のリリースで使用可能になる予定。	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。

8.5.1 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。

- [表 8-3 「ユーザー・アクション・イベントの詳細」](#)
- [表 8-4 「Forms Services のイベントの詳細」](#)
- [表 8-5 「イベントの詳細」](#)
- [表 8-6 「3 層イベントの詳細」](#)
- [表 8-7 「その他のイベントの詳細」](#)

8.5.1.1 ユーザー・アクション・イベント

表 8-3 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

表 8-3 ユーザー・アクション・イベントの詳細 (続き)

アクション	詳細	番号
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

8.5.1.2 Forms Services のイベント

表 8-4 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。今後のリリースで使用可能になる予定。	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

8.5.1.3 イベントの詳細

表 8-5 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。今後のリリースで使用可能になる予定。	100
問合せの実行	ブロック名	101

8.5.1.4 3 層イベント

表 8-6 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

表 8-6 3 層イベントの詳細 (続き)

イベント名	詳細	番号
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

8.5.1.5 その他のイベント

表 8-7 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)。今後のリリースで使用可能になる予定。	196
関数の引数	ファンクションに渡される引数 (イン / アウト)。今後のリリースで使用可能になる予定。	197

8.6 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 コンソールを起動します。
2. Oracle Enterprise Manager Application Server Control コンソールのメイン・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。

8.7 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。Oracle Application Server Forms Services に備わっている Servlet Logging Tools には、次の機能があります。

- 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)
- サイトの構成に関するデバッグ情報の生成 (デバッグ・レベルのロギング)

Servlet Logging Tools に関する情報は、次の各項を参照してください。

- 第 8.7.1 項「ロギングの有効化」
- 第 8.7.2 項「ログ・ファイルのディレクトリ」
- 第 8.7.3 項「各レベルのサーブレット・ロギングの出力例」

8.7.1 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 8-8「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 8-8「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 コンソールの「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、第 8.7.3 項「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 8-8 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コンピュータ）のホスト名と IP アドレス、ランタイム・プロセス ID、および一意の内部セッション ID 番号が含まれます。
/sessionperf	セッション終了メッセージにパフォーマンス・サマリー統計が含まれません。
/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。

8.7.1.1 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/forms/frmservlet?serverURL=/forms/lervlet/perf
```

8.7.1.2 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=/forms/frmservlet/session
```

8.7.1.3 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グ・メッセージをログ・ファイルにも出力します。

```
http://yourserver/forms/frmservlet/debug?serverURL=/forms/lervlet/
debug
```

8.7.2 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/formsapp 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/formsapp/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
```

8.7.3 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

8.7.3.1 (なし)

```
FormsServlet init():
configFileName:    d:¥Oracle/forms/server/formsweb.cfg
testMode:         false
```

8.7.3.2 /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
```

8.7.3.3 /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
```

8.7.3.4 /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
```

8.7.3.5 /debug

http://test-machine:8888/forms/frmservlet/debug&config=myapp&serverURL=/forms/lservlet/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=myapp&serverURL=/forms/lservlet/debug
No current servlet session
File baseie.htm not found, looking in d:¥Oracle/forms/server
The SSO_USERID is: null
===== FormsServlet =====
GET request received, cmd=startsession, qstring=config=myapp&serverURL=
/forms/lservlet/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:¥Oracle/forms/server
envFile = d:¥Oracle¥forms¥server¥default.env
serverURL: /forms/lservlet/debug
rewrittenURL: /forms/lservlet/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 frmweb server webfile=HTTP-0,0,1
Getting stdin, stdout and stderr of child process
Writing working directory to stdin: d:¥Oracle¥forms
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.
*****
```

エンド・ユーザー監視の構成

この章には、次の項が含まれています。

- 第9.1項「エンド・ユーザー監視について」
- 第9.2項「エンド・ユーザー監視の構成」
- 第9.3項「エンド・ユーザー監視の有効化」
- 第9.4項「その他の情報」

9.1 エンド・ユーザー監視について

エンド・ユーザー監視は、Oracle Enterprise Manager のユーティリティの1つです。開発者や管理者は、これを使用してアプリケーションのパフォーマンスを監視できます。ライブ・データのレポート時のシステム・リソースに対するオーバーヘッドがわずかであるため、システム管理者は、他の特別な追加手順を実行することなく、ライブ・アプリケーションのパフォーマンスを正確に監視できます。

エンド・ユーザー監視では、レポートおよびパフォーマンス・グラフが提供され、またシステムがアラートを生成します。Forms の既存の監視メカニズムとは異なり、Oracle Forms Runtime プロセス内の処理時間だけでなく、ユーザーのコンピュータでの遅延も記録されます。

エンド・ユーザー監視からレポートされるデータは定性的です。これらは異なるドメインまたはユーザー・グループの傾向分析に使用したり、比較レポートとして使用できます。たとえば、特定のフォームを開くのに通常2秒かかるところで、突然10秒かかるようになった場合、ユーザー側にこれに相当するパフォーマンスの低下が生じていることを予測できます。ただし、これはストップウォッチを手にして計測すると、操作にちょうど10秒かかるという意味ではありません。同様に、特定の問合せに通常3秒のデータベース時間がかかるところで、急に30秒になった場合、同様のパフォーマンスの低下がユーザー側に生じます。

しかし、同時に、エンド・ユーザー監視はデバッグ・ツールではないということを念頭に入れておく必要があります。データを分析すればどの部分をさらに調査する必要があるかを特定することはできますが、ツール自体からは原因を特定できません。

9.2 エンド・ユーザー監視の構成

以降の項では、Oracle Forms Services およびエンド・ユーザー監視の構成手順について説明します。

- 第 9.2.1 項「[エンド・ユーザー監視の使用要件](#)」
- 第 9.2.2 項「[エンド・ユーザー監視を使用する Web Cache の構成](#)」
- 第 9.2.3 項「[Enterprise Manager Grid Control で監視する Web Cache インスタンスの指定](#)」
- 第 9.2.4 項「[デフォルトの最小ヒット数のしきい値の変更](#)」
- 第 9.2.5 項「[不適切なレスポンス時間を排除する設定の変更](#)」

9.2.1 エンド・ユーザー監視の使用要件

エンド・ユーザー監視を Oracle Forms Services で使用するには、次をインストールおよび構成する必要があります。

- Oracle Forms Services を含む Oracle Application Server 10g リリース 2 (10.1.2)
- Oracle Enterprise Manager Grid Control へのアクセスまたはそのインストール
Enterprise Manager Grid Control インスタンスがあるコンピュータの名前、構成ポート、パスワードが必要です。
- Enterprise Manager Grid Control の一部である Oracle Management Agent
Oracle Management Agent は、監視対象の各ホストに配置されるプロセスです。Oracle Management Agent は、ホスト上のすべてのターゲットを監視し、中間層の管理サービスにその情報を通信し、ホストとターゲットを管理および維持します。

Oracle Management Agent は Oracle Application Server の中間層にインストールします。インストール時に、Enterprise Manager Grid Control を実行するコンピュータ、その構成ポート番号およびパスワードを指定する必要があります。

9.2.2 エンド・ユーザー監視を使用する Web Cache の構成

Web Cache は (Forms ランタイム・プロセスを含む) Oracle Application Server と Oracle Management Agent の間で仲介役を果たします。次の手順に従って、エンド・ユーザー監視を使用するための特定の Web Cache インスタンスを構成します。

エンド・ユーザー監視を使用する Web Cache を構成するには：

1. Oracle Application Server の「Web Cache Admin」ページを開きます (デフォルトのポートは 9400)。
2. 「Web Cache Admin」をクリックして、Web Cache の管理者としてログインします。
3. 「Logging and Diagnostics」で「End-User Performance Monitoring」をクリックします。
4. 「Cache-Specific End-User Performance Monitoring」表からターゲットの Web Cache の横のラジオ・ボタンを選択して、「Enable」をクリックします。
5. 「Site-Specific End-User Performance Monitoring」表から、Oracle Monitoring Agent が位置する中間層の横のラジオ・ボタンを選択して、「Enable」をクリックします。
6. 「Logging and Diagnostics」で「Access Logs」をクリックします。
7. access_log の形式スタイルを「End-User Performance Monitoring Format」に変更します。
8. 「Apply Changes」をクリックして、Web Cache を再起動します。
9. Web ブラウザでターゲットの Oracle Application Server にアクセスして、Web ページのソースをブラウザで開きます。

HTML ページの最後に `<SCRIPT SRC="/oracle_smp_EndUserMonitoring/oracle_smp_EndUserMonitoring.js"></SCRIPT>` が表示される場合、エンド・ユーザー管理は有効になっています。

以降の項では、Enterprise Manager Grid Control の監視対象としてこの Web Cache インスタンスを選択し、Forms Services を構成してアプリケーションを監視させます。

9.2.3 Enterprise Manager Grid Control で監視する Web Cache インスタンスの指定

構成のこの部分では、Enterprise Manager Grid Control で監視する Web Cache インスタンスを指定します。この Web Cache インスタンスに対しては、エンド・ユーザー監視を有効にする必要があります。

Enterprise Manager Grid Control で監視する Web Cache インスタンスを追加するには：

1. `https://em.computer.company.com/em` を開きます。
2. ユーザー名とパスワードを入力してログインします。
3. 「ターゲット」 → 「Web アプリケーション」 → 「追加」を選択します。
4. `http://computer.company.com/forms/frmservlet` など、ホーム・ページ URL の名前と Forms URL を入力し、「次へ」をクリックします。
5. 「使用可能なターゲット」で「Forms」を選択して、インスタンス名を選択し、「次へ」をクリックします。
6. エージェントのポート (1831 など) を選択して、「次へ」、「終了」の順でクリックします。ステータスの表示には数分かかることがあります。
7. ステータスが表示されたら、「ページ・パフォーマンス」に進み、「Web アプリケーションの Web キャッシュの構成」をクリックします。
8. 「Web アプリケーションの Web キャッシュの構成」で、「Web アプリケーション・コンポーネントの追加または削除」をクリックします。
9. 「使用可能なターゲット」から「Web Cache」を選択して、Web Cache のターゲットを選択し、「次へ」、「適用」の順にクリックします。

10. 「**収集**」を選択して、間隔を1分に設定します。
11. 「**適用**」をクリックします。

9.2.4 デフォルトの最小ヒット数のしきい値の変更

デフォルトの最小ヒット数しきい値の設定を変更すると、指定した最小ヒット数に基づき、データの更新を大幅に高速化できます。ヒット数を低く設定すると、ヒット数に達するたびにデータは更新されるので、データは頻繁に更新されます。ヒット数を高く設定すると、指定した高いヒット数に達したときにデータは更新されます。

デフォルトの最小ヒット数のしきい値を変更するには：

- Enterprise Manager データベースに対して SQL スクリプトを実行します。

```
update mgmt_parameters set parameter_value = 1 where parameter_name = 'mgmt_rt_min_hits';
commit;
```

9.2.5 不適切なレスポンス時間を排除する設定の変更

不適切なしきい値はデフォルトで 60,000 ミリ秒に設定されており、この値は Oracle Forms アプリケーションでは短かすぎる場合があります。必要に応じて、このデフォルト値を 1 時間に変更します。

不適切なレスポンス時間の排除を変更するには：

- Enterprise Manager データベースに対して SQL スクリプトを実行します。

```
update mgmt_parameters set parameter_value = 3600000 where parameter_name = 'mgmt_rt_max_elapsed_time';
commit;
```

9.3 エンド・ユーザー監視の有効化

エンド・ユーザー監視の構成後、formsweb.cfg に新しいセクションを作成し、エンド・ユーザー監視を有効にしてアプリケーションを監視できます。

9.3.1 formsweb.cfg の変更

特定のアプリケーションを監視するには、formsweb.cfg に新しいセクションを作成します。formsweb.cfg に特定のセクションを作成する方法の詳細は、[第 4.3.2 項「構成セクションの管理」](#)を参照してください。

1. ORACLE_HOME/forms/server/formsweb.cfg を開きます。
2. EndUserMonitoringEnabled=true を設定します。
3. EndUserMonitoringURL=http://computername:7777/oracle_smp_EndUserMonitoring/oracle_smp_EndUserMonitoring_sdk.gif を設定します。

注意： コンピュータ名は、Web Cache が実行されている中間層インストールです。

9.4 その他の情報

エンド・ユーザー監視に関する追加情報およびメトリックの解析方法の詳細は、次を参照してください。

- Enterprise Manager オンライン・ヘルプのメトリックに関するページ
- Oracle Application Server のドキュメント CD に収録されている『Oracle Enterprise Manager 概要』

10

パフォーマンス・チューニングに関する考慮事項

この章には、次の項が含まれています。

- 第 10.1 項「Forms Services の組み込み最適化機能」
- 第 10.2 項「OracleAS Forms Services アプリケーションのチューニング」
- 第 10.3 項「Web Cache と Forms の統合」

Oracle Application Server Forms Services と Oracle Database Server 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。

10.1 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。

- 第 10.1.1 項「Forms Services の監視」
- 第 10.1.2 項「Forms Services Web ランタイム・プーリング」
- 第 10.1.3 項「Forms Services のユーティリティ」
- 第 10.1.4 項「クライアント・リソース要件の最小化」
- 第 10.1.5 項「Forms Services リソース要件の最小化」
- 第 10.1.6 項「ネットワーク使用量の最小化」
- 第 10.1.7 項「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」
- 第 10.1.8 項「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」

10.1.1 Forms Services の監視

Oracle Enterprise Manager 10g Application Server Control コンソールを使用して 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

10.1.1.1 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 コンソールを起動します。
2. Oracle Enterprise Manager Application Server Control コンソールのメイン・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。
- レスポンス時間。これは、Enterprise Manager ページがロードされるときの、Forms Enterprise Management エージェントから Forms Servlet への ping 時間です。

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

10.1.1.2 Forms イベントの監視

Oracle Enterprise Manager Application Server Control コンソールを使用して、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。

10.1.1.3 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 コンソールを起動します。
2. Oracle Enterprise Manager Application Server Control コンソールのメイン・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・コンピュータの IP アドレス。
- データベースのユーザー名: 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。
- 構成セクション: 特定の Forms セッションで使用される構成セクションの「編集セクション」ページを開きます。

10.1.1.4 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。

10.1.1.5 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（大文字と小文字の区別あり）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。

10.1.2 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・エンジンを実前に起動できます。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、サーバー構成の許容量が小さく、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。

10.1.2.1 事前起動パラメータの構成

Oracle Enterprise Manager Application Server Control コンソールを使用して、表 10-1 に示すパラメータで Forms Services のランタイム・プーリングを構成します。

表 10-1 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
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

prestartMin は、特定のアプリケーションに常時存在する必要がある、事前起動済のランタイム数です。最小値は、prestartInit パラメータに定義した値と同等またはそれ未満に設定してください。prestartMin パラメータは、いつでも変更可能で、アプリケーション・サーバーの再起動を必要としません。クライアントが事前起動済のランタイム・プロセスへの接続をリクエストし、このランタイム・プロセスがタイムアウトしていない場合、この新しいエントリが取得されます。プロセスがタイムアウトすると、アプリケーションはデフォルトの動作になり、最小のしきい値は維持されません。

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。

10.1.2.2 ランタイム・プーリングの起動

管理者は Oracle Enterprise Manager Application Server Control コンソールを使用して、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。

10.1.3 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、中間層での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。

10.1.3.1 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。

10.1.4 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て済みます。

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ルック・アンド・フィールの変更などの、機能サブコンポーネントにグループ化されます。機能サブコンポーネントを使用すると、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。

10.1.5 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。

10.1.6 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。

10.1.7 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レスポンス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、それらを単一のパケットとして 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。

10.1.8 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) ツリーの一部分です。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パケット数を最小化することが必要です。

10.2 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。

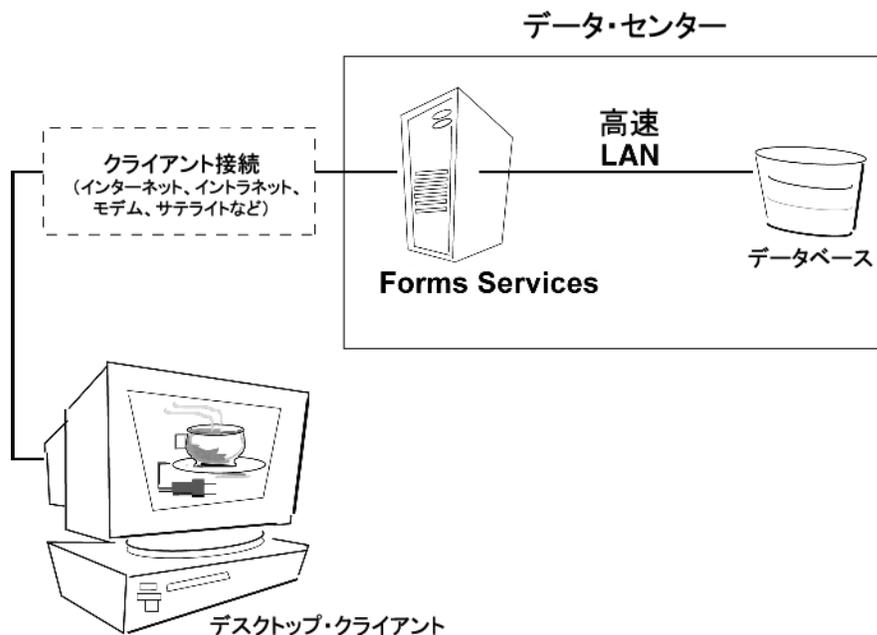
10.2.1 データ・サーバーに対する Oracle Application Server Forms Services の位置

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 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 10-1 の構成では、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 10-1 OracleAS Forms Services とデータベース・サーバーの並存



10.2.2 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、オーバーヘッドと見なされます。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](#)

10.2.2.1 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。

10.2.2.1.1 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。

- `frmall.jar` : 必要なクラスがすべて含まれています。
- `frmall_jinit.jar` : `frmall.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 (デフォルト)。
- `frmmain.jar` : `frmall.jar` よりもクラスの数が増えています。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=frmall_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、frmmain.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 frmall.jar と frmall_jinit.jar 内には存在するため、frmmain.jar を利用する場合のみ使用します。

- frmoracle_laf.jar: Oracle のルック・アンド・フィールのクラス
- frmgeneric_laf.jar: 汎用 (標準) ルック・アンド・フィールのクラス
- frmresources.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 frmall.jar、frmall_jinit.jar および frmmain.jar に含まれています。frmresources.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せん。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 B 「JInitiator」](#) を参照してください。

10.2.2.1.2 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。

- frmall.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=frmall.jar
```

10.2.2.2 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。

10.2.3 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ず。
 - 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うえに）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及ぶ場合でも、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。次のキャンバス・プロパティを使用します。

エントリでレイズ = はい（キャンバスのみ）

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。

10.2.4 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。タイマーの多くは時計やアニメーションを提供するために使用されます。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 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。ログイン・フォームなど、初期フォームを再度コールしない場合以外は、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、ユーザーのコンピュータ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。

10.3 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. ホスト B 上にある Oracle HTTP Server 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。
3. ホスト C 上にある Oracle HTTP Server 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。

Oracle HTTP Server 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 Oracle Application Server クラスタリングを利用できないため、Oracle HTTP Server 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 Oracle HTTP Server 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: /forms/frmservlet?ifcmd=status)。

Web Cache でセッション・バインドを構成するには：

1. 次のコードを \$ORACLE_HOME¥2ee¥OC4J_BI_Forms¥application-deployments¥formsapp¥formsweb¥orion-web.xml にある orion-web.xml ファイルに追加します。

```
<session-tracking
  cookies="enabled">
</session-tracking>
```
2. 次のコマンドを発行します。

```
dcmctl updateconfig -ct oc4j
```
3. 次のコマンドで OC4J_BI_Forms を再起動します。

```
opmnctl restartproc gid="OC4J_BI_Forms"
```
4. Web Cache Manager にログインします。
5. ナビゲータ・ペインで「**Origin Servers, Sites, and Load Balancing**」 → 「**Session Binding**」を選択します。
6. 「**Session Binding**」画面で、「**Default Session Binding**」を選択して、「**Edit Selected**」を選択します。
7. 「**Edit Session Binding**」ダイアログ・ボックスが表示されます。
8. 「**Please select a session:**」プルダウン・リストで、「**JSESSIONID**」を選択します。
9. Oracle Forms アプリケーション D のセッション・バインディング方式として、ドロップダウン・リストで「**Cookie-based**」を選択します。
10. 「**Submit**」をクリックします。
11. 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 Oracle HTTP Server 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 Oracle HTTP Server 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 Oracle HTTP Server 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 Oracle HTTP Server 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 Oracle HTTP Server 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。

OracleAS Forms Services への アップグレード

Oracle Forms 6i から Oracle Application Server 10g へのアップグレードがサポートされています。アップグレード・プロセスを開始する前に、この章に目を通してください。この章は次の項で構成されています。

- [第 11.1 項「OracleAS Forms Services アップグレード項目」](#)
- [第 11.2 項「OracleAS Forms Services に関連するコンポーネント」](#)
- [第 11.3 項「OracleAS Forms Services アップグレード・タスク」](#)
- [第 11.4 項「OracleAS Forms Services のアップグレードの検証」](#)

Oracle9iAS Forms Services から OracleAS Forms Services にアップグレードする場合は、Oracle Application Server のアップグレードおよび互換性ガイドを参照してください。このマニュアルは、Oracle Application Server 10g のドキュメント CD に収録されています。

11.1 OracleAS Forms Services アップグレード項目

表 11-1 では、アップグレードされる項目について説明します。これらの項目にはファイル、実行可能ファイル、および設定があり、それらを Oracle Application Server Forms Services で追加、変更、削除または置換する必要があります。

表 11-1 OracleAS Forms Services アップグレード項目

アップグレード項目	6i の Oracle ホームでの場所	10g (10.1.2) の Oracle ホームでの場所	説明と注意
Oracle HTTP Server 構成ファイル: 6iserver.conf (forms.conf にアップグレード)	6iserver/conf/	/forms/server	仮想パスのマッピングを含んでいます。
サーブレット環境ファイル: default.env	6iserver/forms60/server	/forms/server	Forms Servlet ランタイム・プロセスの環境変数設定を含んでいます。
Forms Servlet の別名を含む構成ファイル: jserv.properties (web.xml にアップグレード)	/Apache/jserv/conf	/j2ee/OC4J_BI_Forms/applications/formsapp/formsweb/WEB-INF/web.xml	Forms Servlet の別名を含んでいます。
アプリケーション構成ファイル: formsweb.cfg	6iserver/forms60/server	/forms/server	Forms Services アプリケーションの構成情報を含んでいます。
Forms Servlet のテンプレート HTML ファイル: (*.htm, *.html)	6iserver/forms60/server	/forms/server	デフォルトおよびユーザー定義の Forms Servlet テンプレート HTML ファイル。
Forms アプリケーション・モジュール (fmb/fmx ファイル)			Oracle Forms Services 6i に配置された Forms モジュール (fmb および fmx ファイル) をアップグレードして OracleAS Forms Services に配置する必要があります。

11.2 OracleAS Forms Services に関連するコンポーネント

この項では、OracleAS Forms Services とその他のコンポーネントの関係について説明します。OracleAS Forms Services と Oracle Application Server の統合は、一連の OracleAS Forms Services 構成ファイルに依存します。これらの依存関係を、次の表 11-2 で示します。

表 11-2 Oracle Application Server コンポーネントと OracleAS Forms Services 構成ファイルの依存関係

コンポーネント	構成ファイル
Oracle HTTP Server	forms.conf
Oracle Application Server Containers for J2EE	web.xml、formsapp.ear
Oracle Application Server Single Sign-On/Oracle Internet Directory	formsweb.cfg
Oracle Enterprise Manager 10g	formsweb.cfg
Oracle Reports	Forms の内部 PL/SQL ビルトイン

11.3 OracleAS Forms Services アップグレード・タスク

この項では、Oracle Application Server 10g Forms Services のアップグレードを実行する方法について説明します。この項は、次の項目に分かれています。

- 第 11.3.1 項「アップグレードに関する推奨事項およびトラブルシューティングのヒント」(11-3 ページ)
- 第 11.3.2 項「OracleAS Forms Services アプリケーション・モジュールのアップグレード」(11-4 ページ)
- 第 11.3.3 項「Common Gateway Interface (CGI) から Oracle Forms Servlet へのアップグレード」(11-4 ページ)
- 第 11.3.4 項「静的 HTML 開始ファイルから汎用アプリケーション HTML 開始ファイルへのアップグレード」(11-5 ページ)
- 第 11.3.5 項「Forms 6i Listener から Forms Listener Servlet へのアップグレード」(11-7 ページ)
- 第 11.3.6 項「Forms Listener Servlet アーキテクチャから OracleAS Forms Services へのアップグレード」(11-9 ページ)
- 第 11.3.7 項「ロード・バランシングのアップグレード」(11-10 ページ)
- 第 11.3.8 項「使用方法」(11-10 ページ)

11.3.1 アップグレードに関する推奨事項およびトラブルシューティングのヒント

次にあげる、Forms アプリケーションを Oracle Application Server 10g にアップグレードする際の推奨事項と考慮事項に注意してください。

- アプリケーションを Oracle Application Server 10g に配布してテストが完了するまでは、Oracle Forms Services 6i をインストールしたままにしてください。
- 先にソース・ファイルをアップグレードしてから、すべてのアプリケーション・ファイルをバックアップし、保護します。
- 統合された Reports に対する Run_Product のコールを、Oracle Reports に対する Run_Report_Object のコールに置き換えます（または、Oracle Forms で PL/SQL 変換ユーティリティ Forms Migration Assistant を使用します）。
- Oracle Application Server をインストールし、各自のアプリケーションが使用する情報で forms/server/formsweb.cfg ファイルを構成します。
- アプリケーションで使用されている環境ファイルを、同じ相対ディレクトリにコピーします。
- アップグレードされた Oracle Forms アプリケーション・モジュール・ファイルを、Oracle Application Server がインストールされたコンピュータにコピーします（同じコンピュータでない場合）。
- Oracle Application Server を起動した後に、次の URL を使って Forms Services Listener Servlet のテスト・ページにアクセスします。

```
http://<hostname>:<port>/forms/frmservlet?form=test.fmx
```

- すべてのアプリケーション設定が formsweb.cfg に追加されていること、および環境変数 Forms_Path にアプリケーション・モジュールのディレクトリが含まれていることを確認します。
- SQL*Plus を使用してデータベースに接続できることを確認します。
- 次の URL を使用して、アップグレードされたアプリケーションを起動します。

```
http://<hostname>:<port>/forms/frmservlet?config=<your application name>
```

11.3.2 OracleAS Forms Services アプリケーション・モジュールのアップグレード

この項では、Oracle Forms Services 6i に配置された Forms アプリケーション・モジュール (fmb ファイル) からアップグレードする方法を示します。Oracle Forms Services 6i に配置された Forms アプリケーション・モジュール (fmb ファイル) を、OracleAS Forms Services のインストールにアップグレードするには、次の手順に従います。

1. Forms アプリケーション・ファイルを新しいディレクトリにコピーします。
2. Forms Migration Assistant を使用して、Forms アプリケーション・モジュール (.fmb ファイル) をアップグレードします。
3. Forms Migration Assistant を使用して、Forms メニュー・モジュール (.mmx ファイル) をアップグレードします。
4. Forms Migration Assistant を使用して、ライブラリ・モジュール (.pll ファイル) およびメニュー・モジュール (.mmb ファイル) をアップグレードします。
5. Forms Migration Assistant を使用して、ライブラリ・モジュール (.plx ファイル) をアップグレードします。
6. Forms コンパイラ (UNIX の場合は frmcmp.sh、Windows の場合は frmcmp.exe) を使用して、Forms アプリケーションの実行可能ファイル (fmx ファイル) を再生成します。

詳細は、次の URL から『Oracle Forms Forms 6i からの Forms アプリケーションの移行』を参照してください。

<http://www.oracle.com/technology/documentation/>

11.3.3 Common Gateway Interface (CGI) から Oracle Forms Servlet へのアップグレード

この項では、Forms CGI を Forms Servlet にアップグレードする方法を示します。Oracle Forms Services 6i Common Gateway Interface を使用して、アプリケーションの Forms アプレット開始 HTML ファイルを動的にレンダリングしている場合は、次の手順に従います。

Forms アプリケーションの CGI は、Forms アプレット開始 HTML ファイルの動的なレンダリングを可能にする目的で、Oracle Forms Services リリース 6i に導入されました。Forms CGI では、formsweb.cfg 構成ファイルおよび HTML テンプレートを使用して、アプリケーションの開始 HTML ファイルが作成されます。CGI インタフェースは、Forms HTTP 構成ファイル 6iserver.conf のエントリ (Oracle HTTP Server oracle_apache.conf ファイルの Include ディレクティブで参照されている) によって構成されます。このファイルには、ifcgi60.exe ファイルを含んでいるディレクトリ構造の dev60cgi を識別する ScriptAlias ディレクティブが含まれています。

Forms Servlet は、CGI と同じ方法で HTML をレンダリングするほかに、ブラウザ・タイプの自動検出機能も備えています。Forms Servlet は、OracleAS Forms Services をインストールするときに構成され、frmservlet と命名されます。

Forms Servlet にアクセスするには、次の URL をリクエストします。

```
http://<hostname>:<port>/forms/frmservlet
```

この URL は、Oracle Forms Services 6i の CGI インタフェースで使用される URL と似ています。forms/server/formsweb.cfg ファイルのカスタム構成セクションで myapp として構成されているアプリケーションをコールするには、次の URL をリクエストします。

```
http://<hostname>:<port>/forms/frmservlet?config=myapp
```

Forms Servlet は、インストール時に自動的に構成されます。インストーラにより、Oracle Application Server Containers for J2EE Business Intelligence and Forms インスタンス (<destination_MT_OH>/j2ee/oc4j/OC4J_BI_Forms) 内に、OracleAS Forms Services 構成である formsapp および formsweb を指す仮想パス /forms/ が作成されます。

Oracle Forms Services リリース 6i の CGI 環境を、OracleAS Forms Services のサーブレット環境にアップグレードするには、次の手順に従います。

1. `<source_MT_OH>/Forms60/Server/formsweb.cfg` からアプリケーション固有の構成をすべてコピーし、`<destination_MT_OH>/forms/server/formsweb.cfg` に追加します。

注意： `<source_MT_OH>` 内の `formsweb.cfg` ファイル全体を、`<destination_MT_OH>` にコピーして置き換えないでください。このファイルは、リリース 6i と OracleAS Forms Services とで異なっています。アプリケーション構成だけを `<destination_MT_OH>/forms/server/formsweb.cfg` へコピーします。

2. `forms/server/default.env` ファイルの `Forms_Path` が、アップグレードされた OracleAS Forms Services アプリケーション・モジュールを指すように構成します。

注意： 新しい環境ファイルを作成できます。そのためには、`default.env` をコピーし、そのファイルを特定のアプリケーションで使用できるように変更し、さらに `envFile=<作成した環境ファイル>` を `formsweb.cfg` ファイルのカスタム・アプリケーション・セクションに追加します。

3. Oracle Forms 6i の HTML テンプレート・ファイルを変更した場合は、OracleAS Forms Services の HTML テンプレート・ファイルも同じように変更します。

注意： これらの変更は、`basejini.htm` および `basejpi.htm` で行う必要があります。これは、サーブレットが JInitiator および Java Plug-in をサポートしているためです。

11.3.4 静的 HTML 開始ファイルから汎用アプリケーション HTML 開始ファイルへのアップグレード

`formsweb.cfg` 構成ファイルには、OracleAS Forms Services に配布されたアプリケーションごとに、カスタム・アプリケーション定義が構成されています。この定義は、使用される JInitiator のバージョンやベース HTML テンプレート・ファイルの名前と場所などの一般的なシステム設定を自動的に継承します。

カスタム・アプリケーション定義の名前は、Forms アプリケーションの URL に組み込まれます。次のカスタム設定は、2つの異なるアプリケーションを定義しています。

```
[MyHR_app]
serverURL=/forms/lservlet
Form = hr_main.fmx
lookAndFeel=oracle
Otherparams=myParam1=12
userid=scott/tiger@orcl
```

次の URL により、このアプリケーションが起動されます。

```
http://<hostname>:<port>/forms/frmservlet?config=MyHR_app
```

もう 1 つのカスタム・アプリケーション定義は、次のようになっています。

```
[booking_app]
ServerURL=/forms/lservlet
Form = book.fmx
lookAndFeel=oracle
Otherparams=
userid=
```

次の URL により、このアプリケーションが起動されます。

```
http://<hostname>:<port>/forms/frmservlet?config=booking_app
```

静的 HTML ファイルごとに、カスタム・アプリケーション定義を作成する必要があります。静的 HTML ファイル内の `archive` パラメータ・ディレクティブには、少なくとも OracleAS Forms Services の `frmall.jar` ファイルが指定されています。カスタム・アーカイブ・ファイルを追加した場合は、`archive` パラメータ・ディレクティブが

`Archive=frmall.jar,custom.jar` のようになります。アーカイブの設定は、Forms Servlet および `formsweb.cfg` ファイルを使用して、User Parameter セクションに定義されます。カスタムのアプリケーション設定はすべてこれらの値を継承するため、アプリケーションの必要に応じて `custom.jar` ファイルを追加しないかぎり、このパラメータを明示的に設定する必要はありません。

`custom.jar` が追加されている場合は、次の行をカスタム・アプリケーション定義に追加できます。次の例は、JInitiator または別の VM を使用していることを前提としていますが、Internet Explorer 固有ではありません。

```
[booking_app]
archive_jini=frmall_jinit.jar, custom.jar
archive=frmall.jar, custom.jar
ServerURL=/forms/lervlet
Form = book.fmx
lookAndFeel=oracle
Otherparams=
Userid=
```

アプリケーションをアップグレードするには、次の手順に従います。

1. `forms/server/default.env` ファイルを編集して、OracleAS Forms Services アプリケーション・モジュールの場所を `Forms_Path` に追加します。
2. `forms/server/formsweb.cfg` ファイルを編集して、置換する静的 HTML アプリケーションごとにカスタム・アプリケーション・セクションを追加します。
3. 各カスタム・アプリケーション・セクションに名前を付けます。名前は、空白を含めず、角カッコで囲んで指定します（たとえば、`[booking_app]` や `[MyHR_app]` など）。
4. 次の URL を使用してアプリケーションを起動します。

```
http://<hostname>:<port>/forms/frmservlet?config=<name>
```

11.3.4.1 OracleAS Forms Services での静的 HTML ファイルの使用

必要であれば、OracleAS Forms Services でも引き続き静的 HTML ファイルを使用できます。ただし、静的 HTML ファイルを使用する場合は、Forms アプリケーションで一部の機能（Oracle Application Server Single Sign-On など）を使用できなくなります。

インストール後は、Forms Listener Servlet がデフォルトで `/forms/lervlet` を示します。OracleAS Forms Services で静的 HTML ファイルを使用するには、`serverURL` パラメータの値を含むように各静的開始 HTML ファイルを変更する必要があります。`serverPort` パラメータと `serverHost` パラメータは使用されなくなったため、未定義のままにしておかまいません。OracleAS Forms Services では JInitiator バージョン 1.3.x が使用されるため、これらの設定も変更する必要があります。必要な値は、`forms/server/formsweb.cfg` ファイルで見つかります。

OracleAS Forms Services で静的 HTML ファイルを使用するには、次の手順に従います。

1. `forms/server/default.env` ファイルで、アップグレードされた OracleAS Forms Services アプリケーション・モジュールを示すように `Forms_Path` を構成します。
2. `<destination_MT_OH>/forms/server/forms.conf` ファイルに、静的 HTML 開始ファイルの場所を示す仮想ディレクトリを作成します。

3. アプリケーションの開始 HTML ファイルを次のように変更します。
 - a. `serverURL` の値 `/forms/lervlet` を追加します。
 - b. `JInitiator` のバージョン番号を変更します。
4. `codebase` パラメータを `forms/java` に変更します。
5. `<destination_MT_OH>/j2ee/OC4J_BI_Forms/applications/formsapp/formsweb/WEB-INF` に移動し、`web.xml` ファイルを編集します。
6. 環境ファイルを指すように、Listener Servlet の `envFile` 初期化パラメータを設定します (通常は `<destination_MT_OH>/forms/server/default.env`)。

編集後は、`web.xml` ファイルの Forms Listener Servlet に対応するエントリが次のようになります。

```
<!--Forms listener servlet-->
<servlet>
  <servlet-name>lervlet</servlet-name>
  <servlet-class>oracle.forms.servlet.ListenerServlet</servlet-class>
  <init-param>
    <param-name>envFile</param-name>
    <param-value>destination_MT_OH/forms/server/default.env</param-value>
  </init-param>
</servlet>
```

11.3.5 Forms 6i/Listener から Forms Listener Servlet へのアップグレード

Forms 6i Listener は C プログラムで、受信する Forms Web リクエストに代わって Forms ランタイム・プロセスを起動します。起動された Forms Web ランタイム・プロセスは、Forms クライアント・アプレットから直接ソケット接続または HTTP ソケット接続を使用して直接アクセスされます。この時点で Forms Listener はアプリケーションの Web クライアント / サーバー間通信に関与しなくなり、他の Web リクエストを処理できるようになります。

Java プログラムである Forms Listener Servlet も、Forms アプリケーションに対する Web リクエストを処理して、Forms のサーバー側 Web ランタイム・プロセスを起動します。Forms 6i Listener とは異なり、Forms Listener Servlet は Forms アプリケーションのアプレット / サーバー間通信に介在したままとなります。

Forms 6i Listener は特定のポート (デフォルトは 9000) でリスニングしますが、Forms Servlet は追加のポートを必要とせず、HTTP リスナー・ポート経由でアクセスされます。Forms Listener Servlet は、Forms 6i バッチ 4 で導入されたもので、Forms Services でサポートされている唯一のリスナーです。

Forms Listener Servlet は、Oracle Application Server のインストール時に自動的に構成されます。インストーラにより、Oracle Application Server Containers for J2EE Business Intelligence and Forms インスタンス (`<destination_MT_OH>/j2ee/oc4j/OC4J_BI_Forms`) 内に、OracleAS Forms Services 構成である `formsapp` および `formsweb` を指す仮想パス `/forms/` が作成されます。

Forms Listener Servlet のテスト・フォームにアクセスするには、次の URL をリクエストします。

```
http://<hostname>:<port>/forms/frmservlet?form=test.fmx
```

このページにアクセスできるということは、Forms Listener Servlet が構成されていて使用できるようになっていることを意味します。`frmservlet` は、インストール時に Forms Servlet によって構成されたアクセス名です。Listener Servlet の名前は、`lervlet` です。

Forms Listener Servlet が Forms Servlet によってアクセスされる場合は、`Forms60/server/formsweb.cfg` ファイルのカスタム・アプリケーション設定のみを `forms/server/formsweb.cfg` ファイルに追加します。すべてのアプリケーション構成は、グローバル・システム・パラメータ設定から自動的に `serverURL` パラメータの値 `/forms/lervlet` を継承します。

注意： <source_MT_OH> 内の formsweb.cfg ファイル全体を、<destination_MT_OH> にコピーして置き換えないでください。このファイルは、リリース 6i と OracleAS 10g リリース 2 (10.1.2) の Forms Services とで異なります。アプリケーション構成だけを <destination_MT_OH>/forms/server/formsweb.cfg へコピーします。

Forms アプリケーションの配布を Forms Listener アーキテクチャから Listener Servlet アーキテクチャに変更するには、formsweb.cfg ファイルの serverURL パラメータに値を指定します。このパラメータは、インストール時に /forms/lervlet に設定されています。

Forms Listener Servlet にアップグレードするには、次の手順に従います。

1. Forms アプリケーションのファイルを新しいディレクトリにコピーし、11-4 ページの第 11.3.2 項「OracleAS Forms Services アプリケーション・モジュールのアップグレード」で説明されているように、それらのファイルを OracleAS Forms Services のモジュールにアップグレードします。
2. アップグレードされた Forms アプリケーション・モジュールの場所が Forms_Path 変数に追加されるように、forms/server/default.env ファイルを編集します。
3. <source_MT_OH>/Forms60/Server/formsweb.cfg からすべてのカスタム・アプリケーション設定をコピーして、<destination_MT_OH>/forms/server/formsweb.cfg に追加します。

注意： <source_MT_OH> 内の formsweb.cfg ファイル全体を、<destination_MT_OH> にコピーして置き換えないでください。このファイルは、Oracle Forms Services 6i と OracleAS Forms Services とで異なります。アプリケーション構成だけを <destination_MT_OH>/forms/server/formsweb.cfg へコピーします。

4. アプリケーションに独自の環境ファイルが必要な場合は、Listener Servlet にサブレットの別名を定義するかわりに、<destination_MT_OH>/forms/server/formsweb.cfg のカスタム・アプリケーション定義セクションで envFile パラメータが新しい環境ファイルを指すように設定します。次に例を示します。

```
envFile=myEnvFile.env
```

myEnvFile.env は、forms/server ディレクトリにあります。

5. Oracle Forms Services 6i の HTML テンプレート・ファイルを変更した場合は、OracleAS Forms Services の HTML テンプレート・ファイルも同じように変更します。

注意： 基本の HTML ファイルを変更する必要がある場合は、編集する前に付属のテンプレート・ファイルのコピーを作成します。編集済の HTML ファイルを別の名前で作成して、インストールにより提供されたデフォルトのテンプレートには変更を加えないようにします。これにより、アプリケーションにパッチ・セットが適用された際にカスタマイズ HTML テンプレートが上書きされることを防ぎます。

アプリケーションで作成した独自のテンプレート・ファイルを使用するには、システム・セクションまたはカスタム・アプリケーションの定義の 1 つで次のパラメータを使用します。

```
baseHTML=<your base template>.htm
baseHTMLJinitiator=<your base jinit>.htm
```

6. 次の URL でアプリケーションを起動します。

```
http://<hostname>:<port>/forms/frmservlet?
config=<application>
```

11.3.6 Forms Listener Servlet アーキテクチャから OracleAS Forms Services へのアップグレード

Oracle9iAS Forms Services リリース 6i では、別名の付いていない Listener Servlet が `oracle.forms.servlet.ListenerServlet` でアクセスされます。Listener Servlet 構成は、`jserv.properties` ファイルおよび `zone.properties` ファイルにあります。

OracleAS Forms Services での Forms Listener Servlet は、サーブレット名 (`frmservlet` および `lservlet`) およびサーブレット・コンテナ (現在は Oracle Application Server Containers for J2EE (OC4J)) を除いて同じです。Oracle9iAS リリース 1 (1.0.2.2.x) と同様に、構成はインストール中に行われます。OC4J での Listener Servlet 構成は、`ORACLE_HOME/j2ee/OC4J_BI_Forms/applications/formsapp/formsweb/WEB-INF/web.xml` に格納されます。一部の初期化パラメータは、`formsweb.cfg` ファイルに移されたため、`envFile` パラメータと同様にサーブレット・エンジンで構成する必要がなくなりました。

Forms Listener Servlet は、Oracle Application Server のインストール時に自動的に構成されます。インストーラにより、Oracle Application Server Containers for J2EE Business Intelligence and Forms インスタンス (`<destination_MT_OH>/j2ee/oc4j/OC4J_BI_Forms`) 内に、OracleAS Forms Services 構成である `formsapp` および `formsweb` を指す仮想パス `/forms/` が作成されます。

Forms Listener Servlet のテスト・フォームにアクセスするには、次の URL をリクエストします。

```
http://<hostname>:<port>/forms/frmservlet?form=test.fmx
```

このページにアクセスできるということは、Forms Listener Servlet が構成されていて使用できるようになっていることを意味します。`frmservlet` は、インストール時に Forms Servlet によって構成されたアクセス名です。Listener Servlet の名前は、`lservlet` です。

Listener Servlet アーキテクチャを OracleAS Forms Services にアップグレードするには、次の手順に従います。

1. Forms アプリケーションのファイルを新しいディレクトリにコピーし、それらを OracleAS Forms Services モジュールにアップグレードします。
2. `forms/server/default.env` ファイルを編集して、アップグレードされた Forms アプリケーション・モジュールの場所を `Forms_Path` 変数に追加します。
3. `<source_MT_OH>/Forms60/Server/formsweb.cfg` からすべてのカスタム・アプリケーション設定をコピーして、`<destination_MT_OH>/forms/server/formsweb.cfg` に追加します。

注意: `<source_MT_OH>` 内の `formsweb.cfg` ファイル全体を、`<destination_MT_OH>` にコピーして置き換えないでください。このファイルは、リリース 6i と OracleAS 10g リリース 2 (10.1.2) の Forms Services とで異なります。アプリケーション構成だけを `<destination_MT_OH>/forms/server/formsweb.cfg` へコピーします。

4. アプリケーションに独自の環境ファイルが必要な場合は、Listener Servlet にサーブレットの別名を定義するかわりに、`<destination_MT_OH>/forms/server/formsweb.cfg` のカスタム・アプリケーション定義セクションで、`envFile` パラメータが新しい環境ファイルを指すように設定します。次に例を示します。

```
envFile=myEnvFile.env
```

`myEnvFile.env` は、`forms/server` ディレクトリにあります。

5. Forms Services リリース 6i の HTML テンプレート・ファイルを変更した場合は、OracleAS Forms Services の HTML テンプレート・ファイルも同じように変更します。

注意： 基本の HTML ファイルを変更する必要がある場合は、編集する前に付属のテンプレート・ファイルのコピーを作成します。編集済の HTML ファイルを別の名前で保存して、インストールにより提供されたデフォルトのテンプレートには変更を加えないようにします。これにより、アプリケーションにパッチ・セットが適用された際にカスタマイズ HTML テンプレートが上書きされることを防ぎます。

アプリケーションで作成した独自のテンプレート・ファイルを使用するには、システム・セクションまたはカスタム・アプリケーションの定義の 1 つで次のパラメータを使用します。

```
baseHTML=<your base template>.htm
baseHTMLJinitiator=<your base jinit>.htm
```

6. 次の URL でアプリケーションを起動します。

```
http://<hostname>:<port>/forms/frmservlet?
config=<application>
```

11.3.7 ロード・バランシングのアップグレード

Forms Services 6i のロード・バランシングをアップグレードする方法は、使用した配置方法によって異なります。

- Forms 6i リスナーの場合は、Metrics Server (別プロセス) によってロード・バランシングが実行されます。
- Forms 6i サーブレットの場合は、JServ エンジン間のラウンド・ロビン・ロード・バランシングを使用して、JServ サーブレット・エンジンでロード・バランシングが構成されます。
- OracleAS Forms Services では、Oracle HTTP Server のモジュールである mod_oc4j によってロード・バランシングが管理されます。このモジュールは、Forms Servlet および Forms Listener Servlet を処理するサーブレット・コンテナに Web リクエストをバインドします。

11.3.8 使用方法

この項では、アップグレードで役立つヒントを紹介します。

11.3.8.1 Forms Servlet によるアイコン・イメージの配置

Forms Services リリース 6i では、静的 HTML 開始ファイルを使用することで、開始 HTML ファイルを基準とした相対的な場所にイメージを格納できました。OracleAS Forms Services の Forms Servlet では、これがサポートされていません。

かわりに、imagebase パラメータに codebase 値を指定し、それをアプリケーションが使用するアイコン・イメージの場所として使用します。codebase 値は、Forms クライアントの Java アーカイブ・ファイルをすべて格納する forms/java ディレクトリを参照します。パフォーマンス上の理由から、ここにイメージを格納することはお勧めできません。

かわりに、アイコンを別のアーカイブ・ファイルにまとめて格納します。これによって、クライアント上でアーカイブが永続的にキャッシュされるため、パフォーマンスが向上します。このアーカイブ・ファイルを作成するには、次の手順に従います。

1. jar コマンドが成功することを確認します。成功しない場合は、システムにインストールされている JDK に、正しい PATH 環境変数エントリが設定されていることを確認する必要があります (この変数は JDK_HOME/bin ディレクトリを指しています)。
2. アプリケーション・イメージが含まれているディレクトリに移動し、次のコマンドを発行します。

```
jar -cvf <application>_images.jar *.<extension>
```

ここで、

- `application` はアプリケーションの名前です。
- `extension` はイメージ・ファイルの拡張子です (.gif など)。

JAR ファイル `<application>_images.jar` は、現在のディレクトリに作成されます。

3. `<application>_images.jar` を `forms/java` ディレクトリにコピーします。
4. `formsweb.cfg` ファイルを編集して、アプリケーションのカスタム・アプリケーション・セクションに `imageBase=codebase` パラメータを追加します。
5. `<application>_images.jar` ファイルを、アプリケーションが使用するアーカイブ・パスに追加します。そのためには、カスタム・アプリケーション・セクションに次の行を追加します。

```
archive_jini=frmall_jinit.jar,<application>_images.jar
archive_frmall.jar,<application>_images.jar
```

OracleAS Forms Services でのカスタム・アイコン・ファイルの配置の詳細は、[第 4.9 項「Forms Services で使用するアイコンとイメージの配置」](#)を参照してください。

11.3.8.2 Oracle Reports に対する統合されたコールから Oracle Reports の使用へのアップグレード

Oracle Application Server では、Forms での Oracle Reports に対する統合されたコールが、クライアント側のバックグラウンド・エンジンで処理されなくなりました。OracleAS Forms Services では、アプリケーションが `RUN_REPORT_OBJECT` ビルトインを使用することで、Oracle Reports をコールして統合レポートを処理する必要があります。Oracle Reports は、Business Intelligence and Forms インストールの一部として設定されます。

コールをアップグレードするには、次の手順に従います。

1. すべての `RUN_PRODUCT (Reports, ...)` を、`RUN_REPORT_OBJECT ()` を使用する同等のコールに変更します。
2. Oracle Reports の `Reports_Path` が使用されるように、アプリケーションの Reports モジュールの場所を追加します。
3. Oracle Reports を参照するように `RUN_REPORT_OBJECT` を変更します。

詳細は、『Oracle Application Server Reports Services レポート Web 公開ガイド』および <http://www.oracle.com/technology/products/forms/pdf/10g/frmrepparamform.pdf> を参照してください。

11.3.8.3 OC4J での Forms Listener Servlet 別名の作成

パッチ 8 が適用される前の Forms Services リリース 6i では、アプリケーションごとに個別の環境ファイルを使用するために、Forms Servlet の別名を `ORACLE_HOME/Apache/Apache/JServ/conf/zone.properties` ファイルに作成する必要がありました。OracleAS Forms Services の Forms Servlet ではこれが必要ありません。次に示すように、`formsweb.cfg` ファイルで `envFile` パラメータを使用して環境ファイル名を設定できます。

```
envFile=myApp.env
```

Forms Servlet の別名は、`ORACLE_HOME/Apache/Apache/JServ/conf/zone.properties` に作成されなくなりました。かわりに、`<destination_MT_OH>/j2ee/OC4J_BI_Forms/applications/formsapp/formsweb/WEB-INF/web.xml` に作成されます。

別名を作成するには、`<servlet>` タグと `</servlet>` タグの内容をコピーし、サーブレットの名前を変更します。新しいサーブレットの別名に対して URL マッピングを作成するには、ファイルに次の行を追加します。

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

11.3.8.4 Listener Servlet 管理ページへのアクセス

Oracle9iAS Forms Services リリース 6i で次の URL にアクセスすると、Listener Servlet のテスト・ページが表示されます。

```
http://<hostname>:<port>/servlet/  
oracle.forms.servlet.ListenerServlet
```

表示される情報は、初期化パラメータ TestMode の値に依存します。このパラメータは、<source_MT_OH>/Apache/Apache/JServ/conf/zone.properties ファイルに設定されます。

OracleAS Forms Services のテスト・ページは、次の URL で表示できます。

```
http://<hostname>:<port>/forms/frmservlet/admin
```

表示される情報は、初期化パラメータ TestMode の値に依存します。このパラメータは、<destination_MT_OH>/j2ee/OC4J_BI_Forms/applications/formsapp/formsweb/WEB-INF/web.xml ファイルで設定されます。次に例を示します。

```
<init-param>  
<!-- Display sensitive options on the /admin page ? -->  
  <param-name>TestMode</param-name>  
  <param-value>>true</param-value>  
</init-param>
```

11.4 OracleAS Forms Services のアップグレードの検証

アップグレード・タスクが完了したら、OracleAS Forms Services のアップグレード・バージョンが予期したとおりに動作することを確認します。各サイト固有のアプリケーションや構成要素に合わせたテストを考案し、実行する必要があります。アップグレード前およびアップグレード後のインストール環境で、各アプリケーションのパフォーマンスと特性を比較します。

Oracle9iAS リリース 1 (1.0.2.2.x) での一般的な Forms アプリケーションの URL は次のとおりです。

```
http://<hostname>:<port>/servlet/<forms servlet alias>?<forms  
application name>
```

Oracle Application Server 10g での一般的な Forms アプリケーションの URL は次のとおりです。

```
http://<hostname>:<port>/forms/<forms servlet alias>?<forms application  
name>
```

Oracle Forms Services の トラブルシューティング

この章には、次の項が含まれています。

- 第 A.1 項 「インストールの確認」
- 第 A.2 項 「FRM-XXXXX エラーの診断」
- 第 A.3 項 「スタック・トレースを使用したサーバーのクラッシュの診断」
- 第 A.4 項 「クライアント・クラッシュの診断」
- 第 A.5 項 「Forms Trace と Servlet Logging Tools」
- 第 A.6 項 「メモリーの問題の解決」
- 第 A.7 項 「トラブルシューティングのヒント」
- 第 A.8 項 「さらに調査が必要な場合」

この章では、Oracle Forms を使用して Web 上でアプリケーションを実行する際に発生する可能性のある問題の解決に役立つ情報を提供します。この中では、エラーの一般的な原因について概説し、インストール内容を確認する方法と、問題を診断するためのツールとテクニックを紹介します。

この章は、ホワイト・ペーパー 『Oracle Forms Diagnostic Techniques』
(<http://www.oracle.com/technology/products/forms/>) のサブセットでもあります。

A.1 インストールの確認

インストールになんらかの問題があると、構成が不正になり、Oracle Forms が正しく動作しなくなります。Oracle Universal Installer で Oracle Application Server または Developer Suite が正常にインストールされた後に、Oracle Forms Services が正しく構成されているかどうかを確認できます。解決策として次のツールを使用します。

第 A.1.1 項「Web Form Tester の使用」

第 A.1.2 項「ポート情報の確認」

A.1.1 Web Form Tester の使用

Form Tester は、Oracle Application Server または Developer Suite のインストールに付属します。OracleAS のインストールおよび Forms Services の構成が正しいかどうかを確認するには、中間層で Web Form Tester を実行します。Windows コンピュータでは、この操作を次のように実行します。

1. Oracle HTTP Server がまだ起動されていない場合は、「スタート」→「プログラム」→「Oracle Application Server-AS Home」→「Oracle HTTP Server」→「Start HTTP Server」を選択して起動します。

Oracle Developer Suite の場合、OC4J インスタンスがまだ起動されていないときは、「スタート」→「プログラム」→「Oracle Developer Suite - DevSuiteHome」→「Forms Developer」→「Start OC4J Instance」を選択して起動します。

2. URL に ORACLE_HOME/tools/web/html/runform.htm と入力し、[Enter] キーを押してブラウザのインスタンスを開きます。ORACLE_HOME は、OracleAS または Developer Suite の実際の Oracle ホームで置き換えます。
3. Web Forms Tester を実行します。OracleAS の場合は、Windows の「スタート」メニューから「プログラム」→「OracleAS – Home for AS」→「Forms Developer」→「Run a Form on the Web」を選択します。

Oracle Developer Suite の場合は、Windows の「スタート」メニューから「プログラム」→「Oracle Developer Suite – Home for DS」→「Forms Developer」→「Run a Form on the Web」を選択します。

4. Web ポートを入力し、「フォームの実行」ボタンをクリックします。Web ポートを調べる方法については、第 A.1.2 項「ポート情報の確認」を参照してください。
5. OracleAS または Developer Suite のインストールに問題がなければ、Web ブラウザに成功メッセージが表示されます。また、クライアント・コンピュータからも、インストーラによって中間層で OracleAS または Developer Suite の Forms が正しく基本設定されているかをテストできます。クライアント・コンピュータからテスト・フォームを実行するには、ブラウザから URL `http://myserver.com:NNNN/forms90/f90servlet?form=test.fmx` を実行します。

A.1.2 ポート情報の確認

インストール後に問題が疑われる場合や、Forms の実行に使用するポート番号がわからない場合は、ORACLE_HOME/install/portlist.ini ファイルのポート情報を参照してください。

インストールに適したポート番号を使用します。

A.2 FRM-XXXXX エラーの診断

FRM-XXXXX エラーの診断と解決には、次のツールを使用します。

- 第 A.2.1 項「Oracle Forms アプレット」
- 第 A.2.2 項「JInitiator Java コンソール」
- 第 A.2.3 項「FRM-92XXX エラー・メッセージ」

A.2.1 Oracle Forms アプレット

問題の基本的な原因の特定に、簡潔な FRM エラーのメッセージが役立ちます。多くの場合、Forms アプレットからレポートされるエラーには、FRM エラーの原因の特定が可能なすべての情報が含まれています。FRM エラーが発生すると、エラー・ダイアログに「詳細」ボタンが表示されます。「詳細」ボタンを押すと、現在の Java スタックが表示されます。このスタックには、根本的な原因と Oracle Forms のバージョンが関連付けられています。これは、リリースが異なると、アプレット・クラス・ファイルに使用されるパッケージ構造も異なるためです。

A.2.2 JInitiator Java コンソール

JInitiator を使用している場合に Java エラーが発生すると、エラーがブラウザのステータス行に表示されます。ただし、ここには完全には Java エラー・スタックが表示されません。JInitiator Java コンソールで該当するエラー・スタックを探する必要があります。

Windows コンピュータの JInitiator コントロール・パネル・アプレットで、Java コンソールをオンにした場合は、ブラウザでフォームを実行すると JInitiator コンソールのウィンドウが表示されます。JInitiator コントロール・パネル・アプレットは、Windows で「スタート」→「設定」→「コントロールパネル」を選択して表示します。JInitiator の Java コンソールが表示されない場合は、タスクバーのトレイ・アイコンをダブルクリックすることでいつでも起動できます。

A.2.3 FRM-92XXX エラー・メッセージ

Forms アプリケーションの実行中に、様々な FRM エラーが発生する場合があります。これらのエラーは、何種類かの状況で発生します。これらのエラーが発生した場合は、詳細な情報入手して解決する必要があります。この項では、よく発生するエラーをいくつか取り上げて、その解決方法を示します。

FRM エラーが発生する原因は、大きく分けて次のとおりです。

- 構成の問題：
 - 一部の FRM エラーは、構成の問題が原因で発生します。たとえば、Forms Services が起動されていない場合や、HTML ファイルに指定されたポートと異なるポートでリスニングしている場合などです。通常、これらのエラーは常に再現されます。
- Forms Server プロセスのクラッシュ：
 - 正常に接続されフォームが起動された後に発生する FRM エラーの大半は、サーバーのクラッシュが原因です。サーバー・プロセスがクラッシュすると、クライアントは実行を継続できなくなります。サーバー・プロセスと通信できない状態では、アプレットは単独で存続できず稼働できません。
 - これらのエラーは、通常は診断が困難です。問題を常に再現できなかつたり、クラッシュに至った一連の原因をユーザーが知らないためです。
- ネットワークの問題：
 - アプレットと Forms Server プロセス間の通信でネットワークの問題が発生し、通信が遮断されました。

表 A-1 は、これらの原因により発生する FRM-92xxx エラーの一覧です。個々のエラーの簡単な説明も記載しています。

表 A-1 FRM-92XXX の説明

エラー	説明
FRM-92000	クラス・ファイルをロードするときに、Java 言語から <code>IllegalAccessException</code> がスローされて発生する内部エラーです。通常は、システムの構成になんらかの誤りがあることを示します。詳細メッセージを見れば、たいていは発生の原因を特定できます。
FRM-92010	クライアントの構成エラーです。アプレットのパラメータ <code>serverArgs</code> が存在しないか、または NULL 値に設定されていると発生します。
FRM-92020	要求した URL またはブラウザ・ターゲットが、なんらかの形でブラウザによって拒否されたことを示します。
FRM-92030	クライアントの構成エラーです。Java クラス・ファイルの欠落およびレジストリの構成ミス（またはそのどちらか）が原因です。このエラーは、サーバーが数値の <code>handlerClassId</code> で要求した Java クラスが、レジストリに存在しないためにクライアントが処理できない場合に発生します。
FRM-92040	サーバーの構成エラーです。Java クラス・ファイルの欠落が原因です。このエラーは、クライアントの要求した Java クラスがサーバーで見つからなかった場合に発生します。
FRM-92050	クライアントが、指定されたソケット（ポート）でサーバー・コンピュータ（ホスト）との接続を確立できませんでした。
FRM-92060	ホスト / ポートの組合せのフォーマットが不正なために、クライアントがサーバーとの接続を確立できませんでした。
FRM-92070	クライアントがなんらかの理由で新しいオブジェクトを作成できませんでした。詳細を確認すると、エラーの発生原因がすべて示されている可能性があります（このエラーが発生してもフォームの動作は停止しません。エラーがログ・ファイルに記録されるだけです）。
FRM-92080	外部ブラウザ・モジュールを起動しようとしてオペレーティング・システムのコマンドを実行したときに、なんらかの問題が発生しました。
FRM-92090	予期しないエラーが発生しました。
FRM-92095	使用されている <code>JInitiator</code> のバージョンが古いために、要求された機能（ <code>Listener Servlet</code> に対する実行など）がサポートされていません。ユーザーは、指定されたバージョン（またはそれ以上）をインストールする必要があります。
FRM-92100	予期しないネットワーク・エラーまたはサーバー障害が発生しました。
FRM-92101	サーバー側の構成ミスが原因で、予期しないサーバー障害が発生しました。

表 A-1 FRM-92XXX の説明 (続き)

エラー	説明
FRM-92102	Forms が特定の回数再接続を試行した後に、予期しないネットワーク・エラーが発生しました。
FRM-92120	サーバー構成エラーで、クライアントが重要なファイル (レジストリ) を見つけることができませんでした。
FRM-92145	シングル・サインオン認証の失敗を説明するテキストです。
FRM-92150	クライアントのバージョンがサーバーのバージョンより新しいです。
FRM-92160	クライアントのバージョンがサーバーのバージョンより古いです。
FRM-93000	汎用的な内部例外メッセージです。予期しないコード・エラーが発生した場合に使用されます。

前述の FRM エラーではメッセージによりほとんどの原因を特定できますが、別に原因があり、診断が難しい場合も少数存在します。次のトピックでは、このような FRM エラーについて説明し、考えられる原因とその解決方法を示します。

A.2.3.1 FRM-92010

原因:

このエラーは、JInitiator がブラウザのプロキシ設定を使用する際に発生する場合があります。

解決策:

「コントロールパネル」から「JInitiator 1.3.x.x」→「プロキシ」選択し、「ブラウザの設定を使用」の選択を解除してから、プロキシ設定の詳細を入力します。

A.2.3.2 FRM-92050

- サーバーへの過剰な負荷

原因:

多数のリクエストが同時に発生したためにサーバーで処理できません。この現象は主にサーバー・コンピュータのパフォーマンスと構成に関係します。

解決策:

- このような状況で役立つのが、Oracle Application Server Forms Services 10g の実行時プレスタート機能です。この機能は、クライアントからのリクエストを処理するランタイム・エンジンを必要な数だけ事前に起動することで、リクエストが急増してもアプリケーションやサーバーがハングしないようにします。
- サーバー・コンピュータのハードウェアをアップグレードして、多数のリクエストを同時に処理できるようにします。

- serverURL パラメータの欠落

原因:

構成ファイル (formsweb.cfg) に serverURL パラメータがないか、またはこのパラメータが間違っています。

解決策:

Forms 構成ファイルを編集して、有効な serverURL パラメータ値を入力します。

- 不正な FORMS_TIMEOUT
原因：
FORMS_TIMEOUT パラメータの値が不正です。
解決策：
環境ファイル (default.env) およびレジストリで、FORMS_TIMEOUT パラメータの値を確認します。この値を適切な整数にします。次のように引用符で囲まないでください。
FORMS_TIMEOUT="10"。このエントリは不正です。
FORMS_TIMEOUT=10。これは正しいエントリです。
- RUN_PRODUCT の不適切な使用
原因：
RUN_PRODUCT は、Oracle Forms で Oracle Graphics 6i を統合するためにのみ使用します。
解決策：
Oracle Forms を Oracle Reports と統合するために使用されている RUN_PRODUCT ビルトイン・コールを、新しい RUN_REPORT_OBJECT ビルトインで置き換えます。
- ServerArgs パラメータの欠落
原因：
HTML にアプレットのロードを行う ServerArgs パラメータがありません。
解決策：
 - フォーム・アプレットのロードに使用される HTML ファイルに、必ず ServerArgs パラメータが含まれるようにします。
 - ServerArgs の値が NULL になっていないことを確認します。ServerArgs ではフォーム名が必要です。これらのパラメータは、Oracle Forms 構成ファイル (formsweb.cfg) で定義するか、フォームの実行に使用される URL に直接渡します。
- jvm.dll の欠落
原因：
Forms Web の実行可能ファイル frmweb.exe で、jvm.dll を検索できません。
解決策：
jvm.dll が、PATH 環境変数に指定されたディレクトリにあることを確認します。formsweb.cfg で、PATH 環境変数が jvm.dll の場所を示すように設定します。通常この変数の値は ORACLE_HOME/forms/server/default.env です。

A.2.3.3 FRM-92100

原因：

このエラーは、Web サーバーが停止しているときにユーザーがアプリケーションにアクセスすると発生します。

解決策：

Web サーバーが稼働しているかをチェックします。URL `http://servercomputer:portno` を試行します。OC4J のホーム・ページが表示されない場合は、Web サーバーは停止しています。Forms またはサーバーの管理者に連絡して Web サーバーを起動します。

A.2.3.4 FRM-92101

- 不正な作業ディレクトリ

原因:

このエラーは、指定された作業ディレクトリが存在しない場合に発生します。

解決策:

- application.log ファイルで「Unable to switch to Working Directory:<workingDirectory>」のようなログ・メッセージを検索します。存在する場合、これが原因であると考えられます。application.log ファイルは、Oracle Forms が配置されている OC4J インスタンスの application-deployments/formsapp ディレクトリにあります。Forms 構成ファイルを編集して正しい作業ディレクトリを指定します。

- FORMS_TIMEOUT および heartbeat

原因:

このエラーは、Forms アプレットの heartbeat パラメータが FORMS_TIMEOUT より大きな値に設定されている場合に発生する可能性があります。

解決策:

一般に、heartbeat を FORMS_TIMEOUT より大きな値に設定するのは、一定時間非アクティブなアプリケーションをタイムアウトにする場合に限られます。そのときに FRM-92120 が発生します。

この設定が望ましくないアプリケーションには、heartbeat を FORMS_TIMEOUT より小さな値に設定します。

A.2.3.5 FRM-92102

原因:

このエラーは、Web サーバーとクライアント間のネットワーク上の問題が原因で発生する可能性があります。指定されたポートでクライアントはサーバーと通信できません。

解決策:

Forms 構成ファイルにパラメータ networkRetries を追加します。ネットワークの特性と必要性に応じて、値を適切な数値に設定します（たとえば networkRetries=30）。このパラメータでは、Forms クライアントが最終的にタイムアウトになるまで中間層への再接続を試行する回数を指定します。

- ポートのブロック

原因:

networkRetries に適切な値を設定してもこのエラーが発生する場合は、TCP/IP レベルで制限されている Web サーバー上のポートが原因と考えられます。

解決策:

ソケット接続では、接続の両端にポートが必要です。ポートがクローズされていると、通信は停止します。ポートの保護にはファイアウォールとプロキシが使用されます。Web サーバーでポートのブロックを解除するとエラーは解消されます。

A.2.3.6 FRM-92120

原因:

クライアントが中間層で Registry.dat を検出できない場合に発生するサーバー構成エラーです。

解決策:

このエラーが発生した場合は、ファイル Registry.dat が中間層の ORACLE_HOME/forms/java/oracle/forms/registry ディレクトリにあることを確認します。存在しない場合は、このファイルを置く必要があります。

稼働中の Forms アプリケーションでこのエラーが突然発生した場合は、HTTP サーバーがダウンした可能性があります。これを確かめるには、ブラウザで URL `http://myserver.com:NNNN` を入力します。この場合、`myserver.com` をホスト名に、`NNNN` を使用する HTTP サーバーのポート番号に置換します。サーバーに接続できなかったことがブラウザで通知された場合は、HTTP サーバーがダウンしているため、システム管理者にサーバーの起動を依頼する必要があります。

HTTP サーバーが稼働している場合は、ブラウザに URL `http://myserver.com:NNNN` と入力すると、OracleAS の「ようこそ」画面が表示されます。

A.2.3.7 FRM-92150/FRM-92160

原因:

不正なパスおよびコードベース（またはそのいずれか）の設定

解決策:

システム・パスの冒頭で、正しい ORACLE_HOME/bin を設定します。HTML ファイルまたは Forms 構成ファイルの CODEBASE エントリで、古いバージョンの JAR ファイルが指されている可能性があります。構成ファイルの CODEBASE エントリを変更するか、またはコードベース・パスの JAR ファイルを適切な JAR ファイルに置き換えます。

クライアント・コンピュータのユーザー・プロファイル・ディレクトリにある Oracle JAR キャッシュの内容を消去すると、確実に新しい Forms JAR ファイルをダウンロードできます。

A.3 スタック・トレースを使用したサーバーのクラッシュの診断

この項は、次の項目に分かれています。

- 第 A.3.1 項「スタック・トレースについて」
- 第 A.3.2 項「スタック・トレースの構成と使用」

Forms Web ランタイムが予期せず終了した場合は、ORACLE_HOME/forms/trace ディレクトリにスタック・トレースが書き込まれます。ファイル名の形式は、`<forms_runtime_process>_dump_<process id>` です。

ダンプ・ファイルには、実行中のプロセスのスタック・トレースが記録され、Forms で最後に成功した操作が示されます。

A.3.1 スタック・トレースについて

スタック・トレースは、次の 2 つの理由で役立ちます。

- スタック内の情報から既知の問題であると同定できます。100% 信頼できるわけではありませんが、スタック・トレースが同一の場合、問題が同一であると考えられます。同一でなくても、既存のバグに対する迂回策やパッチがあれば、それを試すことができます。
- 問題が既知のバグでない場合、スタックから、正確な原因と開発を支援する有益な情報を得ることができる場合があります。

A.3.2 スタック・トレースの構成と使用

この項は、次の項目に分かれています。

- [第 A.3.2.1 項「環境の確認」](#)
- [第 A.3.2.2 項「Solaris のスタック・トレースについて」](#)
- [第 A.3.2.3 項「Windows のスタック・トレースについて」](#)

A.3.2.1 環境の確認

UNIX または Windows でスタック・トレースをテストするには、環境変数 `FORMS_DELIBERATECRASH` を設定します。名前が示すように、この変数を設定すると Forms ランタイム・プロセスがクラッシュします。Oracle Forms では現在 1 と 2 の設定が認識されます。`FORMS_DELIBERATECRASH` を 1 に設定すると、実行時に BELL ビルトインが実行されたときに Forms がクラッシュします。2 に設定すると、実行時に `when-button-pressed` トリガーが起動されたときに Forms がクラッシュします。前述のスタックは、`FORMS_DELIBERATECRASH` を 2 に設定したときに生成されています。この環境変数は、環境ファイル (`default.env` など) で設定できます。

環境変数 `FORMS_DELIBERATECRASH` を 2 に設定し、このマニュアルのスタックに対して生成されたスタック・トレースをチェックすると、シンボル・ファイルが正しくインストールされているかどうかを確認できます。すべてが正常に機能していることを確認した後に、サーバーがクラッシュする場合は、問題の解決にスタック・トレースが役立ちます。

A.3.2.2 Solaris のスタック・トレースについて

Solaris のスタック・トレースでは、上位 2 つの関数 `siehjmptrm()` および `sigacthandler()` が、シグナル処理コードとしてスタック・トレースに頻繁に出現します。エラーが発生したときにプログラムが実行していた関数を確認するには、さらに下方のスタックを参照する必要があります。

A.3.2.3 Windows のスタック・トレースについて

UNIX と Windows ではスタック・トレースの動作が異なります。UNIX では、シンボル情報が実行可能ファイルおよび共有ライブラリに格納されます。Windows では、バイナリ形式の `.sym` ファイルとしてこの情報がリンク時に取り出されています。すべての Oracle Forms 実行可能ファイルまたは DLL に、`.sym` ファイルが 1 つずつ存在します。Windows プラットフォームのメカニズムとして、クラッシュが発生すると、メモリーにロードされている Forms 実行可能ファイルに対応するすべての `.sym` ファイルが、Forms ランタイム・プロセスによって読み取られます。その後に、`.sym` ファイル内の情報を使用してシンボル名が参照されます。

A.4 クライアント・クラッシュの診断

この項は、次の項目に分かれています。

- [第 A.4.1 項「クライアント・クラッシュの診断について」](#)
- [第 A.4.2 項「ハングしているアプリケーションの診断」](#)

A.4.1 クライアント・クラッシュの診断について

Forms アプレットが予期せず表示されなくなり致命的エラーを通知するダイアログが表示される場合は、Forms アプレットはクラッシュしています。Windows では、クラッシュが発生すると無効な操作を通知するダイアログが表示されるか、タスク・マネージャで「応答なし」のフラグが設定されます。

クラッシュを検証するには、クライアントのスタック・トレース・ファイルをチェックします。クライアントがクラッシュした場合は、実行可能ファイルと同じディレクトリに `.rpt` 拡張子の付いたファイルが作成されます。ファイル名のルートは、実行可能ファイルの名前です。ディレクトリ `c:\jdk\1_3_1\bin` から起動した `AppletViewer` を使用している場合は、クライアントのスタック・トレース・ファイルは `c:\jdk\1_3_1\bin\appletviewer.rpt` となります。

JInitiator を使用している場合は、Web ブラウザが実行可能ファイルと見なされます。ブラウザが Netscape の場合はクライアントのスタック・トレース・ファイルは `netscape.rpt` となり、Internet Explorer の場合は `iexplore.rpt` となります。

アプレットがクラッシュしたように見えるのに、対応する `.rpt` ファイルが見つからないことがあります。この場合は、Oracle Forms Server プロセスとクライアントの接続が予期せず突然切断された可能性があります。アプレットはそのまま稼働されますが、すべての Forms ウィンドウがシャットダウンされたために、クライアントがクラッシュしたように見えます。

A.4.2 ハングしているアプリケーションの診断

クライアントがハングしたように見える場合は、サーバー・プロセスがまだ稼働しているかを確認することが重要です。サーバー・プロセスがクラッシュしていないのにクライアントがユーザーの操作に応じなくなったように見える場合は、アプリケーションがハングしていると考えられます。

このような場合は、スレッド・ダンプにデッドロックが示されている可能性があります。

ダンプ・ファイル内の情報は、オラクル社の開発に非常に役立ちます。問題を報告する場合は、提出するバグにこの情報を含めるようにしてください。

A.4.2.1 アプリケーションがハングする原因

原因の 1 つに、Java クラス・ファイルと Oracle Forms サーバーのバージョンの不適合があげられます。アプレットと Forms Server プロセス間の通信は、メッセージ ID に基づいて行われます。これらのメッセージ ID が同期されていないと、アプレットとサーバーは互いの指示を理解できません。JAR ファイルを使用している場合は、`<ARCHIVE>` タグを削除してみてください。問題が解消されない場合は、インストールまたはパッチの CD-ROM から手動で正しいクラス・ファイルを取得してください。

もう 1 つの原因として、Forms ランタイム・プロセスの終了が考えられます。サーバー上で Forms ランタイム・プロセスが稼働しているかどうか確認します。FORMS_TIMEOUT パラメータが設定されていることを確認します。タイムアウト機能は、ハートビートのように機能し、Oracle Forms クライアントがサーバーに対し一定の間隔で ping を実行することを前提として、Forms クライアントから指定された時間内にアクティビティが実行されなかった場合のみ、Oracle Forms Server プロセスをクリーンアップします。この機能の主な目的は、サーバー・プロセスの孤立を防止することですが、サーバー・プロセスが早い段階で不必要にクリーン・アップされないようにする効果もあります。

A.4.2.2 Java スレッド・ダンプの作成

スタック・ダンプは AppletViewer から取得できます。そのためには、AppletViewer を起動したコマンド・プロンプトで `[Ctrl]+[Break]` を押します。

JInitiator の場合は、Java コンソールにダンプ情報の出力方法が示されます。画面の指示に従って情報を表示してください。

スレッド・ダンプ内の情報は、オラクル社の開発でコード内の問題を特定するのに役立ちます。問題を報告する場合は、提出するあらゆるバグにスレッド・ダンプを含めるようにしてください。

A.5 Forms Trace と Servlet Logging Tools

Forms Trace および Servlet Logging の 2 つのツールは、Oracle Forms の環境をトラブルシューティングの際に使用します。Forms Trace の構成と使用の詳細は、[第 8.1 項「Forms Trace について」](#) および [第 8.7 項「Servlet Logging Tools」](#) を参照してください。

A.6 メモリーの問題の解決

この項は、次の項目に分かれています。

- 第 A.6.1 項「Java のメモリーの使用方法」
- 第 A.6.2 項「初期 Java ヒープの設定」
- 第 A.6.3 項「メモリー・リークについて」
- 第 A.6.4 項「キャッシングによるパフォーマンスの向上」
- 第 A.6.5 項「OJDK でのパフォーマンス向上」

A.6.1 Java のメモリーの使用方法

すべてのソフトウェア・プログラムと同様に、Java アプレットもメモリーを使用します。Java の言語仕様ではガベージ・コレクタを必要とします。これは Java 仮想マシン (JVM) の内部メモリーを管理するものです。メモリーが必要になると Java プログラムは JVM にこのメモリーを要求します。メモリーが残っていない場合は、JVM がガベージ・コレクタを使用してある程度のメモリーを解放しようとしています。ガベージ・コレクタは、プログラムで不要となったメモリーを解放して JVM に戻そうとしています。それでも必要なタスクの実行にメモリーが足りない場合は、JVM がオペレーティング・システムからさらにメモリーを取得しようとしています。このメモリーの割当てが失敗すると、Java プログラムは処理を続行できません。

A.6.2 初期 Java ヒープの設定

アプリケーションの初期 Java ヒープ (JVM が使用するメモリー) を、Enterprise Manager または中間層のコマンドラインで指定することができます。たとえば、次のコマンドは初期 Java ヒープを 20MB に設定し、JVM が使用する最大メモリーを 32MB に設定します。

```
appletviewer -J-ms20m -J-mx32Mb dve.html
```

AppletViewer のデフォルトの初期サイズと最大サイズは、それぞれ 16MB と 20MB です。

JInitiator を使用している場合は、JInitiator コントロール・パネルでランタイム・オプションを設定する必要があります。

注意： JVM は、使用を許可されているメモリーのみを使用します。オペレーティング・システムに使用可能なメモリーがあっても、JVM はそれを使用しません。

A.6.3 メモリー・リークについて

メモリー・リークとは、ストアを動的に割り当てるプログラムのロジック・エラーです。このエラーによって、不要となったメモリーを回収できなくなり、メモリーが枯渇して最終的に障害が発生します。

たとえば、プログラムの実行において特定のタスクの実行にメモリーの割当てが必要な場合があります。プログラムがメモリーを使用して作業を終了し、不要になったメモリーをコンピュータ上の他のプログラムに解放できなかった場合は、メモリーはリークしたと判断されます。

メモリー・リークを発見するための一般的な方法は、一連のステップを繰り返し実行してアプリケーションによるメモリーの使用状況を監視します。実行を繰り返すたびにメモリー使用量が増えていく場合は、プログラムでメモリー・リークが発生していると考えられます。

ただし、後から再利用できるように、前に割り当てられたメモリーを保持する複雑なアプリケーションもあります。メモリーの割当ては大きな負荷がかかるため、後でさらにメモリーが必要であると予測されるとき、使用されていないメモリーを再利用できるように保持しておく方が効率的な場合があります。

A.6.3.1 Java でのメモリー・リーク

Java 言語仕様では、JVM にガベージ・コレクタ (GC) を実装することを必要としています。Java では、プログラマが新しいオブジェクトを作成することでメモリーを割り当てます。そのメモリーの割当てを解除する方法はありません。ガベージ・コレクタは、プログラムに割り当てられたメモリーを定期的にすべて検索し、安全に破棄できるオブジェクトを特定してメモリーを解放します。安全に破棄できるオブジェクトを特定するために、ガベージ・コレクタは **mark and sweep** アルゴリズムを使用します。ガベージ・コレクタは、オブジェクトに動的に割り当てられたメモリーをスキャンし、アクティブな参照を持つオブジェクトをマークします。

オブジェクトへのパスがすべて調査された後に、不要と認定されたマークのないオブジェクトがガベージとして収集されます。一般に、Java プログラミングでは、ガベージ・コレクタがあればメモリー・リークが発生しないとされていますが、そのようなことはありません。ガベージ・コレクタは、単にアクティブな参照を持つオブジェクトにマークを付け、そうでないオブジェクトを破棄します。不要となったオブジェクトがアクティブな参照を持っている可能性もあります。これが Java におけるメモリー・リークです。リークを解決するには、不要となったオブジェクトへの参照を破棄して、ガベージ・コレクタがそのオブジェクトを安全に破棄できる対象と判断できるようにします。Java プログラムにメモリー・リークが存在する場合は、ガベージ・コレクタを頻繁にコールしても意味がありません。

問題をさらに複雑にしているのが、JVM が使用されていないメモリーをオペレーティング・システムにあえて戻さない場合がある点です。しかしほとんどのプログラムは追加のメモリーが必要なときに JVM 内の空きメモリーを再使用するため、実際にはこれはほとんど問題となりません。ただし、JVM に割り当てられたすべてのメモリーが、JVM で稼働中のプログラムによって使用されるわけではないことに注意してください。

A.6.3.2 メモリー・リークの特定

通常、ある一連の操作を実行するたびにメモリー使用量が増えていく場合は、メモリー・リークが発生しています。適切な判別方法は次のとおりです。

1. フォームを初期の基本状態にして、メモリー使用量を記録します。
2. 一連の手順を実行して問題を再現します。
3. 初期の基本状態に戻り、メモリー使用量を記録します。

ステップ 2 と 3 を繰り返すことで、継続的にメモリー・リークが発生しているかどうかを判断できます。何度繰り返しても使用量があまり伸びない場合は、リークではなく、JVM が使用されていないメモリーを保持しているか、あるいはガベージ・コレクタが予定の頻度で動作していない可能性があります。

A.6.4 キャッシングによるパフォーマンスの向上

Java プログラムの実行時に、Java 仮想マシンはクラス・ファイルをロードする必要があります。インターネット上で実行する場合は、プログラムが実行されるたびにクラス・ファイルをダウンロードすると時間がかかってパフォーマンスに支障をきたすことがあります。このダウンロードの問題を解決するために、JDK では Java アーカイブ (JAR) ファイルがサポートされています。JAR ファイルは簡単に言えば、一連のクラス・ファイルを 1 つの圧縮ファイルにまとめたものです。一般に、JAR ファイルのサイズは、その中に含まれているクラス・ファイルのサイズの合計よりもはるかに小さくなります。

JAR ファイルを使用すると、転送されるデータ量を削減できるだけでなく、JInitiator および Oracle の JDK でのキャッシングと呼ばれるプロセスが使用可能になります。OJDK 1.1.7.15 以降、キャッシング・プロセスのパフォーマンスは何度か改善されています。

JVM は、最初にクラスを参照する際に、ローカル・コンピュータにキャッシュされている JAR ファイルにそのクラスが存在するかどうかを確認します。事前にキャッシュされた JAR ファイルのいずれかにそのクラスが存在する場合、JVM はキャッシュ内の JAR ファイルより新しいバージョンがアプリケーション・サーバーに存在するかどうかを確認します。新しい JAR ファイルが存在する場合は、JAR ファイルの新しいコピーがクライアント・キャッシュにダウンロードされます。キャッシュされている JAR ファイルが最新のものである場合は、ネットワークからではなくキャッシュされている JAR ファイルからクラス・ファイルがロードされます。

キャッシングは重要な機能です。アプリケーションの 1 回目の実行に必要なすべての JAR ファイルがクライアントにキャッシュされます。アプリケーションの JAR ファイルに変更がなければ、それ以降のアプリケーションの起動では、常にローカルにキャッシュされたコピーからクラスがロードされるようになります。これによって、アプリケーションの起動時間のパフォーマンスが大幅に向上します。アプリケーションの特定の部分の実行に新しいクラスが必要な場合は、必要に応じてそれらがダウンロードされます。

A.6.5 OJDK でのパフォーマンス向上

キャッシングによって、アプリケーションが起動されるたびにサーバーから JAR ファイルがダウンロードされなくなりますが、それでもパフォーマンスに影響を与える問題は残っています。JAR ファイルには圧縮データが含まれるため、ネットワークからダウンロードするデータが少ない場合は、キャッシュされた JAR ファイルからデータを解凍する時間の方がダウンロード時間より多くかかります。

JAR ファイルはネットワークでの伝送中に変更されていないことを保証するため、デジタル署名で認証が行われる場合があります。この手順には、コストのかかる数学上の計算が伴います。OJDK で新たに導入されたキャッシング・メカニズムでは、これらの問題に次のように対処しています。

JAR ファイルが最初にダウンロードされるときに、2つのファイルが作成されます。

- データ・ファイル。JAR ファイルのすべての解凍データが含まれます。データ・ファイルの拡張子は .dxx です。この xx は 00 から 99 までの数値です (例: 10f756b8.d00)。
- 索引ファイル。JAR ファイルに関する情報として、ロード元の URL、ファイルがサーバーで最後に変更された日付、および目次が含まれます。目次には、JAR ファイルのすべてのエントリ、データ・ファイルでのそれらのオフセット、および各エントリの認証ステータスがリストされます。索引ファイルの拡張子は .ixx です。この xx は 00 から 99 までの数値です (例: 10f756b8.i00)。

これらのファイルでは、情報がバイナリ形式で格納されます。これらのファイルは目視で読むことは困難で、またそれを行う意味もほとんどありません。

キャッシュ・ファイルの最初の 8 文字は、JAR ファイルのダウンロード元である URL を表します。これによって、キャッシング・メカニズムによって対応する 8 文字の表現に URL をマッピングし、その名前でファイルを検索することで、キャッシュ内で容易に URL を検出できるようになります。

アプリケーションでクラス・ファイルが必要になると、OJDK は目次の情報を使用して、データ・キャッシュ内でクラス・ファイルを見つけます。データがデジタル署名されている場合は、認証された署名者の一覧がデータ・ファイルから読み取られます。

A.7 トラブルシューティングのヒント

以降のトラブルシューティング一覧は、複雑な問題の対処に役立つ場合がありますが、問題を解決するための決定的なガイドではなく、また Oracle Forms 環境に対する解決策として保証されているわけでもありません。

順序立てて対処する

原因と思われる箇所に直感や推測ですぐに着手するのではなく、まず他の可能性を排除します。人は証拠から判断可能なことに集中せずに、自らの仮説を裏付ける証拠を探すために長い時間を費やしてしまいがちです。

ささいな事項や明白な事項を見落とさないでください。

問題を分割する

- 問題を管理しやすいように分割することで、全体を調査する必要がなくなります。ある部分を調査して問題がないことがわかれば、次の部分に進むことができます。問題を重要な部分別に分けるアプローチは、問題の診断において有用です。オラクル社カスタマ・サポート・センターと問題を協議して解決策を得る必要がある場合は、このアプローチは重要です。

- 具体的な現象、および発生タイミングと頻度を明確にします。これと同様に、発生しなかった現象や発生しなかったタイミングなどを把握することも重要です。たとえば、同じ建物内のユーザー・グループ全体に問題が発生し、その発生が常に午前9時から午前10時の間である場合は、別の建物でその問題が再現しないか、あるいは午前10時以降は再現しないかを把握することも重要です。ユーザーが特定のフォームを午前9時から午前10時の間にのみ使用しているか、あるいはこの時間帯にシステムの負荷が最も高くなっていることが考えられます。

エラー・メッセージを確認する

当たり前のようですが、解決策がエラー・テキストに記述されていることが少なくありません。このマニュアルは、エラー・メッセージを理解して対策を特定する際に役立ちます。

可能であれば問題を再現する

問題を自ら再現できれば、エンド・ユーザーが発見できなかった動作に気付くことができます。その問題が常に発生していたために、エンド・ユーザーがそれを意図された動作であると考えている可能性があります。問題を再現できれば、その解決に一步踏み出したこととなります。

使用するツールを確実に理解する

診断ツールを使用する場合は、その使用方法、およびそのデータの分析方法を把握しておいてください。問題が発生する前にツールの使用方法を理解しておくことは時間の無駄ではありません。ツールを習得するための時間も確保してください。

A.8 さらに調査が必要な場合

ここまでの情報が不十分である場合は、*Oracle MetaLink* (<http://metalink.oracle.com/>) でさらに解決策を検索できます。問題の解決策が見つからない場合は、サービス・リクエストを作成してください。

関連項目：

- Oracle Technology Network から入手可能な Oracle Application Server のリリース・ノート：
<http://www.oracle.com/technology/documentation/appserver101202.html>

B

JInitiator

この章では、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 (frmall.jar および frmall_jinit.jar) が提供されます。frmall.jar は標準のJARファイルであり、frmall_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。

B.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 のほか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 10 章の「[アプリケーションの起動時間の最小化](#)」の項を参照)。

B.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。

B.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・コンピュータ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。

B.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。

B.4.1 Windows 98、NT、2000 および XP

- Netscape Navigator 4.7.x
- Netscape Navigator 7.x
- Internet Explorer 5.x
- Internet Explorer 6.0

B.5 システム要件

Oracle JInitiator をインストールして実行するため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)

注意： この最小システム要件は JInitiator のみの場合です。Oracle Forms の実行には不十分です。

JInitiator とともに Oracle Forms を正常に実行するための最小システム要件を次に示します。

- Windows 98、NT、2000 および XP
- 900MHz 超の Pentium プロセッサ
- 90MB 以上の RAM (128MB を推奨)

B.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 パッケージをダウンロードし、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。

B.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。

B.8 Oracle JInitiator プラグインの設定

サーバーでのテストのみを目的として Oracle JInitiator プラグインをサーバーにインストールし、設定できます。その後に次の作業を行います。

- [Oracle JInitiator マークアップのベース HTML ファイルへの追加](#)
- [Oracle JInitiator ダウンロード・ファイルのカスタマイズ](#)
- [Oracle JInitiator をダウンロード可能にする](#)

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

B.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・ファイルの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。

B.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。

B.9 Oracle JInitiator プラグインの変更

引き続き、次の作業を行い Oracle JInitiator プラグインを変更します。

- [Oracle JInitiator キャッシュ・サイズの変更](#)
- [Oracle JInitiator ヒープ・サイズの変更](#)
- [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 出力の表示](#)

B.9.1 Oracle JInitiator キャッシュ・サイズの変更

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

Oracle JInitiator 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。

B.9.2 Oracle JInitiator ヒープ・サイズの変更

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

Oracle JInitiator 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。

B.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、このボックスを選択しないでください。次に、「プロキシのアドレス」フィールドにプロキシ・サーバーのホスト名を入力します。

B.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」を選択します。

B.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 C.3 項「[base.htm ファイル](#)、[basejini.htm ファイル](#)、および [basejpi.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。

- 第 C.1 項「デフォルトの formsweb.cfg ファイル」
- 第 C.2 項「プラットフォーム固有の default.env ファイル」
- 第 C.3 項「base.htm ファイル、basejini.htm ファイル、および basejpi.htm ファイル」
- 第 C.4 項「web.xml」
- 第 C.5 項「forms.conf」
- 第 C.6 項「Registry.dat」
- 第 C.8 項「デフォルトの webutil.cfg」
- 第 C.9 項「デフォルトの webutilbase.htm」
- 第 C.10 項「デフォルトの webutiljini.htm」
- 第 C.11 項「デフォルトの webutiljpi.htm」

C.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 (firmservlet)
# 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/forms/firmservlet?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: delimiter for parameters in the base HTML files
HTMLdelimiter=%
# System parameter: working directory for Forms runtime processes
# WorkingDirectory defaults to <oracle_home>/forms if unset.
workingDirectory=
# System parameter: file setting environment variables for the Forms runtime processes
envFile=default.env
# 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=/forms/lservlet
# Forms applet parameter
codebase=/forms/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=frmall_jinit.jar
# Forms applet archive setting for other clients (Sun Java Plugin, Appletviewer,
etc)
archive=frmall.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=/forms/jinitiator/us/jinit_download.htm
# Parameter related to the version of JInitiator
jinit_classid=clsid:CAFECAFE-0013-0001-0022-ABCDEFABCDEF
# Parameter related to the version of JInitiator
jinit_exename=jinit.exe#Version=1,3,1,22
# Parameter related to the version of JInitiator
jinit_mimetype=application/x-jinit-applet;version=1.3.1.22
# 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/archive/j2se/1.4.2_06/index.html
# Parameter related to the version of the Java Plugin
jpi_classid=clsid:CAFEEFAC-0014-0002-0006-ABCDEFEDCBA
# Parameter related to the version of the Java Plugin
jpi_codebase=http://java.sun.com/products/plugin/autodl/
jinstall-1_4_2-windows-i586.cab#Version=1,4,2,06
# Parameter related to the version of the Java Plugin
jpi_mimetype=application/x-java-applet;jpi-version=1.4.2_06
# 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=%OID_FORMSID%
# Single Sign-On OID configuration parameter
oracle_home=%ORACLE_HOME%
# Single Sign-On OID configuration parameter
formsid_group_dn=%GROUP_DN%
# 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
# EndUserMonitoring
# EndUserMonitoringEnabled parameter
# Indicates whether EUM/Chronos integration is enabled
EndUserMonitoringEnabled=
# EndUserMonitoringURL
# indicates where to record EUM/Chronos data
EndUserMonitoringURL=
# 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/forms/frmservlet?config=sepwin".
[sepwin]
separateFrame=True
lookandfeel=Generic
# Example Named Configuration Section
# Example 2: configuration forcing use of the Java Plugin in all cases (even if
# the client browser is on Windows)
[jpi]
baseHTMLJInitiator=basejpi.htm
# Example Named Configuration Section
# Example 3: configuration running the Forms ListenerServlet in debug mode
# (debug messages will be written to the servlet engine's log file).
[debug]
serverURL=/forms/lervlet/debug
# Sample configuration for deploying WebUtil. Note that WebUtil is shipped with
# DS but not AS and is also available for download from OTN.
[webutil]
WebUtilArchive=frmwebutil.jar,jacob.jar
WebUtilLogging=off
WebUtilLoggingDetail=normal
WebUtilErrorMode=Alert
WebUtilDispatchMonitorInterval=5
WebUtilTrustInternal=true
WebUtilMaxTransferSize=16384
baseHTMLjinitiator=webutiljini.htm
baseHTMLjpi=webutiljpi.htm
archive_jini=frmall_jinit.jar
archive=frmall.jar
lookAndFeel=oracle

```

C.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 Solaris 用 default.env ファイル

C.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=%FORMS_ORACLE_HOME%

```

```
#
# Search path for Forms applications (.fmx files, PL/SQL libraries)
# If you need to include more than one directory, they should be semi-colon
# separated (e.g. c:¥test¥dir1;c:¥test¥dir2)
#
FORMS_PATH=%FORMS_ORACLE_HOME%¥forms
# webutil config file path
WEBUTIL_CONFIG=%FORMS_ORACLE_HOME%¥forms¥server¥webutil.cfg
# Disable/remove this variable if end-users need access to the query-where
# functionality which potentially allows them to enter arbitrary SQL
# statements when in enter-query mode.
FORMS_RESTRICT_ENTER_QUERY=TRUE
#
# 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=%FORMS_ORACLE_HOME%¥bin;%FORMS_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.
#FORMS_TRACE_PATH=<FORMS_ORACLE_HOME>¥forms¥server
#
# System settings
# -----
# You should not normally need to modify these settings
#
FORMS=%FORMS_ORACLE_HOME%¥forms
#
# Java class path
# This is required for the Forms debugger
# You can append your own Java code here)
# fmrsvr.jar, repository.jar and ldapjclnt10.jar are required for
# the password expiry feature to work (#2213140).
#
CLASSPATH=%FORMS_ORACLE_HOME%¥j2ee¥OC4J_BI_Forms
¥applications¥formsapp¥formsweb¥WEB-INF¥lib¥fmrsvr.jar;
%FORMS_ORACLE_HOME%¥jlib¥repository.jar;%FORMS_ORACLE_HOME%
¥jlib¥ldapjclnt10.jar;%FORMS_ORACLE_HOME%¥jlib¥debugger.jar;%FORMS_ORACLE_HOME%
¥jlib¥ewt3.jar;%FORMS_ORACLE_HOME%¥jlib¥share.jar;%FORMS_ORACLE_HOME%
```

```

¥jlib¥utj.jar;%FORMS_ORACLE_HOME%¥jlib¥zrclient.jar;%FORMS_ORACLE_HOME%
¥reports¥jlib¥rwrun.jar;%FORMS_ORACLE_HOME%¥forms¥java¥frmwebutil.jar

```

C.2.2 デフォルトの Solaris 用 default.env ファイル

```

# default.env - default Forms environment file, Solaris version
#
# This file is used to set the Forms runtime environment parameters.
# If a parameter is not defined here, the value used will be that defined
# in the environment in which the servlet engine (OC4J or JServ) was started.
#
# NOTES
# 1/ The Forms installation process should replace all occurrences of
# <percent>FORMS_ORACLE_HOME<percent> with the correct ORACLE_HOME
# setting, and all occurrences of <percent>O_JDK_HOME<percent> with
# the location of the JDK (usually $ORACLE_HOME/jdk).
# Please make these changes manually if not.
# 2/ Some of the variables below may need to be changed to suite your needs.
# Please refer to the Forms documentation for details.
#
ORACLE_HOME=%FORMS_ORACLE_HOME%
#
# Search path for Forms applications (.fmx files, PL/SQL libraries)
#
FORMS_PATH=%FORMS_ORACLE_HOME%/forms
# webutil config file path
WEBUTIL_CONFIG=%FORMS_ORACLE_HOME%/forms/server/webutil.cfg
# Disable/remove this variable if end-users need access to the query-where
# functionality which potentially allows them to enter arbitrary SQL
# statements when in enter-query mode.
FORMS_RESTRICT_ENTER_QUERY=TRUE
# Java class path
# This is required for the Forms debugger
# You can append your own Java code here)
# fmsrv.jar, repository.jar and ldapjclnt10.jar are required for
# the password expiry feature to work(#2213140).
#
CLASSPATH=%FORMS_ORACLE_HOME%/j2ee/OC4J_BI_Forms
/applications/formsapp/formsweb/WEB-INF/lib/fmsrv.jar:%FORMS_ORACLE_HOME%
/jlib/repository.jar:%FORMS_ORACLE_HOME%/jlib/ldapjclnt10.jar:%FORMS_ORACLE_HOME%
/jlib/debugger.jar:%FORMS_ORACLE_HOME%/jlib/ewt3.jar:%FORMS_ORACLE_HOME%
/jlib/share.jar:%FORMS_ORACLE_HOME%/jlib/utj.jar:%FORMS_ORACLE_HOME%
/jlib/zrclient.jar:%FORMS_ORACLE_HOME%/reports/jlib/rwrun.jar:%FORMS_ORACLE_HOME%
/forms/java/frmwebutil.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=%FORMS_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.
# We no longer need to set REPORTS_CLASSPATH as forms will
# always start the JVM before calling reports.
#
#

```

```

# 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.
#FORMS_TRACE_PATH=<FORMS_ORACLE_HOME>/forms/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
#   (.../jre/lib/sparc)
# - Forms needs two paths to the jre, for libjvm.so and libhpi.so
# - In JDK 1.4.1 the location of libjvm.so is lib/sparc (there is no
#   classic directory) so we do not include the .../classic directory
#   below. There are other versions of libjvm.so (in directories server,
#   client and hotspot) but we will use the version in lib/sparc for now.
#
LD_LIBRARY_PATH=%FORMS_ORACLE_HOME%/lib32:%O_JDK_HOME%/jre/lib/sparc:
%O_JDK_HOME%/jre/lib/sparc/native_threads:%LD_LIBRARY_PATH%

```

C.3 base.htm ファイル、basejini.htm ファイル、および basejpi.htm ファイル

base.htm、basejini.htm、および basejpi.htm の概要とディレクトリは、[第 3.2.1.3 項「base.htm、basejini.htm、および basejpi.htm」](#) を参照してください。

Oracle Application Server 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、「[Application Server Control コンソール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： baseHTML 変数を変更するには、[Application Server Control コンソール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/forms/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。

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、basejpi.htm または base.htm のパラメータを更新します。
2. ORACLE_HOME/forms/server ディレクトリにある、basejini.htm、basejpi.htm または base.htm の初期ファイルをコピーします。
3. ファイル名を変更します（たとえば、order.htm）。
4. ユーザーに表示されるテキストを追加または変更します（たとえば、<TITLE> タグおよび <BODY> タグ内のテキスト）。
5. 必要に応じてパラメータを変更します。ベース HTML ファイルの変数を使用し、「[formsweb.cfg](#)」での説明のように、formsweb.cfg ファイルの実際の値を指定してください。

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

C.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。次に例を示します。

```
<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。

C.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。

C.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%">
<PARAM NAME="EndUserMonitoringEnabled" VALUE="%EndUserMonitoringEnabled%">
<PARAM NAME="EndUserMonitoringURL" VALUE="%EndUserMonitoringURL%">
<PARAM NAME="heartbeat" VALUE="%heartbeat%">

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

%HTMLafterForm%

</BODY>
</HTML>

```

C.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="/forms/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%">
<PARAM NAME="EndUserMonitoringEnabled" VALUE="%EndUserMonitoringEnabled%">
<PARAM NAME="EndUserMonitoringURL" VALUE="%EndUserMonitoringURL%">
<PARAM NAME="heartbeat" VALUE="%heartbeat%">
<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%"
  EndUserMonitoringEnabled="%EndUserMonitoringEnabled%"
  EndUserMonitoringURL="%EndUserMonitoringURL%"
  heartBeat="%heartBeat%"
>
</EMBED>
</COMMENT>
</NOEMBED></EMBED>
</OBJECT>
<!-- Forms applet definition (end) -->

%HTMLafterForm%

</BODY>
</HTML>

```

C.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%">
<PARAM NAME="EndUserMonitoringEnabled" VALUE="%EndUserMonitoringEnabled%">
<PARAM NAME="EndUserMonitoringURL" VALUE="%EndUserMonitoringURL%">
<PARAM NAME="heartBeat" VALUE="%heartBeat%">
<COMMENT>
<EMBED SRC="" PLUGINSOURCE="%jpi_download_page%"
        TYPE="%jpi_mimetype%"
        java_codebase="%codebase%"
        java_code="oracle.forms.engine.Main"
        java_archive="%archive%"
        WIDTH="%Width%"
        HEIGHT="%Height%"
        HSPACE="0"
        VSPACE="0"
        serverURL="%serverURL%"
        networkRetries="%networkRetries%"
        serverArgs="%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%"
        EndUserMonitoringEnabled="%EndUserMonitoringEnabled%"
        EndUserMonitoringURL="%EndUserMonitoringURL%"
        heartBeat="%heartBeat%"
    >
</NOEMBED>
</COMMENT>
</NOEMBED></EMBED>
</OBJECT>
<!-- Forms applet definition (end) -->

%HTMLafterForm%

</BODY>
</HTML>

```

C.4 web.xml

web.xml の説明とディレクトリは、第 3 章の「web.xml」の項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 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 frmServlet の testMode パラメータを true に設定します。次に Web サーバー（または OC4J）を再起動します。追加のオプションが Forms Servlet の管理ページ (http://<your_web_server_hostname>:<port>/forms/frmServlet/admin などの URL でアクセス可能) に表示されます。

- 標準 (ORACLE_HOME/forms/server/formsweb.cfg) 以外の Forms Servlet 構成ファイルを使用する。

frmServlet の configFileName サブレット・パラメータをコメント・アウトして変更します。

- (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)。

C.4.1 デフォルトの web.xml ファイル

```

<?xml version="1.0"?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN"
"http://java.sun.com/j2ee/dtds/web-app_2_2.dtd">
<!-- $Id: web.xml 29-apr-2004.13:43:19 ahousing Exp $
Name
    web.xml
Purpose
    Forms web application (WAR) configuration file
-->
<web-app>

```

```

<display-name>Forms Services</display-name>
<description>Oracle AS: Forms Services</description>
<welcome-file-list>
  <welcome-file>lervlet</welcome-file>
</welcome-file-list>
<!-- Forms page generator servlet -->
<servlet>
  <servlet-name>firmservlet</servlet-name>
  <servlet-class>oracle.forms.servlet.FormsServlet</servlet-class>
  <load-on-startup>1</load-on-startup>
  <!-- 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>/forms/server/formsweb.cfg.
  You can override that value here by editing and uncommenting the
  following servlet parameter setting: -->
  <!--
  <init-param>
    <param-name>configFileName</param-name>
    <param-value><your configuration file name goes here></param-value>
  </init-param>
  -->
  <init-param>
    <!-- Turn on or off sensitive options on the firmservlet/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>lervlet</servlet-name>
  <servlet-class>oracle.forms.servlet.ListenerServlet</servlet-class>
</servlet>
<!-- Forms servlet mappings. Allow these paths to the servlets:
  /forms/firmservlet or /forms/firmservlet/*: FormsServlet
  /forms/lervlet or /forms/lervlet/*: ListenerServlet
-->
<servlet-mapping>
  <servlet-name>firmservlet</servlet-name>
  <url-pattern>/firmservlet*</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>lervlet</servlet-name>
  <url-pattern>/lervlet*</url-pattern>
</servlet-mapping>
<!-- The following context parameter is only defined here so it can be
  overridden by the (site-specific) value in the orion-web.xml file.
-->
<context-param>
  <param-name>configFileName</param-name>
  <param-value></param-value>
</context-param>
</web-app>

```

C.5 forms.conf

forms.conf の説明とディレクトリは、[第 3.2.3.1 項「forms.conf」](#) を参照してください。

[表 C-1](#) 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 C-1 forms.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
/forms/java	別名	ORACLE_HOME/forms/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。
/forms/html	別名	ORACLE_HOME/tools/frmweb/html	runform.htm へのアクセス (テスト用にフォームを実行します)。
/forms/jinitiator	別名	ORACLE_HOME/jinit	Oracle JInitiator のダウンロード。
/forms/frmservlet	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。
/forms/lervlet	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。

C.5.1 デフォルトの forms.conf

```
# Name
# forms.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 "forms" servlet zone with properties file forms.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 ^/forms/java/(.*) "%FORMS_ORACLE_HOME%/forms/java/$1"
# Virtual path for JInitiator downloadable executable and download page
AliasMatch ^/forms/jinitiator/(.*) "%FORMS_ORACLE_HOME%/jinit/$1"
# Virtual path for runform.htm (used to run a form for testing purposes)
AliasMatch ^/forms/html/(.*) "%FORMS_ORACLE_HOME%/tools/web/html/$1"
# Virtual path for webutil
AliasMatch ^/forms/webutil/(.*) "%FORMS_ORACLE_HOME%/forms/webutil/$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 /forms/frmservlet
    # and /forms/lervlet respectively.
    # We map frmservlet to servlet.frm, and lervlet to servlet.frmml.
    # The apJServAction directives (below) will then remap those.
    AliasMatch ^/forms/frmservlet(.*) "/servlet.frm"
    AliasMatch ^/forms/lervlet(.*) "/servlet.frmml"
    ApJServMount /forms/servlet /forms
    #
    # Let the servlets be called by file extension (e.g /servlet.frm)
    #
    ApJServAction .frm /forms/servlet/frmservlet
    ApJServAction .frmml /forms/servlet/lervlet
```

```

# Prevent access to the Forms Servlets by paths other than
# /forms/frmservlet and /forms/lervlet.
# 1. Prevent access via the .frm and .frml file extensions:
<LocationMatch ^.*%.frm.*>
    order deny,allow
    deny from all
</LocationMatch>
# 2. Stop access by class (by paths like
#   /forms/servlet/oracle.forms.servlet.FormsServlet)
<LocationMatch ^/forms/servlet/oracle%.forms.*>
    order deny,allow
    deny from all
</LocationMatch>
</IfModule>
</IfModule>

# Config. for OC4J
<IfModule mod_oc4j.c>
    Oc4jMount /forms          OC4J_BI_Forms
    Oc4jMount /forms/frmservlet OC4J_BI_Forms
    Oc4jMount /forms/frmservlet/* OC4J_BI_Forms
    Oc4jMount /forms/lervlet    OC4J_BI_Forms
    Oc4jMount /forms/lervlet/*  OC4J_BI_Forms
</IfModule>

```

C.6 Registry.dat

Registry.dat の説明とディレクトリは、[第 3.2.4.1 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（「[アプリケーション・アイコンの配置](#)」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,Dialog,Dialog,Serif,Serif,Dialog,SansSerif

```

Windows には Java に対応しないフォントもあります。そのため、対応しないフォントが見つかったときに表示される Java 対応のフォントを指定する（マップさせる）ことができます。前述の例では、default.fontMap.appFontnames の各フォントは、default.fontMap.javaFontnames のフォントに対応しています。その他の例は、「[デフォルトの Registry.dat](#)」を参照してください。

C.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,
Dialog,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

```

C.7 デフォルトの jvmcontroller.cfg

```

# Default JVM Controller
# This section defines the default values for jvm controllers
# under this Oracle Home. These values override the defaults
# for the dejvm executable.
[default]
# Example JVM Controller
# This section shows example values for a jvm controller. These
# value override any values defined for the default controller.
[example]
jvmoptions=-Xms512m -Xmx1024m
# Classpath settings given here is an example only. This should be
# modified to include the required jar files and should be set in
# platform specific manner.
classpath=/myapps/common/jars/common.jar:/myapps/anapp/jars/anapp.jar
maxsessions=50
logdir=/myapps/anapp/log
logging=off

```

C.8 デフォルトの webutil.cfg

```

# -----
# webutil.cfg - WebUtil default configuration file
# -----
# This file provides all of the configuration settings for webutil. These are
# divided into the following sections:
# 1. Logging Options
# 2. Installation Options
# 3. FileUpload and Download Options

# 1. Server Side Logging Options for logging errors and log messages
# You must set logging.enabled to true to allow mid-tier logging. Without this
# mid-tier logging will not take place no matter what PL/SQL or URL options
# are supplied to switch it on. Once logging is enabled the other settings come
# into play.
#
# Details
# -----
# logging.file      : Defines the file name and location of the log file.
#                   Note that WebUtil does no log file management. You may
#                   need to manually clean this file up from time to time.
# logging.enabled   : Can be TRUE or FALSE
# logging.erroronly : Can be TRUE or FALSE. Setting to true will ensure that
#                   only errors and not normal informational log messages
#                   are written to the log file. For product use this would
#                   normally be set to TRUE
# logging.connections: Can be TRUE or FALSE. Setting to true will cause each
#                   connection from a client using WebUtil to write into
#                   the log as it sets up.
logging.file=
logging.enabled=FALSE
logging.erroronly=FALSE
logging.connections=FALSE
# 2. Installation Options
# WebUtil needs to download some files to the client in order to perform
# certain integration operations such as OLE or Registry Access. These files
# are downloaded the first time that you access one of the functions that need
# them. You have to define the location of these files on the server
#
# Details
# -----
# install.syslib.location : The virtual path to the directory holding the
#                           webutil library files on the server side. This
#                           must either be an absolute URL or a URL that is
#                           relative to the documentbase
#
# install.syslib.<os>.<package>.<n> :
#                           The name(s) of the libraries required for
#                           particular webutil beans. The format of this is
#                           name|size|version|showDownloadDialog. Multiple
#                           libraries can be downloaded per package. But
#                           ensure that the <n> values are consecutive and
#                           start at 1
install.syslib.location=/webutil
# Change size and version if necessary, like when upgrading the library.
# Normally this would not be required since most of these libraries come with
# install itself.
install.syslib.0.7.1=jacob.dll|94208|1.0|true
install.syslib.0.9.1=JNIsharedstubs.dll|65582|1.0|true
install.syslib.0.9.2=d2kwut60.dll|192512|1.0|true
# You can also add your own libraries in here, e.g.
#install.syslib.0.user.1=testwebutil.dll|204872|1.0|true

```

```

# 3. Upload / Download options
# For the file upload and download options you can define the default locations
# on the server that weutil can use as a work area. Optionally you can switch
# upload and download off
#
# Details
# -----
# transfer.database.enabled : Can be TRUE or FALSE - allows you to disable
#                             upload and download from the database server.
# transfer.appsrv.enabled   : Can be TRUE or FALSE - allows you to disable
#                             upload and download from the application
#                             server.
# transfer.appsrv.workAreaRoot: The root of the location in which WebUtil can
#                             store temporary files uploaded from the client.
#                             If no location is specified, Application Server
#                             user_home/temp will be assumed.
#                             This location is always readable and writable
#                             no matter what the settings in
#                             transfer.appsrv.* are. This setting is
#                             required if you need the Client side
#                             READ/WRITE_IMAGE_FILE procedures.
# transfer.appsrv.accessControl: Can be TRUE or FALSE - allows you to indicate
#                             that uploads and downloads can only occur from
#                             the directories named in the
#                             transfer.appsrv.read.n and
#                             transfer.appsrv.write.n entries and their
#                             subdirectories. If this setting is FALSE,
#                             transfers can happen anywhere.
# transfer.appsrv.read.<n>: List of directory names that downloads can read
#                             from.
# transfer.appsrv.write.<n>: List of directory names that uploads can write
#                             to.
#NOTE: By default the file transfer is disabled as a security measure
transfer.database.enabled=FALSE
transfer.appsrv.enabled=FALSE
transfer.appsrv.workAreaRoot=
transfer.appsrv.accessControl=TRUE
#List transfer.appsrv.read.<n> directories
transfer.appsrv.read.1=c:¥temp
#List transfer.appsrv.write.<n> directories
transfer.appsrv.write.1=c:¥temp

```

C.9 デフォルトの weutilbase.htm

```

<HTML>
<!-- FILE: weutilbase.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. -->
<!-- and a certificate registration applet for the WebUtil utility -->
<!--
<!-- 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 uses several extra tags that are not present in the -->
<!-- default template files. You should ensure that these are -->
<!-- present in the configuration that uses this template -->
<!-- The extra substitution Tags are: -->

```

```

<!-- %webUtilArchive% = jar file containing the WebUtil code      -->
<!--                      (by default this should be frmweutil.jar) -->
<!-- %WebUtilLogging% = Defines the current logging mode.      -->
<!--                      Valid values: off|on|console|server|all  -->
<!--                      (on == console)                        -->
<!-- %WebUtilLoggingDetail% = Specifies the level of error logging.-->
<!--                      Valid values: normal|detailed          -->
<!-- %WebUtilErrorMode% = Should errors be displayed in an alert -->
<!--                      as well as the programmer defined     -->
<!--                      locations                             -->
<!--                      Valid values: console|server|alert|all -->
<!-- %WebUtilDispatchMonitorInterval% = Counts in second to   -->
<!--                      indicate how often the monitor thread -->
<!--                      checks to see if the Forms session is still-->
<!--                      alive. Used with the WebUtil_Session  -->
<!--                      package.                             -->
<!-- %WebUtilTrustInternal% = Should intranet without domain suffix-->
<!--                      be trusted.                           -->
<!--                      Valid values: true|yes|false|no       -->
<!-- %WebUtilMaxTransferSize% = Size in bytes of file transfer  -->
<!--                      segments. Default and maximum allowed is -->
<!--                      16384, i.e. 16K.                       -->
<HEAD><TITLE>%pageTitle% - WebUtil</TITLE></HEAD>
<BODY %HTMLbodyAttrs%>
%HTMLbeforeForm%
<!-- Registration applet definition (start) -->
<APPLET CODEBASE="%codebase%"
        CODE="oracle.forms.weutil.common.RegisterWebUtil"
        ARCHIVE="%webUtilArchive%"
        WIDTH="0"
        HEIGHT="0"
        HSPACE="0"
        VSPACE="0">
</APPLET>
<!-- Registration applet definition (end) -->

<!-- Forms applet definition (start) -->
<APPLET CODEBASE="%codebase%"
        CODE="oracle.forms.engine.Main"
        ARCHIVE="%archive%,%webUtilArchive%"
        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%">
<PARAM NAME="EndUserMonitoringEnabled" VALUE="%EndUserMonitoringEnabled%">
<PARAM NAME="EndUserMonitoringURL" VALUE="%EndUserMonitoringURL%">
<PARAM NAME="heartbeat" VALUE="%heartbeat%">
<PARAM NAME="heartBeat" VALUE="%heartBeat%">
<!-- Params specific to weutil -->

```

```

<PARAM NAME="WebUtilLogging" VALUE="%WebUtilLogging%">
<PARAM NAME="WebUtilLoggingDetail" VALUE="%WebUtilLoggingDetail%">
<PARAM NAME="WebUtilErrormode" VALUE="%WebUtilErrorMode%">
<PARAM NAME="WebUtilDispatchMonitorInterval" VALUE="%WebUtilDispatchMonitorInterval%">
<PARAM NAME="WebUtilTrustInternal" VALUE="%WebUtilTrustInternal%">
<PARAM NAME="WebUtilMaxTransferSize" VALUE="%WebUtilMaxTransferSize%">

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

%HTMLafterForm%

</BODY>
</HTML>

```

C.10 デフォルトの webutiljini.htm

```

<HTML>
<!-- FILE: webutiljini.htm (Oracle Forms) -->
<!-- -->
<!-- This is the a HTML template file for running a form on the -->
<!-- web using JInitiator-style tags to include the Forms applet. -->
<!-- and a certificate registration applet for the WebUtil utility -->
<!-- -->
<!-- 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 uses several extra tags that are not present in the -->
<!-- default template files. You should ensure that these are -->
<!-- present in the configuration that uses this template -->
<!-- The extra substitution Tags are: -->
<!-- %WebUtilArchive% = jar file containing the WebUtil code -->
<!-- (by default this should be frmwebutil.jar) -->
<!-- %WebUtilLogging% = Defines the current logging mode. -->
<!-- Valid values: off|on|console|server|all -->
<!-- (on == console) -->
<!-- %WebUtilLoggingDetail% = Specifies the level of error logging. -->
<!-- Valid values: normal|detailed -->
<!-- %WebUtilErrorMode% = Should errors be displayed in an alert -->
<!-- as well as the programmer defined -->
<!-- locations -->
<!-- Valid values: console|server|alert|all -->
<!-- %WebUtilDispatchMonitorInterval% = Counts in second to -->
<!-- indicate how often the monitor thread -->
<!-- checks to see if the Forms session is still -->
<!-- alive. Used with the WebUtil_Session -->
<!-- package. -->
<!-- %WebUtilTrustInternal% = Should intranet without domain suffix -->
<!-- be trusted. -->
<!-- Valid values: true|yes|false|no -->
<!-- %WebUtilMaxTransferSize% = Size in bytes of file transfer -->
<!-- segments. Default and maximum allowed is -->
<!-- 16384, i.e. 16K. -->
<HEAD><TITLE>%pageTitle% - WebUtil</TITLE></HEAD>
<BODY %HTMLbodyAttrs%>
%HTMLbeforeForm%
<!-- Registration applet definition (start) -->
<OBJECT classid="%jinit_classid%"
        codebase="/forms/jinitiator/%jinit_exename%"

```

```

        WIDTH="0"
        HEIGHT="0"
        HSPACE="0"
        VSPACE="0">
<PARAM NAME="TYPE"          VALUE="%jinit_mimetype%">
<PARAM NAME="CODEBASE"     VALUE="%codebase%">
<PARAM NAME="CODE"        VALUE="oracle.forms.webutil.common.RegisterWebUtil" >
<PARAM NAME="ARCHIVE"     VALUE="%webUtilArchive%" >
<COMMENT>
<EMBED SRC="" PLUGINSPAGE="%jinit_download_page%"
        TYPE="%jinit_mimetype%"
        java_codebase="%codebase%"
        java_code="oracle.forms.webutil.common.RegisterWebUtil"
        java_archive="%webUtilArchive%"
        WIDTH="1"
        HEIGHT="1"
        HSPACE="0"
        VSPACE="0"
>
<NOEMBED>
</COMMENT>
</NOEMBED></EMBED>
</OBJECT>
<!-- Registration applet definition (end) -->
<!-- Forms applet definition (start) -->
<OBJECT classid="%jinit_classid%"
        codebase="/forms/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%,%webUtilArchive%" >
<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%">
<PARAM NAME="EndUserMonitoringEnabled" VALUE="%EndUserMonitoringEnabled%">
<PARAM NAME="EndUserMonitoringURL" VALUE="%EndUserMonitoringURL%">
<PARAM NAME="heartbeat"   VALUE="%heartbeat%">
<PARAM NAME="WebUtilLogging" VALUE="%WebUtilLogging%">
<PARAM NAME="WebUtilLoggingDetail" VALUE="%WebUtilLoggingDetail%">
<PARAM NAME="WebUtilErrorMode" VALUE="%WebUtilErrorMode%">
<PARAM NAME="WebUtilDispatchMonitorInterval" VALUE="%WebUtilDispatchMonitorInterval%">
<PARAM NAME="WebUtilTrustInternal" VALUE="%WebUtilTrustInternal%">
<PARAM NAME="WebUtilMaxTransferSize" VALUE="%WebUtilMaxTransferSize%">
<COMMENT>
<EMBED SRC="" PLUGINSPAGE="%jinit_download_page%"
        TYPE="%jinit_mimetype%"

```

```

    java_codebase="%codebase%"
    java_code="oracle.forms.engine.Main"
    java_archive="%archive_jini%,%webUtilArchive%"
    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%"
    EndUserMonitoringEnabled="%EndUserMonitoringEnabled%"
    EndUserMonitoringURL="%EndUserMonitoringURL%"
    heartBeat="%heartBeat%"
    WebUtilLogging="%WebUtilLogging%"
    WebUtilLoggingDetail="%WebUtilLoggingDetail%"
    WebUtilErrormode="%WebUtilErrorMode%"
    WebUtilDispatchMonitorInterval="%WebUtilDispatchMonitorInterval%"
    WebUtilTrustInternal="%WebUtilTrustInternal%"
    WebUtilMaxTransferSize="%WebUtilMaxTransferSize%"
>
<NOEMBED>
</COMMENT>
</NOEMBED></EMBED>
</OBJECT>
<!-- Forms applet definition (end) -->
%HTMLafterForm%

</BODY>
</HTML>

```

C.11 デフォルトの webutiljpi.htm

```

<HTML>
<!-- FILE: webutiljpi.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. -->
<!-- and a certificate registration applet for the WebUtil utility -->
<!-- -->
<!-- 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 uses several extra tags that are not present in the -->
<!-- default template files. You should ensure that these are -->
<!-- present in the configuration that uses this template -->
<!-- The extra substitution Tags are: -->

```

```

<!-- %webUtilArchive% = jar file containing the WebUtil code      -->
<!--                    (by default this should be frmwebutil.jar) -->
<!-- %WebUtilLogging% = Defines the current logging mode.        -->
<!--                    Valid values: off|on|console|server|all   -->
<!--                    (on == console)                          -->
<!-- %WebUtilLoggingDetail% = Specifies the level of error logging.-->
<!--                    Valid values: normal|detailed            -->
<!-- %WebUtilErrorMode% = Should errors be displayed in an alert -->
<!--                    as well as the programmer defined        -->
<!--                    locations                                 -->
<!--                    Valid values: console|server|alert|all   -->
<!-- %WebUtilDispatchMonitorInterval% = Counts in second to    -->
<!--                    indicate how often the monitor thread   -->
<!--                    checks to see if the Forms session is still-->
<!--                    alive. Used with the WebUtil_Session    -->
<!--                    package.                                 -->
<!-- %WebUtilTrustInternal% = Should intranet without domain suffix-->
<!--                    be trusted.                              -->
<!--                    Valid values: true|yes|false|no         -->
<!-- %WebUtilMaxTransferSize% = Size in bytes of file transfer  -->
<!--                    segments. Default and maximum allowed is -->
<!--                    16384, i.e. 16K.                         -->
<HEAD><TITLE>%pageTitle% - WebUtil</TITLE></HEAD>
<BODY %HTMLbodyAttrs%>
%HTMLbeforeForm%
<!-- Registration applet definition (start) -->
<OBJECT classid="%jpi_classid%"
        codebase="%jpi_codebase%"
        WIDTH="0"
        HEIGHT="0"
        HSPACE="0"
        VSPACE="0">
<PARAM NAME="TYPE"          VALUE="%jpi_mimetype%">
<PARAM NAME="CODEBASE"     VALUE="%codebase%">
<PARAM NAME="CODE"         VALUE="oracle.forms.webutil.common.RegisterWebUtil" >
<PARAM NAME="ARCHIVE"     VALUE="%webUtilArchive%" >
<COMMENT>
<EMBED SRC="" PLUGINSPAGE="%jpi_download_page%"
        TYPE="%jpi_mimetype%"
        java_codebase="%codebase%"
        java_code="oracle.forms.webutil.common.RegisterWebUtil"
        java_archive="%webUtilArchive%"
        WIDTH="1"
        HEIGHT="1"
        HSPACE="0"
        VSPACE="0"
>
</EMBED>
</COMMENT>
</NOEMBED></EMBED>
</OBJECT>
<!-- Registration applet definition (end) -->
<!-- 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%,%webUtilArchive%" >

```

```

<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%">
<PARAM NAME="EndUserMonitoringEnabled" VALUE="%EndUserMonitoringEnabled%">
<PARAM NAME="EndUserMonitoringURL" VALUE="%EndUserMonitoringURL%">
<PARAM NAME="heartBeat" VALUE="%heartBeat%">
<PARAM NAME="WebUtilLogging" VALUE="%WebUtilLogging%">
<PARAM NAME="WebUtilLoggingDetail" VALUE="%WebUtilLoggingDetail%">
<PARAM NAME="WebUtilErrorMode" VALUE="%WebUtilErrorMode%">
<PARAM NAME="WebUtilDispatchMonitorInterval"
VALUE="%WebUtilDispatchMonitorInterval%">
<PARAM NAME="WebUtilTrustInternal" VALUE="%WebUtilTrustInternal%">
<PARAM NAME="WebUtilMaxTransferSize" VALUE="%WebUtilMaxTransferSize%">
<COMMENT>
<EMBED SRC="" PLUGINSPPAGE="%jpi_download_page%"
TYPE="%jpi_mimetype%"
java_codebase="%codebase%"
java_code="oracle.forms.engine.Main"
java_archive="%archive%,%webUtilArchive%"
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%"
EndUserMonitoringEnabled="%EndUserMonitoringEnabled%"
EndUserMonitoringURL="%EndUserMonitoringURL%"
heartBeat="%heartBeat%"
WebUtilLogging="%WebUtilLogging%"
WebUtilLoggingDetail="%WebUtilLoggingDetail%"
WebUtilErrormode="%WebUtilErrorMode%"
WebUtilDispatchMonitorInterval="%WebUtilDispatchMonitorInterval%"
WebUtilTrustInternal="%WebUtilTrustInternal%"
WebUtilMaxTransferSize="%WebUtilMaxTransferSize%"
>
</EMBED>
</COMMENT>
</NOEMBED></EMBED>
</OBJECT>

```

```
<!-- Forms applet definition (end) -->
%HTMLafterForm%
</BODY>
</HTML>
```

数字

3層のアーキテクチャ, 1-3

A

allow_debug、トレース・ログの表示, 4-11
archive_ie パラメータ, 4-13
archive_jinit パラメータ, 4-13
archive パラメータ, 4-13

B

background パラメータ, 4-13
base.htm, 3-4, C-8
説明, C-9
例, C-10
basejini.htm, 3-4, C-8
説明, C-9
例, C-11
basejini.htm ファイル、OracleAS Forms, 11-5
basejpi.htm, 3-4
説明, C-9
basejpi.htm ファイル
サンプル・デフォルト, C-12
basejpi.htm ファイル、OracleAS Forms, 11-5

C

CGI、Forms のアップグレード, 11-4
CodeBase, 4-26
codebase パラメータ, 4-12
codebase パラメータ、OracleAS Forms, 11-10
colorScheme パラメータ, 4-13

D

DCM プロセス
再起動, 8-2
default.env
Solaris のサンプル, C-7
Windows のサンプル・デフォルト, C-5
default.env ファイル、OracleAS Forms Services, 11-2, 11-5

E

EAR, 3-5

em_mode, 4-14
Enterprise Manager
Application Server Control コンソール, 4-2

F

FORM_PATH, 4-16
Forms, 8-2
Forms CGI
アップグレード, 11-4
説明, 11-4
Forms Listener, 1-4
Forms Listener Servlet, 1-4, 1-5
HTTPS, 5-5
クライアント要件, 5-5
サーバー要件, 5-5
Forms Runtime Diagnostics, 8-2
Forms Services
Web ランタイム・プリーング, 10-3
イベントの監視, 10-3
インスタンスの監視, 10-2
メトリック情報の検索, 10-3
メトリック情報のソート, 10-3
ユーザー・セッションの監視, 10-3
Forms Servlet, 5-2
Forms Services で使用するアイコンとイメージの配置
, 4-22
Forms Services の最適化, 10-2
Forms Services のメトリック
監視, 8-10
Forms Services リソース要件, 10-5
Forms Servlet の別名、web.xml ファイル, 11-2
Forms Trace, 3-5
forms.conf, C-16
説明, 3-6
デフォルト・サンプル, C-16
forms.conf ファイル, 11-2
formsMessageListener, 4-12
FormsServlet.initArgs, 3-13
formsweb.cfg, 3-4
例, C-2
formsweb.cfg ファイル, 11-2
Forms CGI, 11-4
Forms の統合
Web Cache, 10-12
Forms ランタイム・エンジン, 1-4
Forms ランタイム・プロセス, 1-4
Forms リソース

Oracle Internet Directory のデフォルト・プリファレンスでの定義, 6-3
FRD, 8-2
frmall_jinit.jar
説明, 3-13
frmall.jar
説明, 3-13
frmservlet、OracleAS Forms, 11-9
ftrace.cfg, 3-5

G

Graphics, 4-21

H

height パラメータ, 4-13
HTML ベースの Oracle Enterprise Manager, 4-2
HTTP Listener, 5-2
構成ファイル, 3-6
HTTPD, 5-2
HTTPS
Forms Listener Servlet, 5-5

I

ifcgi60.exe ファイル, 11-4
imageBase, 4-12
Internet Explorer と JInitiator, B-3

J

J2EE, 5-1
JAR ファイル, 10-8
JAR ファイル、キャッシュ, 10-9
Java Plug-in, 10-9
Java Plug-in、OracleAS Forms, 11-5
Java クライアント・リソース要件, 10-5
jinit_classid, 4-14
jinit_download_page, 4-14
jinit_exename, 4-14
jinit_mimetype, 4-14
JInitiator, 10-8
説明, 3-13
JInitiator、OracleAS Forms, 11-5, 11-6
JInitiator のキャッシュ・サイズ, B-5
JInitiator の説明, B-2
JInitiator のヒープ・サイズ, B-5
JInitiator プロキシ・サーバー, B-5
jpi_classid, 4-14
jpi_codebase, 4-14
jpi_download_page, 4-14
jserv.properties ファイル
OracleAS Forms, 11-2
OracleAS Forms Listener Servlet, 11-9
JVM コントローラ
JVM コントローラのログ・ファイルの削除, 7-19
JVM プーリングのエラー・メッセージ, 7-19
子 JVM, 7-16
使用コマンド, 7-11
制限, 7-11
デフォルトのプロパティの指定, 7-9
デフォルトのロギング・プロパティ, 7-17

複数, 7-16
プロパティの編集, 7-9
ロギング管理, 7-17
ロギングの有効化と無効化, 7-17
ログ・ファイルのディレクトリの指定, 7-18
ログ・ファイルへのアクセス, 7-18

JVM プーリング

EM による JVM コントローラの管理, 7-6
JVM コントローラの起動と停止, 7-10
Java コードの再インポート, 7-3
JVM コントローラについて, 7-4
JVM コントローラの管理, 7-6
JVM コントローラの削除, 7-8
以前のバージョンの Java Importer, 7-3
概要, 7-2
起動オプション, 7-13
構成ファイル設定, 7-14
コマンドラインによる JVM コントローラの管理, 7-7
新規 JVM コントローラの作成, 7-7
スレッド処理, 7-5
静的変数の共有, 7-4
設計時の考慮事項, 7-3
例, 7-2

L

LD_LIBRARY_PATH, 4-17
Listener Servlet、web.xml の OracleAS Forms エントリ, 11-7
Listener、Forms 6i、説明, 11-7
logo, 4-12
lookAndFeel パラメータ, 4-13
lservlet、OracleAS Forms, 11-9

M

mapFonts, 4-13
MENU_BUFFERING を使用不可に設定, 10-11
mod_oc4j
OracleAS Forms Services のロード・バランシング, 11-10

N

networkRetries, 4-13

O

OC4J, 5-1
構成ファイル, 3-5
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
oid_formsid, 4-15
Oracle Application Server, 1-2
Oracle Application Server Reports Services、構成と OracleAS Forms, 11-2
Oracle Application Server Single Sign-On、OracleAS Forms と静的 HTML, 11-6
Oracle Database, 1-2
Oracle Enterprise Manager、構成と OracleAS Forms, 11-2

Oracle HTTP Listener の構成ファイル, 3-6
Oracle HTTP Server の証明書
 インポート, 5-9
Oracle Identity Management Infrastructure, 2-3
Oracle Internet Directory, 2-2, 6-2
 Forms リソースを定義するデフォルト・プリファレンス, 6-3
 構成のオプション, 2-3
 動的リソースの作成, 2-3
Oracle Internet Directory、構成と OracleAS Forms, 11-2
Oracle JInitiator, 10-8, B-2
 Internet Explorer での使用方法, B-3
 Netscape Navigator での使用方法, B-3
 概要, 3-13
 キャッシュ・サイズの変更, B-5
 サポートされる構成, B-2
 システム要件, B-3
 出力の表示, B-5
 ヒープ・サイズの変更, B-5
 プラグインの設定, B-4
 利点, B-2
Oracle Real Application Clusters, 1-2
Oracle Single Sign-On Server, 6-2
ORACLE_GRAPHICS61_HOME, 4-16
ORACLE_HOME, 4-15, 4-16
OracleAS Forms Services アーキテクチャ、イメージ, 1-3
OracleAS Single Sign On
 Forms からのアクセス, 6-6
OracleAS Single Sign-On
 アプリケーションでの有効化, 6-3
 構成と OracleAS Forms, 11-2
 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, 6-7
oracle.forms.servlet.ListenerServlet、Oracle9iAS Forms, 11-9
Oracle インターネット・プラットフォーム, 1-2

P

PATH, 4-16
Performance Event Collection Services (PECS), 8-2

R

RAD エントリ, 2-2
recordFileName, 4-12
Registry.dat, 3-6, C-17
 サンプル・デフォルト, C-18
 説明, 3-6
 パラメータの値の削除, 4-22
 パラメータの値の追加, 4-22
 パラメータの値の変更, 4-22
Registry.dat、管理, 4-22
REPORTS_CLASSPATH, 4-16
REPORTS_SERVERMAP, 4-16
restrictedURLparams, 4-12
RUN_REPORT_OBJECT ビルトイン、OracleAS Forms, 11-11

S

ScriptAlias ディレクティブ、Oracle9iAS Forms, 11-4
separateFrame パラメータ, 4-13
serverApp パラメータ, 4-13
serverArgs パラメータ, 4-10
serverHost パラメータ、OracleAS Forms, 11-6
serverPort パラメータ、OracleAS Forms, 11-6
serverURL, 4-12
serverURL パラメータ
 OracleAS Forms でのアプリケーションの配布, 11-8
 OracleAS Forms での静的 HTML ファイル, 11-6
Servlet Logging Tools, 8-10
Servlet の別名、Forms、web.xml ファイル, 11-2
splashScreen パラメータ, 4-13
SSL
 Forms Services の構成, 5-7
 Oracle HTTP Server の構成, 5-7
 Web Cache の構成, 5-8
 クライアント側の証明書の有効化, 5-8
 デフォルトの Wallet, 5-8
 フォームの実行, 5-9
 ロード・バランシング・ルーターによる構成, 5-10
ssoCancelUrl, 6-6
ssoDynamicResourceCreate
 概要, 6-5
ssoErrorURL, 6-6
ssoMode
 概要, 6-5
ssoMode パラメータ
 特定のアプリケーションでの有効化の例, 6-5
Sun 社の Java Plug-in, 10-9

U

Upload/Translate ユーティリティ
 起動, 8-6
URL エスケープ・シーケンス, 3-11

W

Web Cache
 Forms の統合, 10-12
 セッション・バインドの構成, 10-12, 10-13
 設定のテスト, 10-13
Web Cache 証明書
 インポート, 5-9
WebUtil 構成ファイル, 3-6
web.xml, 3-5, C-14
 OracleAS Forms, 11-2
web.xml ファイル
 デフォルト・サンプル, C-14
Web 上の Forms アプリケーションの機能制限, 4-27
width パラメータ, 4-12

Z

zone.properties
 ファイル、OracleAS Forms Listener Servlet, 11-9

あ

アイコン

- JAR ファイルの作成, 4-25
 - 検索パス, 4-25
 - 配置, 4-22
- アップグレード
 - CGI から Forms Servlet へ, 11-4
 - Forms 6i Listener から Forms Listener Servlet へ, 11-7
 - Forms Services の検証, 11-12
 - アプリケーション・モジュール, 11-4
 - 構成ファイルの依存関係, 11-2
 - 項目, 11-2
 - 推奨事項, 11-3
 - 静的 HTML 開始ファイル, 11-5
 - タスク, 11-3
 - ロード・バランシング, 11-10
- アプリケーション
 - 環境ファイル、OracleAS Forms Services, 11-5
 - サーバー, 1-3
- アプリケーションの配布
 - 概要, 3-7
 - 手順, 3-7
- アプレット
 - パラメータ, 4-12
- 暗号化されたプログラム単位, 10-5
- イベント、トレース, 8-6
- イベントの詳細、トレース, 8-8
- イベント・バンドル, 10-6
- イメージ, 4-22
 - JAR ファイルの作成, 4-25
 - 検索パス, 4-25
 - スプラッシュ画面, 4-24
 - バックグラウンド, 4-24
- イメージ、配置、OracleAS Forms, 11-10
- インプロセス JVM、定義, 7-2
- インライン IME サポート, 4-27
- 上書き, 3-9
- エンド・ユーザー監視
 - formsweb.cfg の変更, 9-4
 - Oracle Management Agent
 - 概要, 9-2
 - Web Cache の構成, 9-3
 - 監視する Web Cache インスタンスの指定, 9-3
 - 概要, 9-2
 - 構成, 9-2
 - 要件, 9-2
 - デフォルトの最小ヒット数のしきい値の指定, 9-4
 - 不適切なレスポンス時間の排除, 9-4
 - 有効化, 9-4

か

- カスタマイズ HTML テンプレート・ファイル、
 - OracleAS Forms, 11-8, 11-10
- 仮想グラフィック・システム (VGS) ツリー, 10-6
- 環境ファイル、OracleAS Forms Services アプリケーション, 11-5
- キー・マッピング
 - fmrweb.res, 4-28
 - 有効化, 4-28
- 期間イベント, 8-6
- 起動, 4-2
- 起動時間, 10-8
- クライアント層, 1-3

- クライアント・ブラウザのサポート
 - 概要, 3-12
- クライアント・リソース要件, 10-5
- 権限
 - ユーザーのクラス, 2-2
- 権限およびアクセス強制, 2-3
- 言語検出, 4-27
 - 概要, 4-28
 - マルチレベルの継承, 4-28
- 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, 3-13
- 構成ファイル, 3-3, 3-4
 - 6iserver.conf, 11-2

さ

- サーブレットのログ・ファイル
 - サンプル出力, 8-12
 - ディレクトリ, 8-12
- サーブレットのログ・ファイルのディレクトリ, 8-12
- サンプル値, 3-9
- サンプル・ファイル
 - base.htm, C-10
 - basejinit.htm, C-11
- 指定, 3-8
- 使用, 2-3
- シングル・サインオン, 6-2
- スプラッシュ画面, 4-24
- スレッド処理
 - Forms ランタイム・プロセスと JVM, 7-5

た

- タイマー、チューニング, 10-11
- 中間層, 1-3
- チューニング
 - JAR ファイルの使用, 10-8
 - MENU_BUFFERING, 10-11
 - MENU_BUFFERING を使用不可に設定, 10-11
 - アプリケーションのサイズ, 10-12
 - 画面の表示, 10-11
 - タイマー, 10-11
 - ナビゲーションの削減, 10-10
 - ネットワーク帯域幅の削減, 10-10
 - ボイラープレート・オブジェクトの削減, 10-10
 - ボイラープレート項目, 10-10
 - メッセージ順序, 10-10
 - 類似点の活用, 10-10
- テスト・フォーム
 - 保護, 4-19
- 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, 3-12
 - 作成, 4-21
- データ・ストリーム圧縮, 10-10
- データ・セグメント, 10-5
- データベース層
 - 説明, 1-3
- デフォルト動作, 3-10
- デフォルトの formsweb.cfg ファイル
 - サンプル, C-2

デフォルトの `jvmcontroller.cfg`
 サンプル・ファイル, C-19

デフォルトの `webutilbase.htm`
 サンプル・ファイル, C-21
 説明, 3-7

デフォルトの `webutil.cfg`
 サンプル・ファイル, C-20
 説明, 3-6

デフォルトの `webutiljini.htm`
 サンプル・ファイル, C-23
 説明, 3-7

デフォルトの `webutiljpi.htm`
 サンプル・ファイル, C-25
 説明, 3-7

デフォルトの構成パラメータ

- `allow_debug`, 4-11
- `array`, 4-11
- `baseHTMLjiniator`, 4-8, 4-14
- `baseHTMLjpi`, 4-8
- `buffer`, 4-10
- `clientDPI`, 4-10
- `connectionDisallowedURL`, 4-8
- `debug`, 4-10
- `debug_messages`, 4-11
- `defaultcharset`, 4-9
- `em_trace`, 4-11
- `envFile`, 4-8
- `escapeparams`, 4-10
- `form`, 4-10
- `heartBeat`, 4-10
- `host`, 4-11
- `HTMLafterForm`, 4-12
- `HTMLbeforeForm`, 4-12
- `HTMLbodyAttrs`, 4-12
- `HTMLdelimiter`, 4-8
- `ie50`, 4-9
- `jvmcontroller`, 4-9
- `log`, 4-9, 4-11
- `otherparams`, 4-10
- `pageTitle`, 4-12
- `port`, 4-11
- `query_only`, 4-11
- `quiet`, 4-11
- `record`, 4-11
- `render`, 4-11
- `term`, 4-11
- `tracegroup`, 4-11
- `USERID`, 4-10
- `workingDirectory`, 4-8

統合

- Forms と Reports の情報, 6-7

統合されたコール、OracleAS Forms から Reports へ
 , 11-11

特殊なキー・マッピング, 4-29

トレース・イベントの詳細, 8-8

トレース可能イベント, 8-6

トレース・データ

- XML に変換, 8-6

トレースの `tracegroup` パラメータ, 8-4

トレースの URL パラメータ・オプション, 8-4

トレースのログ・パラメータ, 8-4

トレース用 `translate` ユーティリティ, 8-6

な

ネットワーク

- 帯域幅の削減, 10-10
- ネットワーク使用量, 10-6
- ネットワークの待機時間, 10-6
- ネットワーク・パケット, 10-6

は

配布

- フォームと Web, 3-1
- バックグラウンド, 4-24
- 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, 8-2
- パラメータ, 3-8, 3-9
- パラメータ・オプション

 - URL での指定, 8-4

- ビルトイン・イベント, 8-6
- ファイル

 - `basejini.htm`, 11-5
 - `basejpi.htm`, 11-5
 - `default.env`, 11-5
 - `default.env`、OracleAS Forms Services, 11-2
 - `forms.conf`, 11-2
 - `formsweb.cfg`, 11-4
 - `formsweb.cfg`、アプリケーション構成ファイル、
 Forms, 11-2
 - `ifcgi60.exe`、Oracle9iAS Forms, 11-4
 - `jserv.properties`
 OracleAS Forms, 11-2

- ベース HTML ファイル

 - 作成, C-9
 - 選択, 3-13
 - パラメータと変数, C-9
 - 変更, B-6
 - 変数の変更, C-9
 - リスト, 3-4

- 別名、Forms Servlet, 11-11
- 別名、Forms Servlet、`web.xml` ファイル, 11-2
- 保護, 2-2
- ポインテラブル・オブジェクト / イメージ, 10-5
- ポイント・イベント, 8-6

ま

メトリックのロギング

- URL での指定, 8-11
- 有効化, 8-11

や

ユーザー ID / パスワード機能

- 設定, 5-6

ら

ランタイム・プルーニング, 1-3

- 事前起動パラメータの構成, 10-4

ランフォーム・パラメータ, 3-10, 3-11, 4-10

- 定義, 3-10
- 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, 3-10
- リソース, 2-2
 - 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, 10-5
 - 画面のレンダリング, 10-6
 - データ・セグメント, 10-5
 - ネットワーク使用量, 10-6
 - パケットの送信, 10-6
 - ボイラープレート・オブジェクト, 10-5
- ロード・バランシング
 - OracleAS Forms, 11-10
- ログイン機能, 8-11