

Oracle® Application Server Reports Services

レポート Web 公開ガイド

10g リリース 2 (10.1.2)

部品番号 : B25067-02

2006 年 9 月

Oracle Application Server Reports Services レポート Web 公開ガイド, 10g リリース 2 (10.1.2)

部品番号 : B25067-02

原本名 : Oracle Application Server Reports Services Publishing Reports to the Web, 10g Release 2 (10.1.2)

原本部品番号 : B14048-02

原本著者 : Ingrid Snedecor

原本協力者 : Ellen Gravina, Panna Hegde, Vinayak Hegde, Rohit Marwaha, Ratheesh Pai, Vinodkumar Pandurangan, Pravin Prabhakar, Rajesh Ramachandran, Vishal Sharma, Navneet Singh, Puvanenthiran Subbaraj, Philipp Weckerle

Copyright © 2003, 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	xix
対象読者	xx
ドキュメントのアクセシビリティについて	xx
関連ドキュメント	xx
表記規則	xxi
サポートおよびサービス	xxi

第 I 部 環境の準備

1 OracleAS Reports Services のアーキテクチャについて

1.1	OracleAS Reports Services の概要	1-2
1.2	OracleAS Reports Services コンポーネント	1-4
1.3	OracleAS Reports Services のランタイム・プロセス	1-6
1.4	OracleAS Reports Services の通信アーキテクチャ	1-8
1.4.1	ブロードキャスト・メカニズムを使用したサーバーの検出	1-9
1.4.1.1	サブネット内のサーバーの検出	1-10
1.4.1.2	複数のサブネットにわたるサーバーの検出	1-11
1.4.2	COS ネーミング・サービスを使用したサーバーの検出	1-12
1.5	システムの設定	1-13
1.5.1	使用されるリクエスト・タイプの選択	1-13
1.5.2	Reports Servlet、JSP、Web サービスまたは CGI の選択	1-14
1.5.3	シングルマシン構成とマルチマシン構成の選択	1-14
1.5.4	高可用性環境の選択	1-15
1.5.4.1	Oracle Application Server での高可用性の維持	1-15
1.5.4.2	インフラストラクチャの依存性の維持	1-16

2 OracleAS Reports Services の起動と停止

2.1	Reports Server の起動と停止	2-2
2.1.1	Oracle Enterprise Manager 10g からの Reports Server の起動、停止および再起動	2-2
2.1.2	Oracle Process Manager and Notification Server からの Reports Server の起動、 停止および再起動	2-3
2.1.3	Reports Server を起動および停止する別の方法	2-3
2.1.3.1	インプロセス・サーバーの起動 (Windows および UNIX)	2-4
2.1.3.2	コマンドラインからの Reports Server の起動 (Windows および UNIX)	2-4
2.1.3.3	Reports Server の停止	2-4
2.2	Oracle Reportsブリッジの起動と停止	2-6

2.2.1	Oracle Process Manager and Notification Server からの Oracle Reports ブリッジの起動、 停止および再起動	2-6
2.2.2	コマンドラインからの Oracle Reports ブリッジの起動と停止	2-7
2.3	COS ネーミング・サービスの起動と停止	2-8
2.4	Oracle HTTP Server が実行されていることの確認	2-9
2.5	Reports Servlet および Reports Server が実行されていることの確認	2-9

3 OracleAS Reports Services の構成

3.1	OracleAS Reports Services 構成ファイル	3-2
3.2	Reports Server の構成	3-3
3.2.1	Reports Server の構成要素 (rwserverconf.dtd)	3-4
3.2.1.1	server	3-7
3.2.1.2	compatible	3-8
3.2.1.3	cache	3-9
3.2.1.4	engine	3-10
3.2.1.5	security	3-13
3.2.1.6	oidconnection	3-15
3.2.1.7	destination	3-16
3.2.1.8	networkConfig	3-17
3.2.1.9	job	3-18
3.2.1.10	notification	3-19
3.2.1.11	log	3-20
3.2.1.12	jobStatusRepository	3-21
3.2.1.13	trace	3-22
3.2.1.14	connection	3-24
3.2.1.15	ORBPorts	3-26
3.2.1.16	queue	3-27
3.2.1.17	persistFile	3-27
3.2.1.18	jobRecovery	3-28
3.2.1.19	identifier	3-29
3.2.1.20	pluginParam	3-30
3.2.1.21	environment	3-31
3.2.2	動的な環境切替え	3-32
3.2.2.1	例	3-32
3.2.2.2	使用上の注意	3-34
3.2.3	OracleAS Portal への接続	3-36
3.3	Reports Server 検出メカニズムの構成	3-36
3.3.1	ネットワークの構成要素 (rwnetworkconf.dtd)	3-37
3.3.1.1	discoveryService	3-38
3.3.1.2	multicast	3-38
3.3.1.3	namingService	3-39
3.3.2	ブリッジの構成要素 (bridgeconf.dtd)	3-40
3.3.2.1	bridge	3-41
3.3.2.2	networkConfig	3-42
3.3.2.3	identifier	3-42
3.3.2.4	remoteBridges	3-43
3.3.2.5	remoteBridge	3-43
3.3.2.6	trace	3-44
3.4	Reports Servlet の構成	3-45
3.4.1	キー・マップ・ファイルの位置の指定	3-47

3.4.2	キー・マップ・ファイルの再ロード	3-48
3.4.3	Web コマンド出力の非表示	3-48
3.4.4	ログイン・ダイアログ・ボックスの選択	3-49
3.4.5	Reports Servlet と JSP のトレース・オプションの設定	3-49
3.4.6	Reports パラメータのデコードに使用する rwservlet キャラクタ・エンコードの指定 ...	3-50
3.4.7	URL に指定された HTML コードのブラウザでの実行の禁止	3-50
3.4.8	rwservlet に対する同時接続のプール・サイズの指定	3-51
3.4.9	サーバー・エラー・メッセージの外観のカスタマイズ	3-51
3.4.10	インプロセス・サーバーの指定	3-52
3.4.11	インプロセス・サーバーの識別	3-52
3.4.12	Oracle Internet Directory 接続プーリング・パラメータの指定	3-52
3.4.13	動的に生成されたイメージの指定	3-53
3.4.14	データベース認証とシステム認証 Cookie の期限切れの設定	3-53
3.4.15	データベース認証とシステム認証 Cookie 用の暗号化キーの設定	3-53
3.4.16	診断およびデバッグ出力のフォーマットの追加	3-53
3.4.17	rwservlet ヘルプ・ファイルの定義	3-54
3.4.18	OracleAS Single Sign-On 使用の指定	3-54
3.4.19	ネットワーク構成ファイルの指定	3-54
3.4.20	Reports Server クラスタ名を含む OracleAS Forms Services アプリケーションの移行 ...	3-54
3.4.21	代替の Reports Servlet 構成ファイルの指定	3-55
3.5	URL エンジンの構成	3-55
3.6	プロキシ情報の入力	3-56
3.7	Oracle Process Manager and Notification Server および Oracle Enterprise Manager 10g による Reports Server の構成	3-57
3.7.1	opmn.xml	3-58
3.7.1.1	モジュールの指定	3-58
3.7.1.2	スタンドアロン Reports Server の指定	3-58
3.7.1.3	インプロセス Reports Server の指定	3-61
3.7.1.4	Oracle Reports ブリッジの指定	3-62
3.7.1.5	COS ネーミング・サービスの指定	3-63
3.8	Oracle Reports と Oracle Workflow の通信の構成	3-65
3.9	レポート配布の最適化	3-65
3.10	UNIX での DISPLAY およびプリンタ依存性の解消	3-67
3.10.1	ScreenPrinter	3-67
3.10.2	拡張イメージングのサポート	3-68

4 Oracle Reports でのフォントの管理

4.1	フォントの使用	4-2
4.1.1	Reports Builder のフォント	4-2
4.1.2	レポート出力のフォント	4-2
4.1.2.1	フォントの検索	4-2
4.1.2.1.1	フォント検索アルゴリズム	4-3
4.1.3	ユーザー・インタフェースのフォント	4-5
4.2	フォントの追加	4-6
4.2.1	Reports Builder へのフォントの追加	4-6
4.2.2	レポート出力用の新しいフォントの追加	4-7
4.2.2.1	UNIX でのフォントの追加	4-7
4.2.2.2	Windows でのフォントの追加	4-8

4.3	フォント構成ファイル	4-8
4.3.1	ファイルの検索	4-11
4.4	フォント・エイリアシング	4-12
4.4.1	エイリアシング情報の指定	4-12
4.4.2	フォント・エイリアシングのメカニズム	4-14
4.4.3	フォント・エイリアス・ファイルのセクション	4-14
4.4.4	フォント・エイリアシング・ファイルの検証	4-16
4.5	フォントに関する問題のトラブルシューティング	4-16
4.6	フォントのタイプ	4-22
4.6.1	キャラクタ・セット	4-23
4.6.2	Unicode	4-23
4.6.3	Type1 フォント	4-23
4.6.4	TrueType フォント	4-24
4.6.5	TrueType Collection	4-24
4.6.6	バーコード・フォント	4-25
4.6.7	CID フォント	4-25

5 UNIX での Oracle Reports による印刷

5.1	UNIX での印刷の概要	5-2
5.1.1	全体的な印刷のメカニズム	5-2
5.1.2	UNIX および Windows での Oracle Reports の印刷メカニズム	5-2
5.1.3	印刷のサポート	5-3
5.2	UNIX でのプリンタの設定	5-4
5.2.1	UNIX でのプリンタのインストール	5-4
5.2.2	Oracle Reports で使用するプリンタの設定の確認	5-4
5.3	印刷環境の構成	5-4
5.3.1	uiprint.txt ファイルの編集	5-5
5.3.2	環境変数	5-6
5.3.3	印刷プロパティのダイアログ・ボックス	5-7
5.3.3.1	「ページ設定」ダイアログ・ボックス	5-7
5.3.3.2	「ジョブの印刷」ダイアログ・ボックス	5-7
5.4	プリンタ関連のファイル	5-7
5.4.1	ファイルの概要	5-7
5.4.2	PPD ファイル	5-8
5.4.2.1	ローカルでの PPD ファイルのカスタマイズ	5-9
5.4.3	HPD ファイル	5-10
5.4.4	フォント・メトリック・ファイル	5-10
5.4.4.1	AFM ファイル	5-11
5.4.4.2	TFM ファイル	5-11
5.4.5	uifont.ali	5-11
5.4.6	uiprint.txt	5-12
5.4.7	プリンタ関連ファイルの編集	5-12
5.4.7.1	PPD ファイルの編集	5-12
5.4.7.1.1	デフォルトの用紙サイズの変更	5-12
5.4.7.1.2	プリンタのマージン設定の変更	5-13
5.4.7.1.3	PPD ファイルへの新規フォント・エントリの追加	5-14
5.4.7.1.4	プリンタ・トレイ設定のオーバーライド	5-15
5.4.7.2	PCL 印刷を行うための HPD ファイルの編集	5-15

5.4.7.2.1	用紙サイズの変更	5-15
5.4.7.2.2	新規フォント・エントリの追加	5-15
5.5	グローバリゼーション・サポート	5-16
5.5.1	マルチバイト・キャラクタ・セットの印刷	5-16
5.5.2	IX および PASTA の概要	5-16
5.6	デバッグ・オプション	5-17
5.6.1	DEBUG_SLFIND	5-17
5.6.2	TK_DEBUG_POSTSCRIPT	5-18
5.7	よくある質問	5-18
5.7.1	一般的な印刷エラー・メッセージ	5-19
5.7.2	PCL の印刷の問題	5-22
5.7.3	PostScript の印刷の問題	5-23
5.7.4	フォントに関連する印刷の問題	5-25
5.7.5	印刷された出力の問題	5-26

6 Oracle Reports での PDF の使用

6.1	Oracle Reports に含まれる PDF 機能	6-2
6.1.1	圧縮	6-2
6.1.1.1	設定	6-2
6.1.2	フォント関連の機能	6-2
6.1.2.1	フォント・エイリアシング	6-3
6.1.2.1.1	設定	6-4
6.1.2.1.2	トラブルシューティング	6-5
6.1.2.2	フォント・サブセット	6-5
6.1.2.2.1	設定	6-5
6.1.2.2.2	下位互換性	6-7
6.1.2.2.3	トラブルシューティング	6-8
6.1.2.3	フォントの埋込み	6-9
6.1.2.3.1	設定	6-9
6.1.2.3.2	トラブルシューティング	6-10
6.1.2.4	フォント機能のまとめ	6-11
6.1.3	実行の優先順位	6-11
6.1.4	アクセシビリティ	6-12
6.1.5	分類	6-12
6.1.6	グラフのサポート	6-13
6.2	プラットフォーム間の配布時における PDF フォントの問題への対応	6-13
6.2.1	同じプラットフォームでのレポートの設計と配布	6-13
6.2.2	異なるプラットフォームでのレポートの設計と配布	6-13
6.2.2.1	シングルバイト・フォントを使用する PDF レポートの生成	6-14
6.2.2.2	マルチバイト・フォントおよび Unicode フォントを使用する PDF レポートの生成	6-16
6.3	Unicode PDF ファイルの生成	6-18
6.3.1	フォント・サブセット	6-18
6.4	双方向 (BiDi) PDF ファイルの生成	6-19
6.4.1	フォント・サブセット	6-19
6.5	マルチバイト PDF ファイルの生成	6-20
6.5.1	フォント・エイリアシング	6-20
6.5.2	フォント・サブセット	6-20

6.6	バーコード PDF ファイルの生成	6-21
6.6.1	フォントの埋込み	6-21
6.6.2	フォント・サブセット	6-21

7 クロス・プラットフォームの移植問題の解決

7.1	クロス・プラットフォームの問題の概要	7-2
7.1.1	プラットフォームによって異なるフォント	7-2
7.1.2	フォントに関連する問題の解決	7-3
7.2	HTMLCSS、RTF または Web 出力の生成	7-3
7.2.1	レポートの設計	7-4
7.2.2	レポートの配布	7-4
7.2.2.1	トラブルシューティング情報	7-6
7.2.3	よくある質問	7-6
7.3	シングルバイト PDF 出力の生成	7-8
7.3.1	レポートの設計	7-8
7.3.2	レポートの配布	7-9
7.3.2.1	トラブルシューティング情報	7-12
7.3.3	よくある質問	7-12
7.4	マルチバイト PDF 出力の生成	7-12
7.4.1	レポートの設計	7-13
7.4.2	レポートの配布	7-13
7.4.2.1	トラブルシューティング情報	7-16
7.4.3	よくある質問	7-16
7.5	Unicode PDF 出力の生成	7-17
7.5.1	レポートの設計	7-17
7.5.2	レポートの配布	7-18
7.5.2.1	トラブルシューティング情報	7-20
7.5.3	よくある質問	7-20
7.6	PostScript 出力の生成	7-20
7.6.1	レポートの設計	7-21
7.6.2	レポートの配布	7-22
7.6.3	よくある質問	7-23

8 OracleAS Reports Services の宛先の構成

8.1	出力処理の概要	8-2
8.2	サーバーへの宛先タイプの登録	8-4
8.2.1	サーバー構成ファイルの destination セクションの設定	8-4
8.2.2	宛先として有効な値の入力	8-4
8.2.2.1	宛先タイプとクラス	8-4
8.2.2.2	宛先プロパティの名前 / 値のペア	8-5
8.2.3	宛先の例	8-6

9 JDBC PDS の構成と使用

9.1	JDBC 構成ファイル	9-2
9.1.1	プリインストールされているドライバ・エントリの確認	9-6
9.1.2	Merant DataDirect ドライバのインストールと構成	9-6
9.1.2.1	Sybase ドライバ	9-7

9.1.2.2	DB2 ドライバ	9-8
9.1.2.3	SQL Server ドライバ	9-9
9.1.2.4	Informix ドライバ	9-10
9.1.2.5	カスタム・ドライバ	9-11
9.2	JDBC 問合せの定義と実行	9-12
9.2.1	接続情報の例	9-15
9.3	OracleAS Reports Services を使用した JDBC レポートの実行	9-16
9.4	トラブルシューティング情報	9-17
9.4.1	エラー・メッセージ	9-17
9.4.2	トレース情報	9-19
9.5	ユーザー独自の JDBC ドライバの追加	9-21
9.5.1	jdbcpds.conf ファイルの構成	9-21
9.5.2	ドライバの JAR ファイルのインストール	9-21

10 OracleAS Reports Services のセキュリティ保護

10.1	OracleAS Reports Services のセキュリティについて	10-2
10.1.1	リソースの保護	10-2
10.1.1.1	アプリケーション・セキュリティ	10-2
10.1.1.2	リソース・セキュリティ	10-2
10.1.1.3	データソース・セキュリティ	10-3
10.1.2	認証とアクセス権の適用	10-3
10.1.2.1	OracleAS Single Sign-On を使用するレポート・リクエスト処理	10-4
10.1.2.1.1	Single Sign-On を使用するレポート・リクエスト・フロー	10-4
10.1.2.2	OracleAS Single Sign-On を使用しないレポート・リクエスト処理	10-7
10.1.2.2.1	OracleAS Single Sign-On を使用しないレポート・リクエスト・フロー	10-8
10.1.3	Oracle Identity Management インフラストラクチャの利用	10-10
10.1.3.1	OracleAS Single Sign-On	10-10
10.1.3.1.1	Single Sign-On コンポーネント	10-11
10.2	OracleAS Reports Services セキュリティの構成	10-12
10.2.1	OracleAS Reports Services セキュリティ・オプションの構成	10-12
10.2.1.1	OracleAS Portal	10-12
10.2.1.2	セキュリティ・インタフェース	10-13

11 OracleAS Single Sign-On の構成と管理

11.1	前提条件	11-2
11.2	OracleAS Single Sign-On のデフォルトの構成	11-2
11.3	OracleAS Single Sign-On の管理	11-3
11.3.1	OracleAS Single Sign-On の有効化と無効化	11-3
11.3.2	Reports Server のセキュリティの有効化と無効化	11-4
11.3.3	データソースのセキュリティの有効化と無効化	11-4
11.3.3.1	SSOCONN	11-5
11.3.3.1.1	Oracle データベースの例	11-6
11.3.3.1.2	Oracle OLAP Server の例	11-6
11.3.3.1.3	JDBC プラuggブル・データソースの例	11-6
11.3.3.2	Oracle Internet Directory への入力	11-6
11.3.3.2.1	Oracle Delegated Administration Services	11-7
11.3.3.2.2	ユーザー・プロンプト	11-8
11.3.3.2.3	バッチ・ロード	11-8

11.3.3.2.4	すべてのユーザーへのリソースの公開	11-9
11.3.3.3	新しいリソース・タイプの追加	11-10
11.3.4	Oracle Internet Directory への接続	11-13
11.3.4.1	Oracle Internet Directory の接続エンティティの選択	11-13
11.3.4.2	Oracle Internet Directory インスタンスの選択	11-13
11.4	Oracle Internet Directory の接続エンティティの選択	11-13
11.5	OracleAS Forms Services のセキュリティに関する考慮事項	11-14

12 OracleAS Portal でのレポートの配布

12.1	Reports ユーザーおよび名前付きグループの作成	12-2
12.1.1	デフォルトの Reports 関連グループ	12-2
12.1.1.1	RW_BASIC_USER	12-3
12.1.1.2	RW_POWER_USER	12-3
12.1.1.3	RW_DEVELOPER	12-3
12.1.1.4	RW_ADMINISTRATOR	12-3
12.1.2	ユーザーとグループの作成	12-4
12.2	Oracle Reports コンポーネントの登録	12-4
12.2.1	Reports Server の登録	12-4
12.2.2	レポートの登録	12-7
12.2.3	プリンタの登録	12-11
12.2.4	使用可能日カレンダーの作成	12-13
12.2.4.1	シンプルな使用可能日カレンダーの作成	12-14
12.2.4.2	結合された使用可能日カレンダーの作成	12-15
12.2.5	「管理」ポートレット	12-17
12.3	レポートのポートレットとしての公開	12-21
12.3.1	レポートに対するプロバイダの作成	12-21
12.3.2	ページへのレポート・ポートレットの追加	12-21
12.3.3	レポート・コンポーネントのアイテム・リンクとしてのページへの追加	12-23
12.3.4	デフォルトでインストールされていない OracleAS Portal での アイテム・リンクとしてのレポートの実行	12-24
12.4	トラブルシューティング情報	12-25
12.4.1	リソースの作成の試行時の Reports と Portal の統合エラーの解決	12-25

第 II 部 サーバーへの要求送信

13 レポート・リクエストの実行

13.1	Reports の URL 構文	13-2
13.1.1	サブレット	13-2
13.1.2	JSP	13-3
13.1.3	CGI	13-4
13.2	レポート・リクエストの方法	13-5
13.3	レポートの配布	13-6
13.3.1	ペーパー・レイアウトを使用したレポートの配布	13-6
13.3.2	ペーパー・レイアウトを使用したレポートの実行	13-7
13.3.3	Web およびペーパーへの JSP レポートの配布	13-7
13.3.3.1	新規 J2EE アプリケーションの作成	13-8
13.3.3.2	OC4J を使用した J2EE アプリケーションのデプロイ	13-9
13.3.3.2.1	既存の OC4J インスタンスを使用した J2EE アプリケーションのデプロイ	13-9

13.3.3.2.2	新規 OC4J インスタンスを使用した J2EE アプリケーションのデプロイ	13-10
13.3.4	JSP ベースの Web レポートのブラウザからの実行	13-12
13.3.5	ペーパー・レイアウトを使用した JSP レポートの実行	13-12
13.3.6	UNIX で WE8MSWIN1252 キャラクタ・セットを使用したレポートの実行	13-12
13.4	OracleAS Portal でのレポートの公開	13-13
13.5	Web ブラウザからのレポート・リクエストの指定	13-13
13.6	URL エンジンへのリクエスト送信	13-13
13.7	Web サービスを介したレポートの実行	13-14
13.8	Oracle Workflow からのレポートの実行	13-14
13.9	自動実行のためのレポートのスケジューリング	13-14
13.10	追加のパラメータ	13-15
13.11	キャッシュからのレポート出力の再使用	13-15
13.11.1	使用上の注意	13-15
13.12	キー・マップ・ファイルの使用方法	13-16
13.12.1	キー・マッピングを使用可能にする方法	13-16
13.12.2	キー・マップ・ファイルへのキー・マッピング・エントリの追加	13-16
13.12.3	JSP 以外のすべてでキーを使用する場合	13-17
13.12.4	JSP としてのレポート実行でのキーの使用方法	13-17

14 Oracle Reports Web サービスの使用方法

14.1	概要	14-2
14.2	はじめに	14-2
14.2.1	RWWebservice サブレットの起動	14-2
14.2.2	WSDL の表示	14-3
14.3	Oracle Reports Web サービスの操作	14-5
14.3.1	getAPIVersion	14-5
14.3.2	getServerInfo	14-5
14.3.3	getJobInfo	14-6
14.3.4	killJob	14-7
14.3.5	runJob	14-8
14.4	サンプル・プロキシと Java クライアントのインストールと使用方法	14-9

15 拡張配布の作成

15.1	配布の概要	15-2
15.2	配布 XML ファイルの概要	15-2
15.2.1	distribution.dtd ファイル	15-2
15.2.2	属性内での変数の使用方法	15-3
15.3	配布 XML ファイルの要素	15-3
15.3.1	destinations	15-4
15.3.2	foreach	15-4
15.3.3	mail	15-6
15.3.4	body	15-7
15.3.5	attach	15-9
15.3.6	include	15-10
15.3.7	file	15-12
15.3.8	printer	15-14
15.3.9	destype	15-15
15.3.10	property	15-17

15.4	配布 XML ファイルの例	15-17
15.4.1	foreach の例	15-17
15.4.1.1	レポート・グループを個別の添付ファイルとする単一の電子メール	15-18
15.4.1.2	グループ・インスタンスごとに個別の電子メール	15-18
15.4.1.3	単独のセクションを添付ファイルとする個別の電子メール	15-18
15.4.1.4	セクションごとに異なるファイル	15-18
15.4.1.5	レポートごとに実行される印刷	15-19
15.4.1.5.1	Windows	15-19
15.4.1.5.2	UNIX	15-19
15.4.2	mail の例	15-19
15.4.2.1	レポート全体を本文とする電子メール	15-19
15.4.2.2	レポートの1つのセクションを本文とする電子メール	15-20
15.4.2.3	レポートの2つのセクションを本文とする電子メール	15-20
15.4.2.4	外部ファイルを本文とし、レポートを添付ファイルとする電子メール	15-20
15.4.2.4.1	Windows	15-20
15.4.2.4.2	UNIX	15-20
15.4.2.5	レポート全体およびグループ化されたセクションが添付された電子メール	15-21
15.4.2.6	関連する管理者と部門に送信される電子メール	15-21
15.4.3	file の例	15-21
15.4.3.1	レポート全体のファイル	15-22
15.4.3.1.1	Windows	15-22
15.4.3.1.2	UNIX	15-22
15.4.3.2	結合されたレポート・セクションのファイル	15-22
15.4.3.3	結合されたセクションからなる各グループのファイル	15-22
15.4.3.4	レポート・グループ・インスタンスごとのファイル	15-22
15.4.4	printer の例	15-23
15.4.4.1	レポート全体の印刷	15-23
15.4.4.1.1	Windows	15-23
15.4.4.1.2	UNIX	15-23
15.4.4.2	レポートの2つのセクションの印刷	15-23
15.4.4.2.1	Windows	15-23
15.4.4.2.2	UNIX	15-23
15.4.4.3	グループ化されたレポートの印刷	15-23
15.4.4.3.1	Windows	15-23
15.4.4.3.2	UNIX	15-24
15.4.4.4	結合されたセクションをグループ・インスタンスごとに印刷	15-24
15.4.4.4.1	Windows	15-24
15.4.4.4.2	UNIX	15-24
15.4.4.5	レポートの適切なインスタンスをその関連プリンタに印刷	15-24
15.4.5	destype の例	15-25
15.4.5.1	OracleAS Portal 宛先	15-25
15.4.5.2	FTP 宛先	15-25
15.4.5.3	WebDAV 宛先	15-25
15.4.5.4	FAX 宛先	15-26
15.5	実行時の配布 XML ファイルの使用方法	15-26
15.6	配布の使用に関する制限	15-27
15.6.1	OracleAS Portal 宛先	15-27
15.6.2	XML 出力	15-27
15.6.3	デリミタ付きおよび DelimitedData 出力	15-27
15.6.4	スプレッドシート出力	15-27

15.6.5	動的フォーマット値	15-28
--------	-----------------	-------

16 XML によるレポートのカスタマイズ

16.1	カスタマイズの概要	16-2
16.2	XML カスタマイズの作成	16-3
16.2.1	必須 XML タグ	16-3
16.2.2	スタイルの変更	16-4
16.2.3	書式マスクの変更	16-4
16.2.4	書式例外の追加	16-5
16.2.5	プログラム・ユニットとハイパーリンクの追加	16-6
16.2.6	新規問合せの追加と新規ヘッダー・セクションでの結果の使用	16-7
16.2.7	URL のエンコード	16-7
16.3	XML データ・モデルの作成	16-8
16.3.1	複数のデータソースの作成	16-8
16.3.2	データソース間のリンク	16-9
16.3.3	各データソース内でのグループ階層の作成	16-9
16.3.4	クロス積 (マトリックス) グループの作成	16-10
16.3.5	あらゆるレベルでの式、サマリーおよびプレースホルダの作成	16-11
16.3.6	パラメータの作成	16-12
16.4	実行時の XML ファイルの使用方法	16-13
16.4.1	実行時の XML レポート定義の適用	16-13
16.4.1.1	1 つの XML レポート定義の適用	16-13
16.4.1.2	複数の XML レポート定義の適用	16-14
16.4.1.3	PL/SQL での XML レポート定義の適用	16-14
16.4.1.3.1	ファイルに保存された XML 定義の適用	16-14
16.4.1.3.2	メモリーに保存された XML 定義の適用	16-15
16.4.2	XML レポート定義の単独での実行	16-17
16.4.3	バッチ変更の実行	16-17
16.5	XML レポート定義のデバッグ	16-17
16.5.1	XML パーサーのエラー・メッセージ	16-18
16.5.2	トレース・オプション	16-18
16.5.3	rwbuilder	16-18
16.5.4	デバッグを目的とした、ファイルへの XML の書込み	16-19

17 イベント・ドリブンによる公開の使用方法

17.1	Event-Driven Publishing API	17-2
17.1.1	API の要素	17-2
17.1.2	パラメータ・リストの作成と操作	17-2
17.1.2.1	Add_Parameter	17-2
17.1.2.2	Remove_Parameter	17-3
17.1.2.3	Clear_Parameter_List	17-3
17.1.3	パラメータの名前と値での非 ASCII 文字の使用	17-3
17.1.4	ジョブの送信	17-4
17.1.5	ステータスの確認	17-4
17.1.6	サーバーのステータス・レコードの使用方法	17-5
17.2	Event-Driven Publishing API を使用するアプリケーションのデバッグ	17-6
17.3	データベース・イベントからのレポートの起動	17-6

17.4	Oracle Advanced Queuing との統合	17-7
17.4.1	SRW_PARAMLIST タイプのメッセージを保持するキューの作成	17-8
17.4.2	エンキュー・プロシージャの作成	17-8
17.4.3	デキュー・プロシージャの作成	17-9

第 III 部 グローバリゼーション・サポートおよび双方向サポート

18 グローバリゼーションと双方向のサポートの実装

18.1	グローバリゼーション・サポート・アーキテクチャ	18-2
18.1.1	言語に依存しない機能	18-2
18.1.2	言語依存データ	18-2
18.2	グローバリゼーション・サポートの環境変数	18-2
18.2.1	NLS_LANG 環境変数	18-3
18.2.1.1	NLS_LANG 環境変数の定義	18-4
18.2.1.1.1	Windows	18-4
18.2.1.1.2	UNIX	18-4
18.2.1.2	キャラクタ・セット	18-4
18.2.1.2.1	キャラクタ・セットの設計上の考慮事項	18-4
18.2.1.2.2	フォント・エイリアシングの考慮事項	18-5
18.2.1.3	言語と地域	18-5
18.2.2	DEVELOPER_NLS_LANG 環境変数と USER_NLS_LANG 環境変数	18-6
18.3	JSP または XML ファイルでのキャラクタ・セットの指定	18-6
18.4	双方向サポート	18-8
18.5	Unicode	18-9
18.5.1	Unicode のサポート	18-9
18.5.2	Unicode フォントのサポート	18-10
18.5.3	Unicode サポートの有効化	18-11
18.6	アプリケーションの翻訳	18-11
18.7	グローバリゼーションに関する問題のトラブルシューティング	18-12

第 IV 部 パフォーマンス

19 OracleAS Reports Services の管理と監視

19.1	Oracle Enterprise Manager 10g 用の Reports Server の構成	19-2
19.2	Reports Server ホーム・ページへのナビゲート	19-2
19.2.1	Application Server Control での Reports Server ホーム・ページへのナビゲート	19-3
19.2.2	Grid Control での Reports Server ホーム・ページへのナビゲート	19-4
19.3	Reports Server の管理と監視	19-6

20 Oracle Reports のチューニング

20.1	パフォーマンス分析ツール	20-2
20.1.1	Oracle Enterprise Manager	20-2
20.1.2	レポート・トレース	20-3
20.1.3	RW_SERVER_JOB_QUEUE 表	20-5
20.1.3.1	キュー・アクティビティによるデータベースの更新	20-8
20.1.4	SHOWJOBS コマンドライン・キーワード	20-8
20.1.5	効率的な SQL	20-8

20.1.6	PL/SQL	20-10
20.1.7	Java ストアド・プロシージャ	20-11
20.1.8	Java Importer	20-11
20.2	Reports Server の構成のチューニング	20-11
20.3	rwdiag を使用したブリッジおよびネットワークのタイムアウト設定	20-13
20.4	データへのアクセス	20-14
20.4.1	非 SQL データソース	20-15
20.4.2	データベースの索引	20-16
20.4.3	計算	20-16
20.4.4	冗長なデータ	20-17
20.4.5	ブレイク・グループ	20-17
20.4.6	グループ・フィルタ	20-18
20.4.7	リンクの可否	20-18
20.5	データのフォーマット	20-18
20.5.1	ペーパー・レイアウト	20-19
20.5.1.1	フォーマット・トリガー	20-20
20.5.1.2	イメージ出力	20-21
20.5.2	Web レイアウトと JSP レポート定義	20-22
20.6	レイアウトに関する一般的なガイドライン	20-22
20.6.1	事前フェッチ	20-22
20.6.2	一括処理と配布	20-22
20.7	Forms からの Oracle Reports のコール	20-23
20.8	レポートの実行	20-23

第 V 部 付録

A コマンドライン・キーワード

A.1	コマンドラインの使用方法	A-2
A.1.1	全般的な使用上の注意	A-2
A.1.2	ルール	A-2
A.2	Oracle Reports 実行可能ファイルの概要	A-3
A.2.1	キーワードの使用方法の要約	A-3
A.2.2	rwclient	A-7
A.2.3	rwrund	A-8
A.2.4	rwbuilder	A-10
A.2.5	rwconverter	A-10
A.2.6	rwservlet	A-11
A.2.7	rwcgi	A-14
A.2.8	rwservlet	A-17
A.2.9	rwbridge	A-17
A.3	コマンドライン・キーワード	A-18
A.3.1	ACCESSIBLE	A-18
A.3.2	ARRAYSIZE	A-18
A.3.3	AUTHID	A-19
A.3.4	AUTOCOMMIT	A-20
A.3.5	BACKGROUND	A-20
A.3.6	BATCH	A-21
A.3.7	BCC	A-21

A.3.8	BLANKPAGES	A-22
A.3.9	BUFFERS	A-22
A.3.10	CACHELOB	A-23
A.3.11	CC	A-23
A.3.12	CELLWRAPPER	A-24
A.3.13	CMDFILE	A-25
A.3.14	CMDKEY	A-26
A.3.15	COLLATE	A-26
A.3.16	CONTAINSHTMLTAGS	A-27
A.3.17	CONTAINSOLE	A-28
A.3.18	CONTENTAREA	A-28
A.3.19	COPIES	A-29
A.3.20	CUSTOMIZE	A-29
A.3.21	DATEFORMATMASK	A-30
A.3.22	DELAUTH	A-31
A.3.23	DELIMITED_HDR	A-31
A.3.24	DELIMITER	A-32
A.3.25	DESFORMAT	A-33
A.3.26	DESNAME	A-35
A.3.27	DEST	A-37
A.3.28	DESTINATION	A-37
A.3.29	DESTYPE	A-38
A.3.30	DISTRIBUTE	A-44
A.3.31	DTYPE	A-44
A.3.32	DUNIT	A-45
A.3.33	ENGINERESPONSETIMEOUT	A-46
A.3.34	ENVID	A-46
A.3.35	EXPIRATION	A-46
A.3.36	EXPIREDAYS	A-47
A.3.37	EXPRESS_SERVER	A-48
A.3.38	FORMSIZE	A-49
A.3.39	FROM	A-50
A.3.40	GETJOBID	A-50
A.3.41	GETSERVERINFO	A-51
A.3.42	HELP	A-51
A.3.43	ITEMTITLE	A-52
A.3.44	JOBNAME	A-52
A.3.45	JOBTYPE	A-53
A.3.46	JVMOPTIONS	A-53
A.3.47	KILLENGINE	A-54
A.3.48	KILLJOBID	A-55
A.3.49	LONGCHUNK	A-55
A.3.50	MIMETYPE	A-56
A.3.51	MODE	A-56
A.3.52	MODULE REPORT	A-57
A.3.53	NAME	A-57
A.3.54	NONBLOCKSQL	A-58
A.3.55	NOTIFYFAILURE	A-58
A.3.56	NOTIFYSUCCESS	A-59
A.3.57	NUMBERFORMATMASK	A-59
A.3.58	OLAP_CON	A-60

A.3.59	ONFAILURE	A-60
A.3.60	ONSUCCESS	A-61
A.3.61	ORIENTATION	A-61
A.3.62	OUTPUTFOLDER	A-62
A.3.63	OUTPUTIMAGEFORMAT	A-63
A.3.64	OUTPUTPAGE	A-64
A.3.65	OVERWRITE	A-65
A.3.66	P_AVAILABILITY	A-65
A.3.67	P_DESCRIPTION	A-65
A.3.68	P_FORMATS	A-66
A.3.69	P_JDBCPDS	A-66
A.3.70	P_NAME	A-67
A.3.71	P_OWNER	A-67
A.3.72	P_PFORMTEMPLATE	A-68
A.3.73	P_PRINTERS	A-68
A.3.74	P_PRIVILEGE	A-69
A.3.75	P_SERVERS	A-69
A.3.76	P_TRIGGER	A-69
A.3.77	P_TYPES	A-70
A.3.78	PAGEGROUP	A-70
A.3.79	PAGESIZE	A-71
A.3.80	PAGESTREAM	A-72
A.3.81	PARAMFORM	A-72
A.3.82	PARSEQUERY	A-73
A.3.83	PDFCOMP	A-73
A.3.84	PDFEMBED	A-74
A.3.85	PFACTION	A-74
A.3.86	PRINTJOB	A-75
A.3.87	READONLY	A-75
A.3.88	RECURSIVE_LOAD	A-76
A.3.89	REPLYTO	A-77
A.3.90	REPORT MODULE	A-77
A.3.91	ROLE	A-77
A.3.92	RUNDEBUG	A-78
A.3.93	SAVE_RDF	A-78
A.3.94	SCHEDULE	A-79
A.3.95	SERVER	A-80
A.3.96	SHOWAUTH	A-80
A.3.97	SHOWENV	A-81
A.3.98	SHOWJOBID	A-81
A.3.99	SHOWJOBS	A-82
A.3.100	SHOWMAP	A-82
A.3.101	SHOWMYJOBS	A-83
A.3.102	SHUTDOWN	A-83
A.3.103	SITENAME	A-84
A.3.104	SOURCE	A-84
A.3.105	SQLTRACE	A-85
A.3.106	SSOCONN	A-86
A.3.107	STATUSFOLDER	A-87
A.3.108	STATUSFORMAT	A-88
A.3.109	STATUSPAGE	A-88

A.3.110	STYPE	A-89
A.3.111	SUBJECT	A-89
A.3.112	SUPPRESSLAYOUT	A-90
A.3.113	TOLERANCE	A-90
A.3.114	TRACEFILE	A-91
A.3.115	TRACEMODE	A-92
A.3.116	TRACEOPTS	A-92
A.3.117	UPGRADE_PLSQL	A-94
A.3.118	URLPARAMETER	A-94
A.3.119	USEJVM	A-94
A.3.120	USERID	A-95
A.3.121	USERSTYLES	A-96
A.3.122	VALIDATETAG	A-96
A.3.123	WEBSERVER_DEBUG	A-97
A.3.124	WEBSERVER_DOCROOT	A-98
A.3.125	WEBSERVER_PORT	A-98

B 環境変数

B.1	環境変数	B-2
B.1.1	CA_GPREFS	B-5
B.1.2	CA_UPREFS	B-5
B.1.3	DELIMITED_LINE_END	B-5
B.1.4	DOC	B-6
B.1.5	DEVELOPER_NLS_LANG	B-6
B.1.6	NLS_CALENDAR	B-6
B.1.7	NLS_CREDIT	B-6
B.1.8	NLS_CURRENCY	B-6
B.1.9	NLS_DATE_FORMAT	B-6
B.1.10	NLS_DATE_LANGUAGE	B-6
B.1.11	NLS_DEBIT	B-6
B.1.12	NLS_ISO_CURRENCY	B-6
B.1.13	NLS_LANG	B-7
B.1.14	NLS_LIST_SEPARATOR	B-8
B.1.15	NLS_MONETARY_CHARACTERS	B-8
B.1.16	NLS_NUMERIC_CHARACTERS	B-8
B.1.17	NLS_SORT	B-8
B.1.18	ORACLE_AFM	B-8
B.1.19	ORACLE_HOME	B-8
B.1.20	ORACLE_HPD	B-9
B.1.21	ORACLE_PATH	B-9
B.1.22	ORACLE_PPD	B-9
B.1.23	ORACLE_TFM	B-10
B.1.24	ORAINFONAV_DOCPATH	B-10
B.1.25	PRINTER	B-10
B.1.26	REMOTE	B-11
B.1.27	REPORTS_ADD_HWMARGIN	B-11
B.1.28	REPORTS_ARABIC_NUMERAL	B-12
B.1.29	REPORTS_BIDI_ALGORITHM	B-12
B.1.30	REPORTS_CGIDIAGBODYTAGS	B-12
B.1.31	REPORTS_CGIDIAGHEADTAGS	B-13

B.1.32	REPORTS_CGIHELP	B-13
B.1.33	REPORTS_CGIMAP	B-14
B.1.34	REPORTS_CGINODIAG	B-14
B.1.35	REPORTS_CLASSPATH	B-15
B.1.36	REPORTS_CONTAINSHTMLTAGS	B-15
B.1.37	REPORTS_COOKIE_EXPIRE	B-16
B.1.38	REPORTS_DB_AUTH	B-17
B.1.39	REPORTS_DEFAULT_DISPLAY	B-17
B.1.40	REPORTS_DEFAULT_PIXEL_SIZE	B-18
B.1.41	REPORTS_ENABLE_RTF_SPACING	B-18
B.1.42	REPORTS_ENCRYPTION_KEY	B-19
B.1.43	REPORTS_ENHANCED_SUBSET	B-19
B.1.44	REPORTS_GRAPH_IMAGE_DPI	B-19
B.1.45	REPORTS_IGNORE_IMAGE_TAG_RES	B-20
B.1.46	REPORTS_JPEG_QUALITY_FACTOR	B-20
B.1.47	REPORTS_JVM_OPTIONS	B-20
B.1.48	REPORTS_NETWORK_CONFIG	B-21
B.1.49	REPORTS-NLS_XML_CHARSETS	B-21
B.1.50	REPORTS_NO_DUMMY_PRINTER	B-22
B.1.51	REPORTS_NO_HTML_SPACE_REPLACE	B-22
B.1.52	REPORTS_OUTPUTIMAGEFORMAT	B-23
B.1.53	REPORTS_PATH	B-23
B.1.54	REPORTS_RESOURCE	B-24
B.1.55	REPORTS_SERVER	B-24
B.1.56	REPORTS_SOLARIS_9	B-25
B.1.57	REPORTS_SPACE_BREAK	B-25
B.1.58	REPORTS_SRWRUN_TO_SERVER	B-25
B.1.59	REPORTS_SSLPORT	B-26
B.1.60	REPORTS_SYS_AUTH	B-26
B.1.61	REPORTS_TAGLIB_URI	B-26
B.1.62	REPORTS_TMP	B-27
B.1.63	REPORTS_USEREXITS	B-27
B.1.64	REPORTS_UTF8_XMLOUTPUT	B-28
B.1.65	RW	B-28
B.1.66	TK_PRINT	B-28
B.1.67	TK_PRINT_STATUS	B-29
B.1.68	TK_PRINTER	B-29
B.1.69	TK_AFM	B-30
B.1.70	TK_HPD	B-30
B.1.71	TK_PPD	B-31
B.1.72	TK_TFM	B-31
B.1.73	USERNAME	B-31
B.1.74	USER-NLS_LANG	B-32

C OracleAS Portal へのレポートのバッチ登録

C.1	レポート定義ファイルのバッチ登録	C-2
C.1.1	rwconverter の実行による SQL スクリプトの生成	C-2
C.1.2	SQL*Plus でのスクリプトの実行	C-3
C.2	レポート・パッケージのバッチ削除	C-4
C.3	PL/SQL バッチ登録ファンクション	C-5

D OracleAS Reports Services のトラブルシューティング

D.1	問題と解決策	D-2
D.1.1	レポート・リクエストのハング	D-2
D.1.2	Reports Server のアクティビティによりエラー REP-50125 が生成される	D-10
D.1.3	Reports Servlet を使用して長期レポートを実行すると失敗する	D-11
D.1.4	プラットフォームによってフォントの表示が異なる	D-11
D.1.5	UNIX プラットフォームでレポートを実行すると REP-56048 が生成される	D-12
D.1.6	過負荷状態で OPMN を使用するとインプロセス・サーバーに障害が発生する	D-15
D.1.7	右から左に表示する言語のフォントの問題	D-16
D.1.8	RUN_REPORT_OBJECT を使用して Oracle Forms からレポートを 実行する際のエラー	D-17
D.1.9	Microsoft Excel でのレポート出力の表示	D-18
D.1.10	ユーザー・イグジットが含まれるレポートの UNIX での障害	D-19
D.1.11	インプロセス・サーバーを使用する際の印刷エラーとフォント・エラー	D-19
D.2	パフォーマンスの問題の診断	D-20
D.3	フォントの問題の診断	D-20
D.4	印刷の問題の診断	D-21
D.5	JDBC PDS の問題の診断	D-21
D.6	OracleAS Portal の問題の診断	D-21
D.7	グローバリゼーションの問題の診断	D-21
D.8	Oracle Reports ブリッジの問題の診断	D-21
D.9	その他の解決策	D-24

E Reports Server とブリッジの診断ユーティリティ

E.1	rwdiag の概要	E-2
E.1.1	例	E-2
E.1.1.1	例 1	E-2
E.1.1.2	例 2	E-2
E.1.1.3	例 3	E-2
E.1.1.4	例 4	E-3
E.1.1.5	例 5	E-3
E.1.1.6	例 6	E-3
E.2	コマンドライン構文	E-3
E.2.1	構文	E-3
E.2.2	使用上の注意	E-4

用語集

索引

はじめに

このマニュアルでは、レポートを公開するための Oracle Application Server Reports Services ソフトウェアの構成方法について説明します。また、OracleAS Reports Services を使用してレポートを公開するために用意されている各種オプションについても説明します。

注意： このマニュアルの Portable Document Format (PDF) 版では、URL が 2 行に分割されているとき、その URL をクリックしても、URL の完全なデータはブラウザに送られません。PDF に記載されている URL の正確なリンク先にジャンプするには、URL をコピーして、ブラウザのアドレス・フィールドに貼り付けます。このマニュアルの HTML 版では、リンクをクリックすると、そのリンク先がブラウザに直接表示されます。

対象読者

このマニュアルは、OracleAS Reports Services を使用して、レポートを公開する方を対象に記述されています。OracleAS Reports Services を構成する場合は、次のテクノロジーについて確実に理解しておく役に立ちます。

- 使用しているオペレーティング・システム
- Java
- データベース
- CORBA
- JSP ファイル
- XML ファイルおよび DTD ファイル
- Web サーバーの構成
- HTTP

このマニュアルでは、これらのテクノロジーに関するコンポーネントを構成する手順を説明しません。

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

Oracle Reports の詳細は、次のリソースを参照してください。

- 『Oracle Reports チュートリアル』
- 『Oracle Reports レポート作成ガイド』
- Oracle Reports のオンライン・ヘルプには、次のいずれかの方法でアクセスできます。
 - Reports Builder の場合：
 - 「ヘルプ」からヘルプの目次を選択します。
 - ダイアログ・ボックスで、「ヘルプ」をクリックするか [F1] を押します。
 - プロパティ・インスペクタでプロパティをクリックし、[F1] を押してプロパティのヘルプ・トピックを表示します。

- OTN (Oracle Technology Network) の Oracle Reports 10g ページ (<http://www.oracle.com/technology/products/reports/index.html>) の場合:
 - 「Resources」で「Hosted Online Help」をクリックすると、Oracle Reports のオンライン・ヘルプの最新版が Web ベースで表示されます。
 - 「News」で「Oracle Reports Online Help Update」をクリックすると、Reports Builder 内の Oracle Reports のオンライン・ヘルプが、最新版に置き換えられます。ヘルプ・ファイルの置換方法については、ダウンロード・ファイルの `readme.txt` を参照してください。

表記規則

本文では、次の表記規則を使用します。

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。
固定幅フォントのイタリック	固定幅フォントのイタリックは、変数またはユーザー指定の名前を示します。
[]	大カッコは、カッコ内の句を任意に選択できることを表します。ただし選択できるのは1つだけです。

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。

第 I 部

環境の準備

第 I 部では、OracleAS Reports Services 環境の概要を説明し、レポート実行に向けその環境を準備する際に役立つ情報を提供します。この情報には、OracleAS Reports Services の起動と停止、Reports 関連の OracleAS Reports Services コンポーネントの構成、Oracle Reports のフォント関連情報、OracleAS Reports Services の宛先の構成、JDBC PDS の構成と使用、OracleAS Reports Services のセキュリティの設定、OracleAS Single Sign-On、OracleAS Portal でのレポートの配布などが含まれます。

第 I 部には、次の章があります。

- 第 1 章「OracleAS Reports Services のアーキテクチャについて」
- 第 2 章「OracleAS Reports Services の起動と停止」
- 第 3 章「OracleAS Reports Services の構成」
- 第 4 章「Oracle Reports でのフォントの管理」
- 第 5 章「UNIX での Oracle Reports による印刷」
- 第 6 章「Oracle Reports での PDF の使用」
- 第 7 章「クロス・プラットフォームの移植問題の解決」
- 第 8 章「OracleAS Reports Services の宛先の構成」
- 第 9 章「JDBC PDS の構成と使用」
- 第 10 章「OracleAS Reports Services のセキュリティ保護」
- 第 11 章「OracleAS Single Sign-On の構成と管理」
- 第 12 章「OracleAS Portal でのレポートの配布」

OracleAS Reports Services の アーキテクチャについて

この章では、関連する Oracle Reports コンポーネントのアーキテクチャを、レポート公開コンポーネント OracleAS Reports Services とともに説明します。また、レポート実行時処理の概要についても説明し、Reports Server 環境設定時の考慮事項も示します。

この章には、次の項があります。

- [OracleAS Reports Services の概要](#)
- [OracleAS Reports Services コンポーネント](#)
- [OracleAS Reports Services のランタイム・プロセス](#)
- [OracleAS Reports Services の通信アーキテクチャ](#)
- [システムの設定](#)

1.1 OracleAS Reports Services の概要

OracleAS Reports Services は、Oracle Application Server のレポート公開コンポーネントです。これは、質の高い運用レポートを作成する企業レポート・サービスで、動的にあらゆるデータを取得し、あらゆるフォーマットに成形し、そしてあらゆる宛先へ配布します。OracleAS Reports Services を使用すると、Web ベースの環境と非 Web ベースの環境の両方で公開が可能です。

拡張性に優れ柔軟性の高いアーキテクチャを備える OracleAS Reports Services は、レポート生成エンジンを同一のサーバーおよび複数のサーバーに分散し、自動的に管理します。また、同様のリクエストが発行された際に再利用できるよう、レポート出力をキャッシュします。このサービスは、JSP、Java サーブレット、Web サービスおよび CGI（下位互換性を確保するために維持）を使用する標準の Web 環境と統合されます。また、ローカルとリモートの両アプリケーション・サーバーでのレポート実行、およびレポート実行用に複数層アーキテクチャの実装が可能です。

OracleAS Reports Services を、JSP、Java サーブレット、Web サービスまたは CGI（下位互換性を確保するために維持）と組み合わせると、Web ブラウザで標準的な URL 構文を使用して、任意のプラットフォーム上でレポートを実行できます。サーブレットを実装する場合は、応答が速く、管理が簡単なインプロセス・サーバーを使用できます。インプロセス・サーバーを使用すると、プロセス間の通信コストが削減され、結果的に応答時間が短縮されます。

OracleAS Reports Services は、すべてのジョブ・リクエストを 1 つのジョブ・キューに入れてレポートを実行するようクライアント・リクエストを操作します。いずれかのサーバーのエンジンが使用可能になると、キューで待機している次のジョブが実行されます。キューのジョブ数が増えると、サーバーはエンジンをサーバー構成で指定した最大数まで増やすことができます。同様に、エンジンは、指定したアイドル時間（第 3 章「OracleAS Reports Services の構成」を参照）を超えた場合、自動的に停止します。

OracleAS Reports Services は、実行中のジョブ、実行予定のジョブ、完了したジョブ、失敗したジョブなど、サーバーに送信されたすべてのジョブを追跡します。showjobs コマンド（rwservlet を介した Web）、Web サービスのインタフェース、Oracle Enterprise Manager 10g の OracleAS Reports Services ページ、Reports Queue Manager (Windows) および Reports Queue Viewer (UNIX) を使用すると、ジョブのスケジューリング、キューイング、開始、終了およびエラーの時刻情報のほか、ジョブ出力やレポートの最終ステータスに関する情報を表示できます。

OracleAS Reports Services のジョブ情報は永続的です。したがって、Reports Server を停止した後には再起動すると、スケジュールされたジョブだけでなく、すべてのジョブが回復します¹。

Web 環境で使用する場合、OracleAS Reports Services のアーキテクチャは次の 4 つの階層で構成されます。

注意： 階層という用語は、OracleAS Reports Services のアーキテクチャに準ずる構成要素の論理的な位置を意味します。しかしながら、それぞれの階層は、同一マシン上にも、異なるマシン上にも存在することがあります。

- クライアント層 (Web ブラウザ)
- Web サーバー層
- OracleAS Reports Services 層
- データ層 (データベースおよびその他すべてのデータソースを含む)

¹ この場合、同期ジョブおよび現在実行中のジョブのみが消失します。

非 Web 環境で使用する場合は、次の 3 つの階層で構成されます (Web サーバーが不要になりません)。

- クライアント層 (クライアント・マシンごとに、小さな独自アプリケーション)
- OracleAS Reports Services 層
- データ層 (データベースおよびプラグابل・データソースを含む)

これらの層の構成方法は、すべての階層を同一マシン上に設定する方法や各階層をすべて別のマシン上に設定する方法などがあります。また、複数のマシンに複数の Web サーバーをインストールしたり、複数のマシンに複数のアプリケーション・サーバーをインストールすることもできます。サンプル・トポロジの詳細は、Oracle Application Server のインストール・ガイドを参照してください。

Web サーバーを複数のマシンにインストールすると、複数の Oracle Application Server インスタンスをクラスタ化してロード・バランシングを行い、可用性の高いフェイルセーフ環境を実現できます。ロード・バランシングの詳細は、Oracle Application Server のインストール・ガイドを参照してください。エンタープライズ・デプロイメント・アーキテクチャと高可用性の詳細は、『Oracle Application Server エンタープライズ・デプロイメント・ガイド』および『Oracle Application Server 高可用性ガイド』を参照してください。

OracleAS Reports Services は、イベント・ベースのレポートを提供します。この方法では、データベース・イベントを使用して、レポート生成を実行します。たとえば、収益レベルが特定の水際標を上回った (下回った) という変更を通知するイベントを定義できます。データベースでこの変更 (イベント) が発生すると、レポートが自動的に生成されます。詳細は、[第 17 章「イベント・ドリブンによる公開の使用方法」](#)を参照してください。

OracleAS Reports Services には、XML を使用して、レポート配布用の独自の構成を定義する配布モジュールが付属しています。必要な XML ファイルをランタイム・コマンドラインまたは URL からコールしてレポートを 1 つ生成すれば、多様な出力と宛先の処理がサーバーにより実行されます。処理時間が大幅に短縮され、構成変更はすべて XML ファイル内で処理できます。詳細は、[第 15 章「拡張配布の作成」](#)を参照してください。

1.2 OracleAS Reports Services コンポーネント

図 1-1 OracleAS Reports Services のコンポーネント

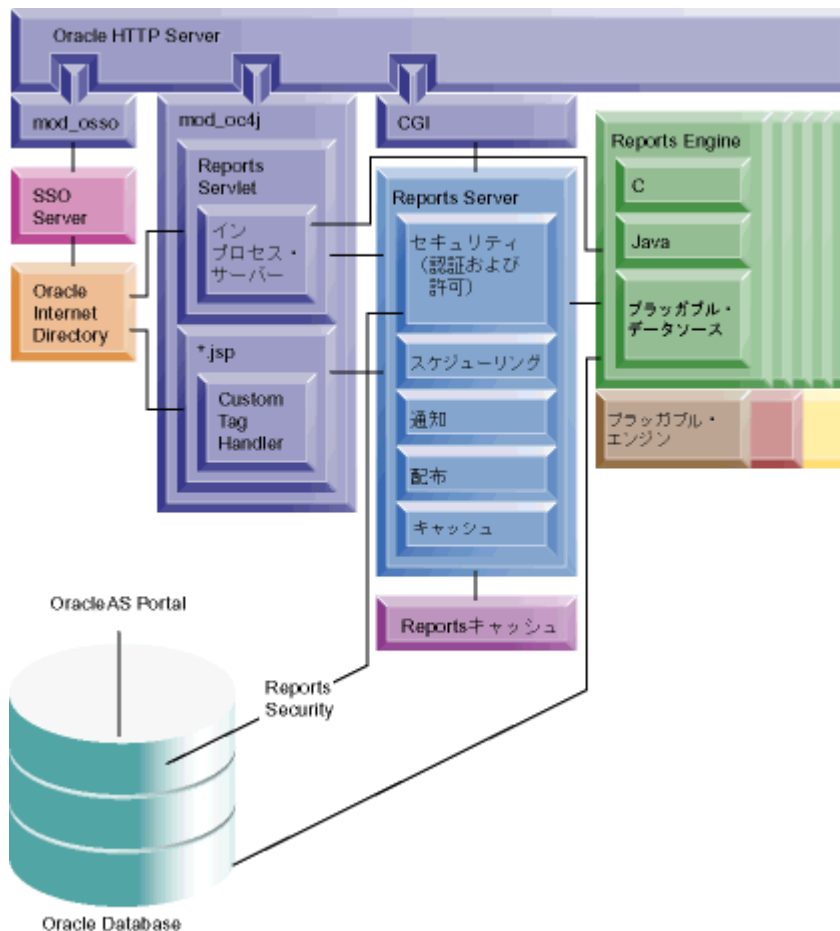


図 1-1 は、稼動中の OracleAS Reports Services 環境のコンポーネントを示しています。これには次のものが含まれます。

1. **Oracle HTTP Server** は、Oracle Application Server から提供される Web サーバーです。このサーバーは、OpenSSL モジュールを組み込んでいるので、Secure Sockets Layer (SSL) および HTTP Secure Sockets Layer (HTTPS) をサポートします。また、Java サーブレット・アプリケーションの実行をサポートするサーブレット・エンジンも備えています。
2. モジュール **mod_oc4j** は、Oracle HTTP Server が、サーブレットや JSP からのリクエストを Oracle Application Server Containers for J2EE (OC4J) にリダイレクトするために使用されます。OC4J は、JSP トランスレータ、JSP サーブレット・エンジン (OJSP) および Enterprise JavaBeans (EJB) コンテナを備える完全な J2EE 環境を提供します。OC4J は、高速、軽量、高拡張性、簡易性を兼ね備えた完全な J2EE 環境を提供します。すべて Java で記述されている OC4J は、標準の Java Development Kit (JDK) 仮想マシン (JVM) 上で動作します。

3. **Reports Servlet** (rwservlet) は、Web サーバーのサーブレット・エンジン内で実行される OracleAS Reports Services のコンポーネントです。Reports Servlet は、HTTP Server と Reports Server との間で情報を変換および配布します。Reports Servlet は、次のコンポーネントで構成されています。
 - **インプロセス・サーバー**は、クライアントからの最初のリクエストを Reports Servlet (rwservlet) または Reports JSP 経由で受信するたびに、サーバーを自動的に起動する手段を提供して、Reports Server のメンテナンスと管理を軽減します。
4. **Custom Tag Handler** は、JSP ファイルに含まれるカスタム Oracle Reports タグを処理します。JSP ファイルでは、Oracle Reports 関連のカスタム・タグは接頭辞が **rw:** です。その他の接頭辞を持つその他のカスタム・タグが存在することもあります。
5. **Reports CGI** (rwcgi) は、Web サーバーと Reports Server との間で情報を変換および配布する Web サーバーのコンポーネントです。これにより、レポートを Web ブラウザから動的に実行できます。

注意： Oracle Reports 10g では、Reports CGI (rwcgi) を使用することはお薦めしません（下位互換性を確保するためにのみ維持されています）。かわりに、Reports JSP、rwservlet (Reports Servlet) または Reports Web サービスを使用してください。

パフォーマンス上の理由により、rwcgi よりも rwservlet を使用することを強くお薦めします。rwcgi では、各リクエストに対して新規プロセスが起動されます。その際、JVM が初期化されるため、多数のレポート・リクエストを実行するとパフォーマンスが低下します。一方、rwservlet は OC4J インスタンスにデプロイされており、サーブレットの機能を利用するため、rwcgi よりも高いパフォーマンスを提供します。

6. **Reports Server** (rwservlet) はクライアント・リクエストを処理します。この処理では、クライアント・リクエストを、認証と許可の確認、スケジューリング、キャッシュ、および配布（カスタムまたはプラグابل・デスティネーションへの配布）など、各種サービスに割り当てます。また、Reports Server は、要求されたレポートを生成するランタイム・エンジンを起動したり、生成されたレポートを Reports Server キャッシュから取り出したり、ジョブの準備が完了したことをクライアントに通知したりします。
7. **Reports Server キャッシュ**では、完了済のジョブ出力が安全に格納されます。
8. **Reports Engine** は SQL ベースおよびプラグابل・データソースベースのレポートを実行するコンポーネントで構成されています。Reports Engine は、要求されたデータをデータソースから取り出したり、レポートをフォーマットしたり、出力をキャッシュに送信したり、ジョブの完了を Reports Server に通知したりします。
9. **プラグابل・エンジン**は、Java API を使用してジョブを Reports Server に渡すほか、スケジューリング、配布、通知、キャッシュなどのサーバー機能を活用するカスタム・エンジンです。OracleAS Reports Services は、URL エンジンと呼ばれる、特別な設定を必要としないプラグابل・エンジンを提供します。URL エンジンを使用すると、任意の公開 URL のコンテンツを電子メール、OracleAS Portal、WebDAV などの宛先に配布できます。

また **Oracle Reportsブリッジ**は、サブネットにわたって Reports Server を検出する機能を提供します。Oracle Reportsブリッジは、サブネットをまたいだ Reports Server および Reports クライアント間でブロードキャストされるパケットのゲートウェイとして機能します。図 1-1 には、ブリッジのメカニズムは示されていません。詳細は、第 1.4.1.2 項「複数のサブネットにわたるサーバーの検出」を参照してください。

1.3 OracleAS Reports Services のランタイム・プロセス

OracleAS Reports Services の各種コンポーネントがレポートの実行プロセスで果たす役割は、次のとおりです。

1. クライアントは、URL (Web) を使用するか、または Web を使用せずに Oracle Reports 関連のコマンド (rwclient など) を使用してサーバーに接続し、レポートを要求します。
 - URL を使用したリクエストは、JSP、rwservlet または rwcgi にリダイレクトされます。これらはすべて Oracle HTTP Server に接続されています。JSP と rwservlet のリクエストは、mod_oc4j にリダイレクトされます (JSP として実行されるジョブでは、mod_oc4j は OJSP を使用して JSP をサーブレットに変換します)。rwcgi リクエストは、CGI コンポーネントにリダイレクトされます。

URL には、ランタイム・パラメータまたは cgicmd.dat 内のランタイム・パラメータ構成セクションを参照するキーワードを含めることができます。両方を含めることもできますが、URL で明示的に指定されているパラメータが cgicmd.dat の関連キーワード・セクションに存在してはなりません。
 - rwclient は、Reports Server に直接リダイレクトされます。

コマンドラインには、ランタイム・パラメータを指定できます。ランタイム・パラメータが多数ある場合には、rwclient コマンドとパラメータ文字列を含むバッチ・ファイルまたはシェル・スクリプトを作成できます。
2. rwservlet (または下位互換性を確保するためにのみ維持される rwcgi) コンポーネントは、Web サーバーまたは J2EE コンテナ (OC4J など) と Reports Server との間で情報を次のように変換および配布します。
 - Reports JSP または rwservlet を使用したサーバー・リクエストは、インプロセス・サーバーで実行されるか、スタンドアロンの Reports Server プロセス (推奨) として実行されます。どちらであるかは、サーブレットの構成ファイル (ORACLE_HOME¥reports¥conf¥rwservlet.properties) で指定されます。インプロセス・サーバーは、スタンドアロンの Reports Server とは異なり、クライアントからのリクエストにตอบสนองして自動的に起動するため、メンテナンスが軽減されます。インプロセス・サーバーを使用すると、プロセス間の通信が削減されます。一方、スタンドアロン・サーバーは、OC4J インスタンスからサーバー・プロセスを切り離すことで、rwservlet の外部のプロセスをより高度に制御します。インプロセス・サーバーとデフォルトのネーミングの指定については、[第 3.4.10 項「インプロセス・サーバーの指定」](#) および [第 3.4.11 項「インプロセス・サーバーの識別」](#) を参照してください。
 - rwcgi を使用したサーバー・リクエストは、スタンドアロン・サーバーにリダイレクトされます。
3. Reports Server は次のようにリクエストを処理します。

リクエストに TOLERANCE オプションが含まれる場合は、Reports Server がキャッシュをチェックして、そのリクエストを満たす既存の出力があるかどうかを判別します。キャッシュ内にリクエストを満たす出力があった場合、サーバーはレポートの再実行よりも、その出力を返すことを優先して行います。

注意： Reports Server に送るどのジョブ・リクエストにも、TOLERANCE オプションを指定できます。TOLERANCE には、要求者が許容できる最も古い出力を定義します。たとえば、要求者が TOLERANCE として 5 分を設定した場合、Reports Server はキャッシュをチェックして過去 5 分以内に生成された最新の複製レポート出力を探します。EXPIRATION オプションには、レポート出力をキャッシュから削除する日時を定義します (たとえば、出力の期限切れと同じ日時を指定できます)。詳細は、[第 A.3.113 項「TOLERANCE」](#) および [第 A.3.35 項「EXPIRATION」](#) を参照してください。

リクエストが現在実行中のジョブと同じ場合、そのリクエストは、レポートを再実行せずに現在のジョブの出力を再利用します。

これらの条件が満たされない場合は、次のように処理が実行されます。

- a. Reports Servlet が SSO 対応の場合、認証が確認されます。次に、セキュアな Reports Server によって、Oracle Internet Directory を使用してユーザーが承認されます。Reports Servlet が SSO 対応でない場合は、セキュアな Reports Server によってユーザーが承認および認証されます。
- b. レポートがスケジュールされていれば、Reports Server はスケジュールされたジョブ・キューにリクエストを挿入し、レポートはスケジュールに従って実行されます。レポートがスケジュールされていなければ、リクエストは現在のジョブ・キューに挿入され、Reports Engine が利用可能になった時点で実行されます。

注意： `server_name.conf` で Reports Server を構成する際に、使用可能な Reports Engine の最大数を指定できます。Reports Server がその最大数を下回る場合は、ジョブをアイドル・エンジンに送信したり、新しいエンジンを起動してリクエストを処理したりできます。そうでない場合は、現在の Oracle Reports Engine のいずれかが現在のジョブを完了するまで、リクエストの実行を待つ必要があります。

- c. 実行時、Reports Server は Reports Engine を起動し、実行対象のリクエストをそのエンジンに送ります。
4. Reports Engine は、データを取り出し、フォーマットします。
5. Reports Engine は、Reports Server キャッシュに書き込みます。
6. Reports Engine は、レポートの準備が完了したことを Reports Server に通知します。
7. Reports Server は、URL、コマンドラインまたは `cgicmd.dat` ファイルのキーワード・セクション (URL リクエストのみ) で指定されたランタイム・パラメータに従って、キャッシュにアクセスし、出力するレポートを送ります。

また、イベント・ドリブンの公開を使用してレポートを作成することもできます。イベント・ドリブンの公開では、クライアントはエンド・ユーザーではなくデータベースです。イベントは、Event-Driven Publishing API で定義されます。イベントは、Event-Driven Publishing API をコールするデータベース・トリガー、アドバンスド・キューイング・アプリケーションまたは PL/SQL パッケージを起動して、ジョブを Reports Server に送ります。イベント・ドリブンの公開の詳細は、第 17 章「イベント・ドリブンの公開の使用法」を参照してください。

Oracle Workflow からのレポートの実行については、第 3.8 項「Oracle Reports と Oracle Workflow の通信の構成」を参照してください。詳細は、OTN

(<http://www.oracle.com/technology/products/reports/features/workflow>) にあるホワイト・ペーパー『Integrating Oracle Workflow with Oracle Reports』を参照してください。

1.4 OracleAS Reports Services の通信アーキテクチャ

Oracle9i Reports および Oracle Reports 10g (9.0.4) は、Borland 社の VisiBroker の osagent 実行可能ファイルを使用して、ネットワーク内の Reports Server を動的に検出します。

Oracle Reports 10g リリース 2 (10.1.2) では、Borland 社の VisiBroker に代わって、Sun 社の業界標準の Java Developer's Kit Object Request Broker (JDK ORB) が使用されています。これによって、クライアントからのサブネットにわたる Reports Server リクエストがサポートされ、さらにはブロードキャスト・メカニズムの使用によりサブネット内外での動的な Reports Server 検出が実現されます。

Oracle Reports 10g リリース 2 (10.1.2) では、特別な設定を必要としないビルトイン・ブロードキャスト・メカニズムを使用して Reports Server を動的に検出できます。Sun 社の JDK ORB で提供される Common Object Service (COS) ネーミング・サービス orbd を使用して Reports Server を検出することもできます。

注意： Reports Server の動的な検出には、ビルトイン・ブロードキャスト・メカニズムを使用することをお勧めします。Common Object Service (COS) ネーミング・サービスは、ビルトイン・ブロードキャスト・メカニズムが使用環境に適さない、次のような場合にのみ使用してください。

- VPN を使用してネットワークに接続しているマシン上に Oracle Reports をインストールする場合
 - ネットワーク上のブロードキャスト・トラフィックを回避する必要がある場合
-
-

この項では、Reports Server を検出する次の 2 つの方法について説明します。

- [ブロードキャスト・メカニズムを使用したサーバーの検出](#)
- [COS ネーミング・サービスを使用したサーバーの検出](#)

注意： Oracle Reports 10g リリース 2 (10.1.2) には、JDK ORB 実装を診断する rwdiag 実行可能ファイルも導入されています。rwdiag は、以前の VisiBroker 実装で使用可能であった osfind に代わる機能として使用できます。これにより、実行中の ORB アプリケーションに関する情報や、ORB 関連のネットワーク・トラフィックを記録するためのオプションが提供されます。rwdiag の詳細は、[付録 E 「Reports Server とブリッジの診断ユーティリティ」](#) を参照してください。

1.4.1 ブロードキャスト・メカニズムを使用したサーバーの検出

ブロードキャスト・メカニズムを使用すると、1つのサブネット内または複数のサブネットにわたって Reports Server を検出できます。

- サブネット内のサーバーの検出
- 複数のサブネットにわたるサーバーの検出

注意： Oracle Reports のビルトイン・ブロードキャスト・メカニズムでは、ホスト・マシンがネットワーク内にある必要があります。この制約によって、ブロードキャスト・メカニズムが機能しない可能性のある例を2つ、それぞれの解決策とあわせて次に示します。

1. ホスト・マシンがネットワーク内にない（つまり、スタンドアロンである）場合。

Windows プラットフォームでの解決策： MS ループバック・アダプタをインストールします。詳細は、Microsoft 社の Web サイト (<http://microsoft.support.com>) を参照してください。次のようにして、マシンの IP アドレスを指定します（XP または Windows 2000 の場合）。

1. デスクトップで「マイ ネットワーク」を右クリックして、「プロパティ」を選択します。
2. MS ループバック・アダプタを右クリックして、「プロパティ」を選択します。
3. 「プロパティ」ダイアログ・ボックスで「インターネットプロトコル (TCP/IP)」を選択して、「プロパティ」をクリックします。
4. 「次の IP アドレスを使用する」を選択して、有効な IP アドレスを入力します。「サブネット マスク」フィールドに、自動的に値が入力されます。ローカル・ホストの IP (127.0.0.1) は使用しないでください。たとえば、「198.162.1.1」と入力します。
5. 「OK」をクリックして、表示された指示に従います。

UNIX プラットフォームでの解決策： ネットワーク構成ファイル (rwnetwork.conf) の multicast 要素を namingService 要素に置き換えることによって、ビルトイン・ブロードキャスト・メカニズムを無効にし、かわりに COS ネーミング・サービスを有効にするように、検出メカニズムを構成します（詳細は、第 3.3.1.3 項「namingService」を参照）。

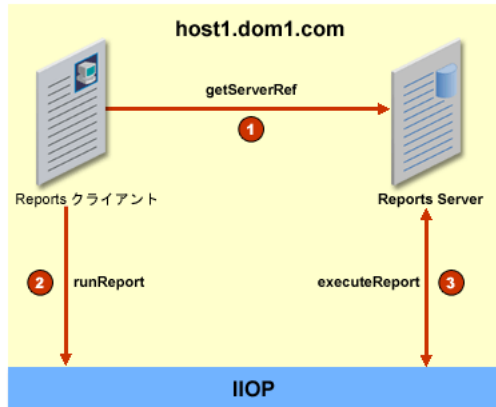
2. ホスト・マシンが、VPN 経由でネットワークに接続されている場合。

Windows および UNIX プラットフォームでの解決策： ネットワーク構成ファイル (rwnetwork.conf) の multicast 要素を namingService 要素に置き換えることによって、ビルトイン・ブロードキャスト・メカニズムを無効にし、かわりに COS ネーミング・サービスを有効にするように、検出メカニズムを構成します（詳細は、第 3.3.1.3 項「namingService」を参照）。

1.4.1.1 サブネット内のサーバーの検出

サブネット内で、クライアントは接続先の Reports Server の名前を指定したパケットをブロードキャストします。指定された名前の Reports Server がネットワーク上にある場合は、レスポンスが返されます。次に、クライアントはレスポンスを返した Reports Server に接続して、レポート・リクエストを実行します。

図 1-2 サブネット内のサーバーの検出



rep_server という名前のサーバーに対してレポート・リクエストを実行する例を次に示します。各手順の番号は図 1-2 の番号に対応しています。

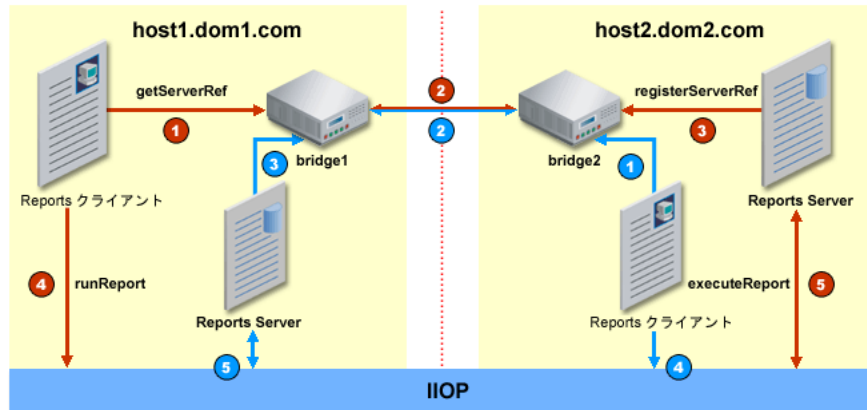
1. getServerRef
 - Oracle Reports クライアント (rwclient、rwservlet または rwrqm など) は、接続先のサーバー名を含むパケットをブロードキャストします。この例では、パケットに rep_server という名前が含まれます。
 - ネットワーク内の rep_server という名前のサーバーが、Interoperable Object Reference (IOR) でレスポンスを返します。
 - クライアントは、この IOR をオブジェクト参照に変換します。
2. runReport
 - Oracle Reports クライアントは、レポートを実行するように、リモートのオブジェクト参照にリクエストを送信します (IIOP コール)。
3. executeReport
 - Reports Server はレポートを実行して、レポートまたはステータスを返します。

1.4.1.2 複数のサブネットにわたるサーバーの検出

Oracle Reports では、セキュアでない複数のサブネットを接続するブリッジ・メカニズムが提供されます。あるサブネットで行われている Oracle Reports ブリッジが別のサブネットで行われている Oracle Reports ブリッジに接続して、Reports Server の参照を取得します。構成の詳細は、第 3.3.2 項「ブリッジの構成要素 (bridgeconf.dtd)」を参照してください。

トラブルシューティングの例と解決策については、第 D.8 項「Oracle Reports ブリッジの問題の診断」を参照してください。

図 1-3 複数のサブネットにわたるサーバーの検出



rep_server という名前のサーバーに対してレポート・リクエストを実行する例を次に示します。各手順の番号は図 1-3 の番号に対応しています。

1. getServerRef
 - Oracle Reports クライアント (rwclient、rwservlet または rwrqm など) は、接続先のサーバー名を含むパケットをブロードキャストします。この例では、パケットに rep_server という名前が含まれます。
2. bridge1 はパケットを傍受して bridge2 に渡します。
3. registerServerRef
 - bridge2 は dom2 でパケットをブロードキャストし、dom2 の rep_server は Interoperable Object Reference (IOR) でレスポンスを返します。
 - bridge2 は返された IOR を bridge1 に渡し、bridge1 はブロードキャスト・メカニズムを使用してこれをクライアントに渡します。
 - Oracle Reports クライアントは、この IOR をオブジェクト参照に変換します。
4. runReport
 - Oracle Reports クライアントは、レポートを実行するように、リモートのオブジェクト参照にリクエストを送信します (IIOP コール)。
5. executeReport
 - Reports Server はレポートを実行して、レポートまたはステータスを返します。

注意： 図 1-3 に青色で示された番号は、Oracle Reports クライアントが dom2 にあり、Reports Server が dom1 にある場合を示しています。

1.4.2 COS ネーミング・サービスを使用したサーバーの検出

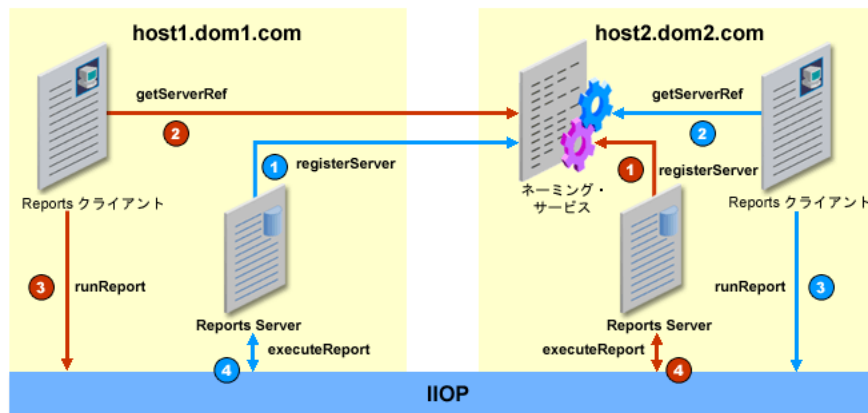
JDK で提供される Common Object Service (COS) ネーミング・サービスを使用して、同じサブネット内の Reports Server にアクセスしたり、セキュアでないサブネットにわたって Reports Server にアクセスしたりすることもできます。ネーミング・サービスを構成するには、第 3.3.1 項「ネットワークの構成要素 (rwnetworkconf.dtd)」を参照してください。

注意： Reports Server の動的な検出には、ビルトイン・ブロードキャスト・メカニズムを使用することをお勧めします。Common Object Service (COS) ネーミング・サービスは、ビルトイン・ブロードキャスト・メカニズムが使用環境に適さない、次のような場合にのみ使用してください。

- VPN を使用してネットワークに接続しているマシン上に OracleAS Reports Services をインストールする場合
- ネットワーク上のブロードキャスト・トラフィックを回避する必要がある場合

OPMN を使用して COS ネーミング・サービスを制御するには、第 3.7.1.5 項「COS ネーミング・サービスの指定」を参照してください。

図 1-4 COS ネーミング・サービスを使用したサーバーの検出



rep_server という名前のサーバーに対してレポート・リクエストを実行する例を次に示します。各手順の番号は図 1-4 の番号に対応しています。

1. registerServer
 - 起動時、rep_server は自身をネーミング・サービスに登録します。ネーミング・サービスは、サーバーを起動するために必ず稼動させておく必要があります。これで、rep_server に対してリクエストが実行されるようになります。
2. getServerRef
 - Oracle Reports クライアントは、ネーミング・サービスに接続して rep_server への参照を要求します。
 - ネーミング・サービスは、参照を rep_server に返します。
3. runReport
 - Oracle Reports クライアントは、レポートを実行するように、リモートのオブジェクト参照にリクエストを送信します (IIOP コール)。

4. executeReport

- Reports Server はレポートを実行して、レポートまたはステータスを返します。

注意： 図 1-4 に青色で示された番号は、Oracle Reports クライアントが dom2 にあり、Reports Server が dom1 にある場合を示しています。

1.5 システムの設定

OracleAS Reports Services の設定方法は、システムへの要求事項に応じて大きく異なります。OracleAS Reports Services を設定する前に、要求事項に基づいていくつかの決定を行う必要があります。事前にそのような決定を行うと、設定プロセスは非常に簡単になります。

次の各項では、これらの決定事項について説明します。

- 使用されるリクエスト・タイプの選択
- Reports Servlet、JSP、Web サービスまたは CGI の選択
- シングルマシン構成とマルチマシン構成の選択
- 高可用性環境の選択

1.5.1 使用されるリクエスト・タイプの選択

OracleAS Reports Services は、Web ベースのリクエストと非 Web ベースのリクエストの両方を受け入れるよう構成できます。

Web ベースの場合、Web ブラウザで URL をクリックするか入力することでレポートを実行できます。URL に従って、レポート出力はブラウザに返されるか、プリンタなど指定された宛先に送信されます。ユーザーがブラウザからレポートを起動できるようにするには、Reports Servlet、JSP または Reports CGI のいずれかのコンポーネントを Web サーバーで使用します。これらのコンポーネントのうちのいずれかと OracleAS Reports Services が通信し、Web クライアントからのレポート・リクエストの処理を可能にするには、そのコンポーネントが Web サーバー上に置かれている必要があります。

注意： 詳細は、第 1.5.2 項「Reports Servlet、JSP、Web サービスまたは CGI の選択」を参照してください。

非 Web ベースの場合、各ユーザー・マシンにインストールされている `rwclient` 実行可能ファイルを使用して、ジョブ・リクエストを送信できます。

構成の観点から見た、Web リクエストと非 Web リクエストの有効化の重要な相違点は、次のとおりです。

- Web リクエストを有効にする場合、Web サーバー側で Reports Servlet、JSP または Reports CGI（下位互換性を確保するためにのみ維持）を選択する必要がありますが、標準 Web ブラウザ以外にクライアント・ソフトウェアをインストールする必要はありません。

注意： Oracle Reports 10g では、Reports CGI (`rwcgi`) を使用することはお薦めしません（下位互換性を確保するためにのみ維持されています）。かわりに、Reports JSP、`rwervlet` (Reports Servlet) または Reports Web サービスを使用してください。

パフォーマンス上の理由により、`rwcgi` よりも `rwervlet` を使用することを強くお薦めします。`rwcgi` では、各リクエストに対して新規プロセスが起動されます。その際、JVM が初期化されるため、多数のレポート・リクエストを実行するとパフォーマンスが低下します。一方、`rwervlet` は OC4J インスタンスにデプロイされており、サーブレットの機能を利用するため、`rwcgi` よりも高いパフォーマンスを提供します。

- 非 Web リクエストを有効にする場合は、リクエストの実行に使用するクライアント・ソフトウェアを各マシンにインストールする必要があります。このため、各クライアント・マシンでクライアント・ソフトウェアを管理する必要があります。

Web ベースにすると、クライアントのメンテナンス・コストが削減されるため、最も費用効果が高くなります。ただし、非 Web リクエストの実行が必要な場合もあります。OracleAS Reports Services は、単一の実行環境で Web リクエストと非 Web リクエストの両方の実装をサポートします。

1.5.2 Reports Servlet、JSP、Web サービスまたは CGI の選択

OracleAS Reports Services を Web 環境で使用するには、サーブレット、JSP、または CGI を実装する必要があります。rwservlet (Reports Servlet)、Reports Web サービスまたは Reports JSP を選択することを強くお勧めします。これは、Reports CGI ではリクエストごとに新規プロセスが内部的に起動されるためです。起動された各プロセスによって JVM が初期化されます。そのため、多数のレポート・リクエストを実行する場合はパフォーマンスが低下します。一方、Reports Servlet は OC4J インスタンスにデプロイされており、サーブレットの機能を利用するため、Reports CGI よりも高いパフォーマンスを提供します。

Reports Servlet と Reports JSP のどちらを選択するかについては、さらに考慮が必要です。JSP のみを実装すると、Web 配信に適したレイアウト (Oracle Reports Web Layout) を公開できます。サーブレットを使用すると、ペーパー・レイアウトをレポート公開ソリューションに含めることができ、OracleAS Reports Services の配布機能を最大限に活用できます。

JSP ファイルには Web レイアウトとペーパー・レイアウトの両方を含めることができるため、Reports Servlet を使用しても JSP ファイルも使用できます。JSP に格納されているレポートを実行するときは、サーブレットを URL で指定し、JSP をコマンドライン・オプション `report=myreport.jsp` でコールします。この場合、レポート出力はペーパー・レイアウトに基づいて作成されます。

レポート実行の詳細は、[第 13 章「レポート・リクエストの実行」](#)を参照してください。

1.5.3 シングルマシン構成とマルチマシン構成の選択

OracleAS Reports Services は、Web サーバーと同じマシン上に置くことも、異なるマシン上に置くこともできます。どちらも長所と短所があります。

たとえば、OracleAS Reports Services と Web サーバーを同じマシンに置くと、マシンのメモリーとディスク領域の消費量が増えますが、ネットワークの通信量は減少します。これは、Web サーバーとアプリケーション・サーバー間でやり取りするリクエストがネットワークを経由せず、受信リクエストのみがネットワークを経由するためです。

インプロセス・サーバーを使用している場合 (Reports Servlet を実装している場合のみ使用可能)、シングル・マシンのパフォーマンス上の利点がさらに高まります。インプロセス・サーバーは、OracleAS Reports Services のコンポーネント間の通信をより高速化および効率化するので、処理時間が短縮されます。Reports Servlet を使用せずにレポートを配布する場合を除き、インプロセス・サーバーを使用することをお勧めします。

一方、シングルマシン構成を使用している場合に、そのマシンに障害が発生すると、すべてが失敗します。

Web サーバーとアプリケーション・サーバーが別々のマシンに置かれている場合、ネットワークの通信量が増大しますが、CPU、ディスク領域、利用可能なメモリーなど、システム・リソースが増えるという利点もあります。マルチマシン構成の場合も、インプロセス・サーバーがあると、OracleAS Reports Services のコンポーネント間の通信が高速化されるので、パフォーマンスが向上します。

また、Web サーバーとアプリケーション・サーバーをそれぞれ複数のマシンに置くという方法もあります。この方法には追加構成が必要ですが、Web サーバーのロード・バランシングが可能になります。

環境切替機能を使用すると、言語などの環境設定が異なる複数の Reports Engine を同じ Reports Server で起動できます。詳細は、[第 3.2.2 項「動的な環境切替え」](#)を参照してください。

1.5.4 高可用性環境の選択

この項では、高可用性環境の選択における次の側面について説明します。

- Oracle Application Server での高可用性の維持
- インフラストラクチャの依存性の維持

1.5.4.1 Oracle Application Server での高可用性の維持

Oracle Application Server は、分散トポロジに配置可能な様々なコンポーネントで構成されています。Oracle Application Server では、高可用性を実現するための基礎となるパラダイムとしてクラスタリングを使用しています。クラスタリングでは、Oracle Application Server の各種コンポーネントが一定の組合せで結合され、スケーラブルな統合機能を提供するとともに、個々のコンポーネントのいずれかで障害が発生した場合に備えて冗長性を提供します。

注意： Oracle Application Server での高可用性を実現する様々なソリューションと技法の詳細は、『Oracle Application Server エンタープライズ・デプロイメント・ガイド』および『Oracle Application Server 高可用性ガイド』を参照してください。

このマニュアルを読み進める前に、『Oracle Application Server 概要』を通読して、Oracle Application Server の様々なコンポーネントに関する知識を得ておくことをお勧めします。高可用性を実現する複数のソリューションと技法を導入することにより、次の目標を達成できます。

冗長性

高可用性システムは、冗長性を確保するためにそのサブシステムを必要とします。『Oracle Application Server 高可用性ガイド』に記載されている手順とソリューションを使用すると、Oracle Application Server のすべてのコンポーネントを冗長に配置できます。コンポーネントは、そのタイプに応じて、アクティブ / アクティブ構成で配置することもアクティブ / パッシブ構成で配置することもできます。

アクティブ / アクティブ構成では、コンポーネントの複数のインスタンスがクライアント・リクエストを同時に処理します。あるインスタンスで障害が発生した場合、そのインスタンスが処理していたリクエストをアクティブな別のインスタンスで実行できます。このインスタンスの障害とフェイルオーバーは、クライアントに対して透過的です。通常、アクティブ / アクティブ構成は、コンポーネントのインスタンスをまとめてクラスタリングすることによって実現します。

アクティブ / パッシブ構成では、通常、リクエストはコンポーネントの1つのインスタンスによって処理されます。そのコンポーネントで障害が発生すると、別のインスタンスがアクティブ化されて、リクエストのワークロードに応答します。

障害の検出と自動再起動

Oracle Application Server コンポーネントに属するローカルまたは分散ソフトウェア・プロセスは、集中的なプロセス管理システムによって管理されます。このプロセス管理システムでは、プロセスの障害を検出して、プロセスが複数のマシンに分散している場合でもプロセスを再起動できます。このシステムを使用すると、プロセスの障害と再起動を定義するパラメータ値（ハートビートの回数など）をカスタマイズできます。プロセス管理システムを実装しているプロセスは、それぞれシャドウ・プロセスを備えており、冗長性を実現しています。

クラスタリング

システムのコンポーネントをまとめてクラスタリングすることにより、クライアント側は、これらのコンポーネントを機能的に1つのエンティティと見なすことができます。クラスタにより、コンポーネントのスケラビリティ、可用性および管理性が向上します。

Oracle Application Server コンポーネントでは、複数のクラスタ・タイプを使用できます。これらのクラスタを作成および構成する手順は、『Oracle Application Server 高可用性ガイド』で詳しく説明しています。『Oracle Application Server エンタープライズ・デプロイメント・ガイド』には、クラスタ環境の設定に関する情報も含まれています。

状態のレプリケーションとルーティング

ステートフル・クライアント・リクエストでは、これらのリクエストを処理しているプロセスに障害が発生した場合、Oracle Application Server は、リクエストのステートフル・フェイルオーバーを有効にするためにクライアント状態をレプリケートできます。J2EE リクエストでは、使用する方法に応じて、J2EE アプリケーションのクライアント状態を宣言的にレプリケートすることも、プログラムによってレプリケートすることもできます。その他の大部分のコンポーネントでは、Cookie を使用した状態ベースのルーティングを使用できます。

接続障害管理

多くの場合、クライアントはサーバー上のサービスに接続し、その接続を再利用します。サーバー上のサービスのいずれかを実装しているプロセスを再起動すると、接続の再確立が必要になることがあります。

Oracle Application Server コンポーネントでは、再利用の接続に失敗すると、失敗の状態がシステムの他の部分に伝播する前に再接続が試行されます。これにより、接続の失敗をクライアントから透過的にすることができます。

バックアップとリカバリ

Oracle Application Server は、システム状態をバックアップして障害からのリカバリに使用する機能を備えています。状況によっては、コンポーネントまたはシステムの障害を修復できないこともあります。Oracle Application Server の Backup and Recovery Tool を使用すると、システムを一定の間隔でバックアップして、修復不可能な障害が発生したときにバックアップをリストアできます。

HTTP リスナーと J2EE コンテナに固有の問題では、ランタイム構成管理システムを使用して、これらのコンポーネントに簡単にチェックポイントを設定できます。さらに、構成エラーを元に戻す操作も可能です。

障害時リカバリ

重要なアプリケーションをホストしている Oracle Application Server サイトが物理的に存在する場所で、自然災害や物理的な障害が発生することもあります。『Oracle Application Server 高可用性ガイド』では、こうした障害からリカバリするためのソリューションについて説明しています。このソリューションは、サイト間リカバリ・ソリューションです。このソリューションでは、Oracle Application Server のサイト全体の状態をバックアップして、そのサイトから物理的に離れている別のサイトにリカバリすることができます。

1.5.4.2 インフラストラクチャの依存性の維持

Oracle Application Server には多数の高可用性の機能が備えられているため、特定のサーバーやコンポーネントに障害が発生しても、その中間層での稼動を継続することができます。Oracle Application Server の高可用性の機能を使用することを強くお勧めします。

注意： Oracle Application Server の高可用性機能の詳細は、『Oracle Application Server 高可用性ガイド』を参照してください。

Oracle Application Server の高可用性の機能を活用して、OracleAS Reports Services では、インフラストラクチャの依存性が損なわれると、次のアクションが実行されます。

- **OracleAS Portal データベース接続の再試行。** Reports Server から OracleAS Portal データベース・スキーマへの接続がなんらかの原因で途絶えている場合、Reports Server は、エラーを生成する前に接続の再確立を試行します。まず、Reports Server は、リポジトリから OracleAS Portal 接続文字列を取得します。OracleAS Portal 接続文字列があれば、Reports Server は再接続を試行できます。再接続が成功した場合、Reports Server の再起動は不要です。また、OracleAS Reports Services は、インフラストラクチャでのコールド・フェイルオーバーおよび RAC、中間層での障害時リカバリもサポートします。
- **Oracle Internet Directory 接続の再試行。** Oracle Internet Directory 接続がなんらかの原因によって無効な状態になった場合、Reports Servlet および Reports Server は、エラーを生成する前に接続の再確立を試行します。再接続が成功した場合、Reports Server の再起動は不要です。

OracleAS Reports Services の起動と停止

この章では、OracleAS Reports Services の起動および停止について説明します。この章の主な項は、次のとおりです。

- Reports Server の起動と停止
- Oracle Reports ブリッジの起動と停止
- COS ネーミング・サービスの起動と停止
- Oracle HTTP Server が実行されていることの確認
- Reports Servlet および Reports Server が実行されていることの確認

注意： この章の各例で使用されている `ORACLE_HOME` は、Oracle Application Server がインストールされている場所を示しています。これには OracleAS Reports Services が含まれます。

2.1 Reports Server の起動と停止

Reports Server を実行する方法として、Oracle Process Manager and Notification Server (OPMN) の使用を第一にお勧めします。OPMN は、Oracle HTTP Server、OC4J プロセスおよび OracleAS Reports Services の初期化、維持、停止を一元化したメカニズムを提供します。OPMN を使用した Reports Server の構成の詳細は、[第 3.7 項「Oracle Process Manager and Notification Server および Oracle Enterprise Manager 10g による Reports Server の構成」](#)を参照してください。

重要： Oracle Enterprise Manager 10g に登録している Reports Server は、Oracle Enterprise Manager 10g または OPMN を使用してのみ起動または停止してください。なんらかの理由で Reports Server が応答しなくなった場合、OPMN はこれを自動的に再起動します。Windows では、OPMN は Windows サービスとして実行されます。

Oracle Reports 10g リリース 2 (10.1.2) から、Windows サービスとしての Reports Server の実行はサポートされなくなりました (`rwserver -install server_name`)。これにより、関連するコマンドライン・キーワード `INSTALL` および `UNINSTALL` の使用も廃止されています。Windows サービスとして実行している OPMN またはコマンドラインを使用して Reports Server を管理している場合、サーバーを起動または停止すると次の問題が発生することがあります。

- Reports Server のステータスが Oracle Enterprise Manager 10g に正確に反映されない。
- Reports Server を起動または停止すると、Oracle Enterprise Manager 10g でエラーが表示される場合がある。

Windows サービスとしての Reports Server の実行の廃止に関する詳細は、OTN (Oracle Technology Network) にある『[A Guide to Changed Functionality Between Oracle Reports 6i and 10g](#)』を参照してください。

2.1.1 Oracle Enterprise Manager 10g からの Reports Server の起動、停止および再起動

OPMN によってスタンドアロンの Reports Server が構成された場合、Oracle Enterprise Manager 10g から起動、停止および再起動できます。

注意： OC4J_BI_Forms の一部として使用可能なインプロセス・サーバーは、Oracle Application Server のインストール時に OPMN で自動的に構成され、Oracle Enterprise Manager 10g に登録されます。Oracle Application Server をインストールした後で Reports Server を追加する場合は、新しいサーバーを、Oracle Enterprise Manager 10g の `targets.xml` ファイルおよび Oracle Process Manager and Notification Server の `opmn.xml` ファイルに登録する必要があります。または、`addnewservertarget.bat` を実行して新しいサーバーを登録することもできます。詳細は、[第 3.7 項「Oracle Process Manager and Notification Server および Oracle Enterprise Manager 10g による Reports Server の構成」](#)を参照してください。

Reports Server を起動、停止または再起動する手順は、次のとおりです。

1. Reports Server のメイン・ページで次の操作を行います。

- サーバーを起動する場合は、「**起動**」をクリックします。
- サーバーを停止する場合は、「**停止**」をクリックします。
- サーバーを再起動する場合は、「**再起動**」をクリックします。

これらのボタンは、サーバーの現在の状態に応じて Reports Server のメイン・ページに表示されます。

- サーバーが停止している場合は、「**起動**」ボタンと「**停止**」ボタンが表示されます。
- サーバーが稼動している場合は、「**再起動**」ボタンと「**停止**」ボタンが表示されます。

注意： Oracle Enterprise Manager では、すべての Reports Server の依存性が Reports Server のメイン・ページにリストされます。

2.1.2 Oracle Process Manager and Notification Server からの Reports Server の起動、停止および再起動

Reports Server が Oracle Process Manager and Notification Server で構成されていれば、次のコマンドラインを使用して、Reports Server を起動、停止および再起動できます。

```
ORACLE_HOME/opmn/bin/opmnctl startproc ias-component=reports_server_name
ORACLE_HOME/opmn/bin/opmnctl startproc process-type=reports_server_name
ORACLE_HOME/opmn/bin/opmnctl stopproc ias-component=reports_server_name
ORACLE_HOME/opmn/bin/opmnctl restartproc ias-component=reports_server_name
```

Reports Server 名は、opmn.xml ファイル内の ias-component id にある名前と一致している必要があります。

Oracle Process Manager and Notification Server のステータスを問い合わせるには、次のコマンドを使用します。

```
ORACLE_HOME/opmn/bin/opmnctl status
```

Oracle Process Manager and Notification Server を使用した Reports Server の構成の詳細は、[第 3.7 項「Oracle Process Manager and Notification Server および Oracle Enterprise Manager 10g による Reports Server の構成」](#)を参照してください。

2.1.3 Reports Server を起動および停止する別の方法

Reports Server を OPMN から実行せずに、Oracle Enterprise Manager 10g を使用して管理する場合は、以前からある次の方法で Reports Server を実行できます。

- [インプロセス・サーバーの起動 \(Windows および UNIX\)](#)
- [コマンドラインからの Reports Server の起動 \(Windows および UNIX\)](#)
- [Reports Server の停止](#)

重要： 前述のとおり、Oracle Reports 10g リリース 2 (10.1.2) から、Windows サービスとしての Reports Server の実行はサポートされなくなりました。

2.1.3.1 インプロセス・サーバーの起動（Windows および UNIX）

Reports Server をインプロセス・サーバー（デフォルトの構成）として使用している場合は、レポート実行リクエストを送信するとインプロセス・サーバーが起動されます。ただし、コマンドラインを介してリクエストを送信する場合は、最初にレポート実行 URL または Web コマンド URL を使用してサーブレットを起動する必要があります。サーブレットの起動に成功した場合は、インプロセス・サーバーの起動も成功したことになります。

URL からインプロセス・サーバーを直接起動するには、Web ブラウザで次のように入力します。

```
http://your_machine_name:your_port_num/reports/rwervlet/startserver
```

2.1.3.2 コマンドラインからの Reports Server の起動（Windows および UNIX）

Windows では、次のコマンドを使用して、Reports Server をスタンドアロン・サーバーとして起動することもできます。

```
rwserver server=server_name
```

サーバーを起動する際に、ダイアログ・ボックスやメッセージを表示しない場合は、コマンドライン・キーワード BATCH を追加します。

```
rwserver server=server_name batch=yes
```

UNIX では、次の構文を使用してこのコマンドを実行できます。

```
rwserver.sh server=server_name
```

または

```
rwserver.sh server=server_name batch=yes
```

重要： DISPLAY が設定されていない場合は、Reports Server をバッチ・モード (batch=yes) で起動する必要があります。

UNIX での DISPLAY およびプリンタ依存性の排除に関する詳細は、[第 3.10 項「UNIX での DISPLAY およびプリンタ依存性の解消」](#)を参照してください。

REPORTS_DEFAULT_DISPLAY 環境変数の詳細は、[第 B.1.39 項「REPORTS_DEFAULT_DISPLAY」](#)を参照してください。

シェル・スクリプトが PATH 環境変数でアクセスできれば、このコマンドは任意のディレクトリから実行できます。

2.1.3.3 Reports Server の停止

Windows および UNIX で Reports Server を停止するには、次のようにいくつかの方法があります。

- Windows で rwserver 実行可能ファイルを使用して Reports Server を実行している場合、または UNIX でシェル・スクリプト rwserver.sh を使用して Reports Server を実行している場合は、Reports Server ダイアログ・ボックスの「**停止**」ボタンをクリックします。
- Reports Server をコマンドラインから実行していない場合は、Oracle Enterprise Manager 10g を起動して、停止する Reports Server にナビゲートし、選択した Reports Server のホーム・ページで「**停止**」をクリックします。Reports Server および Oracle Enterprise Manager 10g の詳細は、[第 19 章「OracleAS Reports Services の管理と監視」](#)を参照してください。

- Reports Servlet を介して Reports Server をインプロセス・サーバーとして実行している場合は、次の URL を発行します。

```
http://your_host_name:port_number/reports/rwervlet/stopserver
```

- Windows または UNIX で Reports Server をコマンドラインから実行している場合は、Reports Server の停止方法に応じて、次のいずれかのコマンドを使用します。

注意： UNIX では、`rwserver` のかわりに `rwserver.sh` を使用します。

サーバーを正常に停止（保留中のジョブを終了してから停止）するには、次のコマンドを実行します。

```
rwserver server=server shutdown=normal authid=username/password
```

サーバーをただちに停止（保留中のジョブを終了せずに停止）するには、次のコマンドを実行します。

```
rwserver server=server shutdown=immediate authid=username/password
```

関連メッセージを表示せずにサーバーを停止するには、次のコマンドを実行します。

```
rwserver server=server shutdown=normal authid=username/password batch=yes
```

`rwserver` コマンドで使用されるキーワードは、付録 A 「コマンドライン・キーワード」で説明します。

注意： `authid` は、Reports Server の管理ユーザー名およびパスワードです。セキュアな Reports Server の場合は、このユーザーが、Oracle Internet Directory の `RW_ADMINISTER` 権限グループのメンバーである必要があります。セキュアでない Reports Server の場合、このユーザーは、`identifier` 要素内で定義されます。セキュアでない Reports Server をコマンドラインから停止する方法を、次で説明します。

- セキュアでない Reports Server をコマンドラインから `rwserver.sh` または `rwrqv.sh` を使用して停止する場合は、サーバー構成ファイルの `identifier` 要素に設定された値と一致する有効な `authid` を指定する必要があります。ただし、`identifier` 要素は、Oracle Application Server 10g のインストール時、Reports を構成する際に設定され、Reports Server によって暗号化されています。この `identifier` 要素は任意の値に変更できます。この Reports Server を Oracle Enterprise Manager 10g および OPMN に登録している場合は、Oracle Enterprise Manager 10g との統合を可能にするため、`targets.xml` で対応するプロパティを変更する必要があります。この手順は次のとおりです。

1. セキュアでない Reports Server の構成ファイル `server_name.conf` で、`identifier` 要素を変更して `username/password` を指定し、`encrypted` 属性を `no` に設定します。次に例を示します。

```
<identifier confidential="yes" encrypted="no">scott/tiger</identifier>
```

2. Reports Server を手動でいったん停止してから再起動し、`server_name.conf` ファイルに行った変更を有効にします。

注意： 構成の変更を有効にするには、Reports Server を再起動する必要があります。

これで、Reports Server で、`identifier` 要素の `username/password` 値が暗号化されます。Reports Server で `server_name.conf` ファイルの変更内容が読み取られると、次のコマンドが正常に実行されます (`username/password` が `scott/tiger` であると想定します)。

```
./rwsrvr.sh server=server_name shutdown=normal authid=scott/tiger
./rwrqvy.sh server=server_name shutdown=normal authid=scott/tiger
```

3. Oracle Enterprise Manager 10g との統合を可能にするため、任意のテキスト・エディタを使用して `targets.xml` ファイル (`ORACLE_HOME/sysman/emd/`) を次のように編集します。
 - `TYPE="oracle_repsrv"` および `DISPLAY_NAME="Reports Server: server_name"` の target を見つけます。
 - このエントリにある `UserName` プロパティと `Password` プロパティを、`server_name.conf` ファイルにある `identifier` 要素と同じユーザー名とパスワードに設定します。両方のプロパティの `ENCRYPTED` 属性を `FALSE` に設定します。
 - 変更を有効にするために、Oracle Enterprise Manager 10g を再起動します。

これで、Oracle Enterprise Manager 10g を使用してセキュアでない Reports Server を停止できます。

注意： これらの手順は、セキュアでない Reports Server にのみ必要なもので、セキュアな Reports Server には必要ありません。

2.2 Oracle Reports ブリッジの起動と停止

Oracle Reports ブリッジは、2つのサブネットを接続するために使用されます。ブリッジは、異なるサブネットで行われている Oracle Reports コンポーネント間のゲートウェイとして機能します。

注意： Oracle Reports 10g リリース 2 (10.1.2) では、Oracle Reports ブリッジは Oracle Enterprise Manager 10g に統合されていません。このため、Oracle Enterprise Manager 10g Application Server Control を使用して Oracle Reports ブリッジのステータスを表示したり、Oracle Reports ブリッジを起動または停止することはできません。

トラブルシューティングの例と解決策については、[第 D.8 項「Oracle Reports ブリッジの問題の診断」](#) を参照してください。

2.2.1 Oracle Process Manager and Notification Server からの Oracle Reports ブリッジの起動、停止および再起動

Oracle Process Manager and Notification (OPMN) で Oracle Reports ブリッジを起動するには、次の例に示すように、ブリッジを OPMN に追加しておく必要があります。

```
cd $ORACLE_HOME/bin
setenv ORACLE_HOME youroraclehome
addNewReportsBridge.sh bridgename
cd $ORACLE_HOME/opmn/bin
opmnctl reload (If opmn is up and running)
opmnctl start (If opmn is not running)
```

Oracle Reports ブリッジが Oracle Process Manager and Notification (OPMN) Server で構成されている場合、Oracle Reports ブリッジを起動するには、次のいずれかのコマンドを使用します。

```
ORACLE_HOME/opmn/bin/opmnctl startproc ias-component=bridgename
ORACLE_HOME/opmn/bin/opmnctl startproc process-type=bridgename
```

Oracle Reports ブリッジを停止するには、次のコマンドを使用します。

```
ORACLE_HOME/opmn/bin/opmnctl stopproc ias-component=bridgename
```

Oracle Reports ブリッジを再起動するには、次のコマンドを使用します。

```
ORACLE_HOME/opmn/bin/opmnctl restartproc ias-component=bridgename
```

Oracle Reports ブリッジ名は、opmn.xml ファイル内の ias-component id にある名前と一致している必要があります。

Oracle Process Manager and Notification ブリッジのステータスを問い合わせるには、次のコマンドを使用します。

```
ORACLE_HOME/opmn/bin/opmnctl status
```

Oracle Process Manager and Notification Server を使用した Oracle Reports ブリッジの構成の詳細は、[第 3.7.1.4 項「Oracle Reports ブリッジの指定」](#)を参照してください。

2.2.2 コマンドラインからの Oracle Reports ブリッジの起動と停止

コマンドラインから Oracle Reports ブリッジを起動するには、次のコマンドを使用します。

Windows の場合：

```
rwbridge.bat name=bridgename
```

UNIX の場合：

```
rwbridge.sh name=bridgename
```

たとえば、foo という名前の Oracle Reports ブリッジを Windows で起動するには、次のコマンドを使用します。

```
rwbridge.bat name=foo
```

関連項目： rwbridge 実行可能ファイルの詳細は、次の項を参照してください。

- [第 A.2.9 項「rwbridge」](#)

Oracle Reports ブリッジが初めて起動するとき、Oracle Reports で構成ファイル repbrg_bridgename.conf が作成されます。このファイルは rwbridge.template ファイルの設定内容に基づいて生成され、ORACLE_HOME/reports/conf ディレクトリに保存されます。repbrg_bridgename.conf ファイルを編集して、他のサブネットに接続するリモートの Oracle Reports ブリッジを指定します。

注意： 構成の変更を有効にするには、Oracle Reports ブリッジを再起動する必要があります。

Oracle Reports ブリッジを停止するには、次のコマンドを使用します。

Windows の場合：

```
rwbridge.bat name=bridgename shutdown=normal authid=username/password
```

UNIX の場合：

```
rwbridge.sh name=bridgename shutdown=normal authid=username/password
```

たとえば、foo という名前の Oracle Reports ブリッジを UNIX で停止するには、次のコマンドを使用します。

```
rwbridge.sh name=foo shutdown= normal authid=scott/tiger
```

構成ファイル `repbrg_bridgename.conf` で、`identifier` 要素を変更して `username/password` を指定し、`encrypted` 属性を `no` に設定します。これは、パスワードが暗号化されないことを示します。パスワードは Oracle Reports ブリッジが起動すると暗号化されます。

次に例を示します。

```
<identifier confidential="yes" encrypted="no">scott/tiger</identifier>
```

使用上の注意

- `identifier` 要素がコメント化されている場合は、`authid` を指定せずに Oracle Reports ブリッジを停止できます。
- Oracle Reports ブリッジをリモートで停止することはできません。

関連項目： [第 3.3.2.3 項「identifier」](#)

2.3 COS ネーミング・サービスの起動と停止

デフォルトのブロードキャスト・メカニズムのかわりに Sun 社の JDK で提供される Common Object Service (COS) ネーミング・サービス `orbd` を使用して Reports Server を検出できます。`orbd` 実行可能ファイルの詳細は、Sun 社の Web サイト (<http://java.sun.com>) にある `JavalDL` のページを参照してください。

ネーミング・サービスを起動するには、次のコマンドを使用します。

Windows の場合：

```
namingservice.bat port_number
```

UNIX の場合：

```
namingservice.sh port_number
```

ネーミング・サービスは、指定されたポート番号を使用して起動します。ネーミング・サービスを使用するには、Reports Server の検出メカニズムを構成する必要があります。Reports Server の検出メカニズムの構成については、[第 3.3 項「Reports Server 検出メカニズムの構成」](#)を参照してください。

ネーミング・サービスを停止するには、次のコマンドを使用します。

Windows の場合：

```
namingservice.bat port_number shutdown
```

UNIX の場合：

```
namingservice.sh port_number shutdown
```

OPMN を使用してネーミング・サービスを制御するには、[第 3.7.1.5 項「COS ネーミング・サービスの指定」](#)を参照してください。

2.4 Oracle HTTP Server が実行されていることの確認

OracleAS Reports Services は Oracle HTTP Server コンポーネントに依存しています。Oracle Enterprise Manager 10g または OPMN で Reports Server を起動する前に、Oracle HTTP Server が稼動していることを確認してください。Oracle Enterprise Manager 10g での確認方法の詳細は、Oracle Enterprise Manager 10g のドキュメントを参照してください。

あるいは、ブラウザで次の URL にナビゲートすることによっても、Oracle HTTP Server が実行されていることを確認できます。

```
http://server_name.domain:port_number/
```

2.5 Reports Servlet および Reports Server が実行されていることの確認

Reports Servlet が実行されていることを確認するには、次の URL にナビゲートします。

```
http://your_machine_name.domain_name:your_port_number/reports/rwservlet/help
```

URL は、大文字と小文字を区別することに注意してください。この URL が正常に実行されると、rwservlet コマンドライン引数を説明するヘルプ・ページが表示されます。

Reports Server が実行されていることを確認するには、次の URL にナビゲートします。

```
http://your_machine_name.domain_name:your_port_number/reports/rwservlet/getserverinfo?server=server_name
```

デフォルトの Reports Server 名 (rep_machine_name) を使用している場合、またはサブレット構成ファイル rwservlet.properties (ORACLE_HOME¥reports¥conf¥) で指定されている Reports Server を使用している場合、server=server_name 引数は必要ありません。この URL が正常に実行されると、指定した Reports Server のジョブ・キューのリストが表示されます。

注意： サブレット構成ファイルの詳細は、[第 3.4 項「Reports Servlet の構成」](#)を参照してください。

OracleAS Reports Services の構成

Oracle Application Server をインストールすると、OracleAS Reports Services が自動的に構成されます。環境をカスタマイズするために調整が必要なこともありますが、環境全体または大部分を設定する必要はありません。

この章では、主に、カスタマイズを行ったり、デフォルト構成に対する理解を深める際の参考となる情報を提供します。サービス関連の構成ファイルを示し、ほとんどのファイルの内容を詳細に説明します。この章の主な項は、次のとおりです。

- [OracleAS Reports Services 構成ファイル](#)
- [Reports Server の構成](#)
- [Reports Server 検出メカニズムの構成](#)
- [Reports Servlet の構成](#)
- [URL エンジンの構成](#)
- [プロキシ情報の入力](#)
- [Oracle Process Manager and Notification Server および Oracle Enterprise Manager 10g による Reports Server の構成](#)
- [Oracle Reports と Oracle Workflow の通信の構成](#)
- [レポート配布の最適化](#)
- [UNIX での DISPLAY およびプリンタ依存性の解消](#)

注意： この章の各例で使用されている `ORACLE_HOME` は、Oracle Application Server がインストールされている場所を示しています。Oracle Application Server には、OracleAS Reports Services が含まれます。

構成の際には、環境変数も設定します。環境変数は、インストール時に自動的に設定されます。環境変数の詳細は、[付録 B 「環境変数」](#) を参照してください。

3.1 OracleAS Reports Services 構成ファイル

この項では、OracleAS Reports Services の構成ファイルについて説明します。多くの場合、これらのファイルをユーザーが操作する必要はありません。これらのファイルは、サーバー環境の多くの部分をコントロールするので、ユーザーが不適切に変更すると、サーバー環境に不具合が発生する可能性があります。構成ファイルをユーザー自身で変更する場合は、必ず現行バージョンのバックアップを作成してください。

OracleAS Reports Services の構成ファイルは、Reports Server (rwservlet) および Reports Servlet (rwservlet) に関連します。これらは、表 3-1 で説明します。

注意： 表 3-1 で指定されているパスは、Windows を前提として円記号 (¥) を使用していますが、Windows および UNIX のどちらの環境でも同じパスです。

表 3-1 OracleAS Reports Services 構成ファイル

コンポーネント	構成ファイル
Reports Server	<p>ORACLE_HOME¥reports¥dtd¥rwservletconf.dtd ORACLE_HOME¥reports¥conf¥server_name.conf</p> <p>rwservletconf.dtd ファイルには、server_name.conf および rebuild.conf の要素と属性のデータ型の定義が格納されています。第 3.2.1 項「Reports Server の構成要素 (rwservletconf.dtd)」を参照してください。</p> <p>server_name.conf ファイルを使用して、Reports Server キャッシュ、Oracle Reports Engine およびセキュリティの初期値の定義、有効な宛先タイプの登録、ログを記録する情報の指定、およびその他のサーバー関連値の設定を行います。このファイルは、サーバー起動時に自動的に作成されます。サーバー名を変更する際、このファイルに入力したカスタム構成設定を保持する場合は、このファイルを新たなサーバー名に変更してからサーバー名を変更します。この順序で行わないと、新しいデフォルト構成ファイルがサーバーによって自動的に作成されます。</p> <p>詳細は、第 3.2 項「Reports Server の構成」を参照してください。</p>
Reports Server (ネットワーク構成)	<p>ORACLE_HOME¥reports¥dtd¥rwnetworkconf.dtd ORACLE_HOME¥reports¥conf¥rwnetwork.conf</p> <p>rwnetworkconf.dtd には、rwnetwork.conf の要素と属性のデータ型の定義が格納されています。第 3.3.1 項「ネットワークの構成要素 (rwnetworkconf.dtd)」を参照してください。</p>
Reports Server (Oracle Reports ブリッジ構成)	<p>ORACLE_HOME¥reports¥dtd¥bridgeconf.dtd ORACLE_HOME¥reports¥conf¥repbrg_bridgename.conf</p> <p>bridgeconf.dtd ファイルには、rwbridge_bridgename.conf の要素と属性のデータ型の定義が格納されています。第 3.3.2 項「ブリッジの構成要素 (bridgeconf.dtd)」を参照してください。</p>

表 3-1 OracleAS Reports Services 構成ファイル (続き)

コンポーネント	構成ファイル
Reports Builder Reports Runtime	<p><code>ORACLE_HOME\reports\dt\rwserverconf.dtd</code> <code>ORACLE_HOME\reports\conf\rbuilder.conf</code></p> <p><code>rwserverconf.dtd</code> ファイルには、<code>server_name.conf</code> および <code>rebuilder.conf</code> の要素と属性のデータ型の定義が格納されています。第 3.2.1 項「Reports Server の構成要素 (rwserverconf.dtd)」を参照してください。</p> <p><code>rebuilder.conf</code> を使用して、Reports Builder および Reports Runtime に埋め込まれている Reports Server を構成します。リクエストはすべて Reports Server 経由で実行される必要があるため、レポートを実行するサーバーが Reports Builder に必要です。Reports Builder は、リクエストを処理する Reports Server を自動的に起動します。Reports Builder からレポートを実行する際、このファイルは起動するインプロセスの Reports Server インスタンスに構成を提供します。<code>server_name.conf</code> ファイルと同様、このファイルはそのデータ型定義を <code>rwserverconf.dtd</code> ファイルに依存します。ただし、<code>compatible</code>、<code>persistFile</code> および <code>security</code> の各要素は除きます。</p> <p>このファイルは <code>server_name.conf</code> と同じ構成要素を多く含むので、このファイルの構成に必要な情報は、第 3.2 項「Reports Server の構成」を参照してください。</p>
Reports Servlet	<p><code>ORACLE_HOME\reports\conf\rwservlet.properties</code></p> <p><code>rwservlet.properties</code> ファイルには、Reports Servlet (<code>rwservlet</code>) の構成に使用するパラメータが格納されています。このファイルの詳細は、第 3.4 項「Reports Servlet の構成」を参照してください。</p>

3.2 Reports Server の構成

OracleAS Reports Services の Reports Server コンポーネントは、`server_name.conf` および `rebuilder.conf` の各 XML ファイルを使用して構成できます。これらのファイルは、Windows および UNIX とともに次のディレクトリにあります。

`ORACLE_HOME\reports\conf`

これらのファイルはどちらも、Windows および UNIX においてデフォルトのサーバー構成値を含む `rwserver.template` ファイル (同じディレクトリにある) によってサポートされています。

`server_name.conf` ファイルは、デフォルトのサーバー構成ファイルです。

`rebuilder.conf` ファイルは、Reports Builder によってインプロセスで使用されるサーバー・インスタンスを構成します。

`server_name.conf` ファイルおよび `rebuilder.conf` ファイルは、ほぼ同じです。唯一の違いは、`rebuilder.conf` は `compatible`、`persistFile` または `security` の構成要素 (この項で後述) を使用しないことです。

これらのファイルは、次の条件で自動的に作成されます。

- `server_name.conf` ファイルは、Reports Server を最初に起動したときに作成されます。このファイルは、`rwserver.template` ファイルをベースにしています。
- `rebuilder.conf` ファイルは、Reports Builder からレポートを最初に実行したときに作成されます。このファイルも、`rwserver.template` ファイルをベースにしています。
- 新しい `server_name.conf` ファイルは、サーバー名の変更後、サーバーを次に起動したときに作成されます。新しい構成ファイルは、`rwserver.template` ファイルのデフォルト値をベースにしています。変更前のサーバー名に付随する構成を保持する場合は、`server_name.conf` ファイルを新たなサーバー名 (`new_server_name.conf`) に変更してから、変更された名前のサーバーを起動します。

- いずれかのファイルを削除すると、削除したファイルはサーバーを次に起動したときに再び作成されます。新しいファイルは、`rwserver.template` ファイルのデフォルト値をベースにしています。

次の項では、次のディレクトリ（Windows および UNIX 共通）にある `rwserverconf.dtd` ファイルを使用して、これらのファイルで許可される構文と値を説明します。

```
ORACLE_HOME\reports\dtd\rwserverconf.dtd
```

この項では、次の項目について説明します。

- [Reports Server の構成要素 \(rwserverconf.dtd\)](#)
- [動的な環境切替え](#)
- [OracleAS Portal への接続](#)

3.2.1 Reports Server の構成要素 (rwserverconf.dtd)

次の例は、`rwserverconf.dtd` を使用して、Reports Server の各側面を構成する方法を示しています。

```
<!--
Copyright 2000, 2005 Oracle Corporation.
500 Oracle Parkway, Redwood Shores, CA 94065, U.S.A. All rights reserved.

This is the DTD defining the Reports Server Configuration file
(XML) format/syntax.
-->

<!ELEMENT server (compatible?,
                  cache?,
                  engine+,
                  security*,
                  oidconnection?,
                  destination*,
                  networkConfig?,
                  job+,
                  notification*,
                  log?,
                  jobStatusRepository?,
                  trace?,
                  connection?,
                  ORBPorts?,
                  queue?,
                  persistFile?,
                  jobRecovery?,
                  identifier?,
                  environment*,
                  pluginParam*)>
<!ATTLIST server
  version      CDATA      #IMPLIED>

<!ELEMENT cache (property*)>
<!-- class specifies full qualified java class name which implements
       oracle.reports.cache.Cache interface -->
<!ATTLIST cache
  class        CDATA      "oracle.reports.cache.RWCache">

<!ELEMENT engine (property*)>
<!-- class specifies full qualified java class name which starts engine -->
<!ATTLIST engine
  id           ID          #REQUIRED
  class        CDATA      #REQUIRED
  classPath    CDATA      #IMPLIED
```

```

initEngine      CDATA      "1"
maxEngine       CDATA      "1"
minEngine       CDATA      "0"
engLife         CDATA      "50"
maxIdle         CDATA      "30"
callbackTimeOut CDATA      "60000"
jvmOptions      CDATA      #IMPLIED
engineResponseTimeOut CDATA "0"
defaultEnvId    CDATA      #IMPLIED>

<!ELEMENT security (property*)>
<!-- class specifies full qualified java class name which implements
      oracle.reports.server.Security interface -->
<!ATTLIST security
  id          ID          #REQUIRED
  class       CDATA       #REQUIRED>

<!ELEMENT oidconnection EMPTY>
<!ATTLIST oidconnection
  init        CDATA      "10"
  increment   CDATA      "10"
  timeout     CDATA      "0">

<!ELEMENT destination (property*)>
<!-- class specifies full qualified java class name which subclass
      oracle.reports.server.Destination abstract class -->
<!ATTLIST destination
  destype     ID          #REQUIRED
  class       CDATA       #REQUIRED>

<!ELEMENT networkConfig EMPTY>
<!ATTLIST networkConfig
  file        CDATA      #REQUIRED>

<!ELEMENT job EMPTY>
<!ATTLIST job
  jobType     CDATA      "report"
  engineId    IDREF      #REQUIRED
  securityId  IDREF      #IMPLIED>

<!ELEMENT notification (property*)>
<!ATTLIST notification
  id          CDATA      "mailNotify"
  class       CDATA      #REQUIRED>

<!ELEMENT log EMPTY>
<!ATTLIST log
  option      (allJobs|succeededJobs|failedJobs|noJob) "noJob">

<!ELEMENT jobStatusRepository (property*)>
<!-- class specifies full qualified java class name which implements
      oracle.reports.server.JobRepository interface -->
<!ATTLIST jobStatusRepository
  class       CDATA      "oracle.reports.server.JobRepositoryDB">

<!ELEMENT queue EMPTY>
<!ATTLIST queue
  maxQueueSize CDATA      "1000">

<!ELEMENT connection (orbClient*, cluster?)*>
<!ATTLIST connection
  maxConnect   CDATA      "20"
  idleTimeOut  CDATA      "15">

```

```

<!ELEMENT ORBPorts EMPTY>
<!ATTLIST ORBPorts
  value          CDATA          #REQUIRED>

<!ELEMENT orbClient EMPTY>
<!ATTLIST orbClient
  id             ID             #REQUIRED
  publicKeyFile  CDATA          #REQUIRED>

<!ELEMENT cluster EMPTY>
<!ATTLIST cluster
  publicKeyFile  CDATA          #REQUIRED
  privateKeyFile CDATA          #REQUIRED>

<!ELEMENT persistFile EMPTY>
<!ATTLIST persistFile
  fileName       CDATA          #IMPLIED>

<!ELEMENT trace EMPTY>
<!ATTLIST trace
  traceFile      CDATA          #IMPLIED
  traceOpts      (trace_prf|trace_brk|trace_app|trace_pls|trace_sql|
                 trace_tms|trace_dst|trace_log|trace_err|trace_inf|
                 trace_dbg|trace_wm|trace_sta|trace_exc|trace_all|none)
                 "trace_all"
  traceMode      (trace_replace|trace_append) "trace_replace"
  traceModule    (all|server|engine) "all">

<!ELEMENT compatible EMPTY>
<!ATTLIST compatible
  version        (6i)          "6i">

<!ELEMENT jobRecovery EMPTY>
<!ATTLIST jobRecovery
  auxDatFiles    (yes|no)      "no">

<!ELEMENT identifier (#PCDATA)>
<!ATTLIST identifier
  confidential    (yes|no)      "yes"
  encrypted       (yes|no)      "no">

<!ELEMENT environment (envVariable*)>
<!ATTLIST environment
  id             ID             #REQUIRED>

<!ELEMENT envVariable EMPTY>
<!ATTLIST envVariable
  name           CDATA          #REQUIRED
  value          CDATA          #IMPLIED>

<!ELEMENT pluginParam (#PCDATA)>
<!ATTLIST pluginParam
  name          ID             #REQUIRED
  type          (text|file|url) "text">

<!ELEMENT property EMPTY>
<!ATTLIST property
  name          CDATA          #REQUIRED
  value         CDATA          #REQUIRED
  confidential   (yes|no)      "no"
  encrypted      (yes|no)      "no">

```

rwserverconf.dtd ファイルで提供される次の要素により、Reports Server が構成されます。

- server
- compatible
- cache
- engine
- security
- oidconnection
- destination
- networkConfig
- job
- notification
- log
- jobStatusRepository
- trace
- connection
- ORBPorts
- queue
- persistFile
- jobRecovery
- identifier
- pluginParam
- environment

これらの要素は、関連する属性およびサブ要素とともに、次の各項で説明します。

これらは XML 要素です。XML は大文字と小文字を区別することに注意してください。

また、これらの任意の要素を `server_name.conf` または `rebuilder.conf` 構成ファイルに追加する場合、`rwserverconf.dtd` で示された要素の順序に従うと、使用中の XML エディタでエラー・メッセージが表示されません。その順序に従わなくても構成ファイルは機能しますが、`rwserverconf.dtd` で指定された大文字と小文字の区別に従わない場合には機能しません。

3.2.1.1 server

例

```
<server>  
  One or more configuration specifications  
</server>
```

必須 / オプション

必須です。構成ファイルに 1 組の `server` 要素の開始タグと終了タグを指定できます。

説明

`server` 要素は、サーバー構成ファイルのコンテンツ領域を開始および終了します。ファイルの階層に関しては、その他すべての要素は `server` 要素に従属します。

3.2.1.2 compatible

注意： compatible 要素は、Oracle Reports 10g リリース 2 (10.1.2) では使用されません。compatible 要素を設定すると、Oracle Reports 6i クライアント・リクエストは、これまで同様 10g リリース 2 (10.1.2) の Reports Server に転送されます。ただし、この構成はサポートされておらず、オラクル社はこの構成が原因で生じる不具合の修正を予定していません。

デフォルトでは、Oracle Reports 10g リリース 2 (10.1.2) と Oracle Reports 10g (9.0.4) クライアント間、およびその逆の間では互換性が保証されています。以前のリリースのクライアントと Oracle Reports 10g リリース 2 (10.1.2) 間の互換性をまとめたマトリックス表は、Oracle Application Server Forms and Reports Services のインストール・ガイドを参照してください。

例

```
<compatible version="6i"/>
```

必須 / オプション

オプションです。サーバー構成ファイルに 1 つの compatible 要素を指定できます。

説明

compatible 要素を使用して、Oracle Reports 6i クライアント (RWCLI60.EXE、RWCGI60.EXE、RWQMU60.EXE、RWRQM60.EXE、RWRQV60.EXE、6i Forms) との下位互換性を確保できます。compatible を 6i に設定すると、Reports Server は、rwproxy という実行可能ファイルを使用して、6i クライアントからのリクエストをリスニングし、それを 10g サーバーに転送します。

compatible 要素の属性を、表 3-2 で説明します。

表 3-2 compatible 要素の属性

属性	有効値	説明
version	6i	version を 6i に設定すると、Oracle Reports で Oracle Reports 6i クライアントを実行できます。ただし、前述のとおり、Oracle Reports 10g リリース 2 (10.1.2) では Oracle Reports 6i との下位互換性が保証されておらず、オラクル社はこの構成が原因で生じる不具合の修正を予定していません。

compatible 要素を使用する場合、Reports Server の 6i バージョンと同様、tnsnames.ora ファイルにも Reports Server に対するエントリが必要です。インストーラによって、デフォルトの Reports Server (rep_machine_name) に対して、tnsnames.ora ファイルが構成されます。

注意： tnsnames.ora ファイルは、次のディレクトリにあります。

```
ORACLE_HOME/network/admin
```

次に例を示します。

```
testsvr.world = (ADDRESS=
  (PROTOCOL=tcp)
  (HOST=testhost.mydomain.com)
  (PORT=1950)
)
```

互換性をオフにすれば、この作業は必要ありません。互換性をオフにするには、compatible 要素を Reports Server 構成ファイルから削除します。

3.2.1.3 cache

例

```
<cache class="oracle.reports.cache.RWCache">
<property name="cacheSize" value="50"/>
<property name="cacheDir" value="D:\orawin\reports\server\cache"/>
</cache>
```

必須 / オプション

オプションです。サーバー構成ファイルに 1 つの cache 要素を指定できます。cache 要素が指定されていない場合は、デフォルトが使用されます (oracle.reports.cache.RWCache)。

説明

cache 要素を使用して、サーバーのキャッシュ実装を定義する Java クラスを指定できます。デフォルトの cache Java クラスを使用するか、OracleAS Reports Services Cache API を使用して独自の实装を定義できます。

注意: cache API の詳細は、Oracle Technology Network (OTN) の Oracle Reports 10g ページ (<http://www.oracle.com/technology/products/reports/index.html>) で「SDK」をクリックすると表示される『Reports Software Development Kit (RSDK)』を参照してください。

cache 要素の属性を、表 3-3 で説明します。

表 3-3 cache 要素の属性

属性	有効値	説明
class	「説明」欄を参照。	デフォルト: oracle.reports.cache.RWCache oracle.reports.cache.Cache インタフェースを実装する完全修飾された Java クラス

また、任意数のプロパティを cache 要素に入力できます。プロパティは、cache に登録する実装クラスによって認識および解釈される名前と値のペアです。たとえば、OracleAS Reports Services で提供されるデフォルトの cache Java クラスを使用する場合、構成エントリは次のようになります。

```
<cache class="oracle.reports.cache.RWCache">
<property name="cacheSize" value="50"/>
<property name="cacheDir" value="D:\orawin\reports\server\cache"/>
</cache>
```

この例では、cacheSize の単位はメガバイトで、キャッシュの位置を示す cacheDir は Windows プラットフォーム向けに指定されています。UNIX では、次のような UNIX 標準を使用します。

```
<property name="cacheDir" value="$ORACLE_HOME/reports/server/cache"/>
```

デフォルトの cache Java クラスは、次のプロパティも提供します。

- maxCacheFileNumber。キャッシュで許可されるファイルの最大数。次に例を示します。
<property name="maxCacheFileNumber" value="250"/>

- ignoreParameters。Reports Server がキャッシュ・キーを作成するときに無視するようにするレポート・パラメータをリストします (Reports Server では、キャッシュ・キーを使用して、受信したジョブ・リクエストがキャッシュ内の既存出力に一致するかどうかを判断されます)。

```
<property name="ignoreParameters" value="param1,param2"/>
```

3.2.1.4 engine

例

```
<engine id="rwEng" class="oracle.reports.engine.EngineImpl" initEngine="1"
  maxEngine="5" minEngine="1" engLife="50" maxIdle="15" callbackTimeOut="90000">
  <property name="sourceDir" value="D:¥orawin¥reports¥myReport"/>
  <property name="tempDir" value="D:¥orawin¥reports¥myTemp"/>
</engine>
```

必須 / オプション

必須です。サーバー構成ファイルに 1 つ以上の engine 要素を指定する必要があります。

説明

engine 要素は、完全修飾された Java クラスを識別します。この Java クラスは、エンジンを起動し、エンジンの動作制御を設定する多数の属性を提供します。OracleAS Reports Services で提供されるデフォルトのエンジンを使用するか、OracleAS Reports Services Engine API を使用して独自の実装を定義できます。カスタム・エンジンの例としては、データベース・イベントの発生時に特定のオペレーティング・システム・コマンドを実行するエンジンなどを開発できます。

注意： engine API の詳細は、Oracle Technology Network (OTN) の Oracle Reports 10g ページ (<http://www.oracle.com/technology/products/reports/index.html>) で「SDK」をクリックすると表示される『Reports Software Development Kit (RSDK)』を参照してください。

engine 要素の属性を、表 3-4 で説明します。

表 3-4 engine 要素の属性

属性	有効値	説明
id	文字列	特定の engine 要素を識別する、構成 XML ファイル内で一意のキーワード。テキスト文字列または数字のどちらでも使用できます。次に例を示します。 id="rwEng"
class	「説明」欄を参照。	デフォルト： oracle.reports.engine.EngineImpl oracle.reports.engine.Engine および oracle.reports.engine.EngineInterface の 2 つのインタフェースを実装する完全修飾された Java クラス。

表 3-4 engine 要素の属性 (続き)

属性	有効値	説明
classPath	文字列	<p>class 属性で指定される Java クラスへのディレクトリ・パス。ディレクトリは、サーバー・プラットフォームの表記規則に従って指定します。次に例を示します。</p> <p>Windows:</p> <pre>classPath="%ORACLE_HOME%\myEngine.jar"</pre> <p>UNIX:</p> <pre>classPath="\$ORACLE_HOME/myEngine.jar"</pre>
initEngine	数値	<p>デフォルト: 1</p> <p>初期化時に Reports Server で起動するエンジンの数。</p>
maxEngine	数値	<p>デフォルト: 1</p> <p>サーバーで実行できるこのタイプのエンジンの最大数。</p>
minEngine	数値	<p>デフォルト: 0</p> <p>サーバーで維持されるこのタイプのエンジンの最小数。</p>
engLife	数値	<p>デフォルト: 50</p> <p>エンジンが終了する前、および必要に応じて新しいエンジンが開始する前に、エンジンで実行できるジョブの数。この機能により、メモリー・リークを防止できます。</p>
maxIdle	数値	<p>デフォルト: 30</p> <p>エンジンの現在数が minEngine を超えている場合に、エンジンが停止する前に許可されるアイドル時間 (分単位)。</p> <p>たとえば minEngine が 0、maxIdle が 30 であり、1 つのエンジンが実行中の場合、このエンジンは 30 分間使用されないと停止します。同じ条件で minEngine が 1 の場合、30 分間アイドル状態であっても実行中のエンジンは停止しません。</p>
callbackTimeOut	数値	<p>デフォルト: 90000</p> <p>サーバーがエンジンを起動してからエンジンがサーバーに回答するまでに許可される待ち時間 (ミリ秒)。</p> <p>サーバーをホスティングするマシンが非常に高速である場合、小さい数を指定するとパフォーマンスが向上します。</p>
defaultEnvId	文字列	<p>(オプション属性) Reports Server がエンジンを起動するときのデフォルト環境を指定します。この属性は、サーバー構成ファイルの environment 要素に関連付けられた ID を取得します。</p> <p>defaultEnvId を指定した場合、Reports Server では、参照された environment 要素で指定される環境変数と、Reports Server の実行に使用されている環境変数を使用してエンジンを起動します。</p> <p>defaultEnvId を指定しない場合、Reports Server では、起動時に有効になっている環境設定を使用してエンジンを起動します。</p> <p>詳細は、第 3.2.2 項「動的な環境切替え」を参照してください。</p>

表 3-4 engine 要素の属性 (続き)

属性	有効値	説明
engineResponseTimeOut	数値	<p>デフォルト: null (タイムアウトなし)</p> <p>ユーザー環境でのレポート実行中にエンジンがジョブのステータス更新に要する最大時間 (分)。ジョブのステータスを更新する時間がなんらかの理由 (エンジンがハングする、または SQL 問合せが長時間ブロックされるなど) でこの最大時間を越えた場合、Reports Server はジョブを終了します。</p>
jvmOptions	文字列	<p>Java Virtual Machine (JVM) でエンジンを起動するときに Reports Server で使用する JVM オプション。たとえば、この属性を使用して JVM の開始ヒープ・サイズと最大ヒープ・サイズ、追加の CLASSPATH エントリなどを指定できます。</p> <p>この属性を指定しない場合、サーバー環境で実行されるエンジンは、REPORTS_JVM_OPTIONS 環境変数値によって指定される JVM オプションを使用します。詳細は、第 B.1.47 項「REPORTS_JVM_OPTIONS」を参照してください。</p>
keepConnection	YES NO	<p>デフォルト: YES</p> <p>keepConnection プロパティは、デフォルトのランタイム・エンジン実装 (oracle.reports.engine.EngineImpl) によって使用されます。</p> <p>YES: デフォルトのランタイム・エンジンは既存のデータベース接続情報を保持します。</p> <p>NO: デフォルトのランタイム・エンジンは既存のデータベース接続情報を破棄し、ジョブに指定されたユーザー ID を使用して再接続します。</p> <p>keepConnection プロパティは、rwbuilder または rwrund を使用して配布されたレポートには影響しません。</p> <p>以前のバージョン (9.0.2.x など) で使用された server_name.conf ファイルが現行の環境で実行されている場合、このプロパティは移行されます。</p>

プロパティ

また、オプションとして、複数のプロパティを engine 要素に入力できます。唯一の要件は、Oracle Reports Engine を実装する Java クラスによって識別される名前と値のペアであることです。

sourceDir および tempDir プロパティ: OracleAS Reports Services に付属するデフォルトの engine Java クラスを使用する場合、engine 構成エントリは次のようになります。

```
<engine id="rwEng" class="oracle.reports.engine.EngineImpl" initEngine="1"
  maxEngine="5" minEngine="1" engLife="50" maxIdle="15" callbackTimeOut="90000">
  <property name="sourceDir" value="D:\orawin\reports\myReport"/>
  <property name="tempDir" value="D:\orawin\reports\myTemp"/>
</engine>
```

この例で、sourceDir と tempDir は Windows 環境用の設定です (UNIX では、sourceDir="ORACLE_HOME/reports/myReport" と tempDir="ORACLE_HOME/reports/myTemp" になります)。

- sourceDir プロパティは、レポート定義ファイルに使用するデフォルト・ディレクトリを識別します。このプロパティは、REPORTS_PATH 環境変数で指定したパス情報を上書きします。

- tempDir プロパティは、OracleAS Reports Services でテンポラリ・ファイルに使用するテンポラリ・ディレクトリの名前と位置を識別します。この値がデフォルト・エンジンに対して指定されていない場合は、REPORTS_TMP 環境変数で指定されるテンポラリ・ディレクトリが使用されます。REPORTS_TMP も指定されていない場合は、オペレーティング・システムのデフォルト・テンポラリ・ディレクトリが使用されます。

この構成ではデフォルトの engine クラスが使用されるので、classPath 属性は指定されません。

diagnosis プロパティ: Oracle Reports 10g リリース 2 (10.1.2) では、エンジンのロギング用に diagnosis プロパティが導入されました。このプロパティを engine 構成要素に含めることで、レポートの特定の機能が正常に完了したかどうかを診断できます。診断ログには、レポートの実行時におけるエンジンの重要なチェックポイントまたはタスクに関する情報が記録されます。この情報は、エンジンがレスポンスを停止しジョブがハング状態になった場合に役立ちます。

エンジンの診断オプションを有効にするには、engine 構成要素を次のように指定します。

```
<engine id="rwEng" class="oracle.reports.engine.EngineImpl" initEngine="2"
  maxEngine="8" minEngine="1" engLife="1" maxIdle="3" callbackTimeOut="90000">
  <property name="diagnosis" value="yes">
</engine>
```

diagnosis プロパティを yes に設定すると、Reports Server とは別個の診断ファイルと、エンジン・トレース・ファイルが作成されます。診断ファイルは、トレース・ファイルと同じ場所に配置されます (第 20.1.2 項「レポート・トレース」を参照)。すべての rwEng エンジンで、インスタンスごとに (つまり、エンジンの存続期間ごとに) 2 つの診断ファイルが作成されます。次に例を示します。

```
rwEng-0-c-2004-08-31_05-24-54.dig
rwEng-0-j-2004-08-31_05-24-54.dig
```

各診断ログは新規ファイルで、ファイル名にタイムスタンプが含まれています。診断ファイルは、同じエンジンの異なるインスタンスに対して、追加使用されたり置換使用されることはありません。

エンジンの診断オプションでは、レポートのトレースよりも詳細な情報が提供されます。通常、トレースは、レポートの実行をデバッグする目的で、現在フォーマット中のファイルや現在実行中のレポート・トリガーなどに関する情報を取得する際に使用されます。

3.2.1.5 security

例

```
<security id="rwSec" class="oracle.reports.server.RWSecurity">
  <!--property name="securityuserid"
    value="portal_db_username/portal_password@portal_db_connection"
    confidential="yes" encrypted="no"/-->
  <property name="oidEntity" value="oidentity_name"/>
</security>
```

注意: Oracle Reports 10g (9.0.4) より前のリリースでは、security 要素の指定が異なっていました。Oracle Reports 10g (9.0.4) および 10g リリース 2 (10.1.2) でもこの従来のプロパティの指定 (securityuserid) がありますが、コメント・アウトされています。securityuserid を使用する必要があるのは、デフォルトのインスタンス以外の OracleAS Portal インスタンスに接続する場合のみです。デフォルトのインスタンスは、Oracle Application Server インフラストラクチャとともにインストールされています。

必須 / オプション

オプションです。security 要素を構成ファイルに入力しないと、Reports Server のセキュリティが確保されません。任意数の security 要素を構成ファイルに入力できます。

説明

security 要素は、サーバー・アクセスを制御する完全修飾された Java クラスを識別します。OracleAS Reports Services で提供されるデフォルトの security クラスを使用するか（このクラスは、Oracle Application Server に含まれる OracleAS Portal 経由で提供されるセキュリティ機能に依存します）、Reports Server Security API を使用して独自の実装を定義することができます。

注意： security API の詳細は、Oracle Technology Network (OTN) の Oracle Reports 10g ページ (<http://www.oracle.com/technology/products/reports/index.html>) で「SDK」をクリックすると表示される『Reports Software Development Kit (RSDK)』を参照してください。

security 要素の属性を、表 3-5 で説明します。

表 3-5 security 要素の属性

属性	有効値	説明
id	文字列	特定の security 要素を識別する、構成 XML ファイル内で一意のキーワード。テキスト文字列または数字のどちらでも使用できます。次に例を示します。 <code>id="rwSec"</code>
class	「説明」欄を参照。	デフォルト： <code>oracle.reports.server.RWSecurity</code> Reports Server Security Java インタフェース (<code>oracle.reports.server.Security</code>) を実装する完全修飾された Java クラス。デフォルトは、Oracle Application Server に含まれる OracleAS Portal 経由で提供されるセキュリティ機能に依存します。

また、複数のプロパティを security 要素に関連付けることもできます。唯一の要件は、Reports Server のセキュリティを実装する Java クラスによって識別される名前と値のペアであることです。たとえば、OracleAS Reports Services で提供されるデフォルトの security Java クラスを使用する場合、security の構成エントリは前述の例のようになります。

oidEntity の値は、インストール時にインストーラによって設定されます。このエンティティを使用して Reports Server を Oracle Internet Directory に接続します。Oracle Application Server コンポーネントはすべて、Oracle Internet Directory に接続できますが、このディレクトリでの権限はコンポーネントごとに異なる場合があります。したがって、各コンポーネントは、Oracle Internet Directory に接続したときに、個々のエンティティ名によって Oracle Internet Directory に識別される必要があります。OracleAS Reports Services エンティティの形式は次のとおりです。

```
reportsApp_hostname_GUID
```

次に例を示します。

```
reportsApp_testhost.mydomain.com_BBEFDCCDAC2343600E0340800020C7BBCC
```

コメント・アウトされたプロパティ (`securityUserid`) は、`security` を指定する従来の方法を示しています。`securityUserid` では、Reports Server で OracleAS Portal のセキュリティ機能にアクセスするために必要な接続情報を提供します。`confidential` および `encrypted` のプロパティ属性は、プロパティ内の情報の暗号化に使用されます。`confidential="yes"` および `encrypted="no"` の属性を入力すると、サーバーの再起動後にプロパティ値が自動的に暗号化されます。構成ファイルを次にオープンする際、パスワード情報が暗号化され、`encrypted` が `yes` に設定されます。構成ファイルに入力したパスワードを忘れた場合は、プロパティを削除して新しい値を再入力し、`encrypted` を `no` に設定します。

注意： `securityUserid` データベース接続文字列については、`thin` (`testhost.mydomain.com:1521:iasdb`) と Oracle Call Interface (`scott/tiger@ordb`) の両方の JDBC フォーマットがサポートされます。

3.2.1.6 oidconnection

例

```
<oidconnection init="10" increment="10" timeout="600"/>
```

必須/オプション

オプションです。

説明

`oidconnection` 要素は、Reports Server の Oracle Internet Directory 接続プーリング・パラメータを指定します。本番環境では、このパラメータを使用して、Reports Server の Oracle Internet Directory 接続プーリングを、次のように詳細に管理できます。

- Reports Server の初期化時にプールで開いておく接続の数
- 使用可能な接続を使用し尽くした場合に、新しいリクエストを受信したときにプールに追加する新しい接続の数
- リソース使用率を減少させるために、開いているアイドル状態の Oracle Internet Directory 接続を閉じるまでのタイムアウト

`oidconnection` 要素の属性を、表 3-6 で説明します。

表 3-6 `oidconnection` 要素の属性

属性	有効値	説明
<code>init</code>	数値	デフォルト: 10 Reports Server の初期化時に作成される Oracle Internet Directory 接続の初期数
<code>increment</code>	数値	デフォルト: 10 すべての接続が使用中になったときに増分される接続数
<code>timeout</code>	数値	デフォルト: 0 (タイムアウトの指定なし) 接続が閉じられるまでの、アイドル状態の秒数

注意： これらの属性に、デフォルト値より非常に大きな値または小さな値を設定すると、OracleAS Reports Services のパフォーマンスに影響する場合があります。一般的な本番環境では、デフォルト値をお勧めします。

Reports Servlet では、`rwervlet.properties` ファイルの `OID_INIT`、`OID_INCREMENT`、`OID_TIMEOUT` の各プロパティを使用して、Oracle Internet Directory 接続プーリング・パラメータを指定できます（詳細は、[第 3.4.12 項「Oracle Internet Directory 接続プーリング・パラメータの指定」](#)を参照）。

3.2.1.7 destination

例

```
<destination destype="oraclePortal" class="oracle.reports.server.DesOraclePortal">
<property name="portalUserId"
    value="portal_db_username/portal_password@portal_db_connection"
    confidential="yes" encrypted="no"/>
</destination>
```

必須 / オプション

オプションです。`destination` 要素をサーバー構成ファイルに入力しないと、提供されている宛先クラスが使用されます（プリンタ、電子メール、ファイル、キャッシュおよび OracleAS Portal。ポータルへのログインに使用されるユーザー ID とパスワードを指定するには、サーバー構成ファイルにエントリが必要なので、ポータルは例外です）。任意数の `destination` 要素をサーバー構成ファイルに入力できます。

説明

`destination` 要素を使用して、サーバーとともに宛先タイプを登録できます。デフォルトの宛先は登録不要です。OracleAS Portal、FTP または WebDAV の宛先は、次の説明にあるように、構成が必要な場合があります。

- OracleAS Portal: この宛先のエントリはデフォルトでサーバー構成ファイルに作成されますが、コメント・アウトされています。この宛先の使用を開始するには、`destination` エントリのコメントを外し、適切なプロパティ値（`portalUserId` プロパティの値など）を指定する必要があります。
- FTP および WebDAV: これらの宛先のエントリはデフォルトでサーバー構成ファイルに作成され、コメント・アウトされていません。つまり、これらは構成済で、デフォルトで使用できます。プロキシを必要とする FTP または WebDAV サーバーに出力を送信する必要がある場合は、デフォルトの場所（`ORACLE_HOME\reports\conf`）にある `proxyinfo.xml` ファイルを編集し、`destination` 要素の `proxy` プロパティのコメントを外し、`value` に `proxyinfo.xml` ファイルへの完全なパスを指定します。次に例を示します。

```
<destination destype="ftp"
class="oracle.reports.plugin.destination.ftp.DesFTP"> <property
name="proxy" value="D:\oracle\reports\conf\proxyinfo.xml"/>
</destination>
```

OracleAS Reports Services Destinations API で作成した新しい宛先タイプは登録する必要があります。

注意： `destination` API の詳細は、Oracle Technology Network (OTN) の Oracle Reports 10g ページ (<http://www.oracle.com/technology/products/reports/index.html>) で「SDK」をクリックすると表示される『Reports Software Development Kit (RSDK)』を参照してください。

宛先の構成の詳細は、[第 8 章「OracleAS Reports Services の宛先の構成」](#)を参照してください。

`destination` 要素の属性を、[表 3-7](#) で説明します。

表 3-7 destination 要素の属性

属性	有効値	説明
destype	文字列	宛先タイプを識別します。次に例を示します。 destype="printer"
class	「説明」欄を参照。	Reports Server Destination Java クラス (oracle.reports.server.Destination) のサブクラスである完全修飾された Java クラス。使用可能な値は、次のとおりです。 oracle.reports.server.DesMail oracle.reports.server.DesFile oracle.reports.server.DesPrinter oracle.reports.server.DesOraclePortal

また、複数のプロパティを destination 要素に入力することもできます。唯一の要件は、Reports Server Destination Java クラスのサブクラスである Java クラスによって識別される名前と値のペアであることです。次に例を示します。

```
<destination destype="oraclePortal" class="oracle.reports.server.DesOraclePortal">
<property name="portalUserid"
value="portal_db_username/portal_password@portal_db_connection"
confidential="yes" encrypted="no"/>
</destination>
```

この例では、接続情報が提供され、Reports Server で OracleAS Portal にアクセスできるようになります。confidential および encrypted の属性が含まれているので、Reports Server の次回起動時に portalUserid 値が自動的に暗号化されます。

注意： portalUserid データベース接続文字列については、thin (testhost.mydomain.com:1521:iasdb) と Oracle Call Interface (scott/tiger@ordb) の両方の JDBC フォーマットがサポートされます。

destination を実装する際に追加情報が必要な場合は、その情報を pluginParam 要素で指定します。

3.2.1.8 networkConfig

例

```
<networkConfig file="net3.conf"></networkConfig>
```

ここで、

net3.conf は、ORACLE_HOME/reports/conf ディレクトリに配置されたカスタムの Reports ネットワーク構成ファイルです。Oracle Reports は、このファイルに指定された Reports Server 検出メカニズムを使用します。

必須 / オプション

オプションです。デフォルトでは、Oracle Reports は ORACLE_HOME¥reports¥conf ディレクトリにある rwnetwork.conf ファイルを使用して、Reports Server 検出メカニズムを決めます。カスタムのネットワーク構成ファイルを指定する場合は、この要素のコメントを外します。

注意： Reports Server 用のネットワーク構成ファイルを変更する場合、そのサーバーに接続するすべてのクライアントで同じ構成設定を使用する必要があります。

説明

この要素は、Reports Server の検出に使用されるネットワーク構成ファイルを指定します。詳細は、第 3.3 項「Reports Server 検出メカニズムの構成」を参照してください。

namingservice が有効化されている場合、Reports Server は COS ネーミング・サービスに自動登録されます。Oracle Reports クライアントはそのネーミング・サービスに問い合わせ、Reports Server 参照を取得します。namingservice 要素の属性は、第 3.3.1.3 項「[namingService](#)」で説明されています。

表 3-8 networkConfig 要素の属性

属性	有効値	説明
ファイル	ネットワーク構成ファイルの名前	ネットワーク構成ファイルは、 <code>ORACLE_HOME\reports\conf</code> ディレクトリにある必要があります。

3.2.1.9 job

例

```
<job jobType="report" engineId="rwEng" securityId="rwSec"/>
```

必須 / オプション

必須です。1 つ以上の job 要素を指定する必要があります。

説明

job 要素は、[engine](#) および [security](#) の要素と連携して動作します。job を使用して、ジョブ・タイプを識別し、そのタイプのジョブで使用するエンジンおよびセキュリティの実装を指定します。たとえば、ユーザーが開発した、データベース・イベントの発生時に特定のオペレーティング・システム・コマンドを実行するエンジンなどです。OracleAS Reports Services のイベントドリブンの公開 API を使用して、イベントを特定のジョブ・タイプとして識別します。そのイベントが発生したとき、ジョブ・タイプ情報が Reports Server に渡され、Reports Server が構成ファイルの job 要素でそのジョブ・タイプを検索し、要素の属性で提供される指示に従います。この指示は、そのタイプのジョブに指定されたエンジン（および場合によってはセキュリティの実装）に対して指定されています。

job 要素の属性を、[表 3-9](#) で説明します。

表 3-9 destination 要素の属性

属性	有効値	説明
jobType	文字列	<p>デフォルト: report</p> <p>サーバーで処理されるジョブのタイプを記述します。Reports Server に備わっているエンジンがそのジョブ・タイプを処理できれば、任意のジョブ・タイプを入力できます。</p> <p>Oracle Reports で提供されるデータベース認証機能は、jobType=report の場合にのみ使用できます。これは、OracleAS Reports Services に提供されるデフォルトのエンジン (rwEng) のジョブ・タイプです。データベース認証機能は、jobType が別の値（たとえば、独自で開発したカスタム・エンジンなど）を指定している場合は実装されません。これは、カスタム・エンジンでは、接続文字列の形式が異なっている必要があるためです。Oracle Reports のデータベース認証機能では、接続文字列はデフォルトのエンジンで使用される Oracle Reports の形式 user/password@dbname に限定されます。</p>

表 3-9 destination 要素の属性 (続き)

属性	有効値	説明
engineId	ID 参照	このジョブ・タイプを処理するエンジンに対して入力された ID を参照します。利用可能な ID は、サーバー構成ファイルで engine 要素の id 属性を使用して指定されています。id は、特定のエンジンを識別する、構成 XML ファイル内で一意のキーワードです。
securityId	ID 参照	このジョブ・タイプに適用されるセキュリティ・メカニズムに対して入力された ID を参照します。利用可能な ID は、サーバー構成ファイルの security 要素で指定されています。

3.2.1.10 notification

例

```
<notification id="tellMe02" class="oracle.reports.server.MailNotify"/>
```

必須 / オプション

オプションです。notification 要素を構成ファイルに入力しないと、通知機能は使用できません。任意数の notification 要素を構成ファイルに入力できます。

説明

notification 要素を使用して、ジョブが成功または失敗したときに送られる通知のタイプを定義する Java クラスを指定します。電子メールを介して通知を送信するデフォルトの notification クラスを使用するか、Oracle Reports Notification API を使用して独自のクラスを定義できます。

注意: notification API の詳細は、Oracle Technology Network (OTN) の Oracle Reports 10g ページ (<http://www.oracle.com/technology/products/reports/index.html>) で「SDK」をクリックすると表示される『Reports Software Development Kit (RSDK)』を参照してください。

notification 要素の属性を、表 3-10 で説明します。

表 3-10 notification 要素の属性

属性	有効値	説明
id	文字列	デフォルト: mailNotify 特定の notification 要素を識別する、構成 XML ファイル内で一意のキーワード。テキスト文字列または数字のどちらでも使用できます。次に例を示します。 id="tellMe01"
class	「説明」欄を参照。	デフォルト: oracle.reports.server.MailNotify Reports Server Notification Java クラス (oracle.reports.server.Notification) を実装する完全修飾された Java クラス。

デフォルトの電子メールによる通知実装を使用する場合は、**pluginParam** 要素を使用して、メールの送信に使用する送信用 SMTP メール・サーバーを指定します。ランタイム・コマンド `notifysuccess` および `notifyfailure` を使用して、`notification` の送信先となる電子メール・アドレスを指定します（詳細は、付録 A 「コマンドライン・キーワード」を参照してください）。たとえば、これらのコマンドを次のようにランタイム URL に挿入できます。

```
notifysuccess=recipient's e-mail address&notifyfailure=recipient's e-mail address
```

デフォルトの電子メール実装では、`notification` のタイプごとに 1 つのアドレスのみを指定できます。一方または両方の `notification` タイプを指定できます。`notification` は、同じアドレスまたはそれぞれ異なるアドレスに送信できます。

サーバー構成ファイルの `notification` 要素は、次のようになります。

```
<notification id="mailNotify" class="oracle.reports.server.MailNotify">
<property name="succNoteFile" value="succnote.txt"/>
<property name="failNoteFile" value="failnote.txt"/>
</notification/>
```

メール・サーバーの中には送信者のドメイン名を確認できるものもあります。このドメイン名の確認によって通知が失敗した場合は、`notification` 要素の一部に次のプロパティを追加する必要があります。

```
<property name="sender" value="valid email address"/>
```

デフォルトの通知実装では、成功または失敗のテキスト・ファイルがデフォルト位置 (`ORACLE_HOME\reports\templates`) にあるかぎり、パスの指定は不要です。デフォルト位置にない場合は、サーバーをホスティングするプラットフォームの要件に従ってディレクトリ・パスとファイル名を入力します。

3.2.1.11 log

例

```
<log option="allJobs"/>
```

必須 / オプション

オプションです。サーバー構成ファイルに 1 つの `log` 要素を指定できます。

説明

`log` 要素は、下位互換性を確保するために用意されています。この要素は、Reports ログ・ファイルの生成および書込みを行います。ログ・ファイルは自動的に生成され、次のパスに格納されます（このパスは、Windows と UNIX で共通です）。

```
ORACLE_HOME\reports\server_name\server.log
```

`log` 要素の属性を、表 3-11 で説明します。

表 3-11 log 要素の属性

属性	有効値	説明
option	allJobs succeededJobs failedJobs noJob	<p>デフォルト: noJob</p> <p>ログの記録が必要なジョブのタイプを記述します。これはデフォルトでログを記録するサーバー・アクティビティに対する追加項目です。次のオプションから選択します。</p> <ul style="list-style-type: none"> ■ allJobs: すべてのジョブのログを記録します。 ■ succeededJobs: 正常に実行されたジョブのログのみを記録します。 ■ failedJobs: 失敗したジョブのログのみを記録します。 ■ noJob: ジョブのログを記録しません。

3.2.1.12 jobStatusRepository

例

```
<jobStatusRepository class="oracle.reports.server.JobRepositoryDB">
<property name="repositoryConn" value="scott/tiger@orcl"
  confidential="yes" encrypted="no"/>
</jobStatusRepository>
```

必須 / オプション

オプションです。サーバー構成ファイルに1つの jobStatusRepository 要素を指定できます。

説明

jobStatusRepository 要素は、ジョブ・ステータス・リポジトリを実装する Java クラスを指定します。この要素は、persistFile 要素に加えてジョブ・ステータス情報を格納するためのもう1つの手段を提供します。

persistFile はバイナリ・ファイルであるため、アプリケーション内でジョブ・ステータス情報を公開する際には使用できません。jobStatusRepository は、ステータス情報を格納する追加手段を提供することで、アプリケーションのステータス情報を保存する手段を提供します。

デフォルトのクラス oracle.reports.server.JobRepositoryDB では、情報はデータベースに格納されます。Oracle Reports API を使用して、目的の位置に情報を格納するよう、Reports Server Job Repository インタフェース (oracle.reports.server.JobRepository) を独自に実装できます。

jobStatusRepository 要素の属性を、表 3-12 で説明します。

表 3-12 jobStatusRepository 要素の属性

属性	有効値	説明
class	「説明」欄を参照。	<p>デフォルト:</p> <p>oracle.reports.server.JobRepositoryDB</p> <p>Reports Server Job Repository Java クラス (oracle.reports.server.JobRepository) を実装する完全修飾された Java クラス</p>

jobStatusRepository 要素では、任意数のプロパティを使用して、オプションをリポジトリに渡すことができます。唯一の要件は、ユーザーが指定した名前と値のペアを、サーバー構成ファイルで指定するクラスが認識することです。

サーバー構成ファイルの `jobStatusRepository` 要素は、次のようになります。

```
<jobStatusRepository class="oracle.reports.server.JobRepositoryDB">
<property name="repositoryConn" value="scott/tiger@ORCL"
  confidential="yes" encrypted="no"/>
</jobStatusRepository>
```

この例で、`repositoryConn` プロパティの値は、リポジトリが格納されているデータベースにアクセスするためのログイン情報です。`confidential` および `encrypted` の属性が使用されているので、Reports Server の再起動時にログイン情報が暗号化されます。

注意： `repositoryConn` データベース接続文字列については、`thin` (`testhost.mydomain.com:1521:iasdb`) と `Oracle Call Interface` (`scott/tiger@orcl`) の両方の JDBC フォーマットがサポートされます。

3.2.1.13 trace

例

```
<trace traceFile="neptune.trc" traceOpts="trace_prf|trace_dbg|trace_wm"
  traceMode="trace_append" traceModule="server"/>
```

必須 / オプション

オプションです。サーバー構成ファイルに1つの `trace` 要素を指定できます。

説明

`trace` 要素を使用して、レポート実行のトレースに使用するファイルを作成し、トレースするオブジェクトとアクティビティを指定します。`trace` 要素は、サーバーとエンジンのみのトレースをコントロールします。

注意： レポートの実行をトレースする各種方法については、[第 20.1.2 項「レポート・トレース」](#)を参照してください。

Reports Server では、印刷ジョブの送信に `rwlpr` 実行可能ファイルが使用されます。Windows の `rwlpr` ロギングでは、`traceModule=all` または `traceModule=server` を使用して Reports Server のトレースを有効化すると、`destype=printer` に対する印刷診断ログ (`server_name-rwlpr-jobid.log`) がログ・ディレクトリ (`ORACLE_HOME\reports\logs\server_name`) に作成されます。このログ・ファイルには、スプーラの問題など、印刷に関連した問題の診断に使用されるメッセージについての情報が含まれます。

デフォルトでは、トレース・ファイルは `ORACLE_HOME\reports\logs` に生成されます。ジョブのトレースが指定されている場合は、トレース・ファイルは `ORACLE_HOME\reports\logs\server_name\job_id` に生成されます。

`trace` 要素の属性を、[表 3-13](#) で説明します。

表 3-13 trace 要素の属性

属性	有効値	説明
traceFile	*.trc	<p>デフォルト: <code>server_name.trc</code></p> <p>トレース・ファイルの名前。パスの指定がない場合、トレース・ファイルは、Windows と UNIX の両方で次のディレクトリに置かれます。</p> <p><code>ORACLE_HOME/reports/logs/</code></p> <p>ジョブのトレースが指定されている場合は、次のディレクトリに置かれます。</p> <p><code>ORACLE_HOME/reports/logs/server_name/job_id</code></p>
traceOpts	表 3-14 を参照。	<p>デフォルト: <code>trace_all</code></p> <p>トレースするアクティビティを定義します。1 つ以上の <code>traceOpts</code> 値を指定できます。次に例を示します。</p> <pre><traceOpts="trace_prf trace_brk"></pre> <p>値は縦線 () で区切ります。</p>
traceMode	trace_replace trace_append	<p>デフォルト: <code>trace_replace</code></p> <p>新しいトレース情報で既存のトレース・ファイルを上書きするか (<code>trace_replace</code>)、既存のトレース情報を残したままトレースの末尾に追加するか (<code>trace_append</code>) を決定します。</p>
traceModule	all server engine	<p>デフォルトでは、トレースは、サーバー・イベントとエンジン・イベントの両方を追跡します。エンジン・イベントをトレースすると、パフォーマンス上の問題が生じる場合があります。 <code>traceModule</code> を使用して、サーバー・イベントのみを追跡するか (<code>server</code>)、エンジン・イベントのみを追跡するか (<code>engine</code>)、両方を追跡するか (<code>all</code>) を指定できます。次に例を示します。</p> <pre><trace traceOpts="trace_all" traceModule="server"/></pre> <p><code>traceModule</code> を指定しない場合は、サーバーとエンジンのトレースが両方ともオンになります。</p> <p>(Windows のみ) Reports Server のトレースを有効化すると、<code>destype=printer</code> に対する印刷診断ログ (<code>server_name-rwlpr-jobid.log</code>) がログ・ディレクトリ (<code>ORACLE_HOME\reports\logs\server_name</code>) に作成されます。このログ・ファイルには、スプーラの問題など、印刷に関連した問題の診断に使用されるメッセージについての情報が含まれます。</p>

`traceOpts` 要素の属性を、表 3-14 で説明します。

表 3-14 traceOpts 属性の有効値

値	説明
trace_prf	サーバーおよびエンジン・プロファイルのログを記録します。
trace_brk	デバッグ・ブレークポイントを示します。
trace_app	すべてのレポート・オブジェクトに関する情報を記録します。
trace_pls	すべての PL/SQL オブジェクトに関する情報を記録します。
trace_sql	すべての SQL に関する情報を記録します。
trace_tms	トレース・ファイルの各エントリのタイムスタンプを入力します。

表 3-14 traceOpts 属性の有効値 (続き)

値	説明
trace_dst	配布リストを示します。 どのレポート・セクションがどの宛先に送信されたかを判断できます。
trace_log	トレース・ファイルのログ情報を複製します。 サーバー構成ファイルで trace 要素のほかに log 要素も指定した場合、ログ・ファイルに出力された情報がトレース・ファイルにも送信されます。
trace_err	サーバーのエラー・メッセージを示します。
trace_inf	他のオプションでトレース・ファイルに記録されない情報をすべて記録します。
trace_dbg	デバッグ情報を記録します。
trace_wrn	サーバーの警告メッセージを示します。
trace_sta	初期化中、準備完了、実行中および停止など、サーバーおよびエンジンの状態に関する情報を提供します。
trace_all	サーバーおよびエンジンに関するすべての情報をトレース・ファイルに記録します。
trace_exc	すべての例外を示します。
none	トレースを無効化し、traceFile で指定されたディレクトリにログ・ファイルのみを生成します (トレース・ファイルは生成されません)。

3.2.1.14 connection

例

```
<connection maxConnect="50" idleTimeOut="20">
  <orbClient id="RWClient" publicKeyFile="clientpub.key"/>
</connection>
```

必須 / オプション

オプションです。サーバー構成ファイルで connection 要素を指定しない場合は、デフォルト値が使用されます (表 3-15 を参照)。サーバー構成ファイルに 1 つの connection 要素を指定できます。

説明

connection 要素は、サーバーとサーバーに接続されているクライアント間の接続ルールを定義します。

connection 要素の属性を、表 3-15 で説明します。

表 3-15 connection 要素の属性

属性	有効値	説明
maxConnect	数値	デフォルト: 20 サーバーが同時にサービスを提供できるリクエストの最大数。リクエスト数が maxConnect 値を超えると、Java 例外が返されます。
idleTimeOut	数値	デフォルト: 15 接続のアイドル状態が許可される時間 (分)。

connection には、orbClient というサブ要素があります。

注意： Oracle Reports 10g リリース 2 (10.1.2) では、Reports Server のクラスタリングは使用されず (詳細は、『A Guide to Changed Functionality Between Oracle Reports 6i and 10g』を参照)、cluster サブ要素は無効です。第 3.4.20 項「Reports Server クラスタ名を含む OracleAS Forms Services アプリケーションの移行」も参照してください。

高可用性のアプリケーション・サーバーレベルのテクニックについては、第 1.5.4 項「高可用性環境の選択」を参照してください。

orbClient は、クライアントが Reports Server との接続に使用する公開鍵ファイルの名前を提供します。Reports Server では公開鍵を使用して、クライアントが Reports Server に接続しようとするときに送信する署名を検証します。この公開鍵を使用して署名が検証できるクライアントのみが Reports Server によって受け付けられます。任意数の orbClient サブ要素をサーバー構成ファイルに入力できます。

orbClient サブ要素の属性を、表 3-16 で説明します。

表 3-16 orbClient サブ要素の属性

属性	有効値	説明
id	文字列	デフォルト: RWClient 公開鍵および秘密鍵を使用するレポート・クライアントを識別します。
publicKeyFile	filename.key	デフォルト: clientpub.key クライアントが Reports Server との接続に使用する公開鍵ファイルの名前を識別します。Reports Server では公開鍵を使用して、クライアントが Reports Server に接続しようとするときに送信する署名を検証します。この公開鍵を使用して署名が検証できるクライアントのみが Reports Server によって受け付けられます。デフォルト・ファイルは、rwrun.jar ファイルに格納されます。

OracleAS Reports Services によってデフォルトで提供されるクライアントの公開鍵ファイルおよび秘密鍵ファイルは、clientpub.key および clientpri.key です。これらの鍵ファイルは、OracleAS Reports Services のすべてのコンポーネントで使用されます。公開鍵ファイルおよび秘密鍵ファイルを再生成し、デフォルトの鍵ペアを置換できます。この作業を行うには、コマンド・プロンプトで次のように入力します。

Windows の場合：

```
rwgenkey.bat path_and_client_public_key_file_name path_and_client_private_key_file_name
```

UNIX の場合：

```
rwgenkey.sh path_and_client_public_key_file_name path_and_client_private_key_file_name
```

これらの鍵を再生成する場合、公開鍵ファイルの位置を publicKeyFile 属性で指定して、ORACLE_HOME¥jlib¥zrclient.jar にある秘密鍵ファイルを置換できます。このためには、jar 形式のファイルを元の形式に戻し、再生成した秘密鍵を入力した後、ファイルを再び jar 形式に変換します。

3.2.1.15 ORBPorts

例

ポート範囲を指定する場合：

```
<ORBPorts value="17000-17010"/>
```

特定のポートを指定する場合：

```
<ORBPorts value="17000,17010,17020,17030,17040"/>
```

必須 / オプション

オプションです。デフォルトでは、CORBA オブジェクトは、通信に使用可能な任意のポートを使用します。Reports Server では CORBA 通信が使用されるため、使用可能な任意の空きポートが通信に使用されます。任意のポートではなく事前定義済みのポートを Reports Server で使用する必要がある場合は、サーバー構成ファイルに ORBPorts 要素を含めます。

説明

ORBPorts 要素では、CORBA 通信に使用するポートを、ポート範囲または特定のポートのどちらかで指定します。ORBPorts を指定すると、ORB 内部通信用に指定されたリストから 1 つのポートが選択されます。Reports Server と各エンジンのそれぞれに、1 つのポートが必要です。

注意： ORBPorts 要素は、レポート関連のリクエストの実行に必要な固有のポートを、Reports Server とエンジンに割り当てる目的で使用されません。このポートを、Oracle Enterprise Manager 10g の「ポート」リンクから表示されるポートと混同しないでください。このリンクから表示されるポートは、Reports Server 検出メカニズムと Oracle Reports ブリッジ・コンポーネント用に予約されたポートで、rwnetwork.conf と rwbridge_bridgename.conf にそれぞれ設定されたものです（第 3.3.1 項「ネットワークの構成要素 (rwnetworkconf.dtd)」および第 3.3.2 項「ブリッジの構成要素 (bridgeconf.dtd)」を参照）。

個々のエンジンに対して、ポート番号を指定することはできません。各エンジンは、リスト内の次のポート番号を取得します。engine 要素の maxengine 属性が rwEng に対して 5 に設定されているときに、URLeng も有効化されている場合は、ORBPorts 要素に最低 7 つのポートを指定する必要があります (Reports Server 用に 1 つ、rwEng 用に 5 つ、rwURLeng 用に 1 つ必要なため)。

ORBPorts 要素の属性を、表 3-17 で説明します。

表 3-17 ORBPorts 要素の属性

属性	有効値	説明
value	ポート範囲 または カンマ区切りの番号	Reports Server とエンジンによる CORBA 通信に使用可能なポート範囲

注意： ORBPorts 要素は、Reports Server が実行されているサーバーで TCP ポート・フィルタリングが有効化されている場合にのみ指定します。ポート・フィルタリングが有効化されている場合は、Reports Server 用のポートをいくつか開いてから、ORBPorts を使用して、Reports Server およびエンジンの通信用ポートをサーバー構成ファイルに指定します。指定されたポートがすべて使用不可の場合、Reports Server またはエンジンは起動に失敗し、エラーが表示されます。

3.2.1.16 queue

例

```
<queue maxQueueSize="1000"/>
```

必須 / オプション

オプションです。サーバー構成ファイルに 1 つの queue 要素を指定できます。queue 要素を指定しない場合、デフォルトの 1000 が使用されます。

説明

queue 要素は、レポートの各キューで保持できるジョブの最大数を指定します。OracleAS Reports Services には、次の 3 つのキュー・コンポーネントがあります。

- スケジュールされたジョブのキュー
- 実行中のジョブのキュー
- 完了したジョブのキュー

queue 要素は、これらの各コンポーネントに許可される値を指定します。

この要素は完了したジョブ・キューにのみ適用されます。ジョブ数が指定された最大値を超えると、完了したジョブ・キューから最も古いジョブが自動的に削除されます。スケジュールされたジョブ・キューと実行中のジョブ・キューは変わりません。デフォルトでは、レポート・サーバーのキュー・サイズは 1000 です。

キュー・サイズを 3000 よりも大きくし、Reports Queue Manager (rwrqm.exe) でキューを監視すると、Queue Manager が失敗する場合があります。キュー・サイズを 3000 以上に設定する必要がある場合は、Oracle Enterprise Manager 10g または Reports Servlet (rwservlet) を使用して、Reports Server のジョブ・キューを管理および監視します。

注意： 詳細は、Reports Queue Manager のオンライン・ヘルプを参照してください。

queue 要素の属性を、表 3-18 で説明します。

表 3-18 queue 要素の属性

属性	有効値	説明
maxQueueSize	数値	デフォルト: 1000 レポートのジョブ・キューに保持できるジョブの最大数

3.2.1.17 persistFile

例

```
<persistFile fileName="neptune.dat"/>
```

必須 / オプション

オプションです。ファイルを指定しない場合、サーバーは `server_name.dat` というデフォルト名で独自のファイルを作成します。1 つの persistFile 要素を指定できます。

説明

persistFile 要素は、すべてのジョブ・ステータスを記録するファイルを識別します。Reports Server は、このファイルを使用してサーバーを停止前のステータスに復元します。

このファイルの名前 persistFile は、サーバーが停止後に再起動しても元の状態のままであることを、つまり永続性があることに由来します。

サーバー永続ファイルは、サーバーの初回起動時、または現在のサーバー永続ファイルを削除または名前変更した後の初回起動時に自動的に作成されます。このファイル名を変更した後も引き続き使用する場合は、新しい名前をサーバー構成ファイルに入力してから、実際にファイル名を変更し、次にサーバーを再起動します。

`persistFile` 要素の属性を、表 3-19 で説明します。

表 3-19 `persistFile` 要素の属性

属性	有効値	説明
<code>fileName</code>	文字列	<p>デフォルト: <code>server_name.dat</code></p> <p>サーバー永続ファイルの名前、およびオプションでパスを指定します。ファイルが次のデフォルト・ディレクトリにある場合は、パスを省略できます。</p> <p><code>ORACLE_HOME¥reports¥server¥</code></p> <p>このパスは、Windows と UNIX の両方で共通です。</p>

3.2.1.18 `jobRecovery`

例

```
<jobRecovery auxDatFiles="yes"/>
```

必須 / オプション

オプションです。ジョブ・リカバリ・メカニズムを有効化する場合は、`jobRecovery` 要素をサーバー構成ファイルに追加します。デフォルトでは、ジョブ・リカバリ・メカニズムは無効化されています。

説明

`jobRecovery` 要素には、`auxDatFiles` 属性があります。`auxDatFiles=yes` を指定すると、より復元力のあるジョブ・リカバリ・メカニズムが Oracle Reports で有効化され、元の `.dat` ファイルがなんらかの原因で破損した場合に、ジョブの復旧が最大化されます。`auxDatFiles=yes` を指定すると、`server_name.dat` (メインの `.dat` ファイル) に加えて、次の 2 つの補助ファイルが作成されます。

- `datfilename_offset.dat`: メインの `.dat` ファイルのジョブの補助情報が格納され、メインの `.dat` ファイルからのジョブの取出しを支援します。
- `datfilename_sc.dat`: スケジュールされているすべてのジョブ情報が格納されます (メインの `.dat` ファイルに格納されている情報に加えて)。

ジョブ・リカバリ・メカニズムが有効化されている場合、Reports Server は起動時にメインの `.dat` ファイルを読み取るときに `datfilename_offset.dat` ファイルと連携し、それに格納されている補助情報を使用します。メインの `.dat` ファイルが破損しており、すべてのジョブ情報を取り出せない場合は、`datfilename_sc.dat` ファイルの読取りを開始して、スケジュールされたジョブをこのファイルにリカバリします。このように、`datfilename_sc.dat` がバックアップ・ファイルとしての役割を果たすことで、メインの `.dat` ファイルが破損した場合の、スケジュールされたジョブのリカバリ可能性が最大化されます。

ジョブ・リカバリ・メカニズムが有効化されているときに `datfilename_offset.dat` ファイルが見つからない場合 (`jobRecovery` 要素が初めて有効化されたときなど)、Reports Server はメインの `.dat` ファイルからジョブを読み取り、残りの 2 つの補助ファイルを一から作成します。

`server_name.dat`、`datfilename_offset.dat` および `datfilename_sc.dat` は 3 ファイルで 1 つのセットを形成し、補助ファイルが有効なのはジョブ・リカバリ・メカニズムが有効化されている場合のみです。ジョブ・リカバリ・メカニズムが無効化されているときに補助ファイルが検出された場合は、補助ファイルがファイル・システムから削除され、ファイル間の整合性が維持されます。こうした整合性上の理由で、これらの 3 つのファイルは必ず一緒に扱う必要があります (たとえば、マシン間でファイルをコピーする場合は、3 ファイルを一緒にコピーする必要があります)。

jobRecovery 要素の属性を、表 3-20 で説明します。

表 3-20 jobRecovery 要素の属性

属性	有効値	説明
auxDatFiles	yes	デフォルト: no
	no	「説明」を参照してください。

3.2.1.19 identifier

例

```
<identifier confidential="yes"
encrypted="yes">fpoiVNFvnlkjRPortn+sneU88=NnN</identifier>
```

必須 / オプション

オプションです。サーバー構成ファイルに 1 つの identifier 要素を指定できます。

説明

identifier 要素は、Oracle Reports をインストールしたときに、Reports Configuration Assistant によって自動的に構成ファイルに書き込まれます。Reports Configuration Assistant は、この値を SERVERACCESSKEY/12312312313 という形式で設定します。ここで、SERVERACCESSKEY はユーザー名で、ランダムに生成された数 (12312312313) はパスワードです。ユーザー名とパスワードは暗号化され、OracleAS Reports Services の構成時に rwservers.template と targets.xml に書き込まれます。インストール後に起動されたすべての Reports Server が、この identifier 情報を構成ファイルに保持することになります。

セキュアでない Reports Server では、identifier 要素の値が次の場合に使用されます。

- Reports Queue Manager を介して Reports Server に接続するとき
- コマンドラインから Reports Server を停止するとき

このどちらの場合も、コマンドラインでは、identifier 要素で指定された値と同じ authid 値を指定する必要があります。特定のパスワードを指定するには (生成されたパスワードは擬似的なランダム数のため)、次の手順を実行する必要があります。

1. サーバー構成ファイルの server_name.conf を編集します。
2. 生成された username/password の暗号化値をカスタムの値に置換します。
3. encrypted=no を設定します。

注意: confidential=no は、username/password の暗号化が不要な場合にのみ設定してください。

次に例を示します。

```
<identifier confidential="yes" encrypted="no">username/password</identifier>
```

4. Reports Server の再起動時に、encrypted=yes が設定されます。
5. targets.xml ファイルを編集し、server_name.conf ファイルと同じ username および password 値を指定します。

この変更を行ったら、ただちに Reports Server を再起動する必要があります。その結果、ユーザー名とパスワードが自動的に暗号化され、encrypted が yes にリセットされます。この値は次のようになります。

```
<identifier confidential="yes"
encrypted="yes">fpoiVNFvnlkjRPortn+sneU88=NnN</identifier>
```

セキュアな Reports Server では、認証はセキュリティ・インフラストラクチャ、つまり Oracle Internet Directory リポジトリを使用して実行されます。そのため、Reports Server の停止時またはコンソール・ウィンドウからの Reports Queue Manager の起動時に、identifier 要素の値を渡すことはできません。

注意： rwservlet.properties ファイルで DIAGNOSTIC=NO が設定されているときは、このユーザー名とパスワードは、rwservlet Web コマンド (getjobid、getserverinfo、showjobs、showenv など) を使用する場合にも使用されます。DIAGNOSTIC=NO の場合は、このユーザー名とパスワードを持つ管理者を除き、Web コマンドは使用できません。

Reports Queue Manager の詳細は、Reports Queue Manager のオンライン・ヘルプを参照してください。rwservlet.properties の詳細は、[第 3.4 項「Reports Servlet の構成」](#)を参照してください。

3.2.1.20 pluginParam

例

```
<pluginParam name="mailServer">smtp01.mycorp.com</pluginParam>
```

必須 / オプション

オプションです。pluginParam 要素は、いくつでも指定できます。

説明

pluginParam 要素は、電子メール、JDBC プラッグابل・データソース (PDS)、テキスト PDS などの一部のビルトイン宛先で使用できるプラグインを指定する手段を提供します。これはビルトイン宛先の FTP および WebDAV では使用されず、Fax などのカスタムのプラグガブル・デスティネーションでは使用できません。

ビルトイン宛先でサポートおよび必要とされるプラグインであれば、任意のプラグイン・パラメータを指定でき、また自由に命名できます。

pluginParam 要素の属性を、[表 3-21](#) で説明します。

表 3-21 pluginParam 要素の属性

属性	有効値	説明
name	文字列	プラグイン・パラメータの名前。
type	text	デフォルト: text
	file	指定されているプラグインのタイプ。
	url	text の場合は、メール・サーバーの名前など、指定されているプラグイン・パラメータを識別できる文字列を指定します。text は、pluginParam 要素の内容がテキストであることを示すので、getPluginParam() メソッドは要素に指定されている内容をそのまま返します。 file の場合は、プラグイン・パラメータ・ファイルのディレクトリ・パスとファイル名を指定します。ディレクトリ・パスは、Reports Server のホスト・マシン (Windows または UNIX) の標準に従って指定します。file は、pluginParam 要素の内容がファイル名であることを示すので、getPluginParam() メソッドは指定されたファイルから読み込んだ内容を返します。 url の場合は、FTP サイトへの完全な URL など、プラグイン・パラメータで必要とする完全な絶対 URL を指定します。url は、pluginParam 要素の内容が URL であることを示すので、getPluginParam() メソッドは、その URL から読み込んだ内容を返します。使用する URL と OracleAS Reports Services は、ファイアウォールの同じ側にある必要があります。 デフォルト・タイプ (text) を使用するときは、pluginParam 文字列で指定する必要はありません。この項の最初に示した例では、プラグイン・パラメータ (メール・サーバー名) がデフォルト・タイプの text であるので、タイプの指定がありません。

3.2.1.21 environment

例

```
<environment id="JP">
  <envVariable name="NLS_LANG" value="Japanese_Japan.JA16SJIS"/>
  <envVariable name="NLS_CURRENCY" value="¥"/>
  <envVariable name="DISPLAY" value="MyServer.MyCompany.com:0.0"/>
</environment>
```

必須 / オプション

オプションです。environment 要素は、必要な数だけ指定できます。

説明

environment 要素は、特定のランタイム環境を設定するために使用する特性 (環境変数) を定義します。environment 要素は、必要な数だけ指定できます (たとえば、サポートが必要な言語や地域ごとに 1 つずつ指定できます)。environment 要素の中では、必要な数だけ envVariable 要素を追加できます。

environment 要素の ID を参照して、その要素の設定を呼び出します。environment 要素の ID は、次のところから参照できます。

- Reports Server 構成ファイルの engine 要素の defaultEnvId 属性。エンジンの起動時に、対応する環境設定をそのエンジンに適用します。詳細は、[第 3.2.1.4 項「engine」](#)を参照してください。
- レポートのジョブ・リクエストのコマンドライン・キーワード ENVID。特定のレポートのジョブ・リクエストについてのみ環境設定を有効にします。

environment 要素の属性を、表 3-22 で説明します。

表 3-22 environment 要素の属性

属性	有効値	説明
id	文字列	環境の名前

environment 要素には、envVariable というサブ要素が 1 つあります。各 envVariable は、名前と値のペアで指定します。これらのサブ要素には標準の環境変数またはユーザー定義の環境変数を設定できます。

envVariable 要素の属性を、表 3-23 で説明します。

表 3-23 envVariable 要素の属性

属性	有効値	説明
name	文字列	使用する環境の名前 (NLS_LANG など)
value	文字列	name 属性で識別される環境変数に割り当てる値

3.2.2 動的な環境切替え

動的な環境切替え機能を使用すると、Reports Server の起動後または特定のジョブ・リクエスト中に環境を動的に変更できます。このことは、Reports Server の 1 つのインスタンスが、言語、通貨、表示設定などの任意の環境設定でレポートを処理できることを意味します。

動的な環境切替え機能を有効化するには、最初に、使用する Reports Server 構成ファイルに environment 要素を追加して、特定のランタイム環境を設定する必要があります。ある environment 要素を一度設定すると、次のどちらの方法でも設定の切替えができます。

- Reports Server 構成ファイルの engine 要素の defaultEnvId 属性の値を environment 要素の id に設定し、エンジンの起動時に対応する環境設定をそのエンジンに適用します。詳細は、第 3.2.1.4 項「engine」を参照してください。
- ENVID コマンドライン・キーワードの値を environment 要素の id に設定し、現行のレポートのジョブ・リクエストについてのみ環境設定を有効にします。詳細は、第 A.3.34 項「ENVID」を参照してください。

3.2.2.1 例

次の例は、動的な環境切替え機能の使用方を示しています。

例 1

日本語のレポートを Reports Server から実行するとします。日本語でのレポートの実行に適した環境は、次のとおりです。

- NLS_LANG = Japanese_Japan.JA16SJIS
- 通貨単位 (NLS_CURRENCY) は、日本通貨の円 (¥) に設定されます。
- UNIX で Reports Server を実行している場合は、DISPLAY の設定が必要な場合があります。

最初に、Reports Server 構成ファイルに environment 要素を追加する必要があります。次に例を示します。

```
<environment id="JP">
  <envVariable name="NLS_LANG" value="Japanese_Japan.JA16SJIS"/>
  <envVariable name="NLS_CURRENCY" value="¥"/>
  <envVariable name="DISPLAY" value="MyServer.MyCompany.com:0.0"/>
</environment>
```

environment 要素を設定すると、次のどちらかの方法を使用して日本語出力のレポートをリクエストできます。

- Reports Server 構成ファイルで、engine 要素の defaultEnvId 属性を次のように使用します。

```
<engine id="rwEng" initEngine="1" minEngine="0" maxEngine="10" engLife="50"
maxIdle="30" defaultEnvId="JP"/>
```

値 JP によって、Reports Server 構成ファイルの environment 要素が識別されます。最初のエンジンは、この environment 要素で指定される環境設定で起動されます。

- ENVID コマンドライン・キーワードを、次のように設定します。

```
http://yourWebServer:port/reports/rwservlet?SERVER=yourreportserver
&REPORT=Japanese.rdf&USERID=username/passwd@db&DESFORMAT=htmlcss
&DESTYPE=cache&ENVID=JP
```

この URL が送信されると、Reports Server ではオプションの ENVID キーワードが検出され、指定された id (この例では JP) と、構成ファイルの environment 要素の対応する id を一致させます。すでに Reports Server でこれらの設定でエンジンを実行している場合は、その既存エンジンを再利用してジョブを処理します。そうでない場合は、現在の環境と、JP environment 要素で指定された 3 つの環境変数を使用してエンジンを起動します。新しいエンジンの起動によって、Reports Server の maxEngines 設定を超える場合、Reports Server ではエンジンを 1 つ停止してから新しいエンジンを起動します。engLife の設定を超えていない場合であっても、エンジンが停止される場合があります。Reports Server で正しい環境でエンジンが実行されていれば、そのエンジンによってジョブが処理され、指定された DESTYPE に出力がルーティングされます。

ENVID をジョブで使用しない場合、Reports Server では defaultEnvId 環境で起動されたエンジンを使用してリクエストが処理されます。Reports Server 構成ファイルの engine 要素に対して defaultEnvId が指定されていない場合、Reports Server インスタンスの起動に使用された設定がエンジンによって継承されます。

例 2

次の例では、この環境切替え機能を使用して、例 1 で日本語のレポートの実行に使用した Reports Server で同時にアラビア語のレポートを実行する方法を示します。

Reports Server 構成ファイルに別の environment 要素を次のように追加します。

```
<environment id="AR">
  <envVariable name="NLS_LANG" value="Arabic_United Arab Emirates.AR8ISO8859P6"/>
  <envVariable name="NLS_CALENDAR" value="Arabic Hijrah "/>
</environment>
```

アラビア語のレポートは、次のコマンドラインを使用して、Reports Server に送信します。

```
http://yourWebServer:port/reports/rwservlet?SERVER=yourreportserver
&REPORT=arabic.rdf&USERID=username/passwd@db&DESFORMAT=htmlcss
&DESTYPE=cache&ENVID=AR
```

ジョブが ENVID=AR で送信されるので、Reports Server は、Reports Server 構成ファイルの要素 AR で指定された環境でエンジンを検索または起動します。ジョブは新しいエンジンで処理され、出力は指定の宛先に配布されます。

例 3

次の例では、環境切替え機能を JSP レポートと組み合わせて、つまり Reports Servlet (rwservlet) と組み合わせずに使用する方法を示します。

Reports Server 構成ファイルに、次の environment 要素があるとします。

```
<environment id="UK">
  <envVariable name="NLS_LANG" value="AMERICAN_UNITED KINGDOM.WE8ISO8859P1"/>
</environment>

<environment id="US">
  <envVariable name="NLS_LANG" value="AMERICAN_AMERICA.WE8ISO8859P1"/>
</environment>
```

JSP レポートで次のような書式マスクを使用すると、通貨記号、グループ化記号、小数点記号が環境に応じて変化します。

```
<rw:field id="sal" src="sal" formatMask="L999G999D999"/>
```

イギリスの通貨記号、グループ化記号、小数点記号を使用してレポートを実行するには、次の URL を使用します。

```
http://myserver:port/test/myjsp?USERID=scott/tiger@orcl&ENVID=UK
```

注意: ENVID=UK を、cgicmd.dat ファイルのキーに追加できます。

3.2.2.2 使用上の注意

- この機能は多数の言語のレポート処理に最適ですが、このほかにも様々な用途に幅広く使用できます。レポートが正常に実行されるために特定の環境が必要であるような場合に、この機能が役に立ちます。
- Reports Server は、特定の環境のリクエストを取得するときに、environment ID ごとに1つまたは複数のエンジンを起動します。ただし、エンジンの総数は、そのエンジン・タイプについて指定された maxEngine を超えることができません。maxEngine は、Reports Server 構成ファイルで指定されている environment 要素の数以上の値に設定することをお勧めします。
- defaultEnvId は、rwEng 以外のプラグガブル・エンジンにも適用できます。Reports Server は、指定された environment ID でプラグガブル・エンジンを起動します。
- インプロセスの Reports Server で使用されるエンジンの場合、環境変数の優先順位の最上位から最下位までの順序は、次のようになります。
 - reports.sh (UNIX のみ)。

注意: 現行の reports.sh ファイルを変更した場合は、バックアップを保存しておき、Oracle Reports をインストールした後で、最新バージョンでインストールされた reports.sh と保存しておいたファイルをマージします。最新の reports.sh には、変更する必要がある内容がいくつかあります。

- Reports Server 構成ファイル内の environment 要素。
- ORACLE_HOME/j2ee/OC4J_BI_Forms/config/oc4j.properties ファイル: oracle.home プロパティは ORACLE_HOME の設定を定義し、oracle.path は PATH の設定を定義します。
- ORACLE_HOME/opmn/conf/opmn.xml の <oc4j instanceName="OC4J_BI_Forms" gid="OC4J_BI_Forms">にある <environment> 要素。
- システム設定とレジストリ (Windows のみ)。

- スタンドアロン・サーバーで使用されるエンジンの場合、環境変数の優先順位の最上位から最下位への順序は、次のようになります。
- reports.sh (UNIX のみ)。

注意： 現行の reports.sh ファイルを変更した場合は、バックアップを保存しておき、Oracle Reports をインストールした後で、最新バージョンでインストールされた reports.sh と保存しておいたファイルをマージします。最新の reports.sh には、変更する必要がある内容がいくつかあります。

- Reports Server 構成ファイル内の environment 要素。
- rwsrvr.sh を起動するコンソールにある環境設定。
- システム設定とレジストリ (Windows のみ)。
- ENVID に設定された同じ環境変数が reports.sh (ORACLE_HOME/bin/) にも設定されている場合、Reports Server は環境変数の値を ENVID ではなく reports.sh から取得します。

たとえば、環境切替え機能を使用して REPORTS_PATH 環境変数を別のエンジンに設定するとします。しかし、reports.sh ファイルにも同じ REPORTS_PATH 環境変数が設定されています。そのため、リクエストを渡したときに、Reports Server では ENVID に設定された REPORTS_PATH ではなく、reports.sh に設定された REPORTS_PATH のみが使用されます。

この問題を回避する手順は、次のとおりです。

1. reports.sh を開き、環境変数の値をコメント化します。たとえば、reports.sh ファイルに設定された REPORTS_PATH 値をコメント化します。
2. server_name.conf ファイルを開きます。
3. reports.sh ファイルの環境変数値を、server_name.conf ファイルにコピーします。次に例を示します。

```
<environment id="default">
  <envVariable name=REPORTS_PATH value="$ORACLE_
    HOME/reports/templates:$ORACLE_
    HOME/reports/samples/demo:$ORACLE_HOME/reports/integ:$ORACLE_
    HOME/reports/printers"/>
</environment>

<environment id="testenv">
  <envVariable name="REPORTS_PATH"
    value="/private/file_path:$ORACLE_HOME/reports/templates:$ORACLE_
    HOME/reports/samples/demo:$ORACLE_HOME/reports/integ:$ORACLE_HOME/
    reports/printers"/>
</environment>
```

4. defaultEnvId の値を、server_name.conf ファイルの該当するタグに追加します。たとえば、defaultEnvId 属性を engine 要素に追加し、エンジンがデフォルトの REPORTS_PATH で起動するようにします。

```
<engine id="rwEng" class="oracle.reports.engine.EngineImpl" initEngine="1"
  maxEngine="1" minEngine="0" engLife="50" maxIdle="30"callbackTimeOut="90000"
  defaultEnvId="default">
```

5. これでレポートを実行します。

3.2.3 OracleAS Portal への接続

デフォルトでは、ポータル・ユーザーのみを使用して Reports Server を OracleAS Portal に接続することができます。最初に適切な権限がスキーマに割り当てられていないかぎり、scott/tiger のような通常のユーザー ID を使用できません。

ポータル・スキーマ以外のスキーマに適切な権限を割り当てるためには、次のスクリプトを SQL*Plus から OracleAS Portal のユーザーとして実行する必要があります。

```
ORACLE_HOME/portal/admin/plsql/wor/rwgrant.sql
```

このスクリプトがロードされると、新しいスキーマの接続文字列（たとえば、repapp/repapp@orc1）を入力するよう指示されます。スクリプトは、この新しいスキーマに適切な権限を割り当てます。これにより、この接続文字列を Reports Server 構成ファイルで指定して、OracleAS Portal に接続することが可能になります。

3.3 Reports Server 検出メカニズムの構成

Oracle Reports 10g リリース 2 (10.1.2) では、Borland 社の VisiBroker にかわって、Sun 社の業界標準の Java Developer's Kit Object Request Broker (JDK ORB) が使用されています。これによって、クライアントからのサブネットにわたる Reports Server リクエストがサポートされ、さらにはブロードキャスト・メカニズムの使用によりサブネット内外での動的な Reports Server 検出が実現されます。

注意： Oracle Reports 10g リリース 2 (10.1.2) では、JDK ORB 実装の診断に使用する rwdiag 実行可能ファイルも導入されています。rwdiag を使用することで、以前の VisiBroker 実装で使用できた osfind の機能を代替でき、実行中の ORB アプリケーションに関する情報と ORB 関連のネットワーク通信のログギング・オプションが提供されます。rwdiag の詳細は、付録 E 「Reports Server とブリッジの診断ユーティリティ」を参照してください。

JDK ORB 実装では、ネットワーク内の Reports Server の検出方法として、次のどちらかを選択できます。

- Oracle Reports ビルトインのブロードキャスト・メカニズム (第 1.4.1 項「ブロードキャスト・メカニズムを使用したサーバーの検出」を参照)
- Common Object Service (COS) ネーミング・サービス (第 1.4.2 項「COS ネーミング・サービスを使用したサーバーの検出」を参照)

この項では、Reports Server の検出に必要な構成要素について説明します。

- ネットワークの構成要素 (rwnetworkconf.dtd) (両方の方法で必要)
- ブリッジの構成要素 (bridgeconf.dtd) (サブネットにわたるブロードキャスト・メカニズムで必要)

注意： Oracle Reports ビルトインのブロードキャスト・メカニズムでは、ホスト・マシンがネットワーク内にある必要があります。この制約によって、ブロードキャスト・メカニズムが機能しない可能性のある例を2つ、それぞれの解決策とあわせて次に示します。

1. ホスト・マシンがネットワーク内にない（つまり、スタンドアロンである）場合。

Windows プラットフォームでの解決策： MS ループバック・アダプタをインストールします。詳細は、Microsoft 社の Web サイト (<http://microsoft.support.com>) を参照してください。次のようにして、マシンの IP アドレスを指定します（XP または Windows 2000 の場合）。

1. デスクトップで「**マイ ネットワーク**」を右クリックして、「**プロパティ**」を選択します。
2. MS ループバック・アダプタを右クリックして、「**プロパティ**」を選択します。
3. 「**プロパティ**」ダイアログ・ボックスで「**インターネット プロトコル (TCP/IP)**」を選択して、「**プロパティ**」をクリックします。
4. 「**次の IP アドレスを使用する**」を選択して、有効な IP アドレスを入力します。「サブネットマスク」フィールドに、自動的に値が入力されます。ローカル・ホストの IP (127.0.0.1) は使用しないでください。たとえば、「198.162.1.1」と入力します。
5. 「**OK**」をクリックして、表示された指示に従います。

UNIX プラットフォームでの解決策： ネットワーク構成ファイル (rwnetwork.conf) の multicast 要素を namingService 要素に置き換えることによって、ビルトインのブロードキャスト・メカニズムを無効にし、かわりに COS ネーミング・サービスを有効にするように、検出メカニズムを構成します（詳細は、第 3.3.1.3 項「**namingService**」を参照）。

2. ホスト・マシンが、VPN 経由でネットワークに接続されている場合。

Windows および UNIX プラットフォームでの解決策： ネットワーク構成ファイル (rwnetwork.conf) の multicast 要素を namingService 要素に置き換えることによって、ビルトインのブロードキャスト・メカニズムを無効にし、かわりに COS ネーミング・サービスを有効にするように、検出メカニズムを構成します（詳細は、第 3.3.1.3 項「**namingService**」を参照）。

3.3.1 ネットワークの構成要素 (rwnetworkconf.dtd)

rwnetworkconf.dtd は、ネットワーク構成ファイル (rwnetwork.conf) 内の様々な要素と属性のデータ型定義が格納された DTD ファイルで、Reports Server の検出にビルトインのブロードキャスト・メカニズムまたは COS ネーミング・サービスのどちらを使用する場合にも必要です。このファイルは、次のディレクトリに格納されています。

```
ORACLE_HOME\reports\%dtd
```

データ型定義では、関連の XML ファイルで使用できるすべての要素、これらの要素に関連する属性、およびこれらの属性のデフォルト値が示されます。

```
<!--
Copyright 2005 Oracle Corp.
500 Oracle Parkway, Redwood Shores, CA 94065, U.S.A. All rights reserved.

This is the DTD defining the Reports network configuration file
(XML) format/syntax.
-->

<!ELEMENT discoveryService ( multicast | namingService )>
<!ATTLIST discoveryService
    version          CDATA "10.1.2">
```

```

<!ELEMENT multicast EMPTY >
<!ATTLIST multicast
  channel      CDATA      #REQUIRED
  port         CDATA      #REQUIRED
  timeout      CDATA      "1000"
  retry        CDATA      "3">

<!ELEMENT namingService EMPTY >
<!ATTLIST namingService
  name         CDATA      #REQUIRED
  host         CDATA      #REQUIRED
  port         CDATA      #REQUIRED>

```

rwnetworkconf.dtd ファイルでは、次の要素を使用して、Reports Server の検出メカニズムを構成します。

- [discoveryService](#)
- [multicast](#)
- [namingService](#)

3.3.1.1 discoveryService

必須 / オプション

必須です。構成ファイルに 1 組の discoveryService 要素の開始タグと終了タグを指定できます。

説明

discoveryService 要素は、ネットワーク構成ファイルのコンテンツ領域を開始および終了します。ファイルの階層に関しては、その他すべての要素は discoveryService 要素に従属します。

3.3.1.2 multicast

例

```

<multicast channel="228.5.6.7" port="one of the port in allotted AS ports"
  timeout="1000" retry="3"/>

```

必須 / オプション

条件付き。namingService 要素と multicast 要素は同時に使用できません。つまり、一度に構成できる要素は、このうちの 1 つのみです。

説明

multicast 要素には、Reports Server でビルトインのブロードキャスト・メカニズムが実行されている場所の識別に必要な情報が含まれます。デフォルトでは、rwnetwork.conf に multicast が指定されています。ただし、次の例のように、ビルトインのブロードキャスト・メカニズムが使用環境に適さないため、Reports Server の検出に Common Object Service (COS) ネーミング・サービスを使用する必要がある場合は、namingService 要素を指定できます。

- VPN を使用してネットワークに接続しているマシン上に Oracle Reports がインストールされている場合
 - ネットワーク上のブロードキャスト・トラフィックを回避する必要がある場合
- multicast 要素の属性を、[表 3-24](#) で説明します。

表 3-24 multicast 要素の属性

属性	有効値	説明
channel	ブロードキャスト・チャンネル。	Reports Server によって使用されるブロードキャスト・チャンネル。
port	ブロードキャスト・ポート。	Reports Server によって使用されるブロードキャスト・ポート。
timeout	レスポンスを待つ時間（ミリ秒単位）。この設定の最適値は 1000 です。	Reports クライアントは、Reports Server からのレスポンスを、指定されたタイムアウト期間だけ待ちます。
retry	再試行カウント。	タイムアウト期間が経過しても Reports Server からのレスポンスがない場合、Reports クライアントは指定された回数だけ再試行します。

注意： デフォルトのチャンネルとポートは、絶対的に必要でないかぎり変更しないでください。rwnetwork.conf のデフォルトのポート値は、Oracle Application Server のインストール時に割り当てられます。

rwnetwork.conf のカスタマイズが必要な場合は、Reports Server 用に予約されている有効なポート範囲（14021 ~ 14030）を指定する必要があります。サブネットにわたる Reports Server の検出に Oracle Reports ブリッジを使用する場合は、ブリッジが正常に機能するように timeout と retry 値の設定には注意する必要があります。timeout 値の設定については、表 3-24 を参照してください。詳細は、第 20.3 項「[rwdiag を使用したブリッジおよびネットワークのタイムアウト設定](#)」を参照してください。トラブルシューティングの例と解決策については、第 D.8 項「[Oracle Reports ブリッジの問題の診断](#)」を参照してください。

3.3.1.3 namingService

例

```
<namingService name="Cos" host="mymachine.mydomain.com" port="14021"/>
```

必須 / オプション

条件付き。namingService 要素と multicast 要素は同時に使用できません。つまり、一度に構成できる要素は、このうちの 1 つのみです。

説明

namingService 要素には、COS ネーミング・サービスが実行されているホスト名とポートの識別に必要な情報が含まれます。この要素は、次の例のように、ビルトインのブロードキャスト・メカニズムが使用環境に適さない場合のみ指定してください。

- VPN を使用してネットワークに接続しているマシン上に Oracle Reports がインストールされている場合
- ネットワーク上のブロードキャスト・トラフィックを回避する必要がある場合

詳細は、第 1.4.2 項「[COS ネーミング・サービスを使用したサーバーの検出](#)」を参照してください。

OPMN を使用して COS ネーミング・サービスを制御するには、第 3.7.1.5 項「[COS ネーミング・サービスの指定](#)」を参照してください。

注意： namingService 要素を使用して COS ネーミング・サービスを構成する前に、第 2.3 項「COS ネーミング・サービスの起動と停止」の説明のよ
うに、ネーミング・サービスを起動する必要があります。

namingService 要素の属性を、表 3-25 で説明します。

表 3-25 namingService 要素の属性

属性	有効値	説明
name	Cos	ネーミング・サービスの名前
host	ホスト名 /IP	ネーミング・サービスが実行されているマシンのホスト名
port	ポート番号	ネーミング・サービスが実行されているマシンのポート番号

3.3.2 ブリッジの構成要素 (bridgeconf.dtd)

bridgeconf.dtd は、ブリッジ構成ファイル (repbrg_bridgename.conf) 内の様々な要素と属性のデータ型定義が格納された DTD ファイルで、サブネットにわたる Reports Server の検出にビルトインのブロードキャスト・メカニズムを使用する場合に必要です。

Oracle Reports ブリッジは、Reports Server と Reports クライアントによってサブネットを越えてブロードキャストされるパケットに対するゲートウェイの役割を果たします。たとえば、設定例では、Oracle Reports コンポーネントは異なるサブネットにインストールされています (Reports Servlet はサブネット A、Reports Server はサブネット B)。この構成を可能にするには、両方のサブネットで Oracle Reports ブリッジを起動する必要があります。ブリッジの構成には、ホストおよびポートの設定が含まれます。サブネット A の Oracle Reports ブリッジは、信頼できる TCP を介してサブネット B の Oracle Reports ブリッジと通信し、サブネット B のサーバー情報を取得します。また、その逆も同様です。詳細は、第 1.4.1.2 項「複数のサブネットにわたるサーバーの検出」を参照してください。

Oracle Reports ブリッジの起動については、第 2.2 項「Oracle Reports ブリッジの起動と停止」を参照してください。

トラブルシューティングの例と解決策については、第 D.8 項「Oracle Reports ブリッジの問題の診断」を参照してください。

```
<!--
Copyright 2005 Oracle Corp.
500 Oracle Parkway, Redwood Shores, CA 94065, U.S.A. All rights reserved.
This is the DTD defining the Reports Bridge Version 10g Configuration file
(XML) format/syntax.
-->
<!ELEMENT bridge (networkConfig?,
                  identifier?,
                  trace?,
                  remoteBridges? ) >
<!ATTLIST bridge
  version      CDATA #REQUIRED
  port         CDATA #REQUIRED
  timeout     CDATA #REQUIRED>

<!ELEMENT networkConfig EMPTY>
<!ATTLIST networkConfig
  file      CDATA #REQUIRED>

<!ELEMENT identifier (#PCDATA)>
<!ATTLIST identifier
  confidential (yes|no)      "yes"
  encrypted   (yes|no)      "no" >
```

```

<!ELEMENT remoteBridges (remoteBridge*)>

<!ELEMENT remoteBridge EMPTY>
<!ATTLIST remoteBridge
  host          CDATA          #REQUIRED
  port          CDATA          #REQUIRED>
<!ELEMENT trace EMPTY>
<!ATTLIST trace
  traceFile     CDATA          #IMPLIED
  traceOpts     (trace_err|trace_inf|
                trace_dbg| trace_all) "trace_all"
  traceMode     (trace_replace|trace_append) "trace_replace">

```

bridgeconf.dtd ファイルでは、次の要素を使用して、サブネットにわたるサーバー検出に使用されるブリッジを構成します。

- [bridge](#)
- [networkConfig](#)
- [identifier](#)
- [remoteBridges](#)
- [remoteBridge](#)
- [trace](#)

3.3.2.1 bridge

例

```

<bridge version="10.1.2" port="14011" timeout="12000">
  <!--networkConfig file="rwnetwork.conf" --></networkConfig-->
  <!--identifier encrypted="no"
        confidential="yes">%USERNAME%/%PASSWORD%</identifier-->
  <!--trace traceOpts="trace_all"></trace-->
  <!-- Specify one or more remote bridges inside remoteBridges element -->
  <!--remoteBridges>
    <remoteBridge host="%HOST%" port="%PORT%"></remoteBridge>
  </remoteBridges-->
</bridge>

```

必須 / オプション

必須です。構成ファイルに 1 組の bridge 要素の開始タグと終了タグを指定できます。

説明

bridge 要素は、ブリッジ構成ファイルのコンテンツ領域を開始および終了します。ファイルの階層に関しては、その他すべての要素は bridge 要素に従属します。

bridge 要素の属性を、[表 3-26](#) で説明します。

表 3-26 bridge 要素の属性

属性	有効値	説明
version	10.1.2	ブリッジのバージョン。
port	Oracle Reports ブリッジ・コンポーネントに割り当てる範囲 (つまり、14011 ~ 14020)	ブリッジがリスニングするポート。

表 3-26 bridge 要素の属性 (続き)

属性	有効値	説明
timeout	1000	ミリ秒 (ms) 単位の数値。ブリッジは、リモート・ブリッジからのレスポンスを、この期間だけ待ちます。

注意： ブリッジ構成ファイルのデフォルトのポート値は、Oracle Application Server のインストール時に割り当てられます。ブリッジ用の構成ファイルの生成に使用される `rwbridge` テンプレート・ファイルには、このデフォルト・ポートが指定されています。ブリッジ名が `mybridge` であれば、その構成ファイル名は `repbrg_mybridge.conf` となります。

ブリッジ用のポート番号のカスタマイズが必要な場合は、Oracle Reports ブリッジ用に予約されている有効なポート範囲 (14011 ~ 14020) を指定する必要があります。

3.3.2.2 networkConfig

`networkConfig` 要素の詳細は、[第 3.2.1.8 項「networkConfig」](#) を参照してください。

3.3.2.3 identifier

例

```
<identifier confidential="yes"
  encrypted="yes">fpoiVNFvnlkjRPortn+sneU88=NnN</identifier>
```

必須 / オプション

オプションです。この要素がコメント化されている場合は、ブリッジ停止コマンドの発行時にセキュリティ・チェックが実行されません。

説明

`identifier` 要素は、ブリッジ停止前のセキュリティ・チェックの実行を指定します。

`identifier` 要素の値を設定する手順は次のとおりです。

1. ブリッジ構成ファイルの `identifier` 要素のコメントを外します。
2. その値を管理者のユーザー名 / パスワードに設定し、属性 `encrypted=no` および `confidential=yes` を設定します。これでユーザー名 / パスワードは、Oracle Reports ブリッジの再起動時に暗号化されます。

次に例を示します。

```
<identifier encrypted="no" confidential="yes">scott/tiger</identifier>
```

3. ブリッジを起動します。

この要素が設定されると、管理者のみがコマンドラインで `username/password` を指定して、ブリッジを停止できます。

関連項目： [第 2.2.2 項「コマンドラインからの Oracle Reports ブリッジの起動と停止」](#)

3.3.2.4 remoteBridges

例

```
<remoteBridges>
  value...
</remoteBridges>
```

必須 / オプション

オプションです。このエントリを指定しないと、対象となるブリッジはリモート・ブリッジに接続して、Reports Server 参照を取得することができません。ただし、リモート・ブリッジはこのブリッジに接続して、このサブネットで行われている Reports Server 参照を取得できます。

説明

remoteBridges 要素には、任意数の remoteBridge 要素を指定できます。

3.3.2.5 remoteBridge

例

```
<remoteBridge host="myhost.mydomain.com" port="14022"></remoteBridge>
```

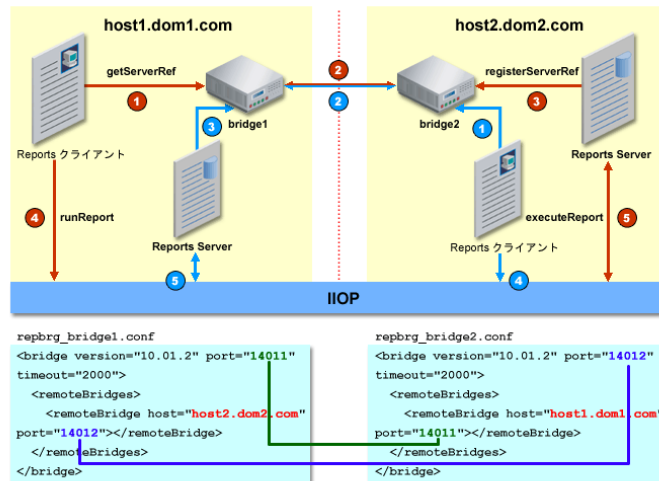
必須 / オプション

オプションです。ブリッジ構成ファイルには、remoteBridge 要素を 1 つ以上指定します。

説明

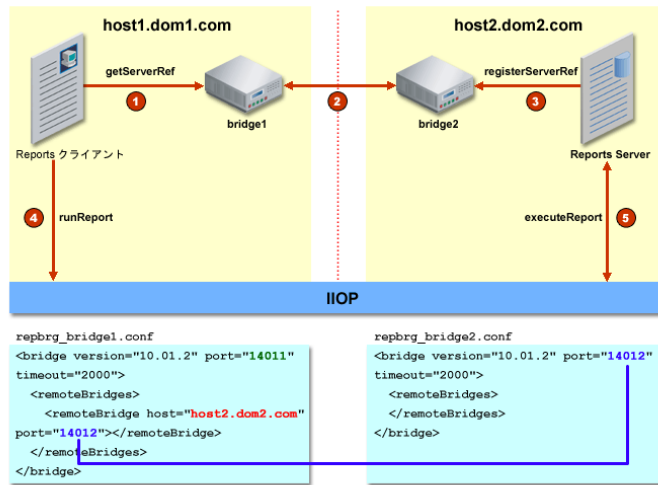
remoteBridge 要素には、リモート・ブリッジが実行されているホストおよびポートを指定します。

図 3-1 Oracle Reports ブリッジの構成 (双方向)



repbrg_bridgename.conf にオプションの remoteBridge 要素を指定すると、該当ブリッジは双方向ブリッジとして機能します。つまり、そのブリッジはリモート・ブリッジからサーバー参照を取得できます。

図 3-2 Oracle Reports ブリッジの構成（片方向）



repbrg_bridgename.conf にオプションの remoteBridge 要素を指定しない場合、該当ブリッジは片方向ブリッジとして機能します。つまり、そのブリッジはリモート・ブリッジとしてのみ機能します。片方向ブリッジはリモート・ブリッジに接続して、サーバー参照を取得することはできません。

remoteBridge 要素の属性を、表 3-27 で説明します。

表 3-27 remoteBridge 要素の属性

属性	有効値	説明
host	リモート・ブリッジのホスト名 / IP アドレス	リモート・ブリッジが実行されているマシンのホスト名または IP アドレス
port	リモート・ブリッジのポート番号	remoteBridge 要素に適用するポート番号

3.3.2.6 trace

Oracle Reports ブリッジのトレースでは、Reports Server トレースで使用可能なオプション（第 3.2.1.13 項「trace」を参照）のサブセットをサポートします。

例

```
<trace traceFile="mybridge.trc" traceOpts="trace_err|trace_inf"
traceMode="trace_append"/>
```

必須 / オプション

オプションです。ブリッジ構成ファイルに 1 つの trace 要素を指定できます。

説明

trace 要素を使用して、Oracle Reports ブリッジの実行のトレースに使用するファイルを作成し、トレースするアクティビティを指定します。

注意： レポートの実行をトレースする各種方法については、第 20.1.2 項「レポート・トレース」を参照してください。

trace 要素の属性を、表 3-28 で説明します。

表 3-28 Oracle Reports ブリッジの trace 要素の属性

属性	有効値	説明
traceFile	*.trc	デフォルト: bridge_name.trc トレース・ファイルの名前。パスの指定がない場合、トレース・ファイルは、Windows と UNIX の両方で次のディレクトリに置かれます。 ORACLE_HOME/reports/logs/
traceOpts	表 3-29 を参照。	デフォルト: trace_all トレースするアクティビティを定義します。1 つ以上の traceOpts 値を指定できます。次に例を示します。 <traceOpts="trace_err trace_inf"> 値は縦線 () で区切ります。
traceMode	trace_replace trace_append	デフォルト: trace_replace 新しいトレース情報で既存のトレース・ファイルを上書きするか (trace_replace)、既存のトレース情報を残したままトレースの末尾に追加するか (trace_append) を決定します。

表 3-29 Oracle Reports ブリッジの traceOpts 属性の有効値

値	説明
trace_err	ブリッジのエラー・メッセージを示します。
trace_inf	他のオプションでトレース・ファイルに記録されない情報をすべて記録します。
trace_dbg	デバッグ情報を記録します。
trace_all	ブリッジに関するすべての情報をトレース・ファイルに記録します。

3.4 Reports Servlet の構成

サーブレット構成ファイルの `rwervlet.properties` を使用して、Reports Servlet (`rwervlet`) を構成できます。このファイルは、Windows および UNIX とともに次のパスにあります。

`ORACLE_HOME\reports\conf\rwervlet.properties`

`rwervlet.properties` を使用して、次の処理を行います。

- キー・マップ・ファイルの位置の指定
- キー・マップ・ファイルの再ロード
- Web コマンド出力の非表示
- ログイン・ダイアログ・ボックスの選択
- Reports Servlet と JSP のトレース・オプションの設定
- Reports パラメータのデコードに使用する `rwervlet` キャラクタ・エンコードの指定
- URL に指定された HTML コードのブラウザでの実行の禁止
- `rwervlet` に対する同時接続のプール・サイズの指定
- サーバー・エラー・メッセージの外観のカスタマイズ
- インプロセス・サーバーの指定
- インプロセス・サーバーの識別
- Oracle Internet Directory 接続プーリング・パラメータの指定

- 動的に生成されたイメージの指定
- データベース認証とシステム認証 Cookie の期限切れの設定
- データベース認証とシステム認証 Cookie 用の暗号化キーの設定
- 診断およびデバッグ出力のフォーマットの追加
- `rwsvrlet` ヘルプ・ファイルの定義
- OracleAS Single Sign-On 使用の指定
- ネットワーク構成ファイルの指定
- Reports Server クラスタ名を含む OracleAS Forms Services アプリケーションの移行

また、次の項の説明に従って代替の構成ファイル名を指定すると、代替ファイルを使用して Reports Servlet を構成することもできます。

- 代替の Reports Servlet 構成ファイルの指定

構成ファイルのエントリでは、大文字と小文字を区別しません。

Windows の場合、`rwsvrlet.properties` では、1つの円記号ではなく2つの円記号 (`%%`) を使用してディレクトリ・パスを指定します。最初の円記号は次の円記号をエスケープします。エスケープしない円記号は、このファイルで別の意味になります。たとえば、Windows の `rwsvrlet.properties` ファイルでは、次のパスはその下のパスのようになります。

```
d:%orawin%reports%conf%filename.ext
```

次のようになります。

```
d:%%orawin%%reports%%conf%%filename.ext
```

UNIX の場合、ディレクトリ・パスは、次のようにプラットフォームの標準に従って指定します。

```
orawin/reports/conf/filename.ext
```

表 3-30 に、Reports Servlet 構成ファイルで定義可能なパラメータと、そのパラメータの説明が記載されている参照先の一覧をアルファベット順に示します。

表 3-30 Reports Servlet の構成パラメータ

パラメータ	参照先
ALLOWHTMLTAGS	第 3.4.7 項「URL に指定された HTML コードのブラウザでの実行の禁止」
CONNECTION_POOLSIZE	第 3.4.8 項「 <code>rwsvrlet</code> に対する同時接続のプール・サイズの指定」
COOKIEEXPIRE	第 3.4.14 項「データベース認証とシステム認証 Cookie の期限切れの設定」
DBAUTH	第 3.4.4 項「ログイン・ダイアログ・ボックスの選択」
DEFAULTCHARSET	第 3.4.6 項「Reports パラメータのデコードに使用する <code>rwsvrlet</code> キャラクタ・エンコードの指定」
DIAGBODYTAGS	第 3.4.16 項「診断およびデバッグ出力のフォーマットの追加」
DIAGHEADTAGS	第 3.4.16 項「診断およびデバッグ出力のフォーマットの追加」
DIAGNOSTIC	第 3.4.3 項「Web コマンド出力の非表示」
ENCRYPTIONKEY	第 3.4.15 項「データベース認証とシステム認証 Cookie 用の暗号化キーの設定」
ERRORTEMPLATE	第 3.4.9 項「サーバー・エラー・メッセージの外観のカスタマイズ」
HELP	第 3.4.17 項「 <code>rwsvrlet</code> ヘルプ・ファイルの定義」

表 3-30 Reports Servlet の構成パラメータ (続き)

パラメータ	参照先
IMAGE_URL	第 3.4.13 項「動的に生成されたイメージの指定」
KEYMAPFILE	第 3.4.1 項「キー・マップ・ファイルの位置の指定」
OIDCON_INCREMENT	第 3.4.12 項「Oracle Internet Directory 接続プーリング・パラメータの指定」
OID_INIT	第 3.4.12 項「Oracle Internet Directory 接続プーリング・パラメータの指定」
OID_TIMEOUT	第 3.4.12 項「Oracle Internet Directory 接続プーリング・パラメータの指定」
RELOAD_KEYMAP	第 3.4.2 項「キー・マップ・ファイルの再ロード」
REPORTS_NETWORK_CONFIG	第 3.4.19 項「ネットワーク構成ファイルの指定」
REPORTS_SERVERMAP	第 3.4.20 項「Reports Server クラスタ名を含む OracleAS Forms Services アプリケーションの移行」
SERVER	第 3.4.11 項「インプロセス・サーバーの識別」
SERVER_IN_PROCESS	第 3.4.10 項「インプロセス・サーバーの指定」
SINGLESIGNON	第 3.4.18 項「OracleAS Single Sign-On 使用の指定」
SYSAUTH	第 3.4.4 項「ログイン・ダイアログ・ボックスの選択」
TRACEFILE	第 3.4.5 項「Reports Servlet と JSP のトレース・オプションの設定」
TRACEMODE	第 3.4.5 項「Reports Servlet と JSP のトレース・オプションの設定」
TRACEOPTS	第 3.4.5 項「Reports Servlet と JSP のトレース・オプションの設定」

3.4.1 キー・マップ・ファイルの位置の指定

レポートのランタイム・コマンドラインには、ユーザーには公開しない情報が含まれることもあります。また、オプションが長い文字列で構成されていて覚えにくかったり、読みづらい URL になることもあります。

レポートのコマンドライン・オプションをキー・マップ・ファイル (cgicmd.dat) に入力して、必要なオプションが格納されているこのファイル内の特定のキー・セクションの名前だけをランタイム・コマンドとして公開することができます。

キー・マップ・ファイルの詳細は、第 13 章「レポート・リクエストの実行」を参照してください。rwservlet.properties を使用して、このファイルの位置を示します。

次に例を示します。

```
KEYMAPFILE=d:¥¥orawin¥¥reports¥¥conf¥¥cgicmd.dat
```

この例では、デフォルトのファイル名と位置を使用しています。サーブレットはすでに検索対象と検索位置を認識しているので、デフォルトではキー・マップ・ファイルの位置とファイル名に対するエントリが rwservlet.properties に表示されません。別の名前または位置のファイルを使用する場合は、ディレクトリ・パスとファイル名を格納する KEYMAPFILE パラメータを rwservlet.properties で指定する必要があります。

3.4.2 キー・マップ・ファイルの再ロード

RELOAD_KEYMAP パラメータを使用して、rwservlet がリクエストを受け取るたびにキー・マップ・ファイル (cgicmd.dat) を再ロードするかどうかを指定します。

次に例を示します。

```
RELOAD_KEYMAP=YES
```

この機能は、マップ・ファイルを頻繁に変更し、変更を自動的にロードする場合に便利です。ファイルの再ロードに要する時間は、ランタイム・パフォーマンスに影響します。

一般にこのパラメータは、本番環境では NO に設定し、テスト環境では YES に設定します。

3.4.3 Web コマンド出力の非表示

セキュアでない Reports Server において、authid に基づく管理者のみが showenv や showjobs などの Web コマンドを実行できるよう、認証メカニズムを導入することが可能です。rwservlet.properties ファイルに DIAGNOSTIC=NO を設定することによってこのような認証メカニズムを実現できます。この認証情報は、コマンドラインから Reports Server を停止するなど、管理作業にも使用されます。

注意： DIAGNOSTIC=NO の設定は、セキュアでない Reports Server に対してのみ機能します。セキュアな Reports Server ユーザーに対しては、Reports Server は、Oracle Internet Directory のエントリに基づいてユーザーの権限を確認します。

セキュアでない Reports Server の Web コマンド表示を無効にする手順は次のとおりです。

1. Reports Server を起動します。
2. rwservlet.properties ファイルで DIAGNOSTIC=NO を設定します。
3. getserverinfo などの任意の Web コマンドを使用して、rwservlet から Reports Server にアクセスします。ここで authid を渡す必要があります。

次に例を示します。

```
http://yourwebserver:portnum/reports/rwservlet/getserverinfo?server=aks+authid=scott/tiger
```

注意： Reports Server では、この authid は identifier 要素に基づき ORACLE_HOME/reports/conf/server_name.conf ファイルに保存されます。<identifier> 要素は、Reports Server 構成ファイルで定義されていない場合、最初の Web コマンドのコールによってサーバー構成ファイルで設定され、その値は、Reports Server の起動時に自動的に暗号化されます。Reports Server 構成ファイルに <identifier> 要素を事前設定するための書式は次のとおりです。

```
<identifier confidential="yes"
encrypted="no">username/password</identifier>
```

クリア・テキストのユーザー名とパスワードは、Reports Server の起動後に暗号化され、encrypted 属性は、暗号化されたことを示す yes に変更されます。

identifier 要素の詳細は、[第 3.2.1.19 項「identifier」](#) を参照してください。

前述の設定後、Web コマンドを使用する Reports Server へのアクセスには、同じ authid を渡すことが必要になります。authid を渡さない場合は、次のエラーが発生します。

```
REP-52262: Diagnostic output is disabled.
```

管理者のユーザー名またはパスワードを変更する手順は次のとおりです。

1. Reports Server 構成ファイル (`server_name.conf`) を開きます。
2. `identifier` 要素を次のように変更します。

```
<identifier encrypted="no"confidential="yes">
admin_name/admin_password</identifier>
```

ヒント: `rwserver.template` ファイルの `identifier` 要素で、管理者の `username/password` を指定できます。これにより、どの Reports Server を起動しても、同じ `authid` を渡すことができます。

3.4.4 ログイン・ダイアログ・ボックスの選択

サーブレット構成ファイルは、ユーザー ID およびパスワードのダイアログ・ボックスのテンプレートを処理する多数のパラメータを提供します。これらのダイアログ・ボックスは、ユーザーがデータベースにログインしたときやセキュア・レポートを実行したときにオープンされます。一般に、これらのパラメータはログイン画面の設定に使用される各種テンプレートを示します。これらのテンプレートは、自社ロゴ、リンク・ボタンまたはその他の HTML を自由に使用してカスタマイズできます。

DBAUTH および SYSAUTH のパラメータを使用して、個別のログイン画面に使用する HTML テンプレートの位置とファイル名を指定できます。デフォルトでは、ファイル名はそれぞれ `rwdbauth.htm` と `rwsysauth.htm` です。

たとえば、次のエントリは、データベース・ログイン画面のテンプレートを示します。

```
DBAUTH=rwdbauth.htm
```

SYSAUTH は、セキュア・レポートのログイン画面を示します。次に例を示します。

```
SYSAUTH=rwsysauth.htm
```

テンプレートが次に示すデフォルトのテンプレート・ディレクトリにある場合は、テンプレートのパスを指定する必要はありません。

```
ORACLE_HOME¥reports¥templates
```

3.4.5 Reports Servlet と JSP のトレース・オプションの設定

OracleAS Reports Services のトレース機能を使用すると、[第 20.1.2 項「レポート・トレース」](#)の説明にあるように、OracleAS Reports Services の各種コンポーネントについて多数の異なるタイプのランタイム情報を記録できます。

Reports Servlet (`rwervlet`) および JSP のランタイム情報を追跡および記録するには、`rwervlet.properties` の `TRACEOPTS` パラメータを使用します。また、任意数のトレース・オプションを入力できます。各オプションは、縦線で区切ります。

次に例を示します。

```
TRACEOPTS=trace_prf|trace_pls|trace_dbg
```

使用可能なすべてのトレース・オプションは、[表 3-14](#) で説明されています。

また、`TRACEFILE` および `TRACEMODE` のパラメータを使用できます。`TRACEFILE` を使用して、トレース・ファイルの名前を指定できます。次に例を示します。

```
TRACEFILE=myrwervlet.trc
```

デフォルトの名前は `rwervlet.trc` です。パスの指定がない場合、トレース・ファイルは、Windows と UNIX の両方で次のディレクトリに置かれます。

```
ORACLE_HOME\reports\logs\machine_name
```

`rwervlet.properties` 構成ファイルでは、オプションごとに 1 行を使用して指定します。次に例を示します。

```
TRACEOPTS=TRACE_ALL
TRACEFILE=rwervlet.trc
TRACEMODE=TRACE_REPLACE
```

TRACEMODE を使用して、新しいトレース情報で既存のトレース・ファイルを上書きするか (`trace_replace`)、既存のトレース情報を残したままトレースの末尾に追加するか (`trace_append`) を決定します。TRACEMODE は、`rwervlet` を含む Oracle Application Server Containers for J2EE コンテナの起動後に蓄積されたトレース情報を置き換えるか、トレース情報に追加します。次に例を示します。

```
TRACEMODE=trace_append
```

TRACEMODE のデフォルトは、`trace_replace` です。

3.4.6 Reports パラメータのデコードに使用する `rwervlet` キャラクタ・エンコードの指定

リクエスト URL に非 ASCII エスケープ・キャラクタまたはパラメータ・フォーム入力に非 ASCII キャラクタを使用できます。使用する場合は、あらかじめ `rwervlet.properties` ファイルにキャラクタ・エンコードを指定しておく必要があります。これにより、パラメータ名と値をデコードするときに、`rwervlet` で必要なエンコードが使用されます。

`rwervlet.properties` ファイルの `DEFAULTCHARSET` の値は次のいずれかに設定できます。

- データベースの `NLS_CHARACTERSET` (`JA16EUC` など)
- IANA 定義のキャラクタ・セット (`EUC-JP` など)

例

```
DEFAULTCHARSET=JA16EUC
```

注意： Event-Driven Publishing API の使用時にユーザー・パラメータと値に非 ASCII キャラクタを使用するには、`rwervlet.properties` ファイルに指定されている `DEFAULTCHARSET` が、パラメータ・リストの `DEFAULTCHARSET` の値と一致する必要があります。詳細は、[第 17.1.3 項「パラメータの名前と値での非 ASCII 文字の使用」](#) を参照してください。

3.4.7 URL に指定された HTML コードのブラウザでの実行の禁止

レポート・リクエスト URL の一部に含まれた HTML コードは、セキュリティの低下をもたらすおそれがあります。ブラウザの中には URL でスクリプトまたはコードを実行するものもあるためです。HTML コードは URL コマンドの一部として使用できません。

URL コマンドの一部として HTML コードを使用できないようにするには、`rwervlet.properties` の `ALLOWHTMLTAGS` プロパティを使用します。デフォルトでは、このプロパティが `NO` に設定されており、レポートの実行時に HTML コードを URL に入力できません。

有効値

- `YES`: URL に HTML コードを使用できます。

注意: ALLOWHTMLTAGS=YES に設定すると、ブラウザによっては悪意のある HTML コードが実行される可能性があります。

- NO: URL に HTML コードを使用できません。

デフォルト

NO

3.4.8 rwservlet に対する同時接続のプール・サイズの指定

rwservlet に対して同時に接続し、ジョブ・リクエストを送信できるユーザー数を定義できません。その場合は、rwservlet.properties ファイルで CONNECTION_POOLSIZE プロパティの値を設定する必要があります。

デフォルト値

500

最小値

0 (非推奨)

CONNECTION_POOLSIZE の値は、同時にジョブ・リクエストを送信するアクティブ・ユーザー数を考慮して設定します。たとえば、約 100 の同時アクティブ・ユーザーを予測できる場合は、この値を 250 に設定します。

注意: この値 (例での 100) は、同時にジョブ・リクエストを送信するアクティブ・ユーザーの数です。リクエストを送信しないでシステムに接続するユーザーの数ではありません。

3.4.9 サーバー・エラー・メッセージの外観のカスタマイズ

OracleAS Reports Services は、サーバー・エラー・メッセージのテンプレートを提供します。これらのメッセージは、原因に従って自動的に生成されます。テンプレートは、エラー・メッセージの表示に使用される視覚的な設定を提供します。

エラー・メッセージの外観は、自社ロゴやエラーと関連するアイコンなどを使用してカスタマイズできます。ヘルプ・システムや自社ホームページを表示したり、前のブラウザ・ウィンドウへ戻るためのボタンを追加することもできます。これには、自動的に生成されたエラー・メッセージに独自の HTML フレームワークを設定します。

rwservlet.properties の ERRORTEMPLATE プロパティは、エラー・メッセージ・テンプレートの名前と場所を指定します。

デフォルトのエントリは次のとおりです。

```
ERRORTEMPLATE=rwerror.htm
```

テンプレートが次に示すデフォルトのテンプレート・ディレクトリにある場合は、エラー・メッセージ・テンプレートのパスを指定する必要はありません。

```
ORACLE_HOME\reports\templates
```

エラー・メッセージ・テンプレート (rwerror.htm) のキャラクタ・セットは、すべてのプラットフォーム間での一貫性を保証する目的で iso-8859-1 になっています。以前のリリースでは、Windows プラットフォームに固有のキャラクタ・セットである windows-1252 が使用されていました。

3.4.10 インプロセス・サーバーの指定

Reports Server を Reports Servlet (rwservlet) と同じプロセスで実行することを選択した場合は、`rwservlet.properties` で次のように指定します。

```
SERVER_IN_PROCESS=YES
```

Reports Servlet と同じプロセスで Reports Server を実行しない場合は、`SERVER_IN_PROCESS=NO` を指定します。

注意： インプロセス・サーバーの実行の長所と短所は、[第 1 章「OracleAS Reports Services のアーキテクチャについて」](#)を参照してください。

OPMN による Reports Server の構成時にインプロセス・サーバーを指定する場合は、[第 3.7.1.3 項「インプロセス Reports Server の指定」](#)を参照してください。

インプロセス・サーバーの使用時における印刷およびフォントに関する問題のトラブルシューティングは、[第 D.1.11 項「インプロセス・サーバーを使用する際の印刷エラーとフォント・エラー」](#)を参照してください。

3.4.11 インプロセス・サーバーの識別

Reports Servlet (rwservlet) では、インプロセス・サーバーの名前として `SERVER` パラメータ値が使用されます。ランタイム URL などに Reports Server の名前が指定されていない場合は、`rwservlet` によって、`SERVER` パラメータで指定された名前のインプロセス・サーバーが起動され (未起動の場合)、それにジョブが送信されます。

AS インストールでは、Oracle Reports 構成ツールによって、`rwservlet.properties` ファイルの `SERVER` パラメータが次のように設定されます。

```
SERVER=rep_hostname_oraclehostname
```

インプロセス・サーバーの名前を変更するには、`rwservlet.properties` ファイルで `SERVER` パラメータを異なる名前に設定します。

```
SERVER=server_name
```

`rwservlet.properties` に `SERVER` パラメータが設定されていない場合、デフォルトのインプロセス・サーバー名は、AS インストールおよび DS インストールの両方で `rep_hostname` です。

`SERVER_IN_PROCESS=NO` が指定されている場合は、`rwservlet` によって、`SERVER` パラメータで指定された名前の外部サーバーへのバインドが試行されます。

3.4.12 Oracle Internet Directory 接続プーリング・パラメータの指定

Reports Servlet への Oracle Internet Directory 接続を指定するには、`rwservlet.properties` ファイルで、次のように Oracle Internet Directory 接続プーリング・パラメータを指定します。

```
OIDCON_INIT=Initial number of Oracle Internet Directory connections to be created when Reports Servlet is initialized
```

```
OIDCON_INCREMENT=Number of connections to be incremented when all connections are used up
```

```
OIDCON_TIMEOUT=Time in seconds for which a connection can be idle
```

これらのパラメータを指定しない場合、デフォルト値は次のようになります。

```
OIDCON_INCREMENT=10
OIDCON_INIT=10
OIDCON_TIMEOUT=0 (which specifies no timeout)
```

Reports Server では、サーバー構成ファイルの `oidconnection` 要素を使用して、Oracle Internet Directory 接続プーリング・パラメータを指定できます (詳細は、[第 3.2.1.6 項「oidconnection」](#)を参照)。

3.4.13 動的に生成されたイメージの指定

オプションで、`IMAGE_URL` パラメータを使用して、動的に生成された Reports イメージにアクセスできる位置を指定できます。

次に例を示します。

```
IMAGE_URL=http://server_or_web_server_name.domain_name:port/reports/rwservlet
```

このパラメータは、Reports Servlet (`rwservlet`) 経由で実行されない JSP で使用されます。これにより、チャートなど動的に生成されたイメージを表示できるのが、レポート実行者に制限されます。Reports Servlet 経由で実行される JSP やその他のレポート・タイプには、この保護が自動的に付加されます。

3.4.14 データベース認証とシステム認証 Cookie の期限切れの設定

`COOKIEEXPIRE` パラメータを使用して、データベースおよびシステム認証 Cookie の存続期間を分単位で指定できます。次に例を示します。

```
COOKIEEXPIRE=20
```

デフォルトは 30 です。

Cookie は、ユーザーの初回認証時に、暗号化されたユーザー名とパスワードをクライアント側で保存します。サーバーは、クライアントから Cookie を受け取ると、Cookie に保存された時間と現在のシステム時間を比較します。この時間が `COOKIEEXPIRE` に定義された時間 (分) を超えていると、サーバーはその Cookie を拒否し、エラー・メッセージとともに認証フォームをクライアントに返します。レポートを実行するには、ユーザーは再度認証を受ける必要があります。

3.4.15 データベース認証とシステム認証 Cookie 用の暗号化キーの設定

`ENCRYPTIONKEY` を使用して、データベース認証とシステム認証 Cookie のユーザー名とパスワードを暗号化するための暗号化キーを指定できます。暗号化キーには任意の文字列を設定できます。次に例を示します。

```
ENCRYPTIONKEY=egbdf
```

3.4.16 診断およびデバッグ出力のフォーマットの追加

`DIAGBODYTAGS` および `DIAGHEADTAGS` のパラメータを使用して、追加の HTML エンコードを、診断およびデバッグの出力に関連する出力ファイルの `<body>` および `<head>` のタグに挿入できます。

`DIAGBODYTAGS` は `<body>` タグ全体を定義し、`DIAGHEADTAGS` は開始および終了の `<head>` と `</head>` タグの間に使用できるタグを定義します。

これらのパラメータを使用して、診断およびデバッグの出力を見やすくするためのフォーマット・オプションを挿入できます。次に例を示します。

```
DIAGBODYTAGS=<BODY>additional HTML encoding</BODY>
```

```
DIAGHEADTAGS=<HEAD>additional HTML encoding</HEAD>
```

3.4.17 rwservlet ヘルプ・ファイルの定義

HELP キーワードを `rwservlet` コマンドとともに使用して、サーブレット関連のヘルプ・トピックを表示できます。ヘルプ・ファイルを呼び出すには、次の URL を指定します。

```
http://your_web_server/your_servlet_path/rwservlet/help
```

注意： HELP キーワードの詳細は、[第 A.3.42 項「HELP」](#) を参照してください。

このパラメータを指定しない場合に表示されるサーブレット用のデフォルトのヘルプ・ファイルもあります (`ORACLE_HOME\reports\templates\help.htm`)。ユーザー独自のヘルプ・ファイルを追加することもできます。そのためには、サーブレットのヘルプ・ファイルの名前と位置の URL を、`rwservlet.properties` の `HELPURL` パラメータで指定します。次に例を示します。

```
HELPURL=http://your_web_server/your_help_file_path/helpfile.htm
```

3.4.18 OracleAS Single Sign-On 使用の指定

OracleAS Reports Services のシングル・サインオン機能を利用するには、`rwservlet.properties` の `SINGLESIGNON` パラメータを `YES` に設定します。インストール直後のデフォルトでは、`SINGLESIGNON=YES` に設定されています。

詳細は、[第 11.3.1 項「OracleAS Single Sign-On の有効化と無効化」](#) を参照してください。

3.4.19 ネットワーク構成ファイルの指定

デフォルトでは、Reports Servlet は `rwnetwork.conf` ファイルを使用して、それが使用する検出メカニズムを決定します。カスタムのネットワーク構成ファイルを使用する場合は、`rwservlet.properties` の `REPORTS_NETWORK_CONFIG` プロパティを、`ORACLE_HOME/reports/conf` ディレクトリにある有効なネットワーク構成ファイル名に設定します。次に例を示します。

```
REPORTS_NETWORK_CONFIG=net3.conf
```

3.4.20 Reports Server クラスタ名を含む OracleAS Forms Services アプリケーションの移行

Oracle Reports 10g リリース 2 (10.1.2) では、Reports Server のクラスタリングは使用されません。Reports Server クラスタ名を含む以前のリリースの OracleAS Forms Services アプリケーションでは、参照先の Reports Server クラスタへのバインドが失敗します。

この問題を解決するために、Oracle Reports 10g リリース 2 (10.1.2) では、クラスタ名を Reports Server 名にマップ可能な `REPORTS_SERVERMAP` が導入されました。これを使用することで、すべての OracleAS Forms Services アプリケーションでクラスタ名を変更する必要がなくなります。

OracleAS Forms Services アプリケーションは、次の方法で Oracle Reports をコールできます。

- `RUN_REPORT_OBJECT` を使用します。Reports Server 名ではなく Reports Server クラスタ名がコールに指定されている場合は、OracleAS Forms Services `default.env` ファイルの環境変数 `REPORTS_SERVERMAP` を設定する必要があります。詳細は、『Oracle Application Server Forms Services 利用ガイド』を参照してください。
- `WEB.SHOW_DOCUMENT` を使用します。この場合、リクエストは `rwservlet` に転送されません。Reports Server 名ではなく Reports Server クラスタ名がコールに指定されている場合は、`rwservlet.properties` ファイルの `REPORTS_SERVERMAP` パラメータを設定する必要があります。次に例を示します。

```
REPORTS_SERVERMAP=cluster:repserver
```

ここで、

`cluster` は、以前のリリース (Oracle Reports 9i および 10g (9.0.4)) で使用されていた Reports Server クラスタ名です。

`repserver` は、Oracle Reports 10g リリース 2 (10.1.2) の Reports Server 名です。

REPORTS_SERVERMAP が `rwervlet.properties` に設定されているときは、OracleAS Forms Services アプリケーションに含まれるクラスタへのリクエストは、すべて `repserver` にリダイレクトされます。

3.4.21 代替の Reports Servlet 構成ファイルの指定

デフォルトでは、Reports Servlet は構成ファイルとして `rwervlet.properties` を使用します。レポートが同じ Oracle Application Server にインストールされた複数の OC4J インスタンスにあるときに、それらを異なる構成ファイルを使用して実行する場合は、`opmn.xml` に次の `start` パラメータを追加します。

```
-DServletPropFile =your_servlet_properties_file
```

次に例を示します。

```
<process-type id="OC4J_BI_Forms" module-id="OC4J">
...
<category id="start-parameters">
  <data id="java-options" value="-server -Djava.security.policy
    =/private/oracle/FRHome_1/j2ee/OC4J_BI_Forms/config/java2.policy
    -Djava.awt.headless=true -Xmx512M" -DServletPropFile=servlet.prop/>
  <data id="oc4j-options" value="-properties -userThreads "/>
</category>
```

別の方法として、Oracle Reports の `web.xml` ファイルを編集し、`servlet.properties` ファイル名を渡す `initparam` を追加することもできます。

次に例を示します。

```
<servlet>
  <servlet-name>Reports servlet - /reports/rwervlet</servlet-name>
  <servlet-class>oracle.reports.rwclient.RWClient</servlet-class>
  <load-on-startup>1</load-on-startup>
  <init-param>
    <param-name>ServletPropFile</param-name>
    <param-value>myservlet.prop</param-value>
  </init-param>
</servlet>
```

3.5 URL エンジンの構成

Reports Server は、URL のコンテンツの取得および配布を行う URL エンジンを搭載していません。URL エンジンを使用すると、Reports Server の強力なスケジューリングおよび配布の機能を利用して、任意の公開 URL のコンテンツを電子メール、OracleAS Portal、WebDAV などの宛先に配布できます。Reports Server の宛先はプラグイン可能なので、URL コンテンツに対するユーザー独自のカスタム宛先を追加することもできます。

さらに、URL エンジンを Reports Server の Event-Driven Publishing API と組み合わせて使用すれば、コンテンツの配布をデータベース・イベントでトリガーできます。たとえば、従業員経費データのテーブルに格納されたデータを高品位の Web で公開するための JSP レポートを作成したとします。この場合、URL エンジンと Event-Driven Publishing API を使用して、新規または更新した従業員経費データが経費アプリケーションによってテーブルに格納されるたびに、その JSP に電子メールで通知することができます。

URL エンジンが有効化されていない場合は、次の手順で有効にします。

1. URL エンジンの **engine** 要素をサーバー構成ファイルに追加します。たとえば、次のようにエンジン要素を入力します。

```
<engine id="rwURLEng"
      class="oracle.reports.engine.URLEngineImpl"
      initEngine="1"
      maxEngine="1"
      minEngine="0"
      engLife="50"
      maxIdle="30"
      callbackTimeOut="60000"
/>
```

2. 適切なジョブ・タイプを URL エンジンと関連付ける **job** 要素をサーバー構成ファイルに追加します。たとえば、次のように job 要素を入力します。

```
<job jobType="rwurl" engineId="rwURLEng"/>
```

3. Reports Server を停止し、再起動します。

注意： これらの新しい要素を含む Reports Server を再起動する場合は、Reports Server のステータス・メッセージ・ボックスでエンジン数の増加を確認できます。前述の例では、Reports Server を再起動するとエンジンの数が1つ増加します (initEngine の値)。

URL エンジンへのリクエスト送信の詳細は、[第 13 章「レポート・リクエストの実行」](#)を参照してください。

3.6 プロキシ情報の入力

OracleAS Reports Services のいくつかの機能では、ファイアウォール経由での情報の送受信をサポートしています。たとえば、URL エンジン、XML データソース、テキスト・データソースおよびメール宛先の機能は、すべてファイアウォール経由で情報を送受信します。これらの機能が適切に動作するために、Reports Server は特定のプロキシ情報を必要とします。処理を簡素化するために、必要なプロキシ情報を 1 箇所に格納し、Reports Server 構成ファイルからそれを指定するようにします。Reports Server にプロキシ情報を構成する手順は、次のとおりです。

1. **pluginParam** 要素をサーバー構成ファイルに追加し、プロキシ情報ファイル (proxyinfo.xml など) を指定するようにします。たとえば、次のように pluginParam 要素を入力します。

```
<pluginParam name="proxy" type="file">proxyinfo.xml</pluginParam>
```

注意： オプションでプロキシ情報ファイルのパスを指定できます。デフォルトでは、このファイルは `ORACLE_HOME/reports/conf` にあります。

- 構成に必要なプロキシ値でプロキシ情報ファイルを更新します。たとえば、次のように proxyinfo.xml に入力します。

```
<proxyInfo>
  <proxyServers>
    <proxyServer name="xyz.abc.com" port="80" protocol="http"/>
    <proxyServer name="www-proxy1.xyz.abc.com" port="80" protocol="ftp"/>
    <proxyServer name="www-prox21.xyz.abc.com" port="80" protocol="https"/>
  </proxyServers>
  <bypassProxy>
    <domain>*.abc.com</domain>
  </bypassProxy>
</proxyInfo>
```

注意： 詳細は、デフォルトのプロキシ情報ファイル（`ORACLE_HOME/reports/conf/proxyinfo.xml`）を参照してください。

3.7 Oracle Process Manager and Notification Server および Oracle Enterprise Manager 10g による Reports Server の構成

Reports Server を起動、シャットダウン、監視、管理する方法としては、Oracle Process Manager and Notification Server (OPMN) および Oracle Enterprise Manager 10g の使用をお勧めします。OPMN は、Reports Server を含む Oracle Application Server コンポーネントの初期化、維持、シャットダウンを一元化したメカニズムを提供します。Oracle Application Server に付属の Oracle Enterprise Manager 10g は、OracleAS Reports Services に対して管理サービスと監視サービスを提供します。Oracle Enterprise Manager 10g を使用して Reports Server を便利に監視することができます。また、なんらかの原因でプロセスが失敗した場合、OPMN が Reports Server を自動的に再起動します。

Reports Server は、Oracle Application Server のインストール中に自動的に OPMN で構成され、Oracle Enterprise Manager 10g に登録されます。Oracle Application Server のインストール後に Reports Server を追加する場合は、新しいサーバーを次の 2 箇所に登録する必要があります。

- Oracle Process Manager and Notification Server の `opmn.xml` ファイル
- Oracle Enterprise Manager 10g の `targets.xml` ファイル

`opmn.xml` と `targets.xml` の両方に新しい Reports Server を登録するには、次のコマンドラインを実行します。

UNIX の場合：

```
ORACLE_HOME/bin/addNewServerTarget.sh reports_server_name
```

Windows の場合：

```
ORACLE_HOME\bin\addNewServerTarget.bat reports_server_name
```

注意： このコマンドラインの実行後は、OPMN 構成ファイルを再ロードして変更を有効にしてください（`opmnctl reload` などを使用）。

3.7.1 opmn.xml

コンポーネントは、OPMN によって、`ORACLE_HOME/opmn/conf` にある `opmn.xml` ファイルに構成されます。次の各項では、OPMN を使用して次のコンポーネントを構成する方法を説明します。

- モジュールの指定
- スタンドアロン Reports Server の指定
- インプロセス Reports Server の指定
- Oracle Reports ブリッジの指定
- COS ネーミング・サービスの指定 (オプション)

関連項目： OPMN の構成および `opmn.xml` の内容の詳細は、次のドキュメントを参照してください。

- 『Oracle Application Server 管理者ガイド』

3.7.1.1 モジュールの指定

`module` タグは、デフォルトでは `opmn.xml` に含まれ、特定のモジュールのロードを OPMN に伝えます。Reports Server の場合は、OracleAS Reports Services モジュールをロードする必要があります。このモジュールは、デフォルトでは、次の情報によって `opmn.xml` にロードされます。

```
<module path="/private/oraclehome/opmn/lib/libopmnreports">
  <module-id id="ReportsServices"/>
</module>
```

3.7.1.2 スタンドアロン Reports Server の指定

スタンドアロン Reports Server の場合、Reports Server は専用コンポーネントで実行されます。したがって、OPMN によってサーバーを制御するには、Reports Server の個別コンポーネントを指定する必要があります。次に例を示します。

```
<ias-component id="<RSName>" status="enabled" id-matching="false">
  <process-type id="ReportsServer" module-id="ReportsServices">
    <process-set id="<RSName>" restart-on-death="true" numprocs="1">
      <environment>
        <variable id="PATH" value="your_shell_path"/>
      </environment>
      <module-data>
        <category id="general-parameters">
          <data id="batch" value="yes"/>
        </category>
        <category id="restart-parameters">
          <data id="reverseping-timeout" value="120"/>
        </category>
      </module-data>
      <dependencies>
        <OID infrastructure="true"/>
        <database infrastructure-key="portal"/>
        <managed-process ias-component="OC4J" process-type="OC4J_BI_Forms"
          process-set="default_island" autostart="true"/>
        <managed-process ias-component="HTTP_Server"
          process-type="HTTP_Server" process-set="HTTP_Server"
          autostart="true"/>
        <managed-process ias-component="WebCache" process type="WebCache"
          process_set="WebCache" autostart="true"/>
      </dependencies>
      <start timeout="600" retry="2"/>
      <stop timeout="120"/>
      <restart timeout="600"/>
    </process-set>
  </process-type>
</ias-component>
```



```

    <ping timeout="30" interval="30"/>
  </process-set>
</process-type>
</ias-component>

```

注意： 前述の例の timeout 値は、すべて秒単位です。

この Oracle Reports の指定で重要なセグメントは次のとおりです。

```
<ias-component id="RSName" ...>
```

このタグは、Reports Server の名前を指定します。この名前は、targets.xml にある、Reports Server の内部名と一致する必要があります。

関連項目： targets.xml の詳細は、第 19 章「OracleAS Reports Services の管理と監視」を参照してください。

```
<process-type id="ReportsServer" module-id="ReportsServices">
```

このタグは、指定された Reports Server のプロセスを定義し、これを OracleAS Reports Services のプロセス・モジュールに関連付けます。

```
<process-set id="RSName" restart-on-death="true" numprocs="1">
```

このタグは、指定の Reports Server のプロセスの特性を定義します。これは、Reports Server の失敗時に再起動するかどうかを指定します。さらに、このプロセス・セットについて起動される Reports Server の数も指定しますが、プロセスセット ID は単一の Reports Server 名を識別するため、その値は 1 である必要があります。

```
<variable id="PATH" value="your_shell_path"/>
```

1 番目のタグは、そのプロセスで使用される PATH 環境変数の値を指定します。この変数の設定は、開始スクリプトが uname を検出するために必要です。この environment 要素は、Microsoft Windows プラットフォームでは不要です。

```
<category id="general-parameters">
  <data id="batch" value="yes"/>
</category>
```

このタグのグループは、プロセスに共通のデータ（パラメータ）をすべて集めます。この例では、BATCH パラメータが Reports Server に送信されるように指定する方法を示しています。batch=yes|no は、Reports Server の起動および停止コマンドのオプションです。構成されない場合、このオプションは Reports Server に渡されません。

```
<category id="restart-parameters">
  <data id="reverseping-timeout" value="120"/>
</category>
```

このタグのグループは、再起動パラメータ・カテゴリを指定します。このカテゴリでは、プロセスが失敗したかどうか、プロセスを再起動するかどうかを検出するときに使用するパラメータを定義します。指定された reverseping-timeout 期間内に通知が受信されなかった場合、プロセスは失敗したと見なされ、再起動されます。

```
<dependencies>
```

このタグは、Reports Server が依存するコンポーネントのリストの範囲を定めます。たとえば、Reports Server は通常、Oracle HTTP Server と Oracle Application Server Containers for J2EE に特に依存しています。

OPMN は、依存性を使用してプロセスを開始するかどうかを決定します。module-data ブロックや environment ブロックと同様に、dependencies ブロックも opmn.xml 内の複数の要素に対して定義できます。OPMN は、プロセス・セット・レベル以上で定義されたすべての依存性を含む集計依存性リストをプロセス・セット・レベルで作成できます。重複する依存性が複数のレベルで定義されている場合、その依存性に対する重複チェックは、プロセスの開始前に行われます。

OPMN には、主に 2 つのタイプの依存性があります。外部依存性と内部依存性です。外部依存性は、OPMN で管理されないリソースです（データベース、Oracle Internet Directory、OracleAS Single Sign-On）。外部リソースに対しては、外部プログラムがリソースのチェックを実行します。内部依存性は、OPMN で管理されるプロセスです。これには、リモート OPMN インスタンスで管理されるプロセスが含まれる場合もあります。内部依存性は、managed-process タグによってリストに表示されます。

OPMN では、依存性の状態のキャッシュが維持されます。このキャッシュには、各依存性の最後に確認された状態と、その確認の日時が格納されます。同じ属性を持つ各依存性について単一のキャッシュ・エントリが存在します。これは、その依存性が複数の場所で（つまり、異なる複数のプロセス・セットに対して）指定されている場合も同様です。各依存性のキャッシュ・タイムアウト・パラメータを使用して、その依存性の状態をキャッシュ内で使用する期間を指定できます。同様に、各依存性の汎用タイムアウト・パラメータは、OPMN がステータス更新を待つ時間を指定します。この時間を過ぎると、OPMN は依存性チェックを中止し、プロセスが開始されます。

依存性のチェックは、各依存性の宣言の順に行われます。この依存性のリストの横断が終了するのは、チェックのシーケンスすべてが正常に完了した場合（リソースが使用可能な場合）か、いずれかのチェックが失敗した場合（リソースが使用不能の場合またはチェックがタイムアウトになった場合）です。

次の例のタグは、Reports Server の依存性の典型的なリストを表します。

```
<OID infrastructure="true"/>
<database infrastructure-key="portal"/>
<managed-process ias-component="OC4J" process-type="OC4J_BI_Forms"
  process-set="default_island" autostart="true"/>
<managed-process ias-component="HTTP_Server"
  process-type="HTTP_Server" process-set="HTTP_Server"
  autostart="true"/>
<managed-process ias-component="WebCache"
  process-type="WebCache" process-set="WebCache"
  autostart="true"/>
```

OID タグは、Reports Server が、この Oracle Application Server インストールのデフォルトの Oracle Internet Directory インスタンスを使用することを表します。

database タグは、Reports Server が使用する OracleAS Portal インスタンスを示します。

1 番目の managed-process タグは、Reports Server が使用する Oracle Application Server Containers for J2EE インスタンスを指定します。2 番目の managed-process タグは、Oracle HTTP Server インスタンスを表します。

関連項目： opmn.xml とその内容の詳細は、次のドキュメントを参照してください。

- 『Oracle Application Server 管理者ガイド』

3.7.1.3 インプロセス Reports Server の指定

インプロセス・サーバーの場合、Reports Server は OC4J コンポーネント内で実行されます。インプロセス・サーバーを使用する場合は、OC4J コンポーネント内で Reports Server データを指定する必要があります。次に例を示します。

```
<ias-component id="OC4J">
  <process-type id="OC4J_BI_Forms">
    <module-data>
      <category id="urlping-parameters">
        <data id="/reports/rwservlet/pingserver?start=auto" value="200"/>
      </category>
    </module-data>
  </process-type>
</ias-component>
```

この指定で重要なセグメントは次のとおりです。

- category 指定

```
<category id="urlping-parameters">
```

ここで、

urlping-parameters は、OC4J モジュールによって ping されるすべての URL を識別するカテゴリです。ping に使用するプロトコルは AJPv1.3 です。

- data 指定

```
<data id="/reports/rwservlet/pingserver?start=auto" value="200"/>
```

ここで、

/reports/rwservlet/pingserver?start=auto は、OC4J モジュールによって ping される URL です。インプロセス・サーバーのコンテキストでこの URL を ping すると、Reports Server アプリケーションが応答するかどうかを OPMN で判別できます。応答しない場合、OPMN は対応する OC4J プロセスを再起動します。

注意： Reports Servlet に対して pingserver コマンドを発行すると、インプロセス・サーバーの起動が試行されます。Reports Server の設定が正しくない場合は、エンジンの起動に失敗します。結果として、Reports Server はデッドロックとなり、成功のステータスが OPMN に返信されません。OPMN はこれを OC4J_BI_FORMS インスタンスの障害と解釈し、このインスタンスの再起動を試行します。この状況から回復するには、Reports Server の設定をチェックして、エンジンを適切に起動できることを確認します。詳細は、[第 3.9 項「レポート配布の最適化」](#)を参照してください。

value="200" は、ping リクエストに対するレスポンスで期待される有効な HTTP コード (200) を指定します。レスポンス HTTP コードがここで構成される値と一致すると、OPMN はアプリケーションが正常に動作して応答していると判断します。一致しない場合は、OPMN が OC4J プロセスを再起動します。

3.7.1.4 Oracle Reports ブリッジの指定

Oracle Reports ブリッジは、それ専用のコンポーネント内で実行されます。したがって、OPMN によってブリッジを制御するには、Oracle Reports ブリッジに対する固有の `ias-component` タグを指定する必要があります。

トラブルシューティングの例と解決策については、第 D.8 項「Oracle Reports ブリッジの問題の診断」を参照してください。

次は、最小のブリッジ構成と完全なブリッジ構成の例です。

例 3-1 Oracle Reports ブリッジの最小構成

```
<ias-component id="your_bridge_name" status="enabled" id-matching="false"
  xmlns="http://www.oracle.com/ias-instance">
  <process-type id="ReportsBridge" module-id="ReportsBridgeServices">
    <process-set id="your_bridge_name" restart-on-death="true" numprocs="1">
      <environment>
        <variable id="PATH" value="your_oracle_home_directory/jdk/jre/bin"
          append="true"/>
        <variable id="PATH" value="your_shell_path" append="true"/>
        <variable id="CLASSPATH" value="your_oracle_home_directory
          /jlib/zrclient.jar" append="true"/>
        <variable id="CLASSPATH" value="your_oracle_home_directory
          /reports/jlib/rwrun.jar" append="true"/>
      </environment>
    </process-set>
  </process-type>
</ias-component>
```

例 3-2 Oracle Reports ブリッジの完全構成

```
<ias-component id="your_bridge_name" status="enabled" id-matching="false">
  <process-type id="ReportsBridge" module-id="ReportsBridgeServices">
    <process-set id="your_bridge_name" restart-on-death="true" numprocs="1">
      <environment>
        <variable id="PATH" value="your_shell_path" append="true"/>
        <variable id="CLASSPATH" value="your_oracle_home_directory
          /jlib/zrclient.jar" append="true"/>
        <variable id="CLASSPATH" value="your_oracle_home_directory
          /reports/jlib/rwrun.jar" append="true"/>
      </environment>
      <module-data>
        <category id="restart-parameters">
          <data id="reverseping-timeout" value="120"/>
        </category>
        <category id="start-parameters">
          <data id="jvm-options" value="-Xms128mb -Xmx256mb"/>
          <data id="bridge-options" value="start_options_if_any"/>
        </category>
        <category id="stop-parameters">
          <data id="jvm-options" value="-Xms128mb -Xmx256mb"/>
          <data id="bridge-options" value="stop_options_if_any"/>
        </category>
      </module-data>
      <start timeout="120" retry="3"/>
      <stop timeout="120"/>
      <restart timeout="120" retry="0"/>
    </process-set>
  </process-type>
</ias-component>
```

3.7.1.5 COS ネーミング・サービスの指定

デフォルトでは、Oracle Reports は Reports Server の検出にビルトインのブロードキャスト・メカニズムを使用します。ただし、次の例のように、ビルトインのブロードキャスト・メカニズムが使用環境に適さない場合は、Oracle Reports クライアントは、Reports Server の検出に Common Object Service (COS) ネーミング・サービスを使用して、レポート・リクエストを送信できます。

- VPN を使用してネットワークに接続しているマシン上に Oracle Reports がインストールされている場合
- ネットワーク上のブロードキャスト・トラフィックを回避する必要がある場合

詳細は、第 1.4.2 項「COS ネーミング・サービスを使用したサーバーの検出」を参照してください。

OPMN を使用して COS ネーミング・サービスを制御するには、次のように、opmn.xml ファイルに ias-component カスタム・タグを含める必要があります。

```
<ias-component id="namingservice">
  <process-type id="namingservice" module-id="CUSTOM">
    <environment>
      <variable id="PATH" value="ORACLE_HOME%jdk%bin"/>
    </environment>
    <process-set id="namingservice" numprocs="1">
      <module-data>
        <category id="start-parameters">
          <data id="start-executable" value="ORACLE_HOME%jdk%bin%orbd"/>
          <data id="start-args" value="-ORBInitialPort port"/>
        </category>
      </module-data>
    </process-set>
  </process-type>
</ias-component>
```

ここで、

ORACLE_HOME は、使用している Oracle ホーム・ディレクトリです。

port は、COS ネーミング・サービスを起動するポートです。第 3.3.1.3 項「namingService」の説明のように、このポートは rwnetwork.conf ファイルで指定する必要があります。

さらに、Reports Server を起動する前に OPMN で COS ネーミング・サービスを起動するようにするには、opmn.xml に次の依存関係を含める必要があります。

```
<ias-component id="vin" status="enabled" id-matching="false">
  <process-type id="ReportsServer" module-id="ReportsServices">
    ...
    <managed-process ias-component="namingservice" process-type="namingservice"
      process-set="namingservice" autostart="true"/>
    ...
  </process-type>
</ias-component>
```

OPMN を使用して COS ネーミング・サービスを制御するには、次の手順を実行します。

1. Reports Server と OC4J_BI_Forms を停止します。
2. opmn.xml ファイルを編集して ias-component タグを追加します。次に例を示します。

```
...
<ias-component id="namingservice">
  <process-type id="namingservice" module-id="CUSTOM">
    <environment>
      <variable id="PATH" value="G:%FRHome_2%jdk%bin"/>
    </environment>
    <process-set id="namingservice" numprocs="1">
      <module-data>
```

```

    <category id="start-parameters">
      <data id="start-executable" value="G:¥FRHome_2¥jdk¥bin¥orbd"/>
      <data id="start-args" value="-ORBInitialPort 8988"/>
    </category>
  </module-data>
</process-set>
</process-type>
</ias-component>
...

```

3. Reports Server の ias-component タグに依存関係を追加します。次に例を示します。

```

...
<ias-component id="vin" status="enabled" id-matching="false">
  <process-type id="ReportsServer" module-id="ReportsServices">
    <process-set id="vin" restart-on-death="true" numprocs="1">
      <environment>
        <variable id="PATH" value="C:/OraHome/bin;C:/WINNT/system32;C:/WINNT"/>
      </environment>
      <module-data>
        <category id="general-parameters">
          <data id="batch" value="yes"/>
        </category>
        <category id="restart-parameters">
          <data id="reverseping-timeout" value="120"/>
        </category>
      </module-data>
      <dependencies>
        <OID infrastructure="true"/>
        <database infrastructure-key="portal"/>
        <managed-process ias-component="namingservice"
          process-type="namingservice" process-set="namingservice"
          autostart="true"/>
        <managed-process ias-component="OC4J" process-type="OC4J_BI_Forms"
          process-set="default_island" autostart="true"/>
        <managed-process ias-component="HTTP_Server" process-type="HTTP_Server"
          process-set="HTTP_Server" autostart="true"/>
      </dependencies>
      <stop timeout="120"/>
      <restart timeout="600"/>
      <ping timeout="30" interval="30"/>
    </process-set>
  </process-type>
</ias-component>
...

```

4. ORACLE_HOME/opmn/bin ディレクトリに移動します。

5. 次のいずれかのコマンドを実行します。

- OPMN が稼働している場合、opmn.xml ファイルへの変更を再ロードします：
opmnctl reload
- OPMN が稼働していない場合：opmnctl start

6. 次のいずれかのコマンドを使用して、COS ネーミング・サービスを起動します。

```
opmnctl startproc ias-component=namingservice
```

または

```
opmnctl startproc process-type=namingservice
```

7. 第 3.3.1.3 項「[namingService](#)」の説明のように、デフォルトのブロードキャスト・メカニズムのかわりに COS ネーミング・サービスを使用するように、rwnetwork.conf ファイルを変更します。

8. OC4J_BI_Forms と Reports Server を起動します。

COS ネーミング・サービスを停止するには、次のコマンドを使用します。

```
opmnctl stopproc ias-component=namingservice
```

COS ネーミング・サービスを再起動するには、次のコマンドを使用します。

```
opmnctl restartproc ias-component=namingservice
```

トラブルシューティング：Reports Server の起動時に COMM_FAILURE エラーが表示された場合、ネーミング・サービスが正しく起動されていないか、ポートが正しく指定されていません。このエラーを解決するには、ネーミング・サービスのプロセス orbd が稼動しているか確認してください。稼動していない場合は起動します。orbd プロセスが実行中の場合は、rwnetwork.conf ファイルの namingService で指定されたポートを確認してください。これは、orbd が起動されるポートと同じである必要があります。

opmnctl stopall コマンドを実行すると、Reports Server が正常に停止せず、OPMN によって強制終了される場合があります。これは、プロセスの停止中に OPMN が依存関係をチェックしないためです。Reports Server を停止する前に OPMN が COS ネーミング・サービスを停止すると、Reports Server は正常に停止しません。これは害がないため、無視してかまいません。

3.8 Oracle Reports と Oracle Workflow の通信の構成

Oracle Workflow は、ビジネス・プロセスの定義、自動化および統合をサポートする完全なビジネス・プロセス管理システムです。Oracle Workflow エンジンには Oracle データベースの一部です。詳細は、Oracle Technology Network (OTN) の Oracle Workflow 10g ページ (<http://www.oracle.com/technology/products/ias/workflow/index.html>) を参照してください。

Oracle Reports を Oracle Workflow に統合することで、Oracle Reports の実行を 1 つのアクティビティとして、ビジネス・プロセス内に配置することができます。たとえば、経費報告書がマネージャによって承認されると、Oracle Workflow によって即座にレポートが生成されます。次に、レポートの実行が完了すると Oracle Workflow エンジンに通知され、ビジネス・プロセス内の次のアクティビティがトリガーされます。

Oracle Workflow と Oracle Reports の統合の詳細は、OTN にあるホワイト・ペーパー『Integrating Oracle Workflow with Oracle Reports』 (<http://www.oracle.com/technology/products/reports/features/workflow>) を参照してください。このホワイト・ペーパーには、Oracle Workflow と Oracle Reports をインストールおよび構成して相互に通信する方法が説明されています。また、ビジネス・プロセスにレポート固有の機能を含める手順も説明されています。具体的には、Oracle Reports と Oracle Workflow のインストールおよび構成方法、レポートを実行する Oracle Workflow プロセスの定義方法、Oracle Workflow プロセスからのレポートの実行方法、トラブルシューティング情報などが説明されています。

3.9 レポート配布の最適化

低速または負荷の高いマシンでは、レポートを配布する前に、次の構成を実行することをお勧めします。

- ping のタイムアウト (OPMN 側) : ping のタイムアウトは、インプロセスの Reports Server (OC4J_BI_FORMS 内の) からのコールバックを待つ時間を決め、そのタイムアウト判定に使用される基準です。

デフォルトのタイムアウト期間は 150 です。この期間は、ping timeout、ping interval および number of retries の値から計算されます。デフォルト値は次のとおりです。

```
ping timeout = 30 seconds
ping interval = 20 seconds
number of retries - 3
```

注意： number of retries は、OPMN が OC4J への接続に成功し、OC4J プロセスから ONS 通知を定期的に受信する場合にのみ適用されます。

これらの値に基づいて、3 回の ping が 30 秒のタイムアウトと 20 秒の間隔で実行されま
す。初回の ping は、指定された ping interval の経過後に実行されます。つまり、
OC4J が OPMN によって起動されてから約 150 秒 (20 + 3 × 30 + 2 × 20) が経過後に、
プロセスが応答不能と判定され再起動されます。ただし、OPMN が OC4J に接続した後の
OC4J による ONS 通知の定期送信が遅すぎる場合は、30 秒のタイムアウトが適用されま
す。

ping のタイムアウトを構成するには、opmn.xml の次の要素に ping エントリを追加して、
マシンの負荷にあった十分なタイムアウト値を指定します。

```
<ias-component id="OC4J">
...
<process-type id="OC4J_BI_Forms" module-id="OC4J">
...
<restart timeout="720" retry="2" />
...
<ping timeout="110" interval="30" />
...
```

また、opmn.xml の要素を削除またはコメントアウトすることで、URL ping を無効化す
ることもできます。

```
<category id="urlping-parameters">
  <data id="/reports/rwservlet/pingserver?start=auto" value="200" />
</category>
```

構成が終わったら、OC4J_BI_Forms を再起動します。

- Report Server の起動または再起動のタイムアウト (OPMN.xml) : 起動または再起動のタイ
ムアウトは、Reports Server プロセス・タイプ (opmn.xml の process-type
id="ReportsServer") の起動または再起動を待つ時間を決め、そのタイムアウト判定
に使用される基準です。

デフォルトのタイムアウト期間は 600 です。デフォルト値は次のとおりです。

```
<start timeout="600" retry="2"/>
<restart timeout="600"/>
```

負荷の高いマシンでは、OPMN がすべての Reports Server の起動を試みたときに、一部の
Reports Server の起動アクティビティが完了せず、それらの Reports Server で起動タイム
アウトが生じる場合があります。また、Reports Server は、server_name.config ファ
イルの engine 要素の initengine プロパティで指定された数のエンジンを起動する必要
があります。負荷の高いマシンでは、これらのエンジン・プロセスの起動に時間がかかる
場合があります。Reports Server プロセスの起動または再起動プロパティのチューニング
と並行して、次に説明する server_name.config の callbackTimeout プロパティも
チューニングしてください。

- コールバックのタイムアウト (Reports Server 側) : コールバックのタイムアウトは、
Reports Server がエンジンからのレスポンスを待つ時間を決め、そのタイムアウト判定に
使用される基準です。この値は server_name.config ファイルで指定できます。タイム
アウト期間はミリ秒単位で指定します。

次に例を示します。

```
<engine id="rwEng" class="oracle.reports.engine.EngineImpl" initEngine="1"
maxEngine="1" minEngine="0"
engLife="50" maxIdle="30" callbackTimeOut="90000">
```

注意： マシンの速度が非常に遅い場合は、コールバックのタイムアウト値を
増やしてください。

3.10 UNIX での DISPLAY およびプリンタ依存性の解消

UNIX の場合、Oracle Reports 10g (9.0.4) よりも前のリリースでは、イメージの作成やピクセル解像度の取得のためにウィンドウ表示システムの表示面を使用するには、DISPLAY 環境変数を設定する必要がありました。Oracle Reports 10g では、この依存性は解消されています。

また、以前のリリースでは、フォントについて UNIX 上で有効なプリンタが必要でした。有効なプリンタがない場合、OracleAS Reports Services ではスクリーン・フォントが使用され、その際にも DISPLAY 環境変数の設定が必要でした。今回のリリースでは、OracleAS Reports Services に組み込まれているデフォルトの画面プリンタ・サーフェス (ScreenPrinter) は、プリンタが使用できない場合、画面またはプリンタをエミュレートしてフォントを入手します。したがって、UNIX 上にプリンタが存在している必要がなくなりました。

デフォルトでは、環境変数 REPORTS_DEFAULT_DISPLAY は YES に設定されています。この指定によって OracleAS Reports Services は次のように設定されます。

- DISPLAY 環境変数への依存性の解消 (UNIX のみ)
- イメージの画面解像度とフォント情報のための ScreenPrinter の使用 (UNIX のみ)
- 拡張イメージングのサポートの有効化 (全プラットフォーム共通)

以前のリリースの DISPLAY 環境変数の依存性に戻すには、REPORTS_DEFAULT_DISPLAY=NO を設定します。

関連項目： 第 B.1.39 項 「REPORTS_DEFAULT_DISPLAY」

3.10.1 ScreenPrinter

PostScript プリンタ・ドライバ screenprinter.ppd は、イメージの画面解像度を実現し、フォント情報を指定します。このドライバは、uiscreenprint.txt の 1 番目のエントリです。このファイルは次の場所にあります (UNIX のみ)。

```
uiscreenprint.txt : ORACLE_HOME/guicommon/tk/admin
screenprinter.ppd : ORACLE_HOME/guicommon/tk/admin/PPD
```

ScreenPrinter は次の目的で使用されます。

- REPORTS_DEFAULT_DISPLAY=YES のときの画面解像度
- プリンタ依存性の解消

レポート出力の生成時に使用できる有効なプリンタ・キューがない場合 (TK_PRINTER、ORACLE_PRINTER、PRINTER または uiprint.txt で見つからない場合) は、screenprinter.ppd に基づく画面が作成され、それを使用してフォント情報が取得されます。新しいフォントを追加するには、screenprinter.ppd の Fonts セクションを変更します。解像度を変更するには、DefaultResolution フィールドを変更します (DefaultResolution は 96 です)。

注意： 新しいフォントを追加する場合は、新しい AFM メトリック・ファイルが AFM ディレクトリにあることを確認します。

UNIX のフォント検索アルゴリズムは次のとおりです。

```
if a valid printer available then
  look up font information from the printer
else
  create a screenPrinter surface
  look up font information from ScreenPrinter
if ScreenPrinter creation fails then
  REP-1800 : Formatter Error if REPORTS_DEFAULT_DISPLAY is set
else
  use Screen Fonts
```

注意： 中国語など、いくつかのマルチバイト言語では、スクリーン・フォントを使用する場合があります。ただし、この場合、レポートを実行するための DISPLAY 変数の設定が必要です。

DISPLAY に戻し、スクリーン・フォント（従来のフォント検索アルゴリズム）を使用する手順は次のとおりです。

- REPORTS_DEFAULT_DISPLAY=NO に設定します。
 - uiscreenprint.txt ファイルの screenprinter.ppd エントリを削除します。
-

関連項目：

- 第4章「Oracle Reports でのフォントの管理」
- 第5章「UNIX での Oracle Reports による印刷」

3.10.2 拡張イメージングのサポート

画質は、レポート（特に Web レポート）の外観全体に大きな影響を与えます。業務の必要性に応じて、レポート出力で様々なイメージ形式が必要になる場合があります。たとえば、航空会社の Web レポートでは、GIF イメージより高画質な JPEG イメージや PNG イメージが適している場合があります。一方、Web ポータルを構築する場合は、より小さいサイズでダウンロードが高速な GIF イメージが適している場合もあります。同様に、このような様々な形式のイメージをユーザーのレポートにインポートする場合があります。

出力の形式によって、様々な形式のイメージを選択することができます。

表 3-31 出力タイプ別のイメージ形式オプション

レポート出力	選択可能なイメージ形式
HTML、HTMLCSS	PNG、JPEG、JPG、GIF
PDF	PNG、JPEG、JPG、GIF
RTF	PNG、JPEG、JPG、BMP

注意： イメージ形式の選択では、画質とサイズを考慮する必要があります。通常は、イメージ形式の画質が高いほど、サイズが大きくなります。たとえば、PNG や JPEG は GIF より高画質ですが、より多くの記憶領域を必要とします。

拡張イメージングを有効にするには、REPORTS_DEFAULT_DISPLAY 環境変数を YES に設定します。REPORTS_OUTPUTIMAGEFORMAT 環境変数で、デフォルトのイメージ・タイプを選択できます。OUTPUTIMAGEFORMAT コマンドライン・キーワードを使用すると、イメージに関するデフォルトの選択を無効にできます。次に例を示します。

```
rwclient server=my_rep_server report=images.rdf destype=file desformat=html
desname=images.html userid=scott/tiger outputimageformat=PNG
```

拡張イメージングを有効にすると、これらの同じ形式のイメージをレポートにインポートすることもできます。

使用上の注意

- UNIX のみ: 拡張イメージングを有効にすることによって、HTML や HTMLCSS 出力で従来の Computer Graphics Metafile (CGM) 形式や Oracle Graphics Data (OGD) 形式が使用できなくなります。これらの形式を入力ソースに使用する必要がある場合は、REPORTS_DEFAULT_DISPLAY=NO に設定する必要があります。この制限は、Windows プラットフォームには適用されません。
- JPEG イメージ付きのレポート (REPORTS_OUTPUTIMAGEFORMAT=JPEG) を RTF 出力に対して実行すると、イメージのサイズとは直接比例することなく、RTF ファイルのサイズが増加します。その理由は、バイナリ・イメージ・ストリームが最初に HEX キャラクタに変換されてから RTF に書き込まれるからです。この変換によってファイルのサイズは増加します。これは、RTF 仕様に沿ったものであり、予測される動作です。ただし、JPEG 付きの RTF ファイルは、BMP イメージ付きの RTF ファイルよりもサイズが小さくなります。

関連項目:

- [第 A.3.63 項「OUTPUTIMAGEFORMAT」](#)
- [第 B.1.39 項「REPORTS_DEFAULT_DISPLAY」](#)
- [第 B.1.52 項「REPORTS_OUTPUTIMAGEFORMAT」](#)

Oracle Reports でのフォントの管理

この章では、Oracle Reports で使用するフォントについて説明します。この章の主な項は、次のとおりです。

- フォントの使用
- フォントの追加
- フォント構成ファイル
- フォント・エイリアシング
- フォントに関する問題のトラブルシューティング
- フォントのタイプ

4.1 フォントの使用

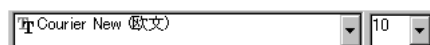
Oracle Reports では、次の場合にフォントが重要になります。

- ビルドの実行時 (Reports Builder において)
- 実行時 (レポート出力内)
- Reports Builder のユーザー・インタフェース内

4.1.1 Reports Builder のフォント

Reports Builder では、システムで使用可能なフォントのリストがフォントの選択ボックスに表示されます。

図 4-1 Reports Builder のフォント・リスト



Windows の場合、フォント・リストは、システムにインストールされているフォントおよび現在のデフォルト・プリンタで使用可能なフォントから作成されます。フォント名の前の小さなプリンタ・アイコンは、プリンタ・フォントであることを示します。TrueType フォントの場合には、TTF アイコンが表示されます。

UNIX の場合、フォント・リストは、アプリケーションが実行されている X サーバーのディスプレイで利用可能なフォントを問い合わせることによって作成されます。このコマンドは、UNIX の `xlsfonts` コマンドに似ています。これにより、X サーバー・ディスプレイで使用可能なすべてのフォントがリストされます。このフォント・リストから、キャラクタ・セットに適合する有効なスタイル、太さ、幅、サイズおよびエンコード特性を持つ使用可能なフォントのリストが Reports Builder によって生成されます。キャラクタ・セットは、`NLS_LANG` 環境変数によって決まります。Reports Builder には、ツールキット・リソース・ファイル内で `Tk2Motif*fontMapCs` で指定されている場合を除き、文字コード体系が `iso8859-1` のフォントのみ組み込まれます。`Tk2Motif*fontMapCs` の詳細は、第 4.3 項「フォント構成ファイル」を参照してください。

4.1.2 レポート出力のフォント

レポートのフォーマット時、まずレイアウト・オブジェクトに関連付けられているフォントがフォント・エイリアス・ファイル (`uifont.ali`) 内に存在するかどうか確認されます (第 4.3 項「フォント構成ファイル」を参照)。フォント・エイリアス・ファイル内にエントリが見つかった場合は、元のフォントではなく、マップされているフォントが使用されます。次に、このマップされたフォントが、システムまたはプリンタで使用可能なフォントのリストにあるかどうか検索されます。フォントが見つからない場合は、かわりに使用できるフォントとして、同じキャラクタ・セットを持つ類似度が最も高いフォントが検索されます。

4.1.2.1 フォントの検索

Windows の場合はプリンタ・ドライバを使用するため、フォント検索メカニズムは単純です。プリンタ・ドライバには、必要に応じてフォントをシステムからアップロードする機能があります。Windows で実行される Oracle Reports の出力に含まれるフォントは次のいずれかになります。

- システム・フォント
- プリンタ・フォント

したがって、Oracle Reports で使用可能なフォントを検索するときは、プリンタ・フォントとシステム・フォントの両方が対象となります。

UNIX の場合、出力生成に使用できるフォントは次のいずれかです。

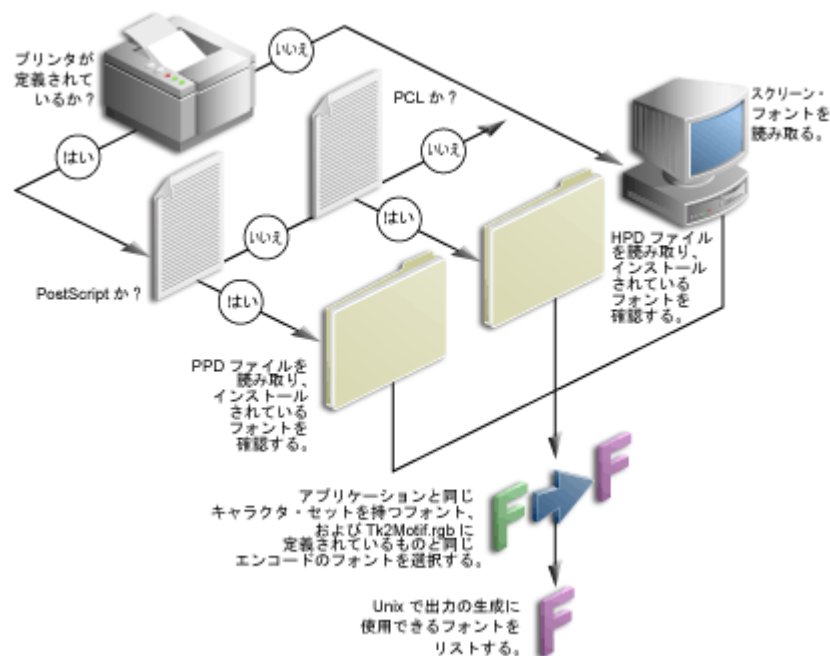
- プリンタで使用可能なフォント（PPD ファイルまたは TFM ファイルで定義されているフォント）
- プリンタが指定されていない場合は、ScreenPrinter（screenprinter.ppd）で使用可能なフォント

関連項目：

- [第 4.3 項「フォント構成ファイル」](#)
- [第 3.10.1 項「ScreenPrinter」](#)

4.1.2.1.1 **フォント検索アルゴリズム** 図 4-2 は、レポート出力の生成時に使用可能なフォントを決定するプロセスを示しています。

図 4-2 フォント検索のフロー



Oracle Reports において、出力（画面用、プリンタ用またはファイル用など）の生成に使用可能なフォントのリストを生成する手順は次のとおりです。

1. プリンタ構成ファイル uiprint.txt を検索し、アプリケーションに対して指定されているすべてのプリンタを調べます。プリンタが定義されていない場合や、使用できるプリンタがない場合は、ScreenPrinter を使用します。

関連項目：

- [第 4.3 項「フォント構成ファイル」](#)
- [第 3.10.1 項「ScreenPrinter」](#)

2. プリンタのタイプ、バージョンおよびプリンタ定義ファイルを取得します。
3. それらのプリンタがシステム内に存在しているかどうか確認します。
4. プリンタが存在する場合は、プリンタ定義ファイルをロードして、そのファイルに記述されている情報と、プリンタで使用可能なフォントに関連する情報を読み取ります。プリンタが見つからない場合は、ScreenPrinter を使用します。

5. AFM ファイルを検索します。ファイル名は、PPD ファイル内で指定されているフォント名と同じです。
6. AFM ファイルが見つかった場合は、そのファイルからすべての有効なキーワードを読み取り、キーワードが正しいかどうかを確認します。なんらかの矛盾がある場合は、そのキーワードに対してデフォルト値を使用します。
7. AFM ファイルが見つからない場合、そのフォントは使用不可能となります。

注意： PCL プリンタの場合と同様に、HP プリンタ定義ファイル (glue ファイル) およびそのファイル内で定義されているすべてのフォントもロードされます。フォントの検索時、HPD ファイル内でそのフォント名のエントリが検索され、そのフォントの TFM ファイルも見つかった場合はそのフォントが使用されます。それ以外の場合は、そのフォントとの類似度が最も高い使用可能なフォントが採用されます。

使用可能なフォントのリストが生成されたら、マップされたフォントをこのフォント・リスト内で検索し、テキストのサイズと太さを計算するために再び AFM を読み取ります。

フォントの置換

必要としているフォントが PPD ファイルに見つからない場合や AFM ファイルが見つからない場合、Oracle Reports では、一致規則に基づいて最も類似度の高いフォントが検索されます。たとえば、あるレポートが簡体字中国語の SimSun フォントを使用するように設計されており、このフォントに対応するマッピングが uifont.ali ファイルに見つからない場合は、[フォント検索アルゴリズム](#)で生成される使用可能なフォントのリストに SimSun フォントがあるかどうか検索されます。このフォントの名前がリストにない場合は、プリンタ定義ファイルで指定されているフォントのリスト内で、類似度が最も高いフォントが検索されます。

SimSun フォントのキャラクタ・セットは 850 です。このキャラクタ・セットに一致するフォントが見つからない場合は、アプリケーションが実行されている環境のキャラクタ・セットと同じキャラクタ・セットのフォントが検索されます。同様のキャラクタ・セットを持つフォントが複数見つかった場合は、要求されたフォントに最も類似度が高いフォントが、フォントの太さやスタイルなどに基づいて選択されます。同じパラメータのフォントが複数ある場合は、最初のフォントが元のフォントのかわりに使用されます。

フォントの一致規則

Oracle Reports では、一致するフォントを見つける際、同じキャラクタ・セットを持つ最も類似度が高いフォントが次の条件に従って検索されます。

fontface > fontsize > fontstyle > fontweight > fontwidth

フォント・フェースが一致しない場合はフォント・サイズが比較され、フォント・サイズが一致しない場合には続いてフォント・スタイルが比較されます。

フォント・サイズのみ一致してそれ以外は異なるフォントと、スタイル、太さ、幅が一致してフォント・サイズが異なるフォントがある場合は、フォント・サイズの一致するフォントが選択されます。

最終的に印刷される出力のフォントは、出力ファイル内のフォントとは関係なく、プリンタにインストールされているフォントにのみ依存することに注意してください。

例： レポートのレイアウト・オブジェクトに、サイズ 8、標準スタイル、標準の太さの Helvetica フォントと、Courier フォントのいずれかが関連付けられているとします。このレポートを PostScript-1 プリンタに対して実行して HTML 出力を生成する場合、フォントは次のように選択されます。

1. フォーマット時、Oracle Reports によって、指定されたフォントのマッピングが uifont.ali に存在するかどうか確認されます。uifont.ali の [Printer:PostScript1] セクションに、次のエントリがあるとします。
Helvetica.8.Plain.Medium.. = "Mkai-Medium" ..

これにより、Helvetica フォントにかわって Mkai-Medium フォントが検索されます。

- このマップされたフォントが、プリンタ定義ファイル (PPD) 内にあるかどうか検索されます。PPD ファイルの ***Font Information** セクションに、次のエントリがあるとします。

```
**Font Mkai-Medium: Standard "(001.004)" Standard ROM"
```

- これにより、対応する AFM ファイル (ファイル名は Mkai-Medium) が AFM ディレクトリ内で検索されます。この名前の AFM ファイルが見つからない場合は、サイズ 8、標準スタイル、標準の太さで、元のフォントと同じキャラクタ・セットの別のフォントが検索されます。

同じレポートが複数の異なる環境で実行されることもあるため、Oracle Reports では、元のフォントが使用できない場合、必ずそれに類似するフォントの検索が行われます。このアルゴリズムは、絶対に失敗しないというものではありません。レポートの作成者は、定義されているフォントを認識し、ユーザーがレポートを実行するプラットフォーム上で同じフォントが使用可能かどうかを常に考慮する必要があります。定義したフォントがランタイム環境で使用できない場合は、そのマシンで使用可能な別のフォントによって置換されます。この処理によって予想と異なる結果が生成される場合があります。たとえば、レポート出力に Wingding 文字などの特異な文字が出現したり、オブジェクトのフォーマットが正しく行われなかったことがあります。

この種の問題が発生した場合は、フォント・エイリアシングを使用して、Oracle Reports によるフォント置換を制御してください。詳細は、[第 4.4 項「フォント・エイリアシング」](#)を参照してください。

Oracle Reports には、PDF を除くすべての出力ファイルが前述のメカニズムに従って生成されます。PDF には、フォント・サブセットおよびフォント埋込みの機能があります。

関連項目： OracleAS Reports Services での PDF 機能および拡張機能の詳細は、[第 6 章「Oracle Reports での PDF の使用」](#)を参照してください。

PDF の一般的な情報は、Adobe PDF のマニュアルを参照してください。

印刷用出力は、Windows の場合はプリンタ・ドライバ、UNIX の場合はプリンタに基づいて生成されます。Windows では、出力の生成はプリンタ・ドライバによって処理されます。この場合、システム・フォントまたはプリンタ・フォントが使用されます。プリンタ上にないフォントについては、Windows API を介してシステムからプリンタにフォントが取得されます。

4.1.3 ユーザー・インタフェースのフォント

Reports Builder のユーザー・インタフェースのテキスト (ウィンドウのタイトル、メニュー項目、メッセージ・ボックス、データ・モデル・オブジェクト名など) には、現行の言語に対応するシステム・リソース・ファイルから取得したフォントが使用されます。これらのシステム・リソース・ファイルは、Oracle Reports のインストールレーションに含まれています。Oracle Reports では、uifont.ali の [rwbuilder] セクションでこれらのフォントのマッピングを指定できます。マップされているフォントが見つかった場合は、元のフォントのかわりにそのフォントが使用されます。それ以外の場合は、元のフォントが使用されます。

UNIX の場合、フォントは Tk2Motif.rgb の Tk2Motif*fontList で定義されます。定義されていないフォントについては、デフォルトのフォント (デフォルトのキャラクタ・セットに対して固定) がかわりに使用されます。デフォルトのシステム・フォントは、Tk2Motif.rgb に定義されているものでなくてもかまいません。定義されているフォントが、アプリケーションを実行する環境のキャラクタ・セットと一致しない場合は、前の項で説明したフォント検索アルゴリズムに従って、かわりの使用可能なフォントが使用されます。

Windows では、ウィンドウの外観と操作性を維持するために、Oracle Reports の多くの部分で Windows のシステム・パラメータから取得されたシステム・フォントが使用されています。非 Unicode 環境では、フォントはアイコン・オブジェクトから取得されます。このフォントを変更するには、「画面のプロパティ」→「デザイン」を選択します。ドロップダウン・リストから「アイコン」を選択し、フォント名とサイズを選択します。日本語 Unicode システムでは、フォントは MS ゴシックです。韓国語の場合は、MS Sans Serif です。簡体字、繁体字および香港の中国語では Arial です。それ以外の言語の場合は、Lucida Sans Unicode になります。

Windows ツールチップのフォントも、前述の手順に従ってアイコンのフォントを変更することによって変更できます。ただし、この変更は Reports Builder の一部には反映されません。これは、ツールチップの中にはフォントをリソース・ファイルから取得するものがあるためです。

Oracle Reports では、エイリアス・ファイルの [rwbuilder] セクションにエントリを作成することによって、Web ソース・ビューのフォントを選択します。この変更に必要なエントリでは、特定のフォントではなくキャラクタ・セットに対してエイリアスを作成するようにします。たとえば、NLS_LANG が UTF8 に設定されているときに Arial Unicode MS が使用されるようにするには、次のようなエントリを作成します。

```
....UTF8 = "Arial Unicode MS"....
```

詳細は、第 4.4 項「フォント・エイリアシング」を参照してください。

サポートされているスタイルは、標準、イタリック、斜線、下線、アウトライン、影、反転、打消し線および点滅です。

サポートされている太さは、超極細、極細、細、中細、標準、中太、太、極太および超極太です。

フォントの太さに Regular を指定しないでください。この太さはサポートされていないため、Reports Builder での表示結果が予期しないものになる可能性があります。

4.2 フォントの追加

Oracle Reports では、次の場合に使用するフォントを追加できます。

- ビルドの実行時 (Reports Builder において)
- 実行時 (出力内)
- ユーザー・インタフェース内

4.2.1 Reports Builder へのフォントの追加

特定のフォントを使用してレポートを作成するには、レポートを設計する際、そのフォントが Reports Builder のフォント選択ボックスで選択できる状態になっている必要があります。フォントがフォント選択ボックスに表示されるようにするには、システムまたは Reports Builder が実行されるディスプレイにそのフォントを追加します。この手順を実行する前に、オペレーティング・システムのマニュアルを参照して、フォントの追加について確認してください。

UNIX 上に Type1 フォントを追加するには：

1. フォント関連のファイルをベンダーから入手します。これらのファイルには、PFB ファイル、PFA ファイル、AFM ファイルなどがあります。
2. 使用可能な変換プログラムを使用して、バイナリの PFB ファイルを ASCII の PFA ファイルに変換します。一般に、このような変換プログラムは、t1ascii などのシェアウェアとして入手できます。
3. プラットフォームごとの指示に従って、フォントのインストール先のディレクトリに PFA ファイルをコピーします。
4. フォントのインストールを確認するために、xlsfonts -u というコマンドを入力します。このコマンドを実行すると、システムで使用可能なすべてのフォントが一覧表示されます。

Reflection X などの UNIX エミュレータを使用している場合は、xlsfonts コマンドを実行しても、システムにインストールされているフォントが表示されないことがあります。これは、エミュレータの構成時に指定されたフォント・パスまたはフォント・サーバーからフォントが取得されるためです。フォント・サーバーを使用している場合は、フォントのインストール後に、フォント・サーバー構成ファイルにフォント・ディレクトリを追加してフォント・サーバーを再起動してください。エミュレータ側では、フォントがインストールされているフォント・サーバーへのフォント・パスを指定します。このようにしても、xlsfonts の実行時にフォントが表示されない場合は、新しいフォント・ディレクトリが構成ファイル内のカタログの最初の要素であることを確認してください。

図 4-3 xlsfonts の出力例

```

端末エミュレータ
ウィンドウ(W)  編集(E)  オプション(O)  ヘルプ(H)
-adobe-times-medium-r-normal--17-120-100-100-p-84-iso8859-1
-adobe-times-medium-r-normal--18-180-75-75-p-94-iso8859-1
-adobe-times-medium-r-normal--18-180-75-75-p-94-iso8859-1
-adobe-times-medium-r-normal--20-140-100-100-p-96-iso8859-1
-adobe-times-medium-r-normal--20-140-100-100-p-96-iso8859-1
-adobe-times-medium-r-normal--24-240-75-75-p-124-iso8859-1
-adobe-times-medium-r-normal--24-240-75-75-p-124-iso8859-1
-adobe-times-medium-r-normal--25-180-100-100-p-125-iso8859-1
-adobe-times-medium-r-normal--25-180-100-100-p-125-iso8859-1
-adobe-times-medium-r-normal--34-240-100-100-p-170-iso8859-1
-adobe-times-medium-r-normal--34-240-100-100-p-170-iso8859-1
-adobe-times-medium-r-normal--8-80-75-75-p-44-iso8859-1
-adobe-times-medium-r-normal--8-80-75-75-p-44-iso8859-1
-adobe-utopia-bold-i-normal--0-0-0-0-p-0-iso8859-1
-adobe-utopia-bold-i-normal--0-0-0-0-p-0-iso8859-1
-adobe-utopia-bold-r-normal--0-0-0-0-p-0-iso8859-1
-adobe-utopia-bold-r-normal--0-0-0-0-p-0-iso8859-1
-adobe-utopia-medium-i-normal--0-0-0-0-p-0-iso8859-1
-adobe-utopia-medium-i-normal--0-0-0-0-p-0-iso8859-1
-adobe-utopia-medium-r-normal--0-0-0-0-p-0-iso8859-1
-adobe-utopia-medium-r-normal--0-0-0-0-p-0-iso8859-1
---

```

5. フォントがインストールされているフォント・サーバーまたはフォントがインストールされているディスプレイを指定したディスプレイ上で、Reports Builder を起動します。

4.2.2 レポート出力用の新しいフォントの追加

Oracle Reports で出力を生成する場合、プリンタ定義ファイルで指定されているフォントのみ使用されます。新しく追加したフォントを出力で使用するには、レポートの設計時にそのフォントをレイアウト・オブジェクトに割り当てることができるように、フォントを Reports Builder に追加します。詳細は、第 4.2.1 項「Reports Builder へのフォントの追加」を参照してください。

注意： Reports Builder で使用するフォントがランタイム・プラットフォームでは使用できないものである場合は、ランタイム・プラットフォーム上でそのフォントのエイリアスを作成してください。詳細は、第 4.4 項「フォント・エイリアシング」を参照してください。

フォント追加の処理は、Windows と UNIX で異なります。

- UNIX でのフォントの追加
- Windows でのフォントの追加

4.2.2.1 UNIX でのフォントの追加

PostScript フォントを追加するには：

1. 新しいフォントの AMF ファイルを、`ORACLE_HOME/guicommon/tk/admin/AFM` にコピーします。
2. 新しいフォントに対応する次のようなエントリを、プリンタ定義ファイル (PPD) の *Font 情報セクションに追加します。

```

*Font new_font_name Standard "(00.1001)" Standard ROM
*Font ...

```

PPD ファイルで指定されている `new_font_name` が AFM ファイル名と同じであることを確認します。これは、Oracle Reports では PPD ファイル内のフォント名に基づいてこのファイルが検索されるためです。また、AFM ファイル名に拡張子の `.afm` が含まれていないことを確認します。

たとえば、AFM ファイル名が CodedreineunBold であれば、PPD ファイルには *Font CodedreineunBold: Standard "(00.1001)" Standard ROM が指定されている必要があります。

- 必要であれば、エイリアス・ファイルに変更を加えて、このフォントへのマッピングを指定します。

レイアウト・オブジェクトに、新しいフォントと同じ名前のフォント名が関連付けられている場合、マッピングは必要ありません。レイアウト・オブジェクトに別のフォント名が使用されている場合に、元のフォントではなく新しいフォントが出力ファイルで使用されるようにするには、元のフォントを新しいフォントにマップします。

たとえば、レイアウト・オブジェクトのフォントが Helvetica の場合に、新しくインストールしたフォントを出力で使用するには、次の行を [Printer:PostScript1] セクションに追加します。

```
Helvetica = CodedreineunBold
```

uiprint.txt で別の PostScript レベルが指定されている場合は、このセクションが異なることに注意してください。詳細は、[第 4.4 項「フォント・エイリアシング」](#)を参照してください。

PCL フォントを追加するには：

新しいフォントを Oracle Reports で使用するには、プリンタの HPD ファイル（プリンタ定義ファイル）および TFM ファイルが必要です。HPD ファイルは、既存のファイルからコピーして作成できます。このファイルの内容が使用するプリンタに適合していることを確認してください。このファイル内で参照されているフォントは、プリンタで使用可能なものである必要があります。TFM ファイル（フォント）が、Oracle Reports のインストレーションで使用できない場合は、フォントおよびプリンタのサプライヤに問い合わせる必要があります。新しい TFM ファイルは、一意のフォント名を付けて HPD ファイルに追加する必要があります。

- HPD ファイルに、新しいフォントのエントリを追加する必要があります。たとえば、新しいフォントが Codedreineun である場合は、次のような行を追加します。

```
FONT= Codedreineun  
/tfm=9nb17035.tfm
```

- 対応する TFM ファイルを TFM ディレクトリにコピーします。

```
ORACLE_HOME/guicommon/tk/admin/TFM
```

- 必要であれば、[第 4.2.2.1 項「UNIX でのフォントの追加」](#)の PostScript プリンタ用の説明に従って、エイリアス・ファイルを変更します。マッピングを定義するセクションは [PCL] である必要があります。

4.2.2.2 Windows でのフォントの追加

Windows に新しいフォントを追加する場合は、オペレーティング・システムのマニュアルを参照して、フォントの追加について確認してください。新しいフォントのキャラクタ・セットが Reports Builder と互換性のあるものであれば、そのフォントがフォント選択ボックスに表示されます。

4.3 フォント構成ファイル

この項では、Oracle Reports のフォント構成に関連するすべてのファイルについて説明します。

- [uiprint.txt](#) (UNIX のみ)
- [screenprinter.ppd](#) (UNIX のみ)
- [uifont.ali](#)
- [PPD ファイルおよび AFM ファイル](#) (UNIX のみ)
- [HPD ファイルおよび TFM ファイル](#)
- [Tk2Motif.rgb](#) (UNIX のみ)

uiprint.txt (UNIX のみ)

このプリンタ構成ファイルには、アプリケーションに対してインストール済みのプリンタの一覧が、プリンタのタイプ、バージョン、プリンタ定義ファイル名とともに記述されています。実行時に使用可能なフォントのリストは、プリンタ定義ファイルから取得されます。プリンタが存在しない場合は、Oracle Reports のデフォルトのプリンタとして PostScript プリンタが選択され、default.ppd ファイルがプリンタ定義ファイルとして使用されます。

関連項目：

- 第 5.3.1 項「uiprint.txt ファイルの編集」
- 第 5.4.6 項「uiprint.txt」

例：

```
Printer: Printer_driver:Driver_specifying_language_and_level:Printer_
description:Printer_definition_file:
```

各行に 5 つのフィールドがあり、それぞれコロンの区切られています。

PCL プリンタを使用する場合、このエントリには HPD ファイルの名前が含まれます。

screenprinter.ppd (UNIX のみ)

screenprinter.ppd は、UNIX 上でプリンタが使用できない場合に使用されます。screenprinter.ppd は、ORACLE_HOME/guicommon/tk/admin/PPD にあります。

uifont.ali

このファイルには、実行時に他のフォントで置換え可能なフォントのマッピング情報が含まれます。詳細は、第 4.4 項「フォント・エイリアシング」を参照してください。

Oracle Reports では、新たに次の 3 つのセクションが uifont.ali ファイルに追加されています。

[PDF]: フォント・エイリアシングおよびマルチバイト言語のサポートに使用されます。

[PDF:Subset]: TrueType フォント・サブセットおよびマルチバイト言語のサポートに使用されます。

[PDF:Embed]: Type1 フォントの埋込みに使用されます。

注意： Oracle Reports では uifont.ali ファイルを解析してキーワード検索が実行されるため、これらのセクション名は変更しないでください。セクションの順序は自由に変更できます。

uifont.ali ファイルに関する一般的な規則を次に示します。

- フォント名やキャラクタ・セット名に複数の単語が含まれる場合、つまり空白が含まれている場合は二重引用符で囲みます。
- コメント行の場合は、行の先頭に # を使用します。
- 行を削除するかわりにコメント・アウトしておけば、その行を後で使用することができます。
- フォント・エイリアシングは、フォント名からフォント名、またはキャラクタ・セットから CID フォント (Adobe より) の場合にのみ可能です。
- フォント・サブセットは、TrueType フォントの場合にのみ可能です。
- フォント・サブセットにはフォント名が使用され、TrueType フォント・ファイル名を使用してサブセットが作成されます。

- フォント埋込みは、Type1 フォントの場合にのみ可能です。Type1 フォントには2つのファイルが使用されます。1つはメトリック・ファイルで、ファイル拡張子が .afm または .pfm のいずれかになります。もう1つは、ファイル拡張子が .pfb のバイナリ・ファイルです。

フォント埋込みにはフォント名が使用され、Type1 フォント・ファイル名を使用して埋込みが行われます (AFM ファイルと PFB ファイルの両方がこの順序で要求されます)。

詳細は、第 4.4 項「フォント・エイリアシング」を参照してください。

PPD ファイルおよび AFM ファイル (UNIX のみ)

PostScript Printer Definition (PPD) ファイルおよび Adobe Font Metrics (AFM) ファイルは、Adobe 社およびプリンタのベンダーによって提供されるものです。これらのファイルには、プリンタに関する情報が含まれます。Oracle Reports では、これらのファイルから、プリンタで使用可能なフォントに関する情報が他のパラメータとともに読み取られます。PPD ファイルに記述されているすべてのフォントについて、フォント名を基に対応する AFM ファイルが検索され、有効な AFM が存在するすべてのフォントがロードされます。

フォントの観点からは、プリンタ用の新しいフォントを追加する場合にこれらのファイルを変更し、変更が Oracle Reports に反映されるようにしてください。

例:

```
*% Font Information =====
*DefaultFont: Error
*Font AvantGarde-Demi: Standard "(001.001)" Standard
*Font AvantGarde-DemiOblique: Standard "(001.001)" Standard
*Font Courier: Standard "(001.004)" Standard
*Font Courier-Bold: Standard "(001.004)" Standard
```

AFM ファイルには、フォントの属性 (スタイル、太さ、幅、コード体系)、固定幅かプロポーショナルかの区別、および各文字の大きさなどの情報が含まれます。

Oracle Reports では、PPD ファイルからフォント名を検索した後、第 4.3.1 項「ファイルの検索」で説明している検索条件に従って、フォント名と同じ名前の AFM ファイルが検索されます。たとえば、AvantGarde-Demi: Standard が PPD ファイル内で見つかった場合は、AvantGarde-Demi という名前の AFM ファイルが AFM ディレクトリにあるかどうか検索されます。

AFM ファイルはフォント・ファイルではなくメトリック・ファイルであることに注意してください。このファイルは、プリンタ用に文字を正しくフォーマットする方法を Oracle Reports に伝えるためのものです。AFM ファイルはあるけれども、そのフォントをプリンタで使用できない場合、Oracle Reports ではフォントを生成できません。

AFM ファイル自体はフォントではないので、使用可能な PostScript プリンタ・フォントを増やすには、次のようにする必要があります。

1. フォントを購入して、プリンタにインストールします。
2. フォントおよびプリンタのベンダーから、修正済の AFM ファイルおよび PPD ファイルを入手します。
3. 対応する X サーバー・ディスプレイ・フォントを入手します (必要な場合)。

HPD ファイルおよび TFM ファイル

PCL では、HPD ファイルおよび TFM ファイルが使用されます。HPD ファイルには、プリンタで使用できるフォントのリストが含まれます。フォントはそれぞれ特定の TFM ファイルを参照します。HPD ファイルは編集可能な ASCII ファイルですが、TFM ファイルはバイナリ・ファイルであるため編集できません。TFM ファイルは編集不可能なバイナリ・ファイルですが、文字列操作を実行すればこのファイルから一部のキーワードを読み取ることができます。Oracle Reports によって認識されるフォント名は、HPD ファイルではなく TFM ファイルに記述されているものです。TFM ファイルはフォント・ベンダーから提供されており、TFM ファイルをインストールすると、新しく追加されたフォントがプリンタの HPD ファイルへ追加されます。

Tk2Motif.rgb (UNIX のみ)

このファイルには、Oracle Toolkit に基づくすべての Oracle Motif ツールのリソース設定が含まれます。フォント固有のリソース設定には、Tk2Motif*fontMapCs および Tk2Motif*fontList が使用されます。

Tk2Motif*fontMapCs は、X-Window ディスプレイ上にあり、アプリケーションによって使用されるフォントのベース・キャラクタ・セットを決定します。

Tk2Motif*fontMapCs: iso8859-2=EE8ISO8859P2 の場合、NLS_LANG は EE8ISO8859P2 に設定される必要があり、文字コード体系が iso8859-2 のフォントのみアプリケーションで使用されます。このコード体系のフォントが見つからない場合は、REP-3000 エラーが発生します。

Tk2Motif*fontList は、アプリケーションによって使用されるデフォルトのシステム・フォントを指定します。次のように指定すると、標準の太さ、標準の幅であるサイズ 12 の Helvetica フォントが使用されます。

```
Tk2Motif*fontList: *-helvetica-medium-r-normal-*-120*
```

このエントリの構文の説明は、Tk2Motif.rgb ファイル内にコメントとして記述されています。

4.3.1 ファイルの検索

ファイル検索の基準は、ファイルのタイプおよび対応する定義された環境変数の値によって決まります。

表 4-1 ファイル情報

ファイル名	タイプ	説明
uiprint.txt	UNKNOWN	プリンタ構成ファイル
uifont.ali	FONTALIAS	フォント・エイリアシング・ファイル
PPD	PPD	PostScript プリンタ定義ファイル
AFM	AFM	Adobe フォント・メトリック・ファイル
HPD	HPD	HP glue ファイル
TFM	TFM	HP glue ファイル

変数の検索は、TK_type、ORACLE_type、グローバル・ディレクトリの順に行われます。たとえば、PPD ファイルの検索を行うディレクトリは、TK_PPD で指定されるディレクトリ、ORACLE_PPD、ORACLE_HOME/guicommon/tk/admin/PPD の順となります。

たとえば、uiprint.txt を検索する場合は、最初に環境変数 TK_UNKNOWN が検索され、次に ORACLE_UNKNOWN、デフォルト・ディレクトリの順に検索されます。

環境変数

REPORTS_NO_DUMMY_PRINTER: この変数に値が定義されている場合、Oracle Reports ではプリンタ・フォントのかわりにスクリーン・フォントが使用されます。

4.4 フォント・エイリアシング

フォント・エイリアシングとは、フォントまたはその属性（スタイル、太さ、幅、サイズ、キャラクタ・セットなど）を他のフォントまたは属性にマップできるようにする、Oracle Reports のメカニズムです。主に、あるプラットフォームから別のプラットフォームにアプリケーションを移植する際、レイアウト内のオブジェクトに関連付けられているフォントが移植先がない場合に使用します。このような場合にフォント・エイリアシングを利用すると、存在しないフォントを別の有効なフォントにマップすることによって、要求どおりの結果を生成することができます。たとえば、Windows から Motif へ移行するときにフォント・エイリアシングを使用すれば、Windows Arial を、Motif で利用可能な Helvetica などのフォントにマップできます。

この項の内容は次のとおりです。

- [エイリアシング情報の指定](#)
- [フォント・エイリアシングのメカニズム](#)
- [フォント・エイリアス・ファイルのセクション](#)
- [フォント・エイリアシング・ファイルの検証](#)

4.4.1 エイリアシング情報の指定

マッピングの指定情報が記述されているファイルは `uifont.ali` です。マッピング・ルールを追加または変更するには、このファイルを編集します。

一般的なフォーマットは次のとおりです。

```
"Original_font"="Font_to_be_aliased"
```

ここで、`Original_font` はフォント名またはその他の属性で、`Font_to_be_aliased` はマップ先のフォント名または属性です。

フォント名と属性は次のように指定できます。

```
Face.Size.Style.Weight.Width.CharSet=  
Face.Size.Style.Weight.Width.CharSet
```

`Face` は、Courier などのフォント・フェースの名前（文字列または識別子）です。`Style`、`Weight`、`Width` および `CharSet` には、数値や定義済の文字列または識別子を指定できます。たとえば、`Style` の値として `Plain` および `0` のどちらも有効であり、同じスタイルを表します。`Size` には、正確なサイズをポイント単位で指定します。

これらの属性は、フォント・エイリアシング、フォント・サブセットおよびフォント埋込みを行うときに有効になります。

たとえば、フォント・サブセットでは次のようになります。

```
Fontname=filename.ttf  
Face.Size.Style.Weight.Width.CharSet=filename.ttf
```

認識可能な名前と対応する数値を次に示します。

表 4-2 スタイル名と対応する数値

スタイル名	対応する数値
Plain	0
Italic	1
Oblique	2
Underline	4
Outline	8
Shadow	16

表 4-2 スタイル名と対応する数値 (続き)

スタイル名	対応する数値
Inverted	32
Blink	64

表 4-3 太さと対応する数値

太さ	対応する数値
Ultralight	1
Extralight	2
Light	3
Demilight	4
Medium	5
Demibold	6

表 4-4 幅と対応する数値

幅	対応する数値
Ultradense	1
Extradense	2
Dense	3
Semidense	4
Normal	5
Expand	7
Extraexpand	8
Ultraexpand	9

複数のスタイルを組み合わせることができます。その場合は、プラス記号 (+) を使用してスタイルの各部分を区切ります。次に例を示します。

```
Arial..Italic+Overstrike = Helvetica.12.Italic.Bold
```

これは、イタリックと打消し線の両方のスタイルが指定されている Arial フォントが、12 ポイント、太字、イタリックの Helvetica フォントにマップされることを表します。

マルチバイト言語をサポートするには、キャラクタ・セットのエイリアスを作成し、Adobe から提供されているアジア言語フォント・パックに含まれる CID フォント (第 4.6.7 項「CID フォント」) を指定する必要があります。たとえば、日本語レポートでは、マルチバイトのキャラクタ・セットである Shift-JIS のエイリアスとして、HeiseiKakuGo-W5-Acro という CID フォントを次のように指定します。.....JA16SJIS = "HeiseiKakuGo-W5-Acro"

マッピングに指定する文字列では、大文字と小文字は区別されません。しかし、フォント・フェースの検索時、プラットフォームや画面によっては大文字と小文字が区別される可能性があります。したがって、名前を指定するときは注意が必要です。たとえば、左側 (元のフォント) に arial というフォント名を使用した場合は、フォントとして arial や Arial が指定されているすべてのレイアウト・オブジェクトがエイリアス先のフォントにマップされます。

4.4.2 フォント・エイリアシングのメカニズム

Oracle Reports のフォント・エイリアシングでは、エイリアス・ファイル内の関連するセクションに、レポートで指定されている元のフォントの属性と一致するエントリがあるかどうか検索されます。フォント・エイリアス・ファイルのセクションの詳細は、[第 4.4.3 項「フォント・エイリアス・ファイルのセクション」](#)を参照してください。完全に一致するエントリが見つかった場合は、左側の元のフォントが右側のフォントにマップされます。

次に例を示します。

```
Arial.8.Italic.Medium.Normal.WE8ISO8859P1 =
Helvetica.12.Plain.Light.Normal.WE8ISO8859P1
```

左側に列記されているすべての属性を持つ Arial フォントが見つかった場合は、右側に列記されているすべての属性を持つ Helvetica フォントにマップされます。

フィールドを空白のままにしておくこともできます。その場合は、そのフィールドに関しては比較が行われません。次に例を示します。

```
Arial..... = Helvetica.12.Plain.Light.Normal.WE8ISO8859P1
```

この場合、サイズなどの属性にかかわらずすべての Arial フォントが、キャラクタ・セット WE8ISO8859P1 のサイズ 12、標準スタイル、細字、標準幅の Helvetica フォントにマップされます。

エイリアス規則を指定するには、次のような方法もあります。

```
Arial = "OCR B"
```

この方法では、指定のフォントの他の属性を維持したまま、フォント名が OCR B に変更されます。この場合は、マップされたフォントの属性が、他のフォントで使用できるかどうか確認しておく必要があります。たとえばこの規則では、スタイルがイタリックの Arial フォントは標準スタイルの OCR B フォントにマップされます。これは、OCR B フォントにイタリックというスタイルがないためです。

マップされたフォントがエイリアス・ファイル (uifont.ali) から読み取られた後、[第 4.1.2.1 項「フォントの検索」](#)で説明するフォント検索手順に従ってフォントの検索が行われます。マップされたフォントがシステム上で見つかった場合は、そのフォントが使用されます。それ以外の場合は、システム上で元のフォントが検索されます。

指定されたキャラクタ・セットの下で、フォント・フェース、サイズ、スタイル、太さ、幅の属性が検索されます。

Oracle Reports では、Web ソース・ビューおよび PL/SQL エディタ用のフォントをマップすることもできます。その場合は、[rwbuilder] セクションにマッピング指定を追加します。この機能は、主にこれらのエディタでの Unicode フォントのサポートを目的として用意されたものです。

4.4.3 フォント・エイリアス・ファイルのセクション

Oracle Reports では、uifont.ali ファイルは次の場所にあります。

```
ORACLE_HOME\tools\common (Windows)
ORACLE_HOME/guicommon/tk/admin (UNIX)
```

[表 4-5](#) に示すように、エイリアス・ファイルは様々なセクションから構成されており、個々のセクションには特定の領域に対するマッピングの指定が記述されます。Oracle Reports では、目的に応じた特定のセクションが検索されるため、目的を達成するには適切なセクションにマッピングのエントリを記述することが重要です。

表 4-5 フォント・マッピング・ファイルのセクション

セクション名	説明
Global	全体に適用されます。
Printer	プリンタ出力にのみ適用されます。

表 4-5 フォント・マッピング・ファイルのセクション (続き)

セクション名	説明
Printer:PostScript1	PostScript レベル 1 のプリンタにのみ適用されます。
Printer:PostScript2	PostScript レベル 2 のプリンタにのみ適用されます。
Printer:PCL5	PCL 5 のプリンタに適用されます。
Display	ディスプレイ (画面) にのみ適用されます。
Display:Motif	Motif ディスプレイにのみ適用されます。
Display:CM	キャラクタ・モード・ディスプレイにのみ適用されます。
PDF	フォント・エイリアシング (Oracle Reports 6i 以降) およびマルチバイト言語サポート (Oracle Reports 以降) に使用されます。
PDF:Embed	(Oracle Reports のみ) Type1 フォントの埋込みに使用されます。
PDF:Subset	(Oracle Reports のみ)
RWBUILDER	(Oracle Reports のみ) Web ソースおよび PL/SQL エディタ用のフォントのマッピングをこのセクションで指定します。
printer_name	特定のプリンタ向けのセクションを表し、次のように指定します。 [Printer:PostScript1:2op813a]

関連項目: 第 4.5 項「フォントに関する問題のトラブルシューティング」の「Web ソース・ビューで正しく表示されないフォントの修正」

優先順位

フォント・エイリアシングを行う際は、そのフォントが使用される状況に応じて 1 つのセクションのみが読み取られます。したがって、3 つのセクションが適用される場合でも、読み取られるセクションは 1 つのみです。たとえば、[Printer]、[Printer:PostScript] および [Printer:PostScript:2op813a] の 3 つのセクションがあるとします。出力生成時に、プリンタが 2op813a ならば、セクション [Printer:PostScript:2op813a] のマッピング・ルールのみ読み取られます。プリンタが 2op813a 以外の場合は、[Printer:PostScript] セクションが使用されます。

エイリアス・ファイルのセクションでは、より限定的に指定しているセクションが、全般的な指定のセクションよりも優先されます。たとえば、[Printer:PostScript1:2op813a] という特定のプリンタ用のセクションは [Printer:PostScript1] セクションよりも優先され、それ以降は [Printer] セクション、[Global] セクションの順に優先順位が低くなります。

uifont.ali ファイルは、Oracle Reports の PDF フォントに関する拡張機能すべてを制御する構成ファイルです。このファイルは、ORACLE_HOME¥tools¥common ディレクトリ (Windows) および ORACLE_HOME/guicommon/tk/admin ディレクトリ (UNIX) にあります。uifont.ali は読み取り可能なテキスト・ファイルです。つまり、標準的なテキスト・エディタで編集できます。このファイルを編集する場合は十分に注意してください。uifont.ali ファイルを保存するときは、テキスト・ファイルとして保存します。ファイルを破損させるおそれのあるフォーマットや特殊文字は使用しないでください。

関連項目: uifont.ali ファイルの各セクションの詳細は、第 6 章「Oracle Reports での PDF の使用」を参照してください。

4.4.4 フォント・エイリアシング・ファイルの検証

uifont.ali ファイルの内容が正しいことを検証するには、`ORACLE_HOME/bin` ディレクトリにあるフォント検証ユーティリティを使用できます。uifont.ali ファイルを変更した場合は、エラーを見つけるために必ずこのユーティリティを実行することをお勧めします。

Windows の場合：

```
fnchk.exe filename
```

UNIX の場合：

```
mfontchk filename
```

ここで、`filename` は uifont.ali ファイルの名前です。ファイル名を指定しない場合は、環境変数に基づいてデフォルトのファイルが検証されます。

エイリアス・ファイルにエラーがある場合は、エラー・メッセージとともに、エラーが見つかったファイルの名前が返されます。次に例を示します。

```
Parsing font alias file "/home/oracle/guicommon/tk/admin/uifont.ali"
Ms san serif
```

```
Error at line 85: Invalid font specification
Parse of font alias file failed
```

このエラーは、uifont.ali の 85 行目の MS San Serif フォントのマッピング・ルールに構文エラーがあることを示しています。

4.5 フォントに関する問題のトラブルシューティング

フォントに関する問題がアプリケーションで生じた場合の解決に役立てるために、この項では、次のトラブルシューティング情報について説明します。

- [PostScript ファイル内で目的のフォントが使用されているかどうかの確認](#)
- [出力の作成](#)
- [出力ファイルの読取り](#)
- [出力ファイルの検証](#)
- [印刷フォントの修正](#)
- [環境変数の確認](#)
- [Web ソース・ビューで正しく表示されないフォントの修正](#)
- [制約事項](#)
- [共通の問題の解決](#)

PostScript ファイル内で目的のフォントが使用されているかどうかの確認

PostScript ファイルには、PPD ファイルを読み取った後、フォントのリストが作成されます。PostScript ファイルで次のタグを検索すると、フォントを確認できます。

- `DocumentNeededResource` には、PPD ファイル内で参照されるフォントのリストが記述されています。
- `DocumentSuppliedResource` には、PostScript ドライバによって AFM ファイルを見つけることができたフォントのリストが記述されています。
- フィールド `%IncludeResource:font` の前の `%%Page` パラグラフには、そのフィールドで使用されるフォントの名前が記述されています。

PCL 出力ファイルの場合は、特定のフォントが使用されているかどうかを確認できます。この情報に応じて、Oracle Reports またはプリンタのフォント設定を変更します。

例：

次に示すのは、Lexmark Optra プリンタに基づくテストの結果です。フォントとその番号および制御コマンドは例であり、他のプリンタではこれとは異なる場合があります。

フォント情報 Lexmark のプリンタには、使用可能なすべてのフォント（PCL エミュレーション・フォント）を印刷するメニュー・オプションがあります。これには、常駐フォント（デフォルト）とフラッシュ・フォント（プリンタに別途インストールされる）の両方が含まれます。

表 4-6 フォント情報の例

フォント名	スタイル	太さ	出力の例
R0 Courier	0	0	... <ESC><symset><ESC>(s0p<pitch>h0s0b4099T...
R39 Courier Bold	0	3	... <ESC><symset><ESC>(s0p<pitch>h0s3b4099T...
R40 Courier Italic	1	0	... <ESC><symset><ESC>(s0p<pitch>h1s0b4099T...
R55 Century Schoolbook Roman	0	0	... <ESC><symset><ESC>(s1p<point>v0s0b24703T ...

表 4-7 フラッシュ・フォント情報の例

フォント名	シンボル・セット	スタイル	太さ	出力の例
F2 OCR-A	0O	0	0	... <ESC>(0O<ESC>(s0p<pitch>h0s0b4200T ...
F3 OCR-B	1O	0	3	... <ESC>(1O<ESC>(s0p<pitch>h0s0b4206T ...

これらの例では、ほかにも多数のフォントがあり、それぞれ異なるコードが定義されています。たとえば、OCRB のコードは 4206 です。この番号の重要性については後述します。

出力の作成

正しいフォントが使用されない問題がある場合は、レポートを単純化します。つまり、出力を単純なものにします。これを行うには、`select sysdate from dual` という問合せを使用し、フォントの数を少なくして単純なレポートを作成します。これにより実行時間が短くなり、出力ファイルのサイズも大幅に縮小されます。

出力ファイルの読取り

作成される PCL ファイルはバイナリ・ファイルですが、テキスト・エディタである程度まで読み取ることができます。最初の短い部分と最後の部分はバイナリですが、中間部分のデータは読取り可能です。

出力ファイルの検証

注目する必要がある情報は、ファイルの中間部分にある読取り可能なデータのみです。テキストを見つけて（これがレポート出力に表示されるテキストです）、そのテキストの直前の部分を確認します。

これは次のようになります。

```
...;SD1,14,2,0,3,10.34,5,0,6,0,7,4099;LB here is your text
```

この例では、フォントにはコード 4099 のものが選択されています。Lexmark のプリンタでは、Courier が選択されます。

別の例として、フォント OCR-B (コード 4206) が必要である場合を考えます。選択されたテキストの直前にこのコードが生成されていなければ、このフォントでは表示されません。これは次のようになります。

```
....;SD1,14,2,0,3,8.57,5,0,6,0,7,4206;LBThis is OCRB font....
```

印刷フォントの修正

出力ファイル内のコードは正しいけれども、プリンタでそのフォントが印刷されない場合は、そのフォントがプリンタでは使用できない状態にある可能性があります。この問題は、出力ファイル内のコード (TFM ファイルから得られます) が、プリンタで処理可能なものと異なる場合にも発生します。このような場合、Lexmark のプリンタでは、これらのフォントがプリンタのデフォルト・フォントによって置き換えられます。

フォントに対応するコードが出力ファイル内にない場合は、Oracle Reports によってそのコードが出力ファイル内に生成されなかったことを表しています。HPD ファイルおよび TFM ファイルを確認してください。

環境変数の確認

DEBUG_SLFIND を利用すると、使用されたファイルを確認できます。フォントに関しては、アプリケーションがプリンタ定義ファイルを読み取った後に検索した AFM ファイルおよび TFM ファイルのリストと、エイリアス処理後に読み取ったフォント・ファイル調べることができます。この方法では、あるフォントがマップされているかどうかを調べることができます。通常は、次の順序でファイルの読取りが行われます。

- 最初にプリンタ定義ファイルを読み取ります。
- このプリンタ定義ファイルで指定されているフォントに対応するフォント・ファイルをすべて読み取ります。
- エイリアス・ファイルを読み取ります。
- フォントのマッピングがある場合は、そのフォントのフォント情報ファイルを読み取ります。出力生成に使用されるフォントについては、最後にもう一度 AFM ファイルを読み取ります。

TK_DEBUG_POSTSCRIPT は、PostScript 出力に影響を及ぼします。この変数には、次の文字列の任意の組合せを設定できます。

- **Functions** を指定すると、コールされる個々のツールキット関数が PostScript 出力のコメントとして出力されます。
- **Long** を指定すると、量が長く時間もかかりますが、理解しやすい PostScript が生成されます。
- **Memory** を指定すると、各ページの下部にメモリ使用率が表示されます。

これらのオプションはいずれも、環境変数内で指定するときに 1 文字に短縮できます。またこれらのオプションは、"/" で区切って自由に組み合わせることができます。この環境変数では、大文字と小文字は区別されません。たとえば、すべてのオプションを使用するには、Func/L/Mem のように指定します。

この変数を使用して生成された出力を顧客向けに使用することは、オラクル社によるサポートの対象外です。この変数は、診断目的にのみ使用してください。

Web ソース・ビューで正しく表示されないフォントの修正

Reports Builder のユーザー・インタフェースのテキスト (ウィンドウのタイトルなど) には、現行の言語に対応するシステム・リソース・ファイルから取得したフォントが使用されます。これらのシステム・リソース・ファイルは、Oracle Reports のインストールに含まれています。Oracle Reports では、uifont.ali の [rwbuilder] セクションでこれらのフォントのマッピングを指定できます。マップされているフォントが見つかった場合は、元のフォントのかわりにそのフォントが使用されます。見つからなかった場合は、元のフォントが使用されます。

注意: マップされたフォントは固定幅フォントである必要があります。

レポート・エディタの Web ソース・ビューでは、アラビア語、中央ヨーロッパ言語、キリル語、ギリシャ語、ヘブライ語、日本語、タイ語、トルコ語などが文字化けして表示されることがあります。uifont.ali で Reports Builder のフォント名を次のように設定することで、この問題を回避できます。

```
[rwbuilder]
.....AR8MSWIN1256="Courier New"
.....CL8MSWIN1251="Courier New"
.....EE8MSWIN1250="Courier New"
.....EL8MSWIN1253="Courier New"
.....IW8MSWIN1255="Courier New"
.....JA16SJIS="MS Gothic"
.....TH8TISASCI="Andale Duospace WT"
.....TR8MSWIN1254="Courier New"
```

Andale Duospace WT (固定幅) フォントは、Oracle Metalink (<http://metalink.oracle.com>) からダウンロードできます。ARU 番号は 2766564 です。

制約事項

Windows の場合:

- Unicode では、True Type Big Font を提供しています。これらのフォントには、複数の言語によるテキストの表示や印刷に必要な文字が含まれています。たとえば、西ヨーロッパ言語、中央ヨーロッパ言語およびアラビア語のテキストをフィールドに入力、表示または印刷したときに予期しない文字が出力された場合は、Big Font を使用していない可能性があります。Microsoft Windows でシングルバイト言語に用意されている Big Font は、Arial、Courier New および Times New Roman です。詳細は、Microsoft 社の Web サイト (<http://www.microsoft.com/typography/fonts/default.aspx>) を参照してください。
- Wingdings フォントは、NLS_LANG が UTF8 に設定されているときは表示されないことがあります。

Wingdings フォントの中で UTF8 の使用時に表示可能なものは、ASC 32 ~ 127 の文字のみです。ASC 252 は UTF8 ではサポートされていないため空白表示されます。

この問題に対処するには、次のいずれかのフォント・セットを使用します。

- Webdings - chr(97)
- Wingdings2 - chr(80)
- Wingdings2 - chr(87)

UNIX の場合:

- AFM がサポートされるのは、日本語エンコードを除き、シングルバイトの PostScript ファイル生成時のみです。AFM ファイルに対してサポートされるコード体系は、AdobeStandardEncoding、ExtJIS12-88-CFEncoding、FontSpecific、HRoman、ISOLatinHebrew、JIS12-88-CFEncoding および JIS12e-88-CFEncoding です。
サポート対象の AFM のバージョンは 2.0 です。
- X11 では、フォント属性として下線はサポートされません。ファイルへの出力は、後述する手順に従って行われます。
- JDK では、バグによって太字の韓国語フォントが適切に表示されません。OracleAS Reports Services では JRE が使用されているため、レポート内のグラフに使用される太字の韓国語文字列がすべて正しく表示されません。

- PostScript を印刷する場合、プリンタへのフォントのロードは行われません。したがって、目的のフォントが印刷結果に出力されるようにするには、そのフォントがプリンタにインストールされていることを確認する必要があります。
- PCL 出力の場合は、TFM フォント・フォーマットのみサポートされます。
- UNIX のディスプレイ・システム (X11 など) は、特定のアプリケーションやプリンタには一切依存しません。印刷と表示の間には、直接の関係はありません。画面に表示されるフォントが印刷されない場合もあります。

ディスプレイ・フォントとプリンタ・フォントにはいくつかの類似点もありますが、それ以上の相違点があります。

X フォント (ディスプレイ・フォント) はビットマップの表示用絵文字であり、X サーバーによって X ターミナル上に表示されます。

プリンタ・フォントは PostScript フォント、つまりフォントを数値として表したものでビットマップではありません。このフォントは PostScript プリンタ内に存在し、プリンタ上の PostScript インタプリタによって生成されます。

- テンプレートを適用すると、フォント・サイズが変更されることがあります。

例えば、テンプレートを作成してフォントをサイズ 10 の Times New Roman に設定し (すべてのフィールドが対象)、レポートでこのテンプレートを使用するように設定した場合でも、レポート・エディタのペーパー・デザイン・ビューで表示されるフォントのサイズが変更されていることがあります。

これは、デフォルト設定のレイアウトでは目的の領域内に収まらない場合に行われます。

最初にテキスト・フィールドのサイズが縮小され、次にフォントのサイズが縮小されます。このようにすれば、テンプレートのフォント・サイズを維持してフィールドを折り返すよりもよい結果を得ることができます。

また、テンプレートを使用する場合に、テンプレートで指定されたものとは異なるフォントが選択されることがあります。これは、最初にキャラクタ・セットの比較が行われるためです。したがって、テンプレートのフォントが現行のキャラクタ・セットをサポートしないものである場合は、サポートするフォントに変更されます。これは、特にヘブライ語またはアラビア語環境で英語のテンプレートを使用した場合にはっきりとわかります。

共通の問題の解決

問題: ラベル・レポートを印刷すると、右マージンの文字が切り捨てられる

Windows マシン上で郵便ラベル・レポートを印刷したのですが、各行の最後の数文字が切り捨てられていることがわかりました。レポートのプレビューでは、これらの文字は欠けてはいませんでした。ページのフォーマットとフォント設定を変えてみましたが、問題は解決しません。

解決策: Preview を DESTYPE にしてレポートが正しく表示されるのであれば、プリンタ・ドライバの問題ではありません。この問題の原因として、枠のプロパティが考えられます。

レイアウト・オブジェクトを囲む枠の水平拡張度が「固定」に設定されており、データの長さが枠のサイズを超える場合は、このようなデータの切捨てが発生します。

「水平拡張度」プロパティを、「拡張」または「可変」に設定してみてください。

問題: ファイルを HTMLCSS として生成すると、列の 1 つが出力から欠落する

レポートを HTMLCSS ファイル・フォーマットとして出力しており、レポート・エディタのペーパー・デザイン・ビューでは問題なく表示されています。新たに作成されたファイルをクリックしてブラウザに表示されるレポートを確認すると、最後の列がレポート出力の中ではありません。

レポートを再度実行すると、ペーパー・デザイン・ビューでは問題なく、最後の列も本来の位置に表示されます。このファイルをもう一度クリックしたときに、最後の列がレポート出力から脱落しているようです。PDF は、ペーパー・デザイン・ビューと Adobe Acrobat Reader で問題なく表示されます。

解決策：

1. Oracle Reports およびその他の実行中のアプリケーションを終了します。
2. Windows の「コントロールパネル」→「画面」→「設定」を選択します。
3. フォントを「小さいフォント」に設定して「適用」ボタンをクリックし、「OK」をクリックして Windows のフォント設定を変更します。
4. コンピュータを再起動して、新しいフォント設定を有効にします。
5. もう一度 Windows の「コントロールパネル」→「画面」→「設定」を選択して、システムのデフォルトが「小さいフォント」に設定されていることを確認します。

HTMMLCSS ファイルをクリックすると、ブラウザにレポートが正しく表示され、すべての列が欠けることなく表示されます。

HTMMLCSS ファイルをブラウザで表示するときは、Windows システムのデフォルト設定として「小さいフォント」を選択しておくことをお勧めします。

大きいフォントがデフォルトで設定されている場合、HTMMLCSS ファイルが正しく表示されないことがあります。

問題： Reports Builder で、8 ポイントより小さいビットマップ・フォント・サイズを選択する方法

解決策： レポートによっては、フォント・サイズを 6 ポイント以下にする必要があることもあります。フォントのマッピングおよびサイズ設定は、実際にはオペレーティング・システムのフォント・ファイルと、ドライバまたはプリンタの指定の両方によって実現されるものであるため、多くのフォントで 6 ポイント以下などの小さなフォント・サイズを指定することが可能です。

Oracle Reports では一般に、フォント・サイズを 8 まで小さくすることができます。このように設定するには、Reports Builder でレポートを開き、レイアウト・モデル・ビューに切り替えて、変更対象のレポート・オブジェクトを選択します。オブジェクトが選択されている状態で、フォント選択ボックスの横のリストからフォント・サイズを選択します。

一般に、選択できるサイズは TrueType フォントの場合は 8～72、その他のフォントの場合はこれより少なくなります。

リストにあるサイズよりも小さいサイズや大きいサイズを入力できます。それにはもう一度オブジェクトを選択して、カーソルをフォント・サイズ・フィールドに置き、[Delete] キーを押してサイズの数値を削除し、目的のフォント・サイズを入力して [Tab] キーを押します。変更は即座に反映されます。

この場合も、すべてのフォント・サイズを指定できるとは限らないことに注意してください。また、フォントと属性の組合せによっては実際に使用できないものもあります。あるフォント・サイズを選択できたとしても、印刷したときにそのフォントを読み取ることができるとは限りません。小さいフォント・サイズを指定し、さらに太字やイタリックなどの属性を組み合わせの場合も、プリンタ・ドライバ、プリンタ、フォント・メトリック、言語、コード・セット、NLS_LANG および人間の視覚などの制約が原因で、印刷または表示したときに読取り不可能の問題が発生することがあります。

問題： レポート出力でのフォント・サイズが、Windows と UNIX とで異なる

Windows 上で設計した単純なレポートで、サイズ 8 の Arial フォントを使用しています。このレポートを Sun Solaris に移植したときに、Solaris 上の出力ではフォント・サイズが変更されていることがわかりました。UNIX 環境では、レポートにフォント・サイズ 9 の Helvetica を使用しています。UNIX 上では、Arial フォントは同等のフォントである Helvetica に uifont.ali を使用してマップされています。

解決策：

1. まず、`xlsfont` コマンドなどの UNIX のフォント・ユーティリティを使用して、UNIX システム上の Helvetica で使用可能なフォント・サイズを調べます。
2. Windows の可変サイズ・フォントを、UNIX の可変サイズ・フォントにマップします。たとえば、Windows でのマッピングを `Arial.8 = Helvetica.8` に変更し (UNIX システム上で Helvetica に対してサイズ 8 が使用可能であるものとします)、`uifont.ali` が正しいディレクトリにあることを確認します (フォント・マッピングの説明を参照)。

マシンにインストールされている Helvetica フォントがビットマップ (ラスタライズ済) であることが考えられますが、その場合は、任意のサイズに自動的に拡大または縮小することはありません。これに該当する場合は、拡張性に優れた Type1 フォントをインストールすれば、どのようなポイント・サイズでも選択できるようになります。

システムが異なれば、インストールされているフォントが同じであっても、フォント間の差が発生する可能性は常にあります。これは、システムによってフォント構成ファイルが異なるためです。

問題： 印刷時に、フォントが非 TrueType フォントによって置き換えられる (ペーパー・デザイン・ビューでは、フォントの問題は発生しない)

解決策： プリンタ設定 (詳細) を調べて、次のような指定がないことを確認します。

True Type Font: *Substitute with Device Font*

UNIX

問題： Oracle Reports を X-Windows エミュレータ上で実行しているときに、UNIX にインストールされているフォントがフォント検索ボックスに表示されない

解決策： X-Windows エミュレータのフォント・パスは、通常、ローカル・マシン上のフォント・ディレクトリになります。インストール済のフォントは使用できず、Oracle Reports のフォント検索ボックスでは、ローカルのフォント・ディレクトリにあるフォントのみが使用されます。このような場合は、フォントがインストールされているリモート・マシン上でフォント・サーバーを起動し、フォント・パスのエントリがこのフォント・サーバーを指すようにします。フォント・サーバーの起動およびフォント・パス・エントリの設定の詳細は、システムのマニュアルおよび X-Windows エミュレータのヘルプを参照してください。

現在使用されているフォント・パスおよびフォント・サーバーを調べるには、UNIX コマンド `xset -` を使用します。

4.6 フォントのタイプ

この項では、Oracle Reports に関係のあるフォントおよびキャラクタ・セットについて説明します。

- [キャラクタ・セット](#)
- [Unicode](#)
- [Type1 フォント](#)
- [TrueType フォント](#)
- [TrueType Collection](#)
- [バーコード・フォント](#)
- [CID フォント](#)

4.6.1 キャラクタ・セット

NLS 環境変数のキャラクタ・セット・コンポーネントでは、各自の環境でデータを表す際に使用されるキャラクタ・セットを指定します。あるキャラクタ・セットを使用するシステムから別のキャラクタ・セットを使用するシステムにデータが転送されると、転送先のシステムでそのデータが処理され、正しく表示されます。ただし、一部の文字がキャラクタ・セットの別のバイナリ値で表される場合があります。

多言語対応アプリケーション、または単一言語対応であっても複数のキャラクタ・セットを使用するアプリケーションを設計する場合は、実行時に最も広く使用されるキャラクタ・セットを決定し、NLS 環境変数 (`NLS_LANG`) をそのキャラクタ・セットに設定してアプリケーションを生成する必要があります。

あるキャラクタ・セットで設計および生成されたアプリケーションを別のキャラクタ・セットで実行すると、パフォーマンスに問題が生じる可能性があります。さらに、生成時のキャラクタ・セットの文字の一部が実行時のキャラクタ・セットに含まれていない場合は、認識されない文字の部分に疑問符が表示されます。Portable Document Format (PDF) は、マルチバイト・キャラクタ・セットをサポートします。特定のフォントを使用してアプリケーションを作成しても、実行してみると別のフォントが使用されている場合があります。この問題は、多くの場合、西ヨーロッパ圏以外の言語環境で英語フォント (MS Sans Serif や Arial など) を使用すると発生します。この問題が起こる理由は、フォントに関連付けられているキャラクタ・セットと言語環境変数 (`NLS_LANG`) によって指定されたキャラクタ・セットが一致しているかどうかを Oracle Reports がチェックするためです。2つのキャラクタ・セットが一致しない場合は、Oracle Reports によって、言語環境変数で指定されたキャラクタ・セットが関連付けられている別のフォントに自動的に置換されます。この自動置換によって、データベースから返されるデータがアプリケーション内で正しく表示されます。注意: 英語フォントを使用して各国語の文字を入力すると、Windows では暗黙的に別のフォントへの関連付けが行われます。ただし、このような自動置換が行われないようにする場合もあります。自動置換が行われないようにするには、希望するすべてのフォントを、フォント・エイリアス・ファイル (uifont.ali) 内の WE8ISO8859P1 キャラクタ・セットにマッピングします。

4.6.2 Unicode

Unicode は、1つのアプリケーションで多言語テキストを表示できるようにするグローバルなキャラクタ・セットです。これによって多国籍企業では、1つの多言語アプリケーションを開発し、それを世界各国に配布できます。多言語アプリケーションでの Unicode の使用の詳細は、第 18.5 項「Unicode」を参照してください。

4.6.3 Type1 フォント

PostScript フォントのフォーマットである Adobe Type1 フォントの保存形式には、.pfa (PostScript Font ASCII) と .pfb (PostScript Font Binary) の 2 種類があります。これらのファイルには文字の形状に関する記述が含まれます。個々の文字は小さなプログラムによって生成されますが、このプログラムはさらに別の小さなプログラムをコールして、フォント内の文字の共通部分を計算します。どちらの場合も、文字の記述内容は暗号化されます。このようなフォントを利用できるようにするには、PostScript インタプリタ、または Adobe Type Manager などの専用のレンダリング・エンジンを使用して、ビットマップのドットにレンダリングする必要があります。Adobe Type Manager は、Apple Macintosh および Microsoft Windows のシステム上で、低解像度のスクリーン・フォントを生成する場合に使用されます。

Type1 バイナリ・ファイル (.pfa、.pfb) には文字情報が含まれますが、メトリック・ファイル (.afm (Adobe Font Metric) または .pfm (Printer Font Metric)) には文字を形成するためのメトリック情報が含まれます。これらのメトリック・ファイルはテキスト・ファイルであり、明確かつ解析しやすい構造で記述されています。

4.6.4 TrueType フォント

パーソナル・コンピュータの出現によって拡張性に優れたフォント・テクノロジーの必要性が高まり、それ以降のあらゆるオペレーティング・システムにおいて不可欠であることが予想されていきました。TrueType はまさに拡張性に優れたフォント・テクノロジーであり、文字を拡大しても出力結果は変わらず、ビットマップ・フォントに顕著なざざざが発生しません。

このテクノロジーは次の 2 つの部分から成り立ちます。

- ラスタライザ
- TrueType フォント

ラスタライザは、Windows と Macintosh の両方のオペレーティング・システムに組み込まれているアプリケーションです。インタプリタとして機能し、フォント情報を、ビデオ・ディスプレイでレンダリング可能な形式へ変換します。

TrueType フォントは、それ自身にタイプフェイスの各文字のアウトラインを表す情報が含まれます。より高品質なフォントには、ヒントング・コードも含まれています。ヒントングとは、縮小されたフォントの形状を最適化する処理です。ベクター・アウトラインにのみ依存するのではなく、文字の線がピクセルに揃うようにするため、フォントの形状がスムーズになり、読みやすさも最大になります。

Adobe 社は、この機能を実現する自社の PostScript コードのライセンス供与を Apple と Microsoft の両社に対して提案しましたが、両社はいずれも、自社のオペレーティング・システムの中核部分に他社の支配力が及ぶということに懸念を抱いていました。Apple 社と Microsoft 社は特許の相互使用と製品開発協定について合意し、Microsoft 社は PostScript スタイルのグラフィック・エンジンを開発し、Apple 社はフォント・システムを開発しました。Apple 社が開発したものは後に TrueType となり、パフォーマンスとレンダリング品質において最高のテクノロジーであることを証明しました。Apple 社と Microsoft 社は、Adobe 社に対抗する戦略的業務提携を発表しました。これは、Apple 社がフォント・システム、Microsoft 社が印刷用エンジンの開発を担当するというものでした。Apple 社は、1991 年 3 月に TrueType を発表しました。最初の TrueType フォントは次のものでした。

- Times Roman
- Helvetica
- Courier

1992 年初頭には、Microsoft 社の Windows バージョン 3.1 に TrueType が導入され、次のフォントから構成されるコア・フォント・セットが作成されました。

- Times New Roman
- Arial
- Courier

Apple 社と Microsoft 社のどちらの TrueType フォントも、スケーラブル・フォントからビットマップを生成する際に、あたかも人間が設計したかのような高い品質をフォントの各サイズで実現しています。

4.6.5 TrueType Collection

TrueType Collection (TTC) は、文字情報や絵文字などの共通するフォント・データを共有する効率的な方法です。こうしたデータの共有化によって、共通の絵文字などが各フォントではなく単一のファイル構造に格納されるため、ファイル・サイズが最適化されます。最も効率的なのは、複数のフォントを組み合わせた 1 つのファイルです。たとえば、同じフォント・ファミリーに属する複数の日本語フォントは、共通の漢字文字セットを共有できます。これらの漢字文字は、TTC ファイルに保持されます。

たとえば、TTC ファイルの msgothic.ttc は、MS Gothic、MS PGothic および MS UI Gothic の 3 種類のフォントで構成される集合ファイルです。

4.6.6 バーコード・フォント

バーコード・フォントという用語が表すものは1つではありません。業種によっては、特定のバーコード・タイプが選定されていることがあります。このような場合は、適切なバーコード・フォントを使用できるようにする必要があります。たとえば、商品パッケージや書籍にバーコードを印刷する場合のバーコードの選択は簡単です。北米では、商品パッケージのバーコードに UPC-A を使用します。ヨーロッパでは、商品のバーコードには EAN を使用します。

書籍の ISBN 番号のバーコードには、Bookland (EAN 13 バーコードに 5 桁を付加したもの) を使用します。フォントはバーコードを印刷する方法の1つですが、唯一の方法ではありません。Oracle Reports では、このほかにバーコード用の Java Bean を使用してバーコードを生成するという方法も利用できます。この Java Bean を使用すれば、一般的に使用されているバーコード・タイプに基づいてバーコードを生成することができます。

4.6.7 CID フォント

CID (Character IDentifier) フォントは、コンポジット (マルチバイト) Type1 フォントの一種で、東アジア言語圏の要求に応じて開発されたものです。Adobe 社が開発した CID-Keyed フォント・ファイル・フォーマットは、PostScript で使用される幅広いキャラクタ・セット・フォントに対応します。中国語、日本語および韓国語のフォントに最適なフォーマットですが、このほかにも様々なキャラクタ・セットのローマン体フォントに使用できます。CID-Keyed とは、文字識別子 (CID) 番号を使用してフォント内の文字に索引を付け、アクセスすることです。CID フォントはそれぞれ、すべての文字のアウトラインが格納された1つの大きなフォント・ファイルと、文字、エンコードおよび文字識別子のリストが格納された1つの小さな CMap ファイルから構成されています。フォント・ファイルと CMap ファイルを組み合わせることにより、特定のキャラクタ・セットとエンコード情報からなるフォントを作成することができます。CID フォントはそれぞれ、多数のキャラクタ・セットとエンコードの組合せをサポートできます。

UNIX での Oracle Reports による印刷

Oracle Reports は、印刷に関する豊富な機能を備えており、様々なプラットフォームですぐに使用できます。UNIX 上で印刷する場合、適切な印刷環境の作成には、いくつかの設定および構成が必要になります。この章では、UNIX での Oracle Reports による印刷について説明します。この章の主な項は、次のとおりです。

- UNIX での印刷の概要
- UNIX でのプリンタの設定
- 印刷環境の構成
- プリンタ関連のファイル
- グローバリゼーション・サポート
- デバッグ・オプション
- よくある質問

5.1 UNIX での印刷の概要

この項では、UNIX で Oracle Reports を使用して印刷を行う方法について、UNIX と Windows のプラットフォームの主な相違点を取り上げながら説明します。また、任意のアプリケーションから正しく印刷を行うための、オペレーティング・システムに関する要件についても説明します。

- 全体的な印刷のメカニズム
- UNIX および Windows での Oracle Reports の印刷メカニズム
- 印刷のサポート

5.1.1 全体的な印刷のメカニズム

UNIX での Oracle Reports による印刷のしくみを理解するには、Microsoft Windows の印刷メカニズムと比較して考えるとよいでしょう。Microsoft Windows では、アプリケーション・レベルの API が、インストールされているプリンタ・ドライバに基づいて様々なタイプのプリンタをサポートします。そしてアプリケーションは、この標準 API を通じて様々なプリンタ・ドライバと通信します。たとえば、印刷用紙のマージンを変更する場合、アプリケーションは Microsoft Windows の該当する API メソッドをコールし、このメソッドが変更内容をプリンタ・ドライバに伝えます。Microsoft Windows では、プリンタ・ドライバはプリンタ固有です。つまり、プリンタごとに特定のプリンタ・ドライバをインストールする必要があります。プリンタ・ドライバはプリンタとの通信方法を理解しており、プリンタに出力を送信するアプリケーションにかわってその処理を行います。アプリケーションからは、プリンタのプロパティの参照、プロパティの変更、および標準 API を介した印刷の実行が可能です。

Motif およびキャラクタ・ベースの UNIX オペレーティング・システムには、Microsoft Windows と異なり、プリンタに対する独自の標準インタフェースがありません。出力をプリンタへのストリーム・ファイルに送出すること、およびプリンタの仕様に従うことは、個々のアプリケーションの責任において行われます。UNIX プラットフォームでは、Oracle Reports の出力をストリームとしてプリンタに送信する前に、PostScript や PCL などの形式で正しくフォーマットする必要があります。UNIX 上で印刷を行うために、Oracle Reports の内部で Microsoft Windows のプリンタ・ドライバの動作と同様の処理が行われます。次の項では、UNIX でのこのメカニズムをさらに詳しく説明します。

5.1.2 UNIX および Windows での Oracle Reports の印刷メカニズム

図 5-1 と図 5-2 は、UNIX と Microsoft Windows での Oracle Reports の印刷の違いを示したものです。

図 5-1 UNIX での Oracle Reports の印刷

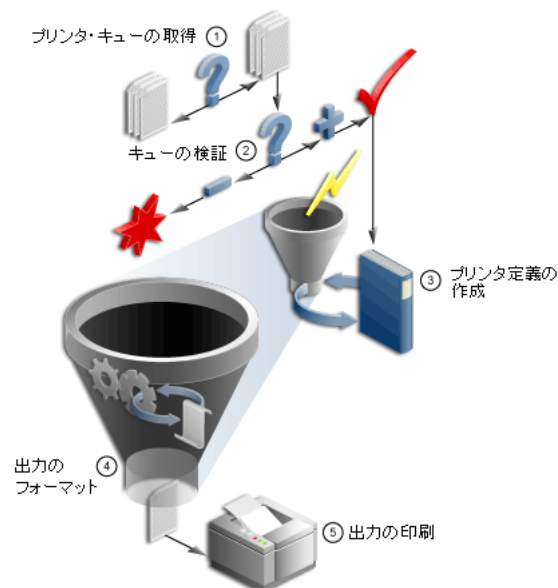
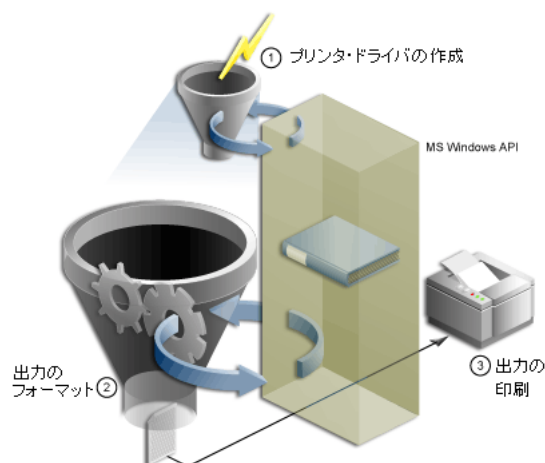


図 5-2 Microsoft Windows での Oracle Reports の印刷



UNIX で印刷を行うために、Oracle Reports では論理プリンタ・ドライバが内部的に作成されます。論理プリンタ・ドライバは Microsoft Windows のプリンタ・ドライバの動作をシミュレートするもので、UNIX での Oracle Reports の印刷サービス・インタフェースとなります。この論理プリンタ・ドライバを通して、Oracle Reports からプリンタのプロパティにアクセスし、プリンタ関連の操作を実行することができます。論理プリンタ・ドライバは、次の処理を行います。

- 最も一般的な印刷標準である、PostScript および PCL の印刷仕様をサポートします。
- プリンタ記述ファイル (PPD や HPD など) を読み取り、プリンタ情報を取得します。
- PostScript または PCL の出力を生成する際に、様々なプリンタ・コマンドを埋め込みます。たとえば、印刷用紙のマージンを変更するには、生成する出力内に対応するプリンタ・コマンドを書き出す必要があります。このコマンドは、PostScript プリンタと PCL プリンタのどちらを使用するかによって異なります。生成された PostScript または PCL のファイルが印刷プログラム (lpr など) を介してプリンタに送信されると、プリンタはこのファイル内のコマンドを解釈し、それに従って処理を実行します。

論理プリンタ・ドライバを正しく動作させるには、次の入力が必要です。

- 印刷要求のプールに使用するプリンタ・キューの名前
- プリンタのプロパティが記述されているプリンタ記述ファイル
- 指定のプリンタ・キューが要求するドライバのタイプ (PostScript または PCL)

この情報を指定するには、uiprint.txt という名前のファイルを使用します。Oracle Reports では、印刷に使用できるプリンタ・キューの名前のリストがこのファイルから取得されます。uiprint.txt には、プリンタ・キューの名前、キューに必要なドライバのタイプ、ドライバのバージョン、およびそのプリンタでプリンタ・ドライバが特別なプリンタ記述ファイルを必要とする場合はその名前 (PostScript ドライバの PPD ファイルなど) を指定する必要があります。この情報を基に内部的な論理プリンタ・ドライバが作成され、このプリンタ・ドライバは指定された定義ファイルを使用してプリンタのプロパティにアクセスします。

5.1.3 印刷のサポート

Oracle Reports は、UNIX の次の印刷標準をサポートしています。

- PostScript レベル 1 および 2
- PCL レベル 3
- ASCII (キャラクタ・モードでの印刷用)

Oracle Reports での印刷には、これらのバージョンに対応しているプリンタを使用してください。

5.2 UNIX でのプリンタの設定

この項では、次の項目について説明します。

- [UNIX でのプリンタのインストール](#)
- [Oracle Reports で使用するプリンタの設定の確認](#)

5.2.1 UNIX でのプリンタのインストール

プリンタ・キューのインストール方法は、UNIX のどのプラットフォームを使用しているかによって少し違いがあります。プラットフォームによっては、インストールに役立つインタフェース・ツールが付属しています。プリンタ・キューを追加する手順は、UNIX プラットフォームのマニュアルを参照してください。

Solaris プラットフォーム上でプリンタ・キューを追加するスクリプトの例を次に示します。この例では、ドメイン情報 `expldomain` と、プリンタ名 `printer1` および `printer2` をハード・コーディングしています。プリンタは、Xerox DCS モデルです。

```
#!/bin/sh
echo "Please enter the Printer Name Either printer1 or printer2\n"
read PRINTER
LOGFILE=/var/adm/config.log
PATH=/usr/bin:/usr/sbin:$PATH
export PRINTER LOGFILE PATH
lpsystem -t bsd expldomain >$LOGFILE 2>&1
lpadmin -p "$PRINTER" -s expldomain!"$PRINTER" -I any >$LOGFILE 2>&1
mkdir -p /usr/Xerox_DCS /usr/Xerox_DCS/original
chown -R 755 /usr/Xerox_DCS /usr/Xerox_DCS/original
cp /usr/bin/lp /usr/Xerox_DCS/original
mv /usr/bin/lp /usr/bin/lp.Xerox
ln -s /tmp /usr/Xerox_DCS/tmp
echo "$PRINTER" > /usr/Xerox_DCS/printer.db
cp /usr/local/packages/dc99cc23.txt /usr/Xerox_DCS
ln -s /usr/Xerox_DCS/dc99cc23.txt /usr/bin/lp
lpadmin -d "$PRINTER"
```

5.2.2 Oracle Reports で使用するプリンタの設定の確認

プリンタ・キューが正しくインストールされたことを確認する手順は次のとおりです。

1. インストールしたプリンタ・キューで使用される PPD ファイルまたは HPD ファイルが、次の場所にあることを確認します。

```
ORACLE_HOME/guicommon/tk/admin/PPD
ORACLE_HOME/guicommon/tk/admin/HPD
```

2. プリンタにインストールされているフォント・メトリックの AFM ファイルまたは TFM ファイルが、次の場所にあることを確認します。

```
ORACLE_HOME/guicommon/tk/admin/AFM
ORACLE_HOME/guicommon/tk/admin/TFM
```

5.3 印刷環境の構成

この項では、UNIX でのプリンタのインストール後に実行する構成の手順について説明します。

- [uiprint.txt ファイルの編集](#)
- [環境変数](#)
- [印刷プロパティのダイアログ・ボックス](#)

5.3.1 uiprint.txt ファイルの編集

第 5.1 項「UNIX での印刷の概要」で説明したように、Oracle Reports によって論理プリンタ・ドライバが作成されます。この内部的なプリンタ・ドライバが作成されるようにするには、使用可能なプリンタ・キューの名前、そのキューで使用されるドライバのタイプ、ドライバのバージョン、プリンタ記述ファイルなどの情報を指定する必要があります。uiprint.txt は、このような情報を伝えるためのメイン・ファイルです。このファイルは次の場所にあります。

`ORACLE_HOME/guicommon/tk/admin`

uiprint.txt はプリンタ構成ファイルです。Oracle Reports では、内部プリンタ・ドライバを作成するときにこのファイルが読み取られます。このファイルは、Oracle Reports のインスタンスごとに変更してください。

uiprint.txt 内のエントリのフォーマットは次のとおりです。

`Printer:DriverType:DriverVersion:PrinterDescription:PrinterDescriptionFile:`

指定のフォーマットで記述された uiprint.txt 内のこの 1 行のエントリによって、Oracle Reports で使用される 1 つのプリンタを定義します。各行に 5 つのフィールドがあり、それぞれコロンで区切られています。表 5-1 は、uiprint.txt のエントリの各要素の説明です。

表 5-1 uiprint.txt のエントリの要素

要素	説明
Printer	<p>プリンタ（またはプリンタ・キュー）の名前を指定します。この名前が <code>lpr</code> コマンドまたは <code>lp</code> コマンドで使用されます。</p> <p>使用可能なプリンタのリストを得るには、次のコマンドを使用します。</p> <p><code>lpstat -a</code></p> <p>プリンタのステータスを調べるには、<code>lpstat</code> コマンドを使用します。</p> <p>Solaris</p> <p><code>lpstat -p printername</code></p> <p>Linux</p> <p><code>lpstat -p printername</code></p> <p>HP-UX</p> <p><code>lpstat -d printername</code></p> <p>HP Tru64</p> <p><code>lpstat -p printername</code></p> <p>IBM AIX</p> <p><code>lpstat -pprintername</code></p> <p>IBM AIX では、<code>-p</code> の後に空白を入れしないでください。</p>
DriverType	<p>このプリンタで使用するプリンタ・ドライバのタイプを指定します。ドライバは、PostScript、PCL または ASCII です。</p>
DriverVersion	<p>使用するドライバ・タイプのバージョンを指定します。PostScript プリンタの場合は 1 または 2、PCL の場合は PCL Version 5 です。</p>
PrinterDescription	<p>プリンタの速度や場所など、プリンタの説明を指定します。この情報は、プリンタ関連のダイアログ・ボックスに表示されます。</p>

表 5-1 uiprint.txt のエントリの要素 (続き)

要素	説明
PrinterDescriptionFile	<p>このプリンタで使用されるプリンタ記述ファイルを指定します。タイプに応じて次のいずれかを指定します。</p> <ul style="list-style-type: none"> ■ PostScript プリンタを使用する場合は、PPD ファイルの名前を指定します。PPD は PostScript Printer Description の略です。指定の PPD ファイルが見つからない場合は、default.ppd が使用されます。Oracle Reports では、次の場所で PPD ファイルが検索されます。 ORACLE_HOME/guicommon/tk/admin/PPD ■ PCL プリンタを使用する場合は、HPD ファイルの名前を指定します。指定の HPD ファイルが見つからない場合は、ui4.hpd が使用されます。Oracle Reports では、次の場所で HPD ファイルが検索されます。 ORACLE_HOME/guicommon/tk/admin/HPD ■ ASCII プリンタを使用する場合は、ここには何も指定しません。このフィールドは、ASCII プリンタでは無視されます。

使用上の注意

- uiprint.txt 内のエントリのフィールドはすべて指定する必要があります。また、行の終わりにはコロンを付けます。
- uiprint.txt には、1 つ以上のエントリを定義する必要があります。または、関連するプリンタ変数 (TK_PRINTER および PRINTER) を設定します。これらが指定されていない場合は、Oracle Reports でプリンタ関連のタスクを実行することはできません。

関連項目： プリンタ関連の環境変数の詳細は、[第 5.3.2 項「環境変数」](#)を参照してください。

内部プリンタ・ドライバは、Oracle Reports の描画用サーフェスです。Oracle Reports では、このサーフェスが印刷に使用されるだけでなく、ファイルへの出力を生成する場合にも内部的に使用されます。したがって、uiprint.txt のエントリを正しく指定するか、プリンタ関連の環境変数のいずれかを設定しておく必要があります。ユーザーがプリンタを簡単に選択できるようにするために、ユーザーが利用可能なプリンタをすべて uiprint.txt に指定しておくことをお勧めします。

例：

uiprint.txt のエントリの例を 2 つ示します。

```
colprt14:PostScript:2:RMSC Atrium HPLaserJet5:default.ppd:
colprt15PCL:5:RMSC 1st Floor HPLaser4:ui4.hpd:
```

5.3.2 環境変数

この項では、印刷に関連する環境変数の一覧を示します。

関連項目： Oracle Reports で設定可能な環境変数の詳細は、[付録 B「環境変数」](#)を参照してください。

- [TK_PRINTER / PRINTER](#)
- [TK_PRINTER](#)
- [TK_PRINT_STATUS](#)
- [REPORTS_NO_DUMMY_PRINTER](#)
- [TK_HPD](#) および [ORACLE_HPD](#)

- [TK_PPD](#) および [ORACLE_PPD](#)
- [TK_TFM](#) および [ORACLE_TFM](#)
- [TK_AFM](#) および [ORACLE_AFM](#)

5.3.3 印刷プロパティのダイアログ・ボックス

UNIX では、Reports Builder の様々なダイアログ・ボックスを通してプリンタ関連の操作が行われます。

5.3.3.1 「ページ設定」ダイアログ・ボックス

「ページ設定」ダイアログ・ボックスでは、印刷されるページの外観を指定します。指定できるオプションは、使用するプリンタ・ドライバのタイプによって異なります。内部プリンタ・ドライバは、このダイアログ・ボックスで指定された情報（スケール、回転、幅、高さなど）を使用して、プリンタに出力するページのフォーマットを行います。

5.3.3.2 「ジョブの印刷」ダイアログ・ボックス

印刷ジョブの特性は、使用するプリンタ・ドライバに応じてそれぞれ異なります。「ジョブの印刷」ダイアログ・ボックスは印刷ジョブの実行直前に表示されるもので、ジョブをプリンタに送信するために必要な印刷ジョブ情報を入力できます。

5.4 プリンタ関連のファイル

この項では、様々な印刷関連のファイルについて説明します。ファイルの概要と、ファイルを一般的な印刷ニーズに合わせて編集する方法を示します。

- [ファイルの概要](#)
- [PPD ファイル](#)
- [HPD ファイル](#)
- [フォント・メトリック・ファイル](#)
- [uifont.ali](#)
- [uiprint.txt](#)
- [プリンタ関連ファイルの編集](#)

5.4.1 ファイルの概要

表 5-2 は、UNIX での印刷時に Oracle Reports で使用されるファイルの一覧です。

表 5-2 プリンタ関連ファイルの概要

ファイル名 / 拡張子	説明
.ppd	PostScript プリンタ定義ファイル
.hpd	HP glue ファイル
.afm	Adobe フォント・メトリック・ファイル
.tfm	PCL フォント・メトリック・ファイル
uifont.ali	フォント・エイリアシング・ファイル
uiprint.txt	プリンタ構成ファイル

5.4.2 PPD ファイル

PostScript は、Adobe 社が開発したページ記述プログラミング言語です。PPD ファイルは、Oracle Reports などのアプリケーションで使用できるプリンタの機能を定義するものです。たとえば、使用可能な用紙トレイ、使用可能な用紙サイズ、用紙の物理的サイズ、使用可能なフォントなどが PPD ファイルで定義されます。現時点では、Oracle Reports によってこのファイルから読み取られる情報は、プリンタの用紙サイズ、プリンタで使用可能なフォントおよびデフォルトの解像度です。将来的には、このほかにイメージ分割を正しく行うためのメモリーなどの情報も使用することが考えられます。

PPD ファイルを変更するのは、新たに追加したフォントまたはメモリーを Oracle Reports が認識できるようにする場合のみです。DefaultPageSize を変更して、優先するページ・サイズを指定することもできます。

注意： ページ・サイズでは、PPD の他のエントリと同様に大文字と小文字が区別されます。基本的に、PPD ファイルの他のエントリは変更しないでください。

「プリンタの選択」ダイアログ・ボックスで、uiprint.txt に記述されていないプリンタを選択する場合や、プリンタのタイプを PostScript タイプに変更する場合は、そのプリンタの PPD ファイルを指定するよう要求されます。使用するプリンタに最も類似度の高いプリンタの PPD ファイルを選択する必要があります。PPD ファイルには、一般に、プリンタのモデル名に似た名前が付けられています。

uiprint.txt では、プリンタごとに PPD ファイルを指定する必要があります。現行のプリンタに対する PPD ファイルの指定が正しくない場合は（PPD ファイルが見つからない、PPD ファイルのフォーマットが不正など）、そのプリンタに対しては default.ppd が使用されます。default.ppd は、デフォルトとなる可能性が最も高いプリンタであるローカル・プリンタの設定を反映する PPD ファイルのコピーとなるように設定してください。

Oracle Reports には、一般的な PPD ファイルのセットが付属していますが、場合によってはプリンタ専用の PPD ファイルをベンダーから入手する必要があります。表 5-3 は、Oracle Reports に付属する PPD ファイルの例です。

表 5-3 Oracle Reports に付属の一般的な PPD ファイル

PPD ファイル名	対応するプリンタ
app1230.ppd	Apple LaserWriter v23.0
datap462.ppd	Dataproducts LZR-2665
declps32.ppd	Digital PrintServer 40
default.ppd	デフォルトのレベル 1 PostScript プリンタ
hpljet41.ppd	HP LaserJet 4/4M PostScript 600DPI
lwntx470.ppd	Apple LaserWriter II NTX
nccps801.ppd	NEC Colormate PS/80
tkphzr33.ppd	Tektronix Phaser III PXi v2011.108
1530_523.ppd	Linotronic 530
screenprinter.ppd	UNIX でプリンタが使用できない場合に使用されるデフォルトの PPD ファイル

Oracle Reports 付属の PPD ファイル以外の PPD ファイルが必要な場合は、次のいずれかを行ってください（推奨する順に示します）。

- PPD ファイルについてプリンタのベンダーに問い合わせます。
- Adobe 社の Web サイトから PPD ファイルをダウンロードします。
- 既存の PPD ファイルをコピーして編集します。
- Adobe 社に PPD の仕様を問い合わせ、PPD ファイルを作成します。

PostScript ファイルに含まれているのはフォント情報のみで、フォント・メトリックは含まれていません。Oracle Reports は、インストールされている AFM ファイルを参照してフォント・メトリック情報を検索します。この AFM ファイルは、フォントのベンダーから提供されるものです。Oracle Reports には、最も一般的に使用されるフォントの AFM ファイルが付属しています。Oracle Reports によって生成された PostScript ファイルが正しく印刷されるようにするには、要求されるフォントがプリンタにインストールされている必要があります。

5.4.2.1 ローカルでの PPD ファイルのカスタマイズ

PPD ファイルは、特定のプリンタの機能を表す静的な情報です。このファイルには、製品出荷時の初期設定が行われています。プリンタをインストールすると、増設メモリー、用紙トレイ、フォントなどがプリンタに追加されることもあります。デバイスを管理する作業は動的で、ディスクにダウンロードされたフォント、エラー・ハンドラ、RAM ベースのフォントとプロシージャ・セット、デフォルト・デバイス設定などを把握する必要があります。この種のデバイス管理は、PPD ファイルで対応できる範囲を越えています。ただし、一定の規則に従えば、PPD ファイルに記述されている情報をカスタマイズして、プリンタの各ローカル・インスタンスに適合させる、あるいは特定のアプリケーションに適合させることも可能です。

元の PPD ファイルを変更するかわりに、新しいファイルを作成して、いくつかのパラメータに関してはローカルでカスタマイズし、それ以外の情報については主ファイルを参照するという方法も考えられます。ローカルでのカスタマイズを行うファイルには、主 PPD ファイルへの参照を次のフォーマットで記述する必要があります。

```
*Include: "filename"
```

ここで、filename は主 PPD ファイルの名前です。このように参照していれば、後で主 PPD ファイルの置換が行われても、ローカルのカスタマイズ・ファイル編集する必要はありません。*Include キーワードで参照されるファイルは、参照元ファイル（ローカルのカスタマイズ・ファイル）内にあるものとして扱われます。

たとえば、default.ppd ファイルが次のように定義されているとします。

```
*PPD-Adobe: "4.0"
*Include: "datap462.ppd"

*% Page definitions
*DefaultPageSize: Letter
.....
*DefaultPageRegion: Letter
```

主 PPD ファイルは datap462.ppd です。

管理者はインクルードするファイルの名前は、サイトのデフォルト・プリンタのタイプに合わせて変更する必要があります。

ローカルのカスタマイズ・ファイルで主 PPD ファイルをインクルードする場合、最終的なファイル内で同じキーワードのインスタンスが複数存在することもあります。したがって、カスタマイズ・ファイルでの主ファイルの位置（先頭または末尾）は重要で、カスタマイズ・ファイルによる変更に影響します。

5.4.3 HPD ファイル

HPD ファイルは PCL プリンタの機能を定義するもので、その役割は PostScript プリンタに対する PPD ファイルに似ています。HPD または HP glue ファイルには、特定の PCL プリンタで利用可能なフォントの情報が含まれます。HPD ファイルのフォーマットは、HP 社の『PCL5 Developer's Guide』を参照してください。

PostScript の場合に AFM ファイルを使用するのと同様に、HP フォントにはそれぞれ対応する TFM ファイルが必要です。TFM ファイルはフォント・ベンダーから提供されており、TFM ファイルをインストールすると、新しく追加されたフォントがプリンタの HPD ファイルへ追加されます。新しいフォントについては、HPD ファイルで次のフィールドを指定します。

```
FONT={fontname}  
/tfm={tfm-filename}
```

ここでは

fontname は、フォントの名前です。

tfm-filename は、TFM ファイルのベース・ファイル名です。

TFM ファイルのみでは指定が不十分な場合は、FONT フィールドの後に次のように指定することもできます。

```
/ptsize={size {size ...}}
```

指定したフォントがビットマップ・フォントであるにもかかわらず、TFM にスケーラブル・フォントとして指定されている場合は、使用可能なサイズを次のように指定することによって、ポイント・サイズを制限することができます。

```
/symset={symset {symset ...}}
```

これにより、指定したシンボル・セットのみがサポートされます。認識されるシンボル・セットの一覧は、HP 社の PCL に関するマニュアルを参照してください。

Oracle Reports では、PCL フォーマットで印刷する場合に defaultpaper フィールドを指定することもできます。このフィールドに設定した defaultpaper は、ツールキットによって使用されます。このフィールドのフォーマットは次のとおりです。

```
<defaultpaper={papername}>
```

たとえば、次のように指定した場合は用紙名が A4 に設定されます。

```
<defaultpaper=A4>
```

用紙名では、大文字と小文字は区別されません。defaultpaper を 2 回以上指定した場合は、最後の defaultpaper が使用されます。プリンタで使用できない用紙名を指定した場合、defaultpaper は無視され、かわりに LETTER が用紙名として使用されます。同様に、用紙名が正しくない場合も LETTER が使用されます。

5.4.4 フォント・メトリック・ファイル

Oracle Reports では、次の 2 種類のフォント・メトリック・ファイルがサポートされています。

- [AFM ファイル](#)
- [TFM ファイル](#)

5.4.4.1 AFM ファイル

AFM ファイルには、それぞれ単一のフォントに対するフォント関連のメトリックが格納されています。このメトリックには、スタイル、太さ、幅、キャラクタ・セットなどの様々なフォント属性が含まれています。AFM ファイルおよび AFM ファイル・フォーマットの説明は、一般に、フォントまたはプリンタのベンダーから入手できます。

AFM ファイルをインストールするには、第 5.2.2 項「[Oracle Reports で使用するプリンタの設定の確認](#)」で説明している AFM ファイルの場所にファイルをコピーします。ファイル名は、.afm 拡張子を付けずに、フォントの名前に一致している必要があります。たとえば、フォント名が CodedreineunBold の場合、ファイル名は CodedreineunBold とする必要があります。

フォント名を確認するには、AFM ファイル内で fontname という文字列を検索します。AFM ファイルはフォント・ファイルではなくメトリック・ファイルであることに注意してください。このファイルは、プリンタ用に文字を正しくフォーマットするための情報を伝えるものです。フォントの AFM ファイルはあるけれども、そのフォントがプリンタ上に存在しない場合は、フォント・メトリックの不一致が発生するため、Oracle Reports によって正しく出力を生成することはできません。レポートの設計に使用したフォントがプリンタでも使用可能であることを確認する必要があります。

5.4.4.2 TFM ファイル

PCL では、HPD ファイルおよび TFM ファイルが使用されます。HPD ファイルには、プリンタで使用できるフォントのリストが含まれます。フォントはそれぞれの TFM ファイルを参照します。TFM ファイルの役割は Adobe の AFM ファイルと同じで、個々のファイルには単一のフォントに関する情報が列記されています。HPD ファイルは編集可能な ASCII ファイルですが、TFM ファイルはバイナリ・ファイルであるため編集できません。

Oracle Reports で使用する新しいフォントが PCL 出力で正しく表示されるようにするには、プリンタ用の HPD ファイルと TFM ファイルを用意する必要があります。既存の HPD ファイルをコピーして新たに作成することができますが、使用するプリンタに適したものであることを確認したうえで行ってください。HPD ファイルには、プリンタで使用できるフォントを指定する必要があります。

Oracle Reports には、一般的な TFM ファイルのセットが付属しています。別のフォント・メトリック・ファイルをプリンタで使用する場合は、フォントまたはプリンタのベンダーから入手してください。TFM ファイルをインストールするには、第 5.2.2 項「[Oracle Reports で使用するプリンタの設定の確認](#)」で説明している TFM ファイルの場所にファイルをコピーします。

5.4.5 uifont.ali

uifont.ali ファイルは、Oracle Reports で使用されるフォント・エイリアスを定義するものです。あるフォントが使用できない場合にどのフォントで置き換えるかを定義できるため、クロスプラットフォーム開発を行う場合はこのツールが非常に役立ちます。uifont.ali は次の場所にあります。

```
ORACLE_HOME/guicommon/tk/admin
```

フォントのエイリアスを指定するには、次の構文を使用します。

```
source_font = destination_font
```

各フォントの次の属性も指定できます。

```
face.size.style.weight.width.character_set
```

複数のスタイルを組み合わせることもできます。その場合は、プラス記号 (+) を使用してスタイルを区切ります。次に例を示します。

```
Arial.Italic+Overstrike = Helvetica.12.Italic.Bold
```

このエントリは、イタリック (Italic) と打消し線 (Overstrike) の両方のスタイルが指定されている Arial フォントが、12 ポイント、太字 (Bold)、イタリック (Italic) の Helvetica フォントにマップされることを表します。プラットフォームおよびサーフェス (プリンタまたはシステム) によっては、フォント・フェースの大文字と小文字が区別されることがあります。

関連項目： フォント関連の詳細は、第 4 章「Oracle Reports でのフォントの管理」を参照してください。

5.4.6 uiprint.txt

uiprint.txt を利用すると、プリンタ・キューに関する詳細情報、たとえばプリンタ・ドライバのタイプやプリンタの説明などを簡単に定義できます。uiprint.txt は、Oracle Reports のインスタンスごとに編集してください。

関連項目： uiprint.txt の詳細は、第 5.3.1 項「uiprint.txt ファイルの編集」を参照してください。

5.4.7 プリンタ関連ファイルの編集

次の各項では、様々なプリンタ関連のファイルを編集する方法を説明します。

- [PPD ファイルの編集](#)
- [PCL 印刷を行うための HPD ファイルの編集](#)

5.4.7.1 PPD ファイルの編集

場合によっては、PPD ファイルに記述されている属性を変更する必要があります。次の各項では、一般に変更が必要となるいくつかの属性について説明します。

- [デフォルトの用紙サイズの変更](#)
- [プリンタのマージン設定の変更](#)
- [PPD ファイルへの新規フォント・エントリの追加](#)
- [プリンタ・トレイ設定のオーバーライド](#)

5.4.7.1.1 デフォルトの用紙サイズの変更 一部のレポートで、用紙サイズを A4 にしなければならないとします。UNIX プラットフォームでは、プリンタ・ドライバは uiprint.txt で指定されており、デフォルトの用紙サイズが A4 に設定されているとは限りません。たとえば、hpljet41.ppd ではデフォルトの用紙サイズは LETTER です。各プリンタ・キューのデフォルト用紙サイズ設定は、対応する PPD ファイルから取得されることに注意してください。

A4 をデフォルトの用紙サイズとして設定する手順は次のとおりです。

1. uiprint.txt を編集し、ページ・サイズとして A4 をサポートしている PostScript プリンタ定義ファイル (拡張子は .ppd) をインクルードします。たとえば、hpljet41.ppd をインクルードします。

2. バックアップとして、hpljet41.ppd をコピーします。

3. 次のように、uiprint.txt にエントリを追加します。

```
Printer_name: PostScript:1: the printer on floor1:hpljet41.ppd
```

4. hpljet41.ppd を編集し、次のように設定を変更します。

```
DefaultPageSize: A4
DefaultPageRegion: A4
DefaultImageableArea: A4
DefaultPaperDimension: A4
```

5.4.7.1.2 プリンタのマージン設定の変更 マージンを変更するには、PPD ファイルの `ImageableArea` セクションを変更します。`ImageableArea` では、指定されたページ・サイズに対してプリンタで印字可能な領域を示す境界を設定します。以後、この指定されたページ・サイズの名前を `mediaOption` とします。デバイスでサポートされている名前付きページ・サイズごとに1つの文を記述します。`*DefaultImageableArea` には、デフォルトの印字可能領域の `mediaOption` 名があります。デフォルトのページ・サイズとして指定できるのは1つのみです。したがって、この値は `*DefaultPageSize`、`*DefaultPageRegion` および `*DefaultPaperDimension` と同じです。

印字可能領域の定義の構文は次のとおりです。

```
*ImageableArea mediaOption: "llx lly urx ury "  
*DefaultImageableArea: mediaOption | Unknown
```

ll は lower left (左下) の略、ur は upper right (右上) の略です。`*ImageableArea` の枠ボックスの値は、4つの実数として指定します。これらの数値は、PostScript 言語のデフォルトのユーザー空間座標系における、領域の左下角と右上角の x 座標および y 座標です。指定ページ・サイズの x 軸および y 軸は、`*PaperDimension` エントリでのそのページ・サイズの x 軸および y 軸に対応しています。

印字可能領域は、プリンタによって実際に印刷可能な、ページ内の部分として定義します。ページ・サイズが同じでも、プリンタによって印字可能領域が異なることがあります。これには、現行の解像度、メモリの大きさ、印刷の向きなどの要素が影響するためです。PPD ファイルで、あるページ・サイズに対する印字可能領域の大きさが複数設定されている場合、記録される印字可能領域の大きさは、そのページ・サイズに対して考えられるすべての印字可能領域の和部分となります。この方式により、印字可能領域が PPD ファイルに記述されているものより小さくなることはなく、印字可能領域内に出力されたものはすべて目視できるようになります。したがって、現行の構成での印字可能領域が、実際には PPD ファイルに記述されている印字可能領域よりも大きいということもありません。

次の表は、デバイスでの特定のページ・サイズを表すために、`mediaOption` で指定可能なオプションのキーワードです。

表 5-4 mediaOption のキーワード

mediaOption (用紙サイズ)	サイズ (ポイント)	サイズ (mm)	サイズ (インチ)
Letter	612 * 792	215.9 * 279.4	8.5 * 11
Legal	612 * 1008	215.9 * 355.6	8.5 * 14
Ledger	1224 * 792	431.8 * 279.4	17 * 11
Tabloid	792 * 1224	279.4 * 431.8	11 * 17
A3	842 * 1191	297 * 420	11.69 * 16.54
A4	595 * 842	210 * 297	8.27 * 11.69
A5	420 * 595	148 * 210	5.83 * 8.27
B4	729 * 1032	257 * 364	10.12 * 14.33
B5	516 * 729	182 * 257	7.17 * 10.12

例

default.ppd で A4 ページのマージンを変更する手順は次のとおりです。

1. 次の各セクションで、デフォルト・ページを Letter から A4 に変更します。

```
*% Page definitions
*DefaultPageSize: A4
*PageSize A4: " "

*% These entries set up the frame buffer. Usually used with manual feed.
*DefaultPageRegion: A4
*PageRegion A4: "A4"

*% These provide the physical dimensions of the paper (by keyword)
*DefaultPaperDimension: A4
*PaperDimension A4: "595 842"
```

2. 次のセクションに、マージン定義を追加します。

```
*% Imageable (writable) areas for each page size, in pixels
*DefaultImageableArea: A4
*ImageableArea A4: "2 2 593 840 "
```

注意： PPD のエントリは、すべて大文字と小文字が区別されます。

5.4.7.1.3 PPD ファイルへの新規フォント・エントリの追加 PostScript プリンタに対して、Oracle Reports で使用できるフォントはそのプリンタで使用可能であることがわかっているもののみです。マルチプロセスのオペレーティング・システムにおいて、個人の要求に応じてプリンタを使用できることはほとんどないため、Oracle Reports ではすべてのフォントのリストが PPD ファイルから取得されます。

プリンタに新しいフォントがインストールされたときは、対応するフォント・エントリをプリンタの PPD ファイルに追加する必要があります。フォント・エントリのフォーマットは次のとおりです。

```
*Font {fontname}: {encoding} "({version})" {charset}
```

ここでは

{fontname} は、PostScript で定義されている Adobe フォント・フェース名です。

{encoding} は PostScript エンコード名です。

{version} は FontInfo のバージョン番号です。

{charset} は Adobe キャラクタ・セットです。

encoding の値の意味は、フォント・タイプによってわずかに異なります。エンコードを判定できない場合、エンコードの値は unknown に設定されます。各フォントの charset の値はそのフォントで可能なエンコードの種類を示しており、それによって通常はアプリケーションによってフォントの再エンコードが行われます。詳細は、Adobe 社から発表されている PPD の仕様を参照してください。

新しいフォントをプリンタに追加したときは、対応する AFM ファイルもフォント・メトリック・ディレクトリに追加する必要があります。AFM ファイルは、Oracle Reports が実際のフォント属性を取得して、印刷ページ上に正しくテキストを配置するために必要です。

例

新しいフォント `CodedreineunBold` を追加したとします。PPD ファイルを編集して、このフォントを追加する必要があります。

1. PPD ファイル内で、次の文字列を検索します。

```
*% Font Information
```

2. 新しいフォントに対する次のエントリを、パラグラフの最後に追加します。

```
*Font CodedreineunBold: Standard "(00.1001)" Standard ROM
```

5.4.7.1.4 プリンタ・トレイ設定のオーバーライド Oracle Reports によって生成される PostScript 出力には、トレイ情報が埋め込まれています。PPD ファイルにはデフォルト・トレイの定義があり、その後にプリンタで使用可能なトレイが定義されています。別のトレイに印刷するには、PPD ファイルの `DefaultInputSlot` エントリを更新します。

PPD ファイルで、デフォルトのトレイと有効な入力スロットが定義されているセクションを検索します。そのセクションは、通常は次のような行で始まっています。

```
*OpenUI *InputSlot: <PickOne>
```

デフォルト・トレイのエントリは、次のように定義されています。

```
*DefaultInputSlot: Lower
```

スロットの定義は、通常はデフォルト・トレイのエントリの後にあり、次のように指定されています。

```
*InputSlot Upper/Multipurpose Tray: "
```

```
...
```

```
*InputSlot Lower/Paper Cassette: "
```

このセクションは、次のような行で終了します。

```
*CloseUI: *InputSlot
```

`DefaultInputSlot` は、定義済スロットのリストの値のいずれかに設定できます。

5.4.7.2 PCL 印刷を行うための HPD ファイルの編集

場合によっては、HPD ファイルに記述されている属性を変更する必要があります。次の各項では、一般に変更が必要となるいくつかの属性について説明します。

- [用紙サイズの変更](#)
- [新規フォント・エントリの追加](#)

5.4.7.2.1 用紙サイズの変更 たとえば、用紙サイズを A4 に変更するには、次の行を使用する HPD ファイルに追加します。

```
<defaultpaper=A4>
```

5.4.7.2.2 新規フォント・エントリの追加 PostScript の場合に AFM ファイルを使用するのと同様に、Oracle Reports で HP フォントを使用するには、各フォントに対応する TFM ファイルが必要です。TFM ファイルは、フォントのベンダーから提供されます。新しいフォントをインストールしたときは、そのフォントを HPD ファイルに追加します。

新しいフォントについては、HPD ファイルで次の設定を指定します。

```
FONT={fontname}          # {fontname} is a descriptive name for the font
/tfm={tfm-filename}     # {tfm-filename} is the base filename for TFM file
```

注意： HPD ファイル内のフォント名のエントリは一意にする必要があります。

5.5 グローバリゼーション・サポート

この項では、Oracle Reports でのマルチバイト・キャラクタ・セットの印刷のサポートについて説明します。Oracle Applications とともにインストールおよび使用する場合にはのみ Oracle Reports でサポートされる、IX および PASTA ユーティリティについても説明します。

- マルチバイト・キャラクタ・セットの印刷
- IX および PASTA の概要

5.5.1 マルチバイト・キャラクタ・セットの印刷

現時点で、Oracle Reports では、PostScript 出力の Unicode キャラクタ・セットをサポートしていません。これにかわる方法として、Oracle Reports のマルチバイト・キャラクタ・セットをサポートする PDF 出力 (desformat=pdf) を利用して、印刷が可能です。

Oracle Reports では、マルチバイト・キャラクタ・セットの AFM ファイルに対応する一連のコード体系をサポートしています。

関連項目： フォント関連の詳細は、第 4 章「Oracle Reports でのフォントの管理」および第 6 章「Oracle Reports での PDF の使用」を参照してください。

PostScript レポート出力を印刷するプリンタに、フォントをインストールしておく必要があります。

例

中国語のキャラクタ・セットが含まれる PostScript 出力を生成するレポートを作成しているとします。まず、マルチバイト・キャラクタ・セットのコード体系に準拠した AFM ファイルと PPD ファイルを用意します。出力先のプリンタにも中国語フォントがインストールされている必要があります。これは、UNIX 上で Oracle Reports によって生成される PostScript ファイルにはフォントが埋め込まれないためです。PostScript ファイルには、フォント名と、AFM ファイルから取得したフォント・メトリックのみ含まれています。レポートの送信先であるプリンタに中国語フォントがインストールされていない場合、中国語の文字は正しく印刷されません。

5.5.2 IX および PASTA の概要

Oracle Applications とともにインストールおよび使用する場合、Oracle Reports には PostScript 出力へのフォント埋込みユーティリティが提供されます。

キャラクタ・モードのレポートでは、PASTA というユーティリティを使用します。ビットマップ・レポートの場合は、ユーティリティ IX を利用すると、PostScript 出力にフォントを埋め込むことにより、そのフォントがプリンタにインストールされていなくても印刷できるようになります。PASTA と IX はいずれも、Oracle Reports を Oracle E-Business Suite とともに使用する場合にはのみサポートされます。

キャラクタ・モード・レポートに対して PASTA を使用する場合は、タグ付きキャラクタ・モード出力 (適切な prt ファイルを介して生成) を PostScript 形式に変換したものが生成されません。IX を使用すると、UTF8 など、どのようなキャラクタ・セットでも、Oracle Reports から PostScript ビットマップ・レポートを PostScript プリンタに印刷することができます。この機能によって、Unicode だけでなく、UNIX でサポートされているすべての言語での PostScript による印刷が可能になります。IX ライブラリは、デフォルトでは Oracle Reports パッチによってオフに設定されています。

Oracle Reports とともに IX および PASTA を使用する時の設定および使用方法は、『Oracle Applications システム管理者ガイド』を参照してください。Oracle Metalink (<http://metalink.oracle.com>) のメンバーは、この情報を MetaLink ノート 189708.1 および 159225 から入手することもできます。

PASTA に関する問題がある場合は、次の方法で問題を特定してください。

1. PASTA 環境変数の設定を解除します。
2. 問題を引き起こした手順を再度実行してみます。
3. 環境変数が設定されていない状態で問題が再現する場合は、Oracle Reports の通常の印刷の問題として、このドキュメントで説明している診断手順に従ってください。

PASTA 環境変数が設定されている場合にのみ問題が再現する場合は、Oracle E-Business Suite のマニュアルに記載されている診断手順に従ってください。

5.6 デバッグ・オプション

この項では、UNIX での印刷の問題に対してデバッグを行うために Oracle Reports で使用できる各種環境変数および方法について説明します。

- [DEBUG_SLFIND](#)
- [TK_DEBUG_POSTSCRIPT](#)

5.6.1 DEBUG_SLFIND

この環境変数が設定されている場合、Oracle Reports では、検索されたファイルとその場所の一覧がファイル検出ルーチンによって出力されます。現行の構成が正しく機能しない場合は、この情報が非常に役立ちます。出力先には、ファイル、stdout (標準出力) または stderr (標準エラーへの出力) を選択できます。出力先として指定したファイルへの書き込みができない場合は、かわりに stderr が使用されます。

処理速度および出力の大きさの理由から、出力先はファイルとすることをお勧めします。DEBUG_SLFIND の出力の例を次に示します。生成されるデバッグ情報によって、設定に関する様々な問題、たとえば参照されている PPD ファイルや AFM ファイルの名前とその場所などを特定できます。

この出力には、次の内容がすべて含まれています。

- TK_PPD や TK_AFM などの各種環境変数とその値
- PPD や AFM などのリソース・ファイルとその場所 (これらのファイルが不足しているかどうかの判断に利用できます)
- ORACLE_HOME 以下の各種リソース・ファイルのデフォルトの場所

```
slsfindfile(): checking environment variable TK_PPD(8).
slsfindfile(): environment variable not set
slsfindfile(): checking environment variable ORACLE_PPD(10).
slsfindfile(): environment variable not set
slfpath(): looking up path
/oraclehome/guicommon/tk/admin/PPD/
slfexist(): testing /oraclehome/guicommon/tk/admin/PPD
slfexist(): testing /oraclehome/guicommon/tk/admin/PPD/default.ppd
slsfindfile():returned
/oraclehome/guicommon/tk/admin/PPD/default.ppd
slfindfile(): type = 39 (AFM)slfindfile(): name = Courier-Bold
slsfindfile(): checking environment variable TK_AFM(8).
slsfindfile(): environment variable not set
slsfindfile(): checking environment variable ORACLE_AFM(10).
slsfindfile(): checking ORACLE_HOME environment variable.
slsfindfile(): environment variable set to /oraclehome (len=18)
slfpath(): looking up path/oraclehome/guicommon/tk/admin/AFM/
slfexist(): testing /oraclehome/guicommon/tk/admin/AFM
slfexist(): testing /oraclehome/guicommon/tk/admin/AFM/Courier-Bold
slsfindfile():returned /oraclehome/guicommon/tk/admin/AFM/Courier-Bold
slfindfile(): name = uiprint.txt
slsfindfile(): checking ORACLE_HOME environment variable.
slfpath(): looking up path/oraclehome/guicommon/tk/admin/
```

```

slfexist(): testing /oraclehome/guicommon/tk/admin
slfexist(): testing /oraclehome/guicommon/tk/admin/uiprint.txt
slsfindfile(): returned /oraclehome/guicommon/tk/admin/uiprint.txt

```

5.6.2 TK_DEBUG_POSTSCRIPT

この環境変数は、Oracle Reports によって生成される PostScript 出力に影響を及ぼします。表 5-5 は、この変数の設定の説明です。

表 5-5 TK_DEBUG_POSTSCRIPT の設定

設定	説明
Functions (Func)	Functions を指定すると、コールされる個々のツールキット関数が PostScript 出力のコメントとして出力されます。
Long (L)	Long を指定すると、理解しやすい PostScript 出力が生成されますが、通常の PostScript 生成よりも時間がかかります。
Memory (Mem)	Memory を指定すると、各ページの下部にメモリ使用率が表示されます。

これらのオプションはいずれも、環境変数内で指定するときに 1 文字に短縮できます。またこれらのオプションは、"/" で区切って自由に組み合わせることができます。この環境変数では、大文字と小文字は区別されません。たとえば、すべてのオプションを使用するには、Func/L/Mem のように指定します。

注意： この環境変数を使用した PostScript 出力は、自身でのデバッグを目的としてのみ使用してください。この出力を、調査を目的としてオラクル社カスタマ・サポート・センターに提出する必要はありません。

5.7 よくある質問

この項では、UNIX での印刷時に発生する一般的な問題について説明します。

- 一般的な印刷エラー・メッセージ
- PCL の印刷の問題
- PostScript の印刷の問題
- フォントに関連する印刷の問題
- 印刷された出力の問題

注意： Reports Server では、印刷ジョブの送信に rwlpr 実行可能ファイルが使用されます。Windows の rwlpr ロギングでは、traceModule=all または traceModule=server を使用して Reports Server のトレースを有効化すると、destype=printer に対する印刷診断ログ (server_name-rwlpr-jobid.log) がログ・ディレクトリ (ORACLE_HOME¥reports¥logs¥server_name¥) に作成されます。このログ・ファイルには、スプーラの問題などの、印刷時の問題の診断に役立つメッセージに関する情報が記録されます。詳細は、[第 3.2.1.13 項「trace」](#) を参照してください。

5.7.1 一般的な印刷エラー・メッセージ

REP-00177: リモート・サーバーで実行中にエラーが発生しました

REP-01800: フォーマット・エラーが発生しました

REP-03300: 回復不能エラー

UI-9: このファンクション呼出しはコンテキストに合いません。

REP-3002: プリンタの初期化中に内部エラーが発生しました

原因:

これらのエラーは一般に、プリンタ構成の問題を示しています。

処置:

設定のオペレーティング・システム・レベルで定義されているプリンタ・キューを調べてください。次のコマンドを使用します。

- `lpc status`
- `lpstat -a`

有効なプリンタ・キューがインストールされている場合は、次の項目を調べてください。

- `uiprint.txt` に、プリンタの有効なエントリが必要です。
- Oracle Reports から、`uiprint.txt` ファイルを開いて読み取ることができる必要があります。
レポートを実行するユーザーには、`uiprint.txt` に対するオペレーティング・システム・レベルの読取り権限が必要です。Oracle Reports から `uiprint.txt` を開くことができる必要があります。UNIX オペレーティング・システムでは、ファイルを開く操作に関する制限があります。この制限を超えている場合は、Oracle Reports で `uiprint.txt` を開くことはできません。
- `uiprint.txt` で指定されているプリンタ記述ファイルが、次の場所に存在している必要があります。
`ORACLE_HOME/guicommon/tk/admin`
- `uiprint.txt` で指定されているプリンタが、オペレーティング・システム・レベルで使用可能に設定されている必要があります。これを簡単に調べるには、コマンドラインで `lp` または `lpr` を使用して任意のファイルを印刷します。これらのコマンドを使用して印刷することができ、プリンタに結果が出力される場合、そのプリンタは使用可能です。
- プリンタ・キューおよび `uiprint.txt` のエントリの構文が有効である必要があります。
プリンタの確認に失敗した場合は、付録 B「環境変数」の環境変数 `TK_PRINT_STATUS` および `REPORTS_NO_DUMMY_PRINTER` の説明を参照してください。

REP-00826: プリンタ・ドライバ: xxx (パラメータ: desformat で指定) は無効です。

REP-00177: リモート・サーバーで実行中にエラーが発生しました (CGI を介して実行した場合)

原因:

DESFORMAT の値が、指定のレポート実行モードに対して正しく指定されていません。

処置:

DESFORMAT パラメータでは、必要な出力フォーマットを指定します。有効なフォーマットは次のとおりです。

- ビットマップ・レポートの場合、Oracle Reports でサポートされている出力フォーマット (PostScript、PCL、PDF、HTML、XML、HTMLCSS) はいずれも DESFORMAT で有効です。この場合は、.PRT ファイル名は指定しません。出力先をファイルとして実行する場合、DESFORMAT パラメータは有効なプリンタ・キューに設定する必要があります。Oracle Reports は、プリンタに関連付けられているプリンタ定義ファイルを使用して出力をフォーマットします。
- キャラクタ・モード・レポートの場合は、DESFORMAT によって ASCII プリンタに対する出力が設定され、エスケープ文字が渡されます。キャラクタ・モード・レポートを実行する場合は、MODE パラメータを必ず Character に変更し、有効な .PRT ファイルを使用してください。

表 5-6 に、コマンドライン・オプション (DESTYPE、DESNAME および DESFORMAT) とプリンタ出力との対応を示します。

表 5-6 出力方法と DESTYPE、DESNAME、DESFORMAT の設定

出力方法	DESTYPE	DESNAME	DESFORMAT
ファイルへの生成	FILE	<i>file_name.ps</i>	<i>printer_name</i>
印刷	PRINTER	<i>printer_name</i>	
DISTRIBUTE=YES			<i>printer_name</i>
MODE=CHARACTER			<i>file_name.prt</i>

REP-1800: フォーマット・エラーが発生しました。

REP-00177: リモート・サーバーで実行中にエラーが発生しました (CGI を介して実行した場合)

原因:

このエラーは、UNIX サーバー上でプリンタ構成の問題が発生したことを示します。物理的に使用可能なプリンタがシステム上にない場合も、そのようなプリンタがあるものとして設定する必要があります。

処置:

1. uiprint.txt に有効なエントリがあることを確認します。
2. uiprint.txt に複数のプリンタ・キューのエントリがあり、デフォルトのプリンタを設定する必要がある場合は、環境変数が、uiprint.txt で指定されたプリンタのいずれかに設定されていることを確認してください。関連する環境変数が設定されていない場合は、uiprint.txt の最初のエントリが使用されます。プリンタ関連の環境変数の詳細は、付録 B 「環境変数」を参照してください。

システムで使用できるプリンタがない場合の代替手段は、第 5.3 項「印刷環境の構成」を参照してください。

名前にスペースがあるプリンタで出力する際のエラー

原因:

Solaris 2.8 の環境で名前にスペースがあるプリンタを使用すると、不具合によりエラーが発生する場合があります。このエラーは、プリンタ名の前後に引用符を付けて `lpr` または `lp` コマンドを実行することが原因です。

処置:

この問題を解決するには、次のいずれかの手順を実行します。

- プリンタ名からスペースを削除します。
 - プリンタ名に引用符を付けて `lpr` または `lp` コマンドを実行できるように修正されている Solaris 2.8 パッチを Sun 社から入手します。
- および
- `rwlp.sh` のセクションを変更し、名前にスペースがあるプリンタに引用符を付けて指定できるようにします。`rwlp.sh` ファイルは、`ORACLE_HOME/bin` ディレクトリにあります。

具体的には、次のように変更します。

```
#either LPR or LP Command was found
if [ -x $PRNCMDPATH ]
then
  if [ `basename $PRNCMDPATH` = "lpr" ]
  then
    #if [ `usr/bin/uname -r` = "5.8" ]
    #then
      $PRNCMDPATH `echo $@ | tr -d "¥"``
    #else
      $PRNCMDPATH "$@"
    #fi
  else
    # parse and Fix the command Line as Required by lp
    #if [ `usr/bin/uname -r` = "5.8" ]
    #then
      #getLpCommandLine `echo $@ | tr -d "¥"``
    #else
      getLpCommandLine "$@"
    #fi
    $PRNCMDPATH
  fi
  # exit with the command's exit code , This will tell the
  # server Print module if the command completed successfully
  # or not.
  exit $?
fi
done
```

Solaris 2.9 での印刷

Solaris 2.9 で DESTYPE=PRINTER および DESNAME=printer_name コマンドライン・オプションを使用してレポートを印刷すると、次のエラーが発生します。

```
REP-0069: Internal error
REP-57054: Inprocess job terminated with error
REP-50157: Error while sending file to printer 2op837a.Exit with error code 1
```

この問題を解決するには、次の手順を実行します。

注意： 手順を実行する前に、rwlpr.sh ファイルのバックアップを作成してください。Solaris では、rwlpr.sh は印刷スクリプト・ファイルで、ORACLE_HOME/bin/ ディレクトリにあります。このスクリプト・ファイルでは、デフォルトで lp と lpr コマンドがサポートされます。

1. ファイルの末尾にある次の行に移動します。

```
#either LPR or LP Command was found
```

2. 既存の if...else 条件に OR 演算子を追加します。

```
if [ `uname -r` = "5.8" ] || [ `uname -r` = "5.9" ] # If Solaris Release 5.8 / 5.9
...
else
# parse and Fix the command Line as Required by lp
...
if [ `uname -r` = "5.8" ] || [ `uname -r` = "5.9" ]# If Solaris Release
5.8/ 5.9
```

3. if...else if 条件では、Solaris リリースのバージョンがチェックされます。バージョン番号に基づいて、プリンタ名から引用符が削除され、print コマンドに渡されます。

5.7.2 PCL の印刷の問題

PC で印刷したときには灰色のフィールドが、UNIX PCL プリンタで印刷すると白くなるのはなぜでしょうか。

PCL カラー印刷はサポートされていません。パターンが透明に設定されている場合、PCL 印刷では白のペン（PCL 言語）が描画に使用されます。パターンがソリッド・パターンに設定されている場合は、黒のペンが使用されます。この処理は、フォアグラウンド・カラーやバックグラウンド・カラーの設定とは無関係に行われます。PostScript の印刷ロジックはこれとは異なります。パターンがソリッドのときはフォアグラウンド・カラーが使用され、パターンが透明のときはバックグラウンド・カラーが使用されます。

Oracle Reports ではどの PCL レベルがサポートされていますか。

現時点では、Oracle Reports の PCL ドライバは PCL レベル 3 の機能をサポートしています。それより後のバージョンの PCL の HPD ファイルはサポートしていますが、PCL レベル 3 以降に導入された新機能には対応していません。

5.7.3 PostScript の印刷の問題

PostScript プリンタで両面印刷を行うにはどうすればよいでしょうか。

両面オプションを持つプリンタがあり、適切な PPD ファイルがあるものとします。次に示す例は、京セラ社の FS-9000 プリンタ用の PPD ファイルを使用してテストした結果です。出力ファイルをフィルタ処理するために、UNIX の `sed` というツールも必要です。

両面印刷の問題とは、ジョブ・レベルでは両面印刷が有効であっても、ページ設定でリセットされてしまうというものです。これは、用紙サイズとプリンタ・トレイの情報がページごとに生成されることが理由です。この問題に対処するには、スクリプトを使用してページ・レベルの設定情報を取り除くことにより、両面設定のリセットを防ぐ必要があります。この対処方法を取る場合、ページ間のプリンタ・トレイの切替えはできなくなります。

1. 次の 3 行からなる `sed` スクリプトを作成します。

```
/^%%BeginPageSetup/, $ {
  /^%%BeginFeature/,/^%%EndFeature/d
}
```

2. このスクリプトを、`duplexsed` という名前のファイルに保存します。
3. `duplexsed` を、`ORACLE_HOME/bin` などの適切なディレクトリにコピーします。
4. 環境変数 `TK_PRINT` を次のように設定します。

```
TK_PRINT="sed -f $ORACLE_HOME/bin/duplexsed | lpr -l -s -P'%n' -#'%c'"
export TK_PRINT
```

注意： 印刷コマンドは、UNIX の種類によって異なります。使用するプラットフォームのインストール・ガイドおよびマニュアル・ページを確認してください。TK_PRINT の説明は、付録 B「環境変数」を参照してください。

TK_PRINT に格納されているコマンドが実行されるのは、`DESTYPE=PRINTER` の場合のみです。`DESTYPE=FILE` の場合は、PostScript ファイルにページ・レベルの設定情報が残ります。これを修正するには、PostScript ファイルに対して `duplexsed` スクリプトを実行します。

Oracle Reports ではどのレベルの PostScript がサポートされていますか。

Oracle Reports では、PostScript レベル 1 および 2 がサポートされています。

印刷ジョブの途中で動的にプリンタ・トレイの設定を変更するにはどうすればよいでしょうか。

場合によっては、レポートの途中でプリンタのトレイを切り替える必要があります。たとえば、レポートの最初のページをレターヘッド付きの用紙に印刷し、それ以降のページは無地の白い用紙に印刷する場合です。キャラクタ・モード・レポートでは、このような結果を得るには `.prt` ファイルを編集してレポートのプロパティを変更します。ビットマップ・レポートの場合は、`SRW.SET_PRINTER_TRAY` ビルトイン・プロシージャを使用します。UNIX では、この機能は PostScript 出力に対してはサポートされていますが、PCL 出力ではサポートされていません。PCL の場合、Oracle Reports では印刷方向と用紙トレイを変更するコマンドは無視されます。UNIX では、PCL の印刷方向とプリンタ・トレイを動的に変更することはできませんが、PCL の印刷ダイアログ・ボックスを使用して実行時に変更することができます。

Before Report、Between Pages またはフォーマット・トリガーを使用することにより、レポートのフォーマットに応じてプリンタ・トレイを切り替えることができます。この方法を利用すれば、同じレポートを異なる種類の用紙に簡単に印刷できます。

注意： SRW.SET_PRINTER_TRAY ビルトイン・プロシージャを含む SRW ビルトイン・パッケージの説明は、Oracle Reports のオンライン・ヘルプを参照してください。

例

BEFORE REPORT トリガーから、最初のページのプリンタ・トレイを設定します。

```
function BeforeReport return boolean is
begin
    srw.set_printer_tray('UPPER PAPER TRAY');
    return (TRUE);
end;
```

以降のページのプリンタ・トレイを動的に設定するには、レポートの各ページに印刷される項目にフォーマット・トリガーを追加します。次のコードは、偶数ページかどうかを調べて、その結果に応じてページ番号を設定します。

```
function B_tbpFormatTrigger return boolean is
page_num number;
begin
    srw.get_page_num(page_num);
    begin
    if mod(page_num, 2) = 0 then
        srw.set_printer_tray('UPPER PAPER TRAY');
    else
        srw.set_printer_tray('LOWER PAPER TRAY');
    end if;
    return (true);
end;
end;
```

Oracle Reports で生成された PostScript 出力を印刷するときに、外部印刷コマンドがトレイ選択オプションを無視するのはなぜでしょうか。

次の印刷コマンドを入力したとします。

```
- lp -dprinter -oupper $report_print_file1
```

この場合、lp コマンドの `-oupper` オプションは無視されます。これは、Oracle Reports によって生成される PostScript 出力の中にトレイ情報があるためです。PostScript でのトレイ選択は、コマンドラインでの指定よりも優先されます。コマンドラインのトレイ情報が優先されるようにするには、PostScript ファイルからトレイ情報を削除する必要があります。このようにするには、PostScript ファイル内の次の文字列を検索して削除します。

```
%%BeginFeature: *InputSlot name of printer tray
....
%%EndFeature
```

プリンタ・トレイ切替えの詳細は、「印刷ジョブの途中で動的にプリンタ・トレイの設定を変更するにはどうすればよいでしょうか。」を参照してください。

5.7.4 フォントに関連する印刷の問題

関連項目：

- フォント関連の詳細は、第4章「Oracle Reports でのフォントの管理」を参照してください。
- プラットフォーム間でのフォント問題への対応方法は、第7章「クロス・プラットフォームの移植問題の解決」を参照してください。

Oracle Reports での印刷時にどのフォントが使用されているかはどのようにして調べればよいでしょうか。

PostScript ファイルには、PPD ファイルを読み取った後、フォントのリストが作成されます。PostScript ファイルで次のタグを検索すると、フォントを確認できます。

- DocumentNeededResource には、PPD ファイル内で参照されるフォントのリストが記述されています。
- DocumentSuppliedResource には、PostScript ドライバによって対応する AFM ファイルを見つけることができたフォントのリストが記述されています。
- フィールド %IncludeResource:font の前の %%Page には、そのフィールドで使用されるフォントの名前が記述されています。

PCL 出力ファイルの場合は、特定のフォントが使用されているかどうかを確認できます。この情報に応じて、Oracle Reports またはプリンタのフォント設定を変更します。

レポートを実行して画面に出力するときとプレビューに出力するときでは、実際に何が異なるのでしょうか。

レポートを画面用にフォーマットした場合、つまり画面フォントを使用する場合は、レポート・エディタのペーパー・デザイン・ビューではレポートが正しく表示されます。ただし、画面フォントを使用してフォーマットされたレポートを印刷しようとする、その結果は多少異なることがあります。これは、画面フォントからプリンタ・フォントへのマップが不完全であることが多いからです。画面ではなくプレビューを選択した場合は、レポートのフォーマットにはプリンタ・フォントが使用され、画面に表示される出力は印刷した場合とほぼ同じになります。

有効なプリンタがインストールされていない場合、フォントに関連するなんらかの問題が発生しますか。

UNIX の場合、Oracle Reports 10g よりも前のリリースでは、イメージの作成やピクセル解像度の取得のためにウィンドウ表示システムの表示面を使用するには、DISPLAY 環境変数を設定する必要がありました。Oracle Reports 10g では、この依存関係は取り除かれています。

また、以前のリリースでは、フォントについて UNIX 上で有効なプリンタが必要でした。有効なプリンタがない場合、OracleAS Reports Services ではスクリーン・フォントが使用され、その際にも DISPLAY 環境変数の設定が必要でした。今回のリリースでは、OracleAS Reports Services に組み込まれているデフォルトの画面プリンタ・サーフェス (ScreenPrinter) は、プリンタが使用できない場合、画面またはプリンタをエミュレートしてフォントを入手します。したがって、UNIX 上にプリンタが存在している必要がなくなりました。

関連項目： 第3.10項「UNIX での DISPLAY およびプリンタ依存性の解消」

5.7.5 印刷された出力の問題

レポートを画面で表示すると問題ないけれども、印刷するとデータが切り捨てられるのはなぜでしょうか。

フィールドの切捨ての原因は多数考えられます。

- そのフィールドが拡張可能かどうかを調べます。
 1. Reports Builder のペーパー・デザイン・ビューまたはペーパー・レイアウト・ビューでフィールドをダブルクリックして、プロパティ・インスペクタを表示します。
 2. 「水平拡張度」プロパティを確認します。
 3. 「固定」が選択されている場合は、設定を「可変」または「拡張」に変更します。
 4. 出力先をプリンタとしてレポートを実行します。
 5. まだ切捨てが発生する場合は、フィールドに複数行が必要であることが考えられます。
 6. このフィールドのプロパティ・インスペクタに戻り、「垂直拡張度」を調べます。
 7. 「固定」が選択されている場合は、設定を「可変」または「拡張」に変更します。
 8. 出力先をプリンタとして再度レポートを実行します。
- ページ右端のフィールドが常に切り捨てられる場合は、プリンタの印刷可能領域の問題が考えられます。PCL プリンタを使用する場合は、印刷可能領域の大きさを計算し、それに応じてマージンのサイズを変更する必要があります。
 1. Reports Builder でレポートを開きます。
 2. ペーパー・レイアウト・ビューに移動します。
 3. 上部のツールバーにある「マージン」ツールをクリックします。レポート本体の終わり、つまりマージンの始まりを示す黒い太線が表示されます。
 4. 黒い線をクリックして、左に約 0.5 インチドラッグします。
 5. レポートを保存し、出力先をプリンタとして再度実行します。
 6. 必要であれば、ステップ 4 と 5 を繰り返して印刷可能領域の境界のおおよその位置を調べ、レポート本体がその中に確実に収まるようにします。
- PostScript プリンタを使用する場合は、次の手順に従い、ペーパー・レイアウト・ビューで印刷可能領域の境界を表示します。
 1. Reports Builder でレポートを開きます。
 2. 「ファイル」→「ページ設定」を選択します。
 3. マージンが小さいことと、印刷方向が正しいことを確認します。
 4. 「OK」をクリックします。これで、ペーパー・レイアウト・ビューで、境界の読取りができるようになります。
 5. ペーパー・レイアウト・ビューに移動します。
 6. 上部のツールバーにある「マージン」ツールをクリックします。レポート本体の終わり、つまりマージンの始まりを示す黒い太線が表示されます。黒い破線も表示されますが、これが印刷可能領域の境界を示します。
 7. 黒い太線が、黒い破線の内側にあることを確認します。そうでない場合は、黒い線をクリックして印刷可能領域の内側までドラッグします。
 8. 「マージン」ツールをクリックして、マージン・モードを終了します。
 9. 必要であれば、変更後の本体の境界内に収まるようにフィールドの位置を変更します。
 10. レポートを保存し、出力先をプリンタとして実行します。

- PCL プリンタでまだ切捨てが発生する場合は、プロポーショナル・フォントのかわりに固定幅フォントを使用してみてください。PCL プリンタでは、プロポーショナル・フォントの解釈時に問題が発生することがあり、これが原因で切捨てが発生します。Courier などの固定幅フォントを使用し、フォント・エイリアシングも試してみてください。

関連項目： フォント関連の詳細は、[第 4 章「Oracle Reports でのフォントの管理」](#)を参照してください。

注意： デフォルト・レイアウトは、一般的なプリンタ向けに作成されています。また印刷可能領域は、プリンタによって異なります。したがって、プリンタに合わせてレポートの設定を変更しなければならないこともあります。理想的には、多数のプリンタを使用することがわかっている場合は、初めから印刷可能領域が最も小さいプリンタに合わせてレポートを設計します。

Oracle Reports での PDF の使用

Adobe Portable Document Format (PDF) は、文書の作成に使用したアプリケーションやプラットフォームの種類に関係なく、あらゆるソース文書のフォント、書式、グラフィックス、カラーをすべて保持する汎用ファイル形式です。Oracle Reports は、いち早くこのテクノロジーを採用し、高品質の PDF 文書を生成してきたレポート生成ツールの 1 つです。

Oracle Reports 10g リリース 2 (10.1.2) では、PDF フォント・サブセットの機能が拡張され、フォントのサポートが改善され、次のような機能を実現しています。

- True Type フォントの直接サブセット (Type0 フォントとして)。Type3 フォントへの変換はなくなりました。その結果、PDF 出力はこれまでより鮮明かつ滑らかになり、検索やアクセスがしやすくなりました。
- True Type Collection (TTC) のサポート改善。これにより、選択する TTC フォント・ファイルを示すゼロベースのインデックスを使用できます。以前のリリースでは、Oracle Reports は True Type Collection (TTC) の最初のフォント・ファイルを選択していました。

この章の主な項は、次のとおりです。

- [Oracle Reports に含まれる PDF 機能](#)

この項では、Oracle Reports でサポートする各種の PDF 機能について説明します。これには、圧縮、フォント・エイリアシング、フォント・サブセット、フォントの埋込み、アクセシビリティ、分類などが含まれます。

- [プラットフォーム間の配布時における PDF フォントの問題への対応](#)

この項では、Windows プラットフォームでレポートを設計し、それを UNIX プラットフォームへ配布するときに発生する、PDF のフォントの問題の解決方法について説明します。

- [Unicode PDF ファイルの生成](#)

この項では、Unicode キャラクタ・セットを使用して PDF ファイルを生成する方法について説明します。

- [双方向 \(BiDi\) PDF ファイルの生成](#)

この項では、ヘブライ語やアラビア語などの双方向 (Bidirectional: BiDi) 言語を使用して PDF ファイルを生成する方法について説明します。

- [マルチバイト PDF ファイルの生成](#)

この項では、マルチバイト・フォントを使用して PDF ファイルを生成する方法について説明します。

- [バーコード PDF ファイルの生成](#)

この項では、バーコードを含む PDF ファイルを生成する方法について説明します。

6.1 Oracle Reports に含まれる PDF 機能

Oracle Reports は、PDF1.4 をサポートし、あらゆるプラットフォームで高品位の PDF レポートを生成できます。Oracle Reports でサポートする PDF 機能は、次のとおりです。

- 圧縮
- フォント関連の機能
- 実行の優先順位
- アクセンビリティ
- 分類
- グラフのサポート

6.1.1 圧縮

PDF 圧縮は、PDF ファイルのサイズを減らすことによって、PDF ファイルのダウンロードに要する時間を短縮します。

圧縮によって節約できる容量は、レポートの内容（たとえば、イメージの数とサイズ）によって変わります。

- **イメージ**: PDF 圧縮は、イメージを含むファイルのサイズにはあまり作用しません。これは、イメージ・ファイルが通常、すでに圧縮されているためです。
- **フォーマットされたデータ**: 高度にフォーマットされたデータは、高い圧縮率を見込めます。ただし、実際の圧縮率はレポートによって異なります。

圧縮されたファイルのサイズは、元のファイルの約 5 分の 1 になります。圧縮テストでは、生成されたファイルが元のファイルの 2 分の 1 から 8 分の 1 のサイズになり、ファイルの内容によってばらつきがあることがわかりました。

6.1.1.1 設定

デフォルトでは、Oracle Reports によって生成される PDF 出力は圧縮されています。圧縮のレベルを指定するには、コマンドラインで PDFCOMP を使用します。詳細は、[第 A.3.83 項「PDFCOMP」](#) を参照してください。

圧縮ファイルはダウンロードが短時間で済みますが、圧縮ファイルの生成にかかる時間は、非圧縮ファイルの場合よりかなり長くなります。

図 6-1 圧縮出力と非圧縮出力

名前	サイズ	種類	更新日時
pdf_no_comp.pdf	725 KB	Adobe Acrobat 文書	2005/09/30 12:25
pdf_yes_comp.pdf	119 KB	Adobe Acrobat 文書	2005/09/30 12:25

注意: 圧縮率は、レポートの内容に依存します。したがって、PDF ファイルのサイズも、PDF ファイルの生成に要する時間もレポートごとに異なります。

6.1.2 フォント関連の機能

この項では、Oracle Reports でサポートされる PDF フォント関連の機能について説明します。

- フォント・エイリアシング
- フォント・サブセット
- フォントの埋込み
- フォント機能のまとめ

6.1.2.1 フォント・エイリアシング

フォント・エイリアシングを使用すると、あるフォントを別のフォントに置き換えること（フォント置換）が可能です。このフォント置換は通常、プラットフォーム間でアプリケーション（ここでは PDF ファイル）を移植する場合に使用されます。その際には、キャラクタ・セットだけでなく、マルチバイト・フォントもエイリアシングできます。マルチバイト・アプリケーションの設計時におけるフォント・エイリアシングの考慮事項については、[第 18.2.1.2.2 項「フォント・エイリアシングの考慮事項」](#)を参照してください。

フォント・エイリアシングは、PDF ファイルを生成する時点で行われます。PDF ファイルには、出力の表示に必要なフォント情報だけが含まれます。使用されるフォントは、PDF ファイルに埋め込まれません。

注意： フォントは、PDF 出力を表示するマシンで使用可能である必要があります。PDF ファイルを生成するマシンでは、フォントが使用可能でなくてもかまいません。

レポートを表示する時点では、Adobe Acrobat は、次の規則に基づいて、エイリアシングされたフォントを置き換えます。

1. 出力を表示するマシンにフォントが存在しない場合、Adobe Acrobat は、そのフォントを Adobe Sans MM フォントに置き換えます。
2. Adobe Sans MM フォントが適合しない場合、出力には、データのかわりにドットが表示されます。

フォント・エイリアシングは、次のいずれか、またはすべてのフォントに対して機能します。

- シングルバイト・フォント。ASCII キャラクタ・セットおよび ISO-Latin キャラクタ・セットの東欧フォントを含みます。
- Adobe のマルチバイト Character ID (CID) フォント ([表 6-1](#)を参照)。これらは、Adobe のサイトから無料でダウンロードできます。
- Type1 PostScript フォント
- TrueType フォント

[表 6-1](#) は、マルチバイト・フォントの PDF フォント・エイリアシングに使用する Oracle NLS_CHARACTERSET、CMap 名、および CID フォント名間のマッピングを示しています。

表 6-1 PDF フォント・エイリアシングのための CID フォント・マッピング

言語	Oracle NLS_CHARACTERSET 名	CMap 名	CID フォント名
日本語	JA16SJIS	90ms-RKSJ-H	"KozMinPro-Regular-Acro" (*)
	JA16EUC	EUC-H	"HeiseiKakuGo-W5-Acro" (**)
			"HeiseiMin-W3-Acro" (**)
韓国語	KO16KSC5601	KSC-EUC-H	"HYSMyeongJoStd-Medium-Acro" (*)
	KO16MSWIN949	KSCms-UHC-H	"HYGothic-Medium-Acro" (**)
			"HYSMyeongJo-Medium-Acro" (**)
繁体字中国語	ZHT32EUC	CNS-EUC-H	"MSungStd-Light-Acro" (*)
	ZHT16BIG5、 ZHT16MSWIN950	ETen-B5-H	"MHei-Medium-Acro" (**)
			"MSung-Light-Acro" (**)
	ZHT16HKSCS	HKscs-B5-H	"MSungStd-Light-Acro" (*)
簡体字中国語	ZHS16CGB231280	GB-EUC-H	"STSongStd-Light-Acro" (*)
	ZHS16GBK	GBK-EUC-H	"STSong-Light-Acro" (**)

(*) これらのフォントは Adobe Acrobat Reader 5.0 以降で利用できます。

(**) これらのフォントは Adobe Acrobat Reader 4.0 以降で利用できます。

予期しないフォント・マッピングにより、マルチバイト・キャラクタが文字化けしないようにするため、Version 5.0 CID Fonts (*) の使用をお勧めします。Version 5.0 フォントは、Adobe Acrobat Reader 5.0 以降と互換性があります。

6.1.2.1.1 設定

フォント・エイリアシングには、コマンドライン・キーワードはありません。

フォント・エイリアシングのエントリは、uifont.ali ファイルに格納します。Oracle Reports では、uifont.ali ファイルのエントリが、生成された PDF ファイルに含まれるフォント情報と一致する場合にのみ、フォント・エイリアシングを行います。

注意： uifont.ali ファイルは、次のディレクトリに格納されています。

- ORACLE_HOME¥tools¥common (Windows の場合)
- ORACLE_HOME/guicommon/tk/admin (UNIX の場合)

uifont.ali ファイルは、Oracle Reports の PDF フォントに関する拡張機能すべてを制御する構成ファイルです。詳細は、[第 4 章「Oracle Reports でのフォントの管理」](#)を参照してください。

uifont.ali ファイル内の、フォント・エイリアシングのセクションは、[PDF] です。

uifont.ali ファイルのエントリは次のとおりです。

- シングルバイト・フォント

```
[PDF]
"font_name"="font_name"
```

注意： フォント名に複数の単語が含まれる場合は、そのフォント名のエントリを二重引用符で囲みます。たとえば、"Brush Script MT"。

- マルチバイト・フォント

```
[PDF]
character_set = "font_name"
```

または

```
"font_name"....character_set="font_name"
```

注意： フォント名に複数の単語が含まれる場合は、そのフォント名のエントリを二重引用符で囲みます。たとえば、"HeiseiKakuGo-W5-Acro"。

uifont.ali ファイル内の、フォント・エイリアシングのエントリの例を次に示します。

```
[ PDF ]
/*Alias TrueType to available Type1 font */
"Kino MT" = UtopiaBold
/*Alias multibyte to available CID font */
.....SJIS = "HeiseiKakuGo-W5-Acro"
```

ここでは

- "Kino MT" = UtopiaBold では、検出されたすべての Kino MT 文字を UtopiaBold 文字に置き換えます。

-SJIS = "HeiseiKakuGo-W5-Acro" では、検出されたすべてのマルチバイト・キャラクタ・セットを HeiseiKakuGo-W5-Acro (CID) のキャラクタ・セットに置き換えます。

6.1.2.1.2 トラブルシューティング

フォント・エイリアシングが機能しない場合は、次のことを確認します。

- Acrobat Reader 6.0 以降の場合は、「ファイル」→「文書のプロパティ」→「フォント」を選択します（それよりも前のリリースで Acrobat Reader 3.0 以降の場合は、「ファイル」→「文書情報」→「フォント」を選択します）。エイリアシングされたフォントがリストに追加されていることを確認します。そのフォントがリストに含まれていない場合は、フォント・エイリアシングが行われていません。フォントが検出されていないか、uifont.ali ファイルのエントリが正しくありません。
- レポート用に指定されたフォントが、レポートを表示するマシン上で使用可能であること。
- uifont.ali ファイルの [PDF] セクション名が変更されていないこと。Oracle Reports では、このセクション名を探してファイルを解析します。
- レポートの表示に使用している Adobe Acrobat Reader がバージョン 3.0 以降であること。マルチバイト・キャラクタのレポートを表示するには、最低でもこのバージョンが必要です。

6.1.2.2 フォント・サブセット

フォント・サブセットを使用すると、PDF のレンダリングに必要なフォント情報が PDF ファイルに含まれます。フォント・サブセットは、レポートの表示に使用されるマシンでそのフォントを使用できるかどうかには依存しません。PDF のフォント・サブセットは、シングルバイト・フォント、マルチバイト・フォントおよび Unicode フォントで使用可能で、マルチバイト・レポートの作成では、最もよく使用される方式です。

PDF ファイル内のフォントのサブセットを作成すると、レポート出力に必要な文字のみのフォント情報が PDF 出力に埋め込まれます。

Oracle Reports 10g リリース 2 (10.1.2) での PDF フォント・サブセットの拡張機能およびフォント・サポートの改善によって、これまでより鮮明かつ滑らかで、検索やアクセスもしやすい PDF 出力を生成できるようになりました。

注意： PDF ファイルの変更には、次のものがが必要です。

1. マシンにインストールされ、レポートに使用されているフォント
 2. PDF ライター
-
-

6.1.2.2.1 設定

フォント・サブセットを使用する前に、次の作業が必要です。

- フォント・ファイルのパスを REPORTS_PATH 環境変数に追加します。Oracle Reports では、PDF ファイルの生成時に、REPORTS_PATH 環境変数で指定されたパスでフォントを探します。
- フォント・サブセットのエントリは、uifont.ali ファイルに格納します。Oracle Reports では、uifont.ali ファイルにリストされているフォント・エントリが、生成中の PDF ファイルに存在する場合にのみ、フォント・サブセットを作成します。

注意： uifont.ali ファイルは、次のディレクトリに格納されています。

- ORACLE_HOME¥tools¥common (Windows の場合)
 - ORACLE_HOME/guicommon/tk/admin (UNIX の場合)
-
-

uifont.ali ファイル内の、フォント・サブセットのセクションは、[PDF:Subset] で、そのエントリは次のとおりです。

```
[PDF:Subset]
font_name = "font_file_name"
```

ここでは

font_name はフォント名です。このフォント名に複数の単語が含まれる場合は、必ず引用符で囲む必要があります。

font_file_name はフォント・ファイル名です。このファイル名は必ず引用符で囲む必要があります、大文字と小文字が区別されます。これが既存のフォント・ファイル名と完全に一致しない場合は、REP-1924 エラーが生成されます。

フォント・ファイルは、*ORACLE_HOME/reports/font_folder* など、任意のフォルダに保存できます。フォント・ファイルのパスを、*REPORTS_PATH* 環境変数に追加します。

注意： この *font_file_name* は、Reports Builder で表示されるフォント名ではありません。

例 1

```
[PDF:Subset]
Arial = "Arial.ttf"
```

TrueType Collection (.ttc) ファイル内にある TrueType フォントを使用する場合は、uifont.ali の [PDF:Subset] セクションのエントリの構文は次のようになります。

```
[PDF:Subset]
font_name = "ttc_file_name[,table_directory_number]"
```

ここでは

font_name はフォント名です。このフォント名に複数の単語が含まれる場合は、必ず引用符で囲む必要があります。

ttc_file_name は TrueType Collection ファイル名です。

table_directory_number は、TrueType Collection ファイル内の TrueType フォントに対する、ゼロベースのインデックスを使用したテーブル・ディレクトリ番号です（たとえば、"MS PGothic" = "msgothic.ttc,1" は、TrueType Collection ファイル内の 2 番目のフォントの使用を指定します）。*table_directory_number* が省略されているか、無効な値が指定されている場合は、TrueType Collection ファイル内の最初のフォント・プログラムのサブセットが使用されます。

例 2

```
[PDF:Subset]
"MS PGothic" = "msgothic.ttc,1"
"MS UI Gothic" = "msgothic.ttc,2"
```

表 6-2 に、Windows プラットフォームにおける東アジア言語用の一般的な TrueType Collection ファイルと、それに含まれるフォント名およびそのテーブル・ディレクトリ番号値を示します。

表 6-2 Windows プラットフォームにおける東アジア言語用の一般的な TrueType Collection ファイル

TTC ファイル名	フォント名	テーブル・ディレクトリ番号
batang.ttc	Batang	0
	BatangChe	1
	Gungsuh	2
	GungsuhChe	3

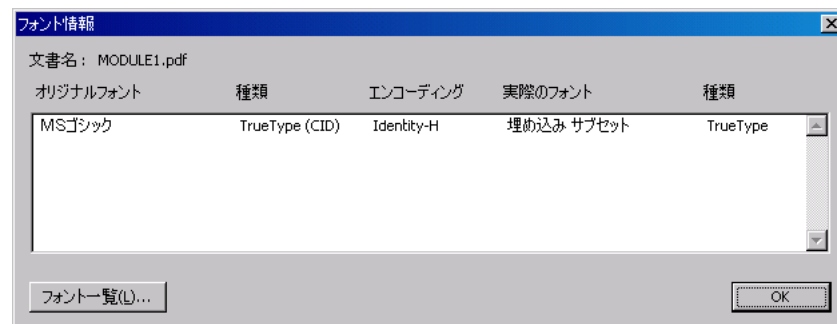
表 6-2 Windows プラットフォームにおける東アジア言語用の一般的な TrueType Collection ファイル (続き)

TTC ファイル名	フォント名	テーブル・ディレクトリ番号
gulim.ttc	Gulim	0
	GulimChe	1
	Dotum	2
	DotumChe	3
mingliu.ttc	MingLiU	0
	PMingLiU	1
msgothic.ttc	MS Gothic	0
	MS PGothic	1
	MS UI Gothic	2
msmincho.ttc	MS Mincho	0
	MS PMincho	1
simsun.ttc	SimSun	0
	NSimSun	1

レポートに使用されているフォントは、次の方法で表示できます。

- Acrobat Reader 6.0 以降の場合は、「ファイル」→「文書のプロパティ」→「フォント」を選択します（それよりも前のリリースで Acrobat Reader 3.0 以降の場合は、「ファイル」→「文書情報」→「フォント」を選択します）。
- 「フォント情報」ダイアログ・ボックスに、「オリジナルフォント」、「種類」、「エンコーディング」、「実際のフォント」（使用しているフォント）および「種類」が表示されます。

図 6-2 フォント・サブセット



注意： フォント・サブセットでは、次のようになります。

- 「エンコーディング」列に「Identity-H」と表示されます。
- 「実際のフォント」列に「埋め込みサブセット」と表示されます。
- 「種類」列に「TrueType」と表示されます。

6.1.2.2.2 下位互換性 Oracle Reports 10g リリース 2 (10.1.2) よりも前のリリースで使用されているフォント・サブセット（つまり、Type3 フォント）の実装に戻すには、環境変数 `REPORTS_ENHANCED_SUBSET=NO` を設定します。

注意： 詳細は、第 B.1.43 項「REPORTS_ENHANCED_SUBSET」を参照してください。

REPORTS_ENHANCED_SUBSET=NO を設定する場合は、最適な表示を確実にします。それには、Adobe Acrobat Reader を使用して、次の手順を実行します。

1. 「編集」→「環境設定」→「ページ表示」を選択します。
2. 「文字のスムージング」、「ラインアートのスムージング」および「画像のスムージング」を選択します。
3. (ラップトップ /LCD 画面の場合)「CoolType 使用」チェックボックスを選択します。
4. 「OK」をクリックします。

注意： この手順は Adobe Acrobat Reader 7.0 に適用されます。

UNIX マシンにおけるレポートの実行の詳細は、第 6.2.2 項「異なるプラットフォームでのレポートの設計と配布」を参照してください。

6.1.2.2.3 トラブルシューティング フォント・サブセットが機能しない場合は、次のことを確認します。

- レポートに使用されているフォントに、太字、イタリックおよび太字のイタリックの各バージョンがあること。PDF フォント・サブセットで、レポートにイタリックまたは太字スタイルが使用されているときに、出力にイタリックまたは太字スタイルが表示されない場合は、Windows の TTF ファイルを確認します。Windows では、すべてのフォントに、太字、イタリックおよび太字のイタリックの各バージョンがあるわけではありません。たとえば、Arial には、arialbd.ttf (太字の Arial)、ariali.ttf (イタリックの Arial) および arialbi.ttf (太字のイタリックの Arial) がありますが、Arial Unicode MS (arialuni.ttf) などの一部のフォントには、太字またはイタリックのバージョンがありません。フォントに太字またはイタリックのバージョンがない場合、Windows では、Windows 上の Oracle Reports と同様に、レポートの表示時にメインのフォント・ファイルから太字またはイタリック・スタイルが合成されます。これらのスタイルは、HTML/HTMLCSS、RTF および PDF (PDF サブセットまたは埋込みのない) 出力で保持されます。ただし、実際にはフォント絵文字がレポートに含まれるため、PDF サブセットまたは埋込みの実行時に、Oracle Reports はスタイルを含む TTF ファイルを必要とします。つまり、レポートに Arial の太字スタイルを含めるには、arialbd.ttf が必要となります。ただし、こうした TTF ファイルを持たない Arial Unicode MS などのフォントでは、PDF のサブセット出力に太字またはイタリック・スタイルが表示されません。
- 「実際のフォント」値が「埋め込みサブセット」、「種類」が「TrueType」であること (Acrobat Reader 6.0 以降の場合は「ファイル」→「文書のプロパティ」→「フォント」を、それよりも前のリリースで Acrobat Reader 3.0 以降の場合は「ファイル」→「文書情報」→「フォント」を選択)。そのフォントが指定されていない場合は、フォント・サブセットが実装されていません。問題の原因として、フォントが検出されていないこと、または uifont.ali ファイルのエントリが正しくないことが考えられます。
- フォント・ファイル名が有効であること。
- フォント・ファイル名の大文字・小文字が、ファイルの定義と一致すること。
- フォント・タイプが TrueType、つまり、filename.ttf/filename.ttc であること。
- フォント名に複数の単語が含まれる場合は、二重引用符で囲まれていること。
- フォント名に埋込みカッコが含まれていないこと。
- フォント・ファイルが、REPORTS_PATH 環境変数で指定されているパスにあること。Oracle Reports では、PDF ファイルの生成時に、REPORTS_PATH 環境変数で指定されたパスでフォントを探します。

- フォント名が正しいこと、および PDF ファイルを生成したマシンでそのフォントが使用可能であること。
- uifont.ali ファイルの [PDF:Subset] セクション名が変更されていないこと。Oracle Reports では、このセクション名を探してファイルを解析します。
- レポートの表示に使用している Adobe Acrobat Reader がバージョン 3.0 以降であること。マルチバイト・キャラクタのレポートを正しく表示するには、最低でもこのバージョンが必要です。
- REPORTS_ENHANCED_SUBSET 環境変数の値が YES に設定されていること。REPORTS_ENHANCED_SUBSET=NO の場合は、PDF ドキュメントの作成に Type3 フォントを使用する、以前のフォント・サブセット実装に戻されます。Type3 フォントは Type1 フォントよりもやや太い、イメージ化された文字です。表示品質を向上させる方法の詳細は、[第 6.1.2.2 項「下位互換性」](#)を参照してください。
- UNIX プラットフォームでは、TrueType フォントの使用について制限があります。UNIX マシンにおけるレポートの実行の詳細は、[第 6.2.2 項「異なるプラットフォームでのレポートの設計と配布」](#)を参照してください。

6.1.2.3 フォントの埋込み

PDF のフォントの埋込みは、データとともにフォント・セット全体を PDF ファイルに含める処理です。PDF のフォント・サブセットとフォントの埋込みは、相互排他的です。

注意： フォントの埋込みは、フォントが PDF ファイルに含まれている場合のみ機能します。フォントの埋込みを行うと、PDF ファイルのサイズが増えます。

Oracle Reports での PDF フォントの埋込みは、Type1 フォント（シングルバイト・フォント）のみが対象であり、TrueType フォントは対象外です。特定の Type1 フォントをレポートに含めるには、市販のサードパーティ製ツールを使用して、TrueType フォントを Type1 フォントに変換します。

Oracle Reports での PDF フォントの埋込みは、フォントとフォント・ファイル名のセットとの間で行われます。

注意： 出力にフォントを埋め込む前に、必要なフォント・ライセンスを保持していることを確認する必要があります。

6.1.2.3.1 設定

PDF の埋込みの設定には、次のものが含まれます。

- コマンドライン・キーワード: PDFEMBED
- uifont.ali ファイル・エントリ: [PDF:Embed]

PDFEMBED

コマンドライン・キーワード PDFEMBED は、Oracle Reports が、uifont.ali ファイルで指定された Type 1 PostScript フォントを PDF 出力に埋め込むかどうかを指定します。詳細は、[第 A.3.84 項「PDFEMBED」](#)を参照してください。

uifont.ali ファイル・エントリ

uifont.ali ファイル内の、フォントの埋込みのセクションは、[PDF:Embed] です。

(Windows のみ) uifont.ali ファイルのエントリは次のとおりです。

```
font_name = "font_name.pfm font_name.pfb"
```

(UNIX のみ) uifont.ali ファイルのエントリは次のとおりです。

```
font_name = "font_name.afm font_name.pfa"
```

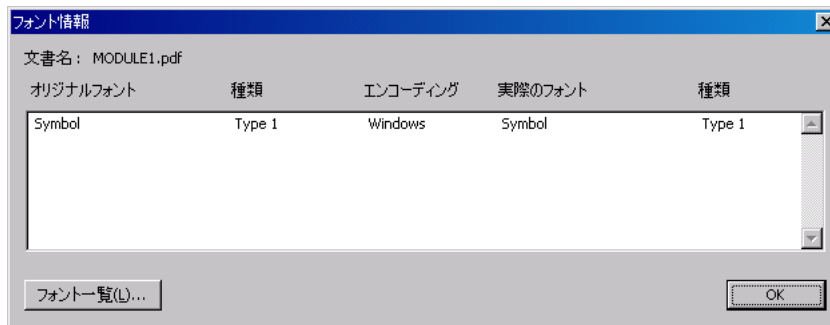
例 6-1 フォントの埋込み

```
[PDF:Embed]
Symbol = "Symbol.pfm Symbol.pfb"
```

例 6-1 では、Symbol フォントが PDF ファイルに埋め込まれています。移植性を確保する手順は次のとおりです。

1. Symbol フォントを含むレポートを作成します。
2. Symbol フォントを PDF ファイルに埋め込みます (図 6-3)。

図 6-3 フォントの埋込み



6.1.2.3.2 トラブルシューティング

PDF フォントの埋込みが機能しない場合は、次のことを確認します。

- Acrobat Reader 6.0 以降の場合は、「ファイル」→「文書のプロパティ」→「フォント」を選択します（それよりも前のリリースで Acrobat Reader 3.0 以降の場合は、「ファイル」→「文書情報」→「フォント」を選択します）。埋め込んだフォントがリストに追加されていることを確認します。フォントが追加されていない場合、フォントの埋込みは行われていません。問題の原因として、フォントが検出されていないこと、または uifont.ali ファイルのエントリが正しくないことが考えられます。
- 正しいフォント・ファイル名が使用されていること。
- REPORTS_PATH 環境変数で指定されているフォント・パスが正しいこと。Oracle Reports では、PDF ファイルの生成時に、REPORTS_PATH 環境変数で指定されたパスでフォントを探します。
- フォント・タイプが Type1 フォントであること。
- フォント名に複数の単語が含まれる場合は、二重引用符で囲まれていること。
- uifont.ali ファイルの [PDF:Embed] セクション名が変更されていないこと。Oracle Reports では、このセクション名を探してファイルを解析します。
- 埋込みフォントを指定する書式が有効であること。

```
font_name="fontfilename.pfm/.afm file fontfilename.pfb/.pfa file"
```

例 (Windows の場合) :

```
UtopiaMediumItalic = "UtopiaMediumItalic.pfm UtopiaMediumItalic.pfb"
```

- フォント名が正しいこと、および PDF ファイルを生成したマシンでそのフォントが使用可能であること。

6.1.2.4 フォント機能のまとめ

表 6-3 に、フォント・エイリアシング、フォントの埋込みおよびフォント・サブセットの長所と短所をまとめます。

表 6-3 各種 PDF フォント機能の比較

PDF タイプ	長所	短所	PDF タイプ
フォント・エイリアシング	マルチバイトをサポートしている。 表示品質がよい。 ファイル・サイズが小さい (日本語では、フォント・サブセットの 130KB に対して、フォント・エイリアシングでは 23KB で済む)。	Unicode キャラクタ・セットをサポートしていない。 クライアントのオペレーティング・システムと Acrobat Reader が現地語バージョンでない場合は、クライアント・マシンに Asian Font Packs が必要。 フォントのサポートに制限がある。たとえば、フォントの強調をサポートしていない。	フォント・エイリアシング
フォントの埋込み	表示が保証されている。	シングルバイトしかサポートしていない。 ファイルのサイズが大きい。	フォントの埋込み
フォント・サブセット	Unicode をサポートしている。 表示が保証されている。 生成されたファイルを、Adobe Acrobat を使用して検索および編集できる。	スタイル (斜体と太字) をサポートしていない。	フォント・サブセット

6.1.3 実行の優先順位

uifont.ali ファイル内の複数の箇所と同じフォントが出現する場合、優先順位は次のとおりです。

1. フォント・エイリアシングがフォントの埋込みより優先される (最高の優先順位)
2. フォント・サブセットがフォントの埋込みより優先される (中間の優先順位)
3. フォントの埋込みは優先されない (最低の優先順位)

たとえば、フォントの埋込みとフォント・サブセットの両方に対して同じフォント・エントリを含めた場合は、フォント・サブセットが優先されます。これには、コマンドライン・オプションの PDFEMBDED=NO を設定していないことが前提となります。

どのフォント機能でも (フォント・エイリアシング、フォント・サブセット、フォントの埋込み)、特殊なエントリを最初に指定してから、汎用エントリを指定します。たとえば、Arial Plain、Arial Bold、Arial Italic、Arial Bold-Italic の各フォントのサブセットを作成する場合、エントリを次の順序にします。

```
[ PDF:Subset ]
Arial..Italic.Bold.. = "Arialbi.ttf"
Arial...Bold.. = "Arialb.ttf"
Arial..Italic... = "Ariali.ttf"
Arial..... = "Arial.ttf"
```

プレーンな Arial..... = "Arial.ttf" エントリが最初にあると、このレイアウトにある Arial フォントのすべてのスタイルが Arial Plain フォントのサブセットとして作成されます。次に、uifont.ali ファイルの一部を抜粋したサンプルを示します。どの PDF エントリにも、3つの PDF セクションがすべて含まれています。

サンプル 1

```
[ PDF ]
Palatino = "Kino MT.ttf"
[ PDF:Subset ]
Garmond..Italic.Bold.. =
"Garmacbi.ttf"
Garmond...Bold... = "Garmacb.ttf"
Garmond..Italic... = "Garmacbi.ttf"
Garmond..... = "Garamac.ttf"
[ PDF:Embed ]
Arial = "Arial.pfm Arial.pfb"
```

サンプル 2

```
[PDF]
Arial.10.Italic = "Times New Roman".12.Italic.Bold
"Courier New" = Symbol
[PDF:Embed]
"Times New Roman".14..Bold = "TimesBold.pfm TimesBold.pfb"
[PDF:Subset]
Verdana..Italic.Bold = "Verdanaz.ttf"
Verdana..Bold = "Verdanab.ttf"
```

6.1.4 アクセシビリティ

Oracle Reports では、様々な方法で PDF ファイルにアクセシビリティ機能を含めることができます。PDF フォーマットのファイルは、PDF 1.4 で定義されているタグ付き PDF の標準に従います。この標準と Acrobat Reader 5（またはそれ以降のバージョン）を使用することにより、ペーパー・レイアウトにアクセシビリティ機能を含めることができます。

Oracle Reports が提供するアクセシビリティ関連の機能をコマンドラインから有効化する方法は、第 A.3.1 項「ACCESSIBLE」を参照してください。障害のあるユーザーが PDF レポート出力にアクセスできるようにする目的で設計された Oracle Reports のアクセシビリティ・プロパティ（「代替テキスト」、「ヘッダ」、「ID」、「レポート言語」および「表キャプション」プロパティ）の使用方法は、Oracle Reports のオンライン・ヘルプを参照してください。

さらに、『Oracle Reports レポート作成ガイド』の第 43 章「アクセス可能な JSP ベースの Web レポートの作成」および OTN の Oracle アクセシビリティ・サイト (<http://www.oracle.com/accessibility/index.html>) を参照してください。このサイトでは、アクセシビリティに関するさらなる説明と、ホワイト・ペーパー『Creating Accessible Enterprise Reports Using Oracle Reports』を参照できます。

6.1.5 分類

PDF 文書には、その文書に関する全体的な情報（文書のタイトル、作成者、作成日時、変更日時など）を含めることができます。このような全体的な情報は、外部データベースで文書をカタログ化したり検索するときには、特に便利です。

Oracle Reports では、そのような分類を行うための次のプロパティが用意されています。

- タイトル
- 作成者
- 件名
- キーワード

表 6-4 分類プロパティ

プロパティ名	型	説明	デフォルト値
タイトル	STRING	文書のタイトル	PDF 文書名
作成者	STRING	文書の作成者	Oracle Reports

表 6-4 分類プロパティ (続き)

プロパティ名	型	説明	デフォルト値
件名	STRING	文書の件名	なし
キーワード	STRING	文書を分類するためのキーワードを指定	なし

分類プロパティの詳細は、Oracle Reports のオンライン・ヘルプを参照してください。

6.1.6 グラフのサポート

Oracle Reports には、PDF 出力内のグラフのイメージ解像度に適用するインチ当たりのドット数 (DPI) 値を指定する機能が備わっています。この機能によって、イメージの品質を損なうことなく、グラフを拡大縮小できます。

詳細は、第 B.1.44 項「REPORTS_GRAPH_IMAGE_DPI」および第 B.1.46 項「REPORTS_JPEG_QUALITY_FACTOR」を参照してください。

6.2 プラットフォーム間の配布時における PDF フォントの問題への対応

シングルバイトまたはマルチバイトのレポートを Windows プラットフォーム上で設計し、これを UNIX プラットフォームに配布すると、フォントや文字位置に関して問題が発生します。原因は、2つのプラットフォームの間でフォント処理およびウィンドウ表示システムがまったく異なることにあります。

6.2.1 同じプラットフォームでのレポートの設計と配布

レポートの設計と配布を同一プラットフォーム上 (たとえば、Windows) で行う場合、次のことが保証されます。

- PDF ファイルでのフォントや文字位置の問題は発生しません。
- フォント・サブセットが有効な状態で生成された PDF ファイルは、異なるプラットフォーム間で同じように表示できます。

6.2.2 異なるプラットフォームでのレポートの設計と配布

レポートを Windows プラットフォームで設計し、UNIX プラットフォームに配布する場合、次の操作を行います。

(Windows) Windows マシンにある TrueType フォントを使用します (通常の場合は、`%windir%/fonts`)。Oracle Reports では、レポートのフォーマット用のフォント情報について、Windows システムへの問合せが行われます。

(UNIX) このレポートが PDF 形式で UNIX プラットフォームに送信されると、次の 2 つの処理が行われます。

1. Oracle Reports ではフォントのフォント・メトリック情報が提供され、この情報を使用して、レポート内の各種オブジェクトをフォーマットします。

注意： Oracle Reports では、プリンタの PPD ファイルで指定されている AFM ファイルからフォント・メトリック情報が提供されます。

2. その後、Oracle Reports では、`uifont.ali` ファイルの [PDF] セクションでエントリが検索されます。フォント・サブセットについては、[PDF:Subset] セクションを参照し、所定の場所にある TrueType フォントのフォント・サブセットを作成します。サブセット・フォントは、PDF ファイルに埋め込まれます。

注意： UNIX マシンでフォーマットが確実に行われるには、レポートで使用されているすべての TrueType フォントに対応する AFM ファイルが、UNIX マシンで使用可能である必要があります。

6.2.2.1 シングルバイト・フォントを使用する PDF レポートの生成

この項では、Windows または UNIX プラットフォーム上で設計したシングルバイト・フォントを使用する PDF レポートを生成する手順の概要について説明します。これらの手順は、PDF 出力にフォント位置の問題が発生している場合にのみ必要です。この項で説明するフォント機能を使用する前に、表 6-3 を参照して、使用するアプリケーションのニーズに最適な機能を確認してください。

この例では、PDF フォント・サブセットを使用します。

1. Windows プラットフォームで、TrueType フォントのレポートを作成します。この手順例で参照されるフォントは、arial.ttf と tahoma.ttf です。
2. フォント (arial.ttf と tahoma.ttf) およびレポートの .rdf ファイルを UNIX プラットフォームにコピーします。フォント・ファイルのパスは、ORACLE_HOME/reports/font_folder にする必要があります。フォント・ファイルのパスを、REPORTS_PATH 環境変数に追加します。
3. ttft2pt1 などのフリー・ユーティリティを使用して、フォント・ファイル (arial.ttf および tahoma.ttf) の AFM ファイルを作成します。TFM ファイルには変換しないでください。予測したとおりの結果にならない場合があります。
4. 作成した AFM ファイル (arial.afm および tahoma.afm) を ORACLE_HOME/guicommon/tk/admin/AFM にコピーします。

注意： AFM ファイルは、.afm 拡張子を付けずに AFM ディレクトリにコピーする必要があります。さらに、AFM ファイルの名前、PPD ファイル内のフォントの名前、および uifont.ali ファイル内のフォントの名前を必ず同じものにします。

5. 任意のテキスト・エディタを使用して、screenprinter.ppd ファイルを編集します。

注意： ORACLE_HOME/guicommon/tk/admin/uiprint.txt にエントリを追加してデフォルトのプリンタを定義している場合は、プリンタの PPD ファイル (PostScript プリンタ) または HPD ファイル (PCL プリンタ) に適切なエントリが必要です。

Oracle Reports 10g (9.0.4) から、デフォルトのプリンタを設定していない場合は、画面を擬似実行するデフォルトのプリンタ・サーフェス (screenprinter.ppd) がフォーマットで使用されるようになりました。デフォルトのプリンタを設定していない場合は、screenprinter.ppd ファイルに必要なフォントや解像度のエントリも追加する必要があります。

PPD ファイルは次の場所にあります。

ORACLE_HOME/guicommon/tk/admin/PPD

HPD ファイルは次の場所にあります。

ORACLE_HOME/guicommon/tk/admin/HPD

screenprinter.ppd ファイルの詳細は、第 3.10.1 項「ScreenPrinter」を参照してください。

使用する PPD/HPD ファイルに、レポートに使用されている AFM ファイルまたは TFM ファイルすべてのエントリがあることを確認します。PPD/HPD ファイルはプリンタ・ドライバの設定を含む構成ファイルであり、そのプリンタでサポートされるすべてのフォントがリストされています。

PPD ファイルの Font Information セクションに移動して、必要なフォント・ファイルのエントリを次のような形式で追加します。

```
*FONTNAME:ENCODING:VERSION:LOCATION
```

次に例を示します。

```
*Font Arial: Standard "(Version 2.76)" Standard ROM
*Font CourierNew: Standard "(Version 2.76)" Standard ROM
```

AFM ファイル名が、PPD ファイルに指定されているフォント名と一致していることを確認します。Oracle Reports は、PPD ファイルのフォント名に基づいて AFM ファイルを検索します。

6. uiprint.txt ファイルに次のエントリを含めます。

```
printer name:PostScript:2:test:default.ppd:
```

次に例を示します。

```
hrprinter:PostScript:2:test:default.ppd:
```

7. PPD ファイルに AFM エントリを追加します。

注意： この PPD ファイルは uiprint.txt ファイルの最初のエントリで、フォント情報を格納しているものです。デフォルトの PPD ファイルは datap462.ppd です。

```
*Font arial: Standard "(001.001)" Standard ROM
*Font tahoma: Standard "(001.001)" Standard ROM
```

8. この時点で、uifont.ali ファイルの [PDF:Subset] セクションにエントリがないことを確認します。
9. レポートを実行して PDF ファイルを生成します。Acrobat Reader 6.0 以降の場合は、「ファイル」→「文書のプロパティ」→「フォント」を選択します（それよりも前のリリースで Acrobat Reader 3.0 以降の場合は、「ファイル」→「文書情報」→「フォント」を選択します）。
 - a. 「オリジナルフォント」列に Arial フォントと Tahoma フォントが表示されます。
 - b. フォントの位置が多少ずれてしまいます。
10. uifont.ali ファイルに次のエントリを追加します。

```
[ PDF:Subset ]
"arial" = "arial.ttf"
"tahoma" = "tahoma.ttf"
```

11. レポートを再度実行して PDF ファイルを生成します。これで、フォントの位置のずれがすべて修正されたはずですが、フォント・サブセットが PDF ファイル内に作成されたことを次のように確認します。
 - a. Acrobat Reader 6.0 以降の場合は、「ファイル」→「文書のプロパティ」→「フォント」を選択します（それよりも前のリリースで Acrobat Reader 3.0 以降の場合は、「ファイル」→「文書情報」→「フォント」を選択します）。
 - b. 「オリジナルフォント」列にフォント名、「エンコーディング」列に「Identity-H」、「種類」列に「TrueType」がそれぞれ表示されているはずですが。

6.2.2.2 マルチバイト・フォントおよび Unicode フォントを使用する PDF レポートの生成

マルチバイト・フォントおよび Unicode フォントのレポートを生成する場合の追加手順について説明します。この項で説明するフォント機能を使用する前に、表 6-3 を参照して、使用するアプリケーションのニーズに最適な機能を確認してください。

マルチバイトおよび Unicode のレポートを UNIX プラットフォームに配布する場合の、PDF フォント・サブセット作成によるフォントの問題を解決する手順は、次のとおりです。

1. TrueType マルチバイト・フォントと適切なキャラクタ・セットを使用して、Windows プラットフォームでレポートを作成します。この手順例で参照されるフォントとキャラクタ・セットは、韓国語フォント h2mjsm.ttf と KO16KSC5601 キャラクタ・セットです。
2. 韓国語フォント (h2mjsm.ttf) およびレポートの .rdf ファイルを UNIX プラットフォームにコピーします。フォント・ファイルのパスは、\$ORACLE_HOME/reports/font_folder にする必要があります。フォント・ファイルのパスを、REPORTS_PATH 環境変数に追加します。
3. 韓国語フォント h2mjsm.ttf の AFM ファイルを作成します。
4. 作成した AFM ファイルを次のディレクトリにコピーします。
\$ORACLE_HOME/guicommon/tk/admin/AFM/
% cp h2mjsm.afm \$ORACLE_HOME/guicommon/tk/admin/AFM/h2mjsm
5. 任意のテキスト・エディタを使用して、screenprinter.ppd ファイルを編集します。

注意： \$ORACLE_HOME/guicommon/tk/admin/uiprint.txt にエントリを追加してデフォルトのプリンタを定義している場合は、プリンタの PPD ファイル (PostScript プリンタ) または HPD ファイル (PCL プリンタ) に適切なエントリが必要です。

Oracle Reports 10g (9.0.4) から、デフォルトのプリンタを設定していない場合は、画面を擬似実行するデフォルトのプリンタ・サーフェス (screenprinter.ppd) がフォーマットで使用されるようになりました。デフォルトのプリンタを設定していない場合は、screenprinter.ppd ファイルに必要なフォントや解像度のエントリも追加する必要があります。

PPD ファイルは次の場所にあります。

\$ORACLE_HOME/guicommon/tk/admin/PPD

HPD ファイルは次の場所にあります。

\$ORACLE_HOME/guicommon/tk/admin/HPD

screenprinter.ppd ファイルの詳細は、第 3.10.1 項「ScreenPrinter」を参照してください。

使用する PPD/HPD ファイルに、レポートに使用されている AFM ファイルまたは TFM ファイルすべてのエントリがあることを確認します。PPD/HPD ファイルはプリンタ・ドライバの設定を含む構成ファイルであり、そのプリンタでサポートされるすべてのフォントがリストされています。

PPD ファイルの Font Information セクションに移動して、必要なフォント・ファイルのエントリを次のような形式で追加します。

*FONTNAME:ENCODING:VERSION:LOCATION

次に例を示します。

*Font Arial: Standard "(Version 2.76)" Standard ROM

*Font CourierNew: Standard "(Version 2.76)" Standard ROM

AFM ファイル名が、PPD ファイルに指定されているフォント名と一致していることを確認します。Oracle Reports は、PPD ファイルのフォント名に基づいて AFM ファイルを検索します。

6. uiprint.txt ファイルに次のエントリを含めます。

```
printer name:PostScript:2:test:default.ppd:
```

次に例を示します。

```
hrprinter:PostScript:2:test:default.ppd:
```

7. PPD ファイルに次の行を追加します。

注意： この PPD ファイルは uiprint.txt ファイルの最初のエントリで、フォント情報を格納しているものです。デフォルトの PPD ファイルは datap462.ppd です。

```
*DefaultFont: h2mjasm
*Font h2mjasm: Special "(001.001)" Special ROM
```

8. ファイル内の Symbol 行をコメント化します。

```
;%Font Symbol: Special "(001.001)" Special ROM
```

9. uifont.ali ファイルの次のセクションを編集して、キャラクタ・セットに使用するフォントを指定します。

注意： uifont.ali ファイルは、次のディレクトリに格納されています。

- ORACLE_HOME¥tools¥common (Windows の場合)
- ORACLE_HOME/guicommon/tk/admin (UNIX の場合)

uifont.ali ファイルは、Oracle Reports の PDF フォントに関する拡張機能すべてを制御する構成ファイルです。詳細は、[第 4 章「Oracle Reports でのフォントの管理」](#)を参照してください。

```
[ Global ]
....ko16ksc5601 ="h2mjasm"
[ Printer:PostScript2 ]
....ko16ksc5601 ="h2mjasm"
```

10. この時点で、uifont.ali ファイルの [PDF] セクションまたは [PDF:Subset] セクションにエントリがないことを確認します。
11. レポートを実行して PDF ファイルを生成します。Acrobat Reader 6.0 以降の場合は、「ファイル」→「文書のプロパティ」→「フォント」を選択します（それよりも前のリリースで Acrobat Reader 3.0 以降の場合は、「ファイル」→「文書情報」→「フォント」を選択します）。
- a. 「オリジナルフォント」列に h2mjasm フォントが表示されます。
 - b. フォントの位置が多少ずれてしまいます。
12. uifont.ali ファイルに次のエントリを追加して、PDF のサブセットを有効にします。

```
[ PDF:Subset ]
"h2mjasm"="h2mjasm.ttf"
```

13. レポートを再度実行して PDF ファイルを生成します。これで、フォントの位置のずれがすべて修正されたはずですが。

フォント・サブセットが PDF ファイル内に作成されたことを次のように確認します。

- a. Acrobat Reader 6.0 以降の場合は、「ファイル」→「文書のプロパティ」→「フォント」を選択します（それよりも前のリリースで Acrobat Reader 3.0 以降の場合は、「ファイル」→「文書情報」→「フォント」を選択します）。
- b. 「オリジナルフォント」列にフォント名、「エンコーディング」列に「Identity-H」、「種類」列に「TrueType」がそれぞれ表示されているはずですが。

注意： UNIX と Windows ではフォント・メトリックの処理方法が異なるため、位置がずれてしまう可能性があります。これにかわる方法は、フォント・ベンダーから適切な AFM ファイルを購入することです。サード・パーティ・ユーティリティによって生成された AFM ファイルは、設計プラットフォームで使用されたフォントと完全に一致するフォント・メトリックを持たない場合があります。

フォント・サブセットを有効にして生成した PDF ファイルでは、一般的にフォント・スタイルの問題は発生しません。ただし、REPORTS_ENHANCED_SUBSET=NO が設定されている場合は、Acrobat Reader で表示する際になんらかのフォント・スタイルの問題が発生する可能性があります（たとえば、一部の内容が太字で表示されるなど）。Type3 フォントの表示をスムージングする方法の詳細は、第 6.1.2.2.2 項「下位互換性」を参照してください。

6.3 Unicode PDF ファイルの生成

この項では、Unicode キャラクタ・セットを使用して PDF ファイルを生成する手順について説明します。この項で説明するフォント機能を使用する前に、表 6-3 を参照して、使用するアプリケーションのニーズに最適な機能を確認してください。

6.3.1 フォント・サブセット

フォント・サブセット機能を使用して Unicode PDF ファイルを生成する手順は、次のとおりです。

1. NLS_LANG=AMERICAN_AMERICA.UTF8 のように UTF8 キャラクタ・セットを設定します。
2. REPORTS_PATH を、TrueType フォントが格納されているフォント・ディレクトリに設定します。たとえば、C:\WINNT\fonts などです。
3. uifont.ali ファイルを開いてから [PDF:Subset] セクションを編集し、TrueType フォントの名前を指定します。

注意： uifont.ali ファイルは、次のディレクトリに格納されています。

- ORACLE_HOME\tools\common (Windows の場合)
 - ORACLE_HOME/gui\common/tk/admin (UNIX の場合)
-

例

```
[ PDF:Subset ]
"Andale Duospace WT J" = "Aduoj.ttf"
"Albany WT J"="AlbanWIJ.ttf"
```

指定されたフォントは、レポートで使用する Unicode 文字の範囲をカバーする必要があります。

4. 多言語データを含むレポートを作成し、そのフォントを Unicode フォントに設定します。
5. 多言語データを含むレポートを DESTYPE=FILE DESFORMAT=PDF で実行します。

6.4 双方向 (BiDi) PDF ファイルの生成

この項では、双方向 (BiDi) 言語の PDF ファイルを生成する手順について説明します。この項で説明するフォント機能を使用する前に、表 6-3 を参照して、使用するアプリケーションのニーズに最適な機能を確認してください。

Oracle Reports では、ヘブライ語やアラビア語などの双方向 (BiDi) 言語でのフォントの整形オプションや数値オプションを決定する次の 2 つの環境変数を備えています。

1. REPORTS_BIDI_ALGORITHM

この環境変数は、双方向 (BiDi) 言語 (アラビア語やヘブライ語など) のレイアウト・アルゴリズムを切り替えます。この環境変数の有効値は、ORACLE または UNICODE です。

関連項目: 第 B.1.29 項「REPORTS_BIDI_ALGORITHM」

2. REPORTS_ARABIC_NUMERAL

この環境変数は、アラビア語の PDF 出力での数値の書式を指定します。

関連項目: 第 B.1.28 項「REPORTS_ARABIC_NUMERAL」

6.4.1 フォント・サブセット

この項の例では、アラビア語環境を使用していることを前提とします。フォント・サブセット機能を使用して双方向 (BiDi) 言語の PDF ファイルを生成する手順は、次のとおりです。

1. NLS_LANG=ARABIC_EGYPT.AR8MSWIN1256 (UNIX では AR8ISO8859P6) のように BiDi キャラクタ・セットを設定します。
2. REPORTS_PATH を、TrueType フォントが格納されているフォント・ディレクトリに設定します。たとえば、C:¥WINNT¥fonts などです。
3. uifont.ali ファイルを開いてから [PDF:Subset] セクションを編集し、TrueType フォントの名前を指定します。

注意: uifont.ali ファイルは、次のディレクトリに格納されています。

- ORACLE_HOME¥tools¥common (Windows の場合)
 - ORACLE_HOME/guicommon/tk/admin (UNIX の場合)
-
-

例

```
[PDF:Subset]
"Andale Duospace WT J" = "Aduoj.ttf"
"Albany WT J"="AlbanWTJ.ttf"
```

4. アラビア語のデータを含むレポートを作成し、それを例で指定したフォントに設定します。
5. DESTYPE=FILE DESFORMAT=PDF を指定してレポートを実行します。

6.5 マルチバイト PDF ファイルの生成

この項では、マルチバイト・フォントを含む PDF ファイルを生成する手順について説明します。この項で説明するフォント機能を使用する前に、表 6-3 を参照して、使用するアプリケーションのニーズに最適な機能を確認してください。

PDF のフォント・サブセット出力では、全幅チルド (U+FF5E) のかわりに波ダッシュ (U+301C) が表示される場合があります。これは、Microsoft 社と他のベンダー間で文字マッピングに互換性がないことによります。この問題を回避するには、PDF のフォント・サブセットに JA16SJISTILDE または JA16EUCTILDE キャラクタ・セットを使用します。ただし、この問題は PDF フォント・エイリアシング機能には見られません。

6.5.1 フォント・エイリアシング

マルチバイト・フォントの PDF フォント・エイリアシングに使用する Oracle NLS_CHARACTERSET、CMap 名および CID フォント名間のマッピングは、表 6-1 を参照してください。

フォント・エイリアシングを使用してマルチバイト・フォントの PDF ファイルを生成する手順は、次のとおりです。

1. NLS_LANG=JAPANESE_JAPAN.JA16SJIS (UNIX では JA16EUC) を設定します。
2. uifont.ali ファイルを開き、[PDF] セクションでフォント・エイリアスを設定します。

注意： uifont.ali ファイルは、次のディレクトリに格納されています。

- ORACLE_HOME¥tools¥common (Windows の場合)
 - ORACLE_HOME/guicommon/tk/admin (UNIX の場合)
-

例

```
[ PDF ]
.....JA16SJIS = "KozMinPro-Regular-Acro"
"MS UI Gothic".....JA16SJIS = "KozMinPro-Regular-Acro"
```

3. 日本語フォント (MS UI Gothic) を指定した日本語データのレポートを作成します。
4. DESTYPE=FILE DESFORMAT=PDF を指定してレポートを実行します。
5. 使用している Acrobat Reader が、日本語以外のオペレーティング・システムにインストールされた他言語バージョンである場合、日本語のフォント・パックを Adobe のサイトから入手してインストールする必要があります。

Windows (日本語版) で日本語バージョンの Acrobat Reader 4.0 または 5.0 を使用して PDF ファイルを表示する場合は、日本語のフォント・パックは不要です。

6.5.2 フォント・サブセット

フォント・サブセット機能を使用してマルチバイト・フォントの PDF ファイルを生成する手順は、次のとおりです。

1. NLS_LANG=JAPANESE_JAPAN.JA16SJIS (UNIX では JA16EUC) を設定します。
2. REPORTS_PATH 環境を、TrueType フォントが格納されているフォント・ディレクトリに設定します。たとえば、C:¥WINNT¥Fonts。
3. uifont.ali ファイルを開いてから [PDF:Subset] セクションを編集し、TrueType フォントの名前を指定します。

注意: uifont.ali ファイルは、次のディレクトリに格納されています。

- ORACLE_HOME¥tools¥common (Windows の場合)
 - ORACLE_HOME/guicommon/tk/admin (UNIX の場合)
-
-

例

```
[ PDF:Subset ]
"Andale Duospace WT J" = "Aduoj.ttf"
"Albany WT J"="AlbanWTJ.ttf"
"MS UI Gothic" = "msgothic.ttc"
```

4. 日本語のデータを含むレポートを作成し、それを例で指定したフォントに設定します。
5. DESTYPE=FILE DESFORMAT=PDF を指定してレポートを実行します。

6.6 バーコード PDF ファイルの生成

この項では、バーコード情報を含む PDF ファイルを生成する手順について説明します。この項で説明するフォント機能を使用する前に、表 6-3 を参照して、使用するアプリケーションのニーズに最適な機能を確認してください。

6.6.1 フォントの埋込み

フォントの埋込みの機能を使用してバーコード PDF ファイルを生成する手順は、次のとおりです。

1. REPORTS_PATH 環境変数を、Type1 フォントが格納されているフォント・ディレクトリに設定します。
2. uifont.ali ファイルを開き、フォントの埋込み [PDF:Embed] セクションで、次のように指定します。

注意: uifont.ali ファイルは、次のディレクトリに格納されています。

- ORACLE_HOME¥tools¥common (Windows の場合)
 - ORACLE_HOME/guicommon/tk/admin (UNIX の場合)
-
-

例

```
[ PDF:Embed ]
SAdHC39a = "SAdHC39a.pfm SAdHC39a.pfb"
```

3. バーコードのデータを含むレポートを作成し、そのフォントを例で指定したフォントに設定します。
4. DESTYPE=FILE DESFORMAT=PDF を指定してレポートを実行します。

6.6.2 フォント・サブセット

フォント・サブセット機能を使用してバーコード PDF ファイルを生成する手順は、次のとおりです。

1. REPORTS_PATH 環境変数を、TrueType フォントが格納されているディレクトリに設定します。たとえば、C:¥WINNT¥Fonts。
2. uifont.ali ファイルを開いてから [PDF:Subset] セクションを編集し、TrueType フォントの名前を指定します。

注意: uifont.ali ファイルは、次のディレクトリに格納されています。

- `ORACLE_HOME\tools\common` (Windows の場合)
 - `ORACLE_HOME/gui\common/tk/admin` (UNIX の場合)
-
-

例

```
[ PDF:Subset ]  
SAdHC39a = "SAdHC39a.ttf"
```

3. バーコードのデータを含むレポートを作成し、それを例で指定したフォントに設定します。
4. `DESTYPE=FILE DESFORMAT=PDF` を指定してレポートを実行します。

クロス・プラットフォームの移植問題の解決

今日のビジネスでは、プラットフォームやインフラストラクチャ間のシームレスな統合と相互運用が求められています。Oracle Reports では、組織の内部と外部を問わずあらゆるレベルに対してデータを作成し、配布することができます。しかし、企業のレポート・ツールは、システム・フォントやプリンタ・フォントなどのプラットフォーム固有の機能に依存しています。そのため、開発プラットフォーム（Windows など）と配布プラットフォーム（UNIX ベースのプラットフォームなど）が異なる場合に、それらのプラットフォーム間でレポートを移植すると、レポートのルック・アンド・フィールが変わるおそれがあります。

この章では、プラットフォームの違いによって、レポート出力のルック・アンド・フィールに影響が生じるシナリオを取り上げます。クロス・プラットフォームで問題となるレポート出力フォーマット（PDF、HTMLCSS および RTF など）を、それぞれの項に分けて説明します。各項目では、プラットフォームが変わってもレポートを同じ外観で出力できる方法を手順を追って説明します。これらの説明の後には、トラブルシューティング情報と FAQ（よくある質問）を記載します。マルチバイトや Unicode レポートには別の手順が必要になるため、これらのトピックも新たな項を設けて説明します。この章では、特に指定がないかぎり、Oracle9i Reports、Oracle Reports 10g (9.0.4)、および 10g リリース 2 (10.1.2) を対象に説明します。

この章を読むには、あらかじめ次の各章で概念と用語を理解しておいてください。

- [第 3 章「OracleAS Reports Services の構成」](#)
- [第 5 章「UNIX での Oracle Reports による印刷」](#)
- [付録 B「環境変数」](#)

この章には、次の項があります。

- [クロス・プラットフォームの問題の概要](#)
 - [プラットフォームによって異なるフォント](#)
 - [フォントに関連する問題の解決](#)
- [HTMLCSS、RTF または Web 出力の生成](#)
- [シングルバイト PDF 出力の生成](#)
- [マルチバイト PDF 出力の生成](#)
- [Unicode PDF 出力の生成](#)
- [PostScript 出力の生成](#)

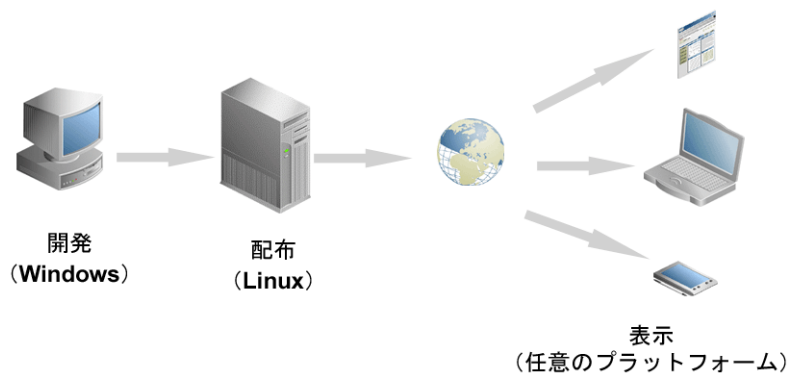
注意： この章では、レポートをクロス・プラットフォームで同じ外観に出力するために別の作業を必要とするシナリオとガイドラインについてのみ説明しています。

7.1 クロス・プラットフォームの問題の概要

Oracle Reports は、Windows、Linux、Sun Solaris、HP-UX、IBM AIX などの様々なプラットフォームで利用できます。Oracle Reports を使用すると、これらのプラットフォーム間でレポートを作成し、配布することができます。最も一般的なシナリオは、レポートを Windows で作成し、Linux などの UNIX ベース環境に配布するシナリオです（図 7-1 を参照）。その場合、配布されたレポートのロック・アンド・フィールドは若干異なって表示されます。たとえば、Windows でレポートを作成する場合、レポート内のテキスト・オブジェクトやフィールドには適切なスペースを割り当てます。しかし、そのレポートを Linux に配布して実行すると、割り当てたはずのスペースにテキストが収まらない場合があります。これが、プラットフォームの違いが直接出力に影響するクロス・プラットフォームの問題です。このような問題が起きるのは、開発プラットフォームで使用してきたフォントが配布プラットフォームで使用できないことが原因になっている場合があります。そのため、レポートを配布プラットフォームで実行する場合は、レポート出力のフォーマットに代替フォントを使用する必要があります。しかし、フォント間でなんらかの違いがあれば、開発プラットフォームと配布プラットフォームのレポート出力も異なって表示されます。

クロス・プラットフォームの問題が起きるもう 1 つのシナリオは、レポートを最終的に表示するプラットフォームに（図 7-1 を参照）適切なフォントがインストールされていない場合です。その場合は、開発プラットフォームと配布プラットフォームでレポートが正しく出力されていても、エンドユーザーがレポートを表示するプラットフォームには、レポートのロック・アンド・フィールドが異なって表示されます。

図 7-1 クロス・プラットフォームの配布シナリオの例



7.1.1 プラットフォームによって異なるフォント

フォントは、特定のスタイルとサイズで印刷または表示できる一連のテキスト文字です。フォントは、レポートを画面上に表示したり、印刷したりするために必要です。Oracle Reports は、レポートをフォーマットするとき、つまりレポート・コマンドを実行するとき、これらのフォント・メトリックを使用します。レポートがフォーマットされ、出力が生成される作業は、フォント・メトリックに基づいて実行されます。

フォント・メトリックはファイルによって提供されます。このファイルは、OracleAS Reports Services を実行するシステムで使用できる必要があります。Windows の場合、フォント・メトリックは TrueType Font (TTF) ファイルまたは TrueType Collection (TTC) ファイルによって提供されます。UNIX プラットフォームの場合、フォント・メトリックは Adobe Font Metrics (AFM) ファイルまたは TeX Font Metrics (TFM) ファイルによって提供されます。使用できるフォントやメトリックは、オペレーティング・システムによって異なります。フォントやレンダリングが異なれば、生成される出力の外観も異なります。

例 1: シングルバイト領域において一般的に使用される Tahoma フォントは、Windows では使用できますが UNIX では使用できません。たとえば、レポートの開発者がレポートの設計時に Tahoma フォントを使用したとします。このレポートは開発プラットフォームである Windows で出力すると、問題なく表示されます。次に、このレポートを配布プラットフォーム (Linux など) に移植するとします。このレポートの実行要求を Reports Server に送信すると、Reports Server は Tahoma フォントのメトリックを探します。しかし、メトリック・ファイルは見つかりません。Tahoma は Windows 固有のフォントであるためです。その場合は、Tahoma に似た別のフォントが使用されます。異なるフォントが使用されたため、レポート出力は異なって表示されます。

例 2: 開発プラットフォームと配布プラットフォームが Windows であるとします。開発プラットフォームと配布プラットフォーム上の Reports Server は、どちらも Windows で稼働しているため、Tahoma フォントにアクセスできます。しかし、エンドユーザーが Linux で出力を表示するとします。すべてのレポート出力フォーマット (HTML、HTMLCSS、RTF および PDF) ではフォントが参照されるだけで、PDF のフォントの埋込み機能を使用しないかぎり出力に埋め込まれることはありません。そのため、クライアント・システムはレポート出力を表示するために Tahoma フォントを探します。しかし、Tahoma は Linux で使用できないため、ユーザーが出力を表示するときにクロス・プラットフォームの問題が発生します。

7.1.2 フォントに関連する問題の解決

これまで説明したように、多くのクロス・プラットフォームの問題は、Reports Server が実行されている本番環境かクライアント・システムのいずれかでフォントが使用できないところに原因があります。このようなフォントの利用可能性に関する問題は、次の 3 つの段階で解決する必要があります。

- 1. 開発プラットフォーム:** レポートを作成する場合は、常に配布プラットフォームで利用できるフォントを考慮します。Windows (TTF ファイル) で使用可能なすべてのフォント・ファイルが、UNIX (AFM ファイルまたは TFM ファイル) でも使用できるとは限りません。UNIX プラットフォームで使用できる適切な AFM または TFM フォント・ファイルがある場合は、そのフォント・ファイルを使用できます (PostScript 出力の場合は AFM ファイル、PCL の場合は TFM ファイル)。UNIX 上に使用可能な AFM フォント・ファイルがない場合、または文字の不揃いなどレポート出力にフォントの問題が発生した場合は、ttf2pt1 など、サード・パーティの無料ユーティリティを使用して Windows の TTF ファイルを変換し、AFM ファイルを生成できます。TFM ファイルには変換しないでください。予測したとおりの結果にならない場合があります。
- 2. 配布プラットフォーム:** レポートに存在するフォントを使用できることを確認します。PDF 出力の場合は、フォント・エイリアシングを使用して、使用不可能なフォントをそれに最も類似したフォントに置換します。どの出力フォーマットでもグローバル・フォント・エイリアシングを使用できます。
- 3. クライアント・プラットフォーム:** クライアント・システムでフォントが使用できることを確認します。たとえば、PDF 出力では、第 6 章「Oracle Reports での PDF の使用」に記載されているように、フォント・サブセットまたはフォントの埋込みを使用できます。HTML、HTMLCSS または RTF 出力フォーマットでは、フォントを埋め込むことができません。そのため、レポートの設計時には、すべてのプラットフォームで利用できることがわかっているフォントを使用するようにします。

7.2 HTMLCSS、RTF または Web 出力の生成

表 7-1 に、宛先フォーマットが HTMLCSS、RTF または Web である場合のクロス・プラットフォームの配布シナリオを示します。

表 7-1 クロス・プラットフォームの配布 - シナリオ 1

開発プラットフォーム	配布プラットフォーム	宛先フォーマット
Windows	UNIX	HTMLCSS、RTF または Web

ここでは、HTMLCSS、RTF または Web 出力を想定したレポートの設計および配布を、次の項で説明します。

- レポートの設計
- レポートの配布
- よくある質問

7.2.1 レポートの設計

UNIX プラットフォームにレポートを配布する前に、次の手順を実行してレポートを準備します。

1. 新しいレポートを作成します。レポートを作成するときは、ボイラープレートやフィールドのオブジェクトに余分なスペースを確保しておきます。これは、配布プラットフォームでレポートを実行した場合に、テキスト幅が拡張されてもボックスのサイズに収まるようにするためです。
2. レポートで使用するフォントについて、次の項目を確認します。
 - UNIX で使用可能なフォントを使用します。Windows (TTF ファイル) で使用可能なすべてのフォント・ファイルが、UNIX (AFM ファイルまたは TFM ファイル) でも使用できるとは限りません。UNIX プラットフォームで使用できる適切な AFM または TFM フォント・ファイルがある場合は、そのフォント・ファイルを使用できます (PostScript の場合は AFM ファイル、PCL の場合は TFM ファイル)。

注意： AFM がサポートされるのは、日本語エンコードを除き、シングルバイトの PostScript ファイル生成時のみです。

AFM ファイルに対してサポートされるコード体系は、次のとおりです。

```
AdobeStandardEncoding
ExtJIS12-88-CFEncoding
FontSpecific
HRoman
ISOLatinHebrew
JIS12-88-CFEncoding
JIS12e-88-CFEncoding
```

- 適切に拡大または縮小可能なフォントを使用します。たとえば、Tahoma は異なるフォント・サイズに適切に拡張または縮小できますが、MS Sans Serif はできません。これは、MS Sans Serif がラスター・フォントであるためです。ラスター・フォントは任意のサイズに変更できず、一般的に表示が粗くなる問題があります。一方、Tahoma は TrueType フォントであり、MS Sans Serif フォントと非常に似ています。また Tahoma はベクトル・フォントであるため、任意のサイズに変更したり、適切な角度にフォントを回転したりできます。

7.2.2 レポートの配布

UNIX 上に使用可能な AFM フォント・ファイルがない場合、または文字の不揃いなどレポート出力にフォントの問題が発生した場合は、ttf2pt1 など、サード・パーティの無料ユーティリティを使用して Windows の TTF ファイルを変換し、AFM ファイルを生成できます。TFM ファイルには変換しないでください。予測したとおりの結果にならない場合があります。

AFM フォント・ファイルを使用できない場合に UNIX プラットフォームにレポートを配布するには、次の作業が必要です。

1. レポートに使用されているフォントに対応する TTF ファイルを探します。これらの TTF ファイルを AFM に変換して、レポートに使用されているフォント用の AFM ファイルを作成します。

TTF ファイルから AFM ファイルへの変換には、TrueType から Type1 へのフォント変換ユーティリティを使用します。たとえば、ttf2pt1 を使用します。

2. 変換後、AFM ファイルの名前から拡張子 .afm を削除します。次に例を示します。

表 7-2 変換後のフォント・ファイル名

変換前	変換後	変更した名前
arial.ttf	arial.afm	Arial
cour.ttf	cour.afm	CourierNew

3. 変換した AFM ファイルを \$ORACLE_HOME/guicommon/tk/admin/AFM ディレクトリにコピーします。
4. 任意のテキスト・エディタを使用して、screenprinter.ppd ファイルを編集します。

注意： ORACLE_HOME/guicommon/tk/admin/uiprint.txt にエントリーを追加してデフォルトのプリンタを定義している場合は、プリンタの PPD ファイル (PostScript プリンタ) または HPD ファイル (PCL プリンタ) に適切なエントリーが必要です。

Oracle Reports 10g (9.0.4) から、デフォルトのプリンタを設定していない場合は、画面を擬似実行するデフォルトのプリンタ・サーフェス (screenprinter.ppd) がフォーマットで使用されるようになりました。デフォルトのプリンタを設定していない場合は、screenprinter.ppd ファイルに必要なフォントや解像度のエントリーも追加する必要があります。

PPD ファイルは次の場所にあります。

ORACLE_HOME/guicommon/tk/admin/PPD

HPD ファイルは次の場所にあります。

ORACLE_HOME/guicommon/tk/admin/HPD

screenprinter.ppd ファイルの詳細は、第 3.10.1 項「ScreenPrinter」を参照してください。

使用する PPD/HPD ファイルに、レポートに使用されている AFM ファイルまたは TFM ファイルすべてのエントリーがあることを確認します。PPD/HPD ファイルはプリンタ・ドライバの設定を含む構成ファイルであり、そのプリンタでサポートされるすべてのフォントがリストされています。

PPD ファイルの Font Information セクションに移動して、必要なフォント・ファイルのエントリーを次のような形式で追加します。

*FONTNAME:ENCODING:VERSION:LOCATION

次に例を示します。

*Font Arial: Standard "(Version 2.76)" Standard ROM

*Font CourierNew: Standard "(Version 2.76)" Standard ROM

AFM ファイル名が、PPD ファイルに指定されているフォント名と一致していることを確認します。Oracle Reports は、PPD ファイルのフォント名に基づいて AFM ファイルを検索します。

5. レポートに使用されているフォントがエイリアシングされていないことを確認します。

たとえば、uifont.ali ファイルを編集し、[Global] セクションのエントリをコメント化します。この例では、Arial と Courier New のエイリアシングの対象に Helvetica と Courier が指定されています。

```
[ Global ] # Put mappings for all surfaces here.
# Mapping from MS Windows
#Arial = helvetica
#"Courier New" = courier
```

これで、Arial と Courier New は別のフォントにエイリアシングされません。

注意: uifont.ali は \$ORACLE_HOME /guicommon/tk/admin ディレクトリにあります。

フォントのエイリアシングは、特定のフォントについて AFM ファイルが生成不可能な場合にのみ使用してください。その場合は、対象フォントを最も類似度が高いフォントにエイリアシングできます。エイリアシングのフォントは、レポート出力を生成するマシンにはなくても、レポート出力を表示するマシンでは必ず使用できる必要があります。

6. レポートを実行します。

```
http://mywebserver.com:reports/rwservlet?server=myserver+report="c:¥test.rdf"+a
uthid=hr/hr@mydb+desformat=htmlcss+destype=cache
```

レポートの HTMLCSS 出力は、Windows で生成する場合とまったく同様に表示されます。

7.2.2.1 トラブルシューティング情報

配布に関する問題が発生した場合は、次のトラブルシューティング情報を参照してください。

- HTMLCSS 出力でフォントが正しく表示されない場合は、環境変数 DEBUG_SLFIND を、debug.txt などのログ・ファイル名に設定してからレポートを実行します。PPD ファイルと使用されているフォントの解析時に参照されるフォント・ファイルが、ログ・ファイル debug.txt に書き込まれます。特に次の項目について確認してください。
 1. 変更した PPD ファイルが参照されているはずですが、参照されていない場合は、構成に問題があります。第 4 章「Oracle Reports でのフォントの管理」を参照してください。
 2. 次に、AFM ディレクトリにコピーした AFM ファイルが参照されているはずですが、DEBUG_SLFIND の詳細は、第 5 章「UNIX での Oracle Reports による印刷」を参照してください。

7.2.3 よくある質問

この項では、レポートを HTMLCSS、RTF または Web に配布する作業に関連したよくある質問 (FAQ) に対して回答します。

質問

Windows でイタリックや太字などのフォント・スタイルを指定してレポートを設計するのに、UNIX で実行すると、Windows の場合と同様に表示されません。なぜでしょうか。

回答

UNIX では、レポートをフォーマットする場合、フォントに対応する AFM ファイルが使用されます。デフォルトでは、これらの AFM ファイルは、インストール時に指定された `$ORACLE_HOME/guicommon/tk/admin/AFM` ディレクトリから取得されます。レポート上のフォント・スタイルに対応する AFM ファイルが UNIX に存在しない場合は、最も類似した AFM ファイルが使用されます。たとえば、Windows で Courier Italic フォントを使用してレポートを設計し、UNIX でそのレポートを実行した場合、標準の Courier フォントしか表示されない場合があります。これは、`$ORACLE_HOME/guicommon/tk/admin/AFM` ディレクトリに、Courier Italic フォントに対応した AFM ファイルが存在しないため、Courier が使用されたことが原因です。この問題を解決するには、AFM ファイルが存在し、同じスタイルを持つ別のフォントにエイリアシングします。たとえば、`uifont.ali` の `global` セクションで、Courier Italic のエイリアシングとして Times Italic を指定します。また、Windows には、太字、イタリック、太字のイタリックなどのバージョンを持つフォントがあります。たとえば、Arial には `arialbd.ttf` (Arial bold)、`ariali.ttf` (Arial italic) および `arialbi.ttf` (Arial bold italic) があります。そのため、太字、イタリック、太字のイタリックの TTF ファイルがあるフォントを使用している場合は、`ttf2pt1` を使用してそれらの TTF ファイルから AFM ファイルを生成し、生成した AFM ファイルを UNIX で使用できます。

質問

レポートを Windows で作成し、HP-UX 11 に配布しました。HP-UX 11 ではフォント・スタイルは正しく表示されますが、行間隔が正しく表示されず、テキストによっては割り当てられたスペースに収まらないものもあります。テキストを正しく表示するには、行間隔をどのように修正すればよいですか。

回答

ドキュメント内のすべてのフォントに対して、対応する AFM ファイルを設定します。詳細は、[第 7.2.1 項「レポートの設計」](#)を参照してください。

質問

レポートを Windows で設計しました。このレポートを別のプラットフォームに配布すると、文字化けが起きます。たとえば、一部のフィールドには、実際の内容のかわりに ***** が表示されます。これは間隔の問題でしょうか。

回答

Oracle Reports が、レポートに使用されているフォントの AFM ファイルを検索して、見つけれなかった可能性があります。これは、レポートの HTML ソースを開き、使用しているフォントを探すと確認できます。

次に、Oracle Reports が最も近いフォントを使用し、そのメトリックが元のフォントよりも大きい可能性があります。そのため、文字がボックスに収まらず、実際の出力のかわりに **** がフィールドに表示された可能性があります。

次の点を確認します。

- レポートに使用されているすべてのフォントについて、AFM ファイルを設定している。詳細は、[第 7.2.1 項「レポートの設計」](#)を参照してください。
- フィールドやテキスト・ボイラープレートに 10% ぐらいの余分なスペースを確保している。

質問

レポートを UNIX で実行すると、HTML または HTMLCSS 出力が縮小されて表示されます。しかし、同じレポートを Windows で実行すると正しく表示されます。レポートを Windows で実行しても UNIX で実行しても同じように表示されるようにするには、どのようにすればよいですか。

回答

Oracle9i Reports を使用している場合に、HTMLCSS 出力で縮小や拡張が見られる場合は、`reports.sh` 内の環境変数 `REPORTS_DEFAULT_PIXEL_SIZE` を 72 ~ 200 の範囲の値に設定してから、Reports Server を再起動します。

次に例を示します。

```
REPORTS_DEFAULT_PIXEL_SIZE =72
export REPORTS_DEFAULT_PIXEL_SIZE
```

Oracle Reports 10g であれば、`screenprinter.ppd` に設定されている解像度が使用されるため、HTMLCSS 出力が縮小または拡張する問題は起こりません。この解像度は編集可能です。

*DefaultResolution: 96dpi (recommended)

質問

レポートを Solaris に配布する場合、`overstrike` プロパティを使用できますか。

回答

AFM ファイルには、`overstrike` プロパティがサポートされないという制約があります。

質問

レポート内の正と負の数値を示すフィールドには、右揃えを指定しています。たとえば、12345.67 と -12345.67 の数値です。しかし、レポートを HTMLCSS 出力で生成すると、位置合せが正しく表示されません。どのようにすれば正しく位置合せできますか。これはプラットフォーム固有の問題でしょうか。

回答

これは、プラットフォーム固有の問題ではありません。HTMLCSS 出力のスペースを ` ` で置換すると、この問題は解決されます。この問題は、Oracle9i Reports パッチ・リリース 1以降では修正されています。この問題を回避するには、最新のリリースの Oracle Reports にアップグレードする必要があります。

7.3 シングルバイト PDF 出力の生成

表 7-3 に、宛先フォーマットが PDF フォント・サブセットを使用して作成したシングルバイト PDF である場合のクロス・プラットフォームの配布シナリオを示します。PDF のフォント機能の詳細は、第 6 章「Oracle Reports での PDF の使用」を参照してください。

表 7-3 クロス・プラットフォームの配布 - シナリオ 2

開発プラットフォーム	配布プラットフォーム	宛先フォーマット
Windows	UNIX	PDF (シングルバイト)

ここでは、シングルバイト PDF 出力を想定したレポートの設計および配布を、次の項で説明します。

- レポートの設計
- レポートの配布
- よくある質問

7.3.1 レポートの設計

UNIX プラットフォームにレポートを配布する前に、次の手順を実行してレポートを準備します。

1. 新しいレポートを作成します。
2. レポートで使用するフォントについて、次の項目を確認します。
 - UNIX で使用可能なフォントを使用します。Windows (TTF ファイル) で使用可能なすべてのフォント・ファイルが、UNIX (AFM ファイルまたは TFM ファイル) でも使用できるとは限りません。UNIX プラットフォームで使用できる適切な AFM または TFM フォント・ファイルがある場合は、そのフォント・ファイルを使用できます (PostScript の場合は AFM ファイル、PCL の場合は TFM ファイル)。

注意: AFM がサポートされるのは、日本語エンコードを除き、シングルバイトの PostScript ファイル生成時のみです。

AFM ファイルに対してサポートされるコード体系は、次のとおりです。

```
AdobeStandardEncoding
ExtJIS12-88-CFEncoding
FontSpecific
HRoman
ISOLatinHebrew
JIS12-88-CFEncoding
JIS12e-88-CFEncoding
```

- 適切に拡大または縮小可能なフォントを使用します。たとえば、Tahoma は異なるフォント・サイズに適切に拡張または縮小できますが、MS Sans Serif はできません。これは、MS Sans Serif がラスター・フォントであるためです。ラスター・フォントは任意のサイズに変更できず、一般的に表示が粗くなる問題があります。一方、Tahoma は TrueType フォントであり、MS Sans Serif フォントと非常に似ています。また Tahoma はベクトル・フォントであるため、任意のサイズに変更したり、適切な角度にフォントを回転したりできます。

7.3.2 レポートの配布

UNIX 上に使用可能な AFM フォント・ファイルがない場合、または文字の不揃いなどレポート出力にフォントの問題が発生した場合は、ttf2pt1 など、サード・パーティの無料ユーティリティを使用して Windows の TTF ファイルを変換し、AFM ファイルを生成できます。TFM ファイルには変換しないでください。予測したとおりの結果にならない場合があります。

PDF フォント・サブセットを使用して UNIX プラットフォームにレポートを配布するには、次の作業が必要です。

1. レポートに使用されているフォントに対応する TTF ファイルを探します。これらの TTF ファイルを AFM に変換して、レポートに使用されているフォント用の AFM ファイルを作成します。

TTF ファイルから AFM ファイルへの変換には、TrueType から Type1 へのフォント変換ユーティリティを使用します。たとえば、ttf2pt1 を使用します。

2. 変換後、AFM ファイルの名前から拡張子 .afm を削除します。次に例を示します。

表 7-4 変換後のフォント・ファイル名

変換前	変換後	変更した名前
arial.ttf	arial.afm	Arial
cour.ttf	cour.afm	CourierNew

3. レポートに使用した Windows の TTF ファイルを、UNIX マシンのフォント・ディレクトリにコピーします。たとえば、\$ORACLE_HOME/reports/fonts にコピーします。
4. REPORTS_PATH 環境変数に TTF ファイルへのパスを追加します。これにより、Reports Runtime はフォント・ファイルを参照できるようになります。
5. 任意のテキスト・エディタを使用して、screenprinter.ppd ファイルを編集します。

注意: `ORACLE_HOME/guicommon/tk/admin/uiprint.txt` にエントリを追加してデフォルトのプリンタを定義している場合は、プリンタの PPD ファイル (PostScript プリンタ) または HPD ファイル (PCL プリンタ) に適切なエントリが必要です。

Oracle Reports 10g (9.0.4) から、デフォルトのプリンタを設定していない場合は、画面を擬似実行するデフォルトのプリンタ・サーフェス (`screenprinter.ppd`) がフォーマットで使用されるようになりました。デフォルトのプリンタを設定していない場合は、`screenprinter.ppd` ファイルに必要なフォントや解像度のエントリも追加する必要があります。

PPD ファイルは次の場所にあります。

`ORACLE_HOME/guicommon/tk/admin/PPD`

HPD ファイルは次の場所にあります。

`ORACLE_HOME/guicommon/tk/admin/HPD`

`screenprinter.ppd` ファイルの詳細は、第 3.10.1 項「ScreenPrinter」を参照してください。

使用する PPD/HPD ファイルに、レポートに使用されている AFM ファイルまたは TFM ファイルすべてのエントリがあることを確認します。PPD/HPD ファイルはプリンタ・ドライバの設定を含む構成ファイルであり、そのプリンタでサポートされるすべてのフォントがリストされています。

PPD ファイルの Font Information セクションに移動して、必要なフォント・ファイルのエントリを次のような形式で追加します。

*FONTNAME:ENCODING:VERSION:LOCATION

次に例を示します。

*Font Arial: Standard "(Version 2.76)" Standard ROM

*Font CourierNew: Standard "(Version 2.76)" Standard ROM

AFM ファイル名が、PPD ファイルに指定されているフォント名と一致していることを確認します。Oracle Reports は、PPD ファイルのフォント名に基づいて AFM ファイルを検索します。

6. 変換した AFM ファイルを `$ORACLE_HOME/guicommon/tk/admin/AFM` ディレクトリにコピーします。
7. `uiprint.txt` に、次のような適切な PPD ファイルのエントリがあることを確認します。

printer name:PostScript:2:test:ppd file

次に例を示します。

printer1:PostScript:2:test:hpljet42.ppd

8. 任意のテキスト・エディタを使用して、`hpljet42.ppd` ファイルを編集します。

注意: `hpljet42.ppd` ファイルを編集する場合は、あらかじめそのファイルのバックアップを作成しておきます。このファイルは次の場所にあります。

`$ORACLE_HOME/guicommon/tk/admin/PPD`

9. 使用する PPD/HPD ファイルに、レポートに使用されている AFM ファイルまたは TFM ファイルすべてのエントリがあることを確認します。

Font Information セクションに移動して、新しい AFM ファイルに必要なエントリを次のような形式で追加します。

```
*FONTNAME:ENCODING:VERSION:LOCATION
```

次に例を示します。

```
*Font Arial: Standard "(Version 2.76)" Standard ROM
*Font CourierNew: Standard "(Version 2.76)" Standard ROM
```

AFM ファイル名が、PPD ファイルに指定されているフォント名と一致していることを確認します。Oracle Reports は、PPD ファイルのフォント名に基づいて AFM ファイルを検索します。

10. レポートに使用されているフォントがエイリアシングされていないことを確認します。

たとえば、uifont.ali を編集し、[Global] セクションのエントリをコメント化します。この例では、Arial と Courier New のエイリアシングの対象に Helvetica と Courier がデフォルトで指定されています。

```
[ Global ] # Put mappings for all surfaces here.
# Mapping from MS Windows
#Arial = helvetica
#"Courier New" = courier
```

注意: uifont.ali は \$ORACLE_HOME /guicommon/tk/admin ディレクトリにあります。

フォントのエイリアシングは、特定のフォントについて AFM ファイルが生成不可能な場合にのみ使用してください。その場合は、対象フォントを最も類似度が高いフォントにエイリアシングできます。エイリアシングのフォントは、PDF ファイルを生成するマシンにはなくても、PDF 出力を表示するマシンでは必ず使用できる必要があります。

11. [PDF:Subset] セクションに、レポートで使用されているフォントのサブセットを作成するために必要なエントリを追加します。

次に例を示します。

```
[ PDF:Subset ]
"arial" = "arial.ttf"
"courier new" = "cour.ttf"
```

PDF ファイルの移植性を実現するため、フォント・サブセットまたはフォント埋込みのいずれかを使用できます。この移植性によって、レポートを表示するマシンにフォントがインストールされているかどうかにかかわらず、PDF レポートを表示できます。

12. レポートを PDF で実行して表示します。PDF には、レポートで使用されているフォントが含まれているはずですが、この例では、Arial と Courier New フォントです。

使用されているフォントを確認するには、次の手順を実行します。

- Acrobat Reader 6.0 の場合は、「ファイル」→「文書のプロパティ」→「フォント」を選択します。
- Acrobat Reader 3.0 以降では、「ファイル」→「文書情報」→「フォント」を選択します。

「オリジナルフォント」列に Arial フォントと Tahoma フォントが表示されます。これで、フォントの位置のずれがすべて修正されたはずですが。

7.3.2.1 トラブルシューティング情報

配布に関する問題が発生した場合は、次のトラブルシューティング情報を参照してください。

- PDF 出力でフォントが正しく表示されない場合は、環境変数 `DEBUG_SLFIND` を、`debug.txt` などのログ・ファイル名に設定してからレポートを実行します。PPD ファイルと使用されているフォントの解析時に参照されるフォント・ファイルが、指定したファイルに書き込まれます。特に次の項目について確認してください。
 1. 変更した PPD ファイルが参照されているはずですが、参照されていない場合は、構成に問題があります。第 4 章「Oracle Reports でのフォントの管理」を参照してください。
 2. 次に、AFM ディレクトリにコピーした AFM ファイルが参照されているはずですが、`DEBUG_SLFIND` の詳細は、第 5 章「UNIX での Oracle Reports による印刷」を参照してください。

7.3.3 よくある質問

この項では、レポートをシングルバイト PDF 出力に配布する作業に関連したよくある質問 (FAQ) に回答します。

質問

PDF レポートを Windows プラットフォームと UNIX プラットフォームに配布したときではページ数が異なります。どのようにすれば解決できますか？

回答

レポートは、デフォルトのプリンタを使用してフォーマットされます。そのため、両方のプリンタで同じ解像度とフォントが使用されている必要があります。その 1 つの方法としては、Windows TTF フォント・ファイルから AFM ファイルを生成し、Windows の TTF ファイルと AFM ファイルを UNIX の適切なフォルダにコピーします。また、PPD/HPD ファイルに Windows と同じ解像度を設定します。詳細は、前述の手順を参照してください。

質問

UNIX の異なるインストールでレポートを実行すると、レポートのページ数が異なります。インストールに関係なくレポートのページ数を同じにするには、どうすればよいでしょうか？

回答

UNIX では、Oracle Reports は各インストールのデフォルト・プリンタの PPD/HPD ファイルを使用してフォーマットを行います。解像度やフォント・リストは、この PPD/HPD ファイルから取得されます。Oracle Reports 10g (9.0.4) から、インストールにデフォルトのプリンタが設定されていない場合は、`screenprinter.ppd` が使用されるようになりました。この PPD ファイルによって画面がエミュレートされます。Oracle Reports の初期のバージョンでは、かわりに `DISPLAY` 環境変数が使用されていました。2 つのインストール間で同じ AFM/TFM ファイルとフォント・ファイルが使用されていることを確認します。これで、PDF 出力のページ数は同じになるはずですが。

7.4 マルチバイト PDF 出力の生成

表 7-5 に、宛先フォーマットが PDF フォント・サブセットを使用して作成したマルチバイト PDF である場合のクロス・プラットフォームの配布シナリオを示します。PDF のフォント機能の詳細は、第 6 章「Oracle Reports での PDF の使用」を参照してください。

表 7-5 クロス・プラットフォームの配布 - シナリオ 4

開発プラットフォーム	配布プラットフォーム	宛先フォーマット
Windows	UNIX	PDF (マルチバイト)

ここでは、マルチバイト PDF 出力を想定したレポートの設計および配布を、次の項で説明します。

- レポートの設計
- レポートの配布
- よくある質問

7.4.1 レポートの設計

UNIX プラットフォームにレポートを配布する前に、次の手順を実行してレポートを準備します。

1. TrueType マルチバイト・フォントを使用して、新しいレポートを作成します。たとえば、AR8ISO8859P6 キャラクタ・セットの Simplified Arabic フォントを使用します。
2. レポートで使用するフォントについて、次の項目を確認します。
 - UNIX で使用可能なフォントを使用します。Windows (TTF ファイル) で使用可能なすべてのフォント・ファイルが、UNIX (AFM ファイルまたは TFM ファイル) でも使用できるとは限りません。UNIX プラットフォームで使用できる適切な AFM または TFM フォント・ファイルがある場合は、そのフォント・ファイルを使用できます (PostScript の場合は AFM ファイル、PCL の場合は TFM ファイル)。

注意： AFM がサポートされるのは、日本語エンコードを除き、シングルバイトの PostScript ファイル生成時のみです。

AFM ファイルに対してサポートされるコード体系は、次のとおりです。

```
AdobeStandardEncoding
ExtJIS12-88-CFEncoding
FontSpecific
HRoman
ISOLatinHebrew
JIS12-88-CFEncoding
JIS12e-88-CFEncoding
```

- 適切に拡大または縮小可能なフォントを使用します。たとえば、Tahoma は異なるフォント・サイズに適切に拡張または縮小できますが、MS Sans Serif はできません。これは、MS Sans Serif がラスター・フォントであるためです。ラスター・フォントは任意のサイズに変更できず、一般的に表示が粗くなる問題があります。一方、Tahoma は TrueType フォントであり、MS Sans Serif フォントと非常に似ています。また Tahoma はベクトル・フォントであるため、任意のサイズに変更したり、適切な角度にフォントを回転したりできます。

7.4.2 レポートの配布

UNIX 上に使用可能な AFM フォント・ファイルがない場合、または文字の不揃いなどレポート出力にフォントの問題が発生した場合は、ttf2pt1 など、サード・パーティの無料ユーティリティを使用して Windows の TTF ファイルを変換し、AFM ファイルを生成できます。TFM ファイルには変換しないでください。予測したとおりの結果にならない場合があります。

レポートを UNIX プラットフォームに配布するには、次の手順を実行します。

1. レポートに使用されているフォントに対応する TTF ファイルを探します。これらの TTF ファイルを AFM に変換して、レポートに使用されているフォント用の AFM ファイルを作成します。

TTF ファイルから AFM ファイルへの変換には、TrueType から Type1 へのフォント変換ユーティリティを使用します。たとえば、`ttf2pt1` を使用します。

2. 変換後、AFM ファイルの名前から拡張子 `.afm` を削除します。次に例を示します。

表 7-6 変換後のフォント・ファイル名

変換前	変換後	変更した名前
<code>simpo.ttf</code>	<code>simpo.afm</code>	<code>SimplifiedArabic</code>

3. レポートに使用した Windows の TTF ファイルを、UNIX マシンのフォント・ディレクトリにコピーします。たとえば、`$ORACLE_HOME/fonts` にコピーします。
4. `REPORTS_PATH` 環境変数に TTF ファイルへのパスを追加します。
5. AFM ファイルを `$ORACLE_HOME/guicommon/tk/admin/AFM` ディレクトリにコピーします。
6. 任意のテキスト・エディタを使用して、`screenprinter.ppd` ファイルを編集します。

注意： `ORACLE_HOME/guicommon/tk/admin/uiprint.txt` にエントリを追加してデフォルトのプリンタを定義している場合は、プリンタの PPD ファイル (PostScript プリンタ) または HPD ファイル (PCL プリンタ) に適切なエントリが必要です。

Oracle Reports 10g (9.0.4) から、デフォルトのプリンタを設定していない場合は、画面を擬似実行するデフォルトのプリンタ・サーフェス (`screenprinter.ppd`) がフォーマットで使用されるようになりました。デフォルトのプリンタを設定していない場合は、`screenprinter.ppd` ファイルに必要なフォントや解像度のエントリも追加する必要があります。

PPD ファイルは次の場所にあります。

`ORACLE_HOME/guicommon/tk/admin/PPD`

HPD ファイルは次の場所にあります。

`ORACLE_HOME/guicommon/tk/admin/HPD`

`screenprinter.ppd` ファイルの詳細は、[第 3.10.1 項「ScreenPrinter」](#) を参照してください。

使用する PPD/HPD ファイルに、レポートに使用されている AFM ファイルまたは TFM ファイルすべてのエントリがあることを確認します。PPD/HPD ファイルはプリンタ・ドライバの設定を含む構成ファイルであり、そのプリンタでサポートされるすべてのフォントがリストされています。

PPD ファイルの `Font Information` セクションに移動して、必要なフォント・ファイルのエントリを次のような形式で追加します。

```
*FONTNAME:ENCODING:VERSION:LOCATION
```

次に例を示します。

```
*Font Arial: Standard "(Version 2.76)" Standard ROM
```

```
*Font CourierNew: Standard "(Version 2.76)" Standard ROM
```

AFM ファイル名が、PPD ファイルに指定されているフォント名と一致していることを確認します。Oracle Reports は、PPD ファイルのフォント名に基づいて AFM ファイルを検索します。

7. uiprint.txt に該当する PPD ファイルのエントリがあることを確認します。
printername: PostScript:2:test:ppd_file

次に例を示します。

```
printer1:PostScript:2:test:hpljet42.ppd
```

8. 任意のテキスト・エディタを使用して、hpljet42.ppd ファイルを編集します。

注意: hpljet42.ppd を編集する場合は、あらかじめそのファイルのバックアップを作成しておきます。このファイルは次の場所にあります。

```
$ORACLE_HOME/guicommon/tk/admin/PPD
```

9. 使用する PPD/HPD ファイルに、レポートに使用されている AFM ファイルまたは TFM ファイルすべてのエントリがあることを確認します。

Font Information セクションに移動して、新しい AFM ファイルに必要なエントリを次のような形式で追加します。

```
*FONTNAME:ENCODING:VERSION:LOCATION
```

次に例を示します。

```
*Font SimplifiedArabic: Standard "(001.01)" Standard ROM
```

10. レポートに使用されているフォントがエイリアシングされていないことを確認します。

uifont.ali を編集して、Simplified Arabic フォントが他のフォントにエイリアシングされている場合は、そのエントリをコメント化します。たとえば、
"SimplifiedArabic"="Arial" のようなエントリです。

注意: uifont.ali は \$ORACLE_HOME /guicommon/tk/admin ディレクトリにあります。

フォントのエイリアシングは、特定のフォントについて AFM ファイルが生成不可能な場合にのみ使用してください。その場合は、対象フォントを最も類似度が高いフォントにエイリアシングできます。エイリアシングのフォントは、PDF ファイルを生成するマシンにはなくても、PDF 出力を表示するマシンでは必ず使用できる必要があります。

11. [PDF:Subset] セクションで、フォントのサブセットを作成するために必要なエントリを追加します。

次に例を示します。

```
[ PDF:Subset ]  
"SimplifiedArabic" = "simpo.ttf"
```

PDF ファイルの移植性を実現するため、フォント・サブセットまたはフォント埋込みのいずれかを使用できます。この移植性によって、レポートを表示するマシンにフォントがインストールされているかどうかにかかわらず、PDF レポートを表示できます。

12. レポートを PDF で実行して表示します。

```
http://mywebserver.com:reports/rwservlet?server=myserver+report=  
"c:¥test.rdf"+authid=hr/hr@mydb+desformat=PDF+destype=cache
```

PDF には、レポートで使用されているフォントが含まれているはずですが、この例では、David フォントなどです。

使用されているフォントを確認するには、次の手順を実行します。

- Acrobat Reader 6.0 の場合は、「ファイル」→「文書のプロパティ」→「フォント」を選択します。
- Acrobat Reader 3.0 以降の場合は、「ファイル」→「文書情報」→「フォント」をクリックします。

「オリジナルフォント」列に David フォントが表示されるはずですが。

7.4.2.1 トラブルシューティング情報

配布に関する問題が発生した場合は、次のトラブルシューティング情報を参照してください。

- PDF 出力でフォントが正しく表示されない場合は、環境変数 `DEBUG_SLFIND` を、`debug.txt` などのログ・ファイル名に設定してからレポートを実行します。PPD ファイルと使用されているフォントの解析時に参照されるフォント・ファイルが、ログ・ファイル `debug.txt` に書き込まれます。特に次の項目について確認してください。
 1. 変更した PPD ファイルが参照されているはずですが。参照されていない場合は、構成に問題があります。第 4 章「Oracle Reports でのフォントの管理」を参照してください。
 2. 次に、AFM ディレクトリにコピーした AFM ファイルが参照されているはずですが。
DEBUG_SLFIND の詳細は、第 5 章「UNIX での Oracle Reports による印刷」を参照してください。

7.4.3 よくある質問

この項では、レポートをマルチバイト PDF 出力に配布する作業に関連したよくある質問 (FAQ) に回答します。

質問

CID マルチバイト・フォントを使用してクロス・プラットフォームで PDF を出力する場合は、どのような点に注意する必要がありますか。

回答

CID マルチバイト・フォントを使用して PDF レポートをマルチバイト言語に対応させるには、PDF ファイルを表示するマシンの Acrobat Reader にアジア言語フォント・パッケージをインストールする必要があります。アジア言語フォント・パッケージは Adobe の Web サイトから入手できます。

質問

UNIX でマルチバイト・フォントを使用する場合、レポートのフォーマットが正しく表示されなかったり、テキストの位置が変わるのはなぜですか。

回答

`ttf2pt1` を使用して AFM ファイルを作成する場合、`ttf2pt1` には、そのフォントの最初の 256 文字のみのメトリック情報を使用して AFM ファイルを作成するという制約があります。最初の 256 文字は Latin-1 文字に使用されます。そのため、フォントに 256 以上の文字が含まれる場合、それを越えた文字に対してメトリック情報を使用できません。Oracle Reports は、AFM ファイルにない文字をレポートで検出した場合、AFM ファイルにあるデフォルトのメトリック情報を使用します。これらのデフォルトのメトリック情報は、レポートに使用されている文字のメトリックと一致しない場合があります。そのため、フォーマットが正しく行われません。256 文字以上あるフォントを使用する場合にこの問題を解決するには、すべての文字が同じ幅になる固定幅フォントを使用します。たとえば、Hebrew の場合は固定幅フォントである Miriam Fixed を使用すると、フォーマットの問題を回避できます。

7.5 Unicode PDF 出力の生成

表 7-7 に、宛先フォーマットが PDF フォント・サブセットを使用して作成した Unicode PDF である場合のクロス・プラットフォームの配布シナリオを示します。PDF のフォント機能の詳細は、第 6 章「Oracle Reports での PDF の使用」を参照してください。

表 7-7 クロス・プラットフォームの配布 - シナリオ 5

開発プラットフォーム	配布プラットフォーム	宛先フォーマット
Windows	UNIX	PDF (Unicode)

ここでは、Unicode PDF 出力を想定したレポートの設計および配布を、次の項で説明します。

- レポートの設計
- レポートの配布
- よくある質問

7.5.1 レポートの設計

UNIX プラットフォームにレポートを配布する前に、次の手順を実行してレポートを準備します。

1. Unicode フォントを使用して新しいレポートを作成します。この例では、Arial Unicode MS フォントなどを使用します。
2. レポートで使用するフォントについて、次の項目を確認します。
 - レポートで使用する Unicode 文字の範囲全体をカバーするフォントを使用します。
 - UNIX で使用可能なフォントを使用します。Windows (TTF ファイル) で使用可能なすべてのフォント・ファイルが、UNIX (AFM ファイルまたは TFM ファイル) でも使用できるとは限りません。UNIX プラットフォームで使用できる適切な AFM または TFM フォント・ファイルがある場合は、そのフォント・ファイルを使用できます (PostScript の場合は AFM ファイル、PCL の場合は TFM ファイル)。

注意： AFM がサポートされるのは、日本語エンコードを除き、シングルバイトの PostScript ファイル生成時のみです。

AFM ファイルに対してサポートされるコード体系は、次のとおりです。

```
AdobeStandardEncoding
ExtJIS12-88-CFEncoding
FontSpecific
HRoman
ISOLatinHebrew
JIS12-88-CFEncoding
JIS12e-88-CFEncoding
```

- 適切に拡大または縮小可能なフォントを使用します。たとえば、Tahoma は異なるフォント・サイズに適切に拡張または縮小できますが、MS Sans Serif はできません。これは、MS Sans Serif がラスター・フォントであるためです。ラスター・フォントは任意のサイズに変更できず、一般的に表示が粗くなる問題があります。一方、Tahoma は TrueType フォントであり、MS Sans Serif フォントと非常に似ています。また Tahoma はベクトル・フォントであるため、任意のサイズに変更したり、適切な角度にフォントを回転したりできます。

7.5.2 レポートの配布

UNIX 上に使用可能な AFM フォント・ファイルがない場合、または文字の不揃いなどレポート出力にフォントの問題が発生した場合は、ttf2pt1 など、サード・パーティの無料ユーティリティを使用して Windows の TTF ファイルを変換し、AFM ファイルを生成できます。TFM ファイルには変換しないでください。予測したとおりの結果にならない場合があります。

レポートを UNIX プラットフォームに配布するには、次の手順を実行します。

1. レポートに使用されているフォントに対応する TTF ファイルを探します。これらの TTF ファイルを AFM に変換して、レポートに使用されているフォント用の AFM ファイルを作成します。

TTF ファイルから AFM ファイルへの変換には、TrueType から Type1 へのフォント変換ユーティリティを使用します。たとえば、ttf2pt1 を使用します。

2. 変換後、AFM ファイルの名前から拡張子 .afm を削除します。次に例を示します。

表 7-8 変換後のフォント・ファイル名

変換前	変換後	変更した名前
arialuni.ttf	arialuni.afm	ArialUnicodeMS

3. レポートに使用した Windows の TTF ファイルを、UNIX マシンのフォント・ディレクトリにコピーします。たとえば、\$ORACLE_HOME/fonts にコピーします。
4. REPORTS_PATH 環境変数に TTF ファイルへのパスを追加します。この例では、arialuni.ttf へのパスを追加します。
5. AFM ファイルを \$ORACLE_HOME/guicommon/tk/admin/AFM ディレクトリにコピーします。この例では、ArialUnicodeMS です。
6. 任意のテキスト・エディタを使用して、screenprinter.ppd ファイルを編集します。

注意： ORACLE_HOME/guicommon/tk/admin/uiprint.txt にエンTRIES を追加してデフォルトのプリンタを定義している場合は、プリンタの PPD ファイル (PostScript プリンタ) または HPD ファイル (PCL プリンタ) に適切なエンTRIES が必要です。

Oracle Reports 10g (9.0.4) から、デフォルトのプリンタを設定していない場合は、画面を擬似実行するデフォルトのプリンタ・サーフェス (screenprinter.ppd) がフォーマットで使用されるようになりました。デフォルトのプリンタを設定していない場合は、screenprinter.ppd ファイルに必要なフォントや解像度のエンTRIES も追加する必要があります。

PPD ファイルは次の場所にあります。

ORACLE_HOME/guicommon/tk/admin/PPD

HPD ファイルは次の場所にあります。

ORACLE_HOME/guicommon/tk/admin/HPD

screenprinter.ppd ファイルの詳細は、[第 3.10.1 項「ScreenPrinter」](#) を参照してください。

使用する PPD/HPD ファイルに、レポートに使用されている AFM ファイルまたは TFM ファイルすべてのエントリがあることを確認します。PPD/HPD ファイルはプリンタ・ドライバの設定を含む構成ファイルであり、そのプリンタでサポートされるすべてのフォントがリストされています。

PPD ファイルの `Font Information` セクションに移動して、必要なフォント・ファイルのエントリを次のような形式で追加します。

```
*FONTNAME:ENCODING:VERSION:LOCATION
```

次に例を示します。

```
*Font Arial: Standard "(Version 2.76)" Standard ROM
*Font CourierNew: Standard "(Version 2.76)" Standard ROM
```

AFM ファイル名が、PPD ファイルに指定されているフォント名と一致していることを確認します。Oracle Reports は、PPD ファイルのフォント名に基づいて AFM ファイルを検索します。

7. `uiprint.txt` に、次のような適切な PPD ファイルのエントリがあることを確認します。

```
printer name:PostScript:2:test:ppd file
```

この例では、次のようになります。

```
printer1:PostScript:2:test:hpljet42.ppd
```

8. 任意のテキスト・エディタを使用して、`hpljet42.ppd` ファイルを編集します。

注意： `hpljet42.ppd` を編集する場合は、あらかじめそのファイルのバックアップを作成しておきます。このファイルは次の場所にあります。

```
$ORACLE_HOME/guicommon/tk/admin/PPD
```

9. 使用する PPD/HPD ファイルに、レポートに使用されている AFM ファイルまたは TFM ファイルすべてのエントリがあることを確認します。

`Font Information` セクションに移動して、新しい AFM ファイルに必要なエントリを次のような形式で追加します。

```
*FONTNAME:ENCODING:VERSION:LOCATION
```

次に例を示します。

```
*Font ArialUnicodeMS: Standard "(Version 2.76)" Standard ROM
```

10. `[PDF:Subset]` セクションで、次のエントリを追加し、レポートに使用するフォントのサブセットを作成します。

次に例を示します。

```
[ PDF:Subset ]
"ArialUnicodeMS" = "arialuni.ttf"
```

PDF サブセット機能を使用してレポートからマルチバイト PDF 出力を生成し、PDF レポートを移植できるようにします。これにより、レポートを配布するマシンにインストールされているフォントに依存しなくなります。

11. レポートを PDF で実行して表示します。

```
http://mywebserver.com:reports/rwservlet?server=myserver+report=
"c:¥test.rdf"+authid=hr/hr@mydb+desformat=PDF+destype=cache
```

PDF には、レポートで使用されている Unicode フォントが含まれているはずですが、この例では、Arial フォントなどです。

7.5.2.1 トラブルシューティング情報

配布に関する問題が発生した場合は、次のトラブルシューティング情報を参照してください。

- PDF 出力でフォントが正しく表示されない場合は、環境変数 `DEBUG_SLFIND` を、`debug.txt` などのログ・ファイル名に設定してからレポートを実行します。PPD ファイルと使用されているフォントの解析時に参照されるフォント・ファイルが、ログ・ファイル `debug.txt` に書き込まれます。特に次の項目について確認してください。
 1. 変更した PPD ファイルが参照されているはずですが、参照されていない場合は、構成に問題があります。第 4 章「Oracle Reports でのフォントの管理」を参照してください。
 2. 次に、AFM ディレクトリにコピーした AFM ファイルが参照されているはずですが、`DEBUG_SLFIND` の詳細は、第 5 章「UNIX での Oracle Reports による印刷」を参照してください。

7.5.3 よくある質問

この項では、レポートを Unicode PDF 出力に配布する作業に関連したよくある質問 (FAQ) に回答します。

質問

UNIX で Unicode フォントを使用する場合、レポートのフォーマットが正しく表示されなかったり、テキストの位置が変わるのはなぜですか。

回答

`ttf2pt1` を使用して AFM ファイルを作成する場合、`ttf2pt1` には、そのフォントの最初の 256 文字のみのメトリック情報を使用して AFM ファイルを作成するという制約があります。複数の言語が含まれるレポートに Arial Unicode MS のような Unicode フォントを設定し、UNIX 用の AFM ファイルを作成する場合、最初の 256 文字を超える文字には対応するメトリック情報がありません。Oracle Reports は、AFM ファイルにない文字をレポートで検出した場合、AFM ファイルにあるデフォルトのメトリック情報を使用します。これらのデフォルトのメトリック情報は、レポートに使用されている文字のメトリックと一致しない場合があります。そのため、フォーマットが正しく行われません。256 文字以上のフォントを使用する場合には、フォーマットの問題を解決するには、すべての文字が同じ幅になる固定幅フォントを使用します。

7.6 PostScript 出力の生成

表 7-9 に、宛先フォーマットが PostScript である場合のクロス・プラットフォームの配布シナリオを示します。

表 7-9 クロス・プラットフォームの配布 - シナリオ 6

開発プラットフォーム	配布プラットフォーム	宛先フォーマット
Windows	UNIX	PostScript

ここでは、マルチバイト PostScript 出力を想定したレポートの設計および配布を、次の項で説明します。

- レポートの設計
- レポートの配布
- よくある質問

7.6.1 レポートの設計

UNIX プラットフォームにレポートを配布する前に、次の手順を実行してレポートを準備します。

1. 新しいレポートを作成します。
2. レポートで使用するフォントについて、次の項目を確認します。
 - UNIX で使用可能なフォントを使用します。Windows (TTF ファイル) で使用可能なすべてのフォント・ファイルが、UNIX (AFM ファイルまたは TFM ファイル) でも使用できるとは限りません。UNIX プラットフォームで使用できる適切な AFM または TFM フォント・ファイルがある場合は、そのフォント・ファイルを使用できます (PostScript の場合は AFM ファイル、PCL の場合は TFM ファイル)。

注意: AFM がサポートされるのは、日本語エンコードを除き、シングルバイトの PostScript ファイル生成時のみです。

AFM ファイルに対してサポートされるコード体系は、次のとおりです。

```
AdobeStandardEncoding
ExtJIS12-88-CFEncoding
FontSpecific
HRoman
ISOLatinHebrew
JIS12-88-CFEncoding
JIS12e-88-CFEncoding
```

- 適切に拡大または縮小可能なフォントを使用します。たとえば、Tahoma は異なるフォント・サイズに適切に拡張または縮小できますが、MS Sans Serif はできません。これは、MS Sans Serif がラスター・フォントであるためです。ラスター・フォントはサイズを変更できず、一般的に表示が粗くなる問題があります。一方、Tahoma は TrueType フォントであり、MS Sans Serif と外観が非常に似ています。また Tahoma はベクトル・フォントであるため、任意のサイズに変更したり、適切な角度にフォントを回転したりできます。
 - Unicode 文字が含まれないフォントを使用します。Oracle Reports では、UNIX プラットフォームで出力された Post Script に対して Unicode キャラクタ・セットがサポートされません。そのかわり、次のいずれかを使用できます。
 - Oracle Reports PDF 出力 (desformat=pdf)。これは、[第 5.5.1 項「マルチバイト・キャラクタ・セットの印刷」](#)に記載したとおり、マルチバイト・キャラクタ・セットをサポートします。
 - Oracle Reports のインストール時に出力され Oracle Applications で使用される PostScript 出力のフォント埋込み用 Oracle Reports ユーティリティである IX および PASTA。詳細は、[第 5.5.2 項「IX および PASTA の概要」](#)を参照してください。
3. PostScript の出力を設計プラットフォーム (Windows) と同様に配布プラットフォーム (UNIX) で表示するには、両方のプラットフォームで同じ用紙サイズを設定する必要があります。Windows でデフォルトの用紙サイズをレターから他のサイズ (A4 など) に変更するには、次の作業が必要です。
 - a. 「設定」→「コントロールパネル」→「プリンタ」を選択します。
 - b. デフォルトのプリンタを右クリックし、「プロパティ」を選択します。
 - c. 「全般」タブの「印刷設定」をクリックします。
 - d. 「用紙/品質」タブの「詳細設定」をクリックします。

- e. 用紙サイズを選択し、「OK」をクリックします。
- f. デフォルトの用紙サイズが設定されたメイン・ダイアログ・ボックスが表示されるまで、「OK」をクリックします。

7.6.2 レポートの配布

UNIX 上に使用可能な AFM フォント・ファイルがない場合、または文字の不揃いなどレポート出力にフォントの問題が発生した場合は、ttf2pt1 など、サード・パーティの無料ユーティリティを使用して Windows の TTF ファイルを変換し、AFM ファイルを生成できます。TFM ファイルには変換しないでください。予測したとおりの結果にならない場合があります。

レポートを UNIX プラットフォームに配布するには、次の手順を実行します。

1. レポートに使用されているフォントに対応する TTF ファイルを探します。これらの TTF ファイルを AFM に変換して、レポートに使用されているフォント用の AFM ファイルを作成します。

TTF ファイルから AFM ファイルへの変換には、TrueType から Type1 へのフォント変換ユーティリティを使用します。たとえば、ttf2pt1 を使用します。

2. 変換後、AFM ファイルの名前から拡張子 .afm を削除します。次に例を示します。

表 7-10 変換後のフォント・ファイル名

変換前	変換後	変更した名前
arial.ttf	arial.afm	Arial

3. レポートに使用した Windows の TTF ファイルを、UNIX マシンのフォント・ディレクトリにコピーします。たとえば、\$ORACLE_HOME/reports/fonts にコピーします。
4. REPORTS_PATH 環境変数に TTF ファイルへのパスを追加します。この例では、arial.ttf へのパスを追加します。
5. AFM ファイルを \$ORACLE_HOME/guicommon/tk/admin/AFM ディレクトリにコピーします。
6. TK_PRINTER 環境変数または PRINTER 環境変数に、デフォルトのプリンタの名前が設定されていることを確認します。この例では、printer1 になります。
7. uiprint.txt に次のような適切な PPD ファイルのエントリがあることを確認します。
printer name: PostScript:2:test:ppd file. この例では、次のようになります。
printer1:PostScript:2:test:hpljet42.ppd
8. 任意のテキスト・エディタを使用して、ファイル hpljet42.ppd を編集します。たとえば、DefaultPageSize、DefaultPageRegion および DefaultPaperDimension を次のように編集して、デフォルトの用紙をレターから A4 に変更します。

```
....
*DefaultPageSize: A4
....
*DefaultPageRegion: A4
....
*DefaultPaperDimension: A4
....
```

注意： hpljet42.ppd を編集する場合は、あらかじめそのファイルのバックアップを作成しておきます。このファイルは次の場所にあります。

```
$ORACLE_HOME/guicommon/tk/admin/PPD
```

9. 使用する PPD ファイルに、レポートに使用されているすべての AFM ファイルのエントリがあることを確認します。

Font Information セクションに移動して、新しい AFM ファイルに必要なエントリを次のような形式で追加します。

```
*FONTNAME:ENCODING:VERSION:LOCATION
```

次に例を示します。

```
*Font Arial: Standard "(Version 2.76)" Standard ROM
```

10. レポートをプリンタに対して実行し、A4 サイズで出力されることを確認します。

```
http://mywebserver.com:reports/rwservlet?server=myserver+report="c:¥test.rdf"+a  
uthid=hr/hr@mydb+desformat=postscript+destype=cache
```

7.6.3 よくある質問

この項では、レポートを PostScript 出力に配布する作業に関連したよくある質問 (FAQ) に回答します。

質問

Oracle Reports は、Unicode PostScript ファイルの生成をサポートしますか。

回答

現在、Oracle Reports は、Windows プラットフォームの場合のみ PostScript 出力の Unicode キャラクタ・セットをサポートします。UNIX プラットフォームでは、次のいずれかを使用できます。

- Oracle Reports PDF 出力 (desformat=pdf)。これは、[第 5.5.1 項「マルチバイト・キャラクタ・セットの印刷」](#)に記載したとおり、マルチバイト・キャラクタ・セットをサポートします。
- Oracle Reports のインストール時に出力され Oracle Applications で使用される PostScript 出力のフォント埋込み用 Oracle Reports ユーティリティである IX および PASTA。詳細は、[第 5.5.2 項「IX および PASTA の概要」](#)を参照してください。

質問

Oracle Reports では、PostScript 出力ファイルにフォントを埋め込むことができますか。

回答

Oracle Reports では、PostScript 出力ファイルにフォントを埋め込むことができません。フォント名と、AFM ファイルを使用して計算されたメトリックが書き込まれます。そのため、レポートでフォントの位置の問題が起こらないようにするには、必要なフォントをプリンタにインストールしておく必要があります。

質問

UNIX の異なるインストールでレポートを実行すると、レポートのページ数が異なります。インストールに関係なくレポートのページ数を同じにするには、どうすればよいでしょうか？

回答

UNIX では、Oracle Reports は各インストールのデフォルト・プリンタの PPD/HPD ファイルを使用してフォーマットを行います。解像度やフォント・リストは、この PPD/HPD ファイルから取得されます。Oracle Reports 10g (9.0.4) から、インストールにデフォルトのプリンタが設定されていない場合は、screenprinter.ppd が使用されるようになりました。この PPD ファイルによって画面がエミュレートされます。Oracle Reports の初期のバージョンでは、かわりに DISPLAY 環境変数が使用されていました。2つのインストール間で同じ AFM/TFM ファイルおよびフォント・ファイルが使用されていることを確認します。これで、PDF 出力のページ数は同じになるはずですが。

OracleAS Reports Services の宛先の構成

レポートを実行するには、レポートの出力方法（宛先）と受信者（配布）を検討する必要があります。配布については、[第 15 章「拡張配布の作成」](#)で説明しています。この章では、OracleAS Reports Services でデフォルトの宛先やカスタムの宛先に対し出力が処理される方法について説明します。まず出力処理の概要を示し、次に OracleAS Reports Services への宛先タイプの登録について説明します。

この章には、次の項があります。

- [出力処理の概要](#)
- [サーバーへの宛先タイプの登録](#)

8.1 出力処理の概要

レポートの出力は、実行時に指定する DESTYPE (宛先タイプ) の値によって制御されます。指定可能な宛先タイプは、destination 要素を使用して各自のサーバー構成ファイル (server_name.conf) に登録した宛先出力タイプによって決まります。詳細は、第 A.3.29 項「DESTYPE」および第 3.2.1.7 項「destination」を参照してください。

次のデフォルトの宛先は登録する必要がありません。

- キャッシュ
- 電子メール
- プリンタ
- ファイル

次のデフォルトの宛先は登録が必要になる場合があります。

- OracleAS Portal: この宛先のエントリーはデフォルトでサーバー構成ファイルに作成されますが、コメント・アウトされます。この宛先の使用を開始するには、destination エントリーのコメントを外し、適切なプロパティ値 (portaluserid プロパティの値など) を指定する必要があります。
- FTP および WebDAV: これらの宛先のエントリーはデフォルトでサーバー構成ファイルに作成され、コメント・アウトされません。登録されているので、デフォルトのまま使用できます。プロキシを必要とする FTP または WebDAV サーバーに出力を送信する必要がある場合は、デフォルトの場所 (ORACLE_HOME¥reports¥conf) にある proxyinfo.xml ファイルを編集し、destination 要素の proxy プロパティのコメントを外し、value に proxyinfo.xml ファイルへの完全なパスを指定します。次に例を示します。

```
<destination destype="ftp"
class="oracle.reports.plugin.destination.ftp.DesFTP"> <property
name="proxy" value="D:¥¥oracle¥¥reports¥¥conf¥¥proxyinfo.xml"/>
</destination>
```

OracleAS Reports Services Destinations API を使用して、FAX、Oracle の Internet File System (iFS) などのカスタム出力タイプや、任意の新しい宛先タイプを定義することもできます。この API を使用して、新しい宛先タイプを定義できます。また、レポートをカスタムの宛先に導くハンドラを作成することもできます。

注意: Oracle Reports で使用可能な API の詳細は、Reports Software Development Kit (RSDK) を参照してください。RSDK を入手するには、OTN (Oracle Technology Network) の Oracle Reports 10g ページ (<http://www.oracle.com/technology/products/reports/index.html>) で「SDK」をクリックします。

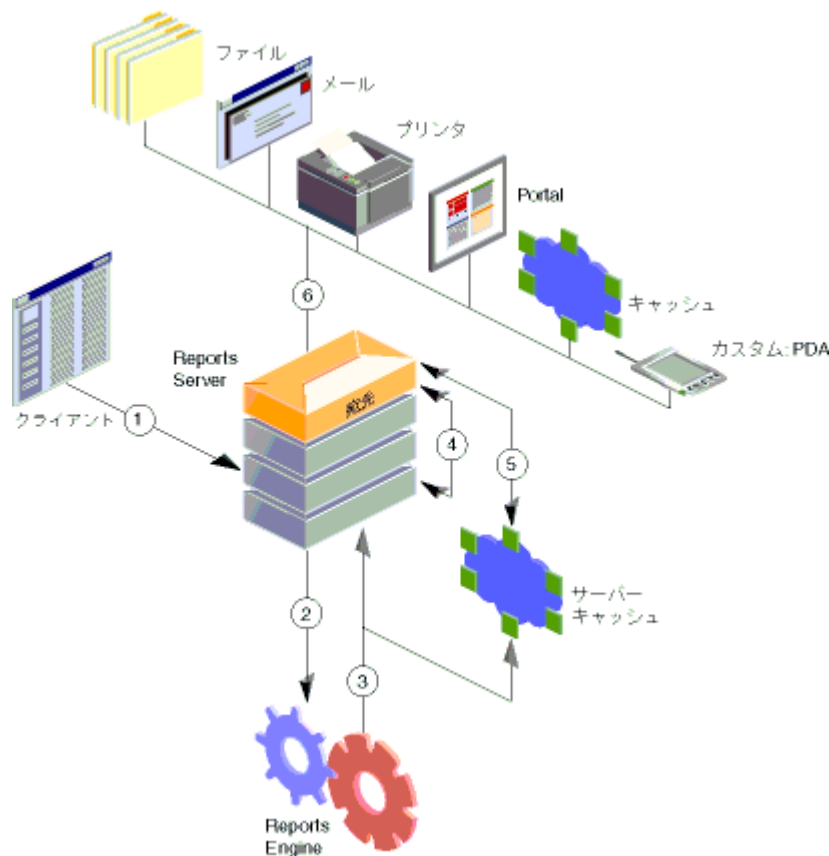
OracleAS Reports Services のアーキテクチャによって、出力の生成と配布の方法が標準化されます。このアーキテクチャでは、Reports Server を通じてレポート出力が適切な宛先に配信され、さらに Oracle Reports エンジンを通じて出力がその宛先とは関係なく生成されます。これによって、1 回のレポート実行が様々な用途で使用できるので、効率が大幅に向上します。また、出力処理アーキテクチャが拡大され、任意数の宛先タイプに対応できるようになっています。

以前は、Reports Runtime エンジンが出力の配信を全面的に処理していました。このため、出力先との通信方法を Runtime エンジンで認識する必要がありました。この結果、エンジンとサポートされている宛先は密接に結び付いていました。

OracleAS Reports Services は、この密接な結び付きとそれに伴う制限を解消します。Runtime エンジンが、すべての宛先を同等に処理するようになりました。そのため、作成される出力の宛先タイプを認識する必要はありません。サーバーが出力を宛先ハンドラに渡し、さらにハンドラがそれぞれに関連付けられた宛先タイプに配信するためのデータを準備します。事前定義の宛先タイプ（事前定義ハンドラを使用）を使用するか、またはサポートするカスタムの宛先タイプ用のハンドラを作成することができます。Oracle Reports では、ほとんどすべてのタイプの宛先に接続することができます。

図 8-1 は、OracleAS Reports Services の出力処理アーキテクチャの主要なコンポーネントを示しています。

図 8-1 宛先 / 配布アーキテクチャの主要コンポーネント



出力処理アーキテクチャでは、リクエストが次の順序で処理されます。

1. ユーザーが、クライアントまたはブラウザから Reports Server にリクエストを送信します。
2. サーバーがこのリクエストを Runtime エンジンに渡します。
3. Runtime エンジンが、宛先オブジェクト（特定の宛先のファイル・リストとそれらの宛先に関連するすべてのプロパティを含んだもの）およびレポート出力を作成 / 処理したうえで、宛先オブジェクトを Reports Server に、レポート出力をキャッシュにそれぞれ送信します。
4. Reports Server が、宛先オブジェクトを Reports Server の宛先コンポーネントに送信します。
5. Reports Server の宛先コンポーネントが、キャッシュからレポート出力をフェッチします。
6. Reports Server の宛先コンポーネントが、レポートと宛先オブジェクト（宛先デバイスによる出力の処理方法を指定するもの）を適切な宛先ハンドラに送信します。

8.2 サーバーへの宛先タイプの登録

OracleAS Reports Services で特定の宛先タイプにレポートを送信するには、そのタイプがデフォルト・タイプ（プリンタ、電子メール、キャッシュまたはファイル）、または事前にサーバーの構成ファイル（`server_name.conf`）に登録されたタイプであることが必要です。構成ファイルには、レポートに有効な宛先タイプを登録するための `destination` 要素が含まれています。任意数の宛先タイプを任意の場所で登録できます。

サーバーへの宛先タイプの登録には、次のタスクが関係します。

- [サーバー構成ファイルの destination セクションの設定](#)
- [宛先として有効な値の入力](#)

これらのタスクについては、次の各項で説明します。

8.2.1 サーバー構成ファイルの destination セクションの設定

`server_name.conf` ファイルの `destination` セクションを設定する手順は、次のとおりです。

1. サーバー構成ファイルを任意のテキスト・エディタでオープンします。

サーバー構成ファイルは、次のディレクトリにあります（Windows と UNIX で共通のパス）。

```
ORACLE_HOME\reports\conf\server_name.conf
```

2. 構成ファイルに `destination` セクションがない場合は、構成ファイルのデータ型定義ファイル（`rwserverconf.dtd`）セクションにおいて `destination` セクションの前にある要素の下に、`destination` セクションを作成します。

注意： サーバー構成ファイルは、関連する Document Type Definition (DTD) ファイル（`ORACLE_HOME\reports\dtd\rwserverconf.dtd`）に定義されている要素の順序に従います。このファイルで宛先の前にある要素がサーバー構成ファイルに存在する場合は、それらの要素の後に `destination` を配置します。

3. レポートを出力する際に使用する宛先タイプを登録するには、常に次の構文を使用します。

```
<destination destype="output_type_1" class="java_class_1">
<property name="valid_destype_property" value="valid_value"/>
<property name="valid_destype_property" value="valid_value"/>
</destination>
<destination destype="output_type_2" class="java_class_2">
<property name="valid_destype_property" value="valid_value"/>
</destination>
```

これらのタグの有効値は、次の項で説明します。

8.2.2 宛先として有効な値の入力

この項では、Oracle Reports でサポートされる宛先について説明します。

8.2.2.1 宛先タイプとクラス

`destype` 属性と `class` 属性は、デフォルト以外の出力タイプの有効な登録に対して必要となります。この2つの属性で、宛先タイプとそれらに関連する Java クラスを指定します。

OracleAS Reports Services に付属の事前定義（デフォルト）の宛先タイプおよびクラスを [表 8-1](#) に示します。

表 8-1 標準の宛先タイプとクラス

宛先	宛先タイプ	クラス
OracleAS Portal コンテンツ領域	oraclePortal	oracle.reports.server.DesOraclePortal
SMTP 対応電子メール	mail	oracle.reports.server.DesMail
ファイル	file	oracle.reports.server.DesFile
キャッシュ	cache	oracle.reports.server.DesCache
プリンタ	printer	oracle.reports.server.DesPrint
FTP	ftp	oracle.reports.plugin.destination.ftp.DesFTP
WebDAV	WebDAV	oracle.reports.plugin.destination.webdav.DesWebDAV

関連項目： 第 A.3.29 項「DESTYPE」には、oraclePortal 宛先タイプを使用してレポートをプッシュする例があります。

使用できる宛先タイプとクラスは、サーバーで提供する事前定義済のものにかぎられません。Destinations API を使用してカスタム・ハンドラを定義すれば、FAX などのカスタムの宛先タイプを登録できます。

注意： Oracle Reports で使用可能な API の詳細は、Reports Software Development Kit (RSDK) を参照してください。RSDK を入手するには、OTN (Oracle Technology Network) の Oracle Reports 10g ページ (<http://www.oracle.com/technology/products/reports/index.html>) で「SDK」をクリックします。

8.2.2.2 宛先プロパティの名前 / 値のペア

サーバー構成ファイルでは、登録された宛先に任意数のプロパティを関連付けることができます。宛先のプロパティは名前 / 値のペアで構成され、それぞれのペアが出力タイプの構成のある側面を定義します。これらは、宛先タイプで認識できる用語で表されます。たとえば、宛先タイプが oraclePortal である宛先は、次の名前 / 値のペアを認識します。

```
<property name="portaluserid" value="portal_id/portal_password@portal_schema"
confidential="yes" encrypted="no"/>
```

この例では、ポータルユーザー ID に関連付ける値を定義しており、confidential および encrypted という属性が指定されています。confidential="yes" は、この要素内で値の暗号化が必要であることを示し、encrypted="no" は値がまだ暗号化されていないことを示しています。次に Reports Server を起動したときは、値が自動的に暗号化され、encrypted が yes にリセットされます。

注意： サーバー構成ファイルに指定できる要素と属性は、rwserverconf.dtd ファイル (ORACLE_HOME¥reports¥dtd¥rwserverconf.dtd) で定義されている構文によって決まります。詳細は、第 3 章「OracleAS Reports Services の構成」を参照してください。

宛先タイプのプロパティに有効な値は、その宛先タイプに完全に依存します。これらの値は、Oracle Reports から取得されるのではなく、Reports Server で使用するわけでもありません。これらは宛先タイプ自体から取得され、宛先で認識する用語を使用します。開発者は、カスタムの宛先の要件を理解し、特定のカスタム出力タイプに関連付けるプロパティを把握する必要があります。

配布の検討に入ると、配布 XML ファイル内の `destype` 要素にもプロパティの名前 / 値のペアが指定できることに気付きます。サーバー構成ファイルの `destination` 要素に入力されるプロパティと、配布 XML ファイルの `destype` 要素に入力されるプロパティを区別することが重要です。

- サーバー構成ファイルの `destination` 要素に入力されるプロパティは、出力タイプの構成のみに関係します。たとえば、宛先 FAX の再試行許容回数を設定します。
- 配布 XML ファイルの `destype` 要素に入力されるプロパティは、ランタイム・パラメータの指定のみに関係します。たとえば、FAX の受信者の識別子を指定します。

8.2.3 宛先の例

次の例は、コンテンツを OracleAS Portal にプッシュするための `destination` 要素を示しています。

```
<destination destype="oraclePortal" class="oracle.reports.server.DesOraclePortal">  
  <property name="portaluserid" value="<the_username_password_tnsname_for_logon_to_portal">  
    encrypted="yes"/>  
</destination>
```

JDBC PDS の構成と使用

JDBC プラuggابل・データソース (PDS) を使用すると、次のような任意の JDBC データソースにアクセスできます。

- Oracle、DB2、Sybase、SQL Server などの RDBMS
- Microsoft Excel などの非リレーショナル・データソース
- JDBC-ODBC Bridge を経由する ODBC データソース

Oracle Reports にはデフォルトで JDBC PDS がインストールされており、あらゆる JDBC 対応データソースへのアクセスが可能です。

この章には、次の項があります。

- [JDBC 構成ファイル](#)
- [JDBC 問合せの定義と実行](#)
- [OracleAS Reports Services を使用した JDBC レポートの実行](#)
- [トラブルシューティング情報](#)
- [ユーザー独自の JDBC ドライバの追加](#)

9.1 JDBC 構成ファイル

Oracle Reports の JDBC PDS 構成ファイルは、`ORACLE_HOME\reports\conf` ディレクトリの `jdbcpds.conf` ファイルです。このファイルは、次のドライバ用に事前構成されています。

- プリインストール・ドライバ。すなわち Oracle JDBC Thin ドライバ、Oracle JDBC OCI (Thick) ドライバおよび JDBC-ODBC ドライバ。
- Oracle Technology Network (<http://www.oracle.com/technology/index.html>) から入手可能な DataDirect Merant ドライバ。

これ以外の JDBC ドライバを使用する場合は、`jdbcpds.conf` ファイル内で関連するエントリを追加または変更する必要があります。

Reports Builder では、`jdbcpds.conf` ファイルのエントリに基づいて、ドライバのリストを JDBC 問合せ接続ダイアログに表示します。このリストで、レポートの JDBC 問合せに使用するドライバを選択します。

Reports Builder では、起動時に `jdbcpds.conf` のエントリが読み取られ、キャッシュに格納されます。Reports Builder を再起動すると、`jdbcpds.conf` ファイルに対する変更（たとえば、新しい JDBC ドライバ・エントリの追加）の結果が表示されます。

`jdbcpds.conf` ファイルには、次の 2 つのセクションがあります。

- XML フォーマットおよびドライバ構成情報を記述する内部 DTD セクション

注意： このセクションは変更不可です。

- ドライバ名、接続文字列の書式、ドライバ・クラスなどのドライバ情報を詳述する XML セクション

注意： このセクションで、ドライバ情報を変更または追加することができます。

例

`jdbcpds.conf` ファイルの内容の例を次に示します。

```
<!-- DTD section - Not to be modified -->

<!DOCTYPE jdbcpds [
<!ELEMENT jdbcpds (driverInfo)>
<!ELEMENT driverInfo (driver+)>
<!ELEMENT driver (property*)>
<!ATTLIST driver name          CDATA #REQUIRED
                 sourceDatabase (oracle |
                                   sqlserver |
                                   sybase |
                                   db2 |
                                   informix |
                                   odbc |
                                   other) "oracle"
                 mainProtocol    ( jdbc ) "jdbc"
                 subProtocol     CDATA #REQUIRED
                 connectString   CDATA #REQUIRED
                 class           CDATA #REQUIRED
                 connection      CDATA #REQUIRED
                 loginTimeout    CDATA "5"
>
<!ELEMENT property EMPTY>
<!ATTLIST property name CDATA #REQUIRED
                  value CDATA #REQUIRED >
```



```
]>

<!-- Add or modify the following section for your driver information -->
<!-- Following drivers are available out-of-box in 9iAS -->

<jdbcps>
<driverInfo>
  <driver name = "oracleThin"
    sourceDatabase = "oracle"
    subProtocol = "oracle:thin"
    connectString = "mainProtocol:subProtocol:@databaseName"
    class= "oracle.jdbc.driver.OracleDriver"
    connection = "oracle.reports.plugin.datasource.jdbcps.
      JDBCConnectionHandling">
  </driver>

  <driver name = "oracle"
    sourceDatabase = "oracle"
    subProtocol = "oracle:oci8"
    connectString = "mainProtocol:subProtocol:@databaseName"
    class = "oracle.jdbc.driver.OracleDriver"
    connection = "oracle.reports.plugin.datasource.jdbcps.
      JDBCConnectionHandling">
  </driver>

  <driver name = "jdbc-odbc"
    sourceDatabase = "odbc"
    subProtocol = "odbc"
    connectString = "mainProtocol:subProtocol:databaseName"
    class = "sun.jdbc.odbc.JdbcOdbcDriver"
    connection = "oracle.reports.plugin.datasource.jdbcps.
      JDBCConnectionHandling">
  </driver>

  <driver name = "sqlserver-merant"
    sourceDatabase = "sqlserver"
    subProtocol = "merant:sqlserver"
    connectString = "mainProtocol:subProtocol://databaseName"
    class = "com.oracle.ias.jdbc.sqlserver.SQLServerDriver"
    connection = "oracle.reports.plugin.datasource.jdbcps.
      JDBCConnectionHandling">
  </driver>

  <driver name = "sybase-merant"
    sourceDatabase = "sybase"
    subProtocol = "merant:sybase"
    connectString = "mainProtocol:subProtocol://databaseName"
    class = "com.oracle.ias.jdbc.sybase.SybaseDriver"
    connection = "oracle.reports.plugin.datasource.jdbcps.
      JDBCConnectionHandling"
    loginTimeout = "0">
  </driver>

  <driver name = "db2-merant"
    sourceDatabase = "db2"
    subProtocol = "merant:db2"
    connectString = "mainProtocol:subProtocol://databaseName"
    class = "com.oracle.ias.jdbc.db2.DB2Driver"
    connection = "oracle.reports.plugin.datasource.jdbcps.
      JDBCConnectionHandling"
    loginTimeout = "0">
  </driver>
```

```

<driver name = "informix-merant"
      sourceDatabase = "informix"
      subProtocol = "merant:informix"
      connectString = "mainProtocol:subProtocol://databaseName"
      class = "com.oracle.ias.jdbc.informix.InformixDriver"
      connection = "oracle.reports.plugin.datasources.jdbcpds.
        JDBCConnectionHandling">
</driver>

</driverInfo>
</jdbcpds>

```

表 9-1 では、ドライバに関連付けることのできる各種の属性を説明しています。

表 9-1 ドライバの属性

属性名	説明	例
name	Oracle Reports で特定の JDBC ドライバを参照するための一意のユーザー定義値。	sybase-merant
sourceDatabase	ドライバが参照するデータベース。有効なエントリは、次のとおりです。 oracle sqlserver sybase db2 informix odbc other	oracle
subProtocol	データベース接続の作成前にデータベース URL とともに追加されるドライバ・サブ・プロトコル。ドライバ固有の情報です。この情報については、各ドライバのドキュメントを参照してください。たとえば、Merant ドライバへの接続に使用されるサブ・プロトコルは次のとおりです。 Sybase の場合は、merant:sybase SQL Server の場合は、merant:sqlserver	merant:sybase
connectString	ドライバの接続文字列の書式は mainProtocol:subProtocol://databaseURL です。たとえば、 jdbc:subProtocol://databaseName になります。subProtocol または databaseName には実際の値を指定せず、固定のプレースホルダ名を使用します。	mainProtocol:subProtocol://databaseName
class	REPORTS_CLASSPATH に登録し、ドライバをロードするためのドライバ・クラス名。ドライバ固有の情報です。この情報については、各ドライバのドキュメントを参照してください。	com.oracle.ias.jdbc.informix.InformixDriver

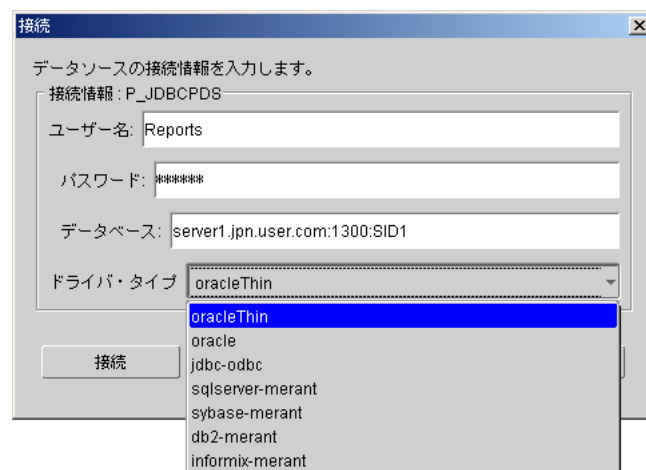
表 9-1 ドライバの属性 (続き)

属性名	説明	例
connection	<p>ドライバの接続処理クラス。JDBC PDS はドライバごとに異なる接続処理クラスを持つことができます。Oracle Reports のデフォルトの接続処理クラスは <code>oracle.reports.plugin.datasource.jdbcpds.JDBCCConnectionHandling</code> であり、ほとんどのドライバで使用できます。</p> <p>JDBC Connection クラスの拡張方法の詳細は、Oracle Reports の Java API リファレンスを参照してください。</p>	<code>oracle.reports.plugin.datasource.jdbcpds.JDBCCConnectionHandling</code>
loginTimeout (オプション)	ドライバ固有のパラメータ。秒単位で値を指定します。詳細は、ドライバのドキュメントを参照してください。	0
property	ドライバの追加プロパティを「属性名」と「値」で任意に指定します。	-

レポートの接続詳細を送信する際には、接続情報と `jdbcpds.conf` ファイルで指定されているドライバ構成情報とが組み合せられます。この組み合せた接続情報が、完全な接続 URL としてデータベースに送信されます。接続情報の例の詳細は、表 9-3、表 9-4、表 9-5、表 9-6 および表 9-7 を参照してください。

図 9-1 では、`jdbcpds.conf` ファイルで構成されている全ドライバの一覧を示しています。

図 9-1 Reports Builder の JDBC 接続ダイアログ・ボックス



9.1.1 プリインストールされているドライバ・エントリの確認

Oracle Reports には、SQL Server や Excel の JDBC-ODBC ドライバ、Oracle JDBC Thin ドライバ、Oracle JDBC OCI (Thick) ドライバなどがインストールおよび構成されています。次のドライバには、追加の JAR ファイルをインストールする必要がありません。

- Oracle JDBC Thin ドライバ
- Oracle JDBC OCI (Thick) ドライバ
- JDBC-ODBC ドライバ

JDBC-ODBC ドライバを使用すると、SQL Server や Excel のデータが扱えます。このドライバのエントリは、`jdbcpds.conf` ファイルに事前構成されています。SQL Server や Excel のデータを JDBC-ODBC ドライバで使用するには、ODBC データソースを作成する必要があります。ODBC データソースの作成方法の詳細は、Windows ヘルプを参照してください。

注意： Oracle Application Server では、SQL Server へのアクセスにも使用できる Merant DataDirect ドライバが用意されています。

9.1.2 Merant DataDirect ドライバのインストールと構成

オラクル社は、Merant DataDirect ドライバ・セットを提供しています。これは、OTN (Oracle Technology Network) (<http://www.oracle.com/technology/index.html>) からダウンロードできます。ドライバ構成ファイル (`jdbcpds.conf`) には、Merant DataDirect ドライバ関連のエントリがあります。さらに、JDBC 接続ダイアログ (図 9-1) には、オラクル社が提供する Merant DataDirect ドライバ・セットのエントリが一覧表示されます。

ただし、適切な JAR ファイルをインストールし、それを Reports Builder および OracleAS Reports Services から使用できるように、Oracle Reports 固有のクラスパス・エントリで指定する必要があります。

オラクル社が提供する、Oracle Application Server および Oracle Developer Suite で使用可能なドライバは次のとおりです。

- Sybase ドライバ
- DB2 ドライバ
- SQL Server ドライバ
- Informix ドライバ

カスタム・ドライバも、Oracle Application Server および Oracle Developer Suite で使用するためにインストールおよび構成できます。

次に、Merant DataDirect ドライバを構成する際の一般的な手順について説明します。個々の Merant DataDirect ドライバを構成する手順については、それぞれ該当する項を参照してください。

Merant DataDirect ドライバを構成する手順は次のとおりです。

1. 適切な JAR ファイルを Oracle Application Server と Oracle Developer Suite のディレクトリにインストールします。
2. `REPORTS_CLASSPATH` にエントリを追加して、Reports Builder と OracleAS Reports Services でそれらのファイルを使用できるようにします。

注意： この `REPORTS_CLASSPATH` 変数は、すべての UNIX プラットフォームで使用可能な `reports.sh` ファイルにあります。

必要な JAR ファイルについては、この項の該当するドライバの記述を参照してください。

- a. **Reports Builder:** REPORTS_CLASSPATH の既存エントリの前に、ドライバの場所を挿入します。この変数は、Windows の場合はレジストリ内に、UNIX の場合は reports.sh ファイル内にあります。この項の該当するドライバの例を参照してください。
- b. **rebuilder.conf:** rebuilder.conf 構成ファイル内にあるエンジンの classPath 属性に、ドライバの場所を追加します。この項の該当するドライバの例を参照してください。
- c. **Reports Server:** Reports Server 構成ファイル内にあるエンジンの classPath 属性に、ドライバの場所を追加します。この項の該当するドライバの例を参照してください。
- d. **jdbcpds.conf:** ORACLE_HOME¥reports¥conf ディレクトリ内にあります。パラメータの詳細は、表 9-1 を参照してください。この項の該当するドライバの例を参照してください。

9.1.2.1 Sybase ドライバ

1. 適切な JAR ファイルを Oracle Application Server と Oracle Developer Suite のディレクトリにインストールします。

必要な JAR ファイルは、YUtil.jar、YMsybase.jar、YMbase.jar です。

2. REPORTS_CLASSPATH にエントリを追加して、Reports Builder と OracleAS Reports Services でそれらのファイルを使用できるようにします。

注意： この REPORTS_CLASSPATH 変数は、すべての UNIX プラットフォームで使用可能な reports.sh ファイルにあります。

- a. **Reports Builder:** REPORTS_CLASSPATH の既存エントリの前に、ドライバの場所を挿入します。この変数は、Windows の場合はレジストリ内に、UNIX の場合は reports.sh ファイル内にあります。

例：

```
D:¥sybase_installed¥YUtil.jar;D:¥sybase_installed¥YMsybase.jar;D:¥sybase_installed¥YMbase.jar;existing classpath entries
```

- b. **rebuilder.conf:** rebuilder.conf 構成ファイル内にあるエンジンの classPath 属性に、ドライバの場所を追加します。

例：

```
<engine id="rwEng" class="oracle.reports.engine.EngineImpl" initEngine="1"
maxEngine="1" minEngine="0" engLife="50" maxIdle="30"
callbackTimeOut="60000" classPath="D:¥sybase_
installed¥YUtil.jar;D:¥sybase_installed¥YMsybase.jar;D:¥sybase_
installed¥YMbase.jar;">
...
</engine>
```

- c. **Reports Server:** Reports Server 構成ファイル内にあるエンジンの classPath 属性に、ドライバの場所を追加します。

例：

```
<engine id="rwEng" class="oracle.reports.engine.EngineImpl" initEngine="1"
maxEngine="1" minEngine="0" engLife="50" maxIdle="30"
callbackTimeOut="60000" classPath="D:¥sybase_
installed¥YUtil.jar;D:¥sybase_installed¥YMsybase.jar;D:¥sybase_
installed¥YMbase.jar;">
...
</engine>
```

- d. **jdbcpds.conf**: `ORACLE_HOME\reports\conf` ディレクトリ内にあります。必須パラメータの詳細は、表 9-1 を参照してください。

例：

```
<driver name = "sybase-merant"
        sourceDatabase = "sybase"
          subProtocol = "merant:sybase"
          connectString = "mainProtocol:subProtocol://databaseName"
          class = "com.oracle.ias.jdbc.sybase.SybaseDriver"
          connection = "oracle.reports.plugin.datasource.jdbcpds.
            JDBCConnectionHandling"
          loginTimeout = "0">

</driver>
```

9.1.2.2 DB2 ドライバ

- 適切な JAR ファイルを Oracle Application Server と Oracle Developer Suite のディレクトリにインストールします。

必要な JAR ファイルは、`YUtil.jar`、`YMdb2.jar`、`YMbase.jar` です。

- `REPORTS_CLASSPATH` にエントリを追加して、Reports Builder と OracleAS Reports Services でそれらのファイルを使用できるようにします。

注意： この `REPORTS_CLASSPATH` 変数は、すべての UNIX プラットフォームで使用可能な `reports.sh` ファイルにあります。

- Reports Builder:** `REPORTS_CLASSPATH` の既存エントリの前に、ドライバの場所を挿入します。この変数は、Windows の場合はレジストリ内に、UNIX の場合は `reports.sh` ファイル内にあります。

例：

```
D:\db2_installed\YUtil.jar;D:\db2_installed\YMdb2.jar;D:\db2_
installed\YMbase.jar;existing classpath entries
```

- rebuilder.conf:** `rebuilder.conf` 構成ファイル内にあるエンジンの `classPath` 属性に、ドライバの場所を追加します。

例：

```
<engine id="rwEng" class="oracle.reports.engine.EngineImpl" initEngine="1"
maxEngine="1" minEngine="0" engLife="50" maxIdle="30"
callbackTimeOut="60000" classPath="D:\db2_installed\YUtil.jar;D:\db2_
installed\YMdb2.jar;D:\db2_installed\YMbase.jar">
...
</engine>
```

- Reports Server:** Reports Server 構成ファイル内にあるエンジンの `classPath` 属性に、ドライバの場所を追加します。

例：

```
<engine id="rwEng" class="oracle.reports.engine.EngineImpl" initEngine="1"
maxEngine="1" minEngine="0" engLife="50" maxIdle="30"
callbackTimeOut="60000" classPath="D:\db2_installed\YUtil.jar;D:\db2_
installed\YMdb2.jar;D:\db2_installed\YMbase.jar">
...
</engine>
```

- d. **jdbcpds.conf**: `ORACLE_HOME\reports\conf` ディレクトリ内にあります。パラメータの詳細は、表 9-1 を参照してください。

例：

```
<driver name = "db2-merant"
      sourceDatabase = "db2"
      subProtocol = "merant:db2"
      connectString = "mainProtocol:subProtocol://databaseName"
      class = "com.oracle.ias.jdbc.db2.DB2Driver"
      connection = "oracle.reports.plugin.datasource.jdbcpds.
      JDBCConnectionHandling"
      loginTimeout = "0">

</driver>
```

9.1.2.3 SQL Server ドライバ

- 適切な .jar ファイルを Oracle Application Server と Oracle Developer Suite のディレクトリにインストールします。

必要な JAR ファイルは、YUtil.jar、YMsqserver.jar、YMbase.jar です。

- REPORTS_CLASSPATH にエントリを追加して、Reports Builder と OracleAS Reports Services でそれらのファイルを使用できるようにします。

注意： この REPORTS_CLASSPATH 変数は、すべての UNIX プラットフォームで使用可能な reports.sh ファイルにあります。

- Reports Builder:** REPORTS_CLASSPATH の既存エントリの前に、ドライバの場所を挿入します。この変数は、Windows の場合はレジストリ内に、UNIX の場合は reports.sh ファイル内にあります。

例：

```
D:\sqlserver_installed\YUtil.jar;D:\sqlserver_
installed\YMsqserver.jar;D:\sqlserver_installed\YMbase.jar;existing
classpath entries
```

- rebuilder.conf:** rebuilder.conf 構成ファイル内にあるエンジンの classPath 属性に、ドライバの場所を追加します。

例：

```
<engine id="rwEng" class="oracle.reports.engine.EngineImpl" initEngine="1"
maxEngine="1" minEngine="0" engLife="50" maxIdle="30"
callbackTimeOut="60000" classPath="D:\sqlserver_
installed\YUtil.jar;D:\sqlserver_installed\YMsqserver.jar;D:\sqlserver_
installed\YMbase.jar;">
...
</engine>
```

- Reports Server:** Reports Server 構成ファイル内にあるエンジンの classPath 属性に、ドライバの場所を追加します。

例：

```
<engine id="rwEng" class="oracle.reports.engine.EngineImpl" initEngine="1"
maxEngine="1" minEngine="0" engLife="50" maxIdle="30"
callbackTimeOut="60000" classPath="D:\sqlserver_
installed\YUtil.jar;D:\sqlserver_installed\YMsqserver.jar;D:\sqlserver_
installed\YMbase.jar;">
...
</engine>
```

- d. **jdbcpds.conf**: `ORACLE_HOME\reports\conf` ディレクトリ内にあります。パラメータの詳細は、表 9-1 を参照してください。

例：

```
<driver name = "sqlserver-merant"
      sourceDatabase = "sqlserver"
      subProtocol = "merant:sqlserver"
      connectionString = "mainProtocol:subProtocol://databaseName"
      class = "com.oracle.ias.jdbc.sqlserver.SQLServerDriver"
      connection = "oracle.reports.plugin.datasources.jdbcpds.
      JDBCCConnectionHandling">

</driver>
```

9.1.2.4 Informix ドライバ

- 適切な JAR ファイルを Oracle Application Server と Oracle Developer Suite のディレクトリにインストールします。

必要な JAR ファイルは、`YMutil.jar`、`YMinformix.jar`、`YMbase.jar` です。

- `REPORTS_CLASSPATH` にエントリを追加して、Reports Builder と OracleAS Reports Services でそれらのファイルを使用できるようにします。

注意： この `REPORTS_CLASSPATH` 変数は、すべての UNIX プラットフォームで使用可能な `reports.sh` ファイルにあります。

- Reports Builder:** `REPORTS_CLASSPATH` の既存エントリの前に、ドライバの場所を挿入します。この変数は、Windows の場合はレジストリ内に、UNIX の場合は `reports.sh` ファイル内にあります。

例：

```
D:\informix_installed\YMutil.jar;D:\informix_
installed\YMinformix.jar;D:\informix_installed\YMbase.jar;existing
classpath entries
```

- rebuilder.conf:** `rebuilder.conf` 構成ファイル内にあるエンジンの `classPath` 属性に、ドライバの場所を追加します。

例：

```
<engine id="rwEng" class="oracle.reports.engine.EngineImpl" initEngine="1"
maxEngine="1" minEngine="0" engLife="50" maxIdle="30"
callbackTimeOut="60000" classPath="D:\informix_
installed\YMutil.jar;D:\informix_installed\YMinformix.jar;D:\informix_
installed\YMbase.jar">
...
</engine>
```

- Reports Server:** Reports Server 構成ファイル内にあるエンジンの `classPath` 属性に、ドライバの場所を追加します。

例：

```
<engine id="rwEng" class="oracle.reports.engine.EngineImpl" initEngine="1"
maxEngine="1" minEngine="0" engLife="50" maxIdle="30"
callbackTimeOut="60000" classPath="D:\informix_
installed\YMutil.jar;D:\informix_installed\YMinformix.jar;D:\informix_
installed\YMbase.jar">
...
</engine>
```


- d. **jdbcpds.conf**: `ORACLE_HOME\reports\conf` ディレクトリ内にあります。パラメータの詳細は、表 9-1 を参照してください。

例:

```
<driver name = "informix-merant"
  sourceDatabase = "informix"
  subProtocol = "merant:informix"
  connectString = "mainProtocol:subProtocol://databaseName"
  class = "com.oracle.ias.jdbc.informix.InformixDriver"
  connection = "oracle.reports.plugin.datasource.jdbcpds.
  JDBCConnectionHandling">
</driver>
```

9.1.2.5 カスタム・ドライバ

オラクル社が提供していないドライバをインストールおよび構成する手順は次のとおりです。

1. 適切な JAR ファイルを Oracle Application Server と Oracle Developer Suite のディレクトリにインストールします。
2. `REPORTS_CLASSPATH` にエントリを追加して、Reports Builder と OracleAS Reports Services でそれらのファイルを使用できるようにします。

注意: この `REPORTS_CLASSPATH` 変数は、すべての UNIX プラットフォームで使用可能な `reports.sh` ファイルにあります。

必要な JAR ファイルについては、該当するドライバのドキュメントを参照してください。

- a. **Reports Builder:** `REPORTS_CLASSPATH` の既存エントリの前に、ドライバの場所を挿入します。この変数は、Windows の場合はレジストリ内に、UNIX の場合は `reports.sh` ファイル内にあります。

例:

```
driver location%1st jar file;driver location%2nd jar file2;existing classpath
entries
```

- b. **rebuilder.conf:** `rebuilder.conf` 構成ファイル内にあるエンジンの `classPath` 属性に、ドライバの場所を追加します。

例:

```
<engine id="rwEng" class="oracle.reports.engine.EngineImpl" initEngine="1"
maxEngine="1" minEngine="0" engLife="50" maxIdle="30"
callbackTimeOut="60000" classPath="driver location%1st jar file;driver
location%2nd jar file;">
...
</engine>
```

- c. **Reports Server:** Reports Server 構成ファイル内にあるエンジンの `classPath` 属性に、ドライバの場所を追加します。

例:

```
<engine id="rwEng" class="oracle.reports.engine.EngineImpl" initEngine="1"
maxEngine="1" minEngine="0" engLife="50" maxIdle="30"
callbackTimeOut="60000" classPath="driver location%1st jar file;driver
location%2nd jar file;">
...
</engine>
```

- d. `jdbcpds.conf`: `ORACLE_HOME\reports\conf` ディレクトリ内にあります。該当するドライバの構成情報を `jdbcpds.conf` ファイルに追加します。必須パラメータの詳細は、表 9-1 を参照してください。

例：

```
<driver name = "<driver name>"
  sourceDatabase = "<sourceDatabase>"
  subProtocol = "<subProtocol>"
  connectionString = "mainProtocol:subProtocol://databaseName"
  class = "<driver class name>"
  connection = "<connection handling class">
</driver>
```

注意： この値は、カスタム接続ダイアログを実装しない場合は、カスタム・ドライバに対して `connection = "oracle.reports.plugin.datasource.jdbcpds.JDBCConnectionHandling"` にすることができます。

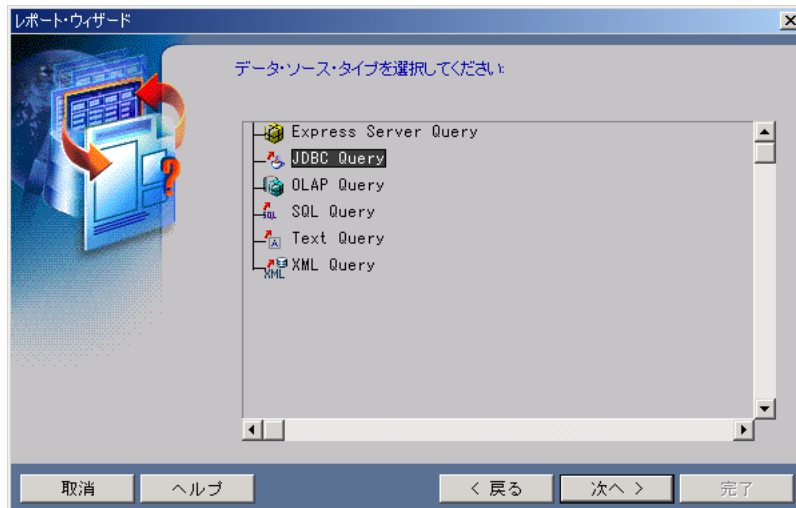
9.2 JDBC 問合せの定義と実行

関連する JDBC ドライバを構成した後、SQL またはストアド・プロシージャを使用して、JDBC 問合せを定義および実行できます。

JDBC 問合せを定義する手順は次のとおりです。

1. Reports Builder を起動します。
2. レポート・ウィザードを起動します。
3. データソース・タイプとして「**JDBC 問合せ**」を選択し、「次へ」をクリックします。レポート・ウィザードを使用する方法の詳細は、Oracle Reports のオンライン・ヘルプを参照してください。

図 9-2 データ・ソース・タイプの選択



4. 「データ・ソース定義」ウィンドウで、「**問合せ定義**」をクリックします。

5. 次のいずれかを定義します。

■ SQL 問合せ :

```
SELECT * FROM DEPARTMENT;
```

■ ストアド・プロシージャ :

使用するデータベースのストアド・プロシージャの完全なコール構文を入力します。次に例を示します。

```
TestProc (40)
```

コール構文の詳細は、データベースのドキュメントを参照してください。

JDBC PDS は、指定されたとおりにコール文をドライバに送信してストアド・プロシージャを起動します。

図 9-3 ストアド・プロシージャのコール

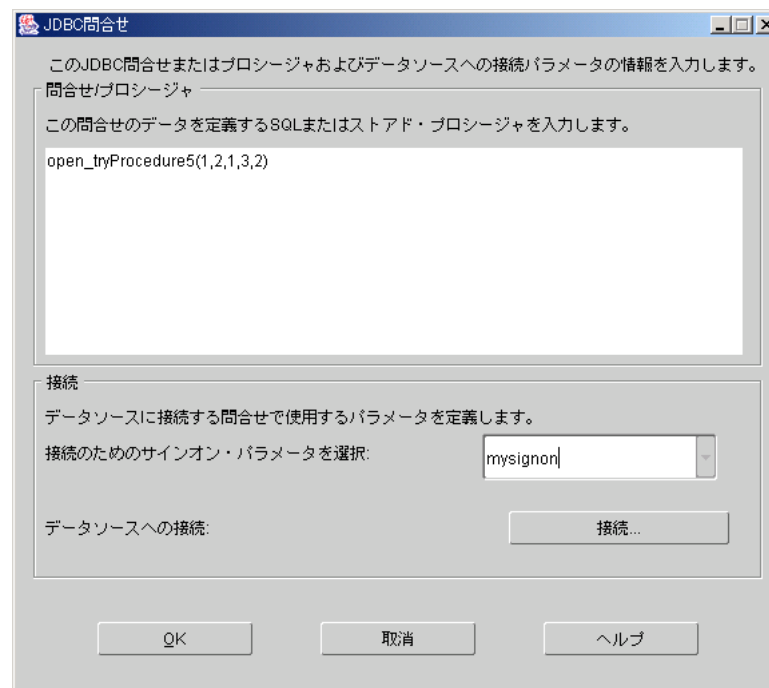
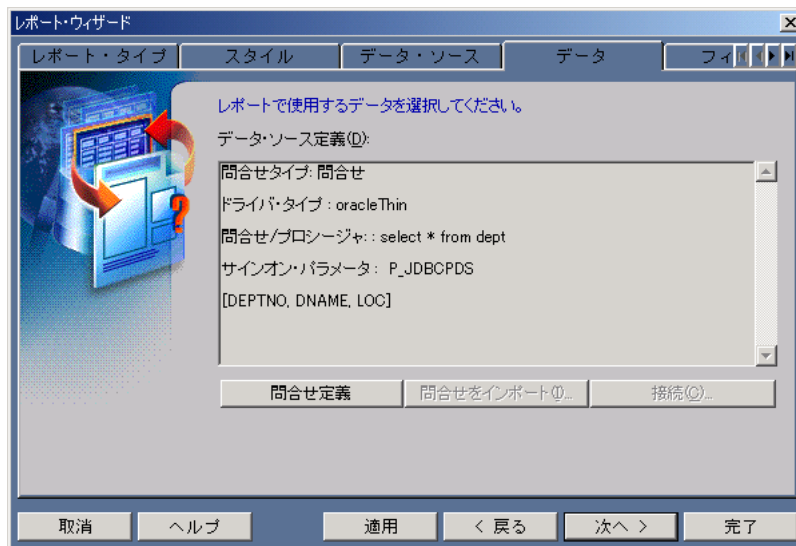


表 9-2 Excel データソースの指定

問合せ (単一ワークシート)	問合せ (複数ワークシート)
<pre>SELECT * FROM [SHEET1\$] または SELECT COL1, COL2, ...COLn FROM [SHEET1\$]</pre> <p>SHEET1\$ は .xls ファイルの名前です。</p> <p>ワークシートの最初の行の値は、問合せの列名として取得されます。</p> <p>注意:最初の行のどの列にも値が指定されていない場合、デフォルトの名前は FcolumnNumber になります。たとえば、8 列目は F8、9 列目は F9 というようになります。</p>	<pre>SELECT * FROM [WORKSHEETNAME\$]</pre> <p>[WORKSHEETNAME\$] は、ワークシートの名前です。</p> <p>ワークシートの最初の行は、問合せの列名として取得されます。</p> <p>注意:最初の行のどの列にも値が指定されていない場合、デフォルトの名前は FcolumnNumber になります。たとえば、8 列目は F8、9 列目は F9 というようになります。</p>

6. サインオン・パラメータ名を指定します。このサインオン・パラメータは、データベースに対して実行されるときに接続情報に関連付けられます。デフォルトのサインオン・パラメータ名は P_JDBCPDS です（詳細は、第 A.3.69 項「P_JDBCPDS」を参照してください）。
 - a. 新しいサインオン名を入力し、「**接続**」をクリックします。OracleAS Reports Services を使用してレポートを実行する際には、このサインオン・パラメータを使用して、データベース接続を指定します。
 - b. そのドライバ・タイプの接続情報（ユーザー名、パスワード、データベース名）を入力します。接続情報の例については、表 9-3、表 9-4、表 9-5、表 9-6 および表 9-7 を参照してください。
 - c. ドライバ・タイプを選択します。jdbcpds.conf ファイルに入力された値に基づいて、ドライバ・リストが表示されます。
 - d. 「**接続**」をクリックすると、新しいサインオンを使用してデータベースにアクセスできます。内部的に作成された接続文字列は、次の 2 つの組合せです。
 - * jdbcpds.conf ファイルで定義されている connectString ドライバ属性（表 9-1）
 - * connectString のデータベース名部分の入力に使用される「**接続**」ダイアログの接続情報
7. 「**OK**」をクリックして、JDBC 問合せを実行します。
8. レポート・ウィザードには、問合せの説明が表示されます（図 9-4）。

図 9-4 問合せの説明



9. ウィザードの手順に従ってレイアウトを定義し、JDBC 問合せに基づいてレポートを実行します。

9.2.1 接続情報の例

表 9-3、表 9-4、表 9-5、表 9-6、表 9-7、表 9-8 および表 9-9 では、次で使用する接続情報の例を示しています。

- プリインストール・ドライバ。すなわち Oracle JDBC Thin ドライバ、Oracle JDBC OCI (Thick) ドライバおよび JDBC-ODBC ドライバ。
- Oracle Technology Network (<http://www.oracle.com/technology/index.html>) から入手可能な DataDirect Merant ドライバ。

表 9-3 Oracle Thin ドライバ

プロパティ	値
ユーザー名	Reports
パスワード	Welcome
データベース	hostname: 接続先サーバーの TCP/IP アドレスまたは TCP/IP ホスト名。 port: TCP/IP ポート番号。 property: 接続プロパティ。接続プロパティと有効な値のリストについては、ドライバのドキュメントを参照してください。 例: server1.us.oracle.com:1300:session1

表 9-4 Oracle Thick ドライバ

プロパティ	値
ユーザー名	Reports
パスワード	Welcome
データベース	n123 n123 は、tnsnames.ora ファイル内の tnsname エントリです。

表 9-5 JDBC-ODBC ドライバ

プロパティ	値
ユーザー名	該当なし
パスワード	このパスワードは、ODBC 接続を確立する時点で設定されます。
データベース	SQLSVR SQLSVR は、ODBC データソースの ODBC データ・エントリです。

表 9-6 Sybase

プロパティ	値
ユーザー名	Reports
パスワード	Welcome
データベース	hostname: 接続先サーバーの TCP/IP アドレスまたは TCP/IP ホスト名。 port: TCP/IP ポート番号。 例: server1.us.oracle.com:1300

表 9-7 DB2

プロパティ	値
ユーザー名	Reports
パスワード	Welcome
データベース	hostname: 接続先サーバーの TCP/IP アドレスまたは TCP/IP ホスト名。 port: TCP/IP ポート番号。 property: 接続プロパティ。接続プロパティと有効な値のリストについては、ドライバのドキュメントを参照してください。 例 1: server1:1654 例 2: server2:1721;PackageName=pkg1

表 9-8 SQL Server

プロパティ	値
ユーザー名	Reports
パスワード	Welcome
データベース	hostname: 接続先サーバーの TCP/IP アドレスまたは TCP/IP ホスト名。 port: TCP/IP ポート番号。 例 1: server1:1654

表 9-9 Informix

プロパティ	値
ユーザー名	Reports
パスワード	Welcome
データベース	hostname: 接続先サーバーの TCP/IP アドレスまたは TCP/IP ホスト名。 port: TCP/IP ポート番号。 InformixServer name: Informix サーバー名。 Database name: 接続先のデータベース名。 例 2: server_name:2003;InformixServer=myinformix_server;DatabaseName=scott/tiger@mydb

9.3 OracleAS Reports Services を使用した JDBC レポートの実行

JDBC 問合せ (Reports Server または rwrun エンジンで) を含むレポートを実行するには、サインオン・パラメータを使用して、JDBC データソースの接続情報を送信します。このサインオン・パラメータは、Reports Builder での設計時に JDBC 問合せに対して定義されます。

たとえば、Sybase データソースに対する JDBC 問合せ、DB2 データソースに対する JDBC 問合せ、および Oracle データソースに対する SQL 問合せがレポートに含まれる場合、これらの要求を次のように定義できます。

```
http://your_ias_server:port/reports/rwservlet?report=my.rdf&userid=user/pwd@oracledb
&desformat=pdf&destype=cache&p_sybasepds=sybaseuser/pw@sybasehost:port
&p_db2pds=db2user/pwd@db2host:port
```

ここでは

- `userid` は、SQL 問合せを実行する際、Oracle データベースに接続するための値です。レポートに SQL 問合せも REF カーソル問合せもない場合は、`userid` の指定は不要です。
- `p_sybasepds` は、Sybase への JDBC 問合せに関連付けられたサインオン・パラメータです。
- `p_db2pds` は、設計時にレポート内で定義された、DB2 への JDBC 問合せに関連付けられたサインオン・パラメータです。

Reports Builder でのレポートの設計時に「JDBC 問合せ」ダイアログ・ボックスで名前を指定しなかった場合は、デフォルトのサインオン・パラメータ名 `P_JDBCPS` が使用されます。

9.4 トラブルシューティング情報

この項では、次の項目の一覧を示します。

- JDBC PDS エラー・メッセージ (エラー・メッセージ)
- JDBC 問合せのトラブルシューティング (トレース情報)

9.4.1 エラー・メッセージ

表 9-10、表 9-11 および表 9-12 は、JDBC PDS に関連するトラブルシューティング情報の一覧です。

表 9-10 データベース接続に関連するエラー・メッセージ

エラー・メッセージ	原因	処置
接続クラス {0} はロードできません	選択したドライバに対して、 <code>jdbcpds.conf</code> ファイルで指定されている接続クラスが無効です。	<code>jdbcpds.conf</code> ファイルで指定されているドライバ接続クラスが有効かつ使用可能であることを確認します。
データソースへの接続に失敗しました	接続情報が無効です。	ユーザー名、パスワード、データベースおよびドライバ・タイプの妥当性を確認します。
無効なサインオン・パラメータ {0}	指定された問合せまたはプロシージャのサインオン・パラメータが無効です。	レポートの JDBC 問合せタイプに対してサインオン・パラメータが使用可能かつ有効であることを確認します。
サインオン・パラメータ {0} に無効な値が与えられました	指定されたサインオン・パラメータに対する接続文字列が無効です。	このサインオン・パラメータに対して指定された接続文字列が、選択したドライバに対して有効であることを確認します。

表 9-11 データソースの実行に関連するエラー・メッセージ

エラー・メッセージ	原因	処置
日付タイプの参照パラメータは、使用している JDBC ドライバでサポートされていません	データベースへの接続に使用しているドライバが、日付データ・タイプを参照パラメータとしてサポートしていません。	次のいずれかを使用します。 文字列データ・タイプ (参照パラメータとして)。 日付データ・タイプをサポートする別の JDBC ドライバ (参照パラメータとして)。
無効な文字パラメータ {0} が問合せに使用されています	問合せまたはプロシージャで使用した文字パラメータが無効です。	問合せやプロシージャで使用している文字パラメータが有効であることを確認します。有効なパラメータがない場合は、新しいパラメータを作成します。

表 9-11 データソースの実行に関連するエラー・メッセージ (続き)

エラー・メッセージ	原因	処置
SQL エラー :	指定した問合せまたはプロシージャで SQL 構文エラーが発生しました。	問合せまたはプロシージャの構文が有効であることを確認します。該当するデータソースのドキュメントを参照してください。
指定されたデータソースに対する無効な問合せ / プロシージャ	問合せまたはプロシージャの構文が無効です。	問合せまたはプロシージャの構文が有効であることを確認します。該当するデータソースのドキュメントを参照してください。
無効な参照パラメータ値	参照パラメータ値が無効です。	参照列のタイプと値が正しいかどうかを確認します。
入力された問合せ / プロシージャはありません	問合せまたはプロシージャのテキスト・フィールドが空です。	有効な問合せまたはプロシージャをテキスト・フィールドに入力します。
データベース URL:	データベース URL が無効です。	指定したデータベース名と選択したドライバ・タイプが有効であることを確認します。
列の数または列のタイプが、問合せ定義と一致していません	フェッチされたデータが、問合せ定義で指定されている列の数または列のタイプと一致していません。	列の数および列のタイプが問合せ定義と一致していることを確認します。
問合せ / プロシージャで使用されている列タイプ {0} は Reports の JDBC 問合せではサポートされていません	この列タイプは、Oracle Reports の JDBC 問合せインタフェースでサポートされていません。	Oracle Reports の JDBC 問合せインタフェースでサポートされている列タイプだけを使用していることを確認します。JDBC 仕様および Oracle Reports のドキュメントで、サポートされているタイプの一覧を参照してください。

表 9-12 ドライバ/PDS の問題の識別

エラー・メッセージ	原因	処置
jdbcpds.conf 構成ファイルのインライン DTD セクションは変更されました	jdbcpds.conf ファイルのインライン DTD セクションのフォーマットが変更されています。	DTD フォーマットを変更した場合は、JDBC PDS 要件に対して構成ファイルが有効であることを確認します。
行番号 :	jdbcpds.conf ファイルの指定された行でエラーが検出されました。	指定された行のエラーを修正します。
構成ファイル jdbcpds.conf が見つかりません	reports/conf ディレクトリで jdbcpds.conf ファイルが見つかりません。	reports/conf ディレクトリに jdbcpds.conf ファイルがあることを確認します。
構成ファイル jdbcpds.conf の解析エラー。エラー数 : {0}	jdbcpds.conf ファイルの XML セクションがそのインライン DTD に準拠していません。	jdbcpds.conf ファイルの XML セクションが適切なインライン DTD を参照していることを確認します。
jdbcpds.conf ファイル内のドライバ {0} には現在エンタリがありません	問合せで使用しているドライバが jdbcpds.conf ファイルで指定されていません。	必要なドライバのエントリと関連するドライバ情報が、ともに jdbcpds.conf ファイルにあることを確認します。

9.4.2 トレース情報

JDBC 問合せをデバッグするには、Oracle Reports で生成される次の詳細トレース情報 (ORACLE_HOME¥reports¥logs¥) を参照します。

- 設計時 (JDBC 問合せの作成) および実行時 (JDBC 問合せの実行) のトレース生成されたトレース情報は、次のものの検出に役立ちます。
 - 文字パラメータやバインド・パラメータ
 - ドライバへの接続用に作成された最終的な接続文字列
 - ドライバから受け取ったメタデータ情報
 - データベースに送信した最終的な問合せ

例 9-1 は、設計時のトレース出力の例です。

例 9-2 は、実行時のトレース出力の例です。

トレース出力の例

例 9-1 「JDBC 問合せ」ダイアログによる JDBC 問合せの作成

Connection handling trace showing final connect string

```
[2003/4/7 5:41:38:686] Debug 50103 (jdbcpds): handleConnectButtonEvent : start
[2003/4/7 5:41:38:686] Debug 50103 (jdbcpds): handleConnectButtonEvent : subProtocol
:sybase-merant
[2003/4/7 5:41:38:686] Debug 50103 (jdbcpds): handleConnectButtonEvent :
connection class :oracle.reports.plugin.datasources.jdbcpds.JDBCConnectionHandling
[2003/4/7 5:41:38:696] Debug 50103 (jdbcpds): handleConnectButtonEvent : combine
string :jdbc:merant:sybase://server1.us.oracle.com:1300
[2003/4/7 5:41:38:696] Debug 50103 (jdbcpds): JDBCDataSource : setJDBCQueryType: sybase
[2003/4/7 5:41:41:350] Debug 50103 (jdbcpds): JDBCUIEventHandler : handleConnectEvent :
Valid Connection
com.oracle.ias.jdbc.sybase.SybaseConnection@56fc16
[2003/4/7 5:41:41:350] Debug 50103 (jdbcpds): JDBCUIEventHandler :
handleConnectEvent : END com.oracle.ias.jdbc.sybase.SybaseConnection@56fc16
```

Design time metadata of query

```
[2003/3/31 6:35:46:363] Debug 50103 (jdbcpds): JDBCUIEventHandler : handleOKEvent :
Serialize XML<jdbc pds DTDVersion="
1.0"><JDBCQuery>jdbc pds pkg.proc_with_
param(1,2,3,4,5)</JDBCQuery><QueryDefinition>1</QueryDefinition><driverType>oracle
</driverType><connectionClass>oracle.reports.plugin.datasources.jdbcpds.JDBCConnect
ionHandling</connectionClass><SignOnParameter>P_
JDBCPDS</SignOnParameter><jdbcElements><elementname = "EMPNO" type = "2"
typeName = "NUMBER" columnSize = "4" columnScale = "0" /><element name = "ENAME"
type = "12" typeName = "VARCHAR2" columnSize = "10" columnScale = "0"
/><element name = "JOB" type = "12" typeName = "VARCHAR2" columnSize = "9"
columnScale = "0" /><element name = "MGR" type = "2" typeName = "NUMBER"
columnSize = "4" columnScale = "0" /><element name = "HIREDATE" type = "93"
typeName = "DATE" columnSize = "16" columnScale = "0" /><element name = "SAL"
type = "2" typeName = "NUMBER" columnSize = "7" columnScale = "2" /><element
name = "COMM" type = "2" typeName = "NUMBER" columnSize = "7" columnScale =
"2" /><element name = "DEPTNO" type = "2" typeName = "NUMBER" columnSize = "2"
columnScale = "0" /></jdbcElements><referenceColumns></referenceColumns></jdbc pds>
[2003/3/31 6:35:46:383] Debug 50103 (jdbcpds): JDBCUIEventHandler :handleOKEvent
END
```

例 9-2 JDBC 問合せの実行

```
[2003/3/18 5:45:17:707] Debug 50103 (jdbcpds): JDBCDataSource : startRuntime
method : START
```

Describing the JDBC Query:

```
[2003/3/18 5:45:17:707] Debug 50103 (jdbcpds): JDBCDataSource : describe : START
[2003/3/18 5:45:17:707] Debug 50103 (jdbcpds): applyXML: Extract the Serilzed XML
containing Query Meta Data <jdbcps DTDVersion=" 1.0"><JDBCQuery>select * from
emp</JDBCQuery><QueryDefinition>0</QueryDefinition><driverType>oracle</driverType>
<connectionClass>oracle.reports.plugin.datasources.jdbcps.JDBCConnectionHandling</
connectionClass>...
```

ConnectionHandling At Runtime:

```
[2003/3/18 5:45:17:737] Debug 50103 (jdbcpds): JDBCDataSource : startRuntime :
Create a new connection and handle it
[2003/3/18 5:45:17:737] Debug 50103 (jdbcpds): JDBCExecuteQuerySource :
handleConnection : START
[2003/3/18 5:45:17:778] Debug 50103 (jdbcpds): JDBCExecuteQuerySource :
handleConnection : set driver
[2003/3/18 5:45:17:778] Debug 50103 (jdbcpds): JDBCExecuteQuerySource :
handleConnection : Check if Connection for the sign on parameter is pooled
[2003/3/18 5:45:17:778] Debug 50103 (jdbcpds): JDBCExecuteQuerySource
:handleConnection : connection available in pool
[2003/3/18 5:45:17:778] Debug 50103 (jdbcpds): handleConnection : END
[2003/3/18 5:45:17:778] Debug 50103 (jdbcpds): JDBCDataSource : startRuntime : END
```

Runtime execution of jdbc query

```
[2003/3/31 6:36:2:836] Debug 50103 (jdbcpds): JDBCDataSource : execute : run Query
[2003/3/31 6:36:2:836] Debug 50103 (jdbcpds): JDBCExecuteQuerySource :
getOutputFromDatabase : START
[2003/3/31 6:36:2:836] Debug 50103 (jdbcpds): JDBCExecuteQuerySource :
getOutputFromDatabase: start Query stringto be submitted
jdbcpspkg.proc_with_param(1,2,3,4,5)
[2003/3/31 6:36:2:836] Debug 50103 (jdbcpds): JDBCExecuteQuerySource :
getOutputFromDatabase : check connection
[2003/3/31 6:36:2:836] Debug 50103 (jdbcpds): JDBCExecuteQuerySource :
getOutputFromDatabase : QSource Id: 1
[2003/3/31 6:36:2:836] Debug 50103 (jdbcpds): JDBCExecuteQuerySource:
executeOracleProcedure:Start
[2003/3/31 6:36:2:836] Debug 50103 (jdbcpds): JDBCExecuteQuerySource:
executeOracleProcedure:Procedure to be submitted { call
jdbcpspkg.proc_with_param(?,?,?,?) }
[2003/3/31 6:36:2:836] Debug 50103 (jdbcpds): JDBCExecuteQuerySource:
executeOracleProcedure: Set parameters for the procedure call
[2003/3/31 6:36:2:836] Debug 50103 (jdbcpds): JDBCExecuteQuerySource:
executeOracleProcedure: execute procedure
[2003/3/31 6:36:2:847] Debug 50103 (jdbcpds): JDBCDataSource : execute : query
execution over andresulset object is oracle.jdbc.driver.OracleResultSetImpl@751a9e
[2003/3/31 6:36:2:847] Debug 50103 (jdbcpds): JDBCDataSource : execute : END
```

Running Report trace with Result set info

```
2003/4/7 5:26:6:996] Debug 50103 (jdbcpds): JDBCDataSource : execute : replace
lexical columns withactual string for the query
[2003/4/7 5:26:6:996] Debug 50103 (jdbcpds): JDBCDataSource : execute : run Query
[2003/4/7 5:26:6:996] Debug 50103 (jdbcpds): JDBCExecuteQuerySource :
getOutputFromDatabase : START
[2003/4/7 5:26:6:996] Debug 50103 (jdbcpds): JDBCExecuteQuerySource :
getOutputFromDatabase: start Query stringto be submitted select * from reports
[2003/4/7 5:26:7:6] Debug 50103 (jdbcpds): JDBCExecuteQuerySource :
getOutputFromDatabase : check connection
[2003/4/7 5:26:7:6] Debug 50103 (jdbcpds): JDBCExecuteQuerySource :
getOutputFromDatabase : QSource Id: 4
```

```

[2003/4/7 5:26:7:6] Debug 50103 (jdbcpds): JDBCExecuteQuerySource :
getOutputFromDatabase : Query source is SQL query
[2003/4/7 5:26:7:6] Debug 50103 (jdbcpds): JDBCExecuteQuerySource:executeQuery
Start
[2003/4/7 5:26:7:6] Debug 50103 (jdbcpds): executeQuery prepareStatement select *
from reports
[2003/4/7 5:26:7:6] Debug 50103 (jdbcpds): executeQuery : bind parameters set for
the query
[2003/4/7 5:26:7:6] Debug 50103 (jdbcpds): executeQuery : JDBC Query executed
[2003/4/7 5:26:7:387] Debug 50103 (jdbcpds): JDBCExecuteQuerySource :
getOutputFromDatabase : Query result col 0 test col 1 10
[2003/4/7 5:26:7:387] Debug 50103 (jdbcpds): JDBCExecuteQuerySource:executeQuery
Start
[2003/4/7 5:26:7:387] Debug 50103 (jdbcpds): executeQuery prepareStatement select
* from reports
[2003/4/7 5:26:7:387] Debug 50103 (jdbcpds): executeQuery : bind parameters set
for the query
[2003/4/7 5:26:7:387] Debug 50103 (jdbcpds): executeQuery : JDBC Query executed
[2003/4/7 5:26:7:767] Debug 50103 (jdbcpds): JDBCDataSource : execute : query
execution over andresultset object is com.oracle.ias.jdbc.base.BaseResultSet@56c3cf
[2003/4/7 5:26:7:767] Debug 50103 (jdbcpds): JDBCDataSource : execute : END

```

9.5 ユーザー独自の JDBC ドライバの追加

注意： Oracle Reports では、PDS API を公開し、またチュートリアルも用意してユーザー独自の PDS の実装方法やカスタマイズ方法の詳細を説明しています。詳細は、Reports Software Development Kit (RSDK) を参照してください。RSDK を入手するには、OTN (Oracle Technology Network) の Oracle Reports 10g ページ (<http://www.oracle.com/technology/products/reports/index.html>) で「SDK」をクリックします。この API を使用すると、PDS を無制限に実装して、ユーザーの持つあらゆる種類のデータソースにアクセスすることが可能になります。

JDBC PDS を追加するには、次の作業が必要です。

- [jdbcpds.conf ファイルの構成](#)
- [ドライバの JAR ファイルのインストール](#)

9.5.1 jdbcpds.conf ファイルの構成

jdbcpds.conf ファイルの構成方法の詳細は、[第 9.1 項「JDBC 構成ファイル」](#)を参照してください。

9.5.2 ドライバの JAR ファイルのインストール

ドライバの JAR ファイルのインストール方法の詳細は、[第 9.1.2.5 項「カスタム・ドライバ」](#)を参照してください。

OracleAS Reports Services の セキュリティ保護

インターネットはあまりにオープンであるために、企業のどの機密情報に誰がアクセスできるかを制御することが課題となります。OracleAS Reports Services では、数多くのセキュリティ・オプションを用意しています。これを利用して、重要なデータに適切なユーザーが安全にアクセスできるようになります。この章では、これらのセキュリティ・オプションについて概説します。

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

10.1 OracleAS Reports Services のセキュリティについて

この項では、レポートやレポートに含まれるデータへ安全にアクセスするための OracleAS Reports Services のセキュリティ動作について説明します。

- [リソースの保護](#)
- [認証とアクセス権の適用](#)
- [Oracle Identity Management インフラストラクチャの利用](#)

10.1.1 リソースの保護

OracleAS Reports Services では、主に次の 3 つの領域のセキュリティを包括的に実現します。

- アプリケーション・セキュリティ（ユーザーがレポート・リクエストを送信した場合にレポート・アプリケーションへのアクセスを制御する）
- リソース・セキュリティ（レポート、プリンタ、カレンダー、Reports Server へのアクセスを制御する）
- データソース・セキュリティ（特定のデータベースへのアクセスを制御する）

10.1.1.1 アプリケーション・セキュリティ

ユーザーがレポートにアクセスして実行するには、通常、アプリケーションやサイトにログインする必要があります。そのような起動アプリケーションは、なんらかのログイン機能（たとえば、OracleAS Single Sign-On）で保護されるのが普通です。起動アプリケーションへのログインに成功すると、リソース・セキュリティが後を引き継ぎ、そのユーザーまたはグループが要求できるレポートおよび宛先を決定します。

OracleAS Single Sign-On では、アプリケーションのセキュリティを確保するために一箇所からのユーザー・ログインと、必要に応じてデータソース・セキュリティを提供します。通常の構成では、OracleAS Single Sign-On を介してログインして、自分のレポートにアクセスして実行するレポート・アプリケーションにアクセスします。

Oracle Internet Directory には、OracleAS Single Sign-On で使用される、ユーザーおよびグループの権限情報が格納されます。また、Oracle Internet Directory には、ユーザーごとのデータソース・セキュリティ情報も格納されます。Oracle Delegated Administration Services では、Oracle Internet Directory に格納されている情報が編集されます。Oracle Delegated Administration Services は OracleAS Portal の内部からアクセスできます。または、OracleAS Portal に関係なく、スタンドアロン・コンポーネントとしてアクセスすることもできます。

あるいは、ユーザー独自のアプリケーションから、独自のログイン・メカニズムおよびユーザー / グループ・リポジトリを使用してレポートを実行することも可能です。この場合、OracleAS Reports Services には、これら Oracle 以外のコンポーネントとの統合を実現するインタフェースが用意されています。

関連項目： [第 10.2 項「OracleAS Reports Services セキュリティの構成」](#)
には、インタフェースの詳細に関する記述があります。

10.1.1.2 リソース・セキュリティ

リソース・セキュリティにより、権限のあるユーザーまたはグループだけに特定のレポートの実行を許可することができます。また、レポートの実行時にユーザーまたはグループが特定のプリンタや Reports Server にアクセスできないようにすることができます。たとえば、特定のプリンタやサーバーが、あるグループのユーザーのために確保されている状況が考えられます。あるいは、一部のプリンタやサーバーを、保守点検のために一定期間、アクセス不可にする場合もあります。

あるレポートを指定の Reports Server から指定の宛先に対して実行する権限がユーザーにあると判断された後は、そのレポートによってアクセスされるデータソースに対する権限がユーザーにあるかどうかの確認も必要になります。

OracleAS Portal では、レポート、プリンタ、カレンダーおよび Reports Server に対するリソース・セキュリティを最初から利用できます。通常の構成では、管理者または開発者が、OracleAS Portal にアクセスするユーザーまたはグループがどのレポート、Reports Server、プリンタを使用できるかを指定します。

アプリケーション・セキュリティの場合と同様、独自のメカニズムでリソースを保護することも可能です。この場合、OracleAS Reports Services には、これら Oracle 以外のコンポーネントとの統合を実現するインタフェースが用意されています。

関連項目： [第 10.2 項「OracleAS Reports Services セキュリティの構成」](#)
には、インタフェースの詳細に関する記述があります。

10.1.1.3 データソース・セキュリティ

データソース・セキュリティでは、特定のデータソースにあるデータにアクセスできるユーザーまたはロールを定義します。1つのレポートが複数のデータソースにアクセスすることがあるため、レポートを実行し、その出力を表示するには、現在のユーザーが、レポートのアクセス対象となるデータソースすべてに対して権限を持っている必要があります。データソースへのアクセスを許可するのは、データソース管理者（通常は DBA と呼ばれる）です。データソース・セキュリティは、レポート環境を構成する前に、確立しておく必要があります。

OracleAS Reports Services では、次の 2 つの異なる方法でデータソース・セキュリティを確立できます。

- **Single Sign-On** ユーザーにデータソース接続情報を関連付けることができます。この場合、ユーザーがデータソースに初めてアクセスしようとするときに、Oracle Delegated Administration Services がユーザーに対して、データソース接続用のリソースを作成するよう指示します。ユーザーがこのデータソース・リソースを作成すると、OracleAS Single Sign-On がこれを Oracle Internet Directory 内でユーザーに関連付けます。Single Sign-On ユーザーに関連付けられたデータソース・リソースは、ユーザーが持つ Single Sign-On の識別情報の一部となり、ユーザーはこのデータソースに別途ログインせずにアクセスできます。この方法には、重要な利点が 2 つあります。最初の利点は、各ユーザーが Single Sign-On の識別情報を使用することによって、その都度ログインしなくても、データソースにアクセスできることです。もう 1 つの利点は、データソースのログイン情報がユーザーの識別情報とともに格納されるため、そのログイン情報をレポートの URL やキー・マッピングにハードコーディングする必要がないことです。そのため、1 つのレポート URL を情報の異なる複数のユーザーが使用できます。
- レポート URL やキー・マッピングで、AUTHID や、レポートに必要な接続パラメータ (USERID など) をコーディングできます。この機能は、OracleAS Reports Services の以前のリリースとほとんど同じです。URL 構文の詳細は、[第 13.1 項「Reports の URL 構文」](#)を参照してください。キー・マッピングの詳細は、[第 13.12 項「キー・マップ・ファイルの使用方法」](#)を参照してください。

アプリケーション・セキュリティやリソース・セキュリティの場合と同様、独自のメカニズムでデータソースを保護することも可能です。この場合、OracleAS Reports Services には、これら Oracle 以外のコンポーネントとの統合を実現するインタフェースが用意されています。

関連項目： [第 10.2 項「OracleAS Reports Services セキュリティの構成」](#)
には、インタフェースの詳細に関する記述があります。

10.1.2 認証とアクセス権の適用

レポート・リクエストに対するアクセス制御は、OracleAS Single Sign-On を使用してもしなくても管理可能です。

- [OracleAS Single Sign-On を使用するレポート・リクエスト処理](#)
- [OracleAS Single Sign-On を使用しないレポート・リクエスト処理](#)

10.1.2.1 OracleAS Single Sign-On を使用するレポート・リクエスト処理

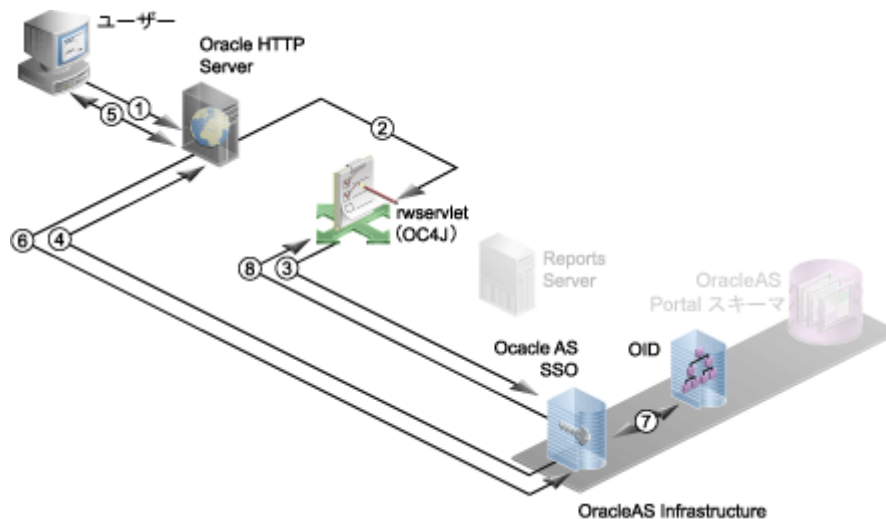
OracleAS Single Sign-On では、認証済アプリケーション・ユーザーの追跡に暗号化された Cookie を使用します。Reports Servlet が、セキュリティ保護された Reports Server 上でのレポート実行を指示するユーザー・リクエストを受け取ると、そのユーザーがすでに OracleAS Single Sign-On を通じてログインしているかどうか（そのユーザーの Single Sign-On Cookie が存在するかどうか）を確認するために、Oracle HTTP Server に `getRemoteUser` メソッドでコールして問い合わせます。

- ユーザーがすでにログインしている場合（Cookie が存在する場合）は、Reports Servlet がユーザーの識別情報を Oracle HTTP Server から取得します。
- ユーザーがまだログインしていない場合（Cookie がまだ存在しない場合）は、Oracle HTTP Server がユーザーを OracleAS Single Sign-On にリダイレクトし、OracleAS Single Sign-On がユーザーにログインを求めます。ユーザーが認証されると、Single Sign-On Cookie が作成され、そのユーザーは Reports Servlet にリダイレクトされます。その後、ユーザーがすでにログインしている場合（Cookie が存在する場合）と同様の処理が行われます。

注意： レポート・リクエストが Reports Servlet からではなく OracleAS Portal 内から送信された場合、OracleAS Reports Services では、レポートに対するユーザーの権限を同様の方法で確認してからレポートを実行します。未認証（PUBLIC）ユーザーが公開ページを閲覧する場合でも、OracleAS Reports Services では、PUBLIC ユーザー・アカウントがレポートに対して適切な権限を持っているかどうかを確認されます。

10.1.2.1.1 Single Sign-On を使用するレポート・リクエスト・フロー このシナリオでは、レポート・リクエストが Single Sign-On の有効な、セキュリティ保護された Reports Server に送信されます。送信されたレポート・リクエストは、[図 10-1](#) の認証と [図 10-3](#) の許可に示すように、2 段階のセキュリティ・チェックを受けます。

図 10-1 SSO を使用する認証プロセス



次の各手順の番号は、[図 10-1](#) の番号に対応しています。

1. ユーザーが URL を使用してレポート・リクエストを送信します。

レポート・リクエストは、次のいずれかの方法で行われます。

- OracleAS Portal の内部から、ユーザーが「実行」リンクをクリックするなどして、レポート・オブジェクトの実行を要求します。ユーザーは、OracleAS Portal にログインする必要があります（つまり、OracleAS Single Sign-On にログインします）。OracleAS Portal では、セキュリティの一環として、レポート・オブジェクトの参照に必要なセキュリティ権限がユーザーにあることを確認します。たとえば、あるページ上にレポート・オブジェクトが存在する場合、ユーザーには、そのページとレポート・オブジェクトを参照するための適切な権限が必要です。その権限がない場合、OracleAS Portal が、ユーザーに対してそのページまたはレポート・オブジェクトを表示しません。
- OracleAS Portal の外部から、ユーザーがレポートを要求する URL を含む Web ページ上のリンクやブックマークを選択します。

注意： この URL には、必要に応じて、次の形式の値を持つ Single Sign-On パラメータ (SSOCONN) が格納されます。または、この URL でキー・マップ・ファイルを介して、そのパラメータを参照します。

`key_name/data_source_type/parameter_name`

Oracle データベースの場合、Single Sign-On 値は次のようになります。

`mykey/OracleDB/userid`

データソース・タイプおよびパラメータ名を指定しない場合は、Oracle データベースであると見なされます。

2. Oracle HTTP Server がリクエストを、OC4J にデプロイされた Reports Servlet にルーティングします。

URL では、このレポートを Reports Servlet 経由で実行するか、JSP 経由で実行するかの設定に応じて、ユーザーを Reports Servlet または JSP にリダイレクトします。

3. Reports Servlet は、OracleAS Single Sign-On にユーザーの認証を要求します。
4. OracleAS Single Sign-On サーバーがユーザー名とパスワードを要求します。
5. Oracle HTTP Server がユーザーにログイン・ページを表示し、ユーザーがユーザー名とパスワードを入力します。
6. ユーザー名とパスワードが OracleAS Single Sign-On に渡されます。
7. OracleAS Single Sign-On が証明書を Oracle Internet Directory に確認します。
8. ユーザーが認証されると、OracleAS Single Sign-On サーバーがユーザーが認証されたというメッセージを Reports Servlet に渡します。

URL で SSOCONN を使用した場合、Reports Servlet が、Oracle Internet Directory で Single Sign-On キーを照合して、Single Sign-On キーがデータソース接続文字列（たとえば、`scott/tiger@my_or_db`）にすでにマッピングされているかどうかを確認します。

SSOCONN を使用しており、そのキーに関連付けられている接続文字列が Oracle Internet Directory に存在する場合は、Reports Servlet が、その接続文字列をレポートのデータソース接続に使用します。

注意： この機能により、異なるデータソース接続文字列を使用する多くのユーザーが、同じレポートの URL を使用できるようになります。

SSOCONN を使用していても、そのキーに対する接続文字列がまだ Oracle Internet Directory に存在しない場合は、Oracle Delegated Administration Services の「リソースの作成」ページが表示され、ユーザーはそこにデータソース接続文字列を入力する必要があります。図 10-2 を参照してください。

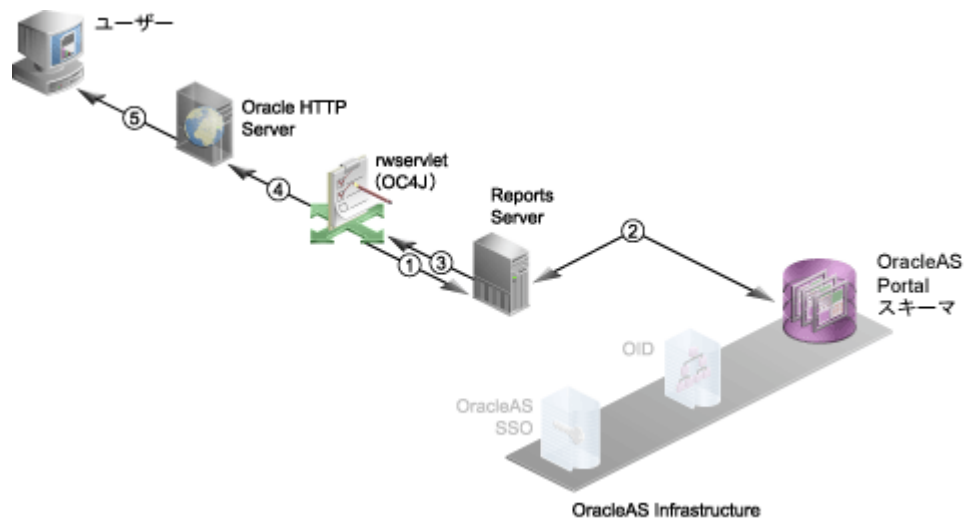
この文字列は今後の使用のために、Oracle Delegated Administration Services によって Oracle Internet Directory に格納され、Reports Servlet が、新しく入力された接続文字列をレポートのデータソース接続文字列として使用します。

図 10-2 Oracle Delegated Administration Services の「リソースの作成」



ユーザーが認証されると、レポート・リクエストは図 10-3 に示すように許可プロセスを経る必要があります。

図 10-3 SSO を使用する許可プロセス



次の各手順の番号は、[図 10-3](#) の番号に対応しています。

1. Reports Servlet がリクエストを Reports Server に転送します。

Reports Servlet が URL から (SSOCONN を使用した場合は Oracle Internet Directory 情報からも) コマンドラインを作成し、これを Reports Server に渡します。

2. Reports Server では、OracleAS Metadata Repository の OracleAS Portal スキーマに対するユーザーの権限が確認されます。

また、Reports Server では、ユーザーが、指定したサーバー上で特定の時刻に指定の宛先に対してレポートを実行する権限を持っているかどうかもチェックされます。この妥当性チェックがなんらかの理由で失敗した場合、ユーザーに対してエラーが返され、この処理が終了します。

注意： ユーザーが showjobs や getserverinfo などの Reports Servlet の Web コマンドを実行している場合、レポートを実行するかわりに、Reports Server ではユーザー証明書の妥当性が Oracle Internet Directory で確認されます。Reports Server で、このユーザーに Reports Servlet の Web コマンドを実行する管理権限があることが確認されます。OracleAS Portal のレポートへのアクセス制御の詳細は、[第 12.1 項「Reports ユーザーおよび名前付きグループの作成」](#)を参照してください。

3. Reports Server でレポート・リクエストが実行され、レポート出力が Reports Servlet に渡されます。

Reports Server がこのジョブをエンジンに委譲し、そこでデータソースのアクセス、データの取出しおよびレポートのフォーマットが行われます。

4. レポート出力が、Oracle HTTP Server に渡されます。

5. レポート出力が、ユーザーに渡されます。

処理された出力は、指定の宛先に送信されます。宛先に従って、この出力はブラウザに返されるか ([図 10-3](#))、プリンタに送信されるか、後で参照するためにファイルに保存されるか、または FTP サーバーなどに送信されます。

10.1.2.2 OracleAS Single Sign-On を使用しないレポート・リクエスト処理

Single Sign-On を使用せずに、Reports Server のセキュリティ保護されたインスタンスへユーザーがアクセスする場合、Oracle Reports 6i の場合と同様に、独自の認証メカニズムを使用して Reports Servlet により識別情報を提供する必要があります。HTTP 1.0 プロトコルはステートレスであるため (つまり、サーバーへの各コールは、事実上独立している)、Cookie が保持されていないかぎり、レポート・リクエストのたびにユーザーの認証が必要になる場合があります。セッションごとに 1 度の認証で済むように、Reports Servlet には独自のクライアント側 Cookie (AUTHID) が保存されます。ここには、現行セッションに必要な認証情報が記録されます。一度ユーザーが認証されると、暗号化された Cookie がブラウザで作成され、ユーザーはリクエストごとに認証を受けることなく、複数のレポート・ジョブを実行できます。

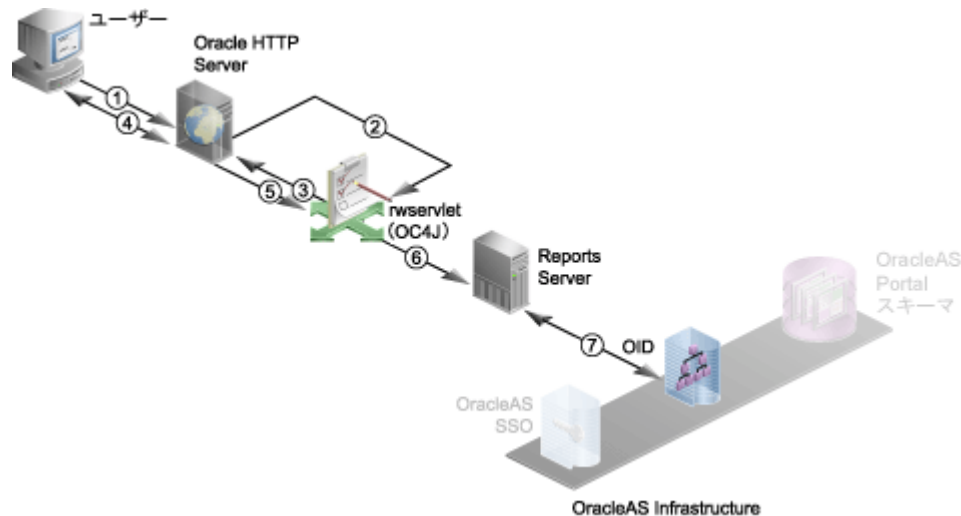
注意： 特定のレポートに対してユーザーに認証を要求する場合は、SHOWAUTH コマンドライン・キーワードを使用できます。または、キー・マップ・ファイル内の対応するレポート・エントリに %S を使用することもできます。このファイルは通常 cgicmd.dat という名前、ORACLE_HOME¥reports¥conf に格納されています。%S を使用すると、ユーザーはレポートをコールするたびに、ユーザー名とパスワードの入力が要求されます。

ユーザーがブラウザ・セッションを終了すると AUTHID Cookie も終了しますが、この方法だけで Cookie を終了するのは十分とはいえません。セッション内の Cookie の存続期間は制限する必要があります。たとえば、ユーザーがログインし、昼食のためにブラウザ・セッションを起動したまま席を外したとします。こうした状況でのセキュリティ侵害の可能性を最小限に抑えるために、管理者は、`rwervlet.properties` ファイル内で `COOKIEEXPIRE` パラメータを指定できます。Reports Servlet でジョブ・リクエストが受信されると、Cookie に保存された時間と現在のシステム時間が比較されます。この時間が環境変数で定義された時間（たとえば、30 分）を超えていると、Cookie は否定され、ユーザーには認証情報を提供する必要が生じます。

関連項目： `COOKIEEXPIRE` パラメータと `rwervlet.properties` ファイルの詳細は、第 3.4 項「Reports Servlet の構成」を参照してください。

10.1.2.2.1 OracleAS Single Sign-On を使用しないレポート・リクエスト・フロー このシナリオでは、レポート・リクエストが Single Sign-On の無効な、セキュリティ保護された Reports Server に送信されます。この場合、Reports Servlet または JSP レポートは、ブックマークを使用してコールするか、OracleAS Portal コンポーネントからコールすることができます。送信されたレポート・リクエストは、図 10-4 の認証と図 10-5 の許可に示すように、2 段階のセキュリティ・チェックを受けます。

図 10-4 SSO を使用しない認証プロセス



次の各手順の番号は、図 10-4 の番号に対応しています。

1. ユーザーが URL を使用してレポート・リクエストを送信します。
ユーザーは、Web ページ上のリンクやブックマークなどを通じて、レポート・リクエストを送信する URL にアクセスし、そこで URL を選択する必要があります。
2. Oracle HTTP Server がリクエストを、OC4J にデプロイされた Reports Servlet にルーティングします。
3. Reports Servlet がユーザーの証明書（ユーザー名とパスワード）を要求します。

Reports Servlet は、URL または既存の Oracle Reports AUTHID Cookie で、AUTHID パラメータがないかどうかを調べます。AUTHID パラメータを見つけると、それを使用してユーザーを認証します。AUTHID パラメータを見つけられない場合は、既存の Oracle Reports AUTHID Cookie を探します（レポートが OracleAS Portal から起動された場合は、AUTHID が URL に自動的に追加されます）。AUTHID パラメータと Oracle Reports AUTHID Cookie のどちらも見つからない場合、Reports Servlet が「システム認証」ページを Oracle HTTP Server に送信し、ユーザーに表示します。

4. Oracle HTTP Server がユーザーにログイン・ページを表示し、ユーザーがユーザー名とパスワードを入力します。

ログイン・ページでは、ユーザーが Single Sign-On ユーザー名とパスワードを入力する必要があります。この情報は、後で参照するために Oracle Reports AUTHID Cookie に保存されます。

5. ユーザー名とパスワードが Reports Servlet に渡されます。

データソース証明書の一部だけが URL に指定されている場合（たとえば、USERID=scott@orqa）は、データベース認証ページにその一部の証明書が表示されます。ユーザーは、証明書の残りの部分を指定しないと、先に進めません。どのデータベース認証ページを使用するかは、rwservlet.properties ファイルの DBAUTH パラメータで制御できます。データソース証明書が指定されていない場合は、データベース認証ページが表示されず、レポートにはデータソースが不要であると見なされます。

関連項目： DBAUTH パラメータと rwservlet.properties ファイルの詳細は、第 3.4 項「Reports Servlet の構成」を参照してください。

データソース証明書は、後で参照するために Oracle Reports USERID Cookie に保存されます。プラグガブル・データソース (PDS) 証明書は、Oracle Reports USERID Cookie には保存されないことに注意してください。

6. Reports Servlet がユーザー名とパスワードを Reports Server に転送します。

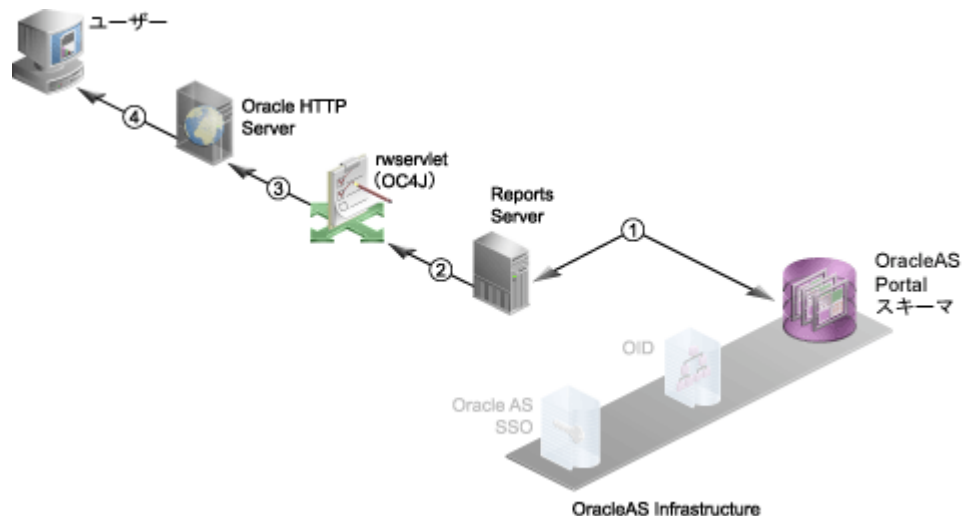
Reports Servlet は、これまでの手順で取得した必要な情報でコマンドラインを作成し、これを Reports Server に渡します。

7. Reports Server が Oracle Internet Directory に問い合せてユーザーを認証（ユーザー名とパスワードを確認）します。

Reports Server では、ユーザー証明書の妥当性が Oracle Internet Directory サーバーで確認されます。この妥当性チェックがなんらかの理由で失敗した場合、ユーザーに対してエラーが返され、この処理が終了します。

ユーザーが認証されると、レポート・リクエストは図 10-5 に示すように許可プロセスを経る必要があります。

図 10-5 SSO を使用しない許可プロセス



次の各手順の番号は、[図 10-5](#) の番号に対応しています。

1. **Reports Server では、OracleAS Metadata Repository の OracleAS Portal スキーマに対するユーザーの権限が確認されます。**

また、Reports Server では、ユーザーが、指定したサーバー上で特定の時刻に指定の宛先に対してレポートを実行する権限を持っているかどうかもチェックされます。この妥当性チェックがなんらかの理由で失敗した場合、ユーザーに対してエラーが返され、この処理が終了します。

注意： ユーザーが showjobs や getserverinfo などの Reports Servlet の Web コマンドを実行している場合、レポートを実行するかわりに、Reports Server ではユーザー証明書の妥当性が Oracle Internet Directory で確認されます。Reports Server で、このユーザーに Reports Servlet の Web コマンドを実行する管理権限があることが確認されます。OracleAS Portal のレポートへのアクセス制御の詳細は、[第 12.1 項「Reports ユーザーおよび名前付きグループの作成」](#) を参照してください。

2. **ユーザーにレポートの実行が許可された場合、Reports Server はレポート・リクエストを実行し、レポート出力を Reports Servlet に渡します。**

Reports Server がこのジョブをエンジンに委譲し、そこでデータソースのアクセス、データの抽出およびレポートのフォーマットが行われます。

3. **レポート出力が、Oracle HTTP Server に渡されます。**
4. **レポート出力が、ユーザーに渡されます。**

処理された出力は、指定の宛先に送信されます。宛先に従って、この出力はブラウザに返されるか ([図 10-5](#))、プリンタに送信されるか、後で参照するためにファイルに保存されるか、または FTP サーバーなどに送信されます。

10.1.3 Oracle Identity Management インフラストラクチャの利用

OracleAS Reports Services では、Oracle Identity Management インフラストラクチャの一部である OracleAS Single Sign-On の機能を利用できます。

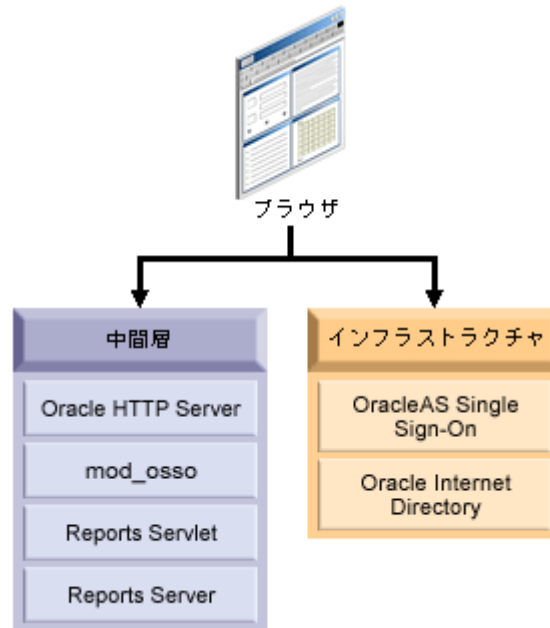
10.1.3.1 OracleAS Single Sign-On

従業員、顧客、パートナーが使用するために企業が配置する Web ベースの E-Business アプリケーションの数が増大し続けているため、多くの企業が、Single Sign-On 機能の採用を検討する必要に迫られています。Single Sign-On とは、複数のセキュリティ・システムに個別にログインするかわりに、単一のセキュリティ・システムに一度だけログインするという機能を指します。Single Sign-On では、各ユーザーがアクセスする必要のあるすべてのデータおよび関連リソースを、1 つの ID とパスワードで管理します。

OracleAS Single Sign-On を使用することによって、Web アプリケーション内での OracleAS Reports Services の利用が容易になります。OracleAS Single Sign-On により、各ユーザーは認証を一度行うだけで済みます。

10.1.3.1.1 Single Sign-On コンポーネント 図 10-6 は、Single Sign-On のコンポーネント・アーキテクチャの概要です。

図 10-6 SSO のアーキテクチャ



Single Sign-On 環境のコンポーネントは、次のとおりです。

- **クライアント Web ブラウザ**
- **Oracle HTTP Server**
Oracle HTTP Server は、クライアント・ブラウザからのリクエストを処理します。
- **Reports Servlet**
- **Oracle Application Server Containers for J2EE**
Reports Servlet は、Oracle HTTP Server の Oracle Application Server Containers for J2EE (OC4J) の内部で実行される、OracleAS Reports Services のコンポーネントです。Oracle HTTP Server がレポート・リクエストを受信すると、Reports Servlet がジョブ・リクエスト Reports Server に渡します。
- **Reports Server**
Reports Server は、クライアント・リクエストを処理します。この処理には、認証と許可の確認、スケジューリング、キャッシュおよび配布が含まれます。
- **OracleAS Single Sign-On**
OracleAS Single Sign-On は、ユーザーの Single Sign-On セッションを管理します。Oracle Internet Directory 内を検索して、ユーザーのログイン証明書を検証します。
- **Oracle Internet Directory**
Oracle Internet Directory は、高度にスケーラブルな、LDAP バージョン 3 の Oracle 固有サービスであり、Oracle 共通のユーザー識別情報をホスティングします。OracleAS Single Sign-On では、Oracle Internet Directory に格納されている情報に対してユーザーが認証されます。すでに説明したとおり、OracleAS Reports Services に対して Single Sign-On が有効であれば、Oracle Internet Directory でユーザーおよびグループの権限情報がチェックされます。さらに、Oracle Internet Directory からデータソース接続情報が取得されます。

- **Oracle Delegated Administration Services**

Delegated Administration Service は、Oracle Internet Directory を更新するための包括的なインタフェースを提供します。OracleAS Reports Services は、Oracle Internet Directory のデータソース接続文字列にまだ関連付けられていない Single Sign-On キーを検出すると、Oracle Delegated Administration Services を表示します。

詳細は、第 11 章「OracleAS Single Sign-On の構成と管理」を参照してください。

10.2 OracleAS Reports Services セキュリティの構成

この項では、OracleAS Reports Services の構成に関する考慮事項について概説します。

10.2.1 OracleAS Reports Services セキュリティ・オプションの構成

OracleAS Reports Services セキュリティのデフォルトの実装には、OracleAS Reports Services インストールと連携して動作するよう事前構成された Oracle コンポーネントすべて（第 10.1.1 項「リソースの保護」を参照）が含まれます。独自のセキュリティ構成を実装する場合は、第 11 章「OracleAS Single Sign-On の構成と管理」および第 12 章「OracleAS Portal でのレポートの配布」の手順に従って、それらのコンポーネントのすべてまたは一部を使用できます。たとえば、データソース・セキュリティまたは OracleAS Portal を実装せずに、OracleAS Single Sign-On を使用することが可能です。また、別の構成では、他のインターネット・ディレクトリを使用して、ユーザーおよびグループの情報を格納することも可能です。前述のどのセキュリティ・コンポーネントも実装しなくても、セキュリティ保護された Reports Server を構成することができます。その場合のセキュリティは、Oracle6i Reports と同程度のものです。

注意： セキュリティの最高レベルでは、Oracle HTTP Server とのすべての通信に SSL を使用するように構成できます。Oracle HTTP Server は、OpenSSL モジュールで Secure Sockets Layer (SSL) および HTTP Secure Sockets Layer (HTTPS) をサポートします。Oracle HTTP Server で SSL が設定されると、『Oracle HTTP Server 管理者ガイド』を参照)、Reports Servlet が SSL ポート番号を自動的に検出します。

10.2.1.1 OracleAS Portal

OracleAS Portal では、OracleAS Reports Services で使用可能な数多くのセキュリティ・オプションを用意しています。これを利用して、重要なデータに適切なユーザーが安全にアクセスできるようになります。OracleAS Portal のセキュリティ機能を正しく使用することで、ユーザーにはそれぞれに許可されたデータのみが表示されます。

OracleAS Portal を使用して、次の制御を行います。

- 各レポートにアクセスできるユーザー
- レポートを実行できる期間
- レポートの実行に使用できるサーバーとプリンタ
- ユーザーが編集できるレポート・パラメータとその値の範囲

OracleAS Portal は、データを公開および開発するためのブラウザ・ベースのソリューションで、情報を Web 上に公開し、Web ベースのデータ駆動型アプリケーションを作成するための Web ベース・ツールを提供します。

OracleAS Portal は OracleAS Reports Services と緊密に統合されているため、データを公開するための堅牢で安全な環境を構築できます。OracleAS Portal には、OracleAS Reports Services のセキュリティを簡単に設定するためのウィザードが用意されています。これには、レポート、Reports Server、プリンタ、出力フォーマットおよびレポート・パラメータに対するユーザー・アクセスをそれぞれ定義するためのウィザードが含まれます。

定義したアクセス制御情報は、OracleAS Portal のリポジトリに格納されます。その後、OracleAS Portal ユーザーとして、登録済の RDF および JSP を OracleAS Portal ページにオプションで公開できます。OracleAS Portal の他の機能と同様、Portal を使用したレポートの配布は必須ではありません。レポートはこれまでどおりコマンドラインから配布でき、さらに OracleAS Portal を通じてアクセス制御機能も利用できます。

OracleAS Reports Services のセキュリティ機能に対するアクセスは、レポート・リンクやレポート内容の公開に Portal を使用しているかどうかに関係しません。Portal を通じて公開しない場合でも、OracleAS Portal で使用可能な OracleAS Reports Services のセキュリティ機能を利用して、すべてのレポートに対するユーザーのアクセスを制御できます。

OracleAS Portal を通じてレポートをポートレットとして公開する場合は、OracleAS Reports Services で Single Sign-On 機能が使用されます。Single Sign-On によって、ユーザーは何度もログインする必要がなくなります。つまり、最初にポータルにログインすれば、ポータル内でポートレットとして公開されている各レポートにログインする必要がありません。OracleAS Single Sign-On でログインすると、OracleAS Portal によって、すべての登録済ポートレット・プロバイダおよびサブシステムへのログインが自動的に行われます。

関連項目： OracleAS Portal 内でのレポート要求フローの詳細は、[第 10.2.1.1 項「OracleAS Portal」](#) を参照してください。

OracleAS Single Sign-On および OracleAS Portal の詳細は、『Oracle Application Server セキュリティ・ガイド』を参照してください。このドキュメントおよびその他の関連ドキュメントは、OTN (Oracle Technology Network) (<http://www.oracle.com/technology/index.html>) に掲載されています。

詳細は、[第 12 章「OracleAS Portal でのレポートの配布」](#) を参照してください。

10.2.1.2 セキュリティ・インタフェース

Reports Software Development Kit (RSDK) の Security API を使用すると、独自のセキュリティ・モデルを Reports Server に統合できます。OracleAS Reports Services では、提供される API を使用して、必要なセキュリティ機能をプラグインできます。

Security API によって、次の制御が可能になります。

- 各レポートにアクセスできるユーザー
- レポートを実行できる期間
- レポートの実行に使用できるサーバーとプリンタ
- ユーザーが編集できるレポート・パラメータとその値の範囲

RSDK に含まれるチュートリアルでは、認証情報の格納に XML ファイルを使用して、独自のセキュリティを統合する方法を示しています。このチュートリアルを完了すると、次のことができますようになります。

- OracleAS Reports Services でのセキュリティ・クラスの実装
- OracleAS Reports Services でのセキュリティ・クラスの登録
- OracleAS Reports Services でのセキュリティ・クラスの使用

チュートリアルおよび詳細は、Reports Software Development Kit (RSDK) を参照してください。RSDK を入手するには、OTN (Oracle Technology Network) の Oracle Reports 10g ページ (<http://www.oracle.com/technology/products/reports/index.html>) で「**SDK**」をクリックします。

OracleAS Single Sign-On の構成と管理

Single Sign-On を使用すると、各ユーザーに一意的な識別子を設定し、その識別子をユーザー固有のリソースやデータソースと結び付けることができます。たとえば、ユーザーは OracleAS Portal などの環境にログインして、必要な権限のある特定のレポートやプリンタにアクセスします。この環境からレポートを実行するとき、ユーザーはレポートに必要なデータソースにアクセスできます。これは、ユーザーのデータソース接続情報が、OracleAS Portal へのログインに使用された単一のユーザー ID とともに格納されているためです。このように、一度ログインすると、レポートの実行に必要なすべてのリソースやデータソースにアクセスできるようになります。

OracleAS Reports Services にはセキュリティに対する柔軟なアプローチが備わっており、前述の構成を様々な方法で実装できます。たとえば、単一のユーザー ID とともにデータソース接続情報を格納しないように選択することもできます。また、OracleAS Portal のようなプラットフォームのかわりに、URL を直接指定してレポートを起動することもできます。さらに、レポートが公開されている場合、セキュリティが不要であれば、レポートのセキュリティをすべてオフにできます。

この章では、OracleAS Reports Services で OracleAS Single Sign-On の様々な構成を実装および管理する方法を説明します。

- [前提条件](#)
- [OracleAS Single Sign-On のデフォルトの構成](#)
- [OracleAS Single Sign-On の管理](#)
- [Oracle Internet Directory の接続エンティティの選択](#)
- [OracleAS Forms Services のセキュリティに関する考慮事項](#)

11.1 前提条件

OracleAS Single Sign-On は、セキュアなサーバー環境にのみ実装できます。つまり、OracleAS Reports Services で OracleAS Single Sign-On の実装を検討する場合、Reports Server 構成ファイルにセキュリティ・ポリシーが正しく設定されていることが前提となります。

注意： セキュリティ設定については、次の各章で取り上げています。第 3 章「OracleAS Reports Services の構成」では、サーバーのセキュリティ・ポリシーを定義する Java クラスを指定する方法を説明し、第 12 章「OracleAS Portal でのレポートの配布」では、OracleAS Portal に OracleAS Reports Services レポートを配布する方法を説明しています。また、第 11.3 項「OracleAS Single Sign-On の管理」では、SSOCONN コマンドライン・キーワードに関する情報を提供しています。

OracleAS Single Sign-On を使用する場合は、管理者が各ユーザーにユーザー ID を設定します。管理者は、Oracle Internet Directory において、そのユーザー・インタフェースである Oracle Delegated Administration Services を使用してこれを行います。Oracle Delegated Administration Services には、スタンドアロンでアクセスすることも、OracleAS Portal からアクセスすることもできます。どちらの場合も、Oracle Internet Directory に情報が保存されます。

ユーザー ID は、ユーザー名とパスワードで構成されます。ユーザーを設定すると、そのユーザーにデータソース接続文字列が関連付けられます。ユーザーは、ログイン時にユーザー名とパスワード（ユーザー ID）を入力する必要があります。それにより、そのユーザー ID と関連付けられたすべてのデータソースにアクセスできるようになります。OracleAS Single Sign-On からセッション Cookie が発行されますが、これはそのセッションに許可されているすべての入力を開くための鍵として効果的に機能します。

注意： SSO 関連コンポーネント（Oracle Internet Directory など）を設定するための要件と手順の詳細は、『Oracle Internet Directory 管理者ガイド』および『Oracle HTTP Server 管理者ガイド』を参照してください。これらはいずれも、Oracle Application Server のドキュメントが収録された CD および Oracle Technology Network (<http://www.oracle.com/technology/index.html>) にあります。

11.2 OracleAS Single Sign-On のデフォルトの構成

デフォルトでは、Reports Server はセキュリティで保護されており、レポートを実行するためには、Single Sign-On の有効なユーザー ID とパスワードを使用してログインする必要があります。Reports Server は、デフォルトで、Oracle Application Server の一部としてインストールされる OracleAS Single Sign-On インスタンスと連携するように構成されています。Oracle Application Server とともにインストールされる Oracle Internet Directory インスタンスが、ユーザーおよびグループ情報のデフォルトのリポジトリとして使用されます。別の Oracle Internet Directory インスタンスを使用するように Reports Server を構成する場合やセキュリティを無効にする場合は、第 11.3 項「OracleAS Single Sign-On の管理」を参照してください。Oracle Internet Directory にユーザーを追加する方法の詳細は、『Oracle Internet Directory 管理者ガイド』を参照してください。また、Oracle Application Server のインストールごとに、Reports Server インスタンスは、その Oracle Application Server のインストールに固有のアプリケーション・エンティティとして Oracle Internet Directory に接続します。この動作の詳細は、第 11.3.4 項「Oracle Internet Directory への接続」を参照してください。

ユーザーがまだ OracleAS Single Sign-On にログインしていない場合、Reports Servlet を使用して Reports Server に対してレポートを実行しようとする、ログインするように要求されます。レポートのユーザー・パラメータに SSOCONN が含まれている場合は、OracleAS Single Sign-On によって Oracle Internet Directory 内にユーザーのデータソース接続情報があるかどうか検索されます。見つからない場合、ユーザーは新しいリソースを作成するように要求されず、Reports Servlet で利用可能なコマンドの詳細は、第 A.2.6 項「[rwservlet](#)」を参照してください。SSOCONN の詳細は、第 11.3.3.1 項「[SSOCONN](#)」を参照してください。

Reports Server は、デフォルトで OracleAS Portal と連携して動作するように構成されています。必要に応じて、ポータルにレポートを追加して、ユーザーがポータルからレポートを起動できるようにすることができます。この場合、ユーザーはポータルにログインする必要があります。ポータルにログインすることによって OracleAS Single Sign-On で識別されるため、レポートの起動時に再度ログインするように要求されることはありません。

必要に応じて、OracleAS Portal で Reports Server に関連付けられているリソース（レポート、プリンタ、Reports Server、カレンダーなど）にアクセス制御を定義することもできます。リソースへのアクセスを制御するには、それらのリソースをポータルに追加して、それぞれのアクセス・オプションを指定する必要があります。OracleAS Portal で指定したリソースのアクセス制御は、ポータルの外部で実行するレポートにも適用されます。たとえば、ユーザーが Reports Servlet を使用してレポートを実行しようとする、OracleAS Portal で指定した任意のアクセス制御が適用されます。

関連項目： OracleAS Portal と OracleAS Reports Services との統合の詳細は、第 12 章「[OracleAS Portal でのレポートの配布](#)」を参照してください。

11.3 OracleAS Single Sign-On の管理

この項では、OracleAS Reports Services のセキュリティを維持するために実行する必要がある管理作業を説明します。

- [OracleAS Single Sign-On の有効化と無効化](#)
- [Reports Server のセキュリティの有効化と無効化](#)
- [データソースのセキュリティの有効化と無効化](#)
- [Oracle Internet Directory への接続](#)

11.3.1 OracleAS Single Sign-On の有効化と無効化

OracleAS Single Sign-On のデフォルト設定を使用する場合、Reports Servlet 構成ファイル (`rwservlet.properties`) の `SINGLESIGNON` パラメータは `YES` に設定されています。つまり、OracleAS Single Sign-On を使用してユーザーの認証が行われます。OracleAS Single Sign-On を使用しないようにするには、このパラメータを `NO` に変更します。`NO` を選択すると、Reports Server では独自の認証メカニズムを使用してユーザー認証が行われます。Reports Servlet 構成ファイルは、通常、次の場所にあります。

```
ORACLE_HOME\reports\conf
```

通常、`SINGLESIGNON` の値はインストール後にコメントアウトされますが、デフォルト値は `YES` です。

注意： OracleAS Reports Services は、OracleAS Single Sign-On のデフォルト設定を使用するように構成されます。オラクル社ではこれを通常のセキュリティ配置モデルと考えており、完全にカスタマイズしたセキュリティ構成で実行する場合にのみオフにしてください。

11.3.2 Reports Server のセキュリティの有効化と無効化

Reports Server のセキュリティのオンおよびオフは、Reports Server 構成ファイルで設定します。Reports Server 構成ファイルはデフォルトで、`ORACLE_HOME/reports/conf/servername.conf` にあります。このファイルには、次のようなセキュリティ要素が格納されています。

```
<security id="rwSec" class="oracle.reports.server.RWSecurity">
<!--property name="securityUserId" value="portal_id/portal_password@portal_schema"
confidential="yes" encrypted="no"-->
<property name="oidEntity" value="%REPORTS_OID_ENTITY%" confidential="yes"
encrypted="no"/>
</security>
```

注意： Oracle Reports 10g より前のリリースでは、`securityUserId` プロパティの指定が異なっていました。Oracle Reports 10g 以降のリリースにもこの古いプロパティが指定されていますが、コメントアウトされています。

このセキュリティ要素は、Reports Server のセキュリティを有効にすることを示すために、デフォルトで、構成ファイル内の次の 2 つのデフォルトの `job` 要素から参照されます。

```
<job jobType="report" engineId="rwEng" securityId="rwSec"/>
<job jobType="rwurl" engineId="rwURLEng" securityId="rwSec"/>
```

Reports Server のセキュリティを無効にするには、`job` 要素の指定から、`security` 要素と `securityId` 属性を削除またはコメントアウトする必要があります。

注意： Oracle Reports セキュリティを構成するには、OracleAS Portal をインストールし、セキュリティ情報を OracleAS Portal リポジトリに保存する必要があります。OracleAS Portal を構成して OracleAS Reports Services セキュリティ情報を保存する詳細は、OTN (<http://www.oracle.com/technology/products/reports/html/ocs/getstart/whitepapers/securing9i.pdf>) にあるホワイトペーパー『Securing Oracle Reports』を参照してください。

11.3.3 データソースのセキュリティの有効化と無効化

OracleAS Single Sign-On を介してデータソースのセキュリティを有効にする手順は次のとおりです。

- レポートを起動する URL に `SSOCONN` を組み込みます。
- 3 つの方法のいずれかを使用して、Oracle Internet Directory にデータソース接続情報を入力します。

独自のプラグgable データソースに対して OracleAS Single Sign-On を介したデータソースのセキュリティを実装する場合は、さらに次の手順を実行します。

- Oracle Internet Directory に新しいリソース・タイプを追加します。

次の各項では、これらの手順を説明します。

11.3.3.1 SSOCONN

OracleAS Single Sign-On を介したデータソースのセキュリティを有効にするには、次の形式の値を持つ OracleAS Single Sign-On パラメータ (SSOCONN) を URL に組み込む (またはキー・マップ・ファイルを介して参照する) 必要があります。

key_name/data_source_type/conn_string_parameter

key_name は、データベースへの接続に必要な情報を提供する Oracle Internet Directory に格納されている文字列にマップされています。Oracle Reports によって *key_name* が検出されると、Oracle Internet Directory に格納されている、対応する鍵を現行のユーザーが所有しているかどうかチェックされます。所有している場合、Oracle Reports は、その鍵に格納されている文字列を使用してデータソースに接続します。所有していない場合は、*key_name* が公開鍵にマップされているかどうかチェックされます。マップされている場合は、その鍵が使用されます。マップされていない場合、ユーザーは Oracle Delegated Administration Services によって、新しいリソースを作成するように要求されます。

関連項目： Oracle Internet Directory へのリソースの入力の詳細は、[第 11.3.3.2 項「Oracle Internet Directory への入力」](#) を参照してください。

data_source_type は、接続するデータソースの種類です。これにより、*key_name* に関連付けられている文字列の形式が認識されます。*data_source_type* 値は、Oracle Internet Directory に格納されている有効なリソース・タイプである必要があります。Oracle Reports には、次のデータソースについてデフォルトのリソース・タイプが用意されています。

- Oracle データベース (OracleDB)
- JDBC PDS (JDBCPDS)
- Oracle OLAP Server (EXPRESSPDS)

独自のプラグابل・データソースについて、Oracle Internet Directory に追加のリソース・タイプを作成することもできます。

関連項目： リソース・タイプの追加の詳細は、[第 11.3.3.3 項「新しいリソース・タイプの追加」](#) を参照してください。

conn_string_parameter は、Oracle Reports に接続文字列を渡すために使用される Oracle Reports のシステム・パラメータまたはユーザー・パラメータを示します。たとえば、OracleDB データソースの場合、Oracle Reports は USERID パラメータを介して接続文字列を受け取り、これを使用して、指定された Oracle データベースに接続します。同様に、EXPRESSPDS の場合は EXPRESS_SERVER パラメータが使用され、JDBCPDS の場合は P_JDBCPDS が使用されます。独自のプラグابل・データソースがある場合、Oracle Reports に接続文字列を渡すための独自のユーザー・パラメータを定義して、それを SSOCONN の *conn_string_parameter* として指定する必要があります。

関連項目： [第 A.3.106 項「SSOCONN」](#)

11.3.3.1.1 Oracle データベースの例 Oracle データベースの場合、SSOCONN を使用してレポートをコールする URL は次のようになります。

```
http://myhost.mycompany.com:7779/reports/rwservlet?server=rs_cped
&report=my.rdf&destype=cache&ssocnn=mykey/OracleDB/userid&desformat=html
```

11.3.3.1.2 Oracle OLAP Server の例 Oracle OLAP Server の場合、Single Sign-On の値は次のようになります。

```
http://myhost.mycompany.com:7779/reports/rwservlet?server=rs_cped
&report=expdp.rdf&destype=cache&ssocnn=exptest1/EXPRESSPDS/express_server&desformat=html
```

11.3.3.1.3 JDBC プラuggابل・データソースの例 JDBC データソースの場合、Single Sign-On の値は次のようになります。

```
http://myhost.mycompany.com:7779/reports/rwservlet?server=rs_cped
&report=Jdbcthin.rdf&destype=cache&desformat=html&ssocnn=jd1/jdbcpds/p_jdbcpds
```

この場合、jd1 は Oracle Internet Directory のリソース名です。

関連項目： JDBC データソースの構成方法の詳細は、[第9章「JDBC PDSの構成と使用」](#)を参照してください。

使用上の注意

- SSOCONN をコマンドラインで使用する場合は、次のことが行えません。
 - 同一のコマンドラインで AUTHID を指定する。
 - セキュアでない Reports Server に対して実行する。
 - rwservlet.properties の SINGLESIGNON を NO に設定する。

SSOCONN をコマンドラインで使用する場合にこのいずれかを実行すると、エラーが発生します。

11.3.3.2 Oracle Internet Directory への入力

OracleAS Single Sign-On によるデータソースのセキュリティを機能させるには、各ユーザーのデータ接続情報を Oracle Internet Directory に格納するか、リソースをすべてのユーザーに利用可能なデフォルトのリソースにする必要があります。次のいずれかの方法で、この情報を Oracle Internet Directory に入力できます。

- [Oracle Delegated Administration Services](#)
- ユーザー・プロンプト
- バッチ・ロード
- すべてのユーザーへのリソースの公開

11.3.3.2.1 Oracle Delegated Administration Services 少数のユーザーの接続情報を入力するだけでよい場合（開発環境など）、Oracle Delegated Administration Services を使用して、ユーザーごとに接続文字列情報を直接 Oracle Internet Directory に入力できます。

注意： ユーザーが Oracle Delegated Administration Services にアクセスするには、あらかじめ管理者が各ユーザーのユーザー ID を Oracle Internet Directory に入力しておく必要があります。この手順は、他のソースの LDAP ディレクトリにすでに入力されている情報をバッチ・ロードすることで実行できます。

バッチ・ロードの詳細は、『Oracle Internet Directory 管理者ガイド』を参照してください。このマニュアルは、Oracle Application Server のドキュメントが収録された CD および Oracle Technology Network (<http://www.oracle.com/technology/index.html>) にあります。

Oracle Application Server をインストールする際に、Oracle Delegated Administration Services の場所を指定します。この URL を使用して、管理を目的として Oracle Delegated Administration Services にアクセスします。Oracle Delegated Administration Services にアクセスしたら、「プリファレンス」タブの「リソースのアクセス情報」セクションでユーザーの情報を入力します。図 11-1 を参照してください。この場合、リソースが入力されていないと「プリファレンス」タブが表示されないことに注意してください。

図 11-1 Delegated Administration Services の「プリファレンス」タブ



多数のユーザーのデータソース情報を入力しなければならない場合、ユーザー・プロンプトまたはバッチを使用して Oracle Internet Directory に入力します。

11.3.3.2.2 ユーザー・プロンプト ユーザーに自身の接続文字列情報を入力させる場合、あらかじめ Oracle Internet Directory にデータソース接続情報を入力しておく必要はありません。レポートの起動時に SSOCONN を使用する場合、Oracle Internet Directory に鍵の接続文字列が格納されておらず、その鍵がすべてのユーザーに利用可能な公開鍵でなければ、ユーザーは表示される Oracle Delegated Administration Services の「リソースの作成」ページで自身のデータソース接続文字列を入力する必要があります。図 11-2 を参照してください。Oracle Delegated Administration Services によって、後で使用するためにユーザーが入力した文字列が Oracle Internet Directory に格納され、Reports Servlet では、新しく入力された接続文字列がレポートのデータソース接続文字列として使用されます。

注意： この機能により、異なるデータソース接続文字列を使用する多くのユーザーが、同じレポートの URL を使用できるようになります。

図 11-2 Oracle Delegated Administration Services の「リソースの作成」

注意： 「リソースの作成」ダイアログで JDBC 接続文字列を入力するには、「データベース」フィールドに hostname:port:sid と入力します。

11.3.3.2.3 パッチ・ロード OracleAS Reports Services のリソースは、次のエントリで Oracle Internet Directory に作成されます。

```
orclresourcename=resource_name, cn=Resource Access Descriptor,
orclownerguid=guid, cn=Extended Properties, cn=OracleContext,
dc=us,dc=oracle,dc=com1
```

¹ dc=us,dc=oracle,dc=com は、このインスタンスでの例にすぎません。通常、これらの項目には環境に応じた値を入力します。

処理を開始する前に リソースのバッチ・ロードを開始するには、Oracle Internet Directory の前述のエントリに `orclownerguid=guid` を作成しておく必要があります。Oracle Delegated Administration Services を使用してユーザーを作成した場合は、`orclownerguid=guid` が自動的に作成されているため、**リソースのバッチ・ロード**を開始できます。

LDIF ファイルを使用して Oracle Internet Directory にユーザーを入力した場合は、**リソースのバッチ・ロード**の手順を実行する前に、次の手順を実行しておく必要があります。

1. ユーザーの GUID を取得します。

Oracle Internet Directory にユーザーを作成した方法によって、GUID を取得する方法がいくつかあります。Oracle Internet Directory LDAP API を使用してユーザーの GUID を取得できます。また、次のように `ldapsearch` コマンドを使用して取得することもできます。

```
D:¥Oracle¥BIN>ldapsearch -h host_name -p port_num -L -D cn=orcladmin
-w orcladmin's_password -b "cn=users,dc=us,dc=oracle,dc=com" -s sub "objectclass=*"
dn orclguid
```

2. `cn=Extended Properties, cn=OracleContext, dc=us, dc=oracle, dc=com` でユーザー・エントリ `orclownerguid=guid` を作成します。

- a. プレースホルダを実際の値に置き換えて、サンプル・スクリプト `ORACLE_HOME¥reports¥samples¥scripts¥createuser.ldif` を変更します。
- b. `ldapadd` を使用して `createuser.ldif` をロードします。次に例を示します。

```
D:¥Oracle¥BIN>ldapadd -D cn=orcladmin -w welcome1
-h host_name -p port_num -f createuser.ldif
```

3. `orclownerguid=guid` を作成したら、「**リソースのバッチ・ロード**」に進みます。

リソースのバッチ・ロード 次の手順に従って、ユーザーのデータソース・リソースをバッチ・ロードします。

1. `orclownerguid=guid, cn=Extended Properties, cn=OracleContext, dc=us, dc=oracle, dc=com` でユーザーのリソース・エントリ `orclresourcename=resource_name, cn=Resource Access Descriptor` を作成します。`orclownerguid=guid` は、「**処理を開始する前に**」で作成した GUID です。

- a. プレースホルダを実際の値に置き換えて、サンプル・スクリプト `ORACLE_HOME¥reports¥samples¥scripts¥createresource.ldif` を変更します。
- b. `ldapadd` を使用して `createresource.ldif` をロードします。次に例を示します。

```
D:¥Oracle¥BIN>ldapadd -D cn=orcladmin -w orcladmin's_password -h host_name -p
port_num -f createresource.ldif
```

11.3.3.2.4 すべてのユーザーへのリソースの公開 すべてのユーザーがリソースを使用できるようにする手順は次のとおりです。

1. Oracle Delegated Administration Services を起動して、「**ホーム**」タブに移動します。
2. 管理者 (orcladmin) としてログインします。
3. 「**構成**」タブをクリックします。
4. 「**プリファレンス**」サブ・タブをクリックすると、[図 11-3](#) のようなページが表示されます。

図 11-3 Oracle Internet Directory の「構成」タブの「プリファレンス」ページ



5. 「デフォルトのリソース・アクセス情報」の「作成」をクリックします。
6. 「リソースの作成」ページで、リソース名を入力し、ドロップダウン・リストからリソース・タイプを選択します。たとえば、「JDBCPSDS」を選択します。
7. 「次へ」をクリックします。
8. 接続情報を入力します。たとえば、scott/tiger@mydb と入力します。
9. 「実行」をクリックします。
10. 「OK」をクリックします。

「デフォルトのリソース・アクセス情報」にそのリソースが表示され、すべてのユーザーが使用できるようになります。

11.3.3.3 新しいリソース・タイプの追加

新しいリソース・タイプを追加して独自のプラグgable・データソースをサポートするには、次の手順を実行する必要があります。

1. Oracle Delegated Administration Services を起動して、「ホーム」タブに移動します。
2. 管理者 (orcladmin) としてログインします。
3. 「構成」タブをクリックします。
4. 「プリファレンス」サブ・タブをクリックすると、図 11-3 のようなページが表示されます。
5. 「リソース・タイプ情報の構成」で「作成」をクリックすると、図 11-4 のようなページが表示されます。

図 11-4 「リソース・タイプの作成」 ページ

6. フィールドにデータを入力します。少なくとも必須フィールドにはデータを入力してください。表 11-1 では、これらのフィールドについて説明します。

表 11-1 「リソース・タイプの作成」 プロパティ

プロパティ	説明
リソース・タイプ名	新しいリソース・タイプの名前です。この名前は、SSOCONN 文字列の <code>data_source_type</code> 部分などでリソース・タイプを参照する必要がある場合に使用されます。
表示名	ユーザー・インタフェースに表示されるリソース・タイプの名前です。
説明	リソース・タイプの説明や、そのリソース・タイプに関連するその他のテキスト情報です。
認証クラス	必須フィールドですが、OracleAS Reports Services には使用されません。このフィールド値にはダミーのテキストを入力しておきます。

表 11-1 「リソース・タイプの作成」プロパティ (続き)

プロパティ	説明
接続文字列フォーマット	<p>そのリソースに関して Oracle Internet Directory に格納されている値を使用して、OracleAS Reports Services で接続文字列がどのように構成されるかを定義します。次に例を示します。</p> <p>Oracle データベースまたは JDBC データソースの場合、接続文字列は次のような形式になります。</p> <pre>orclUserIDAttribute/orclPasswordAttribute @orclFlexAttribute1</pre> <p>この文字列は、ユーザー名の後にスラッシュ、パスワード、アット・マーク (@) および追加属性 1 (データベースの TNS 名など) が続くことを示します。この形式に従うと、接続文字列は次のようになります。</p> <pre>scott/tiger@db1</pre> <p>Oracle OLAP Server の場合、接続文字列は次のような形式になります。</p> <pre>server=orclFlexAttribute1/domain=orclFlexAttribute2/user=orclUserIDAttribute/password=orclPasswordAttribute</pre> <p>この文字列は、server= の後に最初の追加属性、スラッシュ、domain=、2 番目の追加属性、スラッシュ、ユーザー名、スラッシュおよびパスワードが続くことを示します。この形式に従うと、接続文字列は次のようになります。</p> <pre>server=a1/domain=a2/user=scott/password=tiger</pre>
ユーザー名 /ID フィールド名	<p>orclUserIDAttribute の値が格納されるユーザー名フィールドの表示名です。この表示名は、「リソースの作成」ページ (図 11-2) の orclUserIDAttribute に対応するフィールドの横に表示されます。通常、この表示名には「ユーザー名」のような値を入力します。</p>
パスワード・フィールド名	<p>orclPasswordAttribute の値が格納されるパスワード・フィールドの表示名です。この表示名は、「リソースの作成」ページ (図 11-2) の orclPasswordAttribute に対応するフィールドの横に表示されます。通常、この表示名には「パスワード」のような値を入力します。</p>
追加フィールド 1 ~ 3	<p>orclFlexAttribute1、orclFlexAttribute2 および orclFlexAttribute3 の値が格納される追加フィールドの表示名です。これらのフィールドには、ユーザー名やパスワード以外で接続文字列に必要な値を指定します。たとえば、これらのフィールドの 1 つを使用して、サーバー名やドメイン名を格納できます。この表示名は、「リソースの作成」ページ (図 11-2) の orclFlexAttribute1、orclFlexAttribute2 または orclFlexAttribute3 に対応するフィールドの横に表示されます。通常、この表示名には、「サーバー」や「ドメイン」のような、フィールドの内容を説明する値を入力します。</p>

7. 「実行」をクリックします。リソース・タイプが作成され、SSOCONN 値の data_source_type 部分で参照できるようになります。

関連項目: 第 11.3.3.1 項「SSOCONN」

11.3.4 Oracle Internet Directory への接続

第 10 章「OracleAS Reports Services のセキュリティ保護」で説明したように、OracleAS Reports Services でユーザーの権限を確認して既存のデータソース接続情報を取得するには、Oracle Internet Directory に接続する必要があります。Oracle Internet Directory に接続する場合、次の点を検討する必要があります。

- Oracle Internet Directory の接続エンティティの選択
- Oracle Internet Directory インスタンスの選択

11.3.4.1 Oracle Internet Directory の接続エンティティの選択

OracleAS Reports Services は、アプリケーション・エンティティとして Oracle Internet Directory に接続します。デフォルトでは、OracleAS Reports Services のアプリケーション・エンティティは、それぞれ Oracle Application Server のインストールに固有です。同一の Oracle Application Server のインストール先 (`ORACLE_HOME`) から起動されるすべての Reports Server は、同じアプリケーション・エンティティを使用して Oracle Internet Directory に接続します。この設定によって、Reports Server では、それぞれの Oracle Application Server のインスタンスに関連する Oracle Internet Directory の情報にのみアクセスできるようになります。

たとえば、財務グループ用と人事管理グループ用の 2 つの Oracle Application Server インスタンスがあるとします。財務グループ用の Oracle Application Server インスタンスの Reports Server では、人事管理グループのみに関連する情報にアクセスすることはできません。また、その逆についても同じです。このように、Oracle Internet Directory に格納される情報はデフォルトでより安全になりました。

OracleAS Reports Services の以前のリリースでは、すべての Reports Server が同一のアプリケーション・エンティティとして Oracle Internet Directory に接続していました。その結果、Reports Server から Oracle Internet Directory 内の情報へのアクセスを制限できませんでした。

制限が少ないセキュリティ・モードに戻す場合は、Oracle Application Server のリリース・ノート of OracleAS Reports Services に関する章を参照してください。

11.3.4.2 Oracle Internet Directory インスタンスの選択

Reports Server は、デフォルトで、Oracle Application Server とともにインストールされる Oracle Internet Directory インスタンスを使用するように構成されます。システムを新しく構築する場合は、この構成が適切です。しかし、既存の Oracle Internet Directory インスタンスを Reports Server で使用する場合は、構成をいくつか調整する必要があります。

ただし、Oracle Internet Directory インスタンスの変更は、Oracle Application Server の中間層を完全に変更する処理の一部として実行する必要があります。このプロセスの詳細は、『Oracle Application Server 管理者ガイド』のアプリケーション・サーバー・インスタンスの再構成に関する章を参照してください。

11.4 Oracle Internet Directory の接続エンティティの選択

複数のアプリケーション・エンティティをマージすると、異なる `ORACLE_HOME` にインストールされた Reports Server が利用可能な `SSOCONN` リソースを共有できるようになります。マージを行うには、`ldapmodify` コマンドを使用して LDIF ファイルを実行する必要があります。LDIF ファイルの内容は次のとおりです。

```
dn: dn of the group representing the logical grouping of all report instances
changetype: modify
add: uniquemember
uniquemember: dn of the Reports Application Entity
```

ここでは

```
dn of the group representing the logical grouping of all report instances is
cn=Virtual Application Group, orclApplicationCommonName=reports_application_
entity_name**, cn=Reports, cn=Products, cn=OracleContext
```

```
dn of the Reports Application Entity is
orclApplicationCommonName=reports_application_entity_name**, cn=Reports, cn=Products,
cn=OracleContext
```

** reports_application_entity_name is in the format reportsApp_hostname_GUID. For example, reportsApp_serv1.us.oracle.com_C7543D42A9E26726E034080020A46EE

例

LDIF ファイルのエントリ :

```
dn: cn=Virtual Application Group,
    orclApplicationCommonName=reportsApp_Group.us.oracle.com_C7543D42A9E26726E0340
    80020A46EE2, cn=Reports, cn=Products, cn=OracleContext
changetype: modify
add: uniquemember
uniquemember:
    orclApplicationCommonName=reportsApp_serv1.us.oracle.com_A8654E53B0F37837F1451
    91131B57FF3, cn=Reports, cn=Products, cn=OracleContext
```

コマンドラインの対応する ldapmodify コマンド

```
ldapmodify -D cn=orcladmin -w welcome1 -h reportsApp_Group.us.oracle.com -p 389 -f
mergeentity.ldif
```

11.5 OracleAS Forms Services のセキュリティに関する考慮事項

Oracle Application Server Forms Services のデフォルトの構成は、OracleAS Single Sign-On (SSO) モードでは稼働しません。一方、OracleAS Reports Services のデフォルトの構成は、SSO モードで稼働します。

OracleAS Forms Services が非 SSO モードで稼働し、OracleAS Reports Services が SSO モードで稼働していても、Reports Server と要求されたレポートが OracleAS Portal に登録されていないかぎり、RUN_REPORT_OBJECT ビルトイン・プロシージャを使用して、統合された OracleAS Reports Services をコールする Forms アプリケーションで問題が発生することはありません。

その他の要件 :

- プロパティ Reports Server は、Oracle Forms モジュールのすべてのレポート・オブジェクトに対して明示的に設定する必要があります。
- デフォルト以外の Reports Server を使用する場合、そのサーバーはコマンドラインから次のように起動する必要があります。
`rwsrver server=server_name`
- システム変数 REPORTS_PATH は、ファイル `ORACLE_HOME/bin/reports.sh` で、実行されるレポートのパスを参照するように変更する必要があります。
- Reports Server を初めて起動すると、`ORACLE_HOME/server/conf` ディレクトリに `server_name.conf` という構成ファイルが作成されます。
- Reports Server のデフォルトのステータスはセキュアです。Reports Server のステータスを非セキュアに変更するには、`<security>` タグをコメントアウトし、`<job>` タグから `securityId` を削除して、`ORACLE_HOME/server/conf/reports_server_name.conf` を変更します。
- これらの変更が終了したら、Reports Server を停止して再起動する必要があります。
- OracleAS Forms Services が SSO モードで稼働するように構成されている場合、レポート・リクエストは、SSO ユーザー・ログインに基づいて、指定されている `authid` を使用して送信されます。
- 保護されたレポートと Reports Server は、OracleAS Portal に登録できます。

表 11-2 は、考えられる Forms と Reports の組合せと想定される結果のリストです。

表 11-2 Forms が SSO モードまたは非 SSO モードで稼動する場合の Forms と Reports の統合結果

レポートのタイプ	登録されたセキュアな Reports Server (登録済レポートのみ実行)	登録されたセキュアな Reports Server (任意のレポートを実行)	セキュアでない Reports Server
パブリック・アクセスをもつレポート	生成される	生成される	生成される
特定のユーザー・アクセスをもつレポート	生成される	生成される	生成される
特定のユーザー・アクセスをもたないレポート	生成されない	生成されない	生成される
登録されていないレポート	生成されない	生成されない	生成される

OracleAS Portal でのレポートの配布

この章では、Oracle Application Server Reports Services のレポートを OracleAS Portal で配布する方法を説明します。この章には、次の項があります。

- [Reports ユーザーおよび名前付きグループの作成](#)
- [Oracle Reports コンポーネントの登録](#)
- [レポートのポートレットとしての公開](#)
- [トラブルシューティング情報](#)

レポートを配布するには、事前に OracleAS Portal と OracleAS Reports Services の両方がインストールされ、構成されている必要があります。

関連項目： 詳細は、次の内容を参照してください。

- [第3章「OracleAS Reports Services の構成」](#)。OracleAS Reports Services の構成について説明しています。
- 『Oracle Application Server Portal 構成ガイド』。OracleAS Portal の構成について説明しています。
- Oracle Application Server のインストール・ガイド。両方のコンポーネントのインストールについて説明しています。
- Oracle Application Server のドキュメントが収録された CD。
- Oracle Technology Network
(<http://www.oracle.com/technology/index.html>)

12.1 Reports ユーザーおよび名前付きグループの作成

OracleAS Portal のセキュリティ機能を使用してレポートへのアクセスを制御する場合は、すべての Reports ユーザーを Oracle Internet Directory に登録し、すべてのユーザーに OracleAS Portal を通じてセキュリティ権限を割り当てる必要があります。

注意： LDAP 互換ディレクトリにすでに多数のユーザーを入力している場合は、Oracle Internet Directory の機能を使用してディレクトリ間の同期をとることにより、ユーザーを 1 人ずつ入力する手間が省けます。Oracle Internet Directory の Directory Integration Server の詳細は、『Oracle Internet Directory 管理者ガイド』を参照してください。

OracleAS Portal では、個々のユーザーおよびユーザーの名前付きグループにセキュリティ権限を付与できます。名前付きグループは、アクセス権限を付与する処理を簡素化するために役立ちます。名前付きグループにアクセス権限のセットを割り当てておけば、グループに個別のユーザーを追加するだけで、そのユーザーに権限のセット全体を付与することができます。

注意： OracleAS Portal Security、Portal Destination、Job Status Repository などの機能を使用すると、OracleAS Reports Services によって確立された JDBC データベース接続が NLS_LANG の初期設定よりも優先されることがあります。これにより、PDF での双方向出力などの、実行中のレポートの動作に影響が及ぶ場合があります。UNIX プラットフォームでは、環境切替え機能を使用して、レポートの環境を動的に設定することで、この問題を解決できます。詳細は、[第 3.2.2 項「動的な環境切替え」](#)を参照してください。

次の各項では、OracleAS Portal でユーザーとグループを作成する方法について概説します。関連する項は、次のとおりです。

- [デフォルトの Reports 関連グループ](#)
- [ユーザーとグループの作成](#)

12.1.1 デフォルトの Reports 関連グループ

OracleAS Portal をインストールすると、Reports 関連グループが自動的に作成されます。これには次のグループが含まれます。

- [RW_BASIC_USER](#)
- [RW_POWER_USER](#)
- [RW_DEVELOPER](#)
- [RW_ADMINISTRATOR](#)

OracleAS Portal を通じて、グループのメンバーが必要な任意の機能をレポートで実行できるようにするには、これらのグループに適切な権限を割り当てる必要があります。たとえば、あるグループ (RW_BASIC_USER など) のメンバーに実行させたいレポート・オブジェクトがある場合は、個々のレポート・オブジェクトの「アクセス」タブから、そのグループに対して実行権限を付与する必要があります。同様に、あるグループ (RW_ADMINISTRATOR など) のメンバーが、Reports Server、プリンタおよびレポートを管理できるようにするには、それらのオブジェクトの「アクセス」タブから、そのグループに対して管理権限を付与する必要があります。

個々のユーザーにオブジェクト権限を割り当てることも可能ですが、レポートにアクセスするすべてのユーザーは、これらのグループまたは独自に作成したグループに所属させることをお勧めします。これらのグループのメンバーでないユーザーがレポートを実行しようとすると、デフォルトでそれらのユーザーに基本ユーザーの権限が割り当てられます。

注意： Web コマンド (getjobid、getserverinfo、showjobs、showenv など) は、RW_groups に含まれている場合、使用することができます。RW_groups は、OracleAS Portal を構成すると自動的に作成されますが、手動で作成することもできます。Oracle Reports 10g リリース 2 (10.1.2) では、Web コマンドが IASADMINS グループに含まれている場合も、その Web コマンドを実行できます。

12.1.1.1 RW_BASIC_USER

セキュリティ・チェックが失敗した場合、RW_BASIC_USER のユーザーには、他の Oracle Reports グループのユーザーに比べて簡単なエラー・メッセージが表示されます。たとえば、次のようなメッセージです。

```
Security Check Error
```

通常、このグループには最小限の権限を割り当てます。たとえば、レポートを実行する権限のみを付与します。

12.1.1.2 RW_POWER_USER

RW_BASIC_USER グループの権限よりも多くの権限が付与された RW_POWER_USER グループは、基本ユーザーよりも詳細なエラー・メッセージを受け取ります。たとえば、HTML に対する実行が許可されていない場合にそれを実行しようとすると、次のようなメッセージが表示されます。

```
Cannot run report to HTML
```

これは、同じエラーに関して RW_BASIC_USER が受け取るメッセージより詳しく記述されています。

12.1.1.3 RW_DEVELOPER

RW_DEVELOPER グループには、RW_POWER_USER グループの権限よりも多くの権限が付与され、システム環境を表示する SHOWENV や SHOWMAP などの Web コマンドを実行できます。

通常、このグループには、レポートをテストする開発者に必要な権限を割り当てます。インストールによっては、制限付きの管理権限を割り当てることもできます。

12.1.1.4 RW_ADMINISTRATOR

これらのユーザーには、RW_DEVELOPER の権限よりも多くの権限が付与され、Oracle Reports Queue Manager の管理者の機能に対するアクセス権限も与えられます。つまり、サーバー内のジョブの再スケジュール、削除、順序変更、およびサーバーの停止を含めた、サーバー・キューの管理を実行できます。また、RW_ADMINISTRATOR は、Reports Servlet を使用して Web コマンドを実行する権限も持ちます。

通常、このグループには、PORTAL_ADMINISTRATORS グループに割り当てられる権限の一部を割り当てます。

注意： 最初は、PORTAL_ADMINISTRATORS グループのメンバーのみが Oracle Reports オブジェクトの管理権限を持ちます。このグループのメンバーは、OracleAS Portal に登録されているレポート定義ファイル、サーバー、プリンタの各オブジェクトを作成、更新および削除できます。開発ユーザー向けのすべてのリンクに加えて、管理者は OracleAS Portal でアクセス可能なコンポーネントの管理ページの「アクセス」タブにナビゲートできます。ここで管理者は、特定のレポートにアクセスできるユーザーを指定できます。管理者権限を持つユーザーは、他のユーザーにセキュリティ権限を割り当てたり、OracleAS Reports Services から全てのエラー・メッセージを受け取ることができます。

12.1.2 ユーザーとグループの作成

OracleAS Portal は、Oracle Internet Directory の Delegated Administration Services (DAS) インタフェースを使用して、Portal にアクセスできるようにユーザーを登録します。Portal を通じて、DAS インタフェースから新しいユーザーを作成することができます。ユーザーとグループの新規作成の詳細は、Oracle Application Server のドキュメントが収録されている CD 内の『Oracle Application Server Portal 構成ガイド』を参照してください。

グループを作成する際には、グループのメンバーが OracleAS Portal を通じて必要な任意の機能をレポートで実行できるように、グループに適切な権限を割り当てる必要があります。たとえば、あるグループ (RW_BASIC_USER など) のメンバーに実行させたいレポート・オブジェクトがある場合は、個々のレポート・オブジェクトの「アクセス」タブから、そのグループに対して実行権限を付与する必要があります。同様に、あるグループ (RW_ADMINISTRATOR など) のメンバーが、Reports Server、プリンタ、カレンダーおよびレポートを管理できるようにするには、それらのオブジェクトの「アクセス」タブから、そのグループに対して管理権限を付与する必要があります。

オブジェクトに関して必要な権限をユーザーに付与するには、そのロールに適した権限を持つグループにユーザーを割り当てるのが理想的です。たとえば、レポートを実行するだけで管理する必要のないユーザーを作成する場合は、このユーザーを RW_BASIC_USER に割り当てます。必要に応じて、グループのかわりに個々のユーザー (JSMITH など) にオブジェクト権限を割り当てることもできますが、このアプローチは管理が複雑になり、時間がかかります。

12.2 Oracle Reports コンポーネントの登録

ポートレットにアクセスして、アクセス制御の設定に必要なタスクを実行するには、事前に OracleAS Portal で十分なレベルの権限を割り当てられていることが必要です。OracleAS Portal でレポートを管理するには、PORTAL_ADMINISTRATORS グループと RW_ADMINISTRATOR グループの両方に属している必要があります。RW_ADMINISTRATOR にのみ属しているユーザーがレポート・オブジェクトを作成しようとすると、エラーが発生します。

OracleAS Portal の権限グループへの参加の詳細は、『Oracle Application Server Portal 構成ガイド』を参照してください。

この項では、次の操作に必要な手順について説明します。

- [Reports Server の登録](#)
- [レポートの登録](#)
- [プリンタの登録](#)
- [使用可能日カレンダーの作成](#)

既存の OracleAS Portal ポートレットで操作を実行する詳細は、次の項を参照してください。

- [「管理」ポートレット](#)

12.2.1 Reports Server の登録

Reports Server に対するアクセス制御を定義するには、事前に OracleAS Portal 内でサーバーを登録する必要があります。登録によって、利用可能なすべての Reports Server を特定して検索するために必要な情報が、OracleAS Portal に提供されます。これは、個々のレポートを登録する際に特に重要となります。レポートの登録では、Reports Server のリストから選択する必要がありますが、登録されているサーバーのみがこのリストに表示されるためです。

表 12-1 サンプル値

プロパティ	サンプル値
名前 (内部名)	myrep_server
表示名	My Reports Server
Portal DB プロバイダ	PORTAL_APP

表 12-1 サンプル値 (続き)

プロパティ	サンプル値
Reports Server 名	rep_machine_name (rep_myserver1 など)
JSP レポート用の Oracle Reports Web ゲートウェイ URL	http://myias.mycomp.com:7778/
RDF レポート用の Oracle Reports Web ゲートウェイ URL	http://myias.mycomp.com:7778/reports/rwservlet
使用可能日カレンダー	COMCAL

Reports Server を登録する手順は次のとおりです。

1. OracleAS Portal に管理者としてログインします。
2. 「ビルダー」 ページにナビゲートします。
3. 「管理」 タブをクリックします。
4. 「Oracle Reports セキュリティ」 ポートレットの「Oracle Reports セキュリティの設定」 リンクをクリックします。「Oracle Reports セキュリティ」 ポートレットを使用すると、サーバー、プリンタ、カレンダーおよびレポート定義ファイルへのアクセスを定義する際に、OracleAS Portal のセキュリティ機能を利用できます。
5. 「レポート・サーバー・アクセス」 ポートレットの「レポート・サーバー・アクセスの作成」 リンクをクリックします。
6. 表示されるページでは、「名前」 (内部名) フィールドと「Portal DB プロバイダ」 フィールドにデフォルト値が入力されています。カスタムの値を入力する手順は次のとおりです。
 - 「名前」 フィールドに、Reports Server を OracleAS Portal 内部で一意に識別する名前 (MY_REPORTS_SERVER など) を入力します。この名前は、有効なコンポーネント名に関する次のような OracleAS Portal のルールに従っている必要があります。
 - * 30 文字以下で指定する。
 - * 英数字のみ使用する (スペースおよび特殊文字は使用できません)。
 - * 名前の最初は必ず文字で始める (数字は不可)。
 - 「表示名」 フィールドに、このサーバーの表示名を入力します。この名前は、OracleAS Portal を通じてユーザーに公開されます。

注意： 内部名とは異なり、表示名にはスペースを含めることができます。

- 「Portal DB プロバイダ」 値リストから、Reports Server の所有者となるポータル DB プロバイダを選択します。このリストには、コンポーネントを作成する権限がある場合に使用できる Portal DB プロバイダが表示されます。

注意： OracleAS Portal で追加または作成するコンポーネントは、すべてポータル DB プロバイダに属している必要があります。ポータル DB プロバイダの作成方法の詳細は、OracleAS Portal のオンライン・ヘルプを参照してください。

7. 「次へ」 をクリックします。

8. サーバーの定義 ページで、次の作業を行います。

- 「**レポート・サーバー名**」フィールドに、Reports Server の名前を入力します。これは、`rwserver -install repservername` または `rwserver server=repservername` など、インストール時にサーバーに割り当てた一意の名前です。
- (オプション) 「**説明**」フィールドに Reports Server の説明を入力します。
- 「**JSP レポート用の Oracle Reports Web ゲートウェイ URL**」フィールドに、JSP ファイルの位置を示す URL を入力します。URL 構文は、次のような形式になります。

```
http://your_web_server.domain:port/
```

次に例を示します。

```
http://myias.mycomp.com:7779/
```

- 「**RDF レポート用の Oracle Reports Web ゲートウェイ URL**」フィールドに、Reports Servlet の位置を示す URL を入力します。URL 構文は、次のような形式になります。

```
http://your_web_server.domain:port/virtual_path_to_rwservlet/rwservlet
```

関連項目： 仮想パスの指定の詳細は、[第3章「OracleAS Reports Services の構成」](#)を参照してください。

次に例を示します。

```
http://myias.mycomp.com:7778/reports/rwservlet
```

- (オプション) 「**登録されたレポート定義ファイルだけを実行**」チェック・ボックスを選択します。これにより、OracleAS Portal に登録されているレポート定義ファイルのみがこの Reports Server で実行されるようになります。
このボックスの選択を解除したままにすると、この Reports Server は、OracleAS Portal に登録されていないものも含めたあらゆるレポート定義ファイルを受け付けます。ただし、レポート・リクエストを送信するユーザーが、この Reports Server に対するアクセス権を持っている場合に限りです。
- 「**プリンタ**」リストから、この Reports Server で使用できるプリンタを選択します。複数のプリンタを選択するには、**[Ctrl]** を押しながらクリックするか (Windows の場合)、クリックします (UNIX の場合)。

9. 「次へ」 をクリックします。

10. (オプション) カスタムの宛先タイプを定義した場合は、「**カスタム宛先タイプ**」に値を入力します。

関連項目： カスタムの宛先タイプの詳細は、[第8章「OracleAS Reports Services の宛先の構成」](#)を参照してください。

11. 「次へ」 をクリックします。

12. (オプション) この Reports Server でレポート・リクエストを受け付けることができる日時と受け付けることができない日時を指定する使用可能日カレンダー名を入力するか、リスト・ボタンをクリックして使用可能日カレンダーを選択します。

関連項目： [第12.2.4項「使用可能日カレンダーの作成」](#)

13. 「完了」 をクリックします。

この Reports Server に関する設定を要約したページが表示されます。このページでは、設定を編集できます。また、Reports Server に関する詳細な登録情報を確認したり、設定を完全に削除することができます。

関連項目：「開発」タブ、「管理」タブおよび「アクセス」タブなど、「管理」ポートレットに表示されるフィールドや説明の詳細は、[第 12.2.5 項「管理」ポートレット](#)を参照してください。

14. 「閉じる」ボタンをクリックしてこのページを閉じ、「Oracle Reports セキュリティ」ページに戻ります。

これで、Reports Server の登録が完了しました。次は、レポートを登録します。

12.2.2 レポートの登録

レポートの登録は必須のステップです。この登録によって、レポートを実行できるユーザー、レポートの実行時間、レポート・リクエストの処理に使用するサーバー、レポートの配布方法、およびレポートの送信先プリンタを定義できます。

登録では、レポートにアクセスできるユーザーを指定するほかに、OracleAS Portal パラメータ・フォームを通じて、ユーザーとレポートの対話方法を指定することもできます。

ユーザー・パラメータは、レポートの設計時に Reports Builder で作成します。OracleAS Portal でレポートを実行する際に、これらのパラメータに値を割り当てることができます。

注意： OracleAS Portal を通じて利用できるパラメータ設定を使用して、設計時に Reports Builder で定義したパラメータのサブセットを複製または作成できます。設計時に Reports Builder で定義されなかったパラメータは、OracleAS Portal で設定しても、実行時に Reports Server によって無視されます。

OracleAS Portal 内でレポートを登録すると、OracleAS Portal コンポーネントが作成されます。これは、Portal を通じてポートレットとして展開できます。OracleAS Portal では、レポート・ファイルのインスタンスを 1 つだけ登録することをお勧めします。1 つのレポートに複数の OracleAS Portal レポート・オブジェクトを定義すると、実行時にそのすべてのオブジェクトがセキュリティ・チェックを受けます。このうち 1 つでもセキュリティ・チェックに合格しなかった場合は、すべてのオブジェクトが不合格となり、ジョブは実行されません。

注意： OracleAS Portal でレポートを実行するには、HTML の iframe タグを使用する必要があります。このタグは、Netscape 4.x ではサポートされていません。このため、Netscape 4.x では次のような制限があります。

- HTTPS を使用している場合は、レポート・ポートレットは適切な場所に表示されません。別のブラウザ・ウィンドウにレポートを表示するには、ポートレットのタイトルをクリックする必要があります。
 - HTTPS を使用している場合、**カスタマイズ**・リンクを介してレポート・ポートレットを実行するようにスケジュールリングできません。
-

表 12-2 サンプル値

プロパティ	サンプル値
名前 (内部名)	Employee_Report
表示名	Employee Report
Portal DB プロバイダ	PORTAL_APP
Oracle Reports ファイル名	employee_report.jsp
実行	JSP として
名前 (オプションのパラメータ)	userid
表示名 (オプションのパラメータ)	User Identification

レポートを登録する手順は次のとおりです。

1. OracleAS Portal に管理者としてログインします。
2. 「ビルダー」 ページにナビゲートします。
3. 「管理」 タブをクリックします。
4. 「Oracle Reports セキュリティ」 ポートレットの「Oracle Reports セキュリティの設定」 リンクをクリックします。
5. 「レポート定義ファイル・アクセス」 ポートレットの「レポート定義ファイル・アクセスの作成」 をクリックします。
6. 表示されるページでは、「名前」 (内部名) フィールドと「Portal DB プロバイダ」 フィールドにデフォルト値が入力されています。カスタムの値を入力する手順は次のとおりです。
 - 「名前」 フィールドに、レポートを OracleAS Portal 内部で一意に識別する名前 (MY_REPORT など) を入力します。この名前は、有効なコンポーネント名に関する次のような OracleAS Portal のルールに従っている必要があります。
 - * 30 文字以下で指定する。
 - * 英数字のみ使用する (スペースおよび特殊文字は使用できません)。
 - * 名前の最初は必ず文字で始める (数字は不可)。
 - 「表示名」 フィールドに、このレポートの表示名を入力します。この名前は、OracleAS Portal を通じてユーザーに公開されます。

注意： 内部名とは異なり、表示名にはスペースを含めることができます。

- 「Portal DB プロバイダ」 値リストから、Reports Server の所有者となるポータル DB プロバイダを選択します。このリストには、コンポーネントを作成する権限がある場合に使用できる Portal DB プロバイダが表示されます。

注意： OracleAS Portal で追加または作成するコンポーネントは、すべてポータル DB プロバイダに属している必要があります。ポータル DB プロバイダの作成方法の詳細は、OracleAS Portal のオンライン・ヘルプを参照してください。

7. 「次へ」 をクリックします。
8. 次のように情報を入力または選択します。
 - 「Reports Server」 値リストから、このレポートの実行に使用できる Reports Server を選択します。複数のサーバーを選択するには、[Ctrl] を押しながらクリックするか (Windows の場合)、クリックします (UNIX の場合)。
 - 「Oracle Reports ファイル名」 フィールドに、レポート・ファイル名を、その拡張子も含めて入力します。

レポート定義ファイルは、.rdf ファイル、.jsp ファイル、.xml ファイルのいずれかです。このファイルへのパスが REPORTS_PATH 環境変数に含まれている場合は、ここでパスを入力しないでください。パスが REPORTS_PATH 環境変数に含まれていない場合は、ここでファイル名とともにパスを入力します。この作業は、スタンドアロンの JSP として実行するものを除くすべてのレポート定義ファイルに対して実行します。JSP の場合は、`virtual_path/reportname.jsp` のように名前を定義する必要があります。

関連項目：

- Oracle Reports 関連の環境変数の詳細は、付録 B 「環境変数」 を参照してください。
- 仮想パスの指定の詳細は、第 3 章 「OracleAS Reports Services の構成」 を参照してください。

- (オプション) 「説明」 フィールドにこのレポートの説明を入力します。
- 「実行」 フィールドで、「サーブレットを使用」または「JSP として」のいずれかを選択します。ここでの選択は、次のウィザード・ページの選択肢に影響します。
 - * **サーブレットを使用** : Reports Servlet を介してレポートを実行する場合に選択します。
 - * **JSP として** : 配布された JSP レポートを実行する場合に選択します。

9. 「次へ」をクリックします。

10. 「必須パラメータ」 ページで宛先の設定を選択します。これらの設定は、Reports Servlet を介してレポートを実行する場合にのみ適用されます。[Ctrl] を押しながらかlickして複数選択を指定した箇所では、実行時に値リストがユーザーに表示され、その中からユーザーが各自のランタイム情報を選択できるようになります。

- 「タイプ」では、このレポートで受け付ける宛先タイプを指定します。キャッシュ、ファイル、メール、Oracle Portal、Oracle Wireless、プリンタ、FTP、WebDAV またはカスタム宛先タイプの中から宛先タイプを選択します。OracleAS Portal で Reports Server を登録するときに指定したカスタム宛先タイプを、このレポートに関連付けるサーバーがサポートする場合は、指定したタイプがこのリストに表示されます。
- 「フォーマット」では、このレポートで受け付ける出力フォーマットを定義します。HTML、HTMLCSS、PDF、XML、RTF、デリミタ付き、スプレッドシート、PostScript、および文字の中から選択します。
- 「プリンタ」では、このレポートを送信できる登録済プリンタを指定します。このリストに表示されるプリンタは、このレポートに関連付ける Reports Server へのアクセスを設定する際に選択したプリンタによって決まります。ユーザーがランタイム・パラメータ・フォームで Reports Server を選択する場合は、選択した Reports Server に関連付けられているプリンタおよびそのユーザーがアクセスできるプリンタのみが表示されます。

11. 「パラメータ・フォーム・テンプレート」を選択して「テンプレートのプレビュー」をクリックし、選択したテンプレートの外観を確認します。

- 「パラメータ・フォーム・テンプレート」では、レポートを実行する際に使用する Portal パラメータ・フォームのロック・アンド・フィールドを定義するテンプレートを指定します。この値は、レポートが Portal を通じて公開される場合のみ使用されます。テンプレートは値リストから選択します。

注意： このリストに独自のテンプレートを追加する方法の詳細は、OracleAS Portal のオンライン・ヘルプを参照してください。

12. 「次へ」をクリックします。

13. 「オプションのパラメータ」 ページで、レポートの既存のパラメータに対する制限を定義します。

- 「名前」 フィールドに、ユーザーが指定できる値を制限するシステム・パラメータまたはユーザー・パラメータの名前を入力します。たとえば、SALES_REGION や COPIES などを入力します。
- システム・パラメータまたはユーザー・パラメータの表示名を入力します。この名前は、ランタイム・パラメータ・フォームでパラメータを識別するために使用されます。

- 値リストの名前を入力するか、または事前に定義された値リストから値を選択します。このリストは事前に作成しておく必要があります。値リストの作成の詳細は、OracleAS Portal のオンライン・ヘルプを参照してください。
 - 「下限」フィールドに、値の範囲に対して設定する下限値を入力します。
 - 「上限」フィールドに、値の範囲に対して設定する上限値を入力します。
 - パラメータと値を追加するために行を増やす場合は、「**パラメータ追加**」をクリックします。
14. 「次へ」をクリックします。
15. (オプション) 使用可能日カレンダー名を入力するかリスト・ボタンをクリックして、既存の使用可能日カレンダーを選択します。
- 使用可能日カレンダーを使用して、このレポートを実行できる日時を制限します。

関連項目： 第 12.2.4 項「使用可能日カレンダーの作成」

16. 「次へ」をクリックします。
17. (オプション) 有効化トリガーを入力してプログラムによる制限を作成します。
- 有効化トリガーは、条件付き制限を作成するために使用されます。条件付き制限とは、「**必須パラメータ**」ページにも「**オプションのパラメータ**」ページにも定義できない制限です。有効化トリガーは PL/SQL ファンクションです。
- 有効化トリガーとして指定するファンクションは、ブール値 (TRUE または FALSE) を返す必要があります。ファンクションが TRUE を返した場合は、ジョブが実行されます。ファンクションが FALSE を返した場合は、エラー・メッセージが表示され、ジョブは実行されません。
18. 「終了」をクリックして、ウィザードを閉じ、レポートの登録を完了します。
- 結果のページには、登録情報の要約が表示されます。このページで、レポートに追加操作を実行するかどうかを選択できます。

関連項目： OracleAS Portal からレポートを実行する方法の詳細は、第 12.3 項「レポートのポートレットとしての公開」を参照してください。

19. 「アクセス」タブをクリックして、「**ポートレットとして公開**」を選択します。これにより、レポートがポートレット・リポジトリに追加されるので、このレポートをページに追加してポートレットとして公開できます。
20. レポートのランタイム・パラメータ・フォームを表示するには、「**カスタマイズ**」をクリックします。

表 12-3 では、このページで指定可能なオプションをまとめています。

表 12-3 ランタイム・パラメータ・フォームのオプション

オプション	説明
レポート実行	指定したパラメータ値でこのレポートを実行する場合にクリックします。
パラメータの保存	パラメータ値の選択内容を保存する場合にクリックします。
サーバー	このレポート・リクエストを受け取る Oracle Reports Server を選択します。レポートの登録時に選択したサーバーのみが、このリスト・ボックスに表示されます。
プリンタ	レポート出力を印刷するプリンタを選択します。レポートの登録時に選択したプリンタのみが、このリスト・ボックスに表示されます。
Destype	宛先タイプを選択します。レポートの登録時に選択した宛先タイプのみが、このリスト・ボックスに表示されます。
Desformat	宛先フォーマットを選択します。レポートの登録時に選択した宛先フォーマットのみが、このリスト・ボックスに表示されます。

表 12-3 ランタイム・パラメータ・フォームのオプション (続き)

オプション	説明
Desname	DESTYPE が FILE の場合は出力ファイルの名前を入力し、DESTYPE が MAIL の場合は電子メール・アドレスを入力します。複数のアドレスはカンマで区切ります。宛先名は、DESTYPE に FILE または MAIL を選択した場合に必要となります。
SSOCONN	1 つ以上の SSO 接続文字列を入力します。複数の文字列は、スペースではなくカンマで区切ります。SSOCONN の詳細は、 第 11.3.3.1 項「SSOCONN」 を参照してください。
ユーザーに表示	ユーザーがこのレポート・リクエストを実行するときにランタイム・パラメータ・フォームに表示する各パラメータをチェックします。ボックスをチェックしないパラメータは、ユーザーに表示されません。
CGI/Servlet コマンド・キー	オプションで、このレポートに対して実行するコマンドラインを識別する cgicmd.dat ファイルのキーを入力します。
ポートレットの幅	このフィールドは、ポートレットの幅を制御するために使用します。ページの割合 (90% など) またはピクセル (700 など) で値を入力できます。 値を指定しない場合、OracleAS Reports Services ではデフォルト値 (幅が 640 ピクセル) が使用されます。
ポートレットの高さ	このフィールドは、ポートレットの高さを制御するために使用します。ページの割合 (50% など) またはピクセル (400 など) で値を入力できます。 値を指定しない場合、OracleAS Reports Services ではデフォルト値 (高さが 320 ピクセル) が使用されます。
追加されたユーザー・パラメータ	このフィールドは、追加のユーザー・パラメータを入力するために使用します。たとえば、このフィールドを使用して、このレポートの配布方法を定義する配布 XML ファイルのパスと名前を入力します。 その場合、コマンドライン・リクエストまたは cgicmd.dat ファイル内でこれらの値を指定する際の構文を使用します。複数の追加パラメータを入力する場合は、各エントリをスペースで区切るだけです。 配布 XML ファイルの詳細は、 第 15 章「拡張配布の作成」 を参照してください。

12.2.3 プリンタの登録

OracleAS Portal のセキュリティ・フレームワークの中では、プリンタを登録する必要はありません。レポートは、Reports Server で利用できるすべてのプリンタで実行できます。ただし場合によっては、OracleAS Portal ユーザーをそれらのプリンタのサブセットに制限したり、一定期間プリンタの使用を制限したり、特定のレポートの出力を印刷する特別なプリンタを識別することが必要になります。

OracleAS Portal へのプリンタの登録は、OracleAS Portal を通じて実行するレポートだけでなく、スタンドアロンの URL から実行するレポートにも意味があります。

OracleAS Portal 内に登録したプリンタは、Reports Server に関連付けることができます。多数のプリンタを登録できます。ただし、OracleAS Portal にレポートを登録し、特定の Reports Server を選択する場合は、それらの Reports Server に関連付けられているプリンタのみを印刷に使用できます。

登録されたレポートを送信できるプリンタの登録済サブセットを、さらに制限することができます。たとえば、Reports Server が最高経営責任者のオフィスのプリンタに接続されている場合は、最高経営責任者以外の従業員が総勘定元帳レポートを実行する際に、そのプリンタを選択できないようにする必要があります。レポート・リクエストを実行する OracleAS Portal ユーザーに対してプリンタのサブセットを提示し、出力の送信先を選択させることができます。

表 12-4 サンプル値

プロパティ	サンプル値
名前 (内部名)	myrep_printer
表示名	My Reports Printer
Portal DB プロバイダ	PORTAL_APP
OS プリンタ名	¥¥mydomain¥printer1
使用可能日カレンダー	COMCAL

プリンタを登録する手順は次のとおりです。

1. OracleAS Portal に管理者としてログインします。
2. 「ビルダー」 ページにナビゲートします。
3. 「管理」 タブをクリックします。
4. 「Oracle Reports セキュリティ」 ポートレットの「Oracle Reports セキュリティの設定」 リンクをクリックします。「Oracle Reports セキュリティ」 ポートレットを使用すると、サーバー、プリンタ、カレンダーおよびレポート定義ファイルへのアクセスを定義する際に、OracleAS Portal のセキュリティ機能を利用できます。
5. 「レポート・プリンタ・アクセス」 ポートレットの「レポート・プリンタ・アクセスの作成」 リンクをクリックします。
6. 表示されるページでは、「名前」 (内部名) フィールドと「Portal DB プロバイダ」 フィールドにデフォルト値が入力されています。カスタムの値を入力する手順は次のとおりです。
 - 「名前」 フィールドに、プリンタを OracleAS Portal 内部で一意に識別する名前 (MY_PRINTER など) を入力します。この名前は、有効なコンポーネント名に関する次のような OracleAS Portal のルールに従っている必要があります。
 - * 30 文字以下で指定する。
 - * 英数字のみ使用する (スペースおよび特殊文字は使用できません)。
 - * 名前の最初は必ず文字で始める (数字は不可)。
 - 「表示名」 フィールドに、このプリンタの表示名を入力します。この名前は、OracleAS Portal を通じてユーザーに公開されます。

注意： 内部名とは異なり、表示名にはスペースを含めることができます。

- 「Portal DB プロバイダ」 値リストから、プリンタの所有者となるポータル DB プロバイダを選択します。このリストには、コンポーネントを作成する権限がある場合に使用できる Portal DB プロバイダが表示されます。

注意： OracleAS Portal で追加または作成するコンポーネントは、すべてポータル DB プロバイダに属している必要があります。ポータル DB プロバイダの作成方法の詳細は、OracleAS Portal のオンライン・ヘルプを参照してください。

7. 「次へ」 をクリックします。

8. 表示されたページで、必要な値を入力します。

- 「OS プリンタ名」フィールドに、オペレーティング・システムのプリンタ名を入力します。次に例を示します。

UNIX: `printer_name`

Windows: `¥¥printer_server¥¥printer_name` (for a remote printer)
`printer_name` (for a local printer)

このプリンタは、Reports Server で利用できることが必要です。

注意： プリンタの可用性は、Reports Server のホスト・マシンからオペレーティング・システムを通じて設定します。

- (オプション)「説明」フィールドにプリンタの説明を入力します。

9. 「次へ」をクリックします。

10. (オプション) 使用可能日カレンダーを選択してプリンタが使用可能な日時を制限します。

関連項目： [第 12.2.4 項「使用可能日カレンダーの作成」](#)

11. 「完了」をクリックします。

このプリンタに関する設定を要約したページが表示されます。このページでは、設定を編集できます。また、プリンタに関する詳細な登録情報を確認したり、設定を完全に削除することができます。

関連項目： 「開発」タブ、「管理」タブおよび「アクセス」タブなど、「管理」ポートレットに表示されるフィールドや説明の詳細は、[第 12.2.5 項「管理」ポートレット](#)を参照してください。

12. 「閉じる」ボタンをクリックしてこのページを閉じ、OracleAS Portal の「Oracle Reports セキュリティ」ページに戻ります。

これで、OracleAS Portal へのプリンタの登録が完了しました。この登録は、OracleAS Portal を通じて実行されるレポートだけでなく、OracleAS Portal の外部から実行されるレポートにも意味があります。

12.2.4 使用可能日カレンダーの作成

使用可能日カレンダーの定義はオプションのステップで、レポート、サーバーおよびプリンタにアクセスできる時間とできない時間を指定することにより、これらへのアクセスをさらに制限できます。レポート、Reports Server およびプリンタがいつでも利用可能であれば、使用可能日カレンダーは不要です。

この項では、次の項目について説明します。

- [シンプルな使用可能日カレンダーの作成](#)
- [結合された使用可能日カレンダーの作成](#)

レポート、Reports Server またはプリンタに関連付けることができる使用可能日カレンダーは 1 つのみです。本番環境で複数の使用可能日ルールが必要な場合は、使用可能日カレンダーを結合できます。

12.2.4.1 シンプルな使用可能日カレンダーの作成

シンプルな使用可能日カレンダーは、1つの使用可能日ルールを定義したものです（たとえば、日曜日～土曜日の午前 12:00～午後 10:00 など）。

シンプルな使用可能日カレンダーを作成する手順は次のとおりです。

1. OracleAS Portal に管理者としてログインします。
2. 「ビルダー」 ページにナビゲートします。
3. 「管理」 タブをクリックします。
4. 「Oracle Reports セキュリティ」 ポートレットの「Oracle Reports セキュリティの設定」 リンクをクリックします。
5. 「Oracle Reports セキュリティ」 ページで、「レポート・カレンダー・アクセス」 ポートレットの「レポート・シンプル・カレンダー・アクセスの作成」 リンクをクリックします。
6. 表示されるページでは、「名前」（内部名）フィールドと「Portal DB プロバイダ」フィールドにデフォルト値が入力されています。カスタムの値を入力する手順は次のとおりです。
 - 「名前」 フィールドに、使用可能日カレンダーを OracleAS Portal 内部で一意に識別する名前（MY_CALENDAR など）を入力します。この名前は、有効なコンポーネント名に関する次のような OracleAS Portal のルールに従っている必要があります。
 - * 30 文字以下で指定する。
 - * 英数字のみ使用する（スペースおよび特殊文字は使用できません）。
 - * 名前の最初は必ず文字で始める（数字は不可）。
 - 「表示名」 フィールドに、この使用可能日カレンダーが OracleAS Portal を通じて公開される際に表示される、このカレンダーの名前を入力します。内部名とは異なり、表示名にはスペースを含めることができます。
 - プロバイダの値リストから「Portal DB プロバイダ」を選択します。OracleAS Portal で追加または作成されるコンポーネントは、すべてポータル DB プロバイダに属している必要があります。このリストには、コンポーネントを作成する権限がある場合に使用できるプロバイダの名前のみが表示されます。

注意： ポータル DB プロバイダの作成の詳細は、OracleAS Portal のオンライン・ヘルプを参照してください。

7. 「次へ」をクリックします。
8. オプションとして、「説明」にカレンダーの説明を入力できます。
9. 「次へ」をクリックします。
10. 「使用可能日時」 ページで、カレンダーのパラメータを定義します。

「期間」には、1単位の継続期間を構成する時間の長さを指定します。たとえば、レポートのアクセスを、任意の日の午前 9:00 から午後 5:00 まで許可するカレンダーを設定するとします。この場合、月、日および年については「開始」と「終了」の設定が同じになりますが、時間と分については、「開始」の設定が午前 9:00、「終了」の設定が午後 5:00 になります。この例では、任意の日におけるレポートの使用可能期間が、午前 9:00～午後 5:00 になります。

「繰返し」には、継続期間が繰り返される頻度を指定します。

- 「一度のみ繰返し」は、継続期間が繰り返されず、期間が終了すると関連するコンポーネントが利用できなくなることを意味します。たとえば、「一度のみ繰返し」を選択し、1年の継続期間を設定した場合は、1年後に関連コンポーネントが利用できなくなります。

- 「毎年」は、継続期間が毎年繰り返されることを意味します。「毎年」を選択し、「期間」の設定で、開始と終了の日付を同じにして、「開始」時間を午前 9:00、「終了」時間を午後 5:00 に設定した場合は、この使用可能日カレンダーに関連付けられている Reports コンポーネントが毎年同じ日の午前 9:00 ～午後 5:00 に利用可能となります。
- 「毎月」は、「期間」に指定された「開始」と「終了」の日付で表される継続期間が毎月繰り返されることを意味します。「毎月」を選択し、「開始」と「終了」の両方に同じ年と日付（例：2001 年 7 月 25 日）を指定し、「開始」時間を午前 9:00、「終了」時間を午後 5:00 に設定した場合は、関連コンポーネントが毎月 25 日の午前 9:00 ～午後 5:00 に利用可能となります。
- 日付/曜日毎の設定は、「毎月」のみに適用されます。日付/曜日毎では、継続期間が特定の日付で設定されるか（たとえば、月の 25 ～ 29 日は常に利用可能とするなど）、または特定の曜日で設定されるか（たとえば、月曜～金曜は常に利用可能とするなど）を指定します。
- 「毎週」は、「期間」に指定された曜日の期間が毎週繰り返されることを意味します。
- 「毎日」は、「期間」に指定された時間が、毎日繰り返されることを意味します。
- 「頻度」には、繰返し間隔 n （「繰返し」で選択した値に応じて、年数、月数、週数、日数を表す）の値を指定します。たとえば、毎週繰り返すように継続期間を設定し、「頻度」を 2 に設定すると、継続期間が 2 週ごとに（つまり隔週）繰り返されます。
- オプションで、「繰返し期限」をチェックして、カレンダーの終了日時を割り当てます。これによって、「繰返し期限」の日時に、すべての関連コンポーネントが利用できなくなります。

注意： カレンダーでは妥当性チェックが実行されません。継続期間が繰返しの設定を超過しても、エラー・メッセージは生成されません。たとえば、10 日間の継続期間を設定し、繰返しを毎週に設定した場合は、期間が重複しますが、ユーザーにはその重複が通知されません。

11. 「次へ」をクリックします。
12. 「サマリー」ページで「カレンダーの表示」ボタンをクリックして、設定した使用可能日カレンダーをプレビューします。設定を変更する場合は、「前へ」ボタンをクリックして変更を加えます。
13. 「サマリー」ページで「終了」をクリックして、使用可能日カレンダーの作成を完了します。
このカレンダーに関する設定を要約したページが表示されます。このページでは、設定を編集できます。また、カレンダーに関する詳細な情報を確認したり、設定を削除することができます。
14. 「閉じる」ボタンをクリックしてこのページを閉じ、OracleAS Portal の「Oracle Reports セキュリティ」ページに戻ります。

このカレンダーは、他のカレンダーと結合することも、登録されている OracleAS Reports Services コンポーネントにそのまま適用することもできます。

12.2.4.2 結合された使用可能日カレンダーの作成

結合された使用可能日カレンダーは、複数の使用可能日カレンダーを結合して 1 つの使用可能日カレンダーにしたものです。これは、使用可能期間を設定した後に、その期間から特定の日（休日など）を除外する場合に便利です。

カレンダーを結合する際に、一方のカレンダーのすべての日から、もう一方のカレンダーのすべての日を除外するように指定できます。たとえば、あるカレンダーでは月曜～金曜が使用可能日であり、別のカレンダーでは水曜日のみが使用可能日であるとします。この 2 つのカレンダーを結合し、その際に水曜日のカレンダーを除外することで、結合されたカレンダーでは月曜、火曜、木曜、金曜が使用可能日となります。

あるいは、年全体で平日を使用可能日とするシンプルなカレンダーを作成し、その後で新年の休暇を除外するカレンダー、その次の休暇を除外するカレンダー、さらにその次の休暇を除外するカレンダーなどのシンプルなカレンダーを追加します。これらすべてのカレンダーを結合し、休暇のカレンダーすべてを除外することで、コンポーネントは年間を通じて会社営業日のある一定時間のみ利用可能となります。

使用可能日カレンダーを結合する手順は次のとおりです。

1. OracleAS Portal に管理者としてログインします。
2. 「ビルダー」 ページにナビゲートします。
3. 「管理」 タブをクリックします。
4. 「Oracle Reports セキュリティ」 ポートレットの「Oracle Reports セキュリティの設定」 リンクをクリックします。
5. 「レポート・カレンダー・アクセス」 ポートレットの「レポート・シンプル・カレンダー・アクセスの作成」 リンクをクリックします。
6. カレンダーの内部名、表示名およびポータル DB プロバイダを指定します。
 - 「名前」 フィールドに、結合された使用可能日カレンダーを OracleAS Portal 内部で一意に識別する名前 (MY_COMBINED_CALENDAR など) を入力します。この名前は、有効なコンポーネント名に関する次のような OracleAS Portal のルールに従っている必要があります。
 - * 30 文字以下で指定する。
 - * 英数字のみ使用する (スペースおよび特殊文字は使用できません)。
 - * 名前の最初は必ず文字で始める (数字は不可)。
 - 「表示名」 フィールドに、この結合された使用可能日カレンダーの表示名を入力します。この名前は、OracleAS Portal を通じてユーザーに公開されます。

注意: 内部名とは異なり、表示名にはスペースを含めることができます。

 - プロバイダの値リストから「Portal DB プロバイダ」を選択します。Portal で追加または作成するコンポーネントは、すべてポータル DB プロバイダに属している必要があります。このリストには、コンポーネントを作成する権限がある場合に使用できるプロバイダの名前のみが表示されます。

注意: ポータル DB プロバイダの作成の詳細は、OracleAS Portal のオンライン・ヘルプを参照してください。

7. 「次へ」をクリックします。
8. (オプション) 「説明」 フィールドに使用可能日カレンダーの説明を入力します。
9. 「次へ」をクリックします。
10. 「選択」 ページで、「使用可能日カレンダー」 リストの中から結合するカレンダーを選択します。カレンダーは、表示名ではなく内部名で表示されます。複数のカレンダーを選択するには、[Ctrl] を押しながらかlickする (Windows の場合)、クリックします (UNIX の場合)。このページに表示されるのは、結合された使用可能日カレンダーの作成で使用するポータル DB プロバイダに対してすでに定義されている使用可能日カレンダーです。
11. 右矢印をクリックして、選択したカレンダーを「選択された使用可能日カレンダー」 リストに移動します。
12. 「次へ」をクリックします。

13. 「**除く**」 ページで、「**使用可能日カレンダー**」 リストの中から除外する日付を含んだカレンダーを選択します。複数のカレンダーを選択するには、**[Ctrl]** を押しながらクリックするか (Windows の場合)、クリックします (UNIX の場合)。

これらのカレンダーには、使用可能日の指定を取り消す日付が含まれています。

14. 右矢印をクリックして、選択したカレンダーを「**除外された使用可能日カレンダー**」 リストに移動します。

15. 「**次へ**」 をクリックします。

16. 「**サマリー**」 ページで、「**カレンダーの表示**」 ボタンをクリックして、設定したカレンダーをレビューします。

除外した箇所が表示されていない場合は、別のビューを選択します。たとえば、月のビューのかわりに週のビューを選択します。

結合を変更する場合は、カレンダーを閉じ、目的のページに戻るまで「**前へ**」 ボタンをクリックします。

17. 「**終了**」 をクリックして、結合されたカレンダーの作成を完了します。

このカレンダーに関する設定を要約したページが表示されます。このページでは、設定を編集できます。また、カレンダーに関する詳細な情報を確認したり、設定を削除することができます。

関連項目：「開発」タブ、「管理」タブおよび「アクセス」タブなど、「管理」ポートレットに表示されるフィールドや説明の詳細は、[第 12.2.5 項「管理」ポートレット](#) を参照してください。

18. 「**閉じる**」 ボタンをクリックしてこのページを閉じ、OracleAS Portal の「**Oracle Reports セキュリティ**」 ページに戻ります。

このカレンダーは、他のカレンダーと結合することも、登録されている OracleAS Reports Services コンポーネントにそのまま適用することもできます。

12.2.5 「管理」ポートレット

既存の OracleAS Portal Portal ポートレットに対して、実行、編集、コピー、削除または関連情報の表示などの操作を行うには、「管理」ポートレットのページを使用します。

ポートレットに対して実行できる操作は、権限によって異なります。また、ここに記載されているすべての操作を、すべてのポートレットに対して実行できるわけではありません。ページの左上に、これらの操作を実行できるポートレットの名前が表示されます。

[表 12-5](#) では、「**開発**」 タブに表示されるフィールドについて詳しく説明しています。

表 12-5 「開発」タブ

フィールド	説明
(ポートレットのタイプおよび名前)	ポートレットのタイプと名前を表示します。次に例を示します。 フォーム (表) my_form これは、 my_form という表に基づくフォームを表しています。
プロバイダ	ポートレットが作成されたプロバイダの名前を表示します。
バージョン状態 (一部のポートレットには適用不可)	ポートレットのすべてのバージョンと、各バージョンの現行の状態を表示します。ポートレットのバージョンを編集する場合はステータスをクリックします。 注意： ハイパーリンクがない場合、そのポートレットを編集する権限がありません。
最終更新	ポートレットを作成したユーザーまたは最後に編集したユーザーの名前と、ポートレットが作成された日時または最後に編集された日時を表示します。

表 12-5 「開発」タブ (続き)

フィールド	説明
実行リンク (一部のポートレットには適用不可)	<p>ポートレットを実行して表示するプロシージャの URL を表示します。この URL をコピーし、別の Web ページに貼り付けて、このポートレットへのリンクを作成できます。</p> <p>注意: パラメータを指定せずにポートレットを実行するプロシージャには、.show という接尾辞が付いています。パラメータを指定してポートレットを実行するプロシージャには、.show_parms という接尾辞が付いています。</p>
PL/SQL ソース (一部のポートレットには適用不可)	<p>「ポートレット・ビルダー」ウィザードによって、各ポートレットを表す PL/SQL パッケージが作成されます。</p> <p>パッケージ仕様: ポートレットの PL/SQL 仕様を表示します。</p> <p>パッケージ本体: ポートレットの PL/SQL 本体を表示します。</p>
コール・インタフェース (一部のポートレットには適用不可)	<p>ポートレットで実行時にエンド・ユーザーが変更できる引数を表示するには、「表示」をクリックします。PL/SQL スタッド・プロシージャおよび URL を介してポートレットをコールするサンプルも表示されます。PL/SQL で、または URL からコールすることによって、ポートレットが格納されたパッケージを実行する場合、異なる引数を受け付けるようにコール・インタフェースを編集できます。</p> <p>注意: ポートレットのソース・コードを表示するには、ポートレットまたはそのポートレットを所有するプロバイダに対するカスタマイズ権限または実行権限が必要です。</p>
データ・リンクの編集 (一部のポートレットには適用不可)	<p>データが格納された URL に接続し、そのデータを表示したり編集する場合にクリックします。</p>
編集	<p>ポートレットの最新バージョンを編集する場合にクリックします。たとえば、ポートレットの基になっている表の列の再選択、ポートレットに表示されるフィールドやテキストの変更、または新しいロック・アンド・フィールドの選択を行うことができます。</p>
新規として編集	<p>このポートレットの新しいバージョンを作成して編集する場合にクリックします。既存のポートレットのバージョンは変更されません。</p>
データの編集 (一部のポートレットには適用不可)	<p>スプレッドシートを表示し、その中のデータを編集する場合にクリックします。</p>
実行	<p>ポートレットの現在の本稼動バージョンを実行する場合にクリックします。</p> <p>注意: ポートレットの有効なパッケージがない場合、ポートレットは実行されません。</p>
ポートレットとして実行	<p>ポータル・ウィンドウにおけるポートレットの外観を表示します (フル・ページ表示とは異なる場合があります)。</p>
カスタマイズ	<p>ポートレットのカスタマイズ・フォームを表示する場合にクリックします。カスタマイズ・フォームを使用すると、ポートレットの表示に使用する値を指定できます。</p> <p>注意: 現在のポートレットがフォームである場合、このページには「カスタマイズ」ではなく「ブラウザ」が表示されます。</p>
お気に入り追加	<p>OracleAS Portal ホーム・ページの「お気に入り」リストにポートレットを追加する場合にクリックします。</p>
バージョン情報	<p>ポートレットに対して格納されている属性を表示します。</p>
削除	<p>データベースからポートレットを削除する場合にクリックします。</p>

表 12-6 では、「管理」タブに表示されるフィールドについて詳しく説明しています。

表 12-6 「管理」タブ

フィールド	説明
SQL 問合せ情報を表示する /SQL 問合せ情報を非表示にする (一部のポートレットには適用不可)	デバッグを目的として、ポートレットを実行する際に SQL 問合せを表示するか非表示にするかを選択します。
このポートレットのロックを表示 (一部のポートレットには適用不可)	ポートレットで現在アクティブなロック (他のユーザーが編集している場合など) を表示します。
エクスポート	データベースからポートレットをエクスポートする場合にクリックします。
コピー	データベースからポートレットをコピーする場合にクリックします。
改名	(同じプロバイダ内で) ポートレットの名前を変更する場合にクリックします。
生成	PL/SQL パッケージをコンパイルする場合にクリックします。
監視	ポートレットに関するすべてのリクエストとそのリクエストを作成したユーザーのチャートを表示する場合にクリックします。

表 12-7、表 12-8、表 12-9、表 12-10、表 12-11、表 12-12 と表 12-13 では、「アクセス」タブに表示されるフィールドについて詳しく説明しています。

表 12-7 Portal アクセス

フィールド	説明
ポートレットとして公開 (一部のポートレットには適用不可)	ポートレットを1つのポートレットとして使用可能にする場合にクリックします。 注意: ポートレットを1つのポートレットとして公開するには、ポートレットの公開権限が必要です。またアクセス・プロバイダ・ページ (「管理」タブ) の「プロバイダとして公開」で、そのポートレットを所有するプロバイダを使用可能にする必要があります。

表 12-8 権限モード

フィールド	説明
プロバイダから権限を継承	プロバイダのアクセス権限によってポートレットのアクセス権限を上書きする場合は、これを選択します。 ポートレットのアクセス権限によってプロバイダのアクセス権限が上書きされるようにするには、チェック・ボックスの選択を解除し、「適用」をクリックします。「アクセス権限の付与」セクションでは、様々なユーザーやグループについて、ポートレットのアクセス権限 (管理、編集、表示、カスタマイズ、実行など) を付与したり削除することができます。 注意: ユーザーやグループにポートレットのアクセス権限を付与するには、ポートレットまたはそのポートレットを所有するプロバイダに対する管理アクセス権限が必要です。

表 12-9 アクセス権限の付与

フィールド	説明
ユーザー / ロール	プロバイダのアクセス権限を付与するユーザーまたはグループを入力します。
実行	付与する権限を選択します。
追加	プロバイダのアクセス権限を付与する場合にクリックします。

表 12-10 アクセス権限の変更

フィールド	説明
ユーザー / ロール	権限が割り当てられている OracleAS Portal のユーザーまたはグループを表示します。すべての権限を削除するには、「ユーザー / ロール」の横をクリックします。すべての OracleAS Portal ユーザーに権限を付与する場合は、「ユーザー / ロールとして公開」を選択します。
タイプ	ユーザー / ロールが OracleAS Portal ユーザーまたは OracleAS Portal グループのどちらであるかを表示します。
権限	現在付与されている権限を表示します。権限を変更するには、新しい権限を選択して「適用」をクリックします。

表 12-11 セルの権限モード

フィールド	説明
ポートレットから権限を継承 (一部のポートレットには適用不可)	<p>ポートレットのアクセス権限によってセルのアクセス権限を上書きする場合は、これを選択します。</p> <p>セルのアクセス権限によってポートレットのアクセス権限が上書きされるようにするには、チェック・ボックスの選択を解除し、「適用」をクリックします。「アクセスの変更」セクションでは、様々なユーザーやグループについて、セルのアクセス権限（管理、編集、表示、カスタマイズ、実行など）を変更できます。</p> <p>注意: ユーザー / ロールのセルのアクセス権限を変更するには、ポートレットまたはそのポートレットを所有するプロバイダに対する管理アクセス権限が必要です。</p>

表 12-12 アクセスの変更

フィールド	説明
ユーザー / ロール (一部のポートレットには適用不可)	セルのアクセス権限を付与するユーザーまたはグループを入力します。
変更 (一部のポートレットには適用不可)	セルのアクセス権限を変更する場合にクリックします。

表 12-13 キャッシュの無効化

フィールド	説明
キャッシュの消去	次のデータ要求でデータベースからデータが入力されるように、キャッシュされているデータを消去します。

12.3 レポートのポートレットとしての公開

Oracle Reports を登録した後、次の手順を実行すると、レポートをポータルに公開できます。

1. レポートのプロバイダを作成します。この手順では、ポータル内でユーザーに提供するレポートが収納されるプロバイダを定義します。または、OracleAS Portal にデフォルトで含まれている既存のプロバイダを使用することもできます。

詳細は、第 12.3.1 項「レポートに対するプロバイダの作成」を参照してください。

2. レポートをアイテム・リンク¹またはポートレット²としてページに追加し、必要に応じてカスタマイズします。これにより、ユーザーがレポートをページ上で利用できるようにするとともに、ページ設計者がレポート・パラメータを設定し、レポートの自動実行をスケジューリングできるようにします。

詳細は、次の項を参照してください。

- 第 12.3.2 項「ページへのレポート・ポートレットの追加」
- 第 12.3.3 項「レポート・コンポーネントのアイテム・リンクとしてのページへの追加」
- 第 12.3.4 項「デフォルトでインストールされていない OracleAS Portal でのアイテム・リンクとしてのレポートの実行」

12.3.1 レポートに対するプロバイダの作成

レポートを収納するプロバイダをまだ定義していない場合は、これを作成する必要があります。プロバイダの作成の詳細は、OracleAS Portal のオンライン・ヘルプを参照してください。

注意： レポートを収納するプロバイダは、データベース・プロバイダである必要があります。また、その「アクセス」ページで「プロバイダとして公開」の設定が選択されていることが必要です。

12.3.2 ページへのレポート・ポートレットの追加

レポートを OracleAS Portal に登録した後、そのレポートをポータル・ページ上のポートレットとして公開できます。

注意： レポートをポートレットとして公開するには、「ポートレットとして公開」ボックスを選択しておく必要があります。

レポートをポートレットとして公開するには、次の作業が必要です。

1. まだ「ビルダー」ページを表示していない場合は、ページ上部の「ビルダー」をクリックします。
2. 「構築」タブをクリックします。
3. 「ページ・グループ」ポートレットで、作成したレポート・ポートレットを配置するページ・グループの名前を選択します。
4. 「ページの作成」をクリックして新しいページを作成するか、既存のページの名前を入力して「編集」をクリックすることでそのページを編集します。

¹ アイテム領域のページ上にある個々のコンテンツ要素（テキスト、ハイパーリンク、イメージなど）。

² 通常 Web コンテンツ部分を表示するために使用される、再利用可能なプラグgable Web コンポーネント。

- 新しいページを作成する場合は、ウィザードの手順に従います。設定に関する情報が必要な場合は、右上端にある疑問符をクリックしてください。作業が完了したら、「完了」をクリックします。

関連項目： Oracle Reports アイテムをページに追加する方法の詳細は、[第 12.3.3 項「レポート・コンポーネントのアイテム・リンクとしてのページへの追加」](#)を参照してください。

既存のページを編集する場合は、次の手順に進みます。

- レポート・ポートレットを追加するページ領域で、「**ポートレットの追加**」ツールをクリックします。

ヒント： ツールにマウスを合わせると、そのツールに関するヒントが表示されます。

- ポートレット・リポジトリを、レポート・ポートレットを含んでいるプロバイダまでドリルダウンします。レポート・ポートレットは、ポートレット・リポジトリの中の、そのポートレットが属しているポータル DB プロバイダの下に表示されます。プロバイダの位置は、ポートレット・リポジトリがどのように編成されているかによって異なります。ポータル DB プロバイダがかなり新しいプロバイダである場合、ポートレット・リポジトリの下方に表示される可能性があります。
- レポート・ポートレットの名前をクリックして、「**選択したポートレット**」リストに追加します。
- 「OK」をクリックします。
- レポート・ポートレットの右上端にある「**カスタマイズ**」をクリックします。
- 「**パラメータ**」タブにパラメータ値を入力し、必要に応じて「**スケジュール**」タブでジョブの自動実行をスケジュールリングします。
- ポートレットのサイズは、レポート定義ファイル・オブジェクトの「**カスタマイズ**」ページで「**ポートレットの幅**」パラメータと「**ポートレットの高さ**」パラメータを指定して制御できます。これらのパラメータ値は、パーセント (%) またはピクセル数で指定できます。

たとえば、次のように入力します。

ポートレットの幅: 90%

ポートレットの高さ: 480

値を指定しない場合は、OracleAS Reports Services によりデフォルトの値（幅が 640 ピクセル、高さが 320 ピクセル）が使用されます。

「**ポートレットの幅**」と「**ポートレットの高さ**」フィールドをユーザーに表示する場合は、「**カスタマイズ**」を使用して、ユーザーも各ポートレットの幅と高さを調整できます。ユーザーが指定した値によって、レポート定義ファイル・オブジェクト・コンポーネントの「**カスタマイズ**」ページで設定した値は上書きされます。

- レポートのパラメータをユーザーに表示するかどうかは、「レポート定義ファイル・アクセス」コンポーネントのカスタマイズ・ページで指定できます。

レポートのパラメータをユーザーに表示する手順は次のとおりです。

- レポートの「**開発**」タブの最下部にある「**カスタマイズ**」をクリックします。
- 表示するパラメータごとに「**ユーザーに表示**」を選択します。

注意： このページからパラメータのデフォルト値を設定することもできません。

表示するパラメータに、対応する OracleAS Portal ページのパラメータがある場合、「カスタマイズ」ページでパラメータ値を空白にしておくと、ポートレットにはそのページのパラメータ値が継承されます。ユーザーがポートレットのパラメータに値を入力した場合は、その値がページのパラメータより優先します。

注意： OracleAS Portal でレポートを実行するには、HTML の `iframe` タグを使用する必要があります。このタグは、Netscape 4.x ではサポートされていません。このため、Netscape 4.x では次のような制限があります。

- HTTPS を使用している場合またはレポート・ポートレットが JSP レポートでない場合は、レポート・ポートレットは適切な場所に表示されません。別のブラウザ・ウィンドウにレポートを表示するには、ポートレットのタイトルをクリックする必要があります。
 - HTTPS を使用している場合、**カスタマイズ**・リンクを介してレポート・ポートレットを実行するようにスケジューリングできません。
-

12.3.3 レポート・コンポーネントのアイテム・リンクとしてのページへの追加

Oracle Reports アイテム・タイプを使用して、Oracle Reports コンポーネントをアイテム・リンクとしてページに追加できます。OracleAS Portal をデフォルト以外の言語設定でインストールしている場合は、[第 12.3.4 項「デフォルトでインストールされていない OracleAS Portal でのアイテム・リンクとしてのレポートの実行」](#)を参照してください。

注意： このアイテム・タイプは非表示のアイテム・タイプのリストから指定する必要があり、ページ・グループ管理者のみがアイテム・タイプを構成できます。

1. まだ「ビルダー」ページを表示していない場合は、ページ上部の「**ビルダー**」をクリックします。
2. 「**構築**」タブをクリックします。
3. 「ページ・グループ」ポートレットで、作成したレポート・アイテム・リンクを配置するページ・グループの名前を選択します。
4. 「**ページの作成**」をクリックして新しいページを作成するか、既存のページの名前を入力して「**編集**」をクリックすることでそのページを編集します。
5. 新しいページを作成する場合は、ウィザードの手順に従います。設定に関する情報が必要な場合は、右上端にある疑問符をクリックしてください。作業が完了したら、「**完了**」をクリックします。

既存のページを編集する場合は、次の手順に進みます。

6. 「**アイテムの追加**」をクリックします。Oracle Reports アイテム・タイプは非表示のアイテム・タイプとして使用可能です。このアイテム・タイプを使用可能なアイテム・タイプに含めるには、「**使用可能なアイテム・タイプを設定**」リンクをクリックします。
7. 「**非表示のアイテム・タイプ**」リストから Oracle Reports を選択し、右矢印 (>) をクリックして「**表示されるアイテム・タイプ**」リストに移動します。または、「>>」をクリックして「**非表示のアイテム・タイプ**」リスト内のすべてのアイテムを「**表示されるアイテム・タイプ**」リストに移動します。
8. 「**OK**」をクリックします。
9. 「**コンテンツ・アイテム・タイプ**」メニューで Oracle Reports アイテム・タイプを選択し、「**次へ**」をクリックして、Oracle Reports の追加ページを表示します。
10. ポータルのユーザーがレポートをクリックしたときに表示される表示名を入力します。
11. 使用可能なデフォルトの Oracle Reports コンポーネントのリストから選択します。

12. レポートを表示する前にレポートにユーザー入力が必要な場合は、「**パラメータ・フォームの表示**」を選択します。
13. 「**新しいブラウザ・ウィンドウでアイテムを表示するリンク**」を選択し、レポートが別のページに表示されるようにします。
14. 「**完了**」をクリックします。Oracle Reports アイテムがページにリンクとして表示されます。
15. リンクをクリックしてレポートを実行します。「**パラメータ・フォームの表示**」が選択されている場合は、必要なパラメータを入力します。

12.3.4 デフォルトでインストールされていない OracleAS Portal でのアイテム・リンクとしてのレポートの実行

OracleAS Portal をデフォルト以外の言語設定でインストールした場合、レポートをポータル・ページのアイテム・リンクとして公開する際に必要となるエントリーには、自動的にインストールされないものがあります。選択した言語は、`rwlang.sql` スクリプトを使用してインストールする必要があります。

したがって、次の条件がどちらも該当する場合は、スクリプト `rwlang.sql` (ORACLE_HOME/portal/admin/plsql/wwd/) を実行する必要があります。

- OracleAS Portal のインストール時にデフォルト (US) 以外の言語を1つ以上選択した。
- OracleAS Portal のアイテム・リンクとしてレポートを公開するつもりである。

注意： この作業はインストール後に1回実行しておくこと、OracleAS Portal のアイテム・リンクとしてレポートを公開できるようになります。

スクリプトを実行するには、次の作業を実行します。

1. ディレクトリを `ORACLE_HOME/portal/admin/plsql/wwd/` に変更します。
2. `sqlplus` を実行します。
3. ポータル・スキーマを使用して OracleAS Portal にログオンします。
4. これは、OracleAS Portal PL/SQL パッケージのインストールに使用するポータル・スキーマです。
5. 次のパラメータを使用して `rwlang.sql` スクリプトを実行します。

```
@rwlang.sql language_list
```

ここでは

`language_list` は、カンマで区切った言語のリストを示します。

たとえば、フランス語と日本語をインストールするには次のように実行します。

```
@rwlang.sql f,ja
```

使用上の注意

- カンマ (,) の前後にはスペースを入れません。`sqlplus` は言語リストを、カンマ (,) で区切られた1つのパラメータとしてではなく2つのパラメータとして処理するためです。
- `rwlang.sql` スクリプトのヘッダーには、すべての言語の略語が一覧表示されています。スクリプト・ファイルは任意のテキスト・エディタで開いて、各種の略語を調べることができます。

12.4 トラブルシューティング情報

この項では、問題が発生した場合に修正するための各種の手順について説明します。

12.4.1 リソースの作成の試行時の Reports と Portal の統合エラーの解決

OracleAS Portal で、「レポート定義ファイル・アクセス」の「Oracle Reports セキュリティ」設定を構成する場合、レポート定義ファイルの編集時に「実行」または「ポートレットとして実行」をクリックすると、エラーが発生する場合があります。

```
500 Internal Server Error
Unexpected Error. Please contact Administrator
```

このエラーは、次の条件がすべて該当する場合に発生します。

- 相互運用環境を実現し（9.0.2 と 9.0.4 環境を共存させ）、9.0.4MT（中間層）、9.0.4IM（Identity Management）、9.0.2MR（メタデータ・リポジトリ）を同時に実行するように構成している。
- SSOCONN パラメータを使用して、OracleAS Portal 内で Oracle Reports を実行している。
- SSOCONN パラメータで指定した接続リソースが Oracle Internet Directory サーバーに作成されていない。

この問題を解決するには、次の作業を実行します。

1. テキスト・エディタを使用して、9.0.4IM の `ORACLE_HOME` にある次のファイルを開きます。

```
ORACLE_HOME/Apache/Apache/conf/mod_osso.conf
```

2. 次のフラグを追加します。

```
OssRedirectByForm on
```

次に例を示します。

```
<IfModule mod_osso.c>
OssIpCheck off
OssIdleTimeout off
OssConfigFile
/private1/iasinst/install_set1/904infra/Apache/Apache/conf/osso/osso.conf
OssRedirectByForm on
</IfModule>
```


第 II 部

サーバーへの要求送信

第 II 部では、レポート公開に関する有用な詳細情報を説明します。これには、要求の実行方法、高度な自動レポート配布の設定方法、XML カスタマイズ・ファイルを使用して実行時にレポートをカスタマイズする方法、データベース・トリガーを使用してレポートを自動的に起動する方法などが含まれます。

第 II 部には、次の章があります。

- [第 13 章 「レポート・リクエストの実行」](#)
- [第 14 章 「Oracle Reports Web サービスの使用法」](#)
- [第 15 章 「拡張配布の作成」](#)
- [第 16 章 「XML によるレポートのカスタマイズ」](#)
- [第 17 章 「イベント・ドリブンによる公開の使用法」](#)

レポート・リクエストの実行

この章では、Reports Server にレポート・リクエストを送信する様々な方法について説明します。この章には、次の項があります。

- Reports の URL 構文
- レポート・リクエストの方法
- レポートの配布
- OracleAS Portal でのレポートの公開
- Web ブラウザからのレポート・リクエストの指定
- URL エンジンへのリクエスト送信
- Web サービスを介したレポートの実行
- Oracle Workflow からのレポートの実行
- 自動実行のためのレポートのスケジューリング
- 追加のパラメータ
- キャッシュからのレポート出力の再使用
- キー・マップ・ファイルの使用方法

13.1 Reports の URL 構文

この項では、レポートを公開するための URL の作成に関するクイック・リファレンス情報を提供します。ここで取り上げる配布タイプは、次の3つです。

- サブレット
- JSP
- CGI (下位互換性を確保するため)

ここで提供する情報は、Windows 環境と UNIX 環境でほとんど同じです。相違点も示してあります。

13.1.1 サブレット

Reports Servlet を経由したレポート実行の URL 構文は、次のとおりです。

```
http://web_server.domain_name:port/alias/rwservlet?parameters
```

表 13-1 では、サブレットの URL の構成要素を説明します。

表 13-1 Reports Servlet をコールする URL の構成要素

コンポーネント	説明
<i>web_server</i>	Oracle HTTP Server をインストールしたときに各自で付けた名前。
<i>domain_name</i>	所属する組織のドメイン名。
<i>port</i>	Oracle HTTP Server がリクエストをリスニングするポート番号。ポートが指定されない場合は、デフォルト (80) が使用されます。
<i>alias</i>	URL がアクセスするファイルへの絶対パスにかわる仮想パス。
<i>rwservlet</i>	Reports Servlet を起動します。
?	コマンドライン・オプションの開始を示します。
<i>parameters</i>	すべてのコマンドライン・オプション、またはコマンドライン・オプションが指定されているキー・マップ・ファイルへのキー。

Reports Servlet をコールする URL は、次のようになります。

```
http://neptune.world.com:80/reports/rwservlet?keyname
```

keyname は、*cgicmd.dat* ファイル内の一意のヘッダー (キー名) の下に示されているコマンドラインを参照します。JSP ファイルの場合は、この処理が異なるので注意してください。JSP ファイルでは、キーワード / 値のペア (*cmdkey=value*) を使用して、*cgicmd.dat* ファイル内のコマンドラインに対するキー名を指定します。キー・マッピングの使用の詳細は、[第 13.12 項「キー・マップ・ファイルの使用方法」](#)を参照してください。

Reports Servlet を使用する場合、JSP ファイルにペーパー・レイアウトが含まれていれば、JSP レポート・ファイルを実行することもできます。レポートを実行する際に、URL に Reports Servlet を指定し、コマンドライン・オプション (*report=myreport.jsp*) で JSP をコールします。

次に例を示します。

```
http://neptune.world.com:80/reports/rwservlet?report=myreport.jsp&destype=cache&desformat=html
```

コマンドライン・キーワードおよび値の詳細は、[付録 A「コマンドライン・キーワード」](#)を参照してください。

13.1.2 JSP

JSP ベース・レポートの URL の構文は、次のとおりです。

```
http://web_server.domain_name:port/alias/myreport.jsp?parameters
```

表 13-2 では、JSP ベース・レポートの URL の構成要素を説明します。

表 13-2 JSP ベース・レポートの URL の構成要素

コンポーネント	説明
<code>web_server</code>	Oracle HTTP Server をインストールしたときに各自で付けた名前。
<code>domain_name</code>	所属する組織のドメイン名。
<code>port</code>	Oracle HTTP Server がリクエストをリスニングするポート番号。ポートが指定されない場合は、デフォルト (80) が使用されます。
<code>alias</code>	URL がアクセスするファイルへの絶対パスにかわる仮想パス。
<code>myreport.jsp</code>	この URL で実行されるレポートの *.jsp ファイル。
<code>?</code>	コマンドライン・オプションの開始を示します。
<code>parameters</code>	すべてのコマンドライン・オプション、またはコマンドライン・オプションが指定されているキー・マップ・ファイルへのキー。

JSP ベース・レポートの起動に使用される URL は、次のようになります。

```
http://neptune.world.com:80/jsp/myreport.jsp?
```

URL には、追加のコマンドライン・パラメータを含む `cgicmd.dat` ファイル内のコマンドラインを参照するキーを指定できます。その場合は、名前と値のペア (`cmdkey=keyname`) を使用する必要があります。これは、疑問符で示される問合せ文字列の開始点の後であれば、URL の任意の位置に指定できます。次に例を示します。

```
http://neptune.world.com:80/jsp/myreport.jsp?userid=scott/tiger@hrdb&cmdkey=key1
```

URL でパラメータを結合するときは、スペースを一切挿入せずに、アンパサンド (&) を使用します。

JSP を使用する場合は、Reports Servlet を使用することもできます。レポートを実行する際に、URL に Reports Servlet を指定し、コマンドライン・オプション (`report=myreport.jsp`) で JSP をコールします。

次に例を示します。

```
http://neptune.world.com:80/reports/rwservlet?report=myreport.jsp&destype=cache&desformat=html
```

コマンドライン・キーワードの詳細は、付録 A 「コマンドライン・キーワード」を参照してください。cgicmd.dat ファイルの詳細は、第 13.12 項「キー・マップ・ファイルの使用法」を参照してください。

13.1.3 CGI

注意： Oracle Reports 10g では、Reports CGI (rwcgi) を使用することはお薦めしません（下位互換性を確保するために維持されています）。かわりに、JSP、rwservlet (Reports Servlet)、または Web Services を使用することができます。

rwcgi よりも rwservlet をお薦めするのは、パフォーマンス上の理由があるからです。rwcgi では、各リクエストに対して新規プロセスが起動されます。その際、JVM が初期化されるため、多数のレポート・リクエストを実行するとパフォーマンスが低下します。一方、rwservlet は OC4J インスタンスに配置され、サーブレット機能を使用するため、rwcgi よりもパフォーマンスが向上します。

Windows で Reports CGI を経由してレポートを実行する URL の構文は、次のとおりです。

```
http://web_server.domain_name:port/alias/rwcgi.exe?parameters
```

UNIX では、次のようになります。

```
http://web_server.domain_name:port/alias/rwcgi.sh?parameters
```

表 13-3 では、CGI ベース・レポートの URL の構成要素を説明します。

表 13-3 Reports CGI をコールする URL の構成要素

コンポーネント	説明
<i>web_server</i>	Oracle HTTP Server をインストールしたときに各自で付けた名前。
<i>domain_name</i>	所属する組織のドメイン名。
<i>port</i>	Oracle HTTP Server がリクエストをリスニングするポート番号。ポートが指定されない場合は、デフォルト (80) が使用されます。
<i>alias</i>	URL がアクセスするファイルへの絶対パスにかわる仮想パス。
<i>rwcgi.exe</i>	OracleAS Reports Services の CGI コンポーネントを起動する実行可能ファイル。UNIX マシンに OracleAS Reports Services がインストールされている場合は、".exe" のかわりに ".sh" を使用します。
?	コマンドライン・オプションの開始を示します。
<i>parameters</i>	すべてのコマンドライン・オプション、またはコマンドライン・オプションが指定されているキー・マップ・ファイルへのキー。

Windows で CGI 実装の起動に使用される URL は、次のようになります。

```
http://neptune.world.com:80/cgi-bin/rwcgi.exe?key2
```

UNIX では、次のようになります。

```
http://neptune.world.com:80/cgi-bin/rwcgi.sh?key2
```

13.2 レポート・リクエストの方法

レポート・リクエストは、様々な方法で実行できます。次に各方法について説明します。

- **rwclient** コマンドライン

`rwclient` コマンドライン (UNIX では `rwclient.sh`) は、非 Web アーキテクチャ内でコマンドラインからレポート・リクエストを実行するために使用します。このコマンドラインは、コマンドラインを解析して指定の Reports Server に送る実行可能ファイルを参照します。このコマンドラインでは、Reports Runtime の実行可能ファイル `rwrun` (UNIX では `rwrun.sh`) で使用されるものと類似したコマンドライン・オプションを使用できます。

Windows での典型的な `rwclient` コマンドライン・リクエストは、次のようになります。

```
rwclient report=my_report.rdf userid=username/password@my_db server=server_name
destype=cache desformat=html
```

UNIX では、同じコマンドを次のように記述します。

```
rwclient.sh report=my_report.rdf userid=username/password@my_db server=server_name
destype=cache desformat=html
```

コマンドライン・オプションの詳細は、付録 A 「コマンドライン・キーワード」を参照してください。

- **URL**

ブラウザからレポートを実行する場合は、URL 構文を使用します。この URL 構文は、Reports Servlet (および下位互換性のための CGI) によって `rwclient` コマンドライン・リクエストに変換され、その後 OracleAS Reports Services によって処理されます。ユーザーのブラウザからレポート・リクエストを行うのに必要な URL 構文をユーザーに提供するか、または URL 構文をハイパーリンクとして Web サイトに追加することができます。この方法については、この後の項で詳しく説明します。

- **OracleAS Portal 経由**

OracleAS Portal コンポーネントを使用すると、OracleAS Portal ページまたはポートレット内にレポートへのリンクを追加するか、またはレポートの結果を直接ポートレットに出力することができます。各レポート・リンクは、レポート・リクエストに関する情報を含んだパッケージ・プロシージャを示します。OracleAS Reports Services のシステム管理者は、OracleAS Portal のウィザードを使用してパッケージ・プロシージャを作成することで、Web を通じてレポートをより便利で安全に公開できるようにします。OracleAS Portal ページ・グループにアクセスする権限のあるユーザーは、リンクをクリックするだけでレポートを実行できます。システム管理者はウィザードから直接レポートを実行できます。詳細は、OracleAS Portal のオンライン・ヘルプを参照してください。

レポートをポートレットとして公開する方法の詳細は、第 13.4 項「OracleAS Portal でのレポートの公開」を参照してください。

- **パッケージ・プロシージャ**

`SRW.RUN_REPORT` は、Reports Runtime のコマンドを実行するビルトイン・プロシージャです。`SRW.RUN_REPORT` を指定するときは、`SERVER` オプションを Reports Server の名前に設定して、`SRW.RUN_REPORT` コマンドが `rwclient` コマンドと同じように動作するようにします。

詳細は、第 17 章「イベント・ドリブンによる公開の使用方法」を参照してください。`SRW.RUN_REPORT` の詳細は、Oracle Reports のオンライン・ヘルプを参照してください。

- **Web サービス**

OracleAS Reports Services を Web サービスとして公開し、Web サービスをサポートする任意の環境 (Java アプリケーションなど) からコールできます。

詳細は、第 14 章「Oracle Reports Web サービスの使用方法」を参照してください。

13.3 レポートの配布

レポートを作成した後、エンド・ユーザーが参照できるようにレポートを配布できます。この項では、ペーパー・レイアウトによるレポート（REP、RDF、XML または JSP レポート）の配布方法および Web レイアウトによるレポート（JSP レポート）の配布方法について説明します。

注意： JSP ベースの Web レポートの作成とテストの例は、『Oracle Reports チュートリアル』、および『Oracle Reports レポート作成ガイド』の Web レポートの JSP パラメータ・フォームの作成方法に関する項を参照してください。

次の表は、レポートの配布に使用できる方法を、レポートのタイプごとに示しています。

表 13-4 レポートの配布方法

レポートのタイプ	方法	使用目的
ペーパー・レイアウトを使用したレポート（REP、RDF、XML）	ペーパー・レイアウトを使用したレポートの配布	ペーパー・レイアウトのみを使用してレポートを配布する方法。
ペーパー・レイアウトを使用した JSP レポート	ペーパー・レイアウトを使用したレポートの配布	どのタイプのペーパー・レポートに対しても、最も簡単な配布方法。ただし、JSP レポートでペーパー・レイアウトと Web レイアウトの両方を使用する場合は、 第 13.3.3 項「Web およびペーパーへの JSP レポートの配布」 を参照することをお勧めします。
ペーパー・レイアウトおよび Web レイアウトを使用した JSP レポート	Web およびペーパーへの JSP レポートの配布	Web レイアウトとペーパー・レイアウトの両方を使用してレポートを公開する場合に推奨される方法。

13.3.1 ペーパー・レイアウトを使用したレポートの配布

ペーパー・レポートを作成した後、ユーザーがレポートを実行できるように、レポートを Reports Server に配布できます。この項の手順では、RDF、REP、XML または JSP タイプのレポートを配布する方法について説明します。

注意： JSP レポートは、レポート設計者が JSP レポートに使用したレイアウトに応じて、Web またはペーパーのどちらへでも配布できます。この項では、ペーパー・レイアウトを使用して JSP レポートを配布する手順について説明します。ペーパー・レイアウトおよび Web レイアウトを使用した JSP レポートの配布については、[第 13.3.3 項「Web およびペーパーへの JSP レポートの配布」](#)の手順を参照してください。

レポートが Java クラス（たとえば、バーコード・クラスや Web サービス・スタブなど）に依存する場合は、これらのクラスにアクセスするようにプロセスを構成する必要があります。つまり、ペーパー・レイアウトを使用した JSP レポートに Java クラスが含まれる場合は、サーバー構成ファイル（`ORACLE_HOME\reports\conf\server_name.conf`）にある **engine** 要素の `classPath` プロパティを設定する必要があります。

ペーパー・レポートを配布するには：

1. レポート・ファイル (RDF、REP、XML または JSP) とその関連ファイル (PLL、PLX、参照イメージなど) を、アプリケーション・サーバーの配布ディレクトリに転送します。

注意： ファイルを転送するには、FTP や WebDAV などの、サポートされている方法を使用できます。

2. ファイルの転送先であるアプリケーション・サーバーのディレクトリが、Reports Server のアクセス・パスに含まれていることを確認してください。そうでない場合は、`REPORTS_PATH` 環境変数を使用するか、またはサーバー構成ファイルの Reports engine 要素の `sourceDir` プロパティを設定します。

13.3.2 ペーパー・レイアウトを使用したレポートの実行

ペーパー・レポートの配布が終了したら、Web ブラウザからレポートを実行できます。

たとえば、ブラウザで、「アドレス」フィールドに、次の URL を入力します。

```
http://your_web_server:port_num/reports/rwservlet?server=server_name&report=
&userid=&desformat=pdf&destype=cacheusername/password@my_dbmyreport.rdf
```

この例では `desformat=PDF` が指定されているため、目的のレポートがブラウザで PDF として表示されます。

ブラウザからのレポートの実行の詳細は、第 13.5 項「Web ブラウザからのレポート・リクエストの指定」を参照してください。

13.3.3 Web およびペーパーへの JSP レポートの配布

JSP レポートは 2 通りの方法で配布できます。1 つは既存の Oracle Reports アプリケーションを使用する方法で、もう 1 つはユーザー自身が作成した J2EE アプリケーションを使用する方法です。JSP ベースの Web レポートを開発およびテストする場合は、既存のアプリケーションを使用する方法が便利です。ただし、レポートの配布準備ができていない場合は、ユーザー自身が作成したアプリケーションを使用する方をお勧めします。

ペーパー・レイアウトと Web レイアウトの両方を使用した JSP レポートについて

Reports Builder では、ペーパー・レイアウトまたは Web レイアウト（もしくは両方）を使用して JSP レポートを作成できます。これらの各レポートは、次の異なるプロセスを使用して実行します。

- ペーパー・レイアウトを使用した JSP レポートは、Oracle Reports エンジンを通じて実行されます。
- Web レイアウトを使用した JSP レポートは、J2EE コンテナを通じて実行されます。

レポートが Java クラス（たとえば、バーコード・クラスや Web サービス・スタブなど）に依存する場合は、これらのクラスにアクセスするようにプロセスを構成する必要があります。つまり、ペーパー・レイアウトを使用した JSP レポートに Java クラスが含まれる場合は、サーバー構成ファイル (`ORACLE_HOME¥reports¥conf¥server_name.conf`) にある engine 要素の `classpath` プロパティを設定する必要があります。

Web レイアウトを使用した JSP レポートに Java クラスが含まれる場合は、WAR ファイルにクラスまたは JAR を追加するか、J2EE コンテナの `classpath` を変更します。詳細は、Oracle Application Server Containers for J2EE のドキュメントを参照してください。

注意： ペーパーおよび Web レイアウトを使用したレポート作成の例は、『Oracle Reports レポート作成ガイド』の「バーコードを使用したレポートの作成」を参照してください。簡単な JSP ベースの Web レポートの例は、『Oracle Reports チュートリアル』を参照してください。

この項の手順では、J2EE アプリケーションを使用して、ペーパーおよび Web レイアウトによる JSP レポートを配布する方法について説明します。ペーパーおよび Web レイアウトを使用して JSP レポートを配布するために、Oracle Application Server で新たに Oracle Reports J2EE アプリケーションを作成できます。このアプリケーションは、Oracle Application Server Containers for J2EE (OC4J) の既存インスタンスまたは新規インスタンスを使用して作成できます。

13.3.3.1 新規 J2EE アプリケーションの作成

この項では、Oracle Reports 用に新しい J2EE アプリケーションを作成する方法を説明します。アプリケーション情報を含む Web アプリケーション・アーカイブ (WAR) ファイルを作成して、それをエンタープライズ・アーカイブ (EAR) ファイルとしてデプロイします。J2EE アプリケーションを新たに作成するには、Oracle JDeveloper や他の Java 開発ツールを使用するか、または手動で作成することもできます。アプリケーションの作成に Oracle JDeveloper を使用しない場合は、アプリケーションと JSP レポートの両方に若干の変更を加える必要があります。

J2EE アプリケーションを作成するには：

注意： J2EE アプリケーションの作成に関する知識が十分でない場合は、Sun 社の Web サイト (<http://java.sun.com/j2ee>) を参照してください。Oracle JDeveloper の使用方法の詳細は、Oracle JDeveloper のオンライン・ヘルプを参照してください。

1. WAR ファイルを作成する前に、アプリケーションに、WEB-INF などのすべての必要なディレクトリと web.xml ファイルが含まれることを確認します。

注意： WEB-INF ディレクトリには、reports_tld.jar という名前の、Oracle Reports の JSP タグ・ライブラリが必要です。Oracle Developer Suite では、タグ・ライブラリは次のディレクトリにあります。

```
ORACLE_HOME¥reports¥j2ee¥reports_ids¥web¥WEB-INF¥lib
```

ORACLE_HOME は、Oracle Developer Suite がインストールされるディレクトリです。

Oracle Application Server では、タグ・ライブラリは次のディレクトリにあります。

```
ORACLE_HOME¥j2ee¥OC4J_BI_Forms¥applications¥reports¥web¥WEB_INF¥lib.
```

2. JSP ベースの Web レポートが、Oracle Reports の JSP タグ・ライブラリの位置を示していることを確認します。そうでない場合は、レポートは実行されません。

JSP タグ・ライブラリの場所を指定するには、JSP ファイルで taglib ディレクティブを使用します。

```
<%@ taglib uri="/WEB-INF/lib/reports_tld.jar" prefix="rw" %>
```

3. 手動または Oracle JDeveloper などのツールを使用して、新規 EAR ファイルを作成します。適切な J2EE フォーマットに従って、WAR ファイルを作成してください。

4. JSP レポートでペーパー・レイアウトが使用され、ペーパーとして配布する場合は、web.xml ファイルを開きます。

注意： Oracle Developer Suite では、web.xml ファイルは次のディレクトリにあります。

ORACLE_HOME/reports/j2ee/reports_ids/web/WEB-INF

Oracle Application Server では、web.xml ファイルは次のディレクトリにあります。

ORACLE_HOME/j2ee/OC4J_BI_Forms/applications/reports/web/WEB-INF

Web レイアウトのみを使用した JSP レポートを配布する場合は、手順 7 に進みます。

5. web.xml ファイルに、次のコードを追加します。

```
<servlet>
  <servlet-name>rwervlet</servlet-name>
  <servlet-class>oracle.reports.rwclient.RWClient</servlet-class>
  <load-on-startup>yes</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>rwervlet</servlet-name>
  <url-pattern>/rwervlet*</url-pattern>
</servlet-mapping>
```

この新しい定義により、/rwervlet で開始するすべての URL が、定義したサーブレットにリダイレクトされます。

注意： サーブレット名および URL は変更できます。

6. web.xml ファイルを保存します。
7. WAR ファイルから EAR ファイルを作成します。これらのファイルをコンパイルした後で、保存先を書き留めます。

13.3.3.2 OC4J を使用した J2EE アプリケーションのデプロイ

WAR ファイルと EAR ファイルを作成したら、それらを Oracle Application Server にデプロイできます。これにより、アプリケーションは Web で使用可能になります。これらのファイルは、Oracle Enterprise Manager 10g を使用してデプロイできます。この際、OC4J インスタンスは既存または新規のどちらでもかまいません。

この項では、J2EE アプリケーションをデプロイする、次の 2 通りの方法について説明します。

- [既存の OC4J インスタンスを使用した J2EE アプリケーションのデプロイ](#)
- [新規 OC4J インスタンスを使用した J2EE アプリケーションのデプロイ](#)

13.3.3.2.1 既存の OC4J インスタンスを使用した J2EE アプリケーションのデプロイ

1. 第 13.3.3.1 項「[新規 J2EE アプリケーションの作成](#)」で説明されているとおりに、J2EE アプリケーションが作成されていることを確認します。
2. Oracle Enterprise Manager 10g で、中間層の詳細ページを表示します。
3. 「システム・コンポーネント」で、J2EE アプリケーションをデプロイする OC4J インスタンスをクリックします。
4. 「OC4J インスタンス」ページで「[アプリケーション](#)」をクリックします。

- 「アプリケーション」タブ・ページの「デプロイ済アプリケーション」で「**EAR ファイルのデプロイ**」をクリックし、[第 13.3.3.1 項「新規 J2EE アプリケーションの作成」](#)で作成した EAR ファイルをデプロイします。
- 「アプリケーションのデプロイ」ウィザードの最初のページで、「**参照**」をクリックし、デプロイする J2EE アプリケーション (EAR ファイル) を選択するか、作成した EAR ファイルの場所を入力します。
- 「アプリケーション名」に、このアプリケーションの一意のアプリケーション名を指定します。たとえば、MyReportApp のように指定します。
- 「親アプリケーション」リストから親アプリケーションを選択し、「**続行**」をクリックします。
- 「URL マッピング」ページの「URL マッピング」フィールドのテキストが、新規アプリケーションにアクセスする際にユーザーが入力する名前であることを確認します。
- アプリケーション名は URL アドレスの一部であるため、「URL マッピング」フィールドで、アプリケーション名の前にスラッシュ (/) を追加します。次に例を示します。

```
/MyReportApp
```

- 「**終了**」をクリックします。
- 次のページで、「**デプロイ**」をクリックします。
- 表示される詳細ページで、作成したアプリケーション (MyReportApp) が「デプロイ済アプリケーション」の一覧にあることを確認します。
- アプリケーション名 (MyReportApp) をクリックします。
- 「アプリケーション」ページの「プロパティ」で、「**一般**」をクリックします。
- 「ライブラリ・パス」で、rwrun.jar と zrclient.jar が含まれているかどうかを確認します。含まれていない場合は、「**行の追加**」をクリックし、次のパスを追加します。

Windows の場合：

```
%ORACLE_HOME%\reports\jlib\rwrun.jar  
%ORACLE_HOME%\jlib\zrclient.jar
```

UNIX の場合：

```
$ORACLE_HOME/reports/jlib/rwrun.jar  
$ORACLE_HOME/jlib/zrclient.jar
```

- 「**適用**」をクリックし、「**OK**」をクリックします。
- 「**停止**」→「**起動**」をクリックし、新しいライブラリ・パスが有効になるように OC4J インスタンスを再起動します。

13.3.3.2.2 新規 OC4J インスタンスを使用した J2EE アプリケーションのデプロイ

OC4J インスタンスに J2EE アプリケーションをデプロイするには、最初に OC4J インスタンスを作成してから、J2EE アプリケーションをデプロイします。

新規 OC4J インスタンスを作成するには

- [第 13.3.3.1 項「新規 J2EE アプリケーションの作成」](#)で説明されているとおりに、J2EE アプリケーションが作成されていることを確認します。
- Oracle Enterprise Manager 10g で、中間層の詳細ページを表示します。
- 「**OC4J インスタンスの作成**」をクリックします。
- OC4J インスタンスの名前を入力します。
- 「**作成**」をクリックします。
- OC4J インスタンスを作成した後、「**OK**」をクリックします。

J2EE アプリケーションをデプロイするには

1. 「Application Server」 ページの「システム・コンポーネント」に、新規 OC4J インスタンスが表示されていることを確認します。

次に、Reports Server への接続とセキュリティ統合をサポートするために、OC4J インスタンスを手動で構成します。

2. 既存の OC4J インスタンスから `oc4j.properties` ファイル（たとえば、OC4J_BI_FORMS インスタンスであれば、`ORACLE_HOME/j2ee/OC4J_BI_FORM/config/oc4j.properties`）の次のプロパティとその定義を、新規 OC4J インスタンスの `oc4j.properties` ファイル（`ORACLE_HOME/j2ee/your_application/config/oc4j.properties`）にコピーします。

- `oracle.home`
- `oracle.path`

3. `ORACLE_HOME/opmn/conf` ディレクトリの `opmn.xml` ファイルで、`PATH` (Windows) または `LD_LIBRARY_PATH` (UNIX) を新規 OC4J インスタンスに追加します。

- a. `ORACLE_HOME/opmn/conf/opmn.xml` で、新規 OC4J インスタンスを記述する XML 要素を検索します。

```
<process-type id="New OC4J Instance Name" module-id="OC4J">
```

- b. 同じ `opmn.xml` ファイルの既存の OC4J インスタンスに関する記述から、`PATH` (Windows) または `LD_LIBRARY_PATH` (UNIX) モジュール・データ・プロパティに関する記述を含む `environment` タグをコピーして追加します。たとえば、UNIX では次のようになります。

```
<environment>
  <variable id="LD_LIBRARY_PATH"
    value="$ORACLE_HOME/lib:$ORACLE_HOME/network/lib:
      $ORACLE_HOME/jdk/jre/lib/sparc"/>
</environment>
<category id="start-parameters">
  <data id="java-options"
    value="-server -Djava.security.policy=
      $ORACLE_HOME/j2ee/OC4J_BI_Forms/config/java2.policy
      -Djava.awt.headless=true -Xmx512M"/>
  <data id="oc4j-options" value="-properties -userThreads "/>
</category>
<category id="urlping-parameters">
  <data id="MyReportsApp*/rwservlet/pingservlet?start=auto" value="200"/>
</category>
<dependencies>
  <database infrastructure-key="portal"/>
  <managed-process process-type="HTTP_Server"
    process-set="HTTP_Server" ias-component="HTTP_Server"
    autostart="true"/>
</dependencies>
```

***MyReportApp** は、Oracle Reports 用に新規に作成した Web アプリケーションを示します。

- c. OC4J インスタンスを再起動します。
4. 第 13.3.3.2.1 項「既存の OC4J インスタンスを使用した J2EE アプリケーションのデプロイ」の手順を実行します。

13.3.4 JSP ベースの Web レポートのブラウザからの実行

JSP レポートが Web レポートの場合は、作成した JSP レポートを Web ブラウザから実行できます。たとえば、ブラウザで、「アドレス」フィールドに、次の URL を入力します。

```
http://your_computer_name:port/MyReportApp/JSPreportname.jsp?userid=user
ID/password@database_name
```

注意： この URL では、MyReportApp が作成したアプリケーションの名前です。

この時点で JSP ベースの Web レポートを変更する場合は、次のいずれかを実行してください。

- 前述の位置で、レポートを置換します。
- 変更した JSP ベースの Web レポートで WAR ファイルを再作成し、アプリケーションをデプロイし直します。詳細は、[第 13.3.3.1 項「新規 J2EE アプリケーションの作成」](#)を参照してください。

ブラウザからのレポートの実行の詳細は、[第 13.5 項「Web ブラウザからのレポート・リクエストの指定」](#)を参照してください。

13.3.5 ペーパー・レイアウトを使用した JSP レポートの実行

JSP レポートがペーパー・レイアウトの場合は、次の URL を使用して、ブラウザから JSP レポートを実行できます。

```
http://your_web_server:portnum/MyReportApp/rwservlet?report=myreport.jsp&userid=
username/password@my_db&server=server_name&desformat=pdf&destype=cache
```

この例では desformat=PDF が指定されているため、目的のレポートがブラウザで PDF として表示されます。

ブラウザからのレポートの実行の詳細は、[第 13.5 項「Web ブラウザからのレポート・リクエストの指定」](#)を参照してください。

13.3.6 UNIX で WE8MSWIN1252 キャラクタ・セットを使用したレポートの実行

UNIX フォントには、WE8MSWIN1252 キャラクタ・セットが設定されていません。そのため、NLS_LANG=AMERICAN_AMERICA.WE8MSWIN1252 で Oracle Reports を実行すると、正しく表示されない可能性があります。インストールされたフォントのコード・ページ (Tk2Motif.rgb ファイルで定義) を WE8MSWIN1252 キャラクタ・セットにマップする必要があります。TK2Motif.rgb ファイルは、ORACLE_HOME/guicommon/tk/admin/ ディレクトリにあります。

注意： このマッピングが必要なのは、Reports Builder、非バッチ・モード (BATCH=NO) の Reports Converter、REPORTS_DEFAULT_DISPLAY=NO に設定された Reports Server および Reports Runtime です。Reports Server と Reports Runtime では、REPORTS_DEFAULT_DISPLAY を使用して必要なフォントが判断されます。

例 1

Tk2Motif*fontMapCs: ISO8859-1=WE8MSWIN1252 (ISO8859-1 フォントがインストールされている場合)

13.4 OracleAS Portal でのレポートの公開

レポートを公開する最適な方法の 1 つが、OracleAS Portal のセキュリティ保護された宣言型インタフェースを使用する方法です。OracleAS Portal でレポートを公開する場合は、第 12 章「OracleAS Portal でのレポートの配布」を参照してください。その場合は、最初に Oracle Reports コンポーネントを OracleAS Portal で登録してから（第 12.2 項「Oracle Reports コンポーネントの登録」を参照）、レポートをポータルに公開します（第 12.3 項「レポートのポータルレットとしての公開」を参照）。

注意： OracleAS Portal Security、Portal Destination、Job Status Repository などの機能を使用すると、OracleAS Reports Services によって確立された JDBC データベース接続が NLS_LANG の初期設定よりも優先されることがあります。これにより、PDF での双方向出力などの、実行中のレポートの動作に影響が及ぶ場合があります。UNIX プラットフォームでは、環境切替え機能を使用して、レポートの環境を動的に設定することで、この問題を解決できます。詳細は、第 3.2.2 項「動的な環境切替え」を参照してください。

13.5 Web ブラウザからのレポート・リクエストの指定

レポート・リクエストに必要な URL 構文をユーザーに提供できます。あるいは、URL 構文をハイパーリンクとして Web ページに追加することもできます。

URL 構文は、次の形式で提示できます。

- 完全な URL リクエスト。次に例を示します。

```
http://your_webserver.domain_name:port/alias/rwervlet?report=myreport.rdf&userid=username/password@my_db&server=server_name&desformat=html&destype=cache
```

他のコマンドライン・キーワードが必要な場合は、rwclient の有効なコマンドライン・キーワードのリストが掲載されている付録 A「コマンドライン・キーワード」を参照してください。

- キー・マッピングを使用して単純化した URL リクエスト。次に例を示します。

```
http://your_webserver.domain_name:port/alias/rwervlet?key1
```

13.6 URL エンジンへのリクエスト送信

Reports Server の URL エンジンを実態にしている場合は、次のコマンドライン・オプションを使用してジョブ・リクエストを URL エンジンに送信できます。

- urlParameter は、キャッシュに取り込まれる URL を指定します。たとえば、http://www.oracle.com や JSP レポートなどです。
- jobType は、URL エンジンに関連付けられているサーバー構成ファイルでのジョブ・タイプの名前 (rwurl など) です。

注意： URL エンジンのアクティブ化の詳細は、第 3.5 項「URL エンジンの構成」を参照してください。

たとえば、urlParameter に外部 URL を指定するリクエストは、次のようになります。

```
http://your_webserver:portnum/reports/rwervlet?server=ReportsServer+jobType=rwurl+urlParameter="http://www.oracle.com"+destype=mail+desname=foo@bar.com+desformat=htmlcss
```

また、urlParameter に JSP レポートを指定するリクエストは、次のようになります。

```
http://your_webserver:portnum/reports/rwservlet?server=
ReportsServer+jobType=rwurl+destype=cache+urlParameter=
"http%3A%2F%2Flocalhost%2Ffoo.jsp%3Fuserid%3Dscott%2Ftiger@oraDB%3Fserver%3DreportsServ
er"
```

注意： URL に特殊文字が含まれている場合は、
x-www-form-urlencoded 形式でその URL をエンコードする必要があります。

13.7 Web サービスを介したレポートの実行

多くの場合、レポートは、スタンドアロンのアプリケーションではなく、より大規模なアプリケーションを統合して実行されます。したがって、アプリケーション内でレポート・リクエストを生成できれば便利です。この目標は、OracleAS Reports Services を Web サービスとして公開することで達成できます。Web サービスは、Web サービスをサポートする任意の環境 (Java アプリケーションなど) からコールできます。たとえば、システムに Java ベースの経費報告書フォームがあり、ユーザーが経費報告書フォームへの記入を完成させると、毎回その PDF パージョンを作成できるようにするとします。Java プロキシ対応の Oracle Reports Web サービスを作成することで、Java 開発環境 (たとえば、Oracle JDeveloper) からそれを参照し、PDF ファイルを生成する OracleAS Reports Services を起動するボタンを簡単に追加できます。

関連項目： Oracle Reports Web サービスの詳細と、サンプル・プロキシと Java クライアントのインストールおよび使用方法の詳細は、[第 14 章「Oracle Reports Web サービスの使用法」](#)を参照してください。

13.8 Oracle Workflow からのレポートの実行

Oracle Workflow からレポートを実行する詳細は、[第 3.8 項「Oracle Reports と Oracle Workflow の通信の構成」](#)を参照してください。また、OTN (<http://www.oracle.com/technology/products/reports/features/workflow>) にあるホワイト・ペーパー『Integrating Oracle Workflow with Oracle Reports』も参照してください。

13.9 自動実行のためのレポートのスケジューリング

サーバーを使用して、Reports Queue Manager (Windows の場合は rwrqm、Solaris の場合は rwrqv.sh) または OracleAS Portal から、あるいは SCHEDULE コマンドライン・キーワードによって、レポートを自動的に実行することができます。スケジューリング機能では、レポートを実行する時刻と頻度を指定できます。

レポートのスケジューリングの詳細は、Reports Queue Manager のオンライン・ヘルプを参照してください。

レポートを OracleAS Portal ページでポータル・コンポーネントとして公開する場合は、レポート・リクエストが自動的に実行されるようにスケジューリングし、出力されたレポートを指定のページに送信できます。詳細は、OracleAS Portal のオンライン・ヘルプを参照してください。

SCHEDULE コマンドライン・キーワードは、rwclient、rwservlet、rwcgi の各コマンドで使用できます。詳細は、[第 A.3.94 項「SCHEDULE」](#)を参照してください。

13.10 追加のパラメータ

Reports Server にリクエストを送信すると、次に示す追加のパラメータがリクエストとともに暗黙的に送信されます。これらのパラメータの値は変更できません。

表 13-5 レポート・リクエストとともに渡される追加のパラメータ

名前	説明
ACCEPT_LANGUAGE	ブラウザ / ユーザーが受け付ける言語をカンマで区切ったリスト。
REMOTE_ADDR	リクエストの送信元であるリモート IP アドレス。
REMOTE_HOST	リクエストの送信元であるリモート・ホスト名。
SCRIPT_NAME	実行されるスクリプトの仮想パス。
SERVER_NAME	Reports Servlet が実行されているサーバーのホスト名または IP アドレス。
SERVER_PORT	Reports Servlet が実行されているサーバーのポート番号。
SERVER_PROTOCOL	リクエストの送信に使用された情報プロトコルの名前とリビジョン。
USER_AGENT	リモート・クライアントのブラウザの説明。

13.11 キャッシュからのレポート出力の再使用

レポートを実行すると、レポート出力のコピーが OracleAS Reports Services のキャッシュに保存されます。その後で、同じキャッシュ・キーを使用する同一のレポートが実行されると、現行リクエストは重複ジョブとして認識されます。

次に、Reports のキャッシュ機能の効果的な使用例をいくつか紹介します。

- 新規のジョブ・リクエスト "A" が Reports Server に送信された時点で、別のジョブ "B" が現行ジョブ・キュー内で同じキャッシュ・キーを使用している（エンジンが利用可能になるのを待っているか、または実行中）場合は、ジョブ "A" ではジョブ "B" からの出力が使用されます。

ジョブのキャッシュ・キーには、`destype`、`desname`、`server`、`tolerance` の各パラメータを除くほとんどすべてのパラメータが含まれています。

このレベルのキャッシュは、自動的に行われます。コマンドラインにその他のパラメータを指定する必要はありません。

- ユーザーが新規のジョブ・リクエスト "A" で `TOLERANCE=n` (n は分単位) を指定した場合、Reports Server は、終了したジョブ・キュー内で n 分以内に正常終了したジョブの検索を試行します。Reports Server によってこのようなジョブが検出された場合は、新規のジョブ・リクエスト "A" ではジョブ "B" の出力が返されます。

注意： 詳細は、第 A.3.113 項「[TOLERANCE](#)」を参照してください。

OracleAS Reports Services のキャッシュの結果は永続的です。Reports Server を停止して再び起動すると、以前のキャッシュの結果がすべて復元され、再び使用できるようになります。

13.11.1 使用上の注意

キャッシュのサイズは、Reports Queue Manager (Windows の場合は `rwrqm`、Solaris の場合は `rwrqv.sh`) またはサーバー構成ファイル (`server_name.conf`) の `cache` 要素を通じて設定できます。Reports Server は、キャッシュ・ファイルの合計サイズを、設定されたサイズ以下に保つために、最も古いキャッシュ・ファイルを削除します。さらに、Reports Queue Manager を使用してキャッシュを空にすることもできます。

キャッシュの設定の詳細は、Reports Queue Manager のオンライン・ヘルプおよび第 3 章「[OracleAS Reports Services の構成](#)」を参照してください。

13.12 キー・マップ・ファイルの使用法

URL をユーザーに提供する場合、あるいはハイパーリンクを Web サイトに追加する場合は、キー・マップ・ファイルを使用して、URL リクエスト内のパラメータを単純化したり非表示にしたりできます。

キー・マップ・ファイルには、レポートを実行するコマンド文字列が含まれており、それぞれのコマンド文字列の先頭には一意のキー識別子が付加されています。レポートを JSP として実行する場合を除いて、ランタイム URL でこのキーのみを参照することができます。このキー値はサーバーまたはサーブレットによってマップ・ファイル (cgicmd.dat) に送信され、このファイルから指定のキーに関連付けられたコマンドがサーバーまたはサーブレットに返されて処理が進められます。キー・マッピングを使用することによって、すべてのコマンドライン・オプションを非表示にできます。

キー・マッピングは次の場合に便利です。

- URL を短縮して、使いやすくする場合
- 元の URL を変更することなく、ランタイム・コマンドをマッピングし直す場合
- 企業にとって典型的ないくつかの実行構成を標準化する場合
- ユーザーからパラメータの一部 (データベース接続文字列など) を非表示にする場合
- ユーザーがレポートの実行に使用できるパラメータを制限する場合

キー・マップ・ファイル (cgicmd.dat) からキー名を指定するときは、そのキー名をレポート・リクエスト URL の問合せ文字列の開始位置 (疑問符の後) に配置する必要があります。ただし、cmdkey コマンドライン・キーワードの値としてキー名を指定する場合 (CMDKEY=keyname) は例外です。この場合は、レポート・リクエスト URL における問合せ文字列の任意の位置にキー名を配置できます。CMDKEY キーワードは、JSP として実行されるジョブおよび Reports Servlet コマンドで使用できます。

注意： 詳細は、第 A.3.14 項「CMDKEY」を参照してください。

13.12.1 キー・マッピングを使用可能にする方法

キー・マッピングは、次のいずれかの条件が満たされたときに使用可能となります。

- 標準のファイル名 (cgicmd.dat) を持つ有効なファイルが、Web サーバー・マシン (Windows または UNIX) 上の `ORACLE_HOME\reports\conf` ディレクトリ (デフォルトの位置) にある。
- Reports Servlet 構成ファイル (rwservlet.properties) の KEYMAPFILE パラメータの下に、有効なキー・マップ・ファイルが入力されている。
- rwcgi が使用される場合に、Web サーバー上の REPORTS_CGIMAP 環境変数に有効なキー・マップ・ファイルの名前が指定されている。詳細は、付録 B「環境変数」を参照してください。

13.12.2 キー・マップ・ファイルへのキー・マッピング・エントリの追加

キー・マップ・ファイルにキー・マッピング・エントリを追加する手順は、次のとおりです。

1. Reports Server をホスティングするマシン上の cgicmd.dat ファイルにナビゲートし、これをテキスト・エディタで開きます。

このファイルは、Windows と UNIX のどちらの場合も、次のディレクトリにあります。

```
ORACLE_HOME\reports\server\conf\cgicmd.dat
```

2. キー・マッピング・エントリを追加します。次に例を示します。

```
key1: report=your_report.rdf userid=username/password@my_db desformat=html
SERVER=server_name destype=cache
```

この例の key1 は、キーの名前です。

ファイル自体に記述されている特殊なパラメータを除いて、コマンドライン・オプションは `rwclient` の構文規則に従います。詳細は、付録 A 「コマンドライン・キーワード」を参照してください。

3. Web ページでハイパーリンクを追加または更新します。

詳細は、第 13.5 項「Web ブラウザからのレポート・リクエストの指定」を参照してください。

13.12.3 JSP 以外のすべてでキーを使用する場合

レポート・リクエスト URL にキー名を指定するときは、そのキー名が問合せ文字列内の最初の値（疑問符の直後）になるように配置する必要があります。次に例を示します。

```
http://.../rwservlet?keyname
```

次に示すのは、パラメータ・フォームを使用した制限付きの実行に対するキー・マッピングの例です。

URL は次のようになります。

```
http://web_server.domain_name:port/cgi-bin/rwcgi.exe?key&par1&par2&parN
```

キー・マップ・ファイルには、次のような行が含まれます。

```
KEY: REPORT=myreport DEPTNO=%1 MYPARAM=%2 %*
```

これによって、次のコマンドライン・リクエストに相当するリクエストが生成されます。

```
rwclient REPORT=myreport DEPTNO=par1 MYPARAM=par2 parN
```

使用上の注意

`rwservlet` URL では、最初のオプション（疑問符の後の最初の情報）が、名前 / 値のペアの一部でないかぎり、キーとして扱われます。最初のオプションが名前 / 値のペア（つまり、`keyword=value`）でない場合は、`cgicmd.dat` キー・エントリのかわりにコマンドライン全体が使用されます。

13.12.4 JSP としてのレポート実行でのキーの使用方法

レポートを JSP として実行する場合に、`cgicmd.dat` ファイル内のコマンド・キーをコールするときは、URL で `cmdkey` キーワードを使用する必要があります。たとえば、JSP URL は次のようになります。

```
http://.../myreport.jsp?cmdkey=key
```

注意： `rwservlet` コマンドで `cmdkey` を使用することもできます。

JSP または `rwservlet` で `cmdkey` を使用する場合は、問合せ文字列内の任意の位置に `cmdkey` を配置できます。次に例を示します。

```
http://.../example.jsp?parameter1=value1&cmdkey=keyname
http://.../rwservlet?parameter1=value1&cmdkey=keyname
```

使用上の注意

キー・マッピングの使用時に、パラメータが URL からキーに置換される順序は、URL の `cmdkey` の配置により決定されます。たとえば、`cgicmd.dat` ファイルに次のようなキーがあるとします。

```
mykeys: DEPTNO=%1 MYPARAM=%2
```

このキーを参照する JSP レポートを次のように実行します。

```
http://neptune.world.com:80/jsp/myreport.jsp?userid=scott/tiger@hrdb  
&cmdkey=mykeys&10&test
```

この URL での `cmdkey` の配置により、`10` は `%1` に対応し、`test` は `%2` に対応します。これらが URL で 1 番目と 2 番目のパラメータではなくても、URL の `cmdkey` に従い、`10` と `test` は 1 番目と 2 番目のパラメータになります。この例では、URL は次のようになります。

```
http://neptune.world.com:80/jsp/myreport.jsp?userid=scott/tiger@hrdb  
&DEPTNO=10&MYPARAM=test
```

Oracle Reports Web サービスの使用方法

Web サービスは、標準のインターネット技術と XML 技術を基に構築されたアプリケーションで、次の特徴があります。

- XML を使用して定義および記述されたパブリック・インタフェースとバインディングを提供します。
- 他のプログラムから使用できるように、これらのパブリック・インタフェースとバインディングをネットワーク全体に公開します。

Web サービスではリクエストを受信すると、そのリクエストに基づいて適切な機能が実行され、レスポンスを返します。リクエストとレスポンスは同一の操作で実行される場合と、別の操作として実行される場合があります。後者の場合、サービス利用者はレスポンスを待つ必要がありません。通常、リクエストとレスポンスには、ポータブルなデータ交換フォーマットである XML を使用し、HTTP などの通信プロトコルによって伝達されます。

Web サービス・トランザクションの多くは、ビジネス組織間でやり取りされます。特定の Web サービスの提供者であるビジネス組織は、別の Web サービスの利用者である場合もあります。軽量プロトコル経由で Web サービス・プロバイダに接続している Thin クライアントなどのクライアント・デバイスが、Web サービス利用者となる場合もあります。

この章では、Oracle Reports Web サービスについて説明します。この章の主な項は、次のとおりです。

- [概要](#)
- [はじめに](#)
- [Oracle Reports Web サービスの操作](#)
- [サンプル・プロキシと Java クライアントのインストールと使用方法](#)

14.1 概要

Oracle Reports には、ジョブ・リクエストをサーバー・インフラストラクチャに送信して処理するための手段が複数用意されています。

- **rwsvrlet**
rwsvrlet は、HTTP と Reports Server 間で、ジョブ・リクエストを変換および伝達します。Web ブラウザやイベントドリブンの公開 API を使用して送信する場合などです。
- **rwcgi**
rwcgi は、HTTP と Reports Server 間で、ジョブ・リクエストを変換および伝達します。Web ブラウザやイベントドリブンの公開 API を使用して送信する場合などです。rwcgi は、下位互換性を確保するためにのみ維持されています。
- **rwclient**
rwclient は、コマンドラインを解析して送信し、リモートの Reports Server でレポートを実行します。
- **Oracle Forms**
Oracle Forms は、Rapid Application Development (RAD) ツールで、高度なスケーラビリティを持つインターネット・データベース・アプリケーションの構築に使用します。

Oracle Reports テクノロジをカスタム・アプリケーション、特に Java アプリケーションと統合するには、rwsvrlet、rwcgi、rwclient および Oracle Forms を使用して、アプリケーションからサーバーにジョブを送信するメカニズムを実装する必要があります。

Web サービスとしての公開と動作に必要なパブリック・インタフェースとバインディングは、RWWebService サブレットにより提供されます。この機能を使用して、アプリケーション開発者は、各自のアプリケーションに Oracle Reports を統合できます。

14.2 はじめに

この項では、次の操作に必要な手順について説明します。

- [RWWebservice サブレットの起動](#)
- [WSDL の表示](#)

14.2.1 RWWebservice サブレットの起動

RWWebService サブレットを起動する手順は、次のとおりです。

1. Oracle Reports インスタンスが常駐する Oracle Application Server Containers for J2EE (OC4J) インスタンスを起動します。

2. ブラウザの「アドレス」フィールドに、次の URL を入力します。
`http://yourwebserver:port/reports/rwwebservice`

これにより、RWWebService のエンドポイントに進みます。RWWebService のエンドポイント・ページでは、次のことを実行できます。

- a. Oracle Reports Web サービスの WSDL を表示します。
- b. Web ベースの UI を使用して任意の RWWebService コマンドを実行します。
- c. プロキシ JAR ファイルとプロキシ・ソースをダウンロードし、サンプル Java クライアントを使用して Oracle Reports Web サービスを起動します。

14.2.2 WSDL の表示

Web Service Description Language (WSDL) は、利用可能なサービスを、ドキュメント指向またはプロシージャ指向のいずれかの情報を含むメッセージを操作する一連のエンドポイントとして記述する XML フォーマットです。操作とメッセージは抽象的に記述され、エンドポイントを定義するために、具体的なネットワーク・プロトコルとメッセージ・フォーマットにバインドされます。

注意： Oracle Reports Web サービスは、Universal Description Discovery and Integration (UDDI) サーバーへの公開による、WSDL の動的ディスカバリには対応していません。

1. RWWebService の Web ページの「**Service Description**」リンクをクリックし、Oracle Reports Web サービスの WSDL ドキュメントを表示します。

注意： WSDL XML 出力を表示するには、Internet Explorer を使用してください。Netscape (7.2 以降) の場合は、ページを .xml ファイルとして保存し、Internet Explorer を使用してそのファイルを開く必要があります (たとえば、rwwebservice.xml)。

2. WSDL の最後のエントリはサービスの説明で、ここには Web サービスの位置が記述されません。

```
<soap:address location="http://yourwebserver:8888/reports/rwwebservice" />
```

図 14-1 WSDL の表示

RWWebService endpoint

WSDL for Service: RWWebService, generated by O 1.1)

For a formal definition, please review the [Service Description](#) (*rpc style*).

RWWebService service

The following operations are supported.

- [getServerInfo](#)
- [getJobInfo](#)
- [killJob](#)
- [runJob](#)
- [getAPIVersion](#)

oc4j client

The java proxy is packaged in a .jar either as classes or sources files.

- [Proxy Jar](#)
- [Proxy Source](#)

`http://yourwebserver:port/reports/rwwebservice` で、URL とポート番号が正しく定義されていることを確認します。

注意： 指定されているホスト名は、OC4J インスタンスが動作しているホスト名で、Reports Server が動作しているホスト名ではありません。

URL が正しくない場合は、次の手順を実行します。

1. OC4J を停止します。
2. `ORACLE_HOME¥j2ee¥OC4J_INSTANCE_NAME¥application-deployments¥reports¥web¥temp¥` ディレクトリにある `__java_stateless_rpc` を削除します。
3. OC4J を再起動します。
4. 次のように、URL 定義が正しく定義されていることを確認します。
`http://yourwebserver:port/reports/rwwebservice`

Oracle Reports WSDL

```
<?xml version="1.0" encoding="UTF-8" ?>
<definitions name="RWWebService"
targetNamespace="http://oracle.reports.rwclient/RWWebService.wsdl"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:tns="http://oracle.reports.rwclient/RWWebService.wsdl"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
  <documentation>WSDL for Service: RWWebService, generated by Oracle WSDL toolkit
(version: 1.1)</documentation>
  <types>
    <schema targetNamespace="http://oracle.reports.rwclient/RWWebService.xsd"
xmlns:tns="http://oracle.reports.rwclient/RWWebService.xsd"
xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" />
  </types>
  <message name="runJobInput">
    <part name="param0" type="xsd:string" />
    <part name="param1" type="xsd:boolean" />
  </message>
  <message name="getServerInfoOutput">
    <part name="output" type="xsd:string" />
  </message>
  <message name="getAPIVersionInput" />
  <message name="getAPIVersionOutput">
    <part name="output" type="xsd:string" />
  </message>
  ...
  <portType name="RWWebServicePortType">
    <operation name="getServerInfo">
      <input message="tns:getServerInfoInput" />
      <output message="tns:getServerInfoOutput" />
    </operation>
    <operation name="getJobInfo">
      <input message="tns:getJobInfoInput" />
      <output message="tns:getJobInfoOutput" />
    </operation>
    ...
  </portType>
  <service name="RWWebService">
    <port name="RWWebServicePort" binding="tns:RWWebServiceBinding">
      <soap:address location="http://localhost:8888/reports/rwwebservice" />
    </port>
  </service>
</definitions>
```

14.3 Oracle Reports Web サービスの操作

Oracle Reports では、RWWebService サブレットが、XML を使用して定義および記述されたパブリック・インタフェースとバインディングを持つ Web サービスとして公開されます。これらのパブリック・インタフェースとバインディングは、WSDL を介してネットワーク全体に発行されます。

RWWebService のエンドポイントでは、次の操作がサポートされています。

- [getAPIVersion](#)
- [getServerInfo](#)
- [getJobInfo](#)
- [killJob](#)
- [runJob](#)

14.3.1 getAPIVersion

getAPIVersion() 操作では、Reports Server のバージョンの詳細を XML フォーマットで返します。この操作にはパラメータは必要ありません。

注意： getAPIVersion は、SOAP レスポンス全体を結果とともに（文字列で）返す唯一の操作です。runJob などの他の操作では、SOAP レスポンス内に埋め込まれた XML ブロックとしてレスポンスが返されます。

getAPIVersion レスポンスを表示する手順は、次のとおりです。

1. 「getAPIVersion」リンクをクリックします。「テスト」ページが表示されます。このページにはパラメータはなく、リクエストを送信する「**起動**」ボタンのみがあります。
2. 「Invoke」をクリックします。SOAP レスポンスが、新しいウィンドウに表示されます。

次は、getAPIVersion 操作のレスポンス例です。

```
<?xml version="1.0" encoding="UTF-8" ?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:getAPIVersionResponse
      xmlns:ns1="http://oracle.reports.rwclient/RWWebService.wsdl"
      SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <return xsi:type="xsd:string">10.1.2.x.x</return>
    </ns1:getAPIVersionResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

14.3.2 getServerInfo

getServerInfo(String serverName, String authId) 操作では、2つのパラメータを使用し、Reports Server の情報を XML フォーマットで返します。

有効なパラメータは、次のとおりです。

- **serverName:** NULL でない有効なサーバー名です。指定されたサーバーがネットワークで稼動していない場合、この操作によってエラーが発生します。
- **authId:** ユーザー名 / パスワード形式の文字列で、セキュリティ保護されたサーバーに指定する必要があります。セキュリティ保護されていないサーバーの場合、このパラメータは無視されます。

getServerInfo レスポンスを表示する手順は、次のとおりです。

1. 「getServerInfo」リンクをクリックします。「テスト」ページに関連するパラメータ・フィールドとリクエストを送信するための「起動」ボタンが表示されます。
2. Reports Server 名 (param0) と authId (param1) を入力します。
3. 「Invoke」をクリックします。SOAP レスポンスが、新しいウィンドウに表示されます。

次は、getServerInfo 操作の出力例です。

```
<?xml version = '1.0' encoding = 'ISO-8859-1' standalone = 'yes'?>
<serverInfo name="repsevr" version="10.1.2.x.x">
  <host>incq246bc</host>
  <processId>2588</processId>
  <startTime>27-May-2003 10:09:34</startTime>
  <queue maxQueueSize="1000"/>
  <engine id="rwEng" activeEngine="1" runningEngine="0"/>
  <engine id="rwURLEng" activeEngine="1" runningEngine="0"/>
  <performance>
    <property name="successfulJobs" value="6"/>
    <property name="currentJobs" value="0"/>
    <property name="futureJobs" value="0"/>
    <property name="transferredJobs" value="0"/>
    <property name="failedJobs" value="0"/>
    <property name="AverageResponseTime" value="2124"/>
    <property name="executionTimeToDate" value=""/>
  </performance>
</serverInfo>
```

14.3.3 getJobInfo

getJobInfo(Integer jobId, String serverName, String authId) 操作では、ジョブ情報を XML フォーマットで返します。

有効なパラメータは、次のとおりです。

- *jobId*: 情報が必要なジョブの Job ID です。
- *serverName*: NULL でない有効なサーバー名の値を指定する必要があります。指定されたサーバーがネットワークで稼動していない場合、この操作によってエラーが発生します。
- *authId*: ユーザー名 / パスワード形式の文字列で、セキュリティ保護されたサーバーに指定する必要があります。セキュリティ保護されていないサーバーの場合、このパラメータは無視されます。

getJobInfo レスポンスを表示する手順は、次のとおりです。

1. 「getJobInfo」リンクをクリックします。「テスト」ページに関連するパラメータ・フィールドとリクエストを送信するための「起動」ボタンが表示されます。
2. jobId (param0)、Reports Server 名 (param1) および authId (param2) を入力します。
3. 「Invoke」をクリックします。SOAP レスポンスが、新しいウィンドウに表示されます。

次は、job id=3 を指定した場合の getJobInfo 操作の出力例です。

```
<?xml version = '1.0' encoding = 'ISO-8859-1' standalone = 'yes'?>
<serverQueues>
  <job id="3" queueType="past">
    <name>test.rdf</name>
    <type>report</type>
    <status code="4">Finished successfully</status>
    <owner>RWUser</owner>
    <server>repsevr</server>
    <destination>
      <desType>cache</desType>
```

```

        <desFormat>html</desFormat>
        <file>21748116.htm</file>
        <file>217481161.jpg</file>
        <file>217481160.jpg</file>
    </destination>
    <timingInfo>
        <queued>27-May-2003 10:21:50</queued>
        <started>27-May-2003 10:21:50</started>
        <finished>27-May-2003 10:21:51</finished>
    </timingInfo>
</job>
</serverQueues>

```

14.3.4 killJob

killJob(Integer jobId, String serverName, String authId) 操作では、指定されたジョブ ID に基づいてジョブを削除し、操作のステータスを XML フォーマットで返します。

有効なパラメータは、次のとおりです。

- jobId: 情報が必要なジョブの Job ID です。
- serverName: NULL でない有効なサーバー名の値を指定する必要があります。指定されたサーバーがネットワークで稼動していない場合、この操作によってエラーが発生します。
- authId: ユーザー名 / パスワード形式の文字列で、セキュリティ保護されたサーバーに指定する必要があります。セキュリティ保護されていないサーバーの場合、このパラメータは無視されます。

killJob レスポンスを表示する手順は、次のとおりです。

1. 「killJob」リンクをクリックします。「テスト」ページに関連するパラメータ・フィールドとリクエストを送信するための「起動」ボタンが表示されます。
2. jobId (param0)、Reports Server 名 (param1) および authId (param2) を入力します。
3. 「Invoke」をクリックします。SOAP レスポンスが、新しいウィンドウに表示されます。

次は、Job ID=3 を指定した場合の killJob 操作の出力例です。

```

<?xml version = '1.0' encoding = 'ISO-8859-1' standalone = 'yes'?>
<serverQueues>
  <job id="3" queueType="past">
    <name>test.rdf</name>
    <type>report</type>
    <status code="7">Canceled upon user request</status>
    <owner>RWUser</owner>
    <server>repserv</server>
    <destination>
      <desType>cache</desType>
      <desFormat>html</desFormat>
    </destination>
    <timingInfo>
      <queued>27-May-2003 10:21:50</queued>
      <started>27-May-2003 10:21:50</started>
      <finished>27-May-2003 10:22:00</finished>
    </timingInfo>
  </job>
</serverQueues>

```

14.3.5 runJob

runJob(String *commandLine*, Boolean *synchronous*) 操作では、*commandLine* パラメータの一部として指定された Reports Server に対してジョブを実行します。

注意： Oracle Reports Web サービスでは、ジョブ出力または実際のレポートを返しません。

有効なパラメータは、次のとおりです。

- *commandLine*: ジョブを送信するための完全なコマンドライン構文です。次に例を示します。

```
server=repsserv report=test.rdf destype=file desname=output.pdf desformat=pdf
userid=scott/tiger@oraDB
```

注意： コマンドライン・パラメータでは、*paramform=yes* を指定できません。パラメータの実際の値を *commandLine* 引数の一部として渡す必要があります。

- *synchronous*: ジョブを同期で実行するかどうかを指定する BOOLEAN オブジェクトです。

runJob レスポンスを表示する手順は、次のとおりです。

1. 「runJob」リンクをクリックします。「テスト」ページに関連するパラメータ・フィールドとリクエストを送信するための「起動」ボタンが表示されます。
2. コマンドライン構文を入力します (*param0*)。また、ジョブを同期で実行するかどうか (T/F, Y/N) を指定します (*param1*)。
3. 「Invoke」をクリックします。SOAP レスポンスが、新しいウィンドウに表示されます。

次は、runJob 操作の出力例です。

```
<?xml version = '1.0' encoding = 'ISO-8859-1' standalone = 'yes'?>
<serverQueues>
  <job id="7" queueType="current">
    <name>test.rdf</name>
    <type>report</type>
    <status code="1">Waiting in the queue</status>
    <owner>RWUser</owner>
    <server>repsserv</server>
    <destination>
      <desType>file</desType>
      <desName>output.pdf</desName>
      <desFormat>pdf</desFormat>
    </destination>
    <timingInfo>
      <queued>27-May-2003 10:22:00</queued>
      <started>27-May-2003 10:22:00</started>
      <finished>27-May-2003 10:22:00</finished>
    </timingInfo>
  </job>
</serverQueues>
```


14.4 サンプル・プロキシと Java クライアントのインストールと使用方法

RWWebService Web ページには、サンプル・プロキシへのリンクがあります。このサンプル・プロキシは、適切な SOAP メッセージを使用して Web サービスを内部的に起動します。したがって、このプロキシにより、Web サービスで実行する様々な操作へアクセスし、適切なパラメータを使用してこれらの操作を起動します。

次に、プロキシのインストールに必要な手順について説明します。

1. RWWebService の Web ページに表示される Proxy Jar リンクから、rwwebservice.zip ファイルをダウンロードします。
2. rwwebservice.zip ファイルへのパスをシステムのクラスパスに追加します。
3. `ORACLE_HOME¥soap¥lib¥soap.jar` および `ORACLE_HOME¥j2ee¥home¥lib¥http_client.jar` エントリをシステムのクラスパスに追加します。
4. システムのクラスパスに、次のいずれかを追加します。
 - `ORACLE_HOME¥lib¥xmlparserv2.jar` ディレクトリにある `xmlparserv2.jar`
 - Xerces 1.4.4 パーサー、`xerces.jar`

注意： `xmlparser` の位置をシステムのクラスパスに指定する必要があります。この位置を指定しない場合は、`<` 記号のない SOAP レスポンスが表示されます。

5. 正しい Reports Server 名と `authid` を反映するように、サンプルの `RWWebServiceTest.java` を変更します。Java クライアントの作成の詳細は、[例 14-1](#) を参照してください。

注意： セキュリティ保護されていない Reports Server の `authid` は、`null` です。

6. `RWWebServiceTest.java` ファイルをコンパイルして実行します。
7. サンプル Java クライアントを使用して各種操作を実行します。たとえば、API バージョンの取得 (`getAPIVersion`)、ジョブの実行 (`runJob`)、ジョブのステータスのチェック (`getJobInfo`)、Reports Server 情報の取得 (`getServerInfo`) などを実行できます。

注意： サンプル Java クライアントでは、複数のウィンドウから複数のジョブを同時に送信できます。

[例 14-1](#) は、`RWWebServiceTest.java` ファイルの内容を示します。

例 14-1 RWWebServiceTest.java

```

/*
 * $Id: RWWebServiceTest.java
 * @author Anil Sharma
 *
 * Copyright (c) Oracle Corporation 2003. All Rights Reserved
 *
 * FUNCTION
 * This is a sample class to demonstrate how the Oracle Reports WebService
 * Proxy class (oracle.reports.rwclient.proxy.RWWebServiceProxy) can be used
 * to invoke the Reports WebService from Java Clients. Java based Reports
 * Clients will use the demonstrated mechanism for invoking & parsing the
 * results using

```

```

*
* NOTES
* 'oracle.reports.rwclient.proxy.RWWebServiceProxy' class is supplied as part
* of rwebsservice.zip file which can be downloaded by invoking the Reports
* WebService from a browser. Please consult Chapter 14 of
Oracle Application Server Reports Services Publishing Reports to the Web manual,
available on
* the Oracle Technology Network Oracle Reports Documentation page
* (http://www.oracle.com/technology/documentation/reports.html) for details.
*
* CREATED   Anil Sharma   06/13/03
*/

import oracle.reports.rwclient.proxy.RWWebServiceProxy;

/**
 * This class creates an instance of RWWebServiceProxy and makes API
 * calls on it to interact with Reports Server WebService. The result from
 * the webservice is usually an XML object which is printed as-is to the
 * Standard output stream. Java based Reports Clients making use of this
 * WebService class might need to parse the XML to extract meaningful
 * information.
 *
 */

public class RWWebServiceTest
{

    public static void main(String[] args)
    {

        String serverName = "repserv";           //Name of the Reports Server
        String authid     = "portal/welcome1";  //authid, should be null if
                                                //server is not secured.

        String cmdline    = "server=repserv report=test.rdf "+
            "destype=file desname=output.pdf desformat=pdf "+
            "userid=scott/tiger@orcl";

        try
        {
            RWWebServiceProxy proxy = new RWWebServiceProxy();

            /**
             * The following piece of code invokes proxy class' getAPIVersion()
             * to get the Reports Server version information.
             */
            System.out.println("Get Reports Server Version:");
            System.out.println("RESULT:"+proxy.getAPIVersion());

            /**
             * Get the Reports Server Information in XML format. This will contain
             * some server runtime as well as configuration information.
             */
            System.out.println("Get Server Info:");
            System.out.println("RESULT:¥n"+proxy.getServerInfo(serverName, authid));

            /**
             * Get information about a particular job (the job ID needs to be
             * specified.
             */
            System.out.println("Get JobInfo for Job Id=3:");
            System.out.println("RESULT:¥n"+proxy.getJobInfo(new Integer(3), serverName,
authid));
        }
    }
}

```

```
/**
 * Kill a job with a given job ID.
 */
System.out.println("Kill job with Job Id=3:");
System.out.println("RESULT:¥n"+proxy.killJob(new Integer(3), serverName,
authid));

/**
 * Submit a job to the server. The command string takes the same form
 * as the one used for rwclient or any other Oracle Reports client.
 * You can specify whether to run the job synchronously or not. The
 * returned string is in XML format indicating the job status. Please
 * note that with Oracle Reports 10g, you can not get the
 * job output.
 */
System.out.println("Run a job on server");
System.out.println("RESULT:¥n"+proxy.runJob(cmdline, new Boolean(true)));

}
catch (Exception e)
{
    e.printStackTrace();
}
}
```


15

拡張配布の作成

レポートに拡張配布を定義する場合は、配布 XML ファイルの作成を通じて配布を設計できます。このファイルでは、レポートのセクションごとに宛先と出力フォーマットを指定できます。1つの配布 XML ファイルに、各自で設計したカスタムの（プラグガブルな）宛先を含めた複数の異なる宛先を指定できます（第 15.3.9 項「[destype](#)」を参照）。

注意： 配布 XML ファイル (distribution.xml) のサンプルは、Oracle Reports の `ORACLE_HOME\samples\demo` ディレクトリにあります。このファイルを目的に合せて再利用すれば、いちから作成する必要はありません。

この章では、配布 XML ファイルの作成について説明するとともに、いくつかの使用例を示します。この章の主な項は、次のとおりです。

- [配布の概要](#)
- [配布 XML ファイルの概要](#)
- [配布 XML ファイルの要素](#)
- [配布 XML ファイルの例](#)
- [実行時の配布 XML ファイルの使用方法](#)
- [配布の使用に関する制限](#)

15.1 配布の概要

配布 XML ファイルは、レポート出力の配布を指定するために必須ではありませんが、配布が複雑である場合に役立ちます。たとえば、あるレポートの出力を様々な方法で発行する場合があります。レポートのエグゼクティブ・サマリーを上級管理職に、詳細な内訳を個別の部門管理者にメールで送信するとします。この場合、2つのレポート・セクションからなる1つのレポートを作成します。1つは縦長のサマリー・セクション、もう1つは横長のサマリー・セクションです。管理者のリストを含んだデータ・モデル・グループに詳細セクションを関連付けたいので、グループの各インスタンスで宛先を変更し、各部門の出力を関連する管理者に送信します。

配布 XML ファイルを使用すると、ある特定のレポートの複数の出力を1つの XML ファイルに定義し、そのファイルをコマンドラインまたは URL から呼び出すことができるため、配布の複雑さが軽減されます。

15.2 配布 XML ファイルの概要

この項では、配布に関連する XML ファイルの使用方法について説明します。

- [distribution.dtd](#) ファイル
- 属性内での変数の使用方法

15.2.1 distribution.dtd ファイル

配布 XML ファイルを作成する際は、Windows と UNIX の次のディレクトリにある `distribution.dtd` ファイルに定義されている構文に従います。

```
ORACLE_HOME\reports\dtd
```

次の各項では様々な要素や属性が説明されているので、あらかじめ `distribution.dtd` ファイルを印刷し、それを参照しながら読み進めると参考になります。

注意： 配布 XML ファイル内では、データの大文字と小文字が区別されません。各種の要素および属性の大文字と小文字は、`distribution.dtd` ファイルで指定されているとおりに記述する必要があります。

`distribution.dtd` ファイルには、配布 XML ファイル内で有効なすべての要素が示されます。これらの要素にはそれぞれ属性があります。デフォルト値がある属性は、その値を変更する必要がないかぎり、指定しなくてもかまいません。

多数の様々な属性に変数値を使用すると、動的な配布を作成できます。変数値は、配布 XML ファイルを使用するレポート内の列を参照します。

15.2.2 属性内での変数の使用方法

属性内で変数を使用するには、静的値の位置に `&column_name` または `<column_name>` を入力します。

注意： アンパサンド (&) と小なり記号 (<) は、XML で特別の意味を持つ一方で、特定の Oracle Reports コマンドライン・オプションに必要な記号でもあります (たとえば、文字パラメータにはアンパサンド記号が必要です)。変数を設定する際に、XML におけるこれらの記号の意味と競合しないようにするには、エンコード形式のアンパサンド (&) および小なり記号と大なり記号 (< と >) を指定します。次に例を示します。

たとえば、XML ファイル内で間違っってコーディングされた変数は次のようになります。

```
<mail id="a1" to="&<manager>@mycompany.com" ...
```

XML ファイル内で正しくコーディングされた変数は次のようになります。

```
<mail id="a1" to="&lt;manager&gt;@mycompany.com" ...>
```

変数での大なり記号 (>) の使用については特に制約がありませんが、一貫性を持たせるために、エンコード形式 (>) の使用をお勧めします。

使用する変数の構文は、値が単独で表されるか、または他の値や文字列と組み合わせて表されるかによって異なります。たとえば、mail 要素の to 属性の値は、次のいずれかで表されます。

```
<mail id="a2" to="&email" ...>
```

または

```
<mail id="a3" to="&lt;first_name&gt;. &lt;last_name&gt;@myco.com ...>
```

最初の例 (id="a2") では、変数が参照する列 (email) に完全なメール・アドレスが含まれているため、追加情報が必要ありません。2 番目の例 (id="a3") では、変数値 (first_name と last_name) と静的テキストを組み合わせて電子メール・アドレスを表しています (静的テキストは first_name の後のピリオドと @myco.com)。どちらの場合も、電子メールのアドレスが動的に指定されます。どちらの例を使用するかは、変数に必要なすべての情報が含まれているか、または補完のための追加情報が必要とするかによって異なります。

さらに複雑なレイアウトの場合は、PL/SQL 式で作成したレポート列を参照することもできます。たとえば、レポートの中で次の PL/SQL 列を定義するとします。

```
PL/SQL formula CF_MAILID: return(:first_name||'.'||:last_name)
```

この列を、配布 XML ファイルで次のように参照します。

```
to="&lt;CF_MAILID&gt;@mycompany.com"
```

15.3 配布 XML ファイルの要素

配布 XML ファイルの要素は、次のとおりです。

- [destinations](#)
- [foreach](#)
- [mail](#)
- [body](#)
- [attach](#)
- [include](#)

- [file](#)
- [printer](#)
- [destype](#)
- [property](#)

このうちほとんどの要素には、その要素の動作を定義する属性があります。次の各項では、配布 XML ファイルの要素と関連する属性について説明します。第 15.4 項「[配布 XML ファイルの例](#)」では、一般的なシナリオにおける配布 XML ファイルの要素と属性の使用例を示します。

15.3.1 destinations

例

```
<destinations>
  one or more distribution specifications
</destinations>
```

必須 / オプション

必須です。配布 XML ファイルには、1 つの `destinations` 要素を指定する必要があります。

説明

`destinations` 要素は、配布 XML ファイルのコンテンツ領域を開始および終了します。配布 XML ファイルのタグ階層で見ると、`destinations` 要素にそれ以外のすべての要素が従属しています。

15.3.2 foreach

例

```
<foreach>
  <mail id="a1" to="my_addressee@mycompany.com" subject="Fourth Quarter Results">
    <attach format="pdf" name="dept_&lt;department_ID&gt;.pdf"
      srcType="report" instance="this">
      <include src="mainSection"/>
    </attach>
  </mail>
</foreach>
```

または

```
<mail id="a4" to="recipient@mycompany.com" subject="Regional Results">
  <foreach>
    <attach format="pdf" name="report.pdf" srcType="report" instance="all">
      <include src="mainSection"/>
    </attach>
  </foreach>
</mail>
```

必須 / オプション

オプションです。`foreach` 要素は、必要な数だけ指定できます。

説明

`foreach` 要素は、繰返しグループに対して配布を一括処理するために使用します。`foreach` を使用できるのは、関連するレポート定義ファイル (RDF、JSP、XML のいずれか) において、一括処理されるセクションに対する繰返しプロパティが、該当するグループに設定されている場合にかぎられます。`foreach` 要素は、開始と終了タグの間に定義されている配布を、各繰返しグループに対して実行するように指定します。

「繰返し」プロパティは、レポート・セクション（ヘッダー、メイン、トレーラ）でデータ・モデルのブレイク・グループをセクションに関連付けるために設定できます。セクションに「繰返し」プロパティを設定すると、セクションの複数インスタンス（繰返しセクション）を生成できます。

レポートで一括処理および配布を実装する場合は、セクションの「繰返し」プロパティをデータ・モデルのブレイク・グループに設定すると、セクション・レベルの配布を生成することができます。これにより、そのブレイク・グループの列レコードごとにセクションのインスタンスが生成されます。その後、セクションの各インスタンスを適宜配布できます（たとえば、MANAGER グループ内の個々のマネージャへの配布）。

「繰返し」プロパティを複数のレポート・セクション（ヘッダー、メインおよびトレーラ）に対して設定する場合は、「繰返し」プロパティの値は同じデータ・モデルのブレイク・グループに設定する必要があります。ヘッダー、メインおよびトレーラ・セクションのいずれかで「繰返し」プロパティを異なるデータ・モデルのブレイク・グループに設定した場合、Oracle Reports では次のメッセージが表示されます。

```
REP-177: Error while running in remote server
REP-34320: Report sections used in destination '<destination id>' do not repeat on the
same group
```

foreach 要素を mail 要素のサブ要素として使用することもできます。これは、この項の始めにある 2 番目の例に示されているとおりです（この例では、mainSection が G_DEPARTMENT_ID で繰り返されることを前提として、レポートの mainSection のすべてのインスタンスを含んだ 1 つの添付ファイルが 1 つのファイルに作成されます）。

foreach 要素の動作には、attach 要素および file 要素の instance 属性が密接に関係していません。foreach でレコード・グループに従って配布を実行するように指定し、instance で、一括処理されるグループを 1 つのファイルで配布するか (instance="all") または別々のファイルとして配布、つまりグループ・インスタンスごとに 1 つのファイルを配布するか (instance="this") を指定します。

mail 要素とともに使用した場合、foreach の意味は、mail 要素に対するその位置によって変わります。

- foreach が mail 要素および instance="this" の前にある場合は、各グループ・インスタンスが別々のメールとして送信されます。次に例を示します。

```
<foreach>
  <mail id="a1" to="managers@mycompany.com" subject="results">
    <attach name="department_&lt;&lt;department_id&gt;&gt;.pdf" instance="this">
      <include src="mainSection" />
    </attach>
  </mail>
</foreach>
```

レポートが department_id に従ってグループ化され、かつ 4 つの部門が存在する場合は、グループ・インスタンスの数が 4 つになります。つまり、受信者ごとに 4 通の電子メールが作成され、各電子メールにその独自のグループ・インスタンスが添付されます。たとえば、ある電子メールに部門 10 のレポート、別の電子メールに部門 20 のレポートが添付され、残りの電子メールにもそれぞれ違う部門のレポートが添付されます。各受信者が、4 通の電子メールすべてを受け取ります。

- foreach が mail 要素および instance="this" の後にある場合は、各グループ・インスタンスが 1 通の電子メールに添付され、それが各受信者に送信されます。次に例を示します。

```
<mail id="a1" to="managers@mycompany.com" subject="results">
  <foreach>
    <attach name="department_&lt;&lt;department_id&gt;&gt;.pdf" instance="this">
      <include src="mainSection" />
    </attach>
  </foreach>
</mail>
```

15.3.3 mail

例

```
<mail id="a1" to="jsmith@foo.com" subject="Results">
  <body srcType="text">
    Attached are quarterly results.
  </body>
  <attach srcType="report">
    <include src="headerSection"/>
    <include src="mainSection"/>
  </attach>
</mail>
または
<mail id="a4" to="recipient@mycompany.com" subject="Regional Results">
  <foreach>
    <attach format="pdf" name="report.pdf" srcType="report" instance="this">
      <include src="mainSection"/>
    </attach>
  </foreach>
</mail>
```

必須 / オプション

オプションです。mail 要素は、必要な数だけ指定できます。

説明

mail 要素は、SMTP ベースの発信メール・サーバーを介した配布を指定するために使用します。この要素を使用して、電子メールの受信者、件名および優先度を指定します。

mail 要素の開始と終了タグの間には、body サブ要素を 1 つだけ指定できます。また、任意数の attach サブ要素と foreach サブ要素を任意の場所に指定できます。

mail 要素には、関連する属性もあります。これらの属性は、mail タグ内に記述されます。たとえば、id、to、subject の 3 つの属性が記述されます。

```
<mail id="a1" to="jsmith@foo.com" subject="Recent Hires">
```

表 15-1 では、mail 要素に関連する属性を説明します。

表 15-1 mail 要素の属性

属性	有効値	説明
id	文字列	必須です。特定の mail 要素を識別する、配布 XML ファイル内で一意のキーワードです。テキスト文字列と 1 つ以上の数値を組み合わせた値を指定できます (id="a1" など)。id 値は、英字で始まる必要があります。
to	文字列	必須です。変数値を指定できます。電子メールの受信者を指定します。完全に正式な電子メール・アドレスを含みます (to="jsmith@foo.com" など)。複数の受信者は、カンマで区切ります。関連レポートで使用されている列を参照する変数値を指定することもできます。詳細は、 第 15.2.2 項 を参照してください。
cc	文字列	オプションです。変数値を指定できます。電子メールのコピーを受け取る受信者を指定します。

表 15-1 mail 要素の属性 (続き)

属性	有効値	説明
bcc	文字列	オプションです。変数値を指定できます。電子メールのブラインド・コピーを受け取る受信者を指定します。
from	文字列	オプションです。変数値を指定できます。電子メールの送信者を指定します。
replyTo	文字列	オプションです。変数値を指定できます。返信の送信先となる電子メール・アカウントを指定します。
subject	文字列	デフォルト: Mail Sent from & Report オプションです。変数値を指定できます。電子メールの件名を指定します。subject が指定されなかった場合は、件名の行に Mail Sent from [Name of Report] と表示されます。
priority	highest high normal low lowest	デフォルト: normal 電子メールの配信優先度。
returnReceipt	true false	デフォルト: false 電子メールが受信されたときに返信先の個人またはアカウントに通知するかどうかを指定します。
organization	文字列	オプションです。変数値を指定できます。電子メールを配布する組織の名前を指定します。次に例を示します。 organization="Region 10 Sales" または organization="&department_name"

注意: mail 要素が正しく機能するには、Reports Server で、メールの送信先である発信 SMTP メール・サーバーを認識する必要があります。この情報は、Reports Server 構成ファイル (server_name.conf) で指定します。このファイルの pluginParam 要素に、メール・サーバーの名前を入力できます。次に例を示します。

```
<pluginParam name=mailServer>smtp01.mycorp.com</pluginParam>
```

詳細は、第 3 章「OracleAS Reports Services の構成」を参照してください。

15.3.4 body

例

Windows の場合

```
<mail id="a1" to="jsmith@foo.com" subject="Results">
  <body srcType="file">
    <include src="c:\mail\body.html"/>
  </body>
</mail>
```

UNIX の場合

```
<mail id="a1" to="jsmith@foo.com" subject="Results">
  <body srcType="file">
    <include src="/mail/body.html"/>
  </body>
</mail>
```

必須 / オプション

オプションです。1 つの mail 要素には、body 要素を 1 つだけ関連付けることができます。

説明

body 要素は、mail 要素のサブ要素です。この要素は、電子メールの内容 (body) を指定します。body では、body 要素の開始タグと終了タグの間にテキスト文字列を入力するか、または **include** サブ要素を使用して外部ファイル、レポートまたはレポートのセクションを指定できます。次に例を示します。

```
<mail id="a1" to="jsmith@foo.com" subject="Results">
  <body srcType="text">
    Attached are quarterly results.
  </body>
...

```

または

```
<mail id="a1" to="jsmith@foo.com" subject="Results">
  <body srcType="file">
    <include src="d:/reports/admin/results.html"/>
  </body>
...

```

または

```
<mail id="a1" to="&amp;&lt;first_name&gt;. &amp;&lt;last_name&gt;@myco.com"
  subject="Quarterly Results">
  <body srcType="report" format="html">
    <include src="headerSection"/>
  </body>
...

```

body には、srcType、format、instance の 3 つの属性があります。これらの属性について表 15-2 で説明します。

表 15-2 mail の body サブ要素の属性

属性	有効値	説明
srcType	file report text	デフォルト: report 電子メールの内容のソース。内容は電子メールの本文に表示されます。srcType を指定しなかった場合は、デフォルト値が使用されます。
format	html htmlcss ascii	デフォルト: html srcType が report であり、format がデフォルトの html 以外の場合は必須です。それ以外はオプションです。内容のフォーマットを指定します。
instance	this all	デフォルト: all foreach 要素が指定されている場合に使用します。グループ化されたレポートが別々のレポートに一括処理される場合は、instance で、グループをグループ・インスタンスごとに個別の内容へと分割するか (this)、またはすべてを同じ内容に収めるか (all) を指定します。

15.3.5 attach

例

```
<mail id="a1" to="jsmith@foo.com" subject="Results">
  <body srcType="text">
    Attached are quarterly results.
  </body>
  <foreach>
    <attach format="html" name="contacts.htm" srcType="report" instance="all">
      <include src="headerSection"/>
      <include src="mainSection"/>
    </attach>
  </foreach>
</mail>
```

必須 / オプション

オプションです。attach 要素は、1つの mail 要素に対して必要な数だけ指定できます。attach は、foreach のサブ要素でもあることに注意してください。foreach では、そのサブ要素 (mail、file、printer、destype および attach のサブ要素) の少なくとも 1つを使用する必要があります。

説明

attach 要素は、電子メールの添付ファイルを指定します。attach には少なくとも 1つの include サブ要素が必要です。srcType="report" である場合は、このサブ要素を複数指定できます。

表 15-3 では、attach 要素に関連する属性を説明します。

表 15-3 mail の attach サブ要素の属性

属性	有効値	説明
format	pdf html htmlcss rtf ascii xml dflt	デフォルト: pdf srcType が report であり、レポート・フォーマットがデフォルトの pdf 以外である場合は必須です。それ以外はオプションです。添付ファイルのフォーマットを指定します (format="htmlcss" など)。
name	文字列	オプションです。変数値を指定できます。添付ファイルのファイル名を指定します。関連レポートで使用されている列を参照する変数値を指定することもできます。詳細は、第 15.2.2 項を参照してください。
srcType	file report text	デフォルト: report 添付ファイルのソース (ファイル、レポート、テキストのいずれか)。
instance	this all	デフォルト: all foreach 要素が指定されている場合に使用します。グループ化されたレポートが別々のレポートに一括処理される場合は、instance で、グループをグループ・インスタンスごとに個別の内容へと分割するか (this)、またはすべてを同じ内容に収めるか (all) を指定します。

これらの属性を **foreach** 要素とともに使用することで、グループ・インスタンスで繰り返される宛先を設計し、グループの添付ファイルごとに電子メールを生成できます。次に例を示します。

```
<foreach>
  <mail id="a2" to="first.name@myco.com,second.name@myco.com, third.name@myco.com,
    fourth.name@myco.com" subject="Department Summaries">
    <body srcType="text">
      Attached is the breakdown of department summaries for the last quarter.
    </body>
    <attach format="htmlcss" name="deptsum.html" srcType="report" instance="this">
      <include src="report"/>
    </attach>
  </mail>
</foreach>
```

foreach 要素の位置を変えると、複数のファイルが添付された 1 通の電子メールを生成できます。この場合、グループ・インスタンスごとに添付ファイルが異なります。

```
<mail id="a2" to="first.name@myco.com,second.name@myco.com, third.name@myco.com,
  fourth.name@myco.com" subject="Department Summaries">
  <body srcType="text">
    Attached is the breakdown of department summaries for the last quarter.
  </body>
  <foreach>
    <attach format="htmlcss" name="deptsum.html" srcType="report" instance="this">
      <include src="report"/>
    </attach>
  </foreach>
</mail>
```

15.3.6 include

例

```
<mail id="a1" to="jsmith@foo.com" subject="Q4">
  <body srcType="text">
    Attached are quarterly results.
  </body>
  <attach srcType="report" format="pdf">
    <include src="report"/>
  </attach>
</mail>
```

または

```
<mail id="a1" to="jsmith@foo.com" subject="Q4">
  <body srcType="text">
    Attached are quarterly results.
  </body>
  <attach srcType="report" format="htmlcss">
    <include src="headerSection"/>
  </attach>
</mail>
```

または

```
<mail id="a1" to="jsmith@foo.com" subject="Q4">
  <body srcType="text">
    Attached are quarterly results.
  </body>
  <attach srcType="file">
    <include src="d:/management/reports/current/Q4.htm"/>
  </attach>
</mail>
```

必須 / オプション

body および attach とともに使用されるときに、srcType が report または file である場合は必須です。ただし、srcType が text である場合はオプションです。file、printer および destype とともに使用する場合は必須です。include が必要なインスタンスでは、必ず 1 つの include を指定する必要があります。複数指定してもかまいません。

説明

include 要素は、body、attach、file、printer、destype の各要素とともに使用します。この要素は、電子メールの本文に含めるファイル、レポートまたはレポート・セクションを、電子メールの添付ファイル、ファイルの内容、プリンタ出力またはカスタム宛先タイプの内容として指定します。

レポート全体ではなく複数のセクションを指定する場合は、必要なセクションごとに include を入力します。次に例を示します。

```
<mail id="a1" to="jsmith@foo.com" subject="Results">
  <body srcType="text">
    Attached are quarterly results.
  </body>
  <attach srcType="report" format="htmlcss">
    <include src="headerSection"/>
    <include src="mainSection"/>
  </attach>
</mail>
```

先に指定された body 要素または attach 要素の srcType 属性が file である場合は、これに続く include 要素で、そのファイルをディレクトリ・パスとファイル名で指定するか、または単にファイル名で指定できます（そのファイルが REPORTS_PATH 環境変数で示されたディレクトリ内にある場合）。次に例を示します。

```
<mail id="a1" to="jsmith@foo.com">
  <body srcType="file">
    <include src="q4sales.pdf"/>
  </body>
</mail>
```

パスを指定する場合は、プラットフォームに適した標準を使用してください。次に例を示します。

Windows の場合： <include src="c:¥management¥reports¥current¥Q4.htm"/>

UNIX の場合： <include src="/management/reports/current/Q4.htm"/>

include 要素の開始タグと終了タグの間に他の XML 要素は記述されません。

表 15-4 に、include 要素の src 属性について説明します。

表 15-4 body または attach とともに使用される include サブ要素の属性

属性	有効値	説明
src	(パスおよび) ファイル名 report headerSection mainSection trailerSection	<p>必須です。先行する attach、body、file、printer または destype の各要素に指定されているデータのソースを指定します。</p> <p>配布 XML ファイルは、特定のレポートを実行するとき呼び出されるため、src="report" である場合は、src 属性にレポートの名前や位置を指定する必要があります。</p> <p>(パスおよび) ファイル名: 先行する要素は body または attach で srcType=file である必要があります。ディレクトリ・パスとファイル名を指定するか、そのファイルが REPORTS_PATH 環境変数で示されたディレクトリにある場合はファイル名のみを指定します。</p> <p>その他の値: 先行する body または attach 要素で srcType=report を指定する場合、レポート全体を指定するか (report)、あるいは本文に含める (または添付する) レポートのセクションを指定します (たとえば、headerSection、mainSection、または trailerSection)。</p>

15.3.7 file

例

Windows の場合

```
<file id="a7" name="c:\management\reports\report.pdf" format="pdf">
  <include src="report"/>
</file>
```

UNIX の場合

```
<file id="a7" name="/management/reports/report.pdf" format="pdf">
  <include src="report"/>
</file>
```

または

```
<foreach>
  <file id="a7" name="section&lt;department_id&gt;.pdf" format="pdf"
    instance="this">
    <include src="mainSection"/>
  </file>
</foreach>
```

必須 / オプション

オプションです。file 要素は、必要な数だけ指定できます。

説明

file 要素は、ファイルへの配布を指定するために使用します。file 要素には、include というサブ要素が 1 つあります。file 要素の開始タグと終了タグの間には、少なくとも 1 つの include サブ要素が必要です。このサブ要素は複数指定できます。

file 要素を、foreach 要素および instance="this" 属性とともに使用すると、グループ化されたレポートの各グループ・インスタンスを別々のファイルに配布できます。たとえば、レポートを department_id でグループ化し、4 つの部門が存在する場合は、foreach/file/instance="this" を組み合わせて、それぞれが別々の部門のレポートを含んだ 4 つのファイルを生成できます。この場合、配布 XML ファイルにおける file エントリは、次のようになります。

```
<foreach>
  <file id="a3" name="dept_&amp;&lt;department_id&gt;.pdf" format="pdf"
    instance="this">
    <include src="report"/>
  </file>
</foreach>
```

この例では、すべてのレポート・セクション（ヘッダー、メインおよびトレーラ）が同じグループ・インスタンス（department_id など）で繰り返されます。

表 15-5 では、file 要素に関連する属性を説明します。

表 15-5 file 要素の属性

属性	有効値	説明
id	文字列	必須です。特定の file 要素を識別する、配布 XML ファイル内で一意のキーワードです。テキスト文字列と 1 つ以上の数値を組み合わせた値を指定できます (id="a1" など)。id 値は、英字で始まる必要があります。
name	文字列	必須です。変数値を指定できます。宛先ファイルの位置とファイル名を指定します。ディレクトリ・パスを入力します。ファイル名も含めます。次に例を示します。 Windows: name="d:¥reports¥q4sales.pdf" UNIX: name="reports/q4sales.pdf" 関連レポートで使用されている列を参照する変数値を指定することもできます。詳細は、 第 15.2.2 項 を参照してください。
format	pdf html htmlcss rtf ascii xml bitmap	デフォルト: pdf 宛先ファイルのフォーマット。次に例を示します。 format="htmlcss"
instance	this all	デフォルト: all foreach 要素が指定されている場合に使用します。グループ化されたレポートが別々のレポートに一括処理される場合は、instance で、グループをグループ・インスタンスごとに個別のファイルへと分割するか (this)、またはすべてを同じファイルに収めるか (all) を指定します。

15.3.8 printer

例

Windows の場合

```
<printer id="a1" name="¥¥server_name¥printer_name" copies="5">
  <include src="report"/>
</printer>
```

UNIX の場合

```
<printer id="a1" name="alias_to_registered_printer" copies="5" instance="all">
  <include src="report"/>
</printer>
```

必須 / オプション

オプションです。printer 要素は、必要な数だけ指定できます。

説明

printer 要素は、プリンタへの配布を指定するために使用します。printer 要素には、**include** というサブ要素が 1 つあります。printer 要素の開始タグと終了タグの間には、少なくとも 1 つの include サブ要素が必要です。このサブ要素は複数指定できます。

printer 要素を、**foreach** 要素および `instance="this"` 属性とともに使用すると、グループ化されたレポートの各グループ・インスタンスを別々の印刷ジョブに配布できます。たとえば、レポートを `department_id` でグループ化し、4 つの部門が存在する場合は、`foreach/printer/instance="this"` を組み合わせて、それぞれが別々の部門のレポートを含んだ 4 つの印刷レポートを生成できます。この場合、配布 XML ファイルにおける printer エントリは、次のようになります。

```
<foreach>
  <printer id="a7" name="¥¥server_name¥printer_name" instance="this">
    <include src="report"/>
  </printer>
</foreach>
```

この例では、すべてのレポート・セクション（ヘッダー、メインおよびトレーラ）が同じグループ・インスタンス（`department_id` など）で繰り返されます。

表 15-6 では、printer 要素に関連する属性を説明します。

表 15-6 printer 要素の属性

属性	有効値	説明
id	文字列	必須です。特定の file 要素を識別する、配布 XML ファイル内で一意のキーワードです。テキスト文字列と 1 つ以上の数値を組み合わせた値を指定できます（ <code>id="a1"</code> など）。id 値は、英字で始まる必要があります。

表 15-6 printer 要素の属性 (続き)

属性	有効値	説明
name	文字列	<p>必須です。変数値を指定できます。宛先のプリンタを指定します。この情報の入力方法は、Windows と UNIX で異なります。</p> <p>Windows の場合は、プリンタ・サーバー名とプリンタ名を指定します。次に例を示します。</p> <pre>name="%%server_name%%printer_name"</pre> <p>UNIX の場合は、登録されたプリンタに割り当てられた別名を指定します。次に例を示します。</p> <pre>name="sales_printer"</pre> <p>関連レポートで使用されている列を参照する変数値を指定することもできます。詳細は、第 15.2.2 項を参照してください。</p>
copies	文字列	<p>デフォルト: 1</p> <p>各レポートまたは各レポート・グループ・インスタンスの印刷部数。</p>
instance	this all	<p>デフォルト: all</p> <p>foreach 要素が指定されている場合に使用します。グループ化されたレポートが別々のレポートに一括処理される場合は、instance で、グループをグループ・インスタンスごとに別々の印刷レポートに分割するか (this)、またはすべてを同じ印刷レポートに収めるか (all) を指定します。</p>

15.3.9 destype

例

```
<destype id="acustom1" name="fax">
  <include src="headerSection"/>
  <property name="number" value="914925551212"/>
</destype>
```

distribution.xml ファイルに destype 要素を使用して、配布の宛先を OracleAS Portal、FTP、WebDAV、および FAX に指定する例は、[第 15.4 項「配布 XML ファイルの例」](#)を参照してください。

必須 / オプション

オプションです。destype 要素は、必要な数だけ指定できます。

説明

destype 要素は、FAX マシンや FTP サイトなどのカスタムの宛先への配布を指定するために使用します。また、OracleAS Portal で作成されたポータルへの配布を指定する場合も、destype 要素を使用します。destype 要素では、**property** および **include** の 2 つのサブ要素を使用できます。このうち、include は少なくとも 1 つ指定する必要があります。

注意： カスタムの宛先タイプを指定するには、レポートの内容をカスタムの出力先に導くための配布ハンドラを定義する必要があります。カスタムの宛先タイプは、OracleAS Reports Services Destinations API で作成します。

Oracle Reports に使用可能な API の詳細は、Reports Software Development Kit (RSDK) を参照してください。RSDK を入手するには、OTN (Oracle Technology Network) の Oracle Reports 10g ページ (<http://www.oracle.com/technology/products/reports/index.html>) で「SDK」をクリックします。

destype 要素を、foreach 要素および instance="this" 属性とともに使用すると、グループ化されたレポートの各グループ・インスタンスを別々の destype インスタンス（別々の FAX など）に配布できます。たとえば、レポートを department_id でグループ化し、4つの部門が存在する場合は、それぞれが別々の部門のレポートが含まれる4つの destype インスタンスを生成できます。この場合、配布 XML ファイルにおける destype エントリは、次のようになります。

```
<foreach>
  <destype id="a9" name="fax" instance="this">
    <include src="report"/>
    <property name="number" value="&lt;&lt;fax_number&gt;"/>
  </destype>
</foreach>
```

この例では、すべてのレポート・セクション（ヘッダー、メインおよびトレーラ）が同じグループ・インスタンス（fax_number など）で繰り返されます。

カスタムの宛先タイプには、関連する属性もあります。これらの属性は、destype タグ内に記述されます。たとえば、id、name、instance の3つの属性が記述されます。

```
<foreach>
  <destype id="a1" name="name_of_destination_type" instance="all">
    <include src='report'/>
  </destype>
</foreach>
```

表 15-7 では、destype 要素に関連する属性を説明します。

表 15-7 destype 要素の属性

属性	有効値	説明
id	文字列	必須です。特定の file 要素を識別する、配布 XML ファイル内で一意のキーワードです。テキスト文字列と1つ以上の数値を組み合わせた値を指定できます (id="a1" など)。id 値は、英字で始まる必要があります。
name	文字列	必須です。カスタムの宛先の名前を指定します。たとえば、FAX の場合は次のように指定します。 name="fax" OracleAS Portal で作成されたポータルの場合は、次のように指定します。 name="oraclePortal"

表 15-7 destype 要素の属性 (続き)

属性	有効値	説明
instance	this all	<p>デフォルト: all</p> <p>foreach 要素が指定されている場合に使用します。グループ化されたレポートが別々のレポートに一括処理される場合は、instance で、グループをグループ・インスタンスごとに別々の destype インスタンスに分割するか (this)、すべてを同じ destype インスタンスに収めるか (all) を指定します。</p> <p>たとえば、カスタムの宛先タイプが FAX である場合、instance="this" はグループ・インスタンスごとに個別の FAX を使用することを意味し、instance="all" はすべてのグループに同じ FAX を使用することを意味します。</p>

15.3.10 property

例

```
<foreach>
  <destype id="custom1" name="fax" instance="all">
    <include src="headerSection"/>
    <property name="number" value="914925551212"/>
  </destype>
</foreach>
```

必須 / オプション

オプションです。destype 要素の下には、必要な数のプロパティを指定できます。

説明

property 要素には、カスタムの宛先タイプ (destype) で認識できる用語により表された名前 / 値のペアを指定します。プロパティは、単に宛先ハンドラに渡されます。OracleAS Reports Services 内では特に機能を果たしません。プロパティの指定方法は、使用するカスタムの宛先の要件に完全に依存します。

15.4 配布 XML ファイルの例

この項では、配布 XML 要素の単純な例から複雑な例までを紹介します。これらの例は、次に示す distribution.dtd の主要な要素に基づいて編成されています。

- [foreach](#) の例
- [mail](#) の例
- [file](#) の例
- [printer](#) の例
- [destype](#) の例

15.4.1 foreach の例

この項で取り上げる例は、次のとおりです。

- レポート・グループを個別の添付ファイルとする単一の電子メール
- グループ・インスタンスごとに個別の電子メール
- 単独のセクションを添付ファイルとする個別の電子メール
- セクションごとに異なるファイル
- レポートごとに実行される印刷

15.4.1.1 レポート・グループを個別の添付ファイルとする単一の電子メール

この例では、各添付ファイルがヘッダー、メイン、トレーラの各セクションの対応するインスタンスを含んでいます。つまり、レポートが department_id でグループ化され、最初の部門が部門 10 である場合は、部門 10 の情報を含んでいるヘッダー、メイン、トレーラの各セクションからなるレポートが最初の添付ファイルとなります。この例が有効となるのは、同じグループ・インスタンス（この例の場合は department_id）でヘッダー、メイン、トレーラの各セクションが繰り返される場合にかぎられます。

```
<mail id="a1" to="managers@mycompany.com" subject="New Hires">
  <foreach>
    <attach format="html" srcType="report" instance="this">
      <include src="report"/>
    </attach>
  </foreach>
</mail>
```

まず、この例では、managers@mycompany.com が各部門の管理者に配信を行うメーリング・リストのアドレスであることを前提とします。10、20、30、40 の 4 つの部門がある場合は、最初の添付ファイルに部門 10 に対応するヘッダー、メイン、トレーラの各セクションが取り込まれ、後続の各添付ファイルに残りの各部門に対応するセクションが取り込まれます。この例では、4 つの添付ファイルを伴う電子メールが受信者ごとに 1 通ずつ作成されます。

15.4.1.2 グループ・インスタンスごとに個別の電子メール

この例では、各受信者が、グループ化されたレポートごとに異なる電子メールを受け取ります。たとえば、レポートが department_id でグループ化され、4 つの部門が存在する場合は、1 人の受信者がそれぞれに各部門のレポートが添付された 4 通の電子メールを受け取ります。

```
<foreach>
  <mail id="weeklies" to="managers@mycompany.com">
    <attach format="htmlcss" srcType="report" instance="this">
      <include src="mainSection"/>
    </attach>
  </mail>
</foreach>
```

15.4.1.3 単独のセクションを添付ファイルとする個別の電子メール

この例では、個別のセクションが別々のグループで繰り返されます。各受信者が、グループ化されたメイン・セクションごと、およびグループ化されたトレーラ・セクションごとの添付ファイルを伴う個別の電子メールを受信するように配布が設定されます。

```
<foreach>
  <mail id="a6" to="managers@mycompany.com" subject="Personnel Reports">
    <attach format="pdf" name="attach.pdf" srcType="report" instance="this">
      <include src="mainSection"/>
    </attach>
    <attach format="rtf" name="attach.rtf" srcType="report" instance="this">
      <include src="trailerSection"/>
    </attach>
  </mail>
</foreach>
```

15.4.1.4 セクションごとに異なるファイル

この例では、グループ・インスタンスごとに異なるファイルが生成されます。グループは department_id で繰り返されます。各ファイルには、関連する部門 ID から名前が付けられます。

```
<foreach>
  <file id="a10" name="department_&lt;&gt;department_id&gt;.pdf" instance="this">
    <include src="mainSection"/>
  </file>
</foreach>
```

この例では、10 から 40 までの 4 つの部門があることを前提に、department_10.pdf、department_20.pdf などの名前で 4 つのファイルが作成されます。

file 要素には format 属性が含まれていません。これは、srcType が file または text である場合は、この属性が不要であるためです。srcType が report である場合は、この属性が必要となります。

注意： この例で、変数値を使用するために一意のファイル名を指定しない場合（第 15.2.2 項を参照）は、後から作成される各ファイルによって、前に作成されたファイルが上書きされます。つまり、部門 20 の department.pdf ファイルによって部門 10 の department.pdf ファイルが上書きされます。この上書きは、最後の部門（部門 40）レポートの情報を含んだ department.pdf が唯一のファイルとなるまで続きます。

15.4.1.5 レポートごとに実行される印刷

プリンタ名の指定方法は、Windows と UNIX で異なります。最初に Windows の例を示します。次に UNIX の例を示します。

15.4.1.5.1 Windows この例では、レポートが department_id でグループ化されるという前提で、レポートが部門ごとに印刷されます。

```
<foreach>
  <printer id="a7" name="¥¥server_name¥¥printer_name" instance="this">
    <include src="report"/>
  </printer>
</foreach>
```

15.4.1.5.2 UNIX この例では、レポートが department_id でグループ化されるという前提で、レポートが部門ごとに印刷されます。

```
<foreach>
  <printer id="a7" name="printer_alias" instance="this">
    <include src="report"/>
  </printer>
</foreach>
```

15.4.2 mail の例

この項で取り上げる例は、次のとおりです。

- レポート全体を本文とする電子メール
- レポートの 1 つのセクションを本文とする電子メール
- レポートの 2 つのセクションを本文とする電子メール
- 外部ファイルを本文とし、レポートを添付ファイルとする電子メール
- レポート全体およびグループ化されたセクションが添付された電子メール
- 関連する管理者と部門に送信される電子メール

15.4.2.1 レポート全体を本文とする電子メール

レポートがこの電子メールの内容となります。つまり、受信者がこの電子メールを開くと、レポートが表示されます。

```
<mail id="a5" to="managers@mycompany.com" subject="Quarterly Report">
<body srcType="report" format="html">
<include src="report"/>
</body>
</mail>
```

15.4.2.2 レポートの1つのセクションを本文とする電子メール

レポートの1つのセクションがこの電子メールの内容となります。つまり、受信者がこの電子メールを開くと、レポートのセクションが表示されます。

```
<mail id="a6" to="employees@mycompany.com">
<body srcType="report" format="html">
<include src="mainSection"/>
</body>
</mail>
```

この mail 要素には subject 属性が含まれないため、デフォルトの件名である Mail Sent From &Report が使用されます。変数 &Report は、実行時にレポートの名前で置き換えられます。

15.4.2.3 レポートの2つのセクションを本文とする電子メール

レポートの2つのセクションがこの電子メールの内容となります。つまり、受信者がこの電子メールを開くと、1つのレポートに結合された2つのセクション (headerSection と mainSection) が表示されます。

```
<mail id="emp_addresses" to="employees@mycompany.com" subject="Employee Address List">
<body srcType="report" format="html">
<include src="headerSection"/>
<include src="mainSection"/>
</body>
</mail>
```

15.4.2.4 外部ファイルを本文とし、レポートを添付ファイルとする電子メール

このメールでは、外部ファイルが本文の内容となり、レポートが添付ファイルとなります。外部ファイルへのパスの表し方は、Windows と UNIX で異なります。

15.4.2.4.1 Windows

```
<mail id="XQRSN" to="accounting@mycompany.com" subject="Salaries"
<body srcType="file">
<include src="c:\mail\body.html"/>
</body>
<attach format="pdf" name="salaries.pdf" srcType="report">
<include src="report"/>
</attach>
</mail>
```

15.4.2.4.2 UNIX

```
<mail id="XQRSN" to="accounting@mycompany.com" subject="Salaries"
<body srcType="file">
<include src="/mail/body.html"/>
</body>
<attach format="pdf" name="salaries.pdf" srcType="report">
<include src="report"/>
</attach>
</mail>
```


15.4.2.5 レポート全体およびグループ化されたセクションが添付された電子メール

この例では、受信者が複数の添付ファイルを伴う 1 通の電子メールを受け取ります。その内訳は、グループ・インスタンスごとの 1 つの添付ファイルと、レポート全体を含んだ 1 つの添付ファイルです。レポートが department_id でグループ化され、4 つの部門が存在する場合は、受信者が 5 つの添付ファイルを受け取ります（部門ごとに 1 つの添付ファイルとレポート全体の添付ファイル）。

```
<mail id="grx90" to="sales@mycompany.com">
<body srcType="text">
Attached you will find the summary report and breakdown by department of weekly totals.
</body>
<attach format="rtf" name="myAttach.rtf" srcType="report">
<include src="report"/>
</attach>
<foreach>
<attach format="pdf" name="myattach.pdf" srcType="report" instance="this">
<include src="mainSection"/>
</attach>
</foreach>
</mail>
```

15.4.2.6 関連する管理者と部門に送信される電子メール

この例では、部門 10 の管理者が部門 10 のレポートを受け取り、部門 20 の管理者が部門 20 のレポートを受け取ります。このタグを有効にするには、変数の参照先が、添付されるセクションとともに使用される "繰り返し" グループに含まれている列である必要があります。つまり、セクションが G_department_id で繰り返される場合は、manager がそのグループの列である必要があります。

```
<foreach>
<mail id="mgr1090" to="&lt;manager&gt;@mycompany.com">
<attach format="pdf" name="attach.pdf" srcType="report" instance="this">
<include src="mainSection"/>
</attach>
</mail>
</foreach>
```

15.4.3 file の例

グループ化されたレポートを一括処理してファイルに配布する場合は、繰り返しグループに基づいた変数値またはその他の変数情報を使用してファイル名を指定する必要があります。そうしないと、ファイルが作成されるたびに、前に作成されたファイルが上書きされるおそれがあります。たとえば、department.pdf の出力ファイル名を指定し、各部門のレポートのインスタンスを個別に出力すると、2 番目の department.pdf ファイルによって最初の department.pdf ファイルが上書きされ、3 番目の department.pdf ファイルによって 2 番目の department.pdf ファイルが上書きされます。その結果、最後に出力された部門のレポートのみが残ることになります。グループ化されたレポートをグループ・インスタンスごとに個別に出力する場合は、ファイル名を指定する変数値を使用します。たとえば、name="department_<department_id>.pdf" と指定します。

この項で取り上げる例は、次のとおりです。

- レポート全体のファイル
- 結合されたレポート・セクションのファイル
- 結合されたセクションからなる各グループのファイル
- レポート・グループ・インスタンスごとのファイル

15.4.3.1 レポート全体のファイル

この例では、レポート全体を含む report.pdf というファイルが作成されます。

15.4.3.1.1 Windows

```
<file id="a1" name="c:\reports\report.pdf" format="pdf">
<include src="report"/>
</file>
```

15.4.3.1.2 UNIX

```
<file id="a1" name="/reports/report.pdf" format="pdf">
<include src="report"/>
</file>
```

15.4.3.2 結合されたレポート・セクションのファイル

この例では、レポートのヘッダー・セクションとメイン・セクションで構成されるレポートを含んだ、sections.pdf というファイルが作成されます。

```
<file id="a2" name="sections.pdf" format="pdf">
<include src="headerSection"/>
<include src="mainSection"/>
</file>
```

15.4.3.3 結合されたセクションからなる各グループのファイル

この例では、繰り返しグループごとに異なるファイルが作成されます。各ファイルには、関連するグループのメイン・セクションとトレーラ・セクションを結合したレポートが格納されます。メイン・セクションとトレーラ・セクションが、同じグループで繰り返される必要があります。また、変数のファイル名が、"繰り返し"グループ内の列を参照する必要があります。つまり、レポートが department_id で繰り返され、10 から 40 までの 4 つの部門が存在する場合は、あるファイルに部門 10 のメイン・セクションとトレーラ・セクションが収められ、次のファイルに部門 20 のメイン・セクションとトレーラ・セクションが収められ、残りの部門も同様に処理されます。name の中の変数値は、G_department_id グループに含まれている列を参照する必要があります。

```
<foreach>
<file id="file9" name="department_&lt;department_id&gt;.pdf" instance="this">
<include src="mainSection"/>
<include src="trailerSection"/>
</file>
</foreach>
```

15.4.3.4 レポート・グループ・インスタンスごとのファイル

この例では、レポートが department_id でグループ化され、10 から 40 までの 4 つの部門が存在することを前提として、4 つのファイルを作成します。それぞれのファイル名は department_10.pdf、department_20.pdf、department_30.pdf、および department_40.pdf です。

```
<foreach>
<file id="a20" name="department_&lt;department_id&gt;.pdf" instance="this">
<include src="report"/>
</file>
</foreach>
```

15.4.4 printer の例

この項で取り上げる例は、次のとおりです。

- レポート全体の印刷
- レポートの2つのセクションの印刷
- グループ化されたレポートの印刷
- 結合されたセクションをグループ・インスタンスごとに印刷
- レポートの適切なインスタンスをその関連プリンタに印刷

プリンタ名の指定方法は、Windows と UNIX で異なります。各例では、それぞれの指定方法を示します。

15.4.4.1 レポート全体の印刷

この例では、レポート全体が指定のプリンタに送信されます。

15.4.4.1.1 Windows

```
<printer id="a80" name="¥¥neptune¥prtr20">
<include src="report"/>
</printer>
```

15.4.4.1.2 UNIX

```
<printer id="a80" name="10th_floor_printer">
<include src="report"/>
</printer>
```

15.4.4.2 レポートの2つのセクションの印刷

この例では、レポートの2つのセクションがプリンタに送信されます。

15.4.4.2.1 Windows

```
<printer id="a1" name="¥¥neptune¥prtr20">
<include src="headerSection"/>
<include src="mainSection"/>
</printer>
```

15.4.4.2.2 UNIX

```
<printer id="a1" name="10th_floor_printer">
<include src="headerSection"/>
<include src="mainSection"/>
</printer>
```

15.4.4.3 グループ化されたレポートの印刷

この例では、1つのレポートが印刷されます。このレポートは、department_idなどでグループ化されます。そのためには、レポートのすべてのセクションを同じグループで繰り返す必要があります。

15.4.4.3.1 Windows

```
<foreach>
<printer id="prt20" name="¥¥neptune¥prtr20" instance="all">
<include src="report"/>
</printer>
</foreach>
```

15.4.4.3.2 UNIX

```
<foreach>
<printer id="prt20" name="10th_floor_printer" instance="all">
<include src="report"/>
</printer>
</foreach>
```

15.4.4.4 結合されたセクションをグループ・インスタンスごとに印刷

この例では、グループ・インスタンスごとに1つずつ印刷ジョブが生成されます。結合されたセクションを、同じグループで繰り返す必要があります。レポートが department_id で繰り返され、10 から 40 までの4つの部門が存在する場合は、4つの印刷ジョブが生成されます。つまり、部門10に1つのジョブが生成され、残りの部門にも1つずつジョブが生成されます。メイン・セクションとトレーラ・セクションの両方が、department_id で繰り返される必要があります。

15.4.4.4.1 Windows

```
<foreach>
<printer id="prt20" name="¥¥neptune¥prtr20" instance="this">
<include src="mainSection"/>
<include src="trailerSection"/>
</printer>
</foreach>
```

15.4.4.4.2 UNIX

```
<foreach>
<printer id="prt20" name="10th_floor_printer" instance="this">
<include src="mainSection"/>
<include src="trailerSection"/>
</printer>
</foreach>
```

15.4.4.5 レポートの適切なインスタンスをその関連プリンタに印刷

この例が機能するには、repeat on グループにホスト・プラットフォームの適切なプリンタ名を含んだ列が含まれている必要があります。つまり、printer_name 列に適切なプリンタの別名 (UNIX) またはプリンタ・サーバー / 名前の組合せ (Windows) が含まれている必要があります。たとえば、レポートが department_id でグループ化される場合は、G_department_id に printer_name 列が必要となります。printer_name が部門に関連付けられていると仮定して、部門10のレポートが部門10のプリンタに印刷され、同様に残りの部門のレポートがその部門のプリンタに印刷されます。

```
<foreach>
<printer id="a60" name="&printer_name" instance="this">
<include src="mainSection"/>
</printer>
</foreach>
```

各グループ・インスタンスは、個別の印刷ジョブに相当します。各印刷ジョブは、関連する部門のプリンタに送信されます。

15.4.5 destype の例

destype を使用すると、配布時に、Oracle Reports で使用可能なカスタムの宛先、またはプラグラブル・デスティネーションを定義できます。詳細は、[第 15.3.9 項「destype」](#) を参照してください。この項では、次の宛先の例について説明します。

- [OracleAS Portal 宛先](#)
- [FTP 宛先](#)
- [WebDAV 宛先](#)
- [FAX 宛先](#)

15.4.5.1 OracleAS Portal 宛先

この例では、レポート出力を OracleAS Portal の宛先に送信する場合の汎用のタグ構造を示します。DESTYPE=ORACLEPORTAL を使用してレポート出力を OracleAS Portal に配信するときは、レポート出力は PAGEDGROUP フォルダに作成されます。

関連項目： この例にあるプロパティの詳細は、[付録 A「コマンドライン・キーワード」](#) を参照してください。

```
<destinations>
  <destype id="customforPortal" name="oraclePortal">
    <property name="outputpage" value="sample_report"/>
    <property name="statuspage" value="Reports_Status"/>
    <property name="desformat" value="pdf"/>
    <property name="pagegroup" value="REPORTS_OUTPUT"/>
    <property name="itemtitle" value="MyReport"/>
    <include src="report"/>
  </destype>
</destinations>
```

15.4.5.2 FTP 宛先

この例では、レポート出力を FTP の宛先に送信する場合の汎用のタグ構造を示します。

```
<destinations>
  <foreach>
    <destype id="ftpl" name="ftp" instance="this" format="pdf">
      <property name="desname"
        value="ftp://username:passwd@ftpServer/dir/myreport_&lt;
        DEPARTMENT_NAME&gt;.pdf"/>
      <include src="mainSection"/>
    </destype>
  </foreach>
</destinations>
```

15.4.5.3 WebDAV 宛先

この例では、レポート出力を WebDAV の宛先に送信する場合の汎用のタグ構造を示します。

```
<destinations>
  <foreach>
    <destype id="webdav1" name="webdav" instance="this" format="pdf">
      <property name="desname"
        value="http://user:passwd@WebDAVServer/dir/myreport_&lt;
        DEPARTMENT_NAME&gt;.pdf"/>
      <include src="mainSection"/>
    </destype>
  </foreach>
</destinations>
```

15.4.5.4 FAX 宛先

この例では、レポート出力を FAX の宛先に送信する場合の汎用のタグ構造を示します。

```
<destype id="faxdest" name="fax">
  <property name="number" value="123456789"/>
  <include src="report"/>
</destype>
```

これとは別に、使いやすさを考慮して、次のようなより具体的なカスタムのタグ構造を指定することもできます。

```
<fax id="faxdest" number="123456789">
  <include src="report"/>
</fax>
```

注意： `distribution.xsl` ファイルを変更した後は、そのファイルと同じ位置に同じファイル名で保存します。Oracle Reports では、宛先を解決する際に自動的にこの XSL ファイルが検索されます。

15.5 実行時の配布 XML ファイルの使用方法

実行時に配布 XML ファイルを使用する方法は、URL またはコマンドラインのどちらで使用する場合も基本的に同じです。次のオプションを使用します。

```
destination=filename.xml distribute=yes
```

filename は、配布 XML ファイルの名前です。XML ファイルの相対パスまたは絶対パスを指定する必要があります。たとえば、Windows の場合は次のように指定します。

```
destination=c:\%ORACLE_HOME%\reports\distribution\%filename.xml distribute=yes
```

UNIX の場合は次のように指定します。

```
destination=$ORACLE_HOME/reports/distribution/filename.xml distribute=yes
```

URL での完全なコマンドは次のようになります。

```
http://your_server:port/reports/rwservlet?report=rep.jsp&userid=db_credentials
&destination=$ORACLE_HOME/reports/distribution/distribution.xml&distribute=yes
```

例に示されているパスは、一例にすぎません。配布 XML ファイルの格納場所には制約がありません。必要な場所に保存することができます。

注意： Microsoft Internet Explorer では、URL の戻りストリームの MIME タイプが無視され、URL を参照し MIME タイプが設定される場合があります。OracleAS Reports Services の配布機能を使用しているときは、これが問題となることがあります。たとえば、URL の最後に次のような `destination` パラメータがあるとします。

```
...distribute=yes
destination=c:\%oracle%\reports\distribution\%mydist.xml
```

この場合、URL が拡張子 `.xml` で終わるため、Internet Explorer では、URL の戻りストリームが実際は HTML であったとしても、XML として処理されます。結果として、ブラウザ・エラーが発生します。この問題を回避するには、URL の最後に一般的なファイル拡張子を使用しないようにします。この例では、URL で `distribute` パラメータと `destination` パラメータの位置を入れ替えます。

コマンドラインおよび URL からレポートを実行する操作、および `cgicmd.dat` ファイルの使用に関する詳細は、第 13 章「レポート・リクエストの実行」を参照してください。

15.6 配布の使用に関する制限

この項では、Oracle Reports で配布を使用する場合の各種の制限について説明します。

- [OracleAS Portal 宛先](#)
- [XML 出力](#)
- [デリミタ付きおよび DelimitedData 出力](#)
- [スプレッドシート出力](#)
- [動的フォーマット値](#)

15.6.1 OracleAS Portal 宛先

Oracle9i Reports リリース 2 (9.0.2) 以降では、Oracle Reports は OracleAS Portal を宛先としてサポートします。DESTYPE=ORACLEPORTAL を使用すると、OracleAS Portal で指定された出力ページにレポートをプッシュできます。

関連項目： [第 A.3.29 項「DESTYPE」](#)

ただし、この宛先の使用には次の制限があります。

- 配信の宛先に ORACLEPORTAL は使用できません。かわりに、DESTYPE=WEBDAV を使用して、XML ベースの拡張配布を OracleAS Portal に対し実行できます。

注意： OracleAS Portal インスタンスが WebDAV 対応になっている必要があります。WebDAV を有効にする方法の詳細は、OracleAS Portal のオンライン・ヘルプを参照してください。

OracleAS Portal への配布に WebDAV を使用方法の詳細は、Oracle MetaLink (<http://metalink.oracle.com>) のノート 241821.1 「How to Send and Distribute Reports 9i Output to Oracle Portal?」を参照してください。

- DESTYPE=ORACLEPORTAL は、rwrund 実行可能ファイルでは使用できません。これを使用すると、Reports Server からのレスポンスが停止します。この宛先は、rwservlet、rwclient または rwcgi に対してのみ使用できます。

15.6.2 XML 出力

instance=this の指定による、レポート一括処理用参照パラメータの使用は、XML 出力についてはサポートされていません。使用すると、次のエラー・メッセージが表示されます。

```
REP-34310 "Reference parameter not allowed in distribution list for XML destination files"
```

参照パラメータを使用しなくても XML フォーマットを配布に使用できます。

15.6.3 デリミタ付きおよび DelimitedData 出力

デリミタ付き出力では配布と一括処理がサポートされていません。distribution.xml ファイルでは、DELIMITED または DELIMITEDDATA 出力フォーマットを指定できません。

15.6.4 スプレッドシート出力

スプレッドシート出力では配布と一括処理がサポートされていません。distribution.xml ファイルでは、SPREADSHEET 出力フォーマットを指定できません。そのため、配布モジュールにページ区切りのデフォルトの出力を生成する現在のモードは、スプレッドシートで機能しません。

15.6.5 動的フォーマット値

XML 配布では、`distribution.dtd`にあるように、`format` 属性の静的値のみがサポートされます。そのため、実行時に解決される、`format` 属性の文字パラメータは指定できません。したがって、レポート全体についても、特定のセクションについても、`format` を動的に決定することはできません。

XML によるレポートのカスタマイズ

Extensible Markup Language (XML) は、情報をより詳細に識別する方法を実現することにより、Web の機能を向上させることを目的として設計された言語です。XML はメタ言語（他の言語を定義するための言語）です。XML を使用して、様々なタイプのドキュメント用にカスタマイズされたマークアップ言語を設計できます。

XML ドキュメントは、マークアップとその内容の両方で構成されます。

- **要素**は、XML のビルディング・ブロックです。要素インスタンスは、タグ（メインのタグと適切にネストされたタグ）、属性、タグとタグの間にネストされた要素内容からなる構造を持ちます。
- **タグ**は、要素とその内容を定義するために使用されます。
- **属性**は、各タグの追加情報を提供するものです。

XML カスタマイズを使用すると、実行時に元のレポートを変更することなくレポートをカスタマイズできます。ランタイム・コマンドラインに CUSTOMIZE キーワードを追加し、カスタマイズ・ファイルを呼び出して、レポートのレイアウトやデータ・モデルへの追加や変更を実行できます。1 つの XML カスタマイズ・ファイルで、これらすべての作業または任意の作業の組合せを実行できます。XML を使用して、カスタムの JSP ベース・レポートに取り込むレポート・データ・モデルを作成することも可能です。

様々な XML カスタマイズを作成して適用すると、レポート出力をユーザー単位またはユーザー・グループ単位で変更できます。同じレポートを使用して、利用対象者ごとに異なる出力を生成できます。

XML カスタマイズをレポートに適用するときに、結合した定義をファイルに保存することができます。このため、XML カスタマイズを使用して、既存のレポート定義をバッチ更新できます。Reports Builder で 1 つずつファイルを開くことなく、大量のレポート定義をすばやく更新できます。

OracleAS Reports Services では、XML でレポート・データ・モデル全体を作成できるようになり、Oracle Reports XML によるカスタマイズのタイプが拡張されました。これには、複数データソースの作成、データソース間のリンク、および各データソース内でのグループ階層が含まれます。Oracle Reports XML カスタマイズによるデータ・モデルのサポートを通じて、Reports Builder で作成可能なすべてのデータ・モデルを、XML を指定して作成できるようになりました。さらに、データ・モデル・オブジェクトに設定可能なすべてのプロパティを、XML で設定できるようになりました。

この章では、XML を使用して実行中にレポートをカスタマイズする方法、および XML でデータ・モデルを作成する方法について説明します。この章には、次の項があります。

- [カスタマイズの概要](#)
- [XML カスタマイズの作成](#)
- [XML データ・モデルの作成](#)
- [実行時の XML ファイルの使用方法](#)
- [XML レポート定義のデバッグ](#)

この章では、reports.dtd ファイルでサポートされる要素を紹介し、その例を示します。ただし、これらの要素の属性については、一部のみを紹介します。

追加属性、または Oracle Reports XML の要素、タグおよび属性の詳細は、次を参照してください。

- reports.dtd ファイルには、Oracle Reports XML のすべての要素、タグと属性、そして存在する場合は属性のデフォルト値が含まれます。reports.dtd ファイルは、Windows と UNIX どちらの場合も、`ORACLE_HOME¥reports¥dtd¥` にあります。サブ要素の多くは、使用時の規則を示す記号を含んでいます。次に例を示します。
 - プラス記号 (+) は、XML ファイルでそのタイプの要素を 1 つ以上使用できることを意味します。
 - アスタリスク (*) は、XML ファイルでそのタイプの要素を 0 ~ 多数の範囲で使用できることを意味します。
 - 疑問符 (?) は、XML ファイルでそのタイプの要素を 0 または 1 つだけ使用できることを意味します。
 - 記号が付いていない場合は、XML ファイルでそのタイプの要素を 1 つだけ使用する必要がある (必須である) ことを意味します。

複数のサブ要素が括弧で囲まれ、その直後に記号が付いている場合は、その記号が括弧内のすべてのサブ要素に適用されます。

- 選択した Oracle Reports XML タグの詳細は、Oracle Reports のオンライン・ヘルプの「リファレンス」の項のトピック「Oracle Reports XML タグ」を参照してください。
- 作成しようとするカスタマイズのタイプを含んだレポートを作成し、レポートを XML として保存し、さらにテキスト・エディタで保存したファイルを表示します。これは、実際の Oracle Reports XML を参照する優れた手段であると同時に、より複雑なモデルを構築する場合の参考になります。

16.1 カスタマイズの概要

Oracle Reports XML タグを使用して、Reports Builder を使用して作成したレポートをカスタマイズできます。

注意： Oracle Reports XML のタグを使用してレポート全体を手動で作成することも可能ですが、ドキュメントに記載され、サポートされているのは、手動で作成されるカスタマイズおよびデータ・モデルのみです。

XML カスタマイズの作成と適用は、3 つのステップで行われます。

1. Oracle Reports XML のタグを使用してカスタマイズ・ファイルを作成します。

このカスタマイズは、Reports Builder でレポートを作成し、そのレポートを XML として保存することで作成できます。任意のテキスト・エディタまたは高機能の XML エディタを使用して、手動でカスタマイズを作成することもできます。これは、特別な Oracle Reports カスタマイズに必要な XML タグを使用する場合にかざられます。
2. XML カスタマイズを、OracleAS Reports Services でアクセス可能な場所に格納します。
3. XML カスタマイズを、CUSTOMIZE コマンドライン・キーワードまたは SRW.APPLY_DEFINITION ビルトイン・プロシージャを使用して、別のレポートに適用します。カスタマイズに完全なレポート定義が含まれている場合は、REPORT (または MODULE) コマンドライン・キーワードを使用して XML カスタマイズを単独で実行できます。

関連項目： [第 A.3.20 項「CUSTOMIZE」](#)

注意： SRW.APPLY_DEFINITION ビルトイン・プロシージャを含んだ SRW ビルトイン・パッケージの説明については、Oracle Reports のオンライン・ヘルプを参照してください。

16.2 XML カスタマイズの作成

この項では、様々なレポート・カスタマイズの例を示します。ここで取り上げる例は、次のとおりです。

- 必須 XML タグ
- スタイルの変更
- 書式マスクの変更
- 書式例外の追加
- プログラム・ユニットとハイパーリンクの追加
- 新規問合せの追加と新規ヘッダー・セクションでの結果の使用
- URL のエンコード

16.2.1 必須 XML タグ

すべての XML カスタマイズには、次のタグ・ペアが必要です。

```
<report></report>
```

たとえば、最小限の XML カスタマイズは次のとおりです。

```
<report name="emp" DTDVersion="9.0.2.0.0">
</report>
```

この XML カスタマイズには何も含まれていないので、レポートに適用しても効果は一切ありません。必要なタグがあるので解析はできますが、単に必須タグの例として示しています。

<report> タグは、レポート・カスタマイズの開始、レポート・カスタマイズ名およびこの XML カスタマイズで使用するデータ型ディクショナリ (DTD) ファイルのバージョンを示します。</report> タグは、レポート・カスタマイズの終了を示します。

report タグの name 属性には、任意の名前を指定できます。たとえば、XML ファイルでカスタマイズするレポートの名前でも、それ以外の名前でもかまいません。

この例では、<report> タグが最小限で使用されています。<report> タグにも多数の属性がありますが、そのほとんどは暗黙的に示されており、指定する必要はありません。<report> に唯一必要な属性は、DTDVersion です。

注意： XML カスタマイズ・ファイルを適用して、既存のレポート・トリガーを変更するか新しいレポート・トリガーを作成するには、<report> タグの関連トリガー属性を指定する必要があります。

たとえば、Before Report トリガーを変更または作成するには、beforeReportTrigger 属性を使用します。

```
<report DTDVersion="9.0.2.0.0" beforeReportTrigger="BeforeReport">
```

XML カスタマイズ・ファイルを適用してレポート・トリガーを変更または作成するときにこの属性を指定しないと、XML カスタマイズ・ファイルがレポートに適用されたときに、レポート・トリガーの PL/SQL コードがローカル (依存しない) ファンクションとして扱われます。

完全なレポート定義には、データ・モデルとレイアウトの両方が必要です。したがって、次のタグとその内容を定義する必要があります。

- `<data></data>`
- `<layout></layout>`

`data` タグには、属性がありません。`layout` タグには、`panelPrintOrder` と `direction` の 2 つの属性があり、その両方が必須です。これらの属性にデフォルト値（それぞれ `acrossDown` と `default`）を使用する場合は、属性を指定する必要がありません。`data` 要素と `layout` 要素の例は、次の各項で示します。

16.2.2 スタイルの変更

この項の例では、XML を使用して、レポート・フィールド `F_Mincurrent_pricePersymbol` および `F_Maxcurrent_pricePersymbol` に使用されている塗りつぶしカラーと線カラーを変更しています。

```
<report name="anyName" DTDVersion="9.0.2.0.0">
  <layout>
    <section name="main">
      <field name="F_Mincurrent_pricePersymbol"
        source="Mincurrent_pricePersymbol"
        lineColor="black"
        fillColor="r100g50b50"/>
      <field name="F_Maxcurrent_pricePersymbol"
        source="Maxcurrent_pricePersymbol"
        lineColor="black"
        fillColor="r100g50b50"/>
    </section>
  </layout>
</report>
```

この例では、`section` タグと `field` タグの `name` 属性が、この XML ファイルでカスタマイズされるレポートのメイン・セクションに含まれるフィールドの名前と一致していることが前提となっています。この前提を踏まえて、`field` タグのその他の属性は、レポートのメイン・セクション内にある同じ名前のフィールドのみに適用されます。

16.2.3 書式マスクの変更

この項の例では、XML を使用して、レポート・フィールド `f_trade_date` に使用される書式マスクを変更しています。

```
<report name="anyName" DTDVersion="9.0.2.0.0">
  <layout>
    <section name="main">
      <field name="f_trade_date"
        source="trade_date"
        formatMask="MM/DD/RR"/>
    </section>
  </layout>
</report>
```

`field` タグが独自の終了書式 (`/>`) を使用していることに注意してください。`field` タグが追加の子タグを使用する場合は、`field` タグを `</field>` で閉じます。

16.2.4 書式例外の追加

この項の例では、XML を使用して、レポートの `f_p_e` および `f_p_e1` フィールドで 10 より大きな値をハイライトする書式例外を追加しています。

```
<report name="anyName" DTDVersion="9.0.2.0.0">
  <layout>
    <section name="main">
      <field name="f_p_e" source="p_e">
        <exception textColor="red">
          <condition source="p_e" operator="gt" operand1="10"/>
        </exception>
      </field>
      <field name="f_p_e1" source="p_e">
        <exception textColor="blue">
          <condition source="p_e" operator="gt" operand1="10"/>
        </exception>
      </field>
    </section>
  </layout>
</report>
```

この例では、`operator` の値が、" 大なり " を示す `gt` に設定されています。`operator` には、表 16-1 に示すものを指定できます。

表 16-1 operator 属性の値

値	使用方法
<code>eq</code>	等しい
<code>lt</code>	より小さい
<code>lteq</code>	以下
<code>neq</code>	等しくない
<code>gt</code>	より大きい
<code>gteq</code>	以上
<code>btw</code>	範囲内
<code>notBtw</code>	範囲外
<code>like</code>	類似
<code>notLike</code>	非類似
<code>nNULL</code>	NULL
<code>notNull</code>	非 NULL

前の例とは異なり、この例の `field` タグは、子タグを使用しているために、自己完結型の終了書式 (`/>`) ではなく、`</field>` で閉じられていることにも注意してください。

16.2.5 プログラム・ユニットとハイパーリンクの追加

この項の例では、XML を使用してレポートにプログラム・ユニットを追加しています。このプログラム・ユニットによって、従業員の社会保障番号 (:SSN) から従業員の詳細にアクセスするハイパーリンクが追加されます。

```
<report name="anyName" DIDVersion="9.0.2.0.0">
  <layout>
    <section name="header">
      <field name="F_ssn1" source="ssn1">
        <advancedLayout formatTrigger="F_ssn1FormatTrigger"/>
      </field>
    </section>
    <section name="main">
      <field name="F_ssn" source="ssn">
        <advancedLayout formatTrigger="F_ssnFormatTrigger"/>
      </field>
    </section>
  </layout>
  <programUnits>
    <function name="F_ssn1FormatTrigger">
      <textSource>
        <![CDATA[
          function F_ssn1FormatTrigger return boolean is
          begin
            SRW.SET_HYPERLINK('#EMP_DETAILS_&<' || LTRIM(TO_CHAR(:SSN)) || '>');
            return (TRUE);
          end;
        ]]>
      </textSource>
    </function>
    <function name="F_ssnFormatTrigger">
      <textSource>
        <![CDATA[
          function F_ssnFormatTrigger return boolean is
          begin
            SRW.SET_LINKTAG('EMP_DETAILS_&<' || LTRIM(TO_CHAR(:SSN)) || '>');
            return (TRUE);
          end;
        ]]>
      </textSource>
    </function>
  </programUnits>
</report>
```

CDATA タグは、PL/SQL を XML と区別する目的で、PL/SQL の周辺で使用されています。XML ファイルに HTML を埋め込むときも、同じタグ・シーケンスを使用します。この例では、advancedLayout タグの formatTrigger 属性の名前でファンクションが参照されています。

16.2.6 新規問合せの追加と新規ヘッダー・セクションでの結果の使用

この項の例では、XML を使用して、レポートに新しい問合せを追加するとともに、その問合せの結果を利用する新しいヘッダー・セクションを追加しています。

```
<report name="ref" DTDVersion="9.0.2.0.0">
  <data>
    <dataSource name="Q_summary">
      <select>select portid ports, locname locations from portdesc</select>
    </dataSource>
  </data>
  <layout>
    <section name="header">
      <tabular name="M_summary" template="BLAFbeige.tdf">
        <labelAttribute font="Arial" fontSize="10"
          fontStyle="bold" textColor="white"/>
        <field name="F_ports" source="ports" label="Port IDs"
          font="Arial" fontSize="10"/>
        <field name="F_locations" source="locations" label="Port Names"
          font="Arial" fontSize="10"/>
      </tabular>
    </section>
  </layout>
</report>
```

この例の XML は、データ・モデルと完全なレイアウトを備えているため、単独で実行できます。

列名が必ず一意となるように、SELECT 文では別名を使用します。別名を使用しなかった場合は、レポート列のデフォルト名が使用されるため、予定外の名前になる可能性があります（たとえば、portid ではなく portid1 など）。field タグの source 属性を指定する場合は、ソース列（フィールド）の正しい名前を使用する必要がありますので、この名前指定は重要なポイントになります。

labelAttribute 要素は、レイアウト内のフィールド・ラベルの書式を定義します。これは field タグの開始と終了タグの外側にあるため、表形式レイアウトのすべてのラベルに適用されます。この書式をいずれか 1 つのフィールドに関連付ける場合は、この要素を <field></field> タグ・ペアの内側に配置します。グローバルおよびローカルの labelAttribute 要素が両方とも（1 つは <field></field> タグ・ペアの外側に、もう 1 つは内側に）存在する場合は、ローカルなラベル属性がグローバルな属性より優先されます。

16.2.7 URL のエンコード

スペースおよび制御文字が正しく渡されるようにするには、レポートの各フィールドに対して URL エンコードをオンまたはオフにする必要があります。URL エンコードのオンとオフの切替えは、レポートに RW:FIELD タグを使用します。

```
<rw:field
...
urlEncode=yes|no
...
/>
```

urlEncode のデフォルト値は、no です。

16.3 XML データ・モデルの作成

OracleAS Reports Services では、Oracle Reports XML のタグを使用して、より高度なデータ・モデルのタイプを作成できるようになりました。次の処理に Oracle Reports XML を使用できます。

- 複数のデータソースの作成
- データソース間のリンク
- 各データソース内でのグループ階層の作成
- クロス積（マトリックス）グループの作成
- あらゆるレベルでの式、サマリーおよびプレースホルダの作成
- パラメータの作成

この項では、Oracle Reports XML を使用した処理の例を示します。

これらのデータ・モデル・タイプに加えて、OracleAS Reports Services では XML での PL/SQL の使用をサポートしています。これには、ローカルなプログラム・ユニット、レポートレベルのトリガーおよび連結 PL/SQL ライブラリに対するサポートが含まれます。

16.3.1 複数のデータソースの作成

<data> タグが、新しいプラグgable・データソースと複数のデータソースの作成をサポートするようになりました。各データソースは、それぞれの <dataSource> タグ内に記述されます。dataSource 要素のデータ型定義は、次のとおりです。

```
<!ELEMENT dataSource
  ((select|plugin|plsql),
  comment?,
  displayInfo?,
  formula*,
  group*)>
<!ATTLIST dataSource
  name CDATA #IMPLIED
  defaultGroupName CDATA #IMPLIED
  maximumRowsToFetch CDATA #IMPLIED>
```

次の例では、2つの SQL データソースを作成し、それぞれに Q_1 および Q_2 という名前を付けます。また、各データソースに必要なすべての列を作成するとともに、デフォルト・グループを作成します。このグループの名前には、指定の defaultGroupName が設定されます。defaultGroupName が指定されていない場合は、デフォルトで独自の名前が設定されます。

```
<report name="anyname" DIDVersion="9.0.2.0.0">
  <data>
    <dataSource name="Q_1" defaultGroupName="G_DEPARTMENTS">
      <select>
        select * from departments
      </select>
    </dataSource>
    <dataSource name="Q_2" defaultGroupName="G_EMPLOYEES">
      <select>
        select * from employees
      </select>
    </dataSource>
  </data>
</report>
```


16.3.2 データソース間のリンク

複数のデータソースが存在するときは、それらのデータソースをリンクして適切なデータ・モデルを作成することが必要となる場合もあります。Oracle Reports XML では、Oracle Reports データ・モデル・リンク・オブジェクトも公開されています。このオブジェクトは、グループレベルおよび列レベルのリンクをサポートします。任意の数のリンクを指定して、必要なデータ・モデルを作成できます。

link 要素のデータ型定義は、次のとおりです。

```
<!ELEMENT link EMPTY>
<!ATTLIST link
  name CDATA #IMPLIED
  parentGroup CDATA #IMPLIED
  parentColumn CDATA #IMPLIED
  childQuery CDATA #IMPLIED
  childColumn CDATA #IMPLIED
  condition (eq|lt|neq|gt|gteq|like|notLike) "eq"
  sqlClause (startWith|having|where) "where">
```

link 要素は、data 要素の内部に配置され、data 要素内に定義されている dataSource オブジェクトのうち任意の 2 つをリンクします。次に例を示します。

```
<report name="anyname" DIDVersion="9.0.2.0.0">
  <data>
    <dataSource name="Q_1" defaultGroupName="G_DEPARTMENTS">
      <select>
        select * from departments
      </select>
    </dataSource>
    <dataSource name="Q_2" defaultGroupName="G_EMPLOYEES">
      <select>
        select * from employees
      </select>
    </dataSource>
    <link name="L_1" parentGroup="G_DEPARTMENTS"
      parentColumn="DEPARTMENT_ID" childQuery="Q_2"
      childColumn="DEPARTMENT_ID1" condition="eq" sqlClause="where"/>
  </data>
</report>
```

link 要素の内部では、Oracle Reports のデフォルト設定メカニズムによって、DEPARTMENT_ID1 が EMPLOYEES 表における DEPARTMENT_ID 列の別名と認識されます。この場合、ユーザーが明示的に別名を作成する必要はありません。

16.3.3 各データソース内でのグループ階層の作成

OracleAS Reports Services では、完全なグループ階層を利用することができます。各グループ内ですべての列を指定し、それらの列の順序をブレイクすることができます。式、サマリーおよびプレースホルダを使用して、グループ内でさらにオブジェクトをカスタマイズできます。

group 要素のデータ型定義は、次のとおりです。

```
<!ELEMENT group
  (field|exception|rowDelimiter|xmlSettings|displayInfo|dataItem|formula|
  summary|placeholder|filter|comment)*>
<!ATTLIST group
  name CDATA #IMPLIED
  fillColor CDATA #IMPLIED
  lineColor CDATA #IMPLIED
  formatTrigger CDATA #IMPLIED>
```

次の例では、group 要素を使用して、データソースの下にブレイク・グループを作成していません。

```
<report name="anyname" DTDVersion="9.0.2.0.0">
  <data>
    <dataSource name="Q_1">
      <select>
        select * from employees
      </select>
      <group name="G_DEPARTMENTS">
        <dataItem name="DEPARTMENT_ID"/>
      </group>
      <group name="G_EMPLOYEES">
        <dataItem name="EMPLOYEE_ID"/>
        <dataItem name="FIRST_NAME"/>
        <dataItem name="LAST_NAME"/>
        <dataItem name="JOB_ID"/>
        <dataItem name="MANAGER_ID"/>
        <dataItem name="HIRE_DATE"/>
        <dataItem name="SALARY"/>
        <dataItem name="COMMISSION_PCT"/>
      </group>
    </dataSource>
  </data>
</report>
```

16.3.4 クロス積（マトリックス）グループの作成

クロス積グループを使用すると、データ・モデルに任意の数のグループのマトリックスを定義できます。クロス積の各ディメンション・グループは、同じデータソースに属している場合でも、様々なデータソースから結合されている場合でも、マトリックスを作成できます。この柔軟性に対応して、<crossProduct> タグは、すべてのデータソースとグループが作成された後で、<data> タグ内に配置されます。

crossProduct 要素のデータ型定義は、次のとおりです。

```
<!ELEMENT crossProduct
  (xmlSettings|displayInfo|dimension|(formula|summary|placeholder)*|comment)*>
<ATTLIST crossProduct
  name CDDATA #IMPLIED
  mailText CDDATA #IMPLIED>
```

次の例では、単一問合せマトリックスを作成しています。

```
<report name="anyname" DTDVersion="9.0.2.0.0">
  <data>
    <dataSource name="Q_1">
      <select>
        select * from employees
      </select>
      <group name="G_DEPARTMENTS">
        <dataItem name="DEPARTMENT_ID"/>
      </group>
      <group name="G_JOB_ID">
        <dataItem name="JOB_ID"/>
      </group>
      <group name="G_MANAGER_ID">
        <dataItem name="MANAGER_ID"/>
      </group>
      <group name="G_EMPLOYEE_ID">
        <dataItem name="EMPLOYEE_ID"/>
        <dataItem name="FIRST_NAME"/>
        <dataItem name="LAST_NAME"/>
        <dataItem name="HIRE_DATE"/>
      </group>
    </dataSource>
  </data>
</report>
```

```

        <dataItem name="SALARY"/>
        <dataItem name="COMMISSION_PCT"/>
    </group>
</dataSource>
<crossProduct name="G_Matrix">
    <dimension>
        <group name="G_DEPARTMENTS">
    </dimension>
    <dimension>
        <group name="G_JOB_ID">
    </dimension>
    <dimension>
        <group name="G_MANAGER_ID">
    </dimension>
    </crossProduct>
</data>
</report>

```

16.3.5 あらゆるレベルでの式、サマリーおよびプレースホルダの作成

式、サマリーおよびプレースホルダは、データ・モデル内のあらゆるレベルに配置できます。さらに、これらの各オブジェクトの属性は、どれも完全に制御することができます。

次の例では、ソースがグループ・レベルの式列に基づいている、レポートレベルのサマリーを作成しています。

```

<report name="anyname" DIDVersion="9.0.2.0.0">
    <data>
        <dataSource name="Q_1">
            <select>
                select * from employees
            </select>
            <group name="G_EMPLOYEES">
                <dataItem="EMPLOYEE_ID"/>
                <dataItem name="EMPLOYEE_ID"/>
                <dataItem name="FIRST_NAME"/>
                <dataItem name="LAST_NAME"/>
                <dataItem name="HIRE_DATE"/>
                <dataItem name="SALARY"/>
                <dataItem name="COMMISSION_PCT"/>
                <dataItem name="DEPARTMENT_ID"/>
                <formula name="CF_REMUNERATION" source="cf_1formula"
                    datatype="number" width="20" precision="10"/>
            </group>
        </dataSource>
        <summary name="CS_REPORT_LEVEL_SUMMARY" function="sum" width="20"
            precision="10" reset="report" compute="report"/>
    </data>
    <programUnits>
        <function name="cf_1formula" returnType="number">
            <textSource>
                <![CDATA[
                    function CF_1Formula return Number is
                    begin
                        return (:salary + nvl(:commission_pct,0));
                    end;
                ]]>
            </textSource>
        </function>
    </programUnits>
</report>

```

16.3.6 パラメータの作成

Oracle Reports XML では、parameter 要素が data タグの開始と終了の間に配置されます。parameter 要素のデータ型定義は、次のとおりです。

```
<!ELEMENT parameter (comment?|listOfValues?)>
<!ATTLIST parameter
  name CDATA #REQUIRED
  datatype (number|character|date) "number"
  width CDATA "20"
  scale CDATA "0"
  precision CDATA "0"
  initialValue CDATA #IMPLIED
  inputMask CDATA #IMPLIED
  validationTrigger CDATA #IMPLIED
  label CDATA #IMPLIED
  defaultWidth CDATA #IMPLIED
  defaultHeight CDATA #IMPLIED>
```

次の例は、動的値リスト（LOV）、初期値および妥当性チェックトリガーを示しています。

```
<report name="anyname" DTDVersion="9.0.2.0.0">
  <data>
    <dataSource name="Q_1" defaultGroupName="G_DEPARTMENTS">
      <select>
        select * from departments
      </select>
    </dataSource>
    <parameter name="P_LAST_NAME" datatype="character" precision="10"
      initialValue="SMITH" validationTrigger="p_last_namevalidtrigger"
      defaultWidth="0" defaultHeight="0">
      <listOfValues restrictToList="yes">
        <selectStatement hideFirstColumn="yes">
          <![CDATA[select last_name, 'last_name||'-'||employee_id'
            from employees]]>
        </selectStatement>
      </listOfValues>
    </parameter>
  </data>
  <programUnits>
    <function name="p_last_namevalidtrigger" returnType="character">
      <textSource>
        <![CDATA[function P_LAST_NAMEValidTrigger return boolean is
          last_name char(20);
          begin
            select count(*) into last_name from employees
              where upper(last_name)=upper(:p_last_name);
            exception when OTHERS then return(FALSE);
            end;
            return(TRUE);
          end;
        ]]>
      </textSource>
    </function>
  </programUnits>
</report>
```

16.4 実行時の XML ファイルの使用方法

作成した Oracle Reports XML カスタマイズ・ファイルは、次の方法で使用できます。

- CUSTOMIZE コマンドライン・キーワードまたは SRW.APPLY_DEFINITION ビルトイン・プロシージャを指定して、実行時に XML レポート定義を RDF またはその他の XML ファイルに適用します。詳細は、[第 16.4.1 項「実行時の XML レポート定義の適用」](#)を参照してください。

注意： Oracle Reports では、REP ファイルの XML カスタマイズはサポートされません。

- REPORT (または MODULE) コマンドライン・キーワードを指定して、XML レポート定義を単独で (他のレポートなしで) 実行します。詳細は、[第 16.4.2 項「XML レポート定義の単独での実行」](#)を参照してください。
- CUSTOMIZE コマンドライン・キーワードを使用して、rwconverter によるバッチ変更を実行します。詳細は、[第 16.4.3 項「バッチ変更の実行」](#)を参照してください。

この後の項では、各ケースを詳しく説明するとともに、例を示します。

16.4.1 実行時の XML レポート定義の適用

実行時に XML レポート定義を RDF またはその他の XML ファイルに適用するには、CUSTOMIZE コマンドライン・キーワードまたは SRW.APPLY_DEFINITION ビルトイン・プロシージャを使用します。CUSTOMIZE は、rwclient、rwrn、rwbuilder、rwconverter および URL レポート・リクエストとともに使用できます。

注意： CUSTOMIZE を rwconverter とともに使用する方法の詳細は、[第 16.4.3 項「バッチ変更の実行」](#)を参照してください。

16.4.1.1 1 つの XML レポート定義の適用

次に示すコマンドラインは、ジョブ・リクエストを OracleAS Reports Services に送信し、XML レポート定義 (emp.xml) を RDF ファイル (emp.rdf) に適用します。この例では、CUSTOMIZE キーワードが Windows のディレクトリ・パスにあるファイルを参照しています。UNIX の場合は、UNIX の標準に従ってパスを指定してください (つまり myreports/emp.xml)。

```
rwclient REPORT=emp.rdf CUSTOMIZE=%myreports%emp.xml
  USERID=username/password@my_db DESTYPE=file DESNAME=emp.pdf
  DESFORMAT=PDF SERVER=server_name
```

Reports Runtime のコマンドである rwrn を使用する場合は、次のとおりです。

```
rwrn USERID=username/password@my_db REPORT=emp.rdf
  CUSTOMIZE=%myreports%emp.xml DESTYPE=file DESNAME=emp.pdf
  DESFORMAT=PDF
```

XML レポート定義をテストするときは、トレース・ファイルを作成するオプションを追加してレポート・リクエストを実行すると便利な場合があります。次に例を示します。

```
TRACEFILE=emp.log TRACEMODE=trace_replace TRACEOPTS=trace_app
```

関連項目： [第 A.3.114 項「TRACEFILE」](#)、[第 A.3.115 項「TRACEMODE」](#)
および [第 A.3.116 項「TRACEOPTS」](#)

注意： Reports Server 構成ファイルにデフォルトのトレース・オプションを指定した場合は、そのデフォルトを変更する必要がないかぎり、コマンドラインにトレースを含める必要はありません。

トレース・ファイルにより、レポート・オブジェクトの作成および書式に関する詳細なリストが得られます。

16.4.1.2 複数の XML レポート定義の適用

実行時に複数の XML レポート定義をレポートに適用するには、CUSTOMIZE コマンドライン・キーワードにリストを指定します。次に示すコマンドラインは、2つの XML レポート定義 EMP0.XML および EMP1.XML を RDF ファイル EMP.RDF に適用するジョブ・リクエストを、OracleAS Reports Services に送信します。

```
rwclient REPORT=emp.rdf
CUSTOMIZE="(d:¥corp¥myreports¥emp0.xml,d:¥corp¥myreports¥emp1.xml)"
USERID=username/password@my_db DESTYPE=file DESNAME=emp.pdf
DESFORMAT=PDF SERVER=server_name
```

注意： この例では、CUSTOMIZE の値に、Windows プラットフォームに格納されているファイルへのディレクトリ・パスが示されています。UNIX の場合は、そのプラットフォームの標準に従ってディレクトリ・パスを指定してください（つまり円記号 (¥) のかわりに通常のスラッシュを使用）。

Reports Runtime を使用する場合のコマンドラインは、次のとおりです。

```
rwrun REPORT=emp.rdf
CUSTOMIZE="(D:¥CORP¥MYREPOORTS¥EMP0.XML,D:¥CORP¥MYREPORTS¥EMP1.XML)"
USERID=username/password@my_db DESTYPE=file DESNAME=emp.pdf
DESFORMAT=PDF
```

16.4.1.3 PL/SQL での XML レポート定義の適用

PL/SQL で RDF ファイルに XML レポート定義を適用するには、Before Parameter Form または After Parameter Form トリガーで SRW.APPLY_DEFINITION および SRW.ADD_DEFINITION ビルトイン・プロシージャを使用します。この後の項では、これらのビルトイン・プロシージャの例を示します。

注意： SRW.APPLY_DEFINITION および SRW.ADD_DEFINITION ビルトイン・プロシージャを含んだ SRW ビルトイン・パッケージの説明、およびレポート・トリガーの詳細は、Oracle Reports のオンライン・ヘルプを参照してください。

16.4.1.3.1 ファイルに保存された XML 定義の適用 ファイル・システムに保存された XML をレポートに適用するには、レポートの Before Parameter Form または After Parameter Form トリガーで SRW.APPLY_DEFINITION ビルトイン・プロシージャを使用します。

Windows の場合：

```
SRW.APPLY_DEFINITION ('%ORACLE_HOME%\TOOLS\DOC\US\RBBR\COND.XML');
```

UNIX の場合：

```
SRW.APPLY_DEFINITION ('$ORACLE_HOME/TOOLS/DOC/US/RBBR/COND.XML');
```

レポートの実行時に、トリガーが実行され、指定した XML ファイルがレポートに適用されます。

16.4.1.3.2 メモリーに保存された XML 定義の適用 メモリー内に XML レポート定義を作成するには、SRW.ADD_DEFINITION を使用してドキュメント・バッファに定義を追加した後に、SRW.APPLY_DEFINITION ビルトイン・プロシージャを使用してその定義を適用する必要があります。

次の例では、ユーザーが入力したパラメータ値に基づいてメモリー内にいくつかの定義を作成し、それらを適用する方法を示します。この例の PL/SQL は、videosales_custom.rdf というレポートの After Parameter Form トリガーで使用されます。

videosales_custom.rdf ファイルの After Parameter Form トリガーに入っている PL/SQL が実行する内容は、次のとおりです。

- 実行時にユーザーによって入力されたパラメータ値に基づいて、条件と合致したフィールドをハイライトします。
- 実行時にユーザーによって入力されたパラメータ値に基づいて、数値書式マスクを変更します。

次のヒントは、この例を参照する際に役立ちます。

- SRW.APPLY_DEFINITION ビルトイン・プロシージャを使用するたびに、ドキュメント・バッファがフラッシュされるので、そのたびに SRW.ADD_DEFINITION を使用して XML レポート定義を新規作成する必要があります。
- パラメータ hilite_profits、hilite_costs、hilite_sales および money_format を使用して、XML レポート定義に入れる内容を決定している点に注意してください。hilite_profits、hilite_costs および hilite_sales パラメータは、書式例外でも使用されており、これによってハイライトする値が決定します。
- VARCHAR2 列のサイズには上限があるため (4000 バイト)、かなり大きな XML レポート定義の場合は、複数の列にまたがって指定しなければならない場合もあります。その場合は、1 つの大きな定義を作成して一度に適用するのではなく、メモリー内に定義をいくつか作成し、別々に適用する必要があります。

```
function AfterPForm return boolean is
begin
  SRW.ADD_DEFINITION('<report name="vidsales_masks"
author="Generated" DTDVersion="9.0.2.0.0">');
  IF :MONEY_FORMAT='$$$NNN.00' THEN
    SRW.ADD_DEFINITION('<layout>');
    SRW.ADD_DEFINITION('<section name="main">');
    SRW.ADD_DEFINITION('<field name="F_TOTAL_PROFIT" source="TOTAL_PROFIT"
formatMask="LNNNNNNNNNNN0D00"/>');
    SRW.ADD_DEFINITION('<field name="F_TOTAL_SALES" source="TOTAL_SALES"
formatMask="LNNNNNNNNNNN0D00"/>');
    SRW.ADD_DEFINITION('<field name="F_TOTAL_COST" source="TOTAL_COST"
formatMask="LNNNNNNNNNNN0D00"/>');
    SRW.ADD_DEFINITION('<field name="F_SumTOTAL_PROFITPerCITY"
source="SumTOTAL_PROFITPerCITY" formatMask="LNNNNNNNNNNN0D00"/>');
    SRW.ADD_DEFINITION('<field name="F_SumTOTAL_SALESPerCITY"
source="SumTOTAL_SALESPerCITY" formatMask="LNNNNNNNNNNN0D00"/>');
    SRW.ADD_DEFINITION('<field name="F_SumTOTAL_COSTPerCITY"
source="SumTOTAL_COSTPerCITY" formatMask="LNNNNNNNNNNN0D00"/>');
    SRW.ADD_DEFINITION('</section>');
    SRW.ADD_DEFINITION('</layout>');
  ELSIF :MONEY_FORMAT='$$$NNN' THEN
    SRW.ADD_DEFINITION('<layout>');
    SRW.ADD_DEFINITION('<section name="main">');
    SRW.ADD_DEFINITION('<field name="F_TOTAL_PROFIT" source="TOTAL_PROFIT"
formatMask="LNNNNNNNNNNN0"/>');
    SRW.ADD_DEFINITION('<field name="F_TOTAL_SALES" source="TOTAL_SALES"
formatMask="LNNNNNNNNNNN0"/>');
    SRW.ADD_DEFINITION('<field name="F_TOTAL_COST" source="TOTAL_COST"
formatMask="LNNNNNNNNNNN0"/>');
    SRW.ADD_DEFINITION('<field name="F_SumTOTAL_PROFITPerCITY"
source="SumTOTAL_PROFITPerCITY" formatMask="LNNNNNNNNNNN0"/>');
```

```

SRW.ADD_DEFINITION('<field name="F_SumTOTAL_SALESPerCITY"
    source="SumTOTAL_SALESPerCITY" formatMask="LNNNNNNNNNNNO"/>');
SRW.ADD_DEFINITION('<field name="F_SumTOTAL_COSTPerCITY"
    source="SumTOTAL_COSTPerCITY" formatMask="LNNNNNNNNNNNO"/>');
SRW.ADD_DEFINITION('</section>');
SRW.ADD_DEFINITION('</layout>');
END IF;
SRW.ADD_DEFINITION('</report>');
SRW.APPLY_DEFINITION;
SRW.ADD_DEFINITION('<report name="vidsales_hilite_costs" author="Generated"
    DTDVersion="9.0.2.0.0">');
IF :HILITE_COSTS <> 'None' THEN
    SRW.ADD_DEFINITION('<layout>');
    SRW.ADD_DEFINITION('<section name="main">');
    SRW.ADD_DEFINITION('<field name="F_TOTAL_COST" source="TOTAL_COST">');
    SRW.ADD_DEFINITION('<exception textColor="red">');
    SRW.ADD_DEFINITION('<condition source="TOTAL_COST" operator="gt"
        operand1=":hilite_costs"/>');
    SRW.ADD_DEFINITION('</exception>');
    SRW.ADD_DEFINITION('</field>');
    SRW.ADD_DEFINITION('</section>');
    SRW.ADD_DEFINITION('</layout>');
END IF;
SRW.ADD_DEFINITION('</report>');
SRW.APPLY_DEFINITION;
SRW.ADD_DEFINITION('<report name="vidsales_hilite_sales" author="Generated"
    DTDVersion="9.0.2.0.0">');
IF :HILITE_SALES <> 'None' THEN
    SRW.ADD_DEFINITION('<layout>');
    SRW.ADD_DEFINITION('<section name="main">');
    SRW.ADD_DEFINITION('<field name="F_TOTAL_SALES" source="TOTAL_SALES">');
    SRW.ADD_DEFINITION('<exception textColor="red">');
    SRW.ADD_DEFINITION('<condition source="TOTAL_SALES" operator="gt"
        operand1=":hilite_sales"/>');
    SRW.ADD_DEFINITION('</exception>');
    SRW.ADD_DEFINITION('</field>');
    SRW.ADD_DEFINITION('</section>');
    SRW.ADD_DEFINITION('</layout>');
END IF;
SRW.ADD_DEFINITION('</report>');
SRW.APPLY_DEFINITION;
SRW.ADD_DEFINITION('<report name="vidsales_hilite_profits" author="Generated"
    DTDVersion="9.0.2.0.0">');
IF :HILITE_PROFITS <> 'None' THEN
    SRW.ADD_DEFINITION('<layout>');
    SRW.ADD_DEFINITION('<section name="main">');
    SRW.ADD_DEFINITION('<field name="F_TOTAL_PROFIT" source="TOTAL_PROFIT">');
    SRW.ADD_DEFINITION('<exception textColor="red">');
    SRW.ADD_DEFINITION('<condition source="TOTAL_PROFIT" operator="gt"
        operand1=":hilite_profits"/>');
    SRW.ADD_DEFINITION('</exception>');
    SRW.ADD_DEFINITION('</field>');
    SRW.ADD_DEFINITION('</section>');
    SRW.ADD_DEFINITION('</layout>');
END IF;
SRW.ADD_DEFINITION('</report>');
SRW.APPLY_DEFINITION;
return (TRUE);
end;

```


16.4.2 XML レポート定義の単独での実行

XML レポート定義を単独で実行するには、REPORT (または MODULE) オプションに XML ファイルを指定してリクエストを送信します。次のコマンドラインは、レポート emp.xml を単独で実行するジョブ・リクエストを OracleAS Reports Services に送信します。

```
rwclient USERID=username/password@my_db
REPORT=c:\%corp%\myreports\emp.xml
DESTTYPE=file desname=emp.pdf DESFORMAT=pdf
SERVER=server_name
```

Reports Runtime のコマンドである rwrun を使用する場合は、次のとおりです。

```
rwrun USERID=username/password@my_db
REPORT=c:\%corp%\myreports\emp.xml
DESTTYPE=file DESNAME=emp.pdf DESFORMAT=PDF
```

この方法で XML レポート定義を実行する場合は、必ず XML ファイルの拡張子を指定します。また、CUSTOMIZE コマンドライン・キーワードを使用して、XML カスタマイズ・ファイルをこのレポートに適用することもできます。

16.4.3 バッチ変更の実行

更新を必要とするレポートが大量にある場合は、rwconverter で CUSTOMIZE コマンドライン・キーワードを使用して、バッチ変更を実行できます。バッチ変更は、多数のレポートに同じ変更を繰り返し実行する必要がある場合に特に便利です (たとえば、フィールドの書式マスクの変更など)。Reports Builder でレポートを 1 つずつオープンして手動で変更するかわりに、rwconverter を 1 回実行するだけで、一度に多数のレポートに同じ変更を加えることができます。

次の例では、2 つの XML レポート定義 translate.xml および customize.xml を 3 つの RDF ファイル inven1.rdf、inven2.rdf および manu.rdf に適用し、変更後の定義を新しいファイル inven1_new.rdf、inven2_new.rdf および manu_new.rdf に保存します。

```
rwconverter username/password@my_db
STTYPE=rdffile SOURCE="(inven1.rdf, inven2.rdf, manu.rdf)"
DTTYPE=rdffile DEST="(inven1_new.rdf, inven2_new.rdf, manu_new.rdf)"
CUSTOMIZE="(d:\%apps%\trans\translate.xml,d:\%apps%\custom\customize.xml)"
BATCH=yes
```

注意: この例では、CUSTOMIZE の値に、Windows プラットフォームに格納されているファイルへのディレクトリ・パスが示されています。UNIX の場合は、そのプラットフォームの標準に従ってディレクトリ・パスを指定してください (つまり円記号 (¥) のかわりに通常のスラッシュを使用)。

16.5 XML レポート定義のデバッグ

次の機能は、XML レポート・ファイルをデバッグする際に役立ちます。

- XML パーサーのエラー・メッセージ
- トレース・オプション
- rwbuilder
- デバッグを目的とした、ファイルへの XML の書込み

16.5.1 XML パーサーのエラー・メッセージ

XML パーサーは、Oracle データベース・リリースに付属する Oracle XML Development Kit (XDK) に含まれています。XML パーサーは、XML 構文の妥当性をチェックする Java パッケージです。XML パーサーを含む JAR ファイルは、インストール時に自動的に設定され、Oracle Reports で使用可能となります。

XML パーサーは、ほとんどの構文エラーを検出し、エラー・メッセージを表示します。エラー・メッセージは、エラーが発生した XML ファイルの行番号と、エラーの簡単な説明から成ります。

XML パーサーの詳細は、Oracle Technology Network (<http://www.oracle.com/technology/index.html>) を参照してください。その場合、XML parser または XDK で検索します。Oracle データベース付属のドキュメントにも情報が記載されています。

16.5.2 トレース・オプション

XML レポート定義をテストする際に、トレース・ファイルを作成するオプションを追加してレポートを実行すると便利です。次に例を示します。

```
rwrun username/password@my_db REPORT=¥CORP¥MYREPORTS¥EMP.XML
TRACEFILE=emp.log TRACEMODE=trace_replace TRACEOPTS=trace_app
```

このコマンドラインの最後の 3 つのオプションによって、レポート処理の詳細リストを含むトレース・ファイルが生成されます。トレース・ファイル・ログのデフォルトの位置は、Windows と UNIX で同じです。

```
ORACLE_HOME¥reports¥logs¥
```

注意： この例では、REPORT オプションおよびトレース・ログへのパスに、Windows プラットフォームに格納されているファイルへのディレクトリ・パスが示されています。UNIX の場合は、そのプラットフォームの標準に従ってディレクトリ・パスを指定してください（つまり円記号 (¥) のかわりに通常のスラッシュを使用）。

16.5.3 rwbuilder

XML レポート定義の設計時には、Reports Builder でオープンしたほうが便利な場合があります。Reports Builder では、オブジェクトが期待どおりに作成または変更されているかをすぐに確認できます。たとえば、XML レポート定義にサマリーを作成する場合、Reports Builder で定義をオープンすると、サマリーがデータ・モデルの適切なグループに配置されているかが一目でわかります。

Reports Builder で完全なレポート定義をオープンするには、REPORT (または MODULE) キーワードを使用します。次に例を示します。

```
rwbuilder USERID=username/password@my_db REPORT=c:¥corp¥myreports¥emp.xml
```

Reports Builder で部分レポート定義をオープンするには、CUSTOMIZE キーワードを使用します。次に例を示します。

```
rwbuilder USERID=username/password@my_db REPORT=emp.rdf CUSTOMIZE=c:¥myreports¥emp.xml
```

注意： この例では、REPORT オプションによって、Windows プラットフォームに格納されているファイルへのディレクトリ・パスが指定されます。UNIX の場合は、そのプラットフォームの標準に従ってディレクトリ・パスを指定してください（つまり円記号 (¥) のかわりに通常のスラッシュを使用）。

どちらの場合も、Reports Builder で有効な XML レポート定義がオープンします。その後、Reports Builder の様々なビューを使用して、レポートが期待どおりに作成または変更されているかどうかを確認できます。

16.5.4 デバッグを目的とした、ファイルへの XML の書き込み

SRW.ADD_DEFINITION を使用して XML レポート定義をメモリー内に作成する場合は、XML をファイルに書き込むとデバッグに役立ちます。次の例は、各行をメモリー内のドキュメント・バッファおよび（ファイルへの書き込みを選択した場合は）指定したファイルに書き込むプロシージャを示しています。

```
PROCEDURE addaline (newline VARCHAR, outfile Text_IO.File_Type) IS
BEGIN
  SRW.ADD_DEFINITION(newline);
  IF :WRITE_TO_FILE='Yes' THEN
    Text_IO.Put_Line(outfile, newline);
  END IF;
END;
```

この例が実際に機能するには、プロシージャをコールする PL/SQL で TEXT_IO.File_Type タイプの変数を宣言する必要があります。次に例を示します。

```
custom_summary Text_IO.File_Type;
```

また、書き込み用のファイルをオープンし、addaline プロシージャをコールして、書き込む文字列と書き込み先ファイルを渡す必要があります。次に例を示します。

```
custom_summary := Text_IO.Fopen(:file_directory || 'vid_summ_per.xml', 'w');
addaline('<report name="video_custom" author="Generated" DTDVersion="9.0.2.0.0">',
custom_summary);
```

イベント・ドリブンによる公開の使用方法

現代のビジネス・プロセスでは、様々なファンクションやプロシージャを背後で実行することによって暗黙的に実環境に自動化を取り入れるという要求が生じはじめています。背後で実行されるタスクには、発注が処理される際に自動的に印刷される請求書などの出力の自動生成、最新データで自動的に更新される Web サイト、トランザクション完了時の最新レポートの自動メール送信などが挙げられます。

イベントに応じた自動出力は、かつてはかなり複雑な作業でした。特に、Oracle Reports Developer のような対話型の RAD 開発ツールを使用して、可能なかぎり同じ結果を得ようとする場合は非常に困難でした。

自動出力の要求に対応するために、OracleAS Reports Services にはスケジューリング・メカニズムが用意されており、ユーザーの操作を追加することなく定期的にレポートを起動することができます。ただし、これでも解決されない要求が 1 つだけ残されています。それは、レコードの挿入や値の変更といったデータベース内のイベントに応じて、自動的にレポートを実行する機能です。

OracleAS Reports Services の Event-Driven Publishing API を使用すると、レコードの挿入や値の変更といったデータベース内のイベントに応じて自動的にレポートを実行できます。

Event-Driven Publishing API は PL/SQL API で、データベースから OracleAS Reports Services へのジョブの自動送信を可能にします。

この章では、Event-Driven Publishing API の概要とともに、その使用例を示します。この章には、次の項があります。

- [Event-Driven Publishing API](#)
- [Event-Driven Publishing API を使用するアプリケーションのデバッグ](#)
- [データベース・イベントからのレポートの起動](#)
- [Oracle Advanced Queuing との統合](#)

17.1 Event-Driven Publishing API

Event-Driven Publishing API は、データベース内のイベントに応じるプロシージャの開発に必要な、基本的なファンクションを提供する PL/SQL パッケージです。イベント・ドリブン・ジョブは、HTTP プロトコルを使用して送信されます。サーバーは、すべてのコールに一意の `job_ident` レコードを割り当てます。これは、ジョブのステータスを追跡する際に役立ちます。

17.1.1 API の要素

この API は、いくつかの重要な要素から構成されています。

- **SRW** パッケージには、ジョブの送信、ジョブ・ステータスのチェック、ジョブの取消しおよびパラメータ・リストの操作に関連するすべてのファンクションとプロシージャが含まれています。
- **SRW_ParamList** は、パラメータ・リストを定義します。パラメータ・リストは、ジョブを送信する際に値を渡すための主要な手段です。パラメータ・リストは、ジョブ送信ごとに必要となります。このリストには、複数の重要なパラメータが含まれています。
- **SRW_ParamList_Object** は、アドバンスド・キューイングなどの機能に必要です。これらの機能では、パラメータ・リストをデータベースに格納して、メッセージとともに渡せるようにする必要があります。

API の各要素については、この後の項で詳しく説明します。

この API は、OracleAS Reports Services Security および OracleAS Portal とともにインストールされますが、どちらも必須ではありません。OracleAS Portal を使用しないデータベースにも API をインストールできるように、インストール・スクリプトが別途用意されています。

- `srwAPIins.sql` は、Event-Driven Publishing API をインストールします。
- `srwAPIgrant.sql` は、API へのアクセス権限を付与します。このスクリプトは、API へのアクセス権限を付与するユーザーごとに実行します。全員がアクセス可能である場合は、このスクリプトを 1 回実行し、PUBLIC にアクセス権限を付与できます。
- `srwAPIdrop.sql` は、API を削除します。

17.1.2 パラメータ・リストの作成と操作

パラメータ・リストは、`SRW_PARAMLIST` タイプの PL/SQL 変数です。このタイプの変数は、`SRW_PARAMETER` タイプの 255 個の要素からなる配列です。`SRW_PARAMETER` タイプの要素自体は、2 つの属性 `NAME` および `VALUE` から構成されます。API には、パラメータ・リストを操作するための次のプロシージャが用意されています。

- [Add_Parameter](#)
- [Remove_Parameter](#)
- [Clear_Parameter_List](#)

17.1.2.1 Add_Parameter

パラメータ・リストを初めて使用するときは、そのパラメータ・リストを初期化してからパラメータを追加する必要があります。次に例を示します。

```
DECLARE
myPlist SRW_PARAMLIST;
BEGIN
myPlist := SRW_PARAMLIST(SRW_PARAMETER('',''));
srw.add_parameter(myPlist, 'myParameter', 'myValue');
END;
```

パラメータの属性 (`NAME` と `VALUE`) はどちらも `VARCHAR2` タイプで、その長さは `NAME` が 80 文字、`VALUE` が 255 文字に制限されています。

ADD_PARAMETER ファンクションには、オプションで MODE という 4 番目の属性を指定できます。MODE は、同じ名前のパラメータがすでに存在する場合に、そのパラメータを上書きするか、またはエラーが発生するかを決定します。重複する名前があった場合にエラーが発生するように指定するには、定数 CHECK_FOR_EXISTANCE を使用します。これは、MODE 属性のデフォルト値です。重複する名前があった場合にパラメータを上書きするように指定するには、定数 OVERWRITE_IF_EXISTS を使用します。

17.1.2.2 Remove_Parameter

REMOVE_PARAMETER は、パラメータ・リストからパラメータを削除するために使用します。このプロシージャを呼び出し、対象となるパラメータ・リストとともに削除するパラメータの名前を渡します。

次に例を示します。

```
DECLARE
myPlist SRW_PARAMLIST;
BEGIN
myPlist := SRW_PARAMLIST(SRW_PARAMETER('',''));
srw.add_parameter(myPlist,'myParameter','myValue');
srw.remove_parameter(myPlist,'myParameter');
END;
```

17.1.2.3 Clear_Parameter_List

リストからすべてのパラメータを削除するには、CLEAR_PARAMETER_LIST を使用します。次に例を示します。

```
DECLARE
myPlist SRW_PARAMLIST;
BEGIN
myPlist := SRW_PARAMLIST(SRW_PARAMETER('',''));
srw.add_parameter(myPlist,'myParameter','myValue');
srw.clear_parameter_list(myPlist);
END;
```

これによって、リストからすべてのパラメータが削除されます。

17.1.3 パラメータの名前と値での非 ASCII 文字の使用

Event-Driven Publishing API の使用時に、ユーザー・パラメータの名前と値に非 ASCII 文字を使用するには、パラメータ DEFAULTCHARSET を、有効なキャラクタ・セット名に値を設定して、パラメータ・リストに追加する必要があります。このキャラクタ・セット名は、データベースの NLS_CHARACTERSET (JA16SJIS など)、または IANA 定義キャラクタ・セット名 (WINDOWS-31J など) に指定できます。また、DEFAULTCHARSET パラメータの値と、`rwServlet.properties` ファイルに指定されている DEFAULTCHARSET を一致させる必要があります (第 3.4.6 項「Reports パラメータのデコードに使用する `rwServlet` キャラクタ・エンコードの指定」を参照)。OracleAS Reports Services では、ユーザー・パラメータの非 ASCII の名前と値のエンコードに、DEFAULTCHARSET で指定されたキャラクタ・セットを使用するため、パラメータの名前と値に非 ASCII 文字を使用したレポートに Event-Driven Publishing API を使用することができます。

注意： パラメータ・リストにパラメータ DEFAULTCHARSET を追加しないと、OracleAS Reports Services では、ユーザー・パラメータの名前と値のエンコードに、データベースの NLS_CHARACTERSET が使用されます。

17.1.4 ジョブの送信

パラメータ・リストには、ジョブの送信に不可欠なすべてのパラメータが含まれている必要があります。ジョブ・タイプによって、リストに必要なパラメータが決まり、Reports Server でリクエストを処理できるようになります。

リスト内のパラメータは、ブラウザから Reports Servlet にジョブを送信する際に指定するパラメータと同じです。この場合、ジョブがレポートであるときは、少なくとも次のパラメータが必要です（それ以上指定してもかまいません）。

- GATEWAY は、リクエストの処理に使用する Reports Servlet への URL を提供します。
- SERVER は、サーブレットとともに使用される Reports Server を特定します。
- REPORT は、実行するレポート・ファイルを特定します。
- USERID は、レポートを実行するユーザーの名前と ID を特定します。
- AUTHID は、セキュリティで保護されたサーバーにアクセスする場合の認証情報を提供します。

各リクエストからは、`job_ident` レコードが返されます。このレコードには、ジョブを一意に識別するための情報が含まれています。この情報は、`SRW.JOB_IDENT` タイプの変数に格納されます。この情報は `PACKAGE-TYPE` であり、`SRW.JOB_IDENT` として参照する必要がありますことに注意してください。一方、パラメータ・リストは `OBJECT-TYPE` であり、`SRW_PARAMLIST` として参照する必要があります。

次に例を示します。

```
DECLARE
myPlist SRW_PARAMLIST;
myIdent SRW.Job_Ident;
BEGIN
myPlist := SRW_PARAMLIST(SRW_PARAMETER('',''));
srw.add_parameter(myPlist,'GATEWAY','http://...');
srw.add_parameter(myPlist,'SERVER','mySVR');
srw.add_parameter(myPlist,'REPORT','myReport.RDF');
srw.add_parameter(myPlist,'USERID','me/secret');
myIdent := srw.run_report(myPlist);
END;
```

API の `RUN_REPORT` メソッドは、必要不可欠なすべての情報を含んだパラメータ・リストを入力として (`ADD_PARAMETER` を使用)、リクエストを作成して送信するとともに、`job_ident` レコードを返します。

返された `job_ident` レコード (`MyIdent`) には、次のパラメータが含まれます。

- `MyIdent.GatewayURL`
- `MyIdent.ServerName`
- `MyIdent.JobID`
- `MyIdent.AuthID`

これらのパラメータは、`SRW.REPORT_STATUS` ファンクションが、送信されたジョブのステータス情報を取得するために必要となります。

17.1.5 ステータスの確認

Event-Driven Publishing API は、Reports Server との双方向通信を実現します。ユーザーは、サーバーにジョブを送信するとともに、`SRW.REPORT_STATUS` ファンクションを使用して、このジョブのステータスをサーバーに問い合わせることができます。

このファンクションからは、`SRW.STATUS_RECORD` タイプのレコードが返されます。このレコードには、`rwervlet` の Web コマンド `showjobs` を実行したときにジョブ・ステータス表示に示される情報と同じ情報が含まれています。

次に例を示します。

```
DECLARE
myPlist SRW_PARAMLIST;
myIdent SRW.Job_Ident;
myStatus SRW.Status_Record;
BEGIN
myPlist := SRW_PARAMLIST(SRW_PARAMETER('', ''));
srw.add_parameter(myPlist, 'GATEWAY', 'http://...');
srw.add_parameter(myPlist, 'SERVER', 'mySVR');
srw.add_parameter(myPlist, 'REPORT', 'MyReport.RDF');
srw.add_parameter(myPlist, 'USERID', 'me/secret');
myIdent := srw.run_report(myPlist);
myStatus := srw.report_status(myIdent);
END;
```

返されたステータス・レコードを使用して、ジョブのステータスに関する情報をフェッチできます。

17.1.6 サーバーのステータス・レコードの使用法

ステータス・レコードには、ジョブに関する処理情報が含まれています。これには、サーバー・キュー内の情報 (showjobs で表示) と同じ情報が含まれます。さらに、終了したジョブに対して生成されたファイルに関する情報、およびスケジュールされたジョブのシステムに関する情報も含まれます。

ステータス・レコードで最も重要な情報は、ランタイム・エラーとその原因をチェックするために使用される、現行ジョブ・ステータスとステータス・テキストです。

タイミング情報を使用して、完了までの予測時間を超過した場合にジョブが取り消されるかどうかを確認できます。

ステータス・レコードの用途の 1 つに、ジョブの取消しがあります。Event-Driven Publishing API では、サーバーに送信されたジョブを取り消す方法を提供しています。この方法は、許可されている実行時間を超過したジョブを削除する場合、または単にスケジュールされているジョブを取り消す場合に便利です。

ジョブを取り消すには、次のプロシージャを使用します。

```
DECLARE
myPlist SRW_PARAMLIST;
myIdent SRW.JOB_IDENT;
myStatus SRW.STATUS_RECORD;
BEGIN
myPlist := SRW_PARAMLIST(SRW_PARAMETER('', ''));
SRW.ADD_PARAMETER(myPlist, 'GATEWAY', 'http://...');
SRW.ADD_PARAMETER(myPlist, 'SERVER', 'mySVR');
SRW.ADD_PARAMETER(myPlist, 'REPORT', 'myReport.RDF');
SRW.ADD_PARAMETER(myPlist, 'USERID', 'me/secret');
myIdent := SRW.RUN_REPORT(myPlist);
myStatus := SRW.REPORT_STATUS(myIdent);
if myStatus.StatusCode != srw.RUNNING then
SRW.CANCEL_REPORT(myIdent);
END;
```

この例からわかるように、レポートを取り消す場合は、CANCEL_REPORT プロシージャ (SRW.CANCEL_REPORT) をコールし、取り消すジョブの job_ident レコードをそのプロシージャに渡します。このプロシージャでは、オプションのパラメータ・リストによって、必要な追加のパラメータを渡すことができます。

17.2 Event-Driven Publishing API を使用するアプリケーションのデバッグ

これまでに説明した処理はすべて背後で実行されるため、通常の実行ではデバッグ情報を生成できません。このため API には、特別なデバッグ・モードに切り替えるための 2 種類のプロシージャが用意されており、DBMS_OUTPUT によって拡張デバッグ情報が生成されます。

- SRW.START_DEBUGGING
- SRW.STOP_DEBUGGING

デバッグ・モードに切り替えるには、SRW.START_DEBUGGING をコールします。デバッグ・モードをオフにするには、SRW.STOP_DEBUGGING をコールします。デバッグ・モードは、実際のロジックを実行する直前に開始する必要があります。このモードは、パッケージの現行インスタンスがロードされているかぎり持続します。

この情報を表示する 1 つの方法として、スクリプトを実行する前に、SQL*Plus で SERVEROUT を ON に設定します。

このデバッグ方法に加えて、API にはエラー処理に使用する一連の事前定義の例外が用意されています。OracleAS Reports Services のインストールに含まれる srw_test.sql スクリプトには、これらの例外の例が示されています。

17.3 データベース・イベントからのレポートの起動

データベース・トリガーは、Event-Driven Publishing API でレポートを起動するための主要なメカニズムです。Oracle データベースでは、各種イベントに応じて起動される様々な範囲のトリガーを定義できます。データベース・ドリブン・ジョブを送信するには、これまでの項で説明したコードを、データベース・トリガー内で使用します。

イベント・ドリブンの公開には、多数の用途があります。その 1 つとして、権限が付与されたとき、またはユーザーがログイン/ログオフしたときに起動されるトリガーを使用して、セキュリティ・プロトコルを作成できます。また、あるタイプの変更が表内のデータに加えられた場合に実行される自動処理を作成することもできます。たとえば、経費報告書のステータスが DONE (完了) に変わったときにデータベース・トリガーを起動し、それによって従業員の管理者にそのレポートを自動的に送信できます。

次に例を示します。

```
CREATE TRIGGER EXP_REP_TRG
AFTER INSERT OR UPDATE on EXP_REP FOR EACH ROW
myPlist SRW_PARAMLIST;
myIdent SRW_JOB_IDENT;
BEGIN
IF (:new.ExpStat = 'DONE') THEN
myPlist := SRW_PARAMLIST(SRW_PARAMETER('',''));
SRW.ADD_PARAMETER(myPlist,'GATEWAY','http://...');
SRW.ADD_PARAMETER(myPlist,'SERVER','fooSVR');
SRW.ADD_PARAMETER(myPlist,'REPORT','foo.RDF');
SRW.ADD_PARAMETER(myPlist,'USERID','foo/bar');
SRW.ADD_PARAMETER(myPlist,'ExpenseID',:new.ExpID);
myIdent := SRW.RUN_REPORT(myPlist);
END IF;
END;
```

このトリガーは、EXP_REP 表が更新されるたびに起動されます。ステータスが DONE に変わると、レポート・リクエストが実行されます。

cgicmd.dat ファイルに指定されているキーを使用してリクエストを実行する場合は、REPORT パラメータのかわりに CMDKEY パラメータを指定します。このキーにユーザー ID 情報が含まれている場合は、USERID パラメータも省略できます。次に例を示します。

```
CREATE TRIGGER EXP_REP_TRG
AFTER INSERT OR UPDATE on EXP_REP FOR EACH ROW
myPlist SRW_PARAMLIST;
myIdent SRW.JOB_IDENT;
BEGIN
IF (:new.ExpStat = 'DONE') THEN
myPlist := SRW_PARAMLIST(SRW_PARAMETER('', ''));
SRW.ADD_PARAMETER(myPlist, 'GATEWAY', 'http://...');
SRW.ADD_PARAMETER(myPlist, 'SERVER', 'fooSVR');
SRW.ADD_PARAMETER(myPlist, 'CMDKEY', 'keyvalue');
SRW.ADD_PARAMETER(myPlist, 'ExpenseID', :new.ExpID);
myIdent := SRW.RUN_REPORT(myPlist);
END IF;
END;
```

さらに、配布 XML ファイルを通じて拡張配布モデルを定義した場合は、DESTINATION パラメータにそのファイルを指定できます。次に例を示します。

```
CREATE TRIGGER EXP_REP_TRG
AFTER INSERT OR UPDATE on EXP_REP FOR EACH ROW
myPlist SRW_PARAMLIST;
myIdent SRW.JOB_IDENT;
BEGIN
IF (:new.ExpStat = 'DONE') THEN
myPlist := SRW_PARAMLIST(SRW_PARAMETER('', ''));
SRW.ADD_PARAMETER(myPlist, 'GATEWAY', 'http://...');
SRW.ADD_PARAMETER(myPlist, 'SERVER', 'fooSVR');
SRW.ADD_PARAMETER(myPlist, 'REPORT', 'foo.RDF');
SRW.ADD_PARAMETER(myPlist, 'USERID', 'foo/bar');
SRW.ADD_PARAMETER(myPlist, 'DISTRIBUTE', 'YES');
SRW.ADD_PARAMETER(myPlist, 'DESTINATION', 'filename.xml');
SRW.ADD_PARAMETER(myPlist, 'ExpenseID', :new.ExpID);
myIdent := SRW.RUN_REPORT(myPlist);
END IF;
END;
```

これは、この種のロジックをアプリケーションからデータベースに移し、データベースをビジネス・プロセスの中心的なストレージとして使用するための 1 つの方法です。

注意： OracleAS Reports Services のインストールに含まれている `srw_test.sql` デモ・スクリプトには、Event-Driven Publishing API のその他の使用例が示されています。

17.4 Oracle Advanced Queuing との統合

Oracle Advanced Queuing は、キューと 2 つのプロセスを中心として非同期のリクエスト / レスポンス・メカニズムを構築するための手段となります。2 つのプロセスとは、MESSAGES をキューに入れる ENQUEUE、およびキューを読み取る DEQUEUE です。

アドバンスド・キューイングは、キュー間でメッセージを配布し、キューをサブスクライブするための高度なメカニズムを備えています。これらのメカニズムは、いずれも基本要素 (ENQUEUE、DEQUEUE および MESSAGES) をベースに構築されています。

Event-Driven Publishing API では、これらのキューを使用してレポート・ジョブを保存および送信することができます。OracleAS Reports Services で提供するメカニズムが要件に合わない場合は、独自のキューイング・メカニズムを作成することも可能です。

17.4.1 SRW_PARAMLIST タイプのメッセージを保持するキューの作成

キューは、いくつかの管理列とメッセージを表すオブジェクト列で構成されるデータベース内の表です。ここでのメッセージは、パラメータ・リストです。

アドバンスト・キューイング付属の `dbms_AQadm` パッケージには、アドバンスト・キューイング・システムの設定に必要なすべての管理ファンクションが含まれています。

データベースに物理的な表を作成するには、`dbms_AQadm.Create_Queue_Table` を使用します。その場合、表の名前、およびそのキューのメッセージを定義するオブジェクト・タイプの名前を渡す必要があります。

次に例を示します。

```
...
execute dbms_AQadm.Create_Queue_Table
(queue_Table=>'queuename._tab',
queue_Payload_Type=>'SRW_PARAMLIST_OBJECT',
compatible=>'9.0');
```

前述の例で、`SRW_PARAMLIST` タイプをカプセル化するオブジェクト・タイプ `SRW_PARAMLIST_OBJECT` をすでに作成しているため、これをメッセージとして使用できます。

キュー表を作成したら、`dbms_AQadm.Create_Queue` でキューを作成し、`dbms_AQadm.Start_Queue` でキューを開始する必要があります。

次に例を示します。

```
...
execute dbms_AQadm.Create_Queue
(Queue_Name=>'queuename',Queue_Table=>'queuename._tab');
prompt ... starting queue
execute dbms_AQadm.Start_Queue
(Queue_Name=>'queuename');
```

注意： OracleAS Reports Services のインストールに含まれている `srwAQsetup.sql` デモ・ファイルには、単純なキューの設定、作成、開始の完全な例が示されています。

キューを作成して開始した後に必要となるのは、このキューにメッセージを作成するプロシージャと、キューを読み取ってサーバーにジョブを送信するプロシージャです。次の各項では、これらのプロシージャを説明します。

17.4.2 エンキュー・プロシージャの作成

エンキュー・プロシージャは、キューにメッセージを挿入します。このプロシージャは、データベース・トリガーによってコールされるアプリケーション、または外部のメカニズムを通じて提供されるアプリケーションに組み込むことができます。この項では、このキューに単純なメッセージを挿入するストアド・プロシージャの作成例を示します。

ここでは、メッセージがパラメータ・リストそのものであるため、プロシージャはかなり簡単です。パラメータ・リストを作成するには、前述の項で使用したコードを使用します。使用した変数に加えて、キューに入れるメッセージを保持する `object` 変数を定義します。

```
...
  plist_object SRW_ParamList_Object;
...
```

パラメータ・リストを作成したら、オブジェクト・コンストラクタで実際のメッセージ・オブジェクトを作成します。

```
...
plist_object := SRW_PARAMLIST_OBJECT(plist);
...
```

その後、アドバンスド・キューイング付属のエンキュー・プロシージャでメッセージをエンキューします。

```
...
dbms_aq.enqueue(queue_name => 'myQueue',
enqueue_options => enqueue_options,
message_properties => message_properties,
payload => PList_Object,
msgid => message_handle);
...
```

メッセージがキューに挿入されます。メッセージ配布は一切設定していないため、次の項で説明するデキュー・プロシージャによってフェッチされるまで、メッセージはキューに入ったままとなります。

注意: dbms_aq.enqueue の正しい構文は、Advanced Queuing API のリファレンス・ドキュメントで確認してください。

OracleAS Reports Services のインストールに含まれている srwAQsetup.sql ファイルには、その他の例が示されています。

17.4.3 デキュー・プロシージャの作成

デキュー・プロシージャは、キュー内のすべてのメッセージを読み取って処理します。ここでは、メッセージを読み取り、メッセージに添付されたパラメータ・リストを使用してサーバーにジョブを送信します。

この処理には、次の例を使用します。

```
BEGIN
dequeue_options.wait := 1;
loop
DBMS_AQ.DEQUEUE(queue_name => 'myQueue',
dequeue_options => dequeue_options,
message_properties => message_properties,
payload => PList_Object,
msgid => message_handle);
COMMIT;
plist := plist_object.params;
r_jid := SRW.RUN_REPORT(plist);
end loop;
exception when aq_timeout then
begin
NULL;
end;
END;
```

このサンプル・コードでは、すべてのメッセージが処理されるまでキューが読み取られます。処理に使用できる時間は、コードの 2 行目に定義されているタイムアウトによって決まります。このタイムアウトによって、タイムアウト例外が作成されるまでデキュー・プロシージャがメッセージを待機できる秒数が定義されます。

アドバンスド・キューイングには、メッセージを読み取るための DBMS_AQ.DEQUEUE ビルトインが用意されています。このビルトインは、情報を保持するオブジェクトであるメッセージのペイロードを、ペイロード・パラメータで定義されたオブジェクトに挿入します。

`plist` を使用して、ペイロード・オブジェクトから情報を抽出します。すでに説明したように、ここではオブジェクトにパラメータ・リストが含まれています。パラメータ・リストは、オブジェクト内部の `PARAMS` 属性に格納されています。抽出されたパラメータ・リストは、ジョブを送信する `SRW.RUN_REPORT` に渡されます。

このデキュー・プロシージャを手動で起動しないようにするには、データベース内部でジョブとして実行します。

第 III 部

グローバル化・サポート および双方向サポート

第 III 部では、Reports 関連のグローバル化・サポート設定および双方向サポートに関する情報を記載します。第 III 部には、次の章があります。

- [第 18 章「グローバル化と双方向のサポートの実装」](#)

グローバリゼーションと双方向のサポートの実装

各国に配布するレポートを設計する場合は、キャラクタ・セットやテキストの読取り順序などを考慮する必要があります。OracleAS Reports Services では、これらの考慮事項に関連する問題に対処するためのサポートを提供しています。これらのサポートは、キャラクタ・セットに対するグローバリゼーション・サポートと、テキストの読取り順序に対する双方向サポートです。

グローバリゼーション・サポートは、複数の異なる言語で展開できるアプリケーションの設計を可能にします。Oracle では、ヨーロッパ、中東およびアジアで使われている言語のほとんどをサポートしています。グローバリゼーション・サポートを使用すると、次のことが可能になります。

- 各国のキャラクタ・セット（マルチバイト・キャラクタ・セットを含む）の使用
- 適切な言語と地域の表記規則に基づいたデータの表示
- インタフェースに表示される文字列の抽出とそれらの翻訳

双方向サポートは、利用対象者の要件に応じて、左から右または右から左の方向にデータを表示できるようにします。

この章では、グローバリゼーション・サポート・アーキテクチャを、Reports に関連するグローバリゼーション・サポートの設定も含めて概説するとともに、JSP でキャラクタ・セットを指定する方法を説明します。また、双方向、Unicode および翻訳のサポートに関する情報も提供します。この章の主な項は、次のとおりです。

- [グローバリゼーション・サポート・アーキテクチャ](#)
- [グローバリゼーション・サポートの環境変数](#)
- [JSP または XML ファイルでのキャラクタ・セットの指定](#)
- [双方向サポート](#)
- [Unicode](#)
- [アプリケーションの翻訳](#)
- [グローバリゼーションに関する問題のトラブルシューティング](#)

18.1 グローバリゼーション・サポート・アーキテクチャ

グローバル化・サポート・アーキテクチャは、次の2つの部分で構成されます。

- 言語に依存しない機能
- 言語依存データ

18.1.1 言語に依存しない機能

言語に依存しない機能では、実行時にオペレータが使用する言語および地域に応じて、適切にデータが処理されます。データは、各国の日付および時刻の表記規則に従って、自動的に書式設定されます。

18.1.2 言語依存データ

言語依存データを使用すると、データを分離できます。これによって、作成したアプリケーションで、そのアプリケーション固有の文字列の翻訳を処理できるようになります。

言語依存データはコードから分離されているため、グローバル化・サポート機能の動作は実行時に提供されるデータによって制御されます。新規言語を追加するとき、および言語別のアプリケーションの特性を変更するときに、コードを変更する必要はありません。このアーキテクチャでは、セッションごとに言語依存機能を指定することもできます。

18.2 グローバリゼーション・サポートの環境変数

グローバル化・サポートの環境変数は、Oracle Application Server をインストールする際に、自動的にデフォルト値に設定されます。

注意： 環境切替え機能を使用すると、インストール時に設定されるデフォルトの環境に限定されることがありません。また、1つの Reports Server に対して、言語設定など複数の環境を構成できます。詳細は、[第3.2.2項「動的な環境切替え」](#)を参照してください。

[表 18-1](#) では、特に OracleAS Reports Services に関するグローバル化・サポート関連の環境変数を説明します。

注意： グローバリゼーション・サポートのすべての環境変数の詳細は、Oracle Technology Network (<http://www.oracle.com/technology/index.html>) の『Oracle Application Server グローバリゼーション・サポート・ガイド』を参照してください。

表 18-1 OracleAS Reports Services のグローバル化・サポートの環境変数

変数	説明
NLS_LANG	OracleAS Reports Services に関連し、OracleAS Reports Services で使用される言語設定。
DEVELOPER_NLS_LANG	Reports Builder で使用される言語。設定されない場合は、NLS_LANG のデフォルト値が使用されます。
USER_NLS_LANG	Reports Runtime で使用される言語。設定されない場合は、NLS_LANG のデフォルト値が使用されます。

18.2.1 NLS_LANG 環境変数

NLS_LANG 環境変数は、OracleAS Reports Services で使用される言語、地域およびキャラクタ・セットの設定を指定します。具体的な設定項目は次のとおりです。

- ユーザーに表示されるメッセージの言語
- DATE および NUMBER データ型に使用されるデフォルトの書式マスク
- ソート基準
- キャラクタ・セット

注意： この環境変数は、Oracle Application Server をインストールする際に自動的に設定されます。Oracle Application Server のインストール後に環境変数を変更する方法の詳細は、「[NLS_LANG 環境変数の定義](#)」を参照してください。

NLS_LANG の構文は、次のとおりです。

```
NLS_LANG=language_territory.charset
```

各値は次のように定義されます。

- language
メッセージ（エラー・メッセージを含む）および月と曜日の名前を表示する言語とその表記規則を指定します。言語を指定しない場合は、その値がデフォルトで米語に設定されます。
- territory
地域を指定し、デフォルトの日付書式、数値に使用される小数点記号、通貨記号、および週番号と曜日番号の計算に使用される表記規則も指定します。地域を指定しない場合は、その値がデフォルトでアメリカに設定されます。
- charset
データを表示するキャラクタ・セットを指定します。これは、使用する言語とプラットフォームに適合するキャラクタ・セットにする必要があります。このオプションは、メッセージの表示に使用するキャラクタ・セットも指定します。

注意： OracleAS Portal Security、Portal Destination、Job Status Repository などの機能を使用すると、OracleAS Reports Services によって確立された JDBC データベース接続が NLS_LANG の初期設定よりも優先されることがあります。これにより、PDF 出力のアジア言語でのエイリアシングなどの、実行中のレポートの動作に影響が及ぶ場合があります。第 3.2.2 項「[動的な環境切替え](#)」で説明しているように、UNIX プラットフォームでは、環境切替え機能を使用してレポートの環境を動的に設定することで、この問題を回避できます。

NLS_LANG の language、territory、character に一般に使用される値の詳細は、『Oracle Application Server グローバリゼーション・サポート・ガイド』を参照してください。

NLS_LANG の設定では、（基本的に）同じ言語を使用する各国間の地域的な違いを考慮する必要があります。たとえば、フランスで使用されているフランス語で実行する場合は、NLS_LANG 環境変数を次のように設定します。

```
NLS_LANG=FRENCH_FRANCE.WE8ISO8859P1
```

一方、スイスで使用されているフランス語で実行する場合は、NLS_LANG 環境変数を次のように設定します。

```
NLS_LANG=FRENCH_SWITZERLAND.WE8ISO8859P1
```

注意： showjobs、showenv などの rwservlet ページ、オンライン・ヘルプおよびエラー・メッセージの言語は、NLS_LANG ではなく、中間層マシンのロケール (UNIX の場合は LANG) によって決まります。たとえば、中間層のロケールがフランス語に、NLS_LANG が NLS_LANG=JAPANESE_JAPAN.JA16SJIS に設定されている場合、showjobs またはエラー・メッセージは、日本語ではなくフランス語で表示されます。

18.2.1.1 NLS_LANG 環境変数の定義

Windows または UNIX のオペレーティング・システムでは、NLS_LANG 環境変数を他の環境変数と同じ方法で定義します。

18.2.1.1.1 Windows Windows で NLS_LANG 環境変数を定義する手順は、次のとおりです。

1. Windows のレジストリを開きます。

注意： レジストリは、バックアップを作成してから編集してください。

2. 「HKEY_LOCAL_MACHINE」ノードを拡張し、次に「SOFTWARE」ノードを拡張します。
3. 「Oracle」ノードを拡張し、次に OracleAS Reports Services の「HOME」ノードをクリックして、レジストリ・エディタの右側のパネルに Oracle 環境変数を表示します。
4. NLS_LANG 環境変数をダブルクリックします。
5. 「値」データ・テキスト・ボックスに NLS_LANG の新しい値を入力します。
6. 「OK」をクリックします。

18.2.1.1.2 UNIX UNIX プラットフォームで NLS_LANG 環境変数を定義するには、ORACLE_HOME/bin ディレクトリにあるシェル・スクリプト reports.sh でその環境変数を設定します。

18.2.1.2 キャラクタ・セット

グローバリゼーション・サポート環境変数のキャラクタ・セット・コンポーネントでは、各自の環境でデータを表す際に使用されるキャラクタ・セットを指定します。あるキャラクタ・セットを使用するシステムから別のキャラクタ・セットを使用するシステムにデータが転送されると、転送先のシステムでそのデータが処理され、正しく表示されます。ただし、一部の文字がキャラクタ・セットの別のバイナリ値で表される場合があります。

18.2.1.2.1 キャラクタ・セットの設計上の考慮事項 多言語対応アプリケーション、または単一言語対応であっても複数のキャラクタ・セットを使用するアプリケーションを設計する場合は、実行時に最も広く使用されるキャラクタ・セットを決定し、NLS_LANG 環境変数をそのキャラクタ・セットに設定する必要があります。

あるキャラクタ・セットで設計されたアプリケーションを別のキャラクタ・セットで実行すると、パフォーマンスに問題が生じる可能性があります。さらに、設計時のキャラクタ・セットの文字の一部が実行時のキャラクタ・セットに含まれていない場合は、認識されない文字の部分に疑問符が表示されます。

18.2.1.2.2 フォント・エイリアシングの考慮事項 特定のフォントを使用して多言語アプリケーションを作成しても、実行してみると別のフォントが使用されている場合があります。この問題は、多くの場合、西ヨーロッパ圏以外の言語環境で英語フォント（MS Sans Serif や Arial など）を使用すると発生します。この問題が起こる理由は、フォントに関連付けられているキャラクタ・セットと言語環境変数によって指定されたキャラクタ・セットが一致しているかを、OracleAS Reports Services がチェックするためです。2つのキャラクタ・セットが一致しない場合は、OracleAS Reports Services によって、言語環境変数で指定されたキャラクタ・セットが関連付けられている別のフォントに自動的に置換されます。この自動置換によって、データベースから返されるデータがアプリケーション内で正しく表示されます。

注意： 英語フォントを使用して各国語の文字を入力すると、Windows では暗黙的に別のフォントへの関連付けが行われます。

ただし、このような自動置換が行われないようにする場合もあります。自動置換が行われないようにするには、希望するすべてのフォントを、フォント・エイリアス・ファイル (uifont.ali) 内の WE8ISO8859P1 キャラクタ・セットにマッピングします。たとえば、アプリケーションで Arial フォントを使用できないときは、フォント・エイリアス・ファイルに次の行を追加します（このファイルは `ORACLE_HOME\TOOLS\COMMON` にあります）。

```
ARIAL.....=ARIAL.....WE8ISO8859P1
```

この例では、すべての Arial フォントを、WE8ISO8859P1 キャラクタ・セットでの同じ値にマップするように指定しています。

フォント・エイリアシングおよび uifont.ali の詳細は、第 6.1.2.1 項「フォント・エイリアシング」を参照してください。

18.2.1.3 言語と地域

キャラクタ・セットによって各言語に必要な個別の文字が使用可能になると同時に、各国の表記規則がサポートされることで、データ項目が正しくローカライズされて表示されます。

指定の言語では、次の要素についてデフォルトの表記規則が決まります。

- OracleAS Reports Services のメッセージおよび Oracle Database からのメッセージの言語。
- 月と曜日の名前およびそれらの省略形（SQL ファンクションの TO_CHAR および TO_DATE で指定）の言語。
- AM、PM、AD および BC を表す記号
- ORDER BY を指定したときの文字データのデフォルト・ソート基準（ORDER BY が指定されないかぎり、GROUP BY はバイナリ・ソートを使用）。
- 表記の方向（右から左と左から右）

たとえば、言語をフランス語に設定すると、英語による次のメッセージがフランス語に変換されます。

English:
ORA-00942: table or view does not exist
REP-0110: Unable to open file.

French:
ORA-0092: table ou vue inexistante
REP-0110: Impossible d'ouvrir le fichier

指定された地域によって、次のような日付および数値のデフォルトの書式特性に関する表記規則が決まります。

- 日付書式
- 小数点記号および桁グループ・セパレータ
- 各国通貨記号

- ISO 通貨記号
- 週を開始する曜日
- 貸方と借方の記号
- ISO 週フラグ
- リスト・セパレータ

たとえば、地域をフランスに設定すると、カンマを小数点記号として数値が書式設定されます。

18.2.2 DEVELOPER_NLS_LANG 環境変数と USER_NLS_LANG 環境変数

2 か国語のリソース・ファイルとメッセージ・ファイルを同時に使用する必要がある場合は、次の 2 つの言語環境変数を使用します。これらの環境変数は、Oracle Application Server のインストールが完了した後に使用可能となります。

- DEVELOPER_NLS_LANG
- USER_NLS_LANG

DEVELOPER_NLS_LANG と USER_NLS_LANG の構文は、NLS_LANG 環境変数と同じです。したがって次の構文を使用します。

```
DEVELOPER_NLS_LANG=language_territory.charset
USER_NLS_LANG=language_territory.charset
```

これらの環境変数が設定されない場合は、NLS_LANG のデフォルト値が使用されます。環境変数の DEVELOPER_NLS_LANG および USER_NLS_LANG は、次のような状況のときに、NLS_LANG 環境変数のかわりに使用します。

- Reports Builder を特定の言語（英語など）で使用する一方で、別の言語のアプリケーションを開発する場合。DEVELOPER_NLS_LANG および USER_NLS_LANG 環境変数によって、Reports Builder と Reports Runtime で異なる言語設定を使用できます。
- Reports Builder のローカル言語バージョンが用意されていない言語で動作するアプリケーションを作成している場合。

18.3 JSP または XML ファイルでのキャラクタ・セットの指定

JSP レポートの Web レイアウトを実行している場合は、次の構文を使用して JSP ファイルにキャラクタ・セットを追加しなければならないことがあります（次の例では日本語のキャラクタ・セットを指定）。

```
<%@ page contentType="text/html;charset=Shift_JIS" %>
<META http-equiv="Content-Type" content="text/html;charset=Shift_JIS">
```

注意： XML の生成に使用する予定のキャラクタ・セットをペーパー・レイアウトのレポートに設定するには、レポートの XML Prolog Value プロパティに次のキャラクタ・セットを記述する必要があります。

```
<?xml version="1.0" encoding="&Encoding" ?>
&Encoding は、実行時に適切な設定で置き換えられます。
```

詳細は、『Oracle Application Server グローバリゼーション・サポート・ガイド』を参照してください。

このキャラクタ・セットを表す値によって、OracleAS Reports Services に指定されているキャラクタ・セットと互換性のあるキャラクタ・セットがコールされます。Web で使用されるキャラクタ・セット（IANA 定義のキャラクタ・セット）の値は、NLS_LANG 環境変数で表される値と異なります。表 18-2 では、charset パラメータで一般に使用される IANA 定義のキャラクタ・セットを示します。

注意： IANA charset の値には、大文字と小文字の区別がありません。大文字と小文字のどちらでも入力できます。

表 18-2 IANA 定義のキャラクタ・セットの有効値

言語	有効な IANA 定義のキャラクタ・セット
米語	ISO-8859-1、ISO-8859-15、windows-1252、US-ASCII、UTF-8
アラビア語	ISO-8859-6、windows-1256、UTF-8
アッサム語	UTF-8
ベンガル語 (バングラデシュ)	UTF-8
ベンガル語	UTF-8
ポルトガル語 (ブラジル)	ISO-8859-1、ISO-8859-15、windows-1252、UTF-8
ブルガリア語	ISO-8859-5、windows-1251、KOI8-R、UTF-8
フランス語 (カナダ)	ISO-8859-1、ISO-8859-15、windows-1252、UTF-8
カタロニア語	ISO-8859-1、ISO-8859-15、windows-1252、UTF-8
クロアチア語	ISO-8859-2、windows-1250、UTF-8
チェコ語	ISO-8859-2、windows-1250、UTF-8
デンマーク語	ISO-8859-1、ISO-8859-15、windows-1252、UTF-8
オランダ語	ISO-8859-1、ISO-8859-15、windows-1252、UTF-8
エジプト語	ISO-8859-6、windows-1256、UTF-8
英語	ISO-8859-1、ISO-8859-15、windows-1252、US-ASCII、UTF-8
エストニア語	ISO-8859-4、ISO-8859-13、windows-1257、UTF-8
フィンランド語	ISO-8859-1、ISO-8859-15、windows-1252、UTF-8
フランス語	ISO-8859-1、ISO-8859-15、windows-1252、UTF-8
ディンカ語 (ドイツ)	ISO-8859-1、ISO-8859-15、windows-1252、UTF-8
ドイツ語	ISO-8859-1、ISO-8859-15、windows-1252、UTF-8
ギリシア語	ISO-8859-7、windows-1253、UTF-8
グジャラート語	UTF-8
ヘブライ語	ISO-8859-8-I、windows-1255、UTF-8
ヒンディー語	UTF-8
ハンガリー語	ISO-8859-2、windows-1250、UTF-8
アイスランド語	ISO-8859-1、ISO-8859-15、windows-1252、UTF-8
インドネシア語	ISO-8859-1、ISO-8859-15、windows-1252、UTF-8
イタリア語	ISO-8859-1、ISO-8859-15、windows-1252、UTF-8
日本語	EUC-JP、Shift_JIS、UTF-8
カンナダ語	UTF-8
韓国語	EUC-KR、UTF-8
スペイン語 (ラテン・アメリカ)	ISO-8859-1、ISO-8859-15、windows-1252、UTF-8
ラトビア語	ISO-8859-4、ISO-8859-13、windows-1257、UTF-8

表 18-2 IANA 定義のキャラクタ・セットの有効値 (続き)

言語	有効な IANA 定義のキャラクタ・セット
リトアニア語	ISO-8859-4、ISO-8859-13、windows-1257、UTF-8
マライ語	ISO-8859-1、ISO-8859-15、windows-1252、UTF-8
マラヤーラム語	UTF-8
マラーティー語	UTF-8
スペイン語 (メキシコ)	ISO-8859-1、ISO-8859-15、windows-1252、UTF-8
ノルウェー語	ISO-8859-1、ISO-8859-15、windows-1252、UTF-8
オリヤー語	UTF-8
ポーランド語	ISO-8859-2、windows-1250、UTF8
ポルトガル語	ISO-8859-1、ISO-8859-15、windows-1252、UTF-8
パンジャブ語	UTF-8
ルーマニア語	ISO-8859-2、windows-1250、UTF-8
ロシア語	ISO-8859-5、windows-1251、KOI8-R、UTF-8
簡体字中国語	GBK、GB18030、UTF-8
スロバキア語	ISO-8859-2、windows-1250、UTF-8
スロベニア語	ISO-8859-2、windows-1250、UTF-8
スペイン語	ISO-8859-1、ISO-8859-15、windows-1252、UTF-8
スウェーデン語	ISO-8859-1、ISO-8859-15、windows-1252、UTF-8
タミール語	UTF-8
テルグ語	UTF-8
タイ語	TIS-620、UTF-8
繁体字中国語	Big5、Big5-HKSCS、UTF-8
トルコ語	ISO-8859-9、windows-1254、UTF-8
ウクライナ語	ISO-8859-5、windows-1251、KOI8-U、UTF-8
ヴェトナム語	windows-1258、UTF-8

18.4 双方向サポート

双方向サポートは、中東圏および北アフリカ圏の言語など、通常の文字表記方向が右から左の言語でアプリケーションを設計できるようにする機能です。双方向サポートによって、次の制御が可能になります。

- レイアウト方向。項目の右にラベルを配置し、チェック・ボックスおよびラジオ・ボタンを適切に配置して項目を表示することなど。
- 読む方向。テキストの方向 (右から左または左から右) を含みます。
- 整列。原点の左上から右上への切替えを含みます。

双方向のアプリケーションを設計する際に、NLS_LANG の設定を継承するかわりに、グローバル化・サポートの環境変数 DEVELOPER_NLS_LANG および USER_NLS_LANG を使用することができます。たとえば、Windows 環境で英語のインタフェースを使用しながらアラビア語のアプリケーションを開発する場合は、これらの環境変数を次のように設定します。

```
DEVELOPER_NLS_LANG=AMERICAN_AMERICA.AR8MSWIN1256
USER_NLS_LANG=ARABIC_UNITED_ARAB_EMIRATES.AR8MSWIN1256
```


この例では、`DEVELOPER_NLS_LANG` 環境変数がアラビア語のキャラクタ・セットを使用していることに注意してください。詳細は、[第 18.2 項「グローバル化・サポートの環境変数」](#)を参照してください。

Windows 版の Oracle Reports 10g リリース 2 (10.1.2) では、右から左に記述する言語（ヘブライ語やアラビア語など）に対応したフォント・サブセットを使用して PDF 出力を生成したときに、テキストを正しく右揃えする拡張機能が導入されました（[第 6.4 項「双方向 \(BiDi\) PDF ファイルの生成」](#)を参照）。

UNIX 版では、右から左に記述する言語の場合、PDF 出力の右揃えテキストにまだ不揃いが見られる場合があります。この問題を回避するには、可変幅フォントではなく、固定幅フォントを使用します。たとえば、`Miriam Fixed True Type` フォント（ヘブライ語）は Windows 2000 で使用可能な固定幅フォントであり、UNIX プラットフォームでフォント・サブセットに使用して、ヘブライ語のフォント位置合せに関する問題を修正できます。異なるプラットフォーム間でのフォントに関する問題の解決の詳細は、[第 7 章「クロス・プラットフォームの移植問題の解決」](#)を参照してください。

18.5 Unicode

Unicode は、1 つのアプリケーションで多言語テキストを表示できるようにするグローバルなキャラクタ・セットです。これによって多国籍企業では、1 つの多言語アプリケーションを開発し、それを世界各国に配布できます。

グローバルな市場では、次のようなキャラクタ・セットが必要です。

- 製品の 1 つの実装ですべての言語に対応でき、しかも世界各国で簡単に実装できる。
- 主要なスクリプトをすべて含む。
- 多言語ユーザーおよび多言語組織をサポートする。
- インターネットを通じた全世界的なデータ交換を可能にする。

この項では、Oracle Reports における Unicode の次の側面について説明します。

- [Unicode のサポート](#)
- [Unicode フォントのサポート](#)
- [Unicode サポートの有効化](#)

18.5.1 Unicode のサポート

OracleAS Reports Services では Unicode をサポートしています。UNIX プラットフォームでは、Unicode のサポートにはいくつかの制限があります。次に例を示します。

- UNIX では、PostScript 出力フォーマットの Unicode はサポートされていません。
- PDF や RTF などのビットマップ出力フォーマットでは、文字の不揃いなど、フォントに関する問題が UNIX で発生する場合があります。

このような問題を解決する方法の詳細は、[第 7.1.2 項「フォントに関連する問題の解決」](#)を参照してください。

Unicode を使用すると、西ヨーロッパ、東ヨーロッパ、中東（双方向表記）などで使用されているシングルバイト言語、および中国語、日本語、韓国語 (CJK) などのアジア系マルチバイト言語の両方を、同じアプリケーションで表示できるようになります。

すべての言語を網羅する単一のキャラクタ・セットを使用することにより、各種言語の様々なキャラクタ・セットを用意する必要がなくなります。たとえば、日本語などのマルチバイト言語を表示するには、`NLS_LANG` 環境変数を次のように設定する必要があります。

```
NLS_LANG=JAPANESE_JAPAN.JA16SJIS
```

ドイツ語などのシングルバイト言語を表示するには、NLS_LANG を次のように設定する必要があります。

```
NLS_LANG=GERMAN_GERMANY.WE8ISO8859P1
```

この方式の明らかなデメリットは、アプリケーションで一度に1つのキャラクタ・セットの文字しか表示できない点です。複数のキャラクタ・セットを組み合わせたデータは表示できません。

Unicode キャラクタ・セットを使用する場合は、NLS_LANG のキャラクタ・セット部分に、特定言語のキャラクタ・セットではなく UTF8 を設定します。これによって、異なる言語およびキャラクタ・セットの文字を同時に表示できます。たとえば、画面に日本語とドイツ語を同時に表示するには、*language_territory* を適切に設定して、NLS_LANG 環境変数のキャラクタ・セット部分を UTF8 に設定する必要があります。次に例を示します。

```
NLS_LANG=JAPANESE_JAPAN.UTF8
NLS_LANG=GERMAN_GERMANY.UTF8
NLS_LANG=AMERICAN_AMERICA.UTF8
```

Unicode 機能によって、アプリケーション開発者とエンド・ユーザーは、レポートに多言語テキストを表示できるようになります。これには、Unicode 文字を含むデータベースのテキスト、多言語ボイラープレート・テキスト、グラフィカル・ユーザー・インタフェース (GUI) オブジェクト内のテキスト、キーボード入力されたテキスト、およびクリップボードのテキストが含まれます。

注意： Web 用にアプリケーションを開発する場合は、Unicode を使用できます。これは、Java がブラウザを通じて Unicode をサポートしているためです。

18.5.2 Unicode フォントのサポート

特定の言語でテキストを入力するには、その言語をサポートするオペレーティング・システムのバージョンを実行している必要があります。また、第4章「Oracle Reports でのフォントの管理」で説明しているように、OracleAS Reports Services は出力フォーマットのタイプに応じ、各国語のフォントについてオペレーティング・システムに依存しています。

Windows には True Type Big Font が用意されています。これらのフォントには、複数の言語によるテキストの表示や印刷に必要な文字が含まれています。たとえば、西ヨーロッパ言語、中央ヨーロッパ言語およびアラビア語のテキストをフィールドに入力、表示または印刷したときに予期しない文字が出力された場合は、Big Font を使用していない可能性があります。

Microsoft Windows でシングルバイト言語に用意されている Big Font は、Arial、Courier New および Times New Roman です。詳細は、Microsoft 社の Web サイト (<http://www.microsoft.com/typography/fonts/default.aspx>) を参照してください。

西ヨーロッパ言語、中央ヨーロッパ言語、キリル語、ギリシャ語、トルコ語、ヘブライ語、アラビア語、バルト語、ヴェトナム語、タイ語、簡体字中国語、日本語、韓国語、繁体字中国語に対して、オラクル社は次の2種類の Unicode フォントを用意しています。

- Albany WT フォント (プロポーションアル幅) は、Oracle Application Server 10g リリース 2 (10.1.2) MRUA の CD にあります。
- Andale Duospace WT フォント (固定幅) は、Oracle Metalink (<http://metalink.oracle.com>) からダウンロードできます。ARU 番号は 2638552 です。

サードパーティの Unicode フォントも利用できます。

18.5.3 Unicode サポートの有効化

Unicode サポートを有効にするには、NLS_LANG 環境変数を次のように設定します。

```
NLS_LANG=language_territory.UTF8
```

環境変数の詳細は、第 18.2 項「グローバル化・サポートの環境変数」を参照してください。

18.6 アプリケーションの翻訳

OracleAS Reports Services のどのアプリケーションにも、次のような様々なメッセージが使用されています。

- データベースからのエラー・メッセージ
- OracleAS Reports Services で生成されるランタイム・エラー・メッセージ
- アプリケーションの一部として定義されるメッセージまたはボイラープレート・テキスト

このうち最初の 2 つについては、NLS_LANG 環境変数を正しく設定し、適切なメッセージ・ファイルを用意することで、メッセージが自動的に翻訳されます。アプリケーションの一部として定義されているメッセージおよびボイラープレート・テキストを翻訳するには、Oracle の翻訳ツールである TranslationHub を使用します。また、コードの文字列には PL/SQL ライブラリを使用すると便利です。

注意： TranslationHub の使用方法は、Oracle Developer Suite JP Documentation Library および Oracle Technology Network (<http://www.oracle.com/technology/index.html>) を参照してください。

PL/SQL ブロック内の定数テキストは、手動で翻訳する必要があります。これは、このテキストが明確に区切られておらず、多くの場合、変数や文字列から構成されているためです。このような文字列を翻訳するには、PL/SQL ライブラリを使用して柔軟なメッセージ構造を実装します。

ビルトイン・プロシージャ SRW.MESSAGE によって、またはトリガーやプロシージャから表示項目にメッセージを割り当てることによってプログラムから表示されるメッセージには、連結可能な PL/SQL ライブラリを使用して、柔軟なメッセージ機能を実装できます。ライブラリはホスト上に格納し、実行時に動的に連結できます。実行時には、検索パスに基づいて、連結されたライブラリを読み込むことができます。たとえば、ライブラリにイタリア語のメッセージのみが格納されているとします。

```
FUNCTION nls_appl_mesg(index_no NUMBER)
RETURN CHAR
IS
    msg CHAR(80);
BEGIN
    IF index_no = 1001 THEN
        msg := 'L'impiegato che Voi cercate non esiste...';
    ELSIF index_no = 1002 THEN
        msg := 'Lo stipendio non puo essere minore di zero.';
    ELSIF ...
        .
        .
    ELSE
        msg := 'ERRORE: Indice messaggio inesistente.';
    END IF;
    RETURN msg;
END;
```

このようなルーチンは、一般に文字式が使用可能な場所であればどこでも使用できます。たとえば、適切に翻訳されたアプリケーション・メッセージを使用して警告を表示するには、次のコードを組み込みます。

```
SRW.MESSAGE(1001,nls_appl_mesg(1001));
```

注意： SRW.MESSAGE ビルトイン・プロシージャを含んだ SRW ビルトイン・パッケージの説明については、Oracle Reports のオンライン・ヘルプを参照してください。

アプリケーションを別の言語に変更するには、nls_appl_mes フังก์ションを含む PL/SQL ライブラリを、翻訳済みテキストを指定した nls_appl_mesg フังก์ションを含む同名のライブラリに置き換えるだけです。

18.7 グローバリゼーションに関する問題のトラブルシューティング

多言語アプリケーションで発生する可能性のあるグローバリゼーションに関する問題の解決に役立つように、この項では次のトラブルシューティング情報を示します。

- [JSP ファイルでのキャラクタ・セットの動的な埋込み](#)
- [グローバリゼーション・サポートの環境変数の設定](#)
- [WE8ISO8859P15 キャラクタ・セット使用時に文字化けしたフォントの修正](#)
- [エンコードされた JSP レポートのオープンまたは実行](#)
- [ERR-063001 xxx.dtd null の解決](#)
- [HP-UX の日本語環境での Oracle Reports の実行](#)

JSP ファイルでのキャラクタ・セットの動的な埋込み

Oracle Reports では、Web レポートのテンプレートは、デフォルトで西ヨーロッパ言語のキャラクタ・エンコーディング用に構成されます。ただし、その他の言語については、<Meta> タグの charset 属性および <%@page%> ページ・ディレクティブの両方を使用して、あらゆる JSP ファイルにキャラクタ・エンコーディングを指定できます。

JSP ファイルに適切なキャラクタ・エンコーディングを動的に関連付けるには、次の変更を行います。

1. rw*.html ファイルと blank_template.jsp ファイルを編集します。
 - a. ページ・ディレクティブを次のように変更します。

```
<%@ page contentType="text/html;charset=yourIANAencoding" %>
```

ここでは

yourIANAencoding は IANA 定義のキャラクタ・セット名で、NLS_LANG 変数の NLS_CHARACTERSET 部分に対応しています。

- b. <Head> タグ内の <Meta> タグを次のように変更します。

```
<meta http-equiv="Content-Type"
content="text/html;charset=yourIANAencoding" />
```

注意： テンプレート・ファイルの rw*.html と blank_template.jsp は、ORACLE_HOME/reports/templates/ ディレクトリにあります。

2. (`ORACLE_HOME/reports/templates/`にある) `template.xml` ファイルを編集します。

a. `<xsl:output>` タグを次のように変更します。

```
<xsl:output
  method="jsp"
  indent="yes"
  encoding="yourIANAencoding"
/>
```

ここでは

`yourIANAencoding` は IANA 定義のキャラクタ・セット名で、`NLS_LANG` 変数の `NLS_CHARACTERSET` 部分に対応しています。

b. ファイルにページ・ディレクティブを追加します。

```
<%@ page contentType="text/html; charset=yourIANAencoding" %>
```

c. `<HEAD>` タグ内の `<META>` タグを次のように追加または変更します。

```
<meta http-equiv="Content-Type"
content="text/html; charset=yourIANAencoding" />
```

ここでは

`yourIANAencoding` は IANA 定義のキャラクタ・セット名で、`NLS_LANG` 変数の `NLS_CHARACTERSET` 部分に対応しています。

グローバルゼーション・サポートの環境変数の設定

`Tk2Motif.rgb` ファイルには、Motif 版の Oracle Toolkit のリソース設定が含まれます。たとえば、Oracle Reports で使用するキャラクタ・セット (`NLS_CHARACTERSET` で指定) と X フォント間のフォント・マッピングが指定されます。

Oracle Reports によって、このファイルが `ORACLE_HOME/guicommon/tk/admin/language` ディレクトリで検索されます。この `language` は、`NLS_LANG` の言語設定から導出されます。

このファイルが存在しない場合は、`ORACLE_HOME/guicommon/tk/admin` でデフォルトのバージョンが検索されます。このバージョンは、西ヨーロッパ言語のキャラクタ・セットである `WE8ISO8859P1` 用に構成されています。

`NLS_LANG` または `NLS_CHARACTERSET` で、`NLS_LANG` に設定している言語に通常使用されないキャラクタ・セットを指定すると、Oracle Reports によってエラーが生成されます。

たとえば、`NLS_LANG=AMERICAN_AMERICA.JA16EUC` を設定すると、`ORACLE_HOME/guicommon/tk/admin/` ディレクトリで `Tk2Motif.rgb` が検索されず、`NLS_LANG` に設定した言語は `AMERICAN` であり、`AMERICAN` には言語サブディレクトリが関連付けられていないため、Oracle Reports ではデフォルトのファイルが使用されます。このバージョンは `WE8ISO8859P1` 用に設計されており、使用する `NLS_LANG` キャラクタ・セットは `JA16EUC` であるため、Oracle Reports によってエラー `REP-3000` が生成されます。

この問題を回避するには、環境変数 `TK_UNKNOWN` の値を、使用するキャラクタ・セット固有の `Tk2Motif.rgb` ファイルの場所に設定します。

たとえば、`NLS_LANG=AMERICAN_AMERICA.JA16EUC` の場合は、`TK_UNKNOWN=ORACLE_HOME/guicommon/tk/admin/JA` を設定します。使用する言語が `AMERICAN` に設定されていても、Oracle Reports では、`JA` 言語サブディレクトリの `Tk2Motif.rgb` ファイルが使用されます。

WE8ISO8859P15 キャラクタ・セット使用時に文字化けしたフォントの修正

UNIX で `we8iso8859p15` キャラクタ・セットを使用してマルチバイト・レポートを実行すると、文字化けした出力が表示される場合があります。

この問題を回避する手順は、次のとおりです。

1. デフォルトのプリンタの `.ppd` ファイルに、使用しているフォントのエントリを作成します。このファイルは、次のディレクトリに格納されています。

```
$ORACLE_HOME/guicommon/tk/admin
```

注意： これらのエントリは、レポート出力で文字化けして表示されているフォントに対して作成します。

2. このフォントの AFM ファイルのコード体系が `AdobeStandardEncoding` の場合は、`FontSpecific` に設定します。したがって、AFM ファイルの次のエントリを変更します。

```
EncodingScheme AdobeStandardEncoding
```

これを、次のように変更する必要があります。

```
EncodingScheme FontSpecific
```

エンコードされた JSP レポートのオープンまたは実行

JSP レポートのキャラクタ・エンコーディング (EUC-JP など) が、`NLS_LANG` 環境変数のキャラクタ・セット部分 (`JA16SJIS` など) と異なる場合は、次のエラーが発生します。

- JSP ファイルを実行するとき：

```
REP-6106 or 6104 with javax.servlet.jsp.JspException (multibyte)
REP-0495 Unable to tokenize the query (singlebyte)
```

- Reports Builder を使用して JSP ファイルを開くとき：

```
REP-0069 Internal Error or REP-6106
```

この問題を回避するには、JSP レポートのキャラクタ・エンコーディングを、Oracle Reports の `NLS_LANG` 変数のキャラクタ・セット部分に対応する IANA 定義のキャラクタ・セットに一致させる必要があります。

次に例を示します。

JSP レポートのエンコーディング：

```
<% page contentType="text/html;charset=EUC-JP" %>
<META http-equiv="Content-Type" content="text/html;charset=EUC-JP">
```

この JSP ファイルを、キャラクタ・セット (EUC-JP) でエンコードする必要があります。

Oracle Reports のエンコーディング：

```
NLS_LANG=JAPANESE_JAPAN.JA16EUC
```

この例では、JSP レポートのエンコーディング (EUC-JP) が、Oracle Reports の `NLS_LANG` のキャラクタ・セット部分、つまり `JA16EUC` と一致しています。

ERR-063001 xxx.dtd null の解決

XML データ・ソースに対してレポートを作成するとき、XML ファイル（データ・ソース）と DTD の両方のエンコーディングを、Oracle Reports のエンコーディングに一致させる必要があります。

XML レポートを表（たとえば、日本語の表）に対して作成する場合、グループ要素名は表の言語（つまり日本語）になります。データ・ソースに一致させるには、DTD のグループ要素名を日本語に設定する必要があります。XML ファイルおよび DTD ファイルには、日本語をサポートするものであればどのようなエンコードでも実行できます（Shift_JIS、EUC-JP、UTF-8 など）。ただし、XML データ・ソースと DTD のエンコーディングが Oracle Reports と異なると、次のエラーが表示されます。

ERR-063001 xxx.dtd null

注意： XML スキーマを使用してルールを定義した場合、このエラーは表示されません。

この問題を回避するには、XML レポートのデータ・ソース XML ファイルと DTD ファイルの両方を、Reports Runtime の NLS_LANG のキャラクタ・セット部分にエンコードする必要があります。

たとえば、NLS_LANG=JAPANESE_JAPAN.JA16SJIS の場合、データ・ソース XML ファイルと DTD ファイルの両方を、Shift_JIS にエンコードする必要があります。

HP-UX の日本語環境での Oracle Reports の実行

HP-UX の日本語環境で、NLS_LANG=JAPANESE_JAPAN.JA16SJIS と設定して Oracle Reports を使用する場合、Oracle Reports を使用する前に、該当する Tk2Motif.rgb ファイルを変更する必要があります。これは、EUC にエンコードされた日本語リソースがこのファイルに含まれるためです。

Tk2Motif.rgb ファイルを Shift-JIS エンコーディングに変換するか、このファイルから最後の 7 つのエントリを削除します。そうしないと、Oracle Reports が失敗する可能性があります。

注意： 日本語環境に使用する Tk2Motif.rgb ファイルは、次のとおりです。

`$ORACLE_HOME/guicommon/tk/admin/JA/Tk2Motif.rgb`

第 IV 部

パフォーマンス

第 IV 部では、Oracle Application Server Reports Services 環境の管理、監視およびチューニングに関する情報を記載します。第 IV 部には、次の章があります。

- [第 19 章「OracleAS Reports Services の管理と監視」](#)
- [第 20 章「Oracle Reports のチューニング」](#)

OracleAS Reports Services の管理と監視

Oracle Application Server 付属の Oracle Enterprise Manager 10g は、OracleAS Reports Services に対して管理サービスと監視サービスを提供します。

- Oracle Enterprise Manager 10g Application Server Control は、Reports Server に対して管理機能とリアルタイムのパフォーマンス監視機能の両方を提供します。
- Oracle Enterprise Manager 10g Grid Control には、より多くの監視機能が含まれますが、管理機能は提供されません。またこれ以外に、傾向と分析の履歴データ収集、J2EE アプリケーションの診断、アプリケーションのエンドユーザー動作の指標などの管理機能が提供されます。

この章では、Oracle Enterprise Manager 10g に Reports Server を構成する方法と、Reports Server 関連ページにナビゲートし、Oracle Enterprise Manager 10g を使用して OracleAS Reports Services の管理と監視ができるようにする方法を示します。この章の主な項は、次のとおりです。

- [Oracle Enterprise Manager 10g 用の Reports Server の構成](#)
- [Reports Server ホーム・ページへのナビゲート](#)
- [Reports Server の管理と監視](#)

注意： Oracle Enterprise Manager 10g の Reports Server 関連ページには、状況依存のオンライン・ヘルプ・トピックが用意されており、各ページの項目に関する情報が提供されます。各ページで「ヘルプ」リンクをクリックすると、対応するヘルプ・トピックが表示されます。

関連項目： Oracle Enterprise Manager 10g の詳細は、Oracle Application Server JP Documentation Library の『Oracle Application Server 管理者ガイド』を参照してください。

19.1 Oracle Enterprise Manager 10g 用の Reports Server の構成

Reports Server は、Oracle Application Server をインストールする際、自動的に Oracle Process Manager and Notification Server (OPMN) で構成され、Oracle Enterprise Manager 10g に登録されます。Oracle Application Server をインストールした後で Reports Server を追加する場合は、新しいサーバーを次のファイルに登録する必要があります。

- Oracle Enterprise Manager 10g の `targets.xml` ファイル。`targets.xml` ファイルとその使用方法の詳細は、第 3.8 項「Oracle Reports と Oracle Workflow の通信の構成」を参照してください。
- OPMN の `opmn.xml` ファイル。`opmn.xml` ファイルとその使用方法の詳細は、第 3.7.1 項「`opmn.xml`」を参照してください。

注意： `addNewServerTarget.sh` (UNIX) または `addNewServerTarget.bat` (Windows) を使用して、Reports Server ターゲットを `targets.xml` および `opmn.xml` に追加します。次に、変更を有効にするために、OPMN を再起動します。

警告： Oracle Reports ネットワークまたは Oracle Reports ブリッジのポート番号を変更しても、これらのコンポーネントは Oracle Enterprise Manager 10g を通じて制御されているわけではないため、変更されたポート番号は Oracle Enterprise Manager 10g に反映されません。次に例を示します。

ネットワーク構成ファイル `rwnetwork.conf` を編集して、`multicast` 要素および `namingService` 要素の `port` 属性を更新してから、Reports Server を起動して Oracle Enterprise Manager 10g Application Server Control にログインした場合、「ポート」ページには元のポート番号がそのまま表示されます。

Oracle Reports ブリッジ構成ファイル `repbrg_your_bridge_name.conf` を編集して、`port` 属性を更新してから、ブリッジを起動して Oracle Enterprise Manager 10g Application Server Control にログインした場合、Reports Bridge ポートの「ポート」タブには元のポート番号がそのまま表示されます。

19.2 Reports Server ホーム・ページへのナビゲート

この項では、Oracle Enterprise Manager 10g で Reports Server ホーム・ページを表示する方法について説明します。

- [Application Server Control](#) での Reports Server ホーム・ページへのナビゲート
- [Grid Control](#) での Reports Server ホーム・ページへのナビゲート

注意： Grid Control で Reports Server を監視するには、Management Service をインストールおよび構成しておく必要があります。その方法の詳細は、『Oracle Enterprise Manager Grid Control インストレーションおよび基本構成』を参照してください。このマニュアルは、Oracle Application Server JP Documentation Library および Oracle Technology Network (<http://www.oracle.com/technology/index.html>) から入手できます。

19.2.1 Application Server Control での Reports Server ホーム・ページへのナビゲート

Reports Server ホーム・ページの表示は、様々な方法で行うことができます。次に、Application Server Control から Reports Server ホーム・ページを表示する手順について説明します。

1. Application Server Control を起動します。このコンソールを起動する方法の詳細は、Oracle Enterprise Manager 10g のドキュメントを参照してください。
2. Application Server Control は、[図 19-1](#) に示すように「ファーム」ページに起動されます。ここでは、使用可能な Oracle Application Server インスタンスが表示されます。

図 19-1 Application Server Control の「ファーム」ページ

クラスタ

(クラスタの作成(C))

選択 名前	ステータス	インスタンス
ファームにはクラスタがありません。		

スタンドアロン・インスタンス
これらのインスタンスはファームに属しますが、クラスタの一部ではありません。

(クラスタへの結合(J))

選択 名前	ホスト	Oracleホーム
<input checked="" type="radio"/> INSTANCEA.yourcompany.com	HOST01.yourcompany.com	/work1/oracle/product/AS
<input type="radio"/> INSTANCEB.yourcompany.com	HOST02.yourcompany.com	/work1/oracle/product/AS
<input type="radio"/> INSTANCEC.yourcompany.com	HOST03.yourcompany.com	/work1/oracle/product/AS

3. 情報を表示する Reports Server を含む Oracle Application Server インスタンスをクリックします。このリンクにより、[図 19-2](#) に示すような該当インスタンスのホーム・ページが表示されます。

図 19-2 Oracle Enterprise Manager 10g の Oracle Application Server ホーム・ページ

ORACLE Enterprise Manager 10g
Application Server Control

Application Server: oracles.abc.company.com

Home | J2EE Applications | Ports | Infrastructure | Backup/Recovery

Page Refreshed Mar 17, 2005 3:32:53 PM

General

Status: Up
Host: abc.company.com
Installation: Business Intelligence and Forms
Type: Forms
Oracle Home: C:\orahome
Farm: farm.company.com

Stop All | Restart All | Start All

CPU Usage

Application Server (8%)
Idle (83%)
Other (9%)

Memory Usage

Application Server (53% 536MB)
Free (7% 70MB)
Other (40% 410MB)

System Components

Start | Stop | Restart | Delete OC4J Instance

Select	Name	Status	Start Time	CPU Usage (%)	Memory Usage (MB)
<input type="checkbox"/>	Discoverer	Up	Mar 17, 2005 2:48:32 PM	0.00	10.19
<input checked="" type="checkbox"/>	Forms	Up	Mar 17, 2005 2:48:33 PM	0.00	0.00
<input type="checkbox"/>	home	Up	Mar 17, 2005 2:49:08 PM	0.00	28.36
<input type="checkbox"/>	HTTP_Server	Up	Mar 17, 2005 2:49:09 PM	0.10	70.97
<input type="checkbox"/>	OC4J_RI_Forms	Up	Mar 17, 2005 2:48:39 PM	0.07	75.27
<input type="checkbox"/>	OC4J_Portal	Up	Mar 17, 2005 2:49:08 PM	0.00	66.30
<input type="checkbox"/>	OC4J_Wireless	Up	Mar 17, 2005 2:49:24 PM	0.06	53.23
<input checked="" type="checkbox"/>	Portal.portal	Up	Mar 17, 2005 2:49:08 PM	0.00	66.30
<input type="checkbox"/>	Reports Server.rep_pahegde-pc_oracles2	Up	Mar 17, 2005 2:48:39 PM	0.07	75.27
<input type="checkbox"/>	Web_Cache	Up	Mar 17, 2005 2:49:09 PM	0.00	49.25
<input type="checkbox"/>	Wireless	Up	Mar 17, 2005 2:49:23 PM	0.17	64.53
<input checked="" type="checkbox"/>	Management	Up	Mar 17, 2005 12:36:51 PM	7.81	117.65

Start | Stop | Restart | Delete OC4J Instance

TIP This table contains only the enabled components of the application server. Only components that have the checkbox enabled can be started

Related Links

- Process Management
- All Metrics

Home | J2EE Applications | Ports | Infrastructure | Backup/Recovery

Copyright © 1996, 2005, Oracle. All rights reserved.
About Oracle Enterprise Manager 10g Application Server Control

- コンポーネントのリストから、情報を表示する Reports Server を選択します。このリンクにより、[図 19-3](#) のような Reports Server ホーム・ページが表示されます。

図 19-3 Oracle Enterprise Manager 10g の Reports Server ホーム・ページ

リフレッシュされたページ 2005/09/29 19:09:28

ホーム
エンジン

一般



現行ステータス **稼働中** [停止] [再起動]
 開始時間 **2005/09/12 18:18:17**
 バージョン **10.1.2.0.2**

レスポンスおよびロード

CPU使用率(%) **N/A**
 メモリー使用量(MB) **N/A**
 平均レスポンス時間(ミリ秒)

構成

Reports Server クラスタ **クラスタ化されていません**
 トレース・オプション
 トレース・モード
 最大キュー・サイズ **1000**

ジョブ

現行ジョブ 0
 失敗したジョブ 0
 終了したジョブ 0
 スケジュールされたジョブ 0
 転送されたジョブ数 0

依存コンポーネント

コンポーネント名	ステータス	開始時間
HTTP Server	↑	2005/09/12 18:16:29
Web Cache	↑	2005/09/12 18:16:29
OC4J BI Forms	↑	2005/09/12 18:16:29

管理

[構成](#) [構成ファイルの編集](#) [サーバー・トレース](#) [サーバー・ログ](#)

関連リンク

[すべてのメトリック](#)

ホーム
エンジン

[ログ](#) | [トポロジ](#) | [プリファレンス](#) | [ヘルプ](#)

Copyright (c) 1996, 2005, Oracle. All rights reserved.
 Oracle Enterprise Manager 10g Application Server Control バージョン情報

19.2.2 Grid Control での Reports Server ホーム・ページへのナビゲート

Reports Server ホーム・ページの表示は、様々な方法で行うことができます。次に、Grid Control から Reports Server ホーム・ページを表示する手順について説明します。

- Grid Control を起動します。このコンソールを起動する方法の詳細は、Oracle Enterprise Manager 10g のドキュメントを参照してください。このコンソールは、[図 19-4](#) のように表示されます。ホーム・ページには、企業全体の概略が表示されます。

図 19-4 Grid Control コンソール

The screenshot shows the Oracle Enterprise Manager 10g Grid Control interface. At the top, there are navigation tabs: ホーム (Home), ターゲット (Targets), デプロイ (Deploy), アラート (Alerts), ポリシー (Policies), ジョブ (Jobs), and レポート (Reports). The main content area is divided into several sections:

- 概要 (Summary):** A pie chart showing the status of all targets (すべてのターゲット). The counts are: 停止中 (123) - 35%, 不明 (66) - 19%, ブロックアウト中 (3) - 1%, 稼働中 (156) - 45%.
- ターゲット検索 (Target Search):** A search bar with the text "すべて" (All) and a search button.
- セキュリティポリシー違反 (Security Policy Violations):** A list of violations including Critical (297), Warning (159), and Information (7). It also shows 164 new violations in the last 24 hours.
- デプロイサマリ (Deployment Summary):** A table showing the installation status of Oracle Database 10g instances.
- リソースセンター (Resource Center):** Links to documentation, release notes, and support.

データベースのインストール	ターゲット	インストール	個別パッチ適用
Oracle Database 10g 10.1.0.4.0	1	3	3
Oracle Database 10g 10.1.0.4.2	3	3	3
Oracle Database 10g 10.2.0.1.0	7	3	3

- 「ターゲット」タブをクリックします。
- 「Application Server」サブ・タブをクリックします。これは、図 19-5 のように表示されます。

図 19-5 Grid Control コンソールの「ターゲット」の「Application Server」タブ

The screenshot shows the Oracle Enterprise Manager 10g Grid Control interface with the 'Application Server' sub-tab selected. The navigation tabs now include: ホーム (Home), ターゲット (Targets), デプロイ (Deploy), アラート (Alerts), ポリシー (Policies), ジョブ (Jobs), and レポート (Reports). The main content area shows:

- Application Server:** A search bar with the text "すべて" (All) and a search button.
- フラット・リストの表示 (Flat List View):** A table showing the status of Oracle Application Server instances.
- ヒント (Hint):** A note about using icons for more information.
- 関連リンク (Related Links):** A link to customize the table columns.

選択	Name	タイプ	ステータス	ステータスの詳細	アラート	ポリシー違反	CPU使用率 (%)	メモリー使用量 (MB)
▼	すべてのアプリケーション・サーバー							
▶	INSTANCEA.yourcompany.com	Oracle Application Server	(?)	24 (?:2 ⇕ 22)	1 0	0 0 0		
▶	INSTANCEB.yourcompany.com	Oracle Application Server	(?)	6 (⇕ 6)	0 0	0 0 0		
▶	INSTANCEC.yourcompany.com	Oracle Application Server	(?)	14 (⇕ 14)	1 0	0 0 0		

4. 情報を表示する Reports Server を含む Oracle Application Server インスタンスをクリックします。このページは、[図 19-6](#) のように表示されます。

図 19-6 Grid Control コンソールの Application Server ページ

The screenshot displays the Oracle Enterprise Manager 10g Grid Control interface for an Oracle Application Server instance. The main status area shows the server is '稼働中' (Running) with 100% availability. A line graph shows application URL response times in seconds, with a notable spike around 2005/09/28. Below the graph is a table of components, including Discoverer, Forms, HTTP Server, and Reports Server, all showing '稼働中' status. An alert section below the components indicates that no alerts were detected.

選択	名前	タイプ	現行のステータス
<input type="checkbox"/>	Discoverer	Discoverer	稼働中
<input type="checkbox"/>	Forms	Forms	稼働中
<input type="checkbox"/>	home	OC4J	稼働中
<input type="checkbox"/>	HTTP_Server	Oracle HTTP Server	稼働中
<input type="checkbox"/>	OC4J_BI_Forms	OC4J	稼働中
<input type="checkbox"/>	OC4J_Portal	OC4J	稼働中
<input type="checkbox"/>	OC4J_Wireless	OC4J	稼働中
<input type="checkbox"/>	Portal.portal	Portal	稼働中
<input type="checkbox"/>	Reports_Server_rep_oracleas2	Reports Server	稼働中
<input type="checkbox"/>	Web_Cache	Web Cache	稼働中
<input type="checkbox"/>	Wireless	Wireless	稼働中

5. Reports Server は、その Oracle Application Server インスタンスのコンポーネントのリストに表示されます。それをクリックします。Reports Server ホーム・ページが表示されます。これは、[図 19-3](#) のように表示されます。

19.3 Reports Server の管理と監視

Oracle Enterprise Manager 10g で Reports Server を管理および監視するには、オンライン・ヘルプで順を追って説明した手順を参照してください。

Oracle Enterprise Manager 10g でオンライン・ヘルプにアクセスする手順は、次のとおりです。

1. Oracle Enterprise Manager 10g の任意のページで「ヘルプ」リンクをクリックすると、オンライン・ヘルプが表示されます。
2. オンライン・ヘルプのウィンドウで、次の操作を行います。
 - Application Server Control を使用している場合は、**目次**のリンクをクリックします。
 - Grid Control を使用している場合は、**ヘルプの目次**のリンクをクリックします。

3. 目次の「OracleAS Reports Services の管理」をクリックすると、Reports Server に関連するオンライン・ヘルプのトピックが表示されます。

Application Server Control

「OracleAS Reports Services の管理に Oracle Enterprise Manager Application Server Control を使用」を開いて、次のトピックを表示します。

- Application Server Control コンソールを使用した Reports Services の監視について
- 新しい Reports Server の構成および Oracle Enterprise Manager への追加
- Reports Server の起動、停止および再起動
- ジョブ・キューの表示と管理
 - 現行ジョブ・キューの表示と管理
 - スケジュールされたジョブ・キューの表示と管理
 - 終了したジョブ・キューの表示と管理
 - 失敗したジョブ・キューの表示と管理
- Reports Server 構成ファイルの表示と変更
- サーバー・クラスタ・メンバーの表示とメンバーへのリンク
- ポート番号の表示
- 中間層の変更

Grid Control

「OracleAS Reports Services の管理に Oracle Enterprise Manager Grid Control を使用」を開いて、次のトピックを表示します。

- Grid Control コンソールを使用した OracleAS Reports Services の監視について
- Reports Server に関する一般的な情報の表示
- Reports Server のパフォーマンスの表示
- ジョブのパフォーマンスの表示

注意： Oracle Enterprise Manager 10g の Reports Server 関連ページには、タスクに関するオンライン・ヘルプ・トピックのほかに、状況依存のオンライン・ヘルプ・トピックが用意されており、各ページの項目に関する情報が提供されます。各ページで「ヘルプ」リンクをクリックすると、対応するヘルプ・トピックが表示されます。

Oracle Reports のチューニング

レポート・リクエストのサイズと複雑さが増加し、ユーザー・ベースが増えてくると、レポートのパフォーマンス（レポートの実行時間）の効率化を検討する必要があります。それによって、パフォーマンスが最大化されるとともに、レスポンス時間の最小化が実現します。レポートのパフォーマンスをチューニングする前に、次の要点を検討してください。

- 体感パフォーマンスと測定可能なパフォーマンスの両方を向上させることで生じるパフォーマンスとそのトレードオフ
- 必要となるコスト
- コンピューティング環境の複雑性

これらの考慮事項の中には、その調査結果がパフォーマンスの向上に大きく寄与するものがあります。一方、パフォーマンスの向上に少ししか寄与しないものや、実際のレポートのパフォーマンスには影響はないものの、体感パフォーマンスを向上させる可能性があるものもあります。体感パフォーマンスとは、（最終出力として測定される）最終結果に影響を与えるイベントのことをいいます。体感パフォーマンスの例は、第 20.6.1 項「事前フェッチ」を参照してください。

この章では、個々のレポートを構築、実装およびチューニングする際に、パフォーマンスの向上に役立つガイドラインと提案を多数紹介します。ここで紹介する提案は全般的なものであるため、これらの提案すべてがどんな場合にも当てはまるわけではありません。ただし、アプリケーション環境に応じてこれらの提案の一部またはすべてを実装することで、レポートの実行パフォーマンス（実際のパフォーマンスも体感パフォーマンスも含めて）を向上させることができます。

注意： この章では、Oracle Reports の配置またはスケーラビリティに関する問題は取り上げていません。詳細は、OTN (<http://www.oracle.com/technology/products/reports/index.html>) にあるホワイトペーパー『Oracle Application Server Reports Services Scalability』を参照してください。

この章は、次の点についてレポートをより幅広く検討するために役立ちます。

- アプリケーションの要件
- 基盤となるデータ・モデルの正確性
- レポートが実行される環境（クライアント / サーバー、Web、ファイアウォールの内側など）
- ユーザー操作の必要性の程度

レポートの状況が明確になったら、次の項目を最適化および最小化するチューニング・プロセスへと進みます。

- データソースへのコール
- レイアウトで要求される不要なフォーマットの量

これらの目的を達成するには、レポートの次の各側面にチューニングの焦点を当ててください。

- **実行時間**。レポートの実行時にどの部分で多くの時間が費やされているかを突き止めます。それが明確になったら、適切なパフォーマンス・ツールを使用して、問合せの評価、データベースの最適化の再確認、レポートで使用されるコードの効率性の検証を行います。
- **フォーマットとレイアウト**。レポート情報のフォーマットとレイアウトについて考察します。
- **ランタイム・パラメータ**。パフォーマンスとレポート配布が最大になるように、ランタイム・パラメータを設定します。レポートのパフォーマンスを最大化する配布方法の詳細は、[第 20.6.2 項「一括処理と配布」](#)を参照してください。

この章では、次の各項でこれらの側面について説明します。

- [パフォーマンス分析ツール](#)
- [Reports Server の構成のチューニング](#)
- [rwdiag を使用したブリッジおよびネットワークのタイムアウト設定](#)
- [データへのアクセス](#)
- [データのフォーマット](#)
- [レイアウトに関する一般的なガイドライン](#)
- [Forms からの Oracle Reports のコール](#)
- [レポートの実行](#)

20.1 パフォーマンス分析ツール

レポートのチューニングに向けた最初の手順は、レポートの実行時にどの部分で多くの時間が費やされているかを突き止めることです。データの取出し、取り出したデータのフォーマット、ランタイム・リソースや配布の待機などの部分で、多くの時間が費やされている場合もあります。レポートのレイアウトが最大限に効率化されチューニングされていたとしても、非効率的な SQL が原因でデータの取出しに多くの時間が費やされていれば、効果は限定されます。

この項では、レポートのパフォーマンスの監視に使用できるツールについて説明します。

- [Oracle Enterprise Manager](#)
- [レポート・トレース](#)
- [RW_SERVER_JOB_QUEUE 表](#)
- [SHOWJOBS コマンドライン・キーワード](#)
- [効率的な SQL](#)
- [PL/SQL](#)
- [Java スタアド・プロシージャ](#)
- [Java Importer](#)

20.1.1 Oracle Enterprise Manager

Oracle Enterprise Manager を使用した Reports Server の管理と監視の詳細は、[第 19 章「OracleAS Reports Services の管理と監視」](#)を参照してください。

20.1.2 レポート・トレース

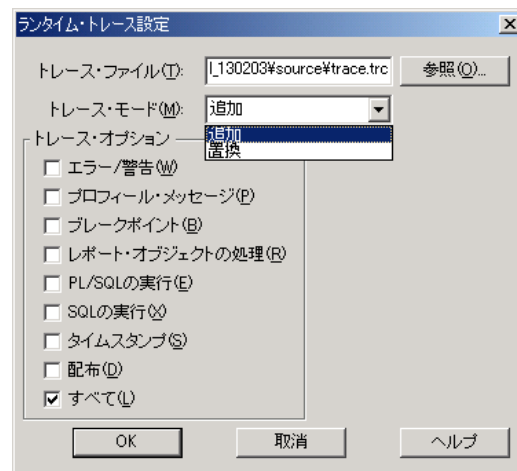
レポート・トレースを有効にすると、レポートの実行時に完了した一連の処理が記録されたテキスト・ファイルが生成されます。トレースでは、設定により、すべてのイベントを記録することも、特定のタイプのイベントのみを記録することもできます。たとえば、SQL の実行のみ、またはレイアウトとフォーマットのみなどに限定してトレースできます。トレース・ファイルに記録される情報は豊富で、パフォーマンス・チューニングだけでなく、レポートのデバッグやパフォーマンスのボトルネックの特定にも役立ちます。

レポート・トレース・ファイルの生成

トレースを有効にするには、次のいずれかを実行します。

- Reports Builder のユーザー・インタフェースで次の操作を行います。
 1. 「プログラム」 → 「トレース」 を選択します。
 2. 「トレース・モード」 を選択します。
 3. 適切な「トレース・オプション」 を選択します。この例では、トレース・ファイルに Reports Builder セッション全体のログ情報が記録されます。

図 20-1 Reports Builder の「ランタイム・トレース設定」ダイアログ・ボックス



- Reports Builder (rwbuilder) および Reports Runtime (rwrun) の場合は、rwbuilder.conf 構成ファイルで、または TRACEFILE、TRACEMODE および TRACEOPTS を使用してコマンドラインで、トレース・オプション (第 3.2.1.13 項「trace」を参照) を指定します。

注意： コマンドラインのトレース・オプションは、rwbuilder.conf ファイルに指定されたトレース・オプションより優先されます。

次に例を示します。

- rwbuilder.conf で、次のように指定します。

```
<trace traceFile="trace_file_name" traceOpts="trace_all"
traceMode="trace_replace"/>
```

- コマンドラインで、次のように指定します。

```
rwrun report=myreport.rdf server=myserver userid=user_id/password@mydatabase
destype=cache desformat=pdf TRACEFILE=trace_file_name TRACEOPTS=trace_all
TRACEMODE=trace_replace
```

注意： `rwbuilder` および `rwrun` のトレース・ファイルの位置は、Oracle Reports ログ・ディレクトリ (`ORACLE_HOME\reports\logs\rep_machinename-rwbuilder`) に対する相対パス、または絶対パス（フルパス名が指定されている場合）です。トレース・ファイルの名前を指定しない場合、デフォルトのトレース・ファイル名は `rwserver.trc` です。

- Reports Server (`rwserver`) では、`server_name.conf` 構成ファイルで、または **TRACEFILE**、**TRACEMODE** および **TRACEOPTS** を使用して Reports Server を起動した場合はコマンドラインで、トレース・オプション（第 3.2.1.13 項「**trace**」を参照）を指定します。Reports Server とエンジンに対して、別々にトレース・ファイルが生成されます。

注意： コマンドラインのトレース・オプションは、`server_name.conf` ファイルに指定されたトレース・オプションより優先されます。

次に例を示します。

- `server_name.conf` で、次のように指定します。

```
<trace traceFile="trace_file_name" traceOpts="trace_all" traceMode="trace_replace"/>
```

注意： トレース・ファイルの位置は、サーバーのログ・ディレクトリ (`ORACLE_HOME\reports\logs\server_name`) に対する相対パス、または絶対パス（フルパス名が指定されている場合）です。トレース・ファイルの名前を指定しない場合、デフォルトのサーバー・トレース・ファイル名は `rwserver.trc`、デフォルトのエンジン・トレース・ファイル名は `rwEng-x.trc` (`x` はエンジン ID) です。

Reports Server (`rwserver`) を通じて実行される個々のレポートでジョブ・トレースを有効にするには、構成ファイル (`server_name.conf` または `rwbuilder.conf`) で `traceOpts` を使用して、またはコマンドラインで **TRACEOPTS** を使用して、必要なトレースオプションを指定します。ジョブ・トレースは、`ORACLE_HOME\reports\logs\server_name\job_id\log.xml` に生成されます。このログ・ファイルの表示には、Oracle Enterprise Manager 10g を使用するのが最適です。詳細は、Oracle Enterprise Manager 10g Application Server Control のオンライン・ヘルプを参照してください。オンライン・ヘルプの目次で「**OracleAS Reports Services の管理**」の下のトピックを開くと、トピック「**失敗したジョブ・キューの表示と管理**」が表示されます。ここには「**失敗したジョブのトレース・ファイルの表示**」という項があります。

- Reports Servlet (`rwservlet`) では、第 3.4.5 項「**Reports Servlet と JSP のトレース・オプションの設定**」で説明しているように、サーブレット構成ファイル (`rwservlet.properties`) でトレース・オプションを指定します。
- Oracle Reports ブリッジのトレースでは、ブリッジ構成ファイル (`rebrg_bridgename.conf`) でトレース・オプション（第 3.3.2.6 項「**trace**」を参照）を指定します。

例

次のコマンドラインの例では、パフォーマンス・トレース情報を記録するトレース・ファイルが生成されます。以前に生成された既存のトレース・ファイルは、このファイルに置き換わります。

```
rwrun report=emp.rdf userid=scott/tiger@orcl destype=file desformat=pdf
desname=emp_pdf.pdf traceopts=trace_prf tracemode=trace_replace
tracefile=emp_tr.txt
```

次は、emp_tr.txt トレース・ファイルへの情報出力の概要です。

例 20-1 Reports Builder

```

+-----+
| Report Builder Profiler statistics |
+-----+
Total Elapsed Time: 8.00 seconds
Reports Time:      7.00 seconds (87.50% of TOTAL)
ORACLE Time:      1.00 seconds (12.50% of TOTAL)
UPI:              0.00 seconds
SQL:              1.00 seconds
TOTAL CPU Time used by process: N/A

```

表 20-1 Reports Builder

フィールド	説明
合計経過時間	レポートの実行にかかった時間。
Reports 時間	取り出したデータのフォーマットにかかった時間。「合計経過時間」に対する割合も表示されます。
Oracle 時間	データの取出しにかかった時間。「合計経過時間」に対する割合も表示されます。
UPI	SQL 問合せのみ。データベース接続の確立および SQL の解析と実行にかかった時間。
SQL	データベース・サーバーによるデータのフェッチにかかった時間 (SRW.DO_SQL 文、EXEC_SQL 文、PL/SQL カーソルなどの実行にかかった時間の割合)。

注意： データソースが非 SQL データソース (テキストや XML のプラグブル・データソースなど) の場合は、「Oracle 時間」、「UPI」および「SQL」の値は 0 になります。

例 20-1 では、チューニング作業を、データの間合せとフェッチではなく、フォーマット (Reports 時間) に絞り込めばよいことがわかります。

20.1.3 RW_SERVER_JOB_QUEUE 表

RW_SERVER_JOB_QUEUE 表では、Reports Server のジョブ・キューに、Enterprise Manager で使用できるウィンドウとは別のウィンドウが提供されます。

Reports Server では、ジョブ・リクエストが送信されるたびに、現在のレポートに関する情報がデータベースに送信されます。この情報は、次のデータを含む RW_SERVER_JOB_QUEUE 表に挿入されます。

- ジョブの名前
- ジョブの送信者
- 出力フォーマット
- ジョブの現行ステータス
- ジョブがキューに入れられた日時、ジョブ実行の開始日時および終了日時

RW_SERVER_JOB_QUEUE 表に含まれる情報は、表 20-2 で説明します。

表 20-2 RW_SERVER_JOB_QUEUE 表の構造

列名	説明
JOB_QUEUE	表示されたジョブの状態 (CURRENT、PAST、SCHEDULED のいずれか)。
JOB_ID	システム生成のジョブ ID 番号。
JOB_TYPE	report や rwurl などのジョブ・タイプ。Reports Server 構成ファイル server_name.conf で定義します。
JOB_NAME	ジョブ送信名 (JOBNAME に値が指定されていない場合はファイル名)。
STATUS_CODE	ジョブの現行ステータス。ステータス・コードの詳細は、表 20-3 を参照してください。
STATUS_MESSAGE	ステータス・コードに関連する完全なメッセージ・テキスト (レポートが終了した場合のエラー・メッセージも含む)。ステータス・コードの詳細は、表 20-3 を参照してください。
COMMAND_LINE	このジョブを送信するために送信される完全なコマンドライン。
OWNER	ジョブを送信したユーザー。Web では、デフォルトのユーザーは Web サーバーを所有する OS ユーザーです。
DESTYPE	レポート出力の送信先。
DESNAME	Reports Server のキャッシュに入れられない場合のレポート出力名。
SERVER	レポートの送信先の Reports Server。
QUEUED	ジョブ送信が所定の Reports Server に受信され、キューに入れられた日時。
STARTED	ジョブ送信が実行された日時。
FINISHED	送信されたジョブが完了した日時。
RUN_ELAPSE	開始時刻から終了時刻までの経過時間 (ミリ秒単位)。
TOTAL_ELAPSE	キューに入れられた時刻から終了時刻までの経過時間 (ミリ秒単位)。
LAST_RUN	スケジュールされたジョブが前回実行された日時。
NEXT_RUN	スケジュールされたジョブが次回実行される日時。
REPEAT_INTERVAL	ジョブの実行頻度。
REPEAT_PATTERN	繰り返しパターン (毎分、毎時、毎日など)。
CACHE_KEY	すでにキャッシュされている結果とリクエストの比較に使用するキャッシュ・キー。このキーは、ジョブが実行された時刻を考慮せずにレポート出力結果を一意に示す文字列です。たとえば、2つのリクエストのキーが同じ場合、これらのリクエストが同時に実行されると、(電子メールへの送信やファイルへの保存など) 異なる目的で出力が使用される場合であっても、同一の出力が生成されません。
CACHE_HIT	ジョブの結果が、ジョブ自体の実行ではなく、キャッシュからフェッチされたかどうかを示す。

表 20-3 ジョブ送信のステータス・コード

ステータス・コード	定義済の PL/SQL 定数	ステータス・コードの説明
0	UNKNOWN	不明なステータス。
1	ENQUEUED	ジョブがキューで待機中。
2	OPENING	サーバーがレポート定義を開いている。
3	RUNNING	レポートが現在実行中。
4	FINISHED	ジョブ送信が正常に完了。
5	TERMINATED_W_ERR	ジョブにエラーが発生して終了。
6	CRASHED	ジョブの実行中にエンジンがクラッシュ。
7	CANCELED	ジョブがユーザーのリクエストによって取り消された。
8	SERVER_SHUTDOWN	ジョブが Reports Server の停止によって取り消された。
9	WILL_RETRY	ジョブが失敗し、RETRY のためにキューに入れられた。
10	SENDING_OUTPUT	ジョブが完了し、出力を返している。
11	TRANSFERRED	ジョブがクラスタ内の別のサーバーに転送された。
12	VOID_FINISHED	ジョブは終了したが、キャッシュ容量の制限に達したため出力は無効。
13	ERROR_FINISHED	出力は正常に生成されたが、宛先への送信に失敗した。
14	DISTRIBUTE	レポート出力の配布中。

SELECT アクセス権が付与されているユーザーは、この表を表示できます。これによって、関心のあるジョブ送信を問い合せて、ジョブの現行ステータスを判別することができます。Reports Server の「キュー」画面の実装によって、このデータをユーザーに表示することもできます。このような画面を実装するには、この表に直接基づいたレポートを作成します。そうすることによって、ユーザーによるジョブ送信としてキューのレポートが表示されます。

逆に、ジョブ送信のステータスを反映してこの表をリアルタイムで更新すると、Reports Server でジョブを同時にリクエストしているユーザー数を、管理者が正確に把握しやすくなります。

RW_SERVER_JOB_QUEUE 表で、ジョブがキューに入れられたが完了していないことを示すステータス・コードを持つエントリの数を数えることによって、サーバー上で現在アクティブなユーザーの数を正確に返すことが可能になります。たとえば、次の問合せを使用できます。

```
SELECT Count (*)
FROM   RW_SERVER_JOB_QUEUE
WHERE  STATUS_CODE IN (1,      -- ENQUEUED
                      2,      -- OPENING
                      3)      -- RUNNING
AND    JOB_TYPE != 'Scheduled'
```

注意： 表には、レポートがキューに入れられた日時、実行された日時、および終了した日時が含まれますが、ジョブに QUEUED および STARTED の時刻は定義されているが FINISHED の値は定義されていない、という事実に基づいて問合せを使用することはお勧めできません。無効な入力などの予期しないエラーによってレポートが終了した場合、FINISHED 列は NULL のままです。ただし、STATUS_CODE と STATUS_MESSAGE の両方に、障害が生じたことが示され、障害の原因が表示されます。

20.1.3.1 キュー・アクティビティによるデータベースの更新

Reports Server のジョブ・キューは、PL/SQL API の使用を通じて実装されます。これには、リクエストが行われたときに、キュー情報を反映してキュー表を更新する機能があります。この実装は、次のパスで定義します。

```
ORACLE_HOME¥reports¥admin¥sql¥RW_SERVER.SQL
```

このスクリプトは、Oracle 10g データベースに対する動作が保証されています。

キューを実装するには、次の手順を実行します。

1. `rw_server.sql` ファイルをデータベースにロードします（このファイルは、OracleAS Reports Services インストールの `ORACLE_HOME¥reports¥admin¥sql` にあります）。
これによって、レポートのキュー情報を所有し、サーバーのキュー API に対して実行権限を持つスキーマが作成されます。Oracle6i Reports との下位互換性のために、`RW_SERVER_QUEUE` というビューも作成されます。
2. サーバー構成ファイル（`ORACLE_HOME¥reports¥conf¥server_name.conf`）の `jobStatusRepository` 要素の `repositoryconn` プロパティを、キュー・データを所有するスキーマの接続文字列に設定します。Reports Server の起動時に接続情報を暗号化する場合、`confidential` 属性を `yes` に設定します。詳細は、[第 3.2.1.12 項「jobStatusRepository」](#) を参照してください。

サーバーは起動時に、定義されたユーザーとして接続し、ジョブ送信をログに記録します。

20.1.4 SHOWJOBS コマンドライン・キーワード

コマンドラインで `showjobs` を使用して、`rwervlet` を通じて実行されているレポートを対象に、Reports Server のキュー・ステータスを Web 形式で表示できます。

詳細は、[第 A.3.99 項「SHOWJOBS」](#) を参照してください。

20.1.5 効率的な SQL

Oracle Reports では、データベースからのデータの取出しに SQL が使用されます。

注意： Oracle Reports は、非 PDS 問合せにのみ SQL を使用します。

非効率的な SQL は、特に大規模なレポートでパフォーマンスを悪化させます。したがって、Oracle Reports チューニングの全担当事者が、SQL に関する十分な実用知識を持ち、データベースで SQL 文がどのように実行されるかについて理解している必要があります。SQL について精通していない場合は、Reports Builder のデータ・ウィザードとクエリー・ビルダーを使用してください。ただし、ウィザードでは、SQL が使用可能な索引を使用しないなど、非効率的な SQL の作成を完全に回避することはできません。

レポートの SQL をチューニングするには、Oracle データベースで使用可能なトレース機能を使用します。SQL トレースでは、データベースに送信された SQL 文を特定できるだけでなく、解析、実行およびデータのフェッチにかかった時間も確認できます。トレース・ファイルの生成後、TKPROF データベース・ユーティリティを使用して EXPLAIN PLAN マップを生成します。EXPLAIN PLAN マップは、Oracle のオプティマイザにより使用された実行計画を図で表示するものです。たとえば、全表スキャンが使用された場所などが示されます。このマップにより、パフォーマンスに対する影響に応じて、その表に索引を作成する必要があるかがわかります。

Reports Builder 内で SQL トレースをオンにするには、次のコードで、`SQL_TRACE` という名前のレポート・レベルの式列を追加します。

```
SRW.DO_SQL('ALTER SESSION SET SQL_TRACE=TRUE');
return(1);
```

注意： SQL_TRACE のコールには、Before Report トリガーまたは Before Parameter Form トリガーを使用することもできます。

次の EXPLAIN PLAN マップは、データベースの SQL トレース機能を使用して生成されたものです。詳細は、『Oracle Database PL/SQL ユーザーズ・ガイドおよびリファレンス』を参照してください。

例

実行される文は次のとおりです。

```
SELECT e.ename, d.dname
FROM emp e, dept d
WHERE e.deptno(+) = d.deptno
```

生成された EXPLAIN PLAN は次のとおりです。

OPERATION	OPTIONS	OBJECT_NAME	POSITION

SELECT STATEMENT			
MERGE JOIN	OUTER		1
SORT	JOIN		1
TABLE ACCESS FULL		DEPT	1
SORT	JOIN		2
TABLE ACCESS FULL		EMP	1

Oracle Reports のデータをチューニングする際、Oracle RDBMS には、コストベース・オブティマイザとルールベース・オブティマイザという 2 種類のオブティマイザがあるということを理解しておく必要があります。デフォルトでは、コストベース・オブティマイザは、スループットの観点から（つまり、アクセスするすべての行を最小限のリソースで処理する）最適な実行計画を作成します。チューニング担当者は、オブティマイザのアプローチと目標を設定し、コストベースの最適化に必要な統計を収集することで、オブティマイザの選択に影響を与えることができます。コストベース・オブティマイザを使用することで、SQL のチューニングに関連する煩雑さはほとんどなくなります。一方、データの配布とオブティマイザ・ルールを理解することで、任意の方法を選択でき、実行計画をより強力に制御できます。たとえば、SQL 文では、次のいずれかを実行できます。

- オブティマイザに、レスポンス時間の最適化（つまり、アクセスする最初の行を最小限のリソースを使用して処理する）を目指すというヒントを指定します。
- 索引が必要ないことを指定します。

注意： 大規模な問合せの場合は、次のいずれかを必ず実行してください。

- コストベース・オブティマイザをアクティブにし、DBMS_STATS パッケージ、COMPUTER STATISTICS オプションまたは ANALYZE コマンドを使用して統計を収集します。
 - ルールベース・オブティマイザにより指定されたルールに従うすべての SQL を最適化します。
-
-

データベース・オブティマイザの機能の詳細は、Oracle Application Server のドキュメントを参照してください。

20.1.6 PL/SQL

ORA_PROF ビルトイン・パッケージを使用して、レポートの PL/SQL プログラム・ユニットをチューニングします。ORA_PROF ビルトイン・パッケージに含まれるプロシージャ、ファンクションおよび例外により、コードの実行にかかった時間を追跡できます。

例

```
PROCEDURE timed_proc (test VARCHAR2) IS
  i PLS_INTEGER;
BEGIN
  ORA_PROF.CREATE_TIMER('loop2');
  ORA_PROF.START_TIMER('loop2');
  ColorBand_Program_Unit;
  ORA_PROF.STOP_TIMER('loop2');
  TEXT_IO.PUTF('Loop executed in %s seconds.¥n',
  ORA_PROF.ELAPSED_TIME('loop2'));
  ORA_PROF.DESTROY_TIMER('loop2');
END;
```

このプロシージャでは、まずタイマーが作成され開始されます。次に、サブプログラムが実行され、それが終了するとタイマーが停止し、サブプログラムの実行にかかった時間が表示されます。完了すると、タイマーが破棄されます。

注意： ORA ビルトイン・パッケージの説明については、Oracle Reports のオンライン・ヘルプを参照してください。

大量のデータベース操作を実行する PL/SQL プログラム・ユニットは、データベース・ストアード・プロシージャとして実装するようにします。ストアード・プロシージャは、Oracle データベースで直接実行されるため、ローカルの PL/SQL プログラム・ユニットに比べて、より高速に操作を実行できます。ローカルの PL/SQL プログラム・ユニットでは、Reports Builder の PL/SQL パーサー、データベースの SQL パーサーが使用されます。さらに、ネットワーク・トリップも含まれます。

データベース操作を実行しない PL/SQL プログラム・ユニットは、オブジェクト・ナビゲータの「プログラム・ユニット」ノードを使用して、可能な限りローカルにコーディングしてください。PL/SQL プログラム・ユニットのローカル化は、外部 PL/SQL ライブラリから PL/SQL を実行する場合よりもパフォーマンス上のメリットがあります。外部 PL/SQL ライブラリは、コード共有のメリットを活用できる場合にのみ使用します。

SRW.DO_SQL ビルトイン・プロシージャの使用は、できるだけ控えめにしてください。SRW.DO_SQL ビルトイン・プロシージャへのコールでは、毎回、通常の間合せと同様に、コマンドの解析とバインドが実行され新しいカーソルがオープンします。ただしこの操作は、SRW.DO_SQL ビルトイン・プロシージャを所有するオブジェクトを起動するたびに発生する点で、通常の間合せとは異なります。

たとえば、式列内の PL/SQL ブロックが SRW.DO_SQL ビルトイン・プロシージャをコールし、データ・モデル・グループにより 100 レコードが返されたとします。この場合、解析、バインド、カーソル作成操作が 100 回発生します。したがって、SRW.DO_SQL ビルトイン・プロシージャは、通常の SQL では実行できない操作（一時表の作成やその他の DDL 操作の実行）、および実行回数が少ないところ（各レポートで 1 回だけ起動されるトリガー内など）で使用してください。

SRW.DO_SQL ビルトイン・プロシージャを使用する主な理由は、一時表の作成や削除などの DDL 操作の実行です。次の例では、SRW.DO_SQL ビルトイン・プロシージャを使用して表を作成します。表の名前は、ランタイム・パラメータ・フォームで入力されるパラメータによって決まります。

注意: SRW.DO_SQL ビルトイン・プロシージャを含んだ SRW ビルトイン・パッケージの説明については、Oracle Reports のオンライン・ヘルプを参照してください。

例

```
SRW.DO_SQL ('CREATE TABLE' || :tname || '(ACCOUNT NUMBER
NOT NULL PRIMARY KEY, COMP NUMBER (10,2))');
```

20.1.7 Java ストアド・プロシージャ

Java ストアド・プロシージャを使用すると、サーバー・レベルでビジネス・ロジックを実装できるため、アプリケーションのパフォーマンス、スケーラビリティおよびセキュリティが向上します。Oracle データベースでは、PL/SQL ストアド・プロシージャと Java ストアド・プロシージャをデータベースに格納できます。一般に、プロシージャの拡張を必要とする SQL プログラムは PL/SQL を使用し、Oracle データへの容易なアクセスを必要とする Java プログラムは Java を使用します。Java ストアド・プロシージャのほうがより高い柔軟性を持ちますが、ある程度のオーバーヘッドが生じます。各自のニーズに応じて、パフォーマンスと柔軟性のトレードオフのバランスをとってください。

Java ストアド・プロシージャの詳細は、『Oracle Database Java 開発者ガイド』を参照してください。

20.1.8 Java Importer

Oracle PL/SQL は強力で生産性の高い開発環境ですが、外部のアプリケーション・サービスやプロバイダとの統合が必要になる場合があります。こうした外部のアプリケーション・サービスやプロバイダの多くが、Java を使用した統合点を提供するケースが増えています。Oracle Reports は、Oracle Java Importer と統合することで、外部の中間層 Java クラスに含まれるビジネス・ロジックを容易に起動できるようになります。Java Importer では、選択した各 Java クラスの PL/SQL ラッパー・パッケージが宣言的に作成され、PL/SQL ファンクションおよびプロシージャ経由でそのクラスの方法が公開されます。これにより、レポートの実行時に、Java オブジェクトのインスタンス化、使用、破棄を行うことが可能になります。この強力な拡張により Java コードを自分で記述する必要はなくなりますが、ある程度のオーバーヘッドが生じます。指定した Java クラスごとに、別々の PL/SQL パッケージが生成されます。PL/SQL ジェネレータは、Java メソッドから PL/SQL パッケージを生成するときに型変換を行います。PL/SQL パッケージの新しいファンクションを使用して Java オブジェクト・インスタンスが作成され、Java Importer により生成されると、常にその結果が JOBJECT 型の変数に格納されます。グローバル参照を削除せずに次々に蓄積していくと、JVM のメモリー消費が増加するため、Java オブジェクトの永続化には注意が必要です。

20.2 Reports Server の構成のチューニング

この項では、Reports Server のパフォーマンスと安定性の向上に役立つヒントを示します。このヒントの対象となる処理は次のとおりです。

- 様々なクライアントからのレポート・リクエストの受入れ
- ジョブの実行スケジューリング
- Oracle Reports のエンジンの管理
- キャッシュの管理
- 様々な宛先の管理
- セキュリティのチェック
- ジョブ・ストア（永続ジョブ・データ）の管理

次に示すように、負荷の大きな操作では、Reports Server の様々なパラメータを最適な値にチューニングすることが重要です。

1. サーバー構成ファイルの engine 要素の initEngine、maxEngine、minEngine の各属性の最適値を決定します。

```
<engine id="rwEng" class="oracle.reports.engine.EngineImpl" initEngine="1"
maxEngine="2" minEngine="1" engLife="50"
maxIdle="30" callbackTimeOut="90000">
```

engine 要素の詳細は、[第 3.2.1.4 項「engine」](#) を参照してください。maxEngine の値によって、レポートの実行を求めるユーザー・リクエストに応じる準備ができていないプロセスの最大数が設定されます。設定した値が低すぎると、キューがユーザー・リクエストでいっぱいになり、マシンの使用可能容量が十分に利用されなくなります。設定した値が高すぎると、Reports Server への割当てマシン容量を超え、同じホストでの実行が必要な他のアクティビティから容量を奪ってしまうことになり、オペレーティング・システムでスラッシングが発生する可能性があります。

エンジン数の簡単な計算例として、実行に平均 10 秒かかるレポートのセットがあるとします。システムへの入力でリクエストされるレポート数は、毎分 6 ~ 12 件です。この例では、計算は次のようになります。

- $\text{initEngine} = (\text{レポートの実行にかかる平均時間}) \times (\text{レポート・リクエストの最小入力率}) = (10/60) \times 6 = 1$
- $\text{maxEngine} = (\text{レポートの実行にかかる平均時間}) \times (\text{レポート・リクエストの最大入力率}) = (10/60) \times 12 = 2$
- minEngine = 負荷の種類に応じ、0 から initEngine の範囲

これらの計算結果を使用して、サーバー構成ファイルに minEngine=1 および maxEngine=2 を指定できます。これにより、ジョブが受け取られたときは常に、アイドル・エンジンが即座に使用されます。

スケーラビリティとパフォーマンスのテストでは、2 ~ 4 のエンジン数に CPU の数を乗じるというガイドラインを使用して maxEngine が構成されている場合、スループットの最大値が表示されます。

URL エンジンを使用していない場合は、サーバー構成ファイルで engine 要素に ID="rwURLEng" というコメントを付けます。

2. cache 要素の cacheSize プロパティ、queue 要素の maxQueueSize 属性および EXPIRATION キーワードの最適値を決定します。

詳細は、[第 3.2.1.3 項「cache」](#)、[第 3.2.1.16 項「queue」](#) および [第 A.3.35 項「EXPIRATION」](#) を参照してください。cacheSize、maxQueueSize および EXPIRATION の値は互いに関連しており、Reports Server が効率的に動作するように慎重に設定する必要があります。

たとえば、EXPIRATION=480 と設定してレポートを実行する場合、これはジョブをキャッシュに 4 時間 (480 分) 入れておくという意味になります。この場合、4 時間ですべてのジョブに対応できるように、maxQueueSize を設定する必要があります。ここで、1 分間に 10 件のジョブが処理されるとすると、計算は次のようになります。

$$\text{maxQueueSize} = (\text{レポート・リクエストの入力率}) \times (\text{有効期間}) = 480 \times 10 = 4800$$

cacheSize も、4800 件のジョブに対応できるように、十分高い値に設定する必要があります。各レポートの平均サイズが 100K の場合、計算は次のようになります。

$$\text{cacheSize} = (\text{maxQueueSize}) \times (\text{レポートの平均サイズ}) = 4800 \times 100 / 1000 = 480\text{MB}$$

cache 要素の maxCacheFileNumber プロパティの値を計算する場合も、同様のロジックを使用できます。

注意： maxQueueSize の最小値には、1000 (デフォルト) をお勧めします。maxQueueSize または cacheSize の値がデフォルト値を大きく下回ると、Reports Server のパフォーマンスが低下する場合があります。

3. サーバー構成ファイルの engine 要素の engineResponseTimeout 属性を、次のように設定します。

```
<engine id="rwEng" class="oracle.reports.engine.EngineImpl" initEngine="1"
maxEngine="2" minEngine="1" engLife="50" maxIdle="30"
callbackTimeout="90000" engineResponseTimeout="5">
```

engine 要素の詳細は、[第 3.2.1.4 項「engine」](#) を参照してください。

エンジンが断続的にハングする場合は、engineResponseTimeout を設定します。この属性によって、Reports Server はハングするエンジンを検出し、クリーンアップを実行できるようになります。Reports Server でハングが早く検出されるほど、システムの安定性が増します。このため、engineResponseTimeout は、次のように慎重に設定する必要があります。

engineResponseTimeout の値を、実在するレポートのセットのうち 1 つのレポートの実行にかかる最大時間に設定してください。たとえば、実行に 10 秒から 5 分かかるレポートのセットがある場合、engineResponseTimeout="5" (5 分) と設定します。

注意： engineResponseTimeout に異なる値を指定して、常に別のサーバーでバッチ・レポートを実行することをお勧めします。対話型レポートとバッチ・レポートは、同じサーバーに送信しないでください。

4. サーバー構成ファイルの connection 要素の maxConnect 属性を、次のように設定します。

```
<connection maxConnect="180" idleTimeout="15">
```

connection 要素の詳細は、[第 3.2.1.14 項「connection」](#) を参照してください。

maxConnect 属性によって、Reports Server が任意の時点で同時に処理できるリクエストの合計数が制御されます。maxConnect の主な目的は、プログラムまたはプロセスの暴走、あるいは DoS 攻撃から、Reports Server を保護することです。これは常に、同時に処理するクライアントの最大数より大きな値に設定する必要があります。

たとえば、150 のクライアントを同時に処理する場合、maxConnect は 150 より大きな値に設定します。次のように、10 ~ 20% の安全係数を使用します。

$$\text{maxConnect} = 150 + 150 * 0.2 = 180$$

5. HTTP タイムアウト値を設定します (AS にのみ適用)。

HTTP タイムアウト値は、システムで実行時間が最長のレポートの実行に必要な時間に基づいて設定してください。実行時間が最長のレポートの実行に 20 分かかる場合は、HTTP タイムアウトを 20 分より長く設定します。そうしないと、サーバーでレポートの実行が終了しないうちに HTTP タイムアウト・エラーが表示されます。このパラメータは、`ORACLE_HOME/Apache/Apache/conf/httpd.conf` ファイルで設定できます。

20.3 rwdiag を使用したブリッジおよびネットワークのタイムアウト設定

Oracle Reports クライアントと Reports Server が異なるサブネットに属する場合、Oracle Reports ブリッジを使用する必要があります。ブリッジを使用するのは、サーバーの検出に使用するメカニズムがブロードキャストの場合のみです。ネーミング・サービスを使用している場合は、ブリッジは不要です。Oracle Reports ブリッジの詳細は、[第 1.4.1.2 項「複数のサブネットにわたるサーバーの検出」](#) を参照してください。

Oracle Reports ブリッジを適切に操作するには、Oracle Reports ブリッジのタイムアウトを適切に構成する必要があります。また、ブリッジの構成は、サーバーとクライアントで使用されているネットワーク構成ファイルによって異なります。rwdiag ユーティリティを使用すると、ブリッジとネットワークのタイムアウトを正確に計算しやすくなります。

ブリッジ構成ファイル `rwbridge_bridgename.conf` (第 3.3.2 項「ブリッジの構成要素 (bridgeconf.dtd)」を参照) は、通常次ようになります。

```
<bridge version="10.1.2" port="14011" timeout="1200">
  <!--identifier encrypted="no"confidential="yes">USERNAME%/%PASSWORD%
  </identifier-->
  <!--networkConfig file="rwnetwork.conf" ></networkConfig-->
  <trace traceOpts="trace_all"></trace>
  <!-- Specify one or more remote bridges inside remoteBridges element -->
  <!--remoteBridges>
    <remoteBridge host="%HOST%" port="%PORT%"></remoteBridge>
  </remoteBridges-->
</bridge>
```

ネットワーク構成ファイル `rwnetwork.conf` (第 3.3.1 項「ネットワークの構成要素 (rwnetworkconf.dtd)」を参照) には通常、ブリッジ構成で次のような `discoveryService` 要素が含まれます。

```
<discoveryService>
  <multicast channel="228.5.6.7" port="14021" timeout="500" retry="3"/>
  <!--namingService name="Cos" host="%HOST%" port="%PORT%">
  </namingService-->
</discoveryService>
```

Oracle Reports ブリッジを構成するには、まずリモート・ブリッジのレスポンスにかかる時間を調べる必要があります。リモート・ブリッジにかかる時間を調べるには、`rwdiag` ユーティリティを使用してリモート・サーバー名を指定します。

```
rwdiag.bat -find server_name
```

このコマンドによって、リモート・ブリッジを経由したリモート・サーバーの検出にかかる時間が出力されます。

返された値が 1100 ミリ秒 (1.1 秒) とします。返された値を基準にして、この ping 時間にブリッジの処理時間 (100 ミリ秒など) を追加します。これによって、ブリッジ構成ファイルのタイムアウト値は 1200 ミリ秒になります。

ブリッジがレスポンスする前にクライアントがタイムアウトにならないように、ネットワークのタイムアウトも設定する必要があります。ネットワークのタイムアウトと再試行回数の値の乗数が、常にブリッジのタイムアウトよりも大きくなるように設定する必要があります。

動作の信頼性を高めるために、再試行回数の値を常に 1 より大きく設定することをお勧めします。再試行回数の値を 3 とした場合、ネットワークのタイムアウト値の計算は次のようになります。

```
1200 < network_timeout * 3
network_timeout > 400
```

この計算に基づくと、ネットワークの適切なタイムアウト値は 500 ミリ秒になります。

20.4 データへのアクセス

パフォーマンス測定ツールの結果が、データソースのデータへのアクセスに多くの時間が費やされていることを示している場合は、データ構造を再確認して、データの使用方法を定める必要があります。非効率的なスキーマ設計は、レポートのパフォーマンスに深刻な影響を及ぼします。たとえば、過度に正規化されたデータ・モデルでは、なくてもよい結合や問合せが定義されていることがあります。

この項では、レポートで使用するデータの効率性を確認し、向上させる方法について説明します。

- 非 SQL データソース
- データベースの索引
- 計算
- 冗長なデータ
- ブレーク・グループ
- グループ・フィルタ
- リンクの可否

20.4.1 非 SQL データソース

データソースのデータを公開するには、Oracle Reports のプラグgable・データソース・アーキテクチャを使用します。Oracle Reports は、XML、テキスト、JDBC のプラグgable・データソースなどの非 SQL データソースをデフォルトでサポートしています。XML およびテキスト・プラグgable・データソースには、どちらもリモート URL を通じてアクセスできます (ファイアウォールも通過できます)。速度が低下する場合は、データをローカルにダウンロードして、リモート URL ではなくローカル・データ・ストリームを使用します。また、プロキシ・サーバーを迂回するドメインを指定することもできます。

XML プラグgable・データソースでは、実行時の XML データ検証がサポートされます。XML Query ウィザードで「**検証データ・ソース**」チェック・ボックスを選択すると、XML データがフェッチ時に、DTD または XML スキーマに指定されているデータ定義に対して検証されます。これは負荷の非常に高い操作であるため、本番環境ではなく、レポートの開発時にのみ有効です。XML データ・ストリームが非常に大規模である場合、パフォーマンスの違いが顕著に現れます。

データ定義には、XML スキーマまたは DTD スキーマを指定できます。XML スキーマでは型チェックが強制されますが、DTD スキーマでは、すべてのデータが文字列として扱われるため、型チェックは必要ありません。

注意： 非 SQL ソースのデータ型が正しい列に対応していることを確認してください。

XML データ・ストリームに Extensible Stylesheet Language (XSL) ファイルを指定して、任意のフォーマットから単純な行セットまたは行データのフォーマットに変換できます。XSL を実行時に適用する必要がない場合は、データを最初から適切なフォーマットで保持しておくことをお勧めします。

プラグgable・テキスト・データソースでは、セルの囲み文字の使用がサポートされます。これにより、囲みが定義されている各フィールドで、ファイル・フォーマット・レベルのデリミタが無視されます。セルの囲みは、本当に必要でないかぎり使用しないでください。

JDBC プラグgable・データソースでは、JDBC ブリッジ、Thick JDBC ドライバおよび Thin JDBC ドライバがサポートされます。ドライバの選択は、データのフェッチに直接影響します。この選択は、使用されているアプリケーションとデータベースにより決まります。一般に、ネイティブなドライバを使用したほうが、パフォーマンスはよくなります。詳細は、[第 9 章「JDBC PDS の構成と使用」](#)を参照してください。

20.4.2 データベースの索引

SQL の WHERE 句で使用する列には、索引を作成してください。レポートのマスター問合せにある列に対する索引は、それほど効果がないかもしれません。なぜならこれらの問合せでは、データベースへのアクセスは1回のみだからです。パフォーマンスを大きく向上させるには、ディテール問合せで結合されているすべての列で索引を使用するようにします。

注意： 適切な索引がない場合は、全表スキャンの回数が多くなり、結果的にパフォーマンスが悪化します。

20.4.3 計算

レポート内では、サマリー列や式列による計算のほとんどを、データソースで実行するようにします。SQL 問合せの場合、計算は、レポートで取り出したデータに対してではなく、データベースで実行されます。データベースに格納されているユーザー定義ファンクションおよびプロシージャを、Oracle Database または JDBC 問合せの問合せ選択リストに含めることもできます。この場合、結果セットの一部としてデータベースから計算されたデータが返されるため、ローカル・ファンクションを使用するよりも効率的です。

例

次の PL/SQL ファンクションを、Oracle Database に格納するとします。

```
CREATE OR REPLACE FUNCTION CityState (  
  p_location_id world_cities.location_id%TYPE)  
  RETURN VARCHAR2 IS  
  v_result VARCHAR2(100);  
BEGIN  
  SELECT city || ', ' || state  
  INTO v_result  
  FROM world_cities  
  WHERE location_id = p_location_id;  
  RETURN v_result;  
END CityState;
```

このファンクションは、都市名とその州名を、カンマとスペースで区切って返します。このフォーマットはデータベース・レベルで実行され、結果のみがレポートに返されて表示されます。

レポートでは、次のような SQL 問合せを使用します。

```
SELECT location_id, citystate(location_id) "City  
& State" FROM world_cities
```

結果は次のようになります。

```
LOCATION_ID CITY & STATE  
-----  
1 Redwood Shores, California  
2 Seattle, Washington  
3 Los Angeles, California  
4 New York, New York
```

20.4.4 冗長なデータ

パフォーマンスへの影響を考えると、レポートの間合せでは、必要な列のみを選択し、不要な列（冗長な間合せ）を選択しないことが理想的です。間合せが少なければ少ないほど、レポートの実行は高速になります。単一間合せによるデータ・モデルは、複数間合せによるデータ・モデルよりも高速に実行されます。しかし、レポートで、ユーザーごとに別々のフォーマットを使い分けるだけでなく、異なる間合せ文を使用することが必要になるケースもあります。この問題は2種類の異なるレポートを作成することで解決されますが、メンテナンスを容易にするには1つのレポートを保持するほうが望ましいことは明らかです。この場合は、`SRW.SET_MAXROW` ビルトイン・プロシージャを使用して、冗長な間合せを無効にします。

注意： `SRW.SET_MAXROW` ビルトイン・プロシージャを含んだ `SRW` ビルトイン・パッケージの説明については、Oracle Reports のオンライン・ヘルプを参照してください。

例

Before Report トリガーに使用されている次のコードでは、ユーザー・パラメータに応じて、`Query_Emp` または `Query_Dept` のいずれかが無効になります。

```
IF :Parameter_1 = 'A' THEN
  SRW.SET_MAXROW('Query_Emp',0);
ELSE
  SRW.SET_MAXROW('Query_Dept',0);
END IF;
```

注意： `SRW.SET_MAXROW` ビルトイン・プロシージャの使用に意味があるのは、Before Report トリガー内（間合せの解析後）のみです。このポイントよりも後に `SRW.SET_MAXROW` ビルトイン・プロシージャをコールすると、`SRW.MAXROW_UNSET` ビルトイン例外が発生します。その場合、間合せは解析およびバインドされますが、データはレポートに返されません。

XML またはテキスト・ブラッグブル・データソースのいずれかに基づいて間合せを定義するには、間合せで使用するフィールド（使用可能なすべてのフィールドまたはサブセット）を選択します。フィールドのサブセットを使用する必要がある場合は、すべての値をフェッチしてからグループ・フィルタやレイアウト・レベルのフォーマット・トリガーを使用してフィルタリングを行うのではなく、パラメータを使用して間合せレベルでサブセットを使用します。

20.4.5 ブレーク・グループ

ブレーク・グループの数を制限して、レポートのパフォーマンスを向上させます。Oracle Reports では、最下位レベルの子グループを除き、「ブレーク順序」プロパティが設定されているデータ・モデルのすべての列にブレーク・レベルが設定されます。

SQL 間合せでは、この設定が追加の列として間合せの `ORDER BY` 句に追加されます。ORDER BY 句にある列の数が少ないほど、データを必要な順序に並べ替えて返すためのデータベース操作が少なく済みます。ブレーク・グループの作成は、間合せの一部として定義されている `ORDER BY` 句を無視して、`ORDER BY` 句を冗長にする場合があります。こうした `ORDER BY` 句は、データベースで余分な処理が必要となるため削除してください。

レポートでブレーク・グループを使用する必要がある場合は、「ブレーク順序」プロパティを設定する列を極力少なくします。ブレーク順序列には、Reports Builder のデータ・モデル・ビューで、グループ内の列名の左に小さな矢印が表示されています。間合せの最下位レベルの子グループを除く各ブレーク・グループでは、少なくとも1つの列に、「ブレーク順序」プロパティを設定する必要があります。ソートが必要ない列でブレーク・グループを解除すると、パフォーマンスが向上します。

ブレイク・グループは、可能であれば1列に限定します。これらの列はできるだけ小さくすると同時に、可能であればデータベース列（サマリー列や式列ではなく）にします。これらの両条件は、データがフォーマットされる前に Oracle Reports により実行されるローカル・キャッシングを最大限に効率化することに役立ちます。これらの条件は常に満たされるとはかぎりませんが、条件が満たされる場合に適用すると、パフォーマンスは確実に向上します。

20.4.6 グループ・フィルタ

グループ・フィルタは、表示されるレコードの数を減らすために使用されます。フィルタリングは、問合せがデータソースから Oracle Reports にデータを返した後に実行されます。上位5レコードのみを表示するようにフィルタを定義していても、結果セットには、問合せによって返されたすべてのレコードが含まれます。そのため、グループ・フィルタ機能を問合せの WHERE 句または「最大行数」プロパティに組み込むと、効率性が高くなります。これにより、データベースから返されるデータを制限できます。

20.4.7 リンクの可否

複数の表からなるデータ・モデルを作成する方法は多数あります。企業の各部門に所属する全従業員を一覧表示するレポートを作成する場合に、dept と emp を結合する標準的なケースを考えてみます。次のいずれかの問合せを作成できます。

- 単一問合せ：

```
SELECT d.dname, e.ename
FROM emp e, dept d
WHERE e.deptno(+) = d.deptno
```

- deptno に基づく列リンクを使用する2つの問合せ：

```
SELECT deptno, dname FROM dept
SELECT deptno, ename FROM emp
```

レポートのデータ・モデルを設計するときは、単純な単一表問合せを多数作成するのではなく、大きな複数表問合せを少数作成することで、問合せの数を少なくします。Oracle Reports では、問合せが実行されるたびに、解析、バインドおよびカーソルが実行されます。単一問合せによるレポートでは、必要なすべてのデータが、複数のカーソルではなく1つのカーソルで返されます。マスター・ディテール問合せでは、取り出された各マスター・レコードに対して、再度、ディテール問合せが解析、バインドおよび実行されます。この例では、2つの問合せをマージし、ブレイク・グループを使用してマスター・ディテール効果を作成したほうが効率的です。

問合せがより大きく複雑になるほど、メンテナンスが難しくなることに注意してください。パフォーマンスとメンテナンスの要件間のバランスを、どの時点で取るかを定める必要があります。

20.5 データのフォーマット

データがデータソースから取り出された後は、Oracle Reports によりレポート・レイアウトが生成され、出力がフォーマットされます。ペーパー・レイアウトに要する時間は多くの要因に依存しますが、一般に次のことが関係しています。

- オブジェクトが別のオブジェクトにより上書きされることを防ぐために必要な処理
- フォーマット・トリガーで実行される計算またはファンクションの効率性

Web レイアウトのルールは少し異なり、Oracle Reports では、Web ページの所有もレンダリング・メカニズムの制御も行われません。データが、通常の JSP ページに送られるだけです。

この項では、レポートのフォーマットの確認およびチューニングについて説明します。

- ペーパー・レイアウト（フォーマット・トリガーおよびイメージ出力を含む）
- Web レイアウトと JSP レポート定義

20.5.1 ペーパー・レイアウト

デフォルトのペーパー・レイアウトを生成すると、Oracle Reports では、オブジェクトが別のオブジェクトに重なって表示されないように、ほとんどすべてのオブジェクトが枠で囲まれます。実行時は、すべてのレイアウト・オブジェクト（枠、フィールド、ボイラープレートなど）が調べられ、オブジェクトが重なって表示される可能性が判別されます。明らかに重なって表示される可能性がない場合（たとえば、ボイラープレート・テキスト列のヘッダー）は、オブジェクトを直接囲む枠が削除されます。これにより、Oracle Reports によるフォーマットが必要なオブジェクトの数が減り、結果として、パフォーマンスが向上します。

可変で、水平または垂直あるいはその両方の方向に拡張または縮小可能に定義されているオブジェクトには、追加の処理が必要です。この場合、Oracle Reports では、そのオブジェクトと周辺のオブジェクトをフォーマットする前に、そのオブジェクト・サイズのインスタンスを決める必要があります。固定サイズが割り当てられたオブジェクトの場合は、そのサイズとオブジェクト間の位置関係がわかっているため、処理のオーバーヘッドは生じません。

次に、ペーパー・レイアウトの作成時にパフォーマンスの向上に役立つガイドラインを示します。

- 非グラフィカル・レイアウト・オブジェクト（ボイラープレート・テキストやテキスト付きフィールドなど）は、そのフィールドの「垂直拡張度」および「水平拡張度」プロパティを「固定」に設定してサイズを固定します。特に、繰返し枠とそのコンテンツのサイズを固定に設定すると、パフォーマンスが向上します。可変サイズの非グラフィカル・オブジェクトでは、オブジェクトをフォーマットする前にそのサイズを決める必要があるため、処理のオーバーヘッドが生じます。一方、固定の非グラフィカル・オブジェクトでは、追加の処理が必要ないため、オーバーヘッドが少なくなります。
- グラフィカル・レイアウト・オブジェクト（イメージやグラフなど）は、そのオブジェクトの「垂直拡張度」および「水平拡張度」プロパティを「可変」に設定してサイズを変更可能にします。固定のグラフィカル・オブジェクトでは、そのコンテンツを適切なサイズにスケールリングする必要があるため、処理のオーバーヘッドが大きくなります。可変オブジェクトはそのコンテンツにあわせて拡張または縮小されるため、スケールリングは必要ありません。
- テキスト・フィールドの行数は1行で済むようにし、その内容が指定した幅に収まるようにします（SUBSTR ファンクションを使用するなどの方法により）。テキスト・フィールドの行数が複数になる場合、Reports Builder では、ワード・ラップ・アルゴリズムによるフィールドのフォーマットが必要になります。テキスト・フィールドは必ず1行になるようにすると、フォーマット時にワード・ラップ・アルゴリズムによる処理のオーバーヘッドを回避できます。
- 同一フィールドやボイラープレート・テキスト内では、複数の異なるフォーマット属性（フォントなど）の使用は極力少なくします。フォーマットにより長い時間が必要となるためです。
- レポート問合せで SUBSTR ファンクションを使用して、データベース・レベルでデータを切り捨てます。Report Builder レイアウトでフィールドから文字列を切り捨てる方法は避けます。
- ペーパー・レイアウトのみのレポートでは、.rdf および .rep ファイルのほうが .jsp ファイルよりも高速に実行されます。.rdf や .rep ファイルのフォーマットはシリアライズされており、解析が不要なためです。また、.rep ファイルは現行プラットフォーム用に最適化されているため、.rdf ファイルよりも高速に実行されます。

20.5.1.1 フォーマット・トリガー

フォーマット・トリガーを使用すると、オブジェクトの外観の無効化、有効化および変更を動的に実行できます。フォーマット・トリガーは、実行時に関連するオブジェクトのインスタンスが作成されフォーマットされるたびに起動されるため、その使用には注意が必要です。

次の例を考えてみます。

表レポートに、垂直に拡張する繰返し枠が1つあり、その「ページ保護」プロパティが「はい」に設定されています。レポートのフォーマット時に、最初のページの下部に1行分の余白が残っていることが検出されると、Oracle Reports は、繰返し枠の次のインスタンスのフォーマットを開始し、関連するフォーマット・トリガーを起動します。繰返し枠内のオブジェクトの1つが拡張されたことがわかると、繰返し枠のこのインスタンスは次のページに移動されます。繰返し枠のフォーマット・トリガーが再起動されます。この場合、繰返し枠は1回だけ表示されますが（2番目のページの最上部）、フォーマット・トリガーは2回起動されます。フォーマット・トリガーが特定のオブジェクトに対して何回実行されるかわからないため、フォーマット・トリガー内ではDMLを実行しないようにしなければなりません。

この例では、フォーマット・トリガー内にINSERT文があると、データ行が2行挿入されてしまいます。

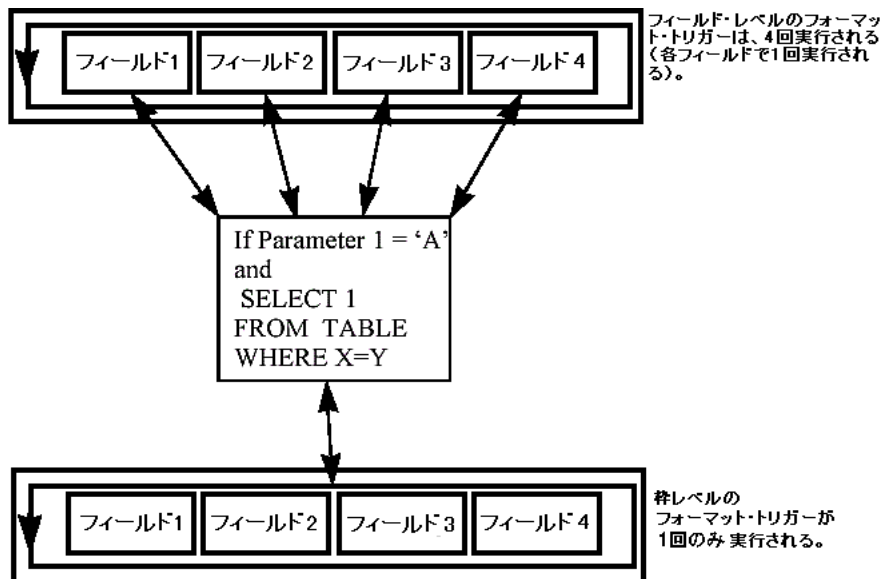
繰返し枠には、データをフィルタリングするフォーマット・トリガーを使用することもできます。ただし、フィルタリングは適切なレベルで使用することで、レポートのパフォーマンスを向上させるだけでなく、この種のレポートで必要とされる複雑さを軽減することもできます。

可能であれば、次の順序でフィルタリングを行います。

- サーバーからデータが返されないようにSQL文を変更します。
- グループ・フィルタを使用して、データ・モデルにフィルタリングを導入します。
- フォーマット・トリガー内でreturn falseを使用します。

フォーマット・トリガーは、その起動回数が最小限になるように、オブジェクトまたは枠階層のできるだけ高いレベルに配置してください。次に例を示します。

図 20-2 フォーマット・トリガー



Oracle Reports 内のトリガーまたは PL/SQL プログラム・ユニットを定義するときは、常にコードの効率性を最大化してください。たとえば、フィールドの表示属性を動的に変更して基準値を超えた値を強調するには、SRW.SET_TEXT_COLOR ビルトイン・プロシージャなどの個別のビルトインを使用してその属性を変更します。

PL/SQL チューニングに関する一般的な問題は、『Oracle Database PL/SQL ユーザーズ・ガイドおよびリファレンス』を参照してください。

レイアウト・オブジェクト（枠や繰返し枠など）はビットマップ・ファイルとしてレンダリングされないため、これらのオブジェクトに透明な境界線と塗りつぶしパターンを割り当てると、パフォーマンスが向上します。

20.5.1.2 イメージ出力

イメージのサポートに関連する環境変数を慎重に設定することによって、イメージを含むレポートのパフォーマンスを向上させることができます。

PDF ファイルまたはプリンタへのグラフ出力のパフォーマンスの向上

REPORTS_GRAPH_IMAGE_DPI 環境変数は、PDF ファイルまたはプリンタにグラフを出力する際のインチ当たりのドット数 (DPI) を指定します。この環境変数のデフォルト値は、72 DPI に設定されています。これは、レポートのファイル・サイズを小さくするとともに、レポートの生成に要する時間を最小化するためです。72 DPI より高い値を指定すると、PDF ファイルやプリンタに出力されるグラフのイメージ解像度が向上します。ただし、これにより、ファイル・サイズが大きくなるだけでなく、レポート出力の生成に、より時間がかかります。

値を 250 に設定すると、Oracle Reports グラフを含むレポートの生成にかかる時間が、同じレポートを値を 72 DPI で生成する場合の 5～6 倍になります。PDF ファイルのサイズも 5～6 倍になります。

この機能は、PDF 出力およびプリンタ出力専用になっているため、現時点の Oracle Reports 配布機能ではサポートされていません。

注意： DPI 値を 250 超に設定し、グラフの大きさを 5"x5" より大きくすると、JVM の Out Of Memory エラーを回避するために、REPORTS_JVM_OPTIONS 環境変数を使用して JVM ヒープ・サイズの値を変更することも必要になります。

詳細は、[第 B.1.44 項「REPORTS_GRAPH_IMAGE_DPI」](#) を参照してください。

JPEG/GIF/PNG 出力イメージ形式のパフォーマンスの向上

入力イメージ形式が JPEG の場合、REPORTS_OUTPUTIMAGEFORMAT 環境変数を GIF または PNG に設定しないことをお勧めします。設定すると、イメージのサイズがさらに大きくなり、パフォーマンスが低下する場合があります。同様に、入力イメージ形式が GIF または PNG の場合、REPORTS_OUTPUTIMAGEFORMAT 環境変数を JPEG に設定しないことをお勧めします。パフォーマンスを向上させるには、入力と出力の両方に同じ形式を使用します。

詳細は、[第 B.1.52 項「REPORTS_OUTPUTIMAGEFORMAT」](#) を参照してください。

JPEG イメージのパフォーマンスの向上

REPORTS_JPEG_QUALITY_FACTOR 環境変数によって、JPEG イメージに求められるイメージ画質のレベルが指定されます。JPEG イメージの画質とイメージのサイズのトレードオフが制御されます。イメージの画質がよいほど、イメージ・ファイルのサイズが大きくなり、パフォーマンスが低下します。パフォーマンスを向上させるには、値を 0 に設定します。デフォルト値は 100（最高画質）です。値を 75 に設定すると、適切な圧縮率を維持したまま、イメージの画質が向上します。

詳細は、[第 B.1.46 項「REPORTS_JPEG_QUALITY_FACTOR」](#) を参照してください。

20.5.2 Web レイアウトと JSP レポート定義

Oracle Reports では、好みの Web オーサリング・ツールを使用して Web ページの静的な部分を設計してから、Reports Builder を使用して、ページの適切なセクションに動的な部分（データ）を挿入できます。設計の悪い Web ページは、体感パフォーマンスに影響を与えます。または、事前定義済の Oracle Database Web テンプレートを使用して、Web ページを構築することもできます。

JSP ファイルには、Java コードを記述しないようにします。表示ロジックにビジネスおよびデータ・アクセス Java コードを混在させると、JSP のフットプリントが増大し、システム・リソースの使用と管理の効率性が制限されます。

Web ページのカスタマイズされたフォーマットは、常に負荷の高い操作になります。Oracle Reports でネイティブに実行できないフォーマット（データ・ブロックのフォアグラウンド・カラーの変更など）は、すべて Java を使用して実行してください。フォーマット目的で、PL/SQL ラッパーを使用することはお勧めしません。

.jsp レポート定義には、ペーパー・レイアウト定義と Web レイアウト定義の両方を含めることができます。Oracle Reports では、レポートの実行時に、常にペーパー・レイアウト定義が最初にフォーマットされます。JSP レポートの Web レイアウト・セクションに、ペーパー・レイアウト・オブジェクトを参照する `<rw:include>` タグが含まれる可能性があります。JSP レポートにペーパー・レイアウト・オブジェクトへの参照がまったくない場合は、SUPPRESSLAYOUT コマンドライン・キーワードを使用して、ペーパー・レイアウトのフォーマットが実行されないようにすることをお勧めします。

20.6 レイアウトに関する一般的なガイドライン

この項では、パフォーマンスの向上を目的としてレポートのレイアウトをデザインする場合の各種ガイドラインを示します。

- 事前フェッチ
- 一括処理と配布

20.6.1 事前フェッチ

Oracle Reports では、レポートのマージンまたはレポートのヘッダー・ページに総ページ数や総計などのデータを表示できます。このオプションは便利ですが、レポート全体を事前にフェッチすることが必要になります。事前フェッチでは、最初のページを出力する前に、レポート全体を処理する必要があります。通常は、必要に応じてページのフォーマットが行われます。

事前フェッチ機能は、レポート生成にかかる合計時間には影響しませんが、必要となる一時記憶域の量と最初のページが表示されるまでの時間に影響します。これは、実際のパフォーマンスとは区別される体感パフォーマンスの例です。レポートを本番環境の画面に出力する場合は、パフォーマンスの変化が許容範囲と思われる場合を除き、事前フェッチは使用しないでください。

20.6.2 一括処理と配布

レポートの一括処理を使用して、レポート・レイアウトをヘッダー、本体、トレーラという異なる 3 つのセクションで構成できます。1 つのレポートにこれら 3 つのセクションを含めることも、1 つのレポート内で 3 つの異なるレポートとして表示することもできます。Oracle Reports では、グループ・レコード・レベルで一括処理を制御でき、より高いレベルの精度が提供されます。これは、個々のセクションの「配布」プロパティと「繰返し」プロパティにより可能になります。一括処理を配布オプションとともに使用すると、パフォーマンスは確実に向上します。これにより、レポートの各セクションに複数のフォーマットを定義し複数の宛先に送信することも可能になります。配布オプションを設定しておけば、レポートの実行は 1 回のみです。つまり、問合せの 1 回の実行で複数の宛先に出力されます。以前は、レポートを複数回実行する必要がありました。

レポートで一括処理および配布を実装する場合は、セクションの「繰返し」プロパティをデータ・モデルのブレイク・グループに設定すると、セクション・レベルの配布を生成することができます。これにより、そのブレイク・グループの列レコードごとにセクションのインスタンスが生成されます。その後、セクションの各インスタンスを適宜配布できます（たとえば、MANAGER グループ内の個々のマネージャへの配布）。

「繰返し」プロパティを複数のレポート・セクション（ヘッダー、メインおよびトレーラ）に対して設定する場合は、「繰返し」プロパティの値は同じデータ・モデルのブレイク・グループに設定する必要があります。ヘッダー、メインおよびトレーラ・セクションのいずれかで「繰返し」プロパティを異なるデータ・モデルのブレイク・グループに設定した場合、Oracle Reports では次のいずれかのメッセージが表示されます。

REP-0069: Internal Error

REP-57054: In-Process job terminated: Terminated with error

REP-594: No report output generated

20.7 Forms からの Oracle Reports のコール

Forms Builder および Reports Builder を使用して構築したアプリケーションでは、そのアプリケーションの OracleAS Forms Services セクションにより取り出し済または更新済のデータを基にしたレポートが必要です。Oracle Reports と OracleAS Forms Services 間の緊密な製品統合により、関連する製品間でデータ・ブロックを渡すことができ、同じ問合せを行う必要がなくなります。問合せのパーティション化と呼ばれる技法を使用することで、Oracle Reports はデータのフォーマットのみを行い、トリガーや文字パラメータによる問合せの動的な変更を無視できます。

OracleAS Forms Services と Oracle Reports 間でのデータの受渡しは、レコード・グループとデータ・パラメータを RUN_REPORT_OBJECT ビルトイン（OracleAS Forms Services から Oracle Reports をコールする）と組み合わせて使用することで行われます。

OracleAS Forms Services アプリケーションからのレポートのコールの詳細は、OTN (<http://www.oracle.com/technology/products/forms/techlisting10g.html>) にあるホワイト・ペーパー『Integrating Oracle Reports Services 10g in Oracle Forms Services 10g』を参照してください。

注意： データ・パラメータが過度に広範な場合や問合せが非常に複雑な場合を除いて、体感パフォーマンスの向上はごくわずかです。また、フォームから渡されるデータを受け取ることができるのは、レポートの最上位レベルのグループのみです。

20.8 レポートの実行

特定のランタイム・オプションを設定することで、全体的なパフォーマンスをさらに向上させることができます。

- Reports Builder では、ペーパー・レイアウト定義とバインド変数に対して、エラー・チェックが自動的に実行されます。ランタイム・パラメータ RUNDEBUG=NO を設定すると、実行時のこのエラー・チェックを無効にできます。
- JSP レポート定義では、タグ検証が実行され、重複フィールドの ID や誤った形式の属性などの項目がチェックされます。この機能は、本番環境ではなく、設計フェーズでのみ有効です。デフォルトでは、OracleAS Reports Services のタグ検証はオフです。このオプションをオンにするには、HTTP リクエストで `validatetag=yes` を指定します（例、`http://my.server.com/myreport.jsp?validatetag=yes`）。

注意： `validatetag=yes` を使用すると、パフォーマンスが低下します。

- デフォルトでは、`rwrun` コマンドと `rwervlet` コマンドの両方で使用される `RECURSIVE_LOAD` コマンドライン・キーワードが `YES` に設定されており、PL/SQL プログラム・ユニットの無効な外部参照が自動的に再コンパイルされます。このキーワードは開発環境でのみ使用されるため、本番環境では、`RECURSIVE_LOAD=NO` と設定します。
- SQL 問合せでは、データのフェッチに Oracle データベースの配列処理機能を利用します。これにより、レコードは 1 回に 1 つではなく一括してデータベースからフェッチされ、結果的にデータベースのコール回数が少なくなります。ただし、配列処理では、返されたレコード配列を格納するために、実行プラットフォームでより多くのメモリーが必要になります。本番環境でネットワークの負荷（ネットワーク・トリップ数）を減らすには、`ARRAYSIZE` コマンドライン・キーワードの値を大きな値に設定します（KB 単位）。
- [第 20.2 項「Reports Server の構成のチューニング」](#) で説明したように、`Reports Servlet` (`rwervlet`) または `Reports Client` (`rwclient`) で多数のレポートを実行する場合は、`maxQueueSize` および `cacheSize` の値を反映するように `EXPIRATION` コマンドライン・キーワードを設定します。たとえば、サーバー構成ファイルの `queue` 要素で `maxQueueSize=6000` と指定している場合は、最大 6000 件のジョブをジョブ・キューに格納できます。`EXPIRATION=1440`（1 日）と設定した状態で、1 日に 6000 件超のジョブを実行すると、`EXPIRATION` の時間に達する前に一部のジョブが失われる場合があります。これは、`maxQueueSize` とサーバーの安定性を確保するために、期限切れになっていないジョブでも `Reports Server` によって削除されるためです。また、サーバー構成ファイルの `cache` 要素で、6000 件のジョブを維持するために十分なキャッシュ・サイズが割り当てられるように `cacheSize` を指定する必要があります。一般的なガイドラインとして、1 日に実行するジョブの数に応じて、`EXPIRATION`、`maxQueueSize` および `cacheSize` を設定します。
- `LONG`、`CLOB` または `BLOB` データ型を使用してレポートで大量のデータを取り出す場合は、`LONGCHUNK` コマンドライン・キーワードをできるだけ大きな値に設定します。これにより、`LONG` 値を取り出すための、Oracle Reports による変更分の数が減ります。Oracle データベース・サーバーでは、`LONG` または `LONG RAW` のかわりに、より効率的な `CLOB` または `BLOB` データ型を使用します。
- ペーパー・パラメータ・フォームが必要ない場合は、`PARAMFORM` コマンドライン・キーワードを `NO` に設定します。
- `PostScript` に出力するときは、`COPIES` コマンドライン・キーワードの使用に注意する必要があります。 `COPIES` を 1 よりも大きな値に設定すると、Oracle Reports では、ページを照合するために、各ページを一時記憶域に保存することが必要になります。これにより、一時的に使用されるディスク領域の量が増加し、追加ファイルの書込みオーバーヘッドによりパフォーマンスが悪化します。
- レポートを PDF 出力に生成する場合は、`PDFCOMP` コマンドライン・キーワードを `NO` に設定します。PDF 出力はデフォルトで圧縮されます。圧縮ファイルはダウンロードが短時間で済みますが、圧縮ファイルの生成にかかる時間は、非圧縮ファイルの場合よりかなり長くなります。

第 V 部

付録

第 V 部は、Reports Builder および OracleAS Reports Services 環境での動作に関する追加の詳細情報が記載されている付録です。これには、Oracle Reports コマンドやそれに関連するコマンドライン・オプションの情報、Reports 関連の環境変数に関する詳細、およびバッチ・スクリプトを使用してレポートを OracleAS Portal に登録する方法について記載されています。

第 V 部には、次の付録があります。

- [付録 A 「コマンドライン・キーワード」](#)
- [付録 B 「環境変数」](#)
- [付録 C 「OracleAS Portal へのレポートのバッチ登録」](#)
- [付録 D 「OracleAS Reports Services のトラブルシューティング」](#)
- [付録 E 「Reports Server とブリッジの診断ユーティリティ」](#)

コマンドライン・キーワード

この付録では、Oracle Reports 実行可能ファイル (rwclient、rwrun、rwbuilder、rwconverter、rwservlet、rwcgi および rserver) とともに使用できるコマンドライン・キーワードについて説明し、例を示します。各キーワードの説明には、そのキーワードを使用できる実行可能ファイルを示した表を記載します。

注意： ランタイム URL でのコマンドライン・キーワードの使用例は、[第 13 章「レポート・リクエストの実行」](#) を参照してください。

この付録には、次の項があります。

- [コマンドラインの使用方法](#)
- [Oracle Reports 実行可能ファイルの概要](#)
- [コマンドライン・キーワード](#)

この付録に記載されている情報は、Oracle Reports のオンライン・ヘルプにも記載されています。このオンライン・ヘルプは、「はじめに」の「[関連ドキュメント](#)」の説明のように、Reports Builder または Oracle Technology Network (OTN) から表示できます。

A.1 コマンドラインの使用法

コマンドラインでの Oracle Reports コマンドは、一般的に次の形式をとります。

```
executable_name keyword=value, keyword=value, ...
```

ここで、各 `keyword=value` のペアはコマンドライン・オプションと呼ばれます。

キーワードは必ず指定する必要がありますが、実行可能ファイル名の後ろに順不同で使用できます。

A.1.1 全般的な使用上の注意

- オプションの等号記号の前後に空白を入れることはできません。
- オプションを区切るには、1つ以上の空白を使用します。コンマで区切ることはできません。
- 値は一重引用符または二重引用符で囲む場合もあります。一重引用符または二重引用符の効果はオペレーティング・システムによって異なります。
- どのオプションでも、`keyword=` 部分では大文字と小文字は区別されません。`value` 部分は、使用しているオペレーティング・システムによって大文字と小文字が区別される場合があります。
- コマンドラインから一重引用符を渡す場合、2つの引用符（1つはエスケープとして、もう1つは実際の引用符として）入力する必要があります。次に例を示します。

```
rwrun REPORT=myrep DESTYPE=file DESNAME=run.out BATCH=yes p_value="Roy"'s Batch Report"
```

- フルパス名は、すべてのファイル参照に対してサポートされています（例：`DESNAME=/revenues/q1/nwsales`）。フルパス名を指定しない場合は、Oracle Reports のファイル検索メソッドが使用されてファイルが検索されます。キーワード値にファイル名を含むパスを指定しないと、Reports Server は `REPORTS_PATH` 環境変数からファイル検索を試みます。
- クライアントのコマンドラインで指定されたすべてのファイル名とパスは、サーバー・マシン上のファイルとディレクトリを参照します。ただし、次のコマンドライン・キーワードに対して指定されたファイルはこれらを参照しません。
 - `CMDFILE=filename`。この場合は、指定された `CMDFILE` が読み取られ、元のコマンドライン（`CMDFILE` が記述されているコマンドライン）に追加されてから、Reports Server に送信されます。ランタイム・エンジンによって、このコマンド・ファイルが再び読み取られることはありません。
 - `DESNAME=filename DESTYPE=LOCALFILE`。この場合、`DESNAME` はクライアント・マシンにあるファイルを参照します。

A.1.2 ルール

- ランタイム・パラメータ・フォームに入力された値は、コマンドラインで入力された値よりも優先されます。たとえば、コマンドラインで `rwrun` を `COPIES=1` で指定しても、ランタイム・パラメータ・フォームで `COPIES=2` と指定した場合は、生成されるレポートの部数は2です。
- コマンドラインに入力した値は、コマンド・ファイルで指定した値よりも優先されます。たとえば、コマンドラインで `rwrun` を `COPIES=1` および `CMDFILE=RUNONE`（コマンド・ファイル）で指定し、コマンド・ファイル `RUNONE` に `rwrun COPIES=2` という記述が含まれている場合でも、生成されるレポートの部数は1です。

- DESTYPE、DESNAME、DESFORMAT、ORIENTATION および COPIES の値は、いくつかの異なる場所で指定できます。次のリストは、これらの値を指定する場所の優先順位を高いほうから順に示しています。
 1. 「ジョブの印刷」ダイアログ・ボックス
 2. ランタイム・パラメータ・フォーム (Runtime Parameter Form)
 3. 「プリファレンス」ダイアログ・ボックスの「ランタイム・パラメータ」タブ / 「ランタイム設定」タブ
 4. コマンドラインのキーワード
 5. レポート定義で指定した値
 6. 「プリンタの選択」ダイアログ・ボックス

A.2 Oracle Reports 実行可能ファイルの概要

この項では、Oracle Reports 実行可能ファイルと各実行可能ファイルで使用できるキーワードについて簡単に説明します。

- キーワードの使用方法の要約
- [rwclient](#)
- [rwrund](#)
- [rwbuilder](#)
- [rwconverter](#)
- [rwservlet](#)
- [rwcgi](#)
- [rwserver](#)
- [rwbridge](#)

A.2.1 キーワードの使用方法の要約

表 A-1 では、すべての Oracle Reports コマンドライン・キーワードをアルファベット順に要約したリストと、各キーワードを使用できる Oracle Reports 実行可能ファイルを示しています。

* Oracle9iAS Portal リリース 1 および Oracle WebDB リリース 2.2 に対する下位互換性を確保するために維持されています。

表 A-1 キーワードおよび各キーワードを使用できる実行可能ファイル

キーワード	rwclient	rwrund	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver	rwbridge
ACCESSIBLE	○	○	○	×	○	○	×	×
ARRAYSIZE	○	○	○	×	○	○	×	×
AUTHID	○	○	×	×	○	○	○	○
AUTOCOMMIT	○	○	○	×	○	○	×	×
BACKGROUND	○	×	×	×	○	○	×	×
BATCH	×	×	×	○	×	×	○	×
BCC	○	○	×	×	○	○	×	×
BLANKPAGES	○	○	○	×	○	○	×	×
BUFFERS	○	○	○	×	○	○	×	×
CACHELOB	○	○	○	×	○	○	×	×
CC	○	○	×	×	○	○	×	×

表 A-1 キーワードおよび各キーワードを使用できる実行可能ファイル (続き)

キーワード	rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver	rwbridge
CELLWRAPPER	○	○	×	×	○	○	×	×
CMDFILE	○	○	○	○	×	×	×	×
CMDKEY	×	×	×	×	○	×	×	×
COLLATE	○	○	×	×	○	○	×	×
CONTAINSHTMLTAGS	○	○	○	×	○	○	×	×
CONTAINSOLE	○	○	○	×	○	○	×	×
CONTENTAREA*	○	○	×	×	○	○	×	×
COPIES	○	○	×	×	○	○	×	×
CUSTOMIZE	○	○	×	○	○	○	×	×
DATEFORMATMASK	○	○	×	×	○	○	×	×
DELAUTH	×	×	×	×	○	○	×	×
DELIMITED_HDR	○	○	×	×	○	○	×	×
DELIMITER	○	○	×	×	○	○	×	×
DESFORMAT	○	○	×	×	○	○	×	×
DESNAM	○	○	×	×	○	○	×	×
DEST	×	×	×	○	×	×	×	×
DESTINATION	○	○	×	×	○	○	×	×
DESTYPE	○	○	×	×	○	○	×	×
DISTRIBUTE	○	○	×	×	○	○	×	×
DTYPE	×	×	×	○	×	×	×	×
DUNIT	×	×	×	○	×	×	×	×
ENGINERESPONSETIMEOUT	○	×	×	×	○	○	×	×
ENVID	○	×	×	×	○	○	×	×
EXPIRATION	○	×	×	×	○	○	×	×
EXPIREDAYS	×	×	×	×	○	○	×	×
EXPRESS_SERVER	○	○	○	×	○	○	×	×
FORMSIZE	×	×	×	○	×	×	×	×
FROM	○	○	×	×	○	○	×	×
GETJOBID	×	×	×	×	○	○	×	×
GETSERVERINFO	×	×	×	×	○	×	×	×
HELP	×	×	×	×	○	○	×	×
ITEMTITLE	○	○	×	×	○	○	×	×
JOBNAME	○	×	×	×	○	○	×	×
JOBTYPE	○	×	×	×	○	○	×	×
JVMOPTIONS	×	○	○	○	×	×	○	×
KILLENGINE	×	×	×	×	○	×	×	×
KILLJOBID	×	×	×	×	○	○	×	×
LONGCHUNK	○	○	○	×	○	○	×	×
MIMETYPE	×	×	×	×	○	○	×	×

表 A-1 キーワードおよび各キーワードを使用できる実行可能ファイル (続き)

キーワード	rwclient	rwruntime	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver	rwbridge
MODE	○	○	×	×	○	○	×	×
MODULE REPORT	○	○	○	×	○	○	×	×
NAME	×	×	×	×	×	×	×	○
NONBLOCKSQL	○	○	○	×	○	○	×	×
NOTIFYFAILURE	○	○	×	×	○	○	×	×
NOTIFYSUCCESS	○	○	×	×	○	○	×	×
NUMBERFORMATMASK	○	○	×	×	○	○	×	×
OLAP_CON	○	○	○	×	○	○	×	×
ONFAILURE	○	○	○	×	○	○	×	×
ONSUCCESS	○	○	○	×	○	○	×	×
ORIENTATION	○	○	×	×	○	○	×	×
OUTPUTFOLDER*	○	○	×	×	○	○	×	×
OUTPUTIMAGEFORMAT	○	○	○	×	○	○	×	×
OUTPUTPAGE	○	○	×	×	○	○	×	×
OVERWRITE	×	×	×	○	×	×	×	×
P_AVAILABILITY	×	×	×	○	×	×	×	×
P_DESCRIPTION	×	×	×	○	×	×	×	×
P_FORMATS	×	×	×	○	×	×	×	×
P_JDBCPS	○	○	○	×	○	○	×	×
P_NAME	×	×	×	○	×	×	×	×
P_OWNER	×	×	×	○	×	×	×	×
P_PFORMTEMPLATE	×	×	×	○	×	×	×	×
P_PRINTERS	×	×	×	○	×	×	×	×
P_PRIVILEGE	×	×	×	○	×	×	×	×
P_SERVERS	×	×	×	○	×	×	×	×
P_TRIGGER	×	×	×	○	×	×	×	×
P_TYPES	×	×	×	○	×	×	×	×
PAGEGROUP	○	○	×	×	○	○	×	×
PAGESIZE	○	○	○	○	○	○	×	×
PAGESTREAM	○	○	×	×	○	○	×	×
PARAMFORM	×	×	×	×	○	○	×	×
PARSEQUERY	×	×	×	×	○	○	×	×
PDFCOMP	○	○	×	×	○	○	×	×
PDFEMBED	○	○	×	×	○	○	×	×
PFACTION	○	×	○	×	×	×	×	×
PRINTJOB	×	×	○	×	×	×	×	×
READONLY	○	○	○	×	○	○	×	×
RECURSIVE_LOAD	○	○	×	○	○	○	×	×
REPLYTO	○	○	×	×	○	○	×	×

表 A-1 キーワードおよび各キーワードを使用できる実行可能ファイル (続き)

キーワード	rwclient	rwrn	rwbuilder	rwconverter	rservlet	rwcgi	rwserver	rwbridge
REPORT MODULE	○	○	○	×	○	○	×	×
ROLE	○	○	×	×	○	○	×	×
RUNDEBUG	○	○	○	×	○	○	×	×
SAVE_RDF	×	○	○	×	×	×	×	×
SCHEDULE	○	×	×	×	○	○	×	×
SERVER	○	×	×	×	○	○	○	×
SHOWAUTH	×	×	×	×	○	○	×	×
SHOWENV	×	×	×	×	○	○	×	×
SHOWJOBID	×	×	×	×	○	×	×	×
SHOWJOBS	×	×	×	×	○	○	×	×
SHOWMAP	×	×	×	×	○	○	×	×
SHOWMYJOBS	×	×	×	×	○	×	×	×
SHUTDOWN	×	×	×	×	×	×	○	○
SITENAME*	○	○	×	×	○	○	×	×
SOURCE	×	×	×	○	×	×	×	×
SQLTRACE	○	○	○	×	○	○	×	×
SSOCONN	×	×	×	×	○	×	×	×
STATUSFOLDER*	○	○	×	×	○	○	×	×
STATUSFORMAT	×	×	×	×	○	×	×	×
STATUSPAGE	○	○	×	×	○	○	×	×
STYPE	×	×	×	○	×	×	×	×
SUBJECT	○	○	×	×	○	○	×	×
SUPPRESSLAYOUT	○	○	○	×	○	○	×	×
TOLERANCE	○	×	×	×	○	○	×	×
TRACEFILE	×	○	○	×	×	×	○	×
TRACEMODE	×	○	○	×	×	×	○	×
TRACEOPTS	○	○	○	×	○	○	○	×
UPGRADE_PLSQL	×	×	×	○	×	×	×	×
URLPARAMETER	○	×	×	×	○	○	×	×
USEJVM	○	×	×	×	×	×	×	×
USERID	○	○	○	○	○	○	×	×
USERSTYLES	○	○	○	×	○	○	×	×
VALIDATETAG	×	×	○	×	×	×	×	×
WEBSERVER_DEBUG	×	×	○	×	×	×	×	×
WEBSERVER_DOCROOT	×	×	○	×	×	×	×	×
WEBSERVER_PORT	×	×	○	×	×	×	×	×

A.2.2 rwclient

rwclient (Reports クライアント) は、コマンドラインを解析し、指定された Reports Server に転送します。

クライアントのコマンドラインで指定されたすべてのファイル名とパスは、サーバー・マシン上のファイルとディレクトリを参照します。ただし、次のコマンドライン・キーワードに対して指定されたファイルはこれらを参照しません。

- **CMDFILE=filename**。この場合は、指定された **CMDFILE** が読み取られ、元のコマンドライン (**CMDFILE** が記述されているコマンドライン) に追加されてから、Reports Server に送信されます。ランタイム・エンジンによって、このコマンド・ファイルが再び読み取られることはありません。
- **DESNAME=filename DESTYPE=LOCALFILE**。この場合、DESNAME はクライアント・マシンにあるファイルを参照します。

rwclient とともに使用できるキーワードは、表 A-1 を参照してください。

例

例 1: ペーパー・レポートを実行してキャッシュ

```
rwclient server=myrepssrv report=test.rdf userid=scott/tiger@mydb
desformat=pdf DESTYPE=cache
```

例 2: レポート出力をファイルに送信

```
rwclient server=myrepssrv report=test.rdf userid=scott/tiger@mydb
desformat=pdf DESTYPE=file DESNAME=c:¥mydir¥test
```

例 3: レポート出力をプリンタに送信

```
rwclient server=myrepssrv report=test.rdf userid=scott/tiger@mydb
desformat=pdf DESTYPE=printer DESNAME=myprinter
```

例 4: レポート出力を電子メールで送信

```
rwclient server=myrepssrv report=test.rdf userid=scott/tiger@mydb
desformat=pdf DESTYPE=mail DESNAME="emp1@comp.com, emp2@comp.com"
cc="emp3@comp.com" bcc="mgr@comp.com" replyto="me@comp.com"
from="me@comp.com"
```

例 5: レポート出力を WebDAV (いずれかの WebDAV サーバーまたは OracleAS Portal WebDAV) に送信

```
rwclient server=myrepssrv report=test.rdf userid=scott/tiger@mydb
desformat=htmlcss DESTYPE=webdav
DESNAME="http://myusername:mypassword@mywebdavserv.com/mydir/test.html"
```

例 6: レポート出力を OracleAS Portal に送信

```
rwclient server=myrepssrv report=test.rdf userid=scott/tiger@mydb
DESTYPE=oracleportal desformat=PDF PAGEGROUP=mypagegrp
OUTPUTPAGE=reports_output ITEMTITLE=pushtportal STATUSPAGE=result
```

例 7: XML PDS レポート出力をファイルに送信

```
rwclient server=myrepssrv report=myxmlpdstest.rdf DESTYPE=file
desformat=PDF desname=c:¥mydir¥my.pdf
```

例 8: JDBC PDS レポート出力をファイルに送信

```
rwclient server=myrepserver report=myjdbcpdstest.rdf DESTYPE=file
desformat=PDF desname=c:¥mydir¥myxml.pdf p_
jdbcpds=sybuser/sybpwd@server1.mydomain.com:1300
```

例 9: レポート出力を複数の宛先に配布

```
rwclient server=myrepserver report=test.rdf userid=scott/tiger@mydb
DISTRIBUTE=yes DESTINATION=c:¥mydistribute.xml
```

例 10: スケジュールされたレポートの実行

```
rwclient server=myrepserver report=test.rdf SCHEDULE="every first fri
of month from 15:53 Oct 23, 1999 retry 3 after 1 hour" destype=file
desformat=pdf desname=test.pdf
```

例 11: 保護された Reports Server の使用

```
rwclient server=myrepserver report=test.rdf userid=scott/tiger@mydb
desformat=pdf destype=file desname=test.pdf AUTHID=myadmin/myadmin
```

例 12: レポートを電子メールによる通知で実行

```
rwclient server=myrepserver report=test.rdf userid=scott/tiger@mydb
destype=file desformat=pdf desname=test.pdf
NOTIFYSUCCESS="emp@comp.com" NOTIFYFAILURE="admin@comp.com"
```

例 13: URL エンジンでフェッチする URL を指定するレポートの実行

```
rwclient server=myrepserver report=test.rdf userid=scott/tiger@mydb
destype=file desformat=pdf desname=test.pdf JOBTYP=rwurl
URLPARAMETER="http://www.oracle.com"
```

例 14: トレースを使用したレポートの実行

```
rwclient server=myrepserver report=test.rdf userid=scott/tiger@mydb
destype=file desformat=pdf desname=test.pdf TRACEOPTS=trace_all
```

A.2.3 rwrn

rwrn (Reports Runtime) は、OracleAS Reports Services のインプロセス・サーバーを使用してレポートを実行します。

注意: rwrn は、テスト目的にのみ使用することをお勧めします。本番環境では、rwservlet と rwclient を使用して、OracleAS Reports Services の威力をフル活用してください。

rwrn とともに使用できるキーワードは、表 A-1 を参照してください。

例

例 1: レポートのカスタマイズ

```
rwrn userid=scott/tiger@mydb report=emp.rdf
CUSTOMIZE=empcustomize.xml destype=file desformat=pdf desname=emp.pdf
```

例 2: レポート出力をファイルに送信

```
rwrn report=test.rdf userid=scott/tiger@mydb desformat=pdf
DESTYPE=file DESNAME=c:¥mydir¥test.pdf
```

例 3: レポート出力をプリンタに送信

```
rwr run report=test.rdf userid=scott/tiger@mydb desformat=pdf
DESTYPE=printer DESNAME=myprinter
```

例 4: レポート出力を電子メールで送信

```
rwr run report=test.rdf userid=scott/tiger@mydb desformat=pdf
DESTYPE=mail DESNAME="emp1@comp.com, emp2@comp.com"
cc="emp3@comp.com" bcc="mgr@comp.com" replyto="me@comp.com"
from="me@comp.com"
```

例 5: レポート出力を WebDAV (いずれかの WebDAV サーバーまたは OracleAS Portal WebDAV) に送信

```
rwr run report=test.rdf userid=scott/tiger@mydb desformat=htmlcss
DESTYPE=webdav
"DESNAME"="http://myusername:mypassword@mywebdavserv.com/mydir/test.html"
```

例 6: XML PDS レポート出力をファイルに送信

```
rwr run report=myxmlpdstest.rdf destype=file desformat=PDF
desname=c:¥mydir¥my.pdf
```

例 7: JDBC PDS レポート出力をファイルに送信

```
rwr run report=myjdbcpdstest.rdf destype=file desformat=PDF
desname=c:¥mydir¥myxml.pdf
P_JDBCPDS=sybusersybpwd@server1.mydomain.com:1300
```

例 8: レポート出力を複数の宛先に配布

```
rwr run report=test.rdf userid=scott/tiger@mydb DISTRIBUTE=yes
DESTINATION=c:¥mydistribute.xml
```

例 9: 保護された Reports Server の使用

```
rwr run report=test.rdf userid=scott/tiger@mydb desformat=pdf
destype=file desname=test.pdf AUTHID=myadmin/myadmin
```

例 10: レポートを電子メールによる通知で実行

```
rwr run report=test.rdf userid=scott/tiger@mydb destype=file
desformat=pdf desname=test.pdf NOTIFYSUCCESS="emp@comp.com"
NOTIFYFAILURE="admin@comp.com"
```

例 11: トレースを有効化してレポートを実行

```
rwr run report=test.rdf userid=scott/tiger@mydb destype=file
desformat=pdf desname=test.pdf TRACEOPTS=trace_prf TRACEMODE=trace_
replace
```

A.2.4 rwbuilder

rwbuilder は、Reports Builder を起動します。コマンドラインで rwbuilder に REPORT|MODULE キーワードを指定すると、Reports Builder が起動し、指定されたレポートがオブジェクト・ナビゲータ内で強調表示されます。指定されたレポートがない場合、Reports Builder は、既存のレポートを開くか新規レポートを作成するかを選択する「ようこそ」ダイアログとともに起動します。

rwbuilder とともに使用できるキーワードは、表 A-1 を参照してください。

例

```
rwbuilder report=myrep.rdf userid=scott/tiger@mydb
```

A.2.5 rwconverter

rwconverter (Reports Converter) によって、1つ以上のレポート定義または PL/SQL ライブラリをある保存形式から別の保存形式に変換できます。たとえば、rwconverter を使用して次のような処理ができます。

- レポート・ファイルと XML ファイルの組合せによる新しいレポートの作成
- .rdf ファイルに格納されたレポートを .rep、.rex、.jsp または .tdf (テンプレート) ファイルに変換

注意： レポートをテンプレートに変換する場合、レポートのヘッダー・セクションとトレーラ・セクションおよびマージン領域にあるオブジェクトのみがテンプレートに使用されます。メイン・セクションにあるオブジェクトは無視されます。

- .rex ファイルに格納されたレポートを .rdf またはテンプレート (.tdf ファイル) に変換
- データベースに格納されたライブラリを .pld または .pll ファイルに変換
- .pld ファイルに格納されたライブラリをデータベース・ライブラリまたは .pll ファイルに変換
- .pll ファイルに格納されたライブラリを .pld ファイルのデータベース・ライブラリに変換

注意： 連結ライブラリ付きのレポートを変換する場合、レポートに連結された .pll ファイルを変換してから、.rdf/.rex ファイルを変換する必要があります。

- OracleAS Portal にレポートをバッチ登録する PL/SQL スクリプトの作成

rwconverter によって、変換処理の一部として、レポートの PL/SQL が自動的にコンパイルされる場合もあります。変換先が .rex ファイルでない場合、次の状況では、rwconverter によって PL/SQL が自動的にコンパイルされます。

- 変換先が .rep ファイルの場合。コンパイルでエラーが発生すると、rwconverter によりエラー・メッセージが表示され、.rep ファイルは作成されません。
- ソースとして .rex ファイルを使用する場合。コンパイルでエラーが発生すると、rwconverter により警告が表示されますが、変換処理は続行します。
- ソースとして別のプラットフォーム上で作成されたレポートを使用する場合。コンパイルでエラーが発生すると、rwconverter により警告が表示されますが、変換処理は続行します。

その他の状況では、各自でレポートの PL/SQL をコンパイルする必要があります (Reports Builder の「プログラム」→「コンパイル」→「すべて」を使用します)。

注意: フォントは、変換時ではなく、Reports Builder や Reports Runtime でレポートを開くときにマップされます。

rwconverter とともに使用できるキーワードは、表 A-1 を参照してください。

例:

```
rwconverter scott/tiger@mydb stype=rdffile source=inven1.rdf
dtype=xmlfile dest=inven1_new.xml
```

A.2.6 rwservlet

rwservlet (Reports Servlet) は、Web サーバーまたは J2EE コンテナ (OC4J など) と Reports Server との間で情報を変換および送信します。これにより、Web ブラウザからレポートを動的に実行できるようになります。オプションでインプロセス・サーバーを使用することもできます。これは、クライアントから最初のリクエストを受信するたびにサーバーを自動的に起動する手段が提供されるので、Reports Server のメンテナンスと管理を軽減できます。

注意: rwservlet を使用して JSP を実行する場合、rwservlet に適用可能なキーワードをすべて使用できます。rwservlet による JSP 実行の詳細は、第 13 項「レポート・リクエストの実行」を参照してください。

注意: キーワード SHOWENV、SHOWJOBS、SHOWMAP、SHOWMYJOBS、KILLJOBID、KILLENGINE、PARSEQUERY、DELAUTH、GETJOBID および GETSERVERINFO は、keyword=value のペアではなくコマンドです。つまり、対応する値を指定せずに、コマンドだけを入力します。

rwservlet とともに使用できるキーワードは、表 A-1 を参照してください。

例

次の例では、myias.mycomp.com は、Oracle Application Server インスタンスであり、7779 は rwservlet が実行されるポートです。

例 1: ペーパー・レポートをブラウザ (キャッシュ) で実行

```
http://myias.mycomp.com:7779/reports/rwservlet?server=myrepserver+report=test.rdf+userid=scott/tiger@mydb+desformat=pdf+DESTYPE=cache
```

例 2: レポート出力をファイルに送信

```
http://myias.mycomp.com:7779/reports/rwservlet?server=myrepserver+report=test.rdf+userid=scott/tiger@mydb+desformat=pdf+DESTYPE=file+DESNAME=c:¥mydir¥test
```

例 3: レポート出力をプリンタに送信

```
http://myias.mycomp.com:7779/reports/rwservlet?server=myrepserver+report=test.rdf+userid=scott/tiger@mydb+desformat=pdf+DESTYPE=printer+DESNAME=myprinter
```

例 4: レポート出力を電子メールで送信

```
http://myias.mycomp.com:7779/reports/rwservlet?server=myrepserver+report=
test.rdf
f+userid=scott/tiger@mydb+desformat=pdf+DESTYPE=mail+DESNAME="emp1@co
mp.com,emp2@comp.com"+CC="emp3@comp.com"+BCC="mgr@comp.com"+
REPLYTO="me@comp.com"+FROM=me@comp.com"
```

例 5: レポート出力を WebDAV (いずれかの WebDAV サーバーまたは OracleAS Portal WebDAV) に送信

```
http://myias.mycomp.com:7779/reports/rwservlet?server=myrepserver+report=
test.rdf
f+userid=scott/tiger@mydb+desformat=htmlcss+DESTYPE=webdav+DESNAME="h
ttp://myusername:mypassword@mywebdavserv.com/mydir/test.html"
```

例 6: レポート出力を OracleAS Portal に送信

```
http://myias.mycomp.com:7779/reports/rwservlet?server=myrepserver+report=
test.rdf
f+userid=scott/tiger@mydb+destype=oracleportal+desformat=PDF+PAGEGROU
P=
mypagegrp+OUTPUTPAGE=reports_
output+ITEMTITLE=pushtoportal+STATUSPAGE=result
```

例 7: XML PDS レポート出力をファイルに送信

```
http://myias.mycomp.com:7779/reports/rwservlet?server=myrepserver+report=
myxmlpdstest.rdf+destype=file+desformat=PDF+DESNAME=c:¥mydir¥my.pdf
```

例 8: JDBC PDS レポート出力をファイルに送信

```
http://myias.mycomp.com:7779/reports/rwservlet?server=myrepserver+report=
myjdb
cpdstest.rdf+destype=file+desformat=PDF+desname=c:¥mydir¥myxml.pdf+
P_JDBCPDS=sybusersybpwd@server1.mydomain.com:1300
```

例 9: レポート出力を複数の宛先に配布

```
http://myias.mycomp.com:7779/reports/rwservlet?server=myrepserver+report=
test.rdf+userid=scott/tiger@mydb+DISTRIBUTE=yes+DESTINATION=c:¥mydistr
ibute.xml
```

例 10: スケジュールされたレポートの実行

```
http://myias.mycomp.com:7779/reports/rwservlet?server=myrepserver+repor
t=test.rdf+SCHEDULE="every first fri of month from 15:53 Oct 23, 1999
retry 3 after 1 hour"+destype=file+desformat=pdf+desname=test.pdf
```

例 11: 保護された Reports Server の使用

```
http://myias.mycomp.com:7779/reports/rwservlet?server=myrepserver+report=
test.rdf+userid=scott/tiger@mydb+desformat=pdf+destype=file+desname=tes
t.pdf+AUTHID=myadmin/myadmin
```


例 12: キー・ファイルの使用

```
http://myias.mycomp.com:7779/report/rwservlet?key1
```

ここでは

key1 は、cgicmd.dat ファイルで定義されたキーです (キー名は最初のパラメータです)。

または

```
http://myias.mycomp.com:7779/report/rwservlet?server=myrepserver+userparam=12+CMDKEY=keyname
```

例 13: レポートをパラメータ・フォームで実行

```
http://myias.mycomp.com:7779/rwservlet?server=myrepserver+report=test.rdf+userid=scott/tiger@mydb+destype=cache+desformat=htmlcss+PARAMFORM=html
```

例 14: レポートを電子メールによる通知で実行

```
http://myias.mycomp.com:7779/rwservlet?server=myrepserver+report=test.rdf+user
```

```
id=scott/tiger@mydb+destype=file+desformat=pdf+desname=test.pdf+
```

```
NOTIFYSUCCESS="emp@comp.com"+NOTIFYFAILURE="admin@comp"
```

例 15: URL エンジンでフェッチする URL を指定するレポートの実行

```
http://myias.mycomp.com:7779/rwservlet?server=myrepserver+report=test.rdf+user
```

```
id=scott/tiger@mydb+destype=file+desformat=pdf+desname=test.pdf+JOBTYPE=rw
```

```
url+URLPARAMETER="http://www.oracle.com"
```

例 16: トレースを有効化してレポートを実行

```
http://myias.mycomp.com:7779/rwservlet?server=myrepserver+report=test.rdf+userid=scott/tiger@mydb+destype=file+desformat=pdf+desname=test.pdf+TRACEOPTS=trace_prf
```

例 17: サーバー myrepserver の環境情報の表示

```
http://myias.mycomp.com:7779/reports/rwservlet/SHOWENV?server=myrepserver+authid=myrepuser/myreppassword
```

例 18: サーバー myrepserver の過去のジョブ情報の表示

```
http://myias.mycomp.com:7779/reports/rwservlet/SHOWJOBS?server=myrepserver+authid=myrepuser/myreppassword+queuetype=past
```

例 19: cgicmd.dat キー・マッピングの表示

```
http://myias.mycomp.com:7779/reports/rwservlet/SHOWMAP?authid=myrepuser/myreppassword
```

例 20: ユーザー myrepuser の現行ジョブ情報の表示

```
http://myias.mycomp.com:7779/reports/rwservlet/SHOWMYJOBS?server=myrepserver+authid=myrepuser/myreppassword+queuetype=current
```

例 21: ジョブ ID 30 のジョブのステータスの取得

```
http://myias.mycomp.com:7779/reports/rwservlet/SHOWJOBID30?server=myrepserver+authid=myrepuser/myreppassword
```

例 22: ジョブ ID 122 の現在実行中のジョブの取消し

`http://myias.mycomp.com:7779/reports/rwservlet/KILLJOBID122?server=myreps
server+authid=myrepuser/myreppassword`

例 23: コマンドの解析済問合せの表示

`http://myias.mycomp.com:7779/reports/rwservlet/PARSEQUERY?server=myreps
server+authid=myrepuser/myreppassword+report=test.rdf+userid=scott/tiger
@db+destype=cache+desformat=htmlcss`

例 24: DB 認証ページの表示

`http://myias.mycomp.com:7779/reports/rwservlet/SHOWAUTH?server=myreps
server+authid=myrepuser/myreppassword+authtype=D`

例 25: rwservlet によって設定された Cookie の削除

`http://myias.mycomp.com:7779/reports/rwservlet/DELAUTH?server=myreps
server+authid=myrepuser/myreppassword`

例 26: サーバー myrepserver からのジョブ ID 87 のジョブの出力の取得

`http://myias.mycomp.com:7779/reports/rwservlet/GETJOBID87?server=myreps
server+authid=myrepuser/myreppassword`

例 27: サーバー myrepserver のサーバー情報の表示

`http://myias.mycomp.com:7779/reports/rwservlet/GETSERVERINFO?server=my
repserver+authid=myrepuser/myreppassword`

例 28: サーバー myrepserver のエンジン rwEng-1 の強制終了

`http://myias.mycomp.com:7779/reports/rwservlet/KILLENGINE1?type=rwEng+
server=myrepserver+authid=myrepuser/myreppassword`

A.2.7 rwcgi

rwservlet と同様に、rwcgi (Common Gateway Interface: CGI) は、Web サーバーと Reports Server との間で情報を変換および送信します。これにより、Web ブラウザからレポートを動的に実行できるようになります。

注意: Oracle Reports 10g では、Reports CGI (rwcgi) を使用することはお勧めしません (下位互換性を確保するために維持されています)。代わりに、JSP、rwservlet (Reports Servlet) または Web サービスを使用します。

パフォーマンス上の理由から、rwcgi よりも rwservlet を使用することを強くお勧めします。rwcgi では、各リクエストに対して新規プロセスが起動されます。その際、JVM が初期化されるため、多数のレポート・リクエストを実行するとパフォーマンスが低下します。一方、rwservlet は OC4J インスタンスにデプロイされており、サーブレットの機能を利用するため、rwcgi よりも高いパフォーマンスを提供します。

rwcgi とともに使用できるキーワードは、表 A-1 を参照してください。

例

```
http://mywebserver.com1:77792/cgi-bin/rwcgi.exe3?server=myrepserver+report=myrepo.rdf+desname=sample.pdf+desformat=pdf+destype=file
```

```
http://mywebserver.com:7779/cgi-bin/rwcgi.sh4?server=myrepserver+authid5=myrepuser/myreppassword+report=myrepo.rdf+desname=sample.pdf+desformat=pdf+destype=file
```

例 1: ペーパー・レポートをブラウザ（キャッシュ）で実行

```
http://mywebserver.com:7779/cgi-bin/rwcgi.exe?server=myrepserver+report=test.rdf+userid=scott/tiger@mydb+DESFORMAT=pdf+DESTYPE=cache
```

例 2: レポート出力をファイルに送信

```
http://mywebserver.com:7779/cgi-bin/rwcgi.exe?server=myrepserver+report=test.rdf+userid=scott/tiger@mydb+desformat=pdf+DESTYPE=file+DESNAME=c:\mydir\test
```

例 3: レポート出力をプリンタに送信

```
http://mywebserver.com:7779/cgi-bin/rwcgi.exe?server=myrepserver+report=test.rdf+userid=scott/tiger@mydb+desformat=pdf+DESTYPE=printer+DESNAME=myprinter
```

例 4: レポート出力を電子メールで送信

```
http://mywebserver.com:7779/cgi-bin/rwcgi.exe?server=myrepserver+report=test.rdf+userid=scott/tiger@mydb+desformat=pdf+DESTYPE=mail+DESNAME="emp1@comp.com,emp2@comp.com+CC=emp3@comp.com"+BCC="mgr@comp.com"+REPLYTO="me@comp.com+from=me@comp.com"
```

例 5: レポート出力を WebDAV（いずれかの WebDAV サーバーまたは OracleAS Portal WebDAV）に送信

```
http://mywebserver.com:7779/cgi-bin/rwcgi.exe?server=myrepserver+report=test.rdf
+userid=scott/tiger@mydb+desformat=htmlcss+DESTYPE=webdav+DESNAME="http://myusername:mypassword@mywebdavserv.com/mydir/test.html"
```

例 6: レポート出力を OracleAS Portal に送信

```
http://mywebserver.com:7779/cgi-bin/rwcgi.exe?server=myrepserver+report=test.rdf
+userid=scott/tiger@mydb+DESTYPE=oracleportal+desformat=PDF+PAGEGROUP=
mypagegrp+OUTPUTPAGE=reports_
output+ITEMTITLE=pushtoportal+STATUSPAGE=result
```

¹ CGI スクリプトを実行している Web サーバー

² Web サーバー・リスナー・ポート

³ Windows オペレーティング・システムにインストールされている Web サーバー

⁴ UNIX オペレーティング・システムにインストールされている Web サーバー

⁵ 保護されたサーバーのみ

例 7: XML PDS レポート出力をファイルに送信

```
http://mywebserver.com:7779/cgi-bin/rwcgi.exe?server=myrepserver+report=myxmlpdstest.rdf+DESTYPE=file+desformat=PDF+desname=c:¥mydir¥my.pdf
```

例 8: JDBC PDS レポート出力をファイルに送信

```
http://mywebserver.com:7779/cgi-bin/rwcgi.exe?server=myrepserver+report=myjdbc
```

```
pdstest.rdf+destype=file+desformat=PDF+desname=c:¥mydir¥myxml.pdf+
```

```
P_JDBCPS=sybpwd@server1.mydomain.com:1300
```

例 9: レポート出力を複数の宛先に配布

```
http://mywebserver.com:7779/cgi-bin/rwcgi.exe?server=myrepserver+report=test.rdf+userid=scott/tiger@mydb+DISTRIBUTE=yes+DESTINATION=c:¥mydistribute.xml
```

例 10: スケジュールされたレポートの実行

```
http://mywebserver.com:7779/cgi-bin/rwcgi.exe?server=myrepserver+report=test.rdf+SCHEDULE="every first fri of month from 15:53 Oct 23, 1999 retry 3 after 1 hour"+destype=file+desformat=pdf+desname=test.pdf
```

例 11: 保護された Reports Server の使用

```
http://mywebserver.com:7779/cgi-bin/rwcgi.exe?server=myrepserver+report=test.rdf+userid=scott/tiger@mydb+desformat=pdf+destype=file+desname=test.pdf+AUTHID=myadmin/myadmin
```

例 12: キー・ファイルの使用

```
http://mywebserver.com:7779/cgi-bin/rwcgi.exe?key1
```

ここで、key1 は cgicmd.dat ファイルで定義されたキーです（キー名は最初のパラメータです）。

または

```
http://mywebserver.com:7779/cgi-bin/rwcgi.exe?server=myrepserver+userparam=12+CMDKEY=key1
```

*cmdkey とともに使用する場合、URL の任意の位置を指定できます。

例 13: レポートをパラメータ・フォームで実行

```
http://mywebserver.com:7779/cgi-bin/rwcgi.exe?server=myrepserver+report=test.rdf+userid=scott/tiger@mydb+destype=cache+desformat=htmlcss+PARAMFORM=yes
```

例 14: レポートを電子メールによる通知で実行

```
http://mywebserver.com:7779/cgi-bin/rwcgi.exe?server=myrepserver+report=test.rdf
```

```
f+userid=scott/tiger@mydb+destype=file+desformat=pdf+desname=test.pdf+
```

```
NOTIFYSUCCESS="emp@comp.com"+NOTIFYFAILURE="admin@comp.com"
```

例 15: URL エンジンでフェッチする URL を指定するレポートの実行

```
http://mywebserver.com:7779/cgi-bin/rwcgi.exe?server=myrepserver+report
=test.rd
f+userid=scott/tiger@mydb+destype=file+desformat=pdf+desname=test.pdf+
JOBTYPE=rwurl+URLPARAMETER="http://www.oracle.com"
```

例 16: トレースを有効化してレポートを実行

```
http://mywebserver.com:7779/cgi-bin/rwcgi.exe?server=myrepserver+report
=test.rdf+userid=scott/tiger@mydb+destype=file+desformat=pdf+desname=te
st.pdf+TRACEOPTS=trace_prf
```

A.2.8 rwsrver

rwsrver (Reports Server) は、クライアント・リクエストを処理します。この処理では、クライアント・リクエストを、認証と許可の確認、スケジューリング、キャッシュ、および配布 (ブラッガブル出力先への配布など) など、各種サービスに割り当てます。また、Reports Server は、リクエストされたレポートを生成するランタイム・エンジンを起動したり、生成されたレポートを Reports キャッシュから取り出したり、ジョブの準備が完了したことをクライアントに通知したりします。

rwsrver とともに使用できるキーワードは、[表 A-1](#) を参照してください。

例 1 (推奨) :

Oracle Process Manager and Notification Server (OPMN) で構成されている場合は、次のコマンドで Reports Server が起動します。

```
$ORACLE_HOME/opmn/bin/opmnctl startproc ias-component=reports_server_name
$ORACLE_HOME/opmn/bin/opmnctl startproc process-type=reports_server_name
```

OPMN による Reports Server の起動の詳細は、[第 2.1.2 項「Oracle Process Manager and Notification Server からの Reports Server の起動、停止および再起動」](#) を参照してください。

例 2 (非推奨) :

コマンドラインから次のコマンドを使用すると、Reports Server が起動します。

```
rwsrver server=myrepserver batch=yes
```

A.2.9 rwbridge

rwbridge (Oracle Reports ブリッジ) は Reports Server と Reports クライアントが異なるサブネットにあるときに使用します。Reports クライアントは、デフォルトのブロードキャスト・メカニズムを使用して、サーバーを検出します。このメカニズムではサブネット内のみで転送されるパケットが送信されます。Oracle Reports ブリッジは、1つのネットワーク内の2つのサブセットをブリッジできます。これは、Reports Server と Reports Client によってブロードキャストされたパケットを取得し、ブリッジ構成ファイルで構成されたりリモート・ブリッジに転送します。Oracle Reports ブリッジの構成の情報は、[第 3.3.2 項「ブリッジの構成要素 \(bridgeconf.dtd\)」](#) を参照してください。

トラブルシューティングの例と解決策については、[第 D.8 項「Oracle Reports ブリッジの問題の診断」](#) を参照してください。

rwbridge とともに使用できるキーワードは、次のとおりです。

- NAME
- SHUTDOWN
- AUTHID

例 1: ブリッジの開始

UNIX の場合: `rwbridge.sh name=mybridge`

Windows の場合: `rwbridge.bat name=mybridge`

例 2: ブリッジの停止

UNIX の場合: `rwbridge.sh name=mybridge shutdown=immediate
authid=scott/tiger`

Windows の場合: `rwbridge.bat name=mybridge name=mybridge shutdown=immediate
authid=scott/tiger`

関連項目: [第 2.2.2 項「コマンドラインからの Oracle Reports ブリッジの起動と停止」](#)

A.3 コマンドライン・キーワード

この項では、Oracle Reports で使用できる各コマンドライン・キーワードについて説明します。

A.3.1 ACCESSIBLE

表 A-2 では、ACCESSIBLE キーワードを使用できる実行可能ファイルを示します。

表 A-2 ACCESSIBLE キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwsgi	rwserver
○	○	○	×	○	○	×

説明 ACCESSIBLE を使用して、Oracle Reports のアクセシビリティ関連の機能が PDF 出力で有効化されるか (YES) または無効化されるか (NO) を指定します。

構文 ACCESSIBLE={YES|NO}

値

- YES の場合、アクセシビリティの機能が PDF 出力で有効化されます。
- NO の場合、アクセシビリティの機能が PDF 出力で無効化されます。

デフォルト NO

A.3.2 ARRAYSIZE

表 A-3 では、ARRAYSIZE キーワードを使用できる実行可能ファイルを示します。

表 A-3 ARRAYSIZE キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwsgi	rwserver
○	○	○	×	○	○	×

説明 ARRAYSIZE を使用して、Oracle の配列処理で使用するサイズ (KB) を指定します。通常、配列サイズが大きくなるに従って、レポートの実行速度も高まります。

構文 ARRAYSIZE=*n*

値

n には 1 ~ 9999 までの数字を使用します (位取りのカンマは使用しないでください)。Reports Runtime では、レポートでの問合せごとに、指定された KB 数のメモリーを使用できません。

デフォルト 10

使用上の注意 ARRAYSIZE は、コマンドラインから JSP ベースの Web レポートを実行する場合に使用できません。

A.3.3 AUTHID

表 A-4 では、AUTHID キーワードを使用できる実行可能ファイルを示します。

表 A-4 AUTHID キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver	rwbridge
○	○	×	×	○	○	○	○

説明

- AUTHID を使用して、制限付きの Reports Server のユーザーの認証に使用されるユーザー名、およびパスワードを指定します。ユーザー認証によって、レポート・リクエストを行っているユーザーがリクエストされたレポートを実行するアクセス権限を持っていることが保証されます。
- rwbridge の場合、AUTHID を使用して、Oracle Reports ブリッジを停止する権限を持たせるユーザー名とパスワードを指定します。ブリッジ構成ファイルで **identifier** 要素を管理者のユーザー名とパスワードに指定し、ブリッジをセキュリティで保護できます。これにより、管理者のみがブリッジを停止できるようになります。

構文 AUTHID=*username/password*

値

- *username/password* には、OracleAS Portal で生成した有効なユーザー名、およびパスワードを指定します。OracleAS Portal に新しいユーザー・アカウントを生成するときは、データベース管理者に問い合わせてください。
- rwbridge の場合：
username/password は、ブリッジ構成ファイル (*rwbridge_bridgename.conf*) の **identifier** 要素で指定されたユーザー名とパスワードです。

デフォルト なし

使用上の注意

- AUTHID は、コマンドラインから JSP ベースの Web レポートを実行する場合に使用できません。
- Single Sign-On 環境では、Oracle Application Server Single Sign-On Server が認証処理を実行し、AUTHID 内のユーザー名のみが Reports Server に渡されます。

A.3.4 AUTOCOMMIT

表 A-5 では、AUTOCOMMIT キーワードを使用できる実行可能ファイルを示します。

表 A-5 AUTOCOMMIT キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
○	○	○	×	○	○	×

説明 AUTOCOMMIT を使用して、データベースへの変更（CREATE など）をデータベースに自動的にコミットするかどうかを指定します。ある種の非 Oracle データベース（SQL Server など）では、AUTOCOMMIT=YES とする必要があります。

構文 AUTOCOMMIT={YES|NO}

値

- YES の場合、データ変更が自動的にデータベースにコミットされます。
- NO の場合、COMMIT コマンドが実行されるか、データをコミットする PL/SQL コマンドが実行されるまで、データ変更はデータベースにコミットされません。

デフォルト NO

使用上の注意 AUTOCOMMIT は、URL から JSP ベースの Web レポートを実行する場合に使用できます。

A.3.5 BACKGROUND

表 A-6 では、BACKGROUND キーワードを使用できる実行可能ファイルを示します。

表 A-6 BACKGROUND キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
○	×	×	×	○	○	×

説明 BACKGROUND を使用して、サーバー上のレポートを同期で実行するか（NO）非同期で実行するか（YES）を指定できます。

注意： BACKGROUND システム・パラメータは Oracle Reports では使用されません。BACKGROUND は、コマンドラインのみで使用します。

構文 BACKGROUND={YES|NO}

値

- YES の場合、レポートを非同期で実行します。クライアントがコールをサーバーに送信すると、レポート・ジョブが完了するのを待たずに他のプロセスを続行します。クライアント・プロセスが中止されると、そのジョブは取り消されます。
- NO の場合、レポートを同期で実行します。クライアントは、レポートがキューに入れられ、ランタイム・エンジンに割り当てられ、実行されてから完了するまで待ちます。

デフォルト NO

使用上の注意 rwbuilder に BACKGROUND=YES を使用した場合、警告が出力され、キーワードは無視されます。

A.3.6 BATCH

表 A-7 では、BATCH キーワードを使用できる実行可能ファイルを示します。

表 A-7 BATCH キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
×	×	×	○	×	×	○

説明 BATCH は、サーバーを非 UI モードで実行する場合に使用します。BATCH=YES を指定したコマンドラインから実行した場合、ユーザー・インタフェースは表示されません。たとえば、rwserver でこのオプションを使用すると、サーバーをスクリプトおよびリモート・エージェントから実行でき、実行中にサーバー・ダイアログは表示されません。

rwconverter で BATCH=YES を使用すると、レポートまたはライブラリの変換にユーザーが介入しないように、すべての端末入力および出力を抑制します。rwserver で BATCH を使用すると、サーバー・ダイアログ・ボックスがオフ (YES) またはオン (NO) になり、プロセスのメッセージが表示または非表示になります。

構文 BATCH={YES|NO}

値

- YES の場合、すべての端末入力および出力を非表示にします (レポートはバックグラウンドで実行されます)。これは、rwrn の場合、デフォルトです。
- NO の場合、特殊な端末入力および出力を許可します。rwconverter の場合、「変換」ダイアログ・ボックスが表示され、そのダイアログ・ボックスを受け入れると、変換が実行されます。

デフォルト NO

使用上の注意

- BATCH=YES の場合、エラー・メッセージは SYSOUT に送信されます。SYSOUT の詳細は、「DESTTYPE」を参照してください。
- BATCH=YES の場合、PARAMFORM=YES は無効になります。これは、ランタイム・パラメータ・フォームをバッチ・モードで表示する意味がないためです。

A.3.7 BCC

表 A-8 では、BCC キーワードを使用できる実行可能ファイルを示します。

表 A-8 BCC キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
○	○	×	×	○	○	×

説明 BCC を使用して、電子メールの BCC 宛先 (指定された受信者名が他の受信者に見えない (公開されない) 宛先) を指定します。

注意: BCC 宛先とは、指定された受信者名が他の受信者に見えない (公開されない) 宛先をいいます。

構文 BCC=emailid | ("emailid","emailid", ...)

値

emailid には、someone@foo.com 形式の有効な電子メール・アドレスを指定します。

デフォルト なし

使用上の注意

- 複数の電子メール・アドレスを指定するには、アドレスのリストを引用符で囲み、リスト内のアドレスをカンマで区切ります。
- 関連するキーワードは、**BCC**、**CC**、**FROM**、**REPLYTO** および **SUBJECT** です。電子メールの主な宛先を指定するには、**DESNAME** を使用します。
- BCC は、コマンドラインから JSP ベースの Web レポートを実行する場合に使用できます。

A.3.8 BLANKPAGES

表 A-9 では、BLANKPAGES キーワードを使用できる実行可能ファイルを示します。

表 A-9 BLANKPAGES キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwsgi	rwserver
○	○	○	×	○	○	×

説明 BLANKPAGES を使用して、レポートの印刷時に空白ページの印刷をするかどうか指定します。このキーワードは、レポート出力に印刷したくない空白ページがある場合に使用します。

構文 BLANKPAGES={YES|NO}

値

- YES の場合、すべての空白ページを印刷します。
- NO の場合、空白ページを印刷しません。

デフォルト YES

使用上の注意 BLANKPAGES は、論理ページが複数の物理ページ（またはパネル）にわたっていて、すべての空白の物理ページの印刷をしないときに特に便利です。

A.3.9 BUFFERS

表 A-10 では、BUFFERS キーワードを使用できる実行可能ファイルを示します。

表 A-10 BUFFERS キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwsgi	rwserver
○	○	○	×	○	○	×

説明 BUFFERS を使用して、仮想メモリー・キャッシュのサイズを KB で指定します。レポートの実行に十分な領域を確保する一方で、システムのリソースを使い過ぎないように調整する必要があります。

構文 BUFFERS=*n*

値

n には 1 ~ 9999 までの数字を使用します（位取りのカンマや小数点などは使用しないでください）。オペレーティング・システムによっては、上限がこれより低い場合もあります。

デフォルト 640

使用上の注意

- この設定がセッションの途中で変更されても、変更は次のレポートが実行されるまで有効になりません。
- BUFFERS は、コマンドラインから JSP ベースの Web レポートを実行する場合に使用できません。

A.3.10 CACHELOB

表 A-11 では、CACHELOB キーワードを使用できる実行可能ファイルを示します。

表 A-11 CACHELOB キーワードを使用する実行可能ファイル

rwclient	rwrwn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
○	○	○	×	○	○	×

説明 CACHELOB を使用して、取り出した Oracle のラージ・オブジェクトを Reports Server の一時ファイル・ディレクトリにキャッシュするかどうかを指定します。一時ファイル・ディレクトリは、環境変数 REPORTS_TMP、または tempDir プロパティ (Reports Server 構成ファイル server_name.conf の engine 要素のプロパティ) で指定されます。tempDir 設定は REPORTS_TMP 設定よりも優先されます。

構文 CACHELOB={YES|NO}

値

- YES の場合、LOB を一時ファイル・ディレクトリにキャッシュします。
- NO の場合、LOB を一時ファイル・ディレクトリにキャッシュしません。

デフォルト YES

使用上の注意

- このオプションは、コマンドラインでのみ設定できます。
- サーバー上の一時ファイル・ディレクトリの場所に、使用可能なディスク領域が十分でない場合は、この値を NO に設定することをお勧めします。ただし、この値を NO に設定すると、LOB をデータベースから複数回フェッチする必要がある場合があり、それによってパフォーマンスが低下することがあります。
- CACHELOB は、コマンドラインから JSP ベースの Web レポートを実行する場合に使用できます。

A.3.11 CC

表 A-12 では、CC キーワードを使用できる実行可能ファイルを示します。

表 A-12 CC キーワードを使用する実行可能ファイル

rwclient	rwrwn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
○	○	×	×	○	○	×

説明 CC を使用して、電子メールの CC 宛先を指定します。

構文 CC=emailid | ("emailid","emailid", ...)

値

emailid には、someone@foo.com. 形式の有効な電子メール・アドレスを指定します。

デフォルト なし

使用上の注意

- 複数の電子メール・アドレスを指定するには、アドレスのリストを引用符で囲み、リスト内のアドレスをカンマで区切ります。
- 関連するキーワードは、**BCC**、**CC**、**FROM**、**REPLYTO** および **SUBJECT** です。電子メールの主な宛先を指定するには、**DESNAME** を使用します。

A.3.12 CELLWRAPPER

表 A-13 では、CELLWRAPPER キーワードを使用できる実行可能ファイルを示します。

表 A-13 CELLWRAPPER キーワードを使用する実行可能ファイル

rwclient	rwrwn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
○	○	×	×	○	○	×

説明 CELLWRAPPER を使用して、レポート出力のデリミタ付きセルの前後に表示される文字を指定します。

構文 CELLWRAPPER=*value*

値

value には、任意の英数字または英数字の文字列を使用します。

表 A-14 一般的な有効値

値	説明
"	セルの両側に二重引用符が表示されます。
'	セルの両側に一重引用符が表示されます。

表 A-15 予約済の有効値

値	説明
tab	セルの両側にタブが配置されます。
space	セルの両側にシングル・スペースが配置されます。
return	セルの両側に改行が配置されます。
none	セルの囲み文字は使用されません。

表 A-16 ASCII キャラクタ・セットに基づくエスケープ・シーケンスの有効値

値	説明
¥t	セルの両側にタブが配置されます。
¥n	セルの両側に改行が配置されます。

デフォルト なし

使用上の注意

- このキーワードを使用できるのは、DESFORMAT=DELIMITED または DESFORMAT=DELIMITEDDATA と指定しているときのみです。
- セルの囲み文字は実際のデリミタとは異なります。セルの囲み文字は、デリミタ付きデータの周囲に表示される文字を指定します。デリミタは、2つのデータ間の境界または区切り位置を示します。

A.3.13 CMDFILE

表 A-17 では、CMDFILE キーワードを使用できる実行可能ファイルを示します。

表 A-17 CMDFILE キーワードを使用する実行可能ファイル

rwclient	rwrwn	rwbuilder	rwconverter	rwervlet	rwsgi	rwserver
○	○	○	○	×	×	×

説明 CMDFILE を使用して、レポートのコマンドライン・オプションを格納するファイルをコールします。コールするファイルは、.txt またはその他の ASCII 形式ファイルである必要があります。

CMDFILE と cgicmd.dat ファイルの違いは、CMDFILE には1つのレポートに対する1つのコマンドラインを指定できるのに対し、cgicmd.dat ファイルには、複数のレポートに対して、キーによって識別される複数のコマンドを格納できる点です。また、CMDFILE キーワードは、コマンドラインで他の引数とともに使用できますが、cgicmd.dat に関連したキー引数を使用する場合、コマンドラインで使用するはその引数のみです。

CMDFILE キーワードを使用すると、実行コマンドを呼び出すたびに多くのオプションを指定しなくても、レポートを実行できます。

構文 CMDFILE=*filename*

値

filename には、任意の有効なコマンド・ファイル名を指定します。

デフォルト なし

使用上の注意

- rwervlet および rwsgi では、CMDFILE キーワードを使用するのではなく、cgicmd.dat ファイルのキーを参照する CMDKEY キーワードを使用します。
- コマンド・ファイルは、別のコマンド・ファイルを参照できます。
- コマンド・ファイルで指定するコマンドライン構文は、コマンドラインで使用される構文と同じです。
- コマンドラインに入力した値は、コマンド・ファイルで指定した値を上書きします。たとえば、コマンドラインから rwclient を COPIES=1 および CMDFILE=RUNONE (コマンド・ファイル) として指定するとします。一方 RUNONE ファイルには COPIES の値として 2 が指定されています。この場合、コマンドラインで指定した COPIES の値 (1) は、RUNONE ファイルで指定した COPIES の値 (2) よりも優先されます。この場合、生成されるレポートのコピーは1つのみです。
- このキーワードの値は、オペレーティング・システムによって異なります。

A.3.14 CMDKEY

表 A-18 では、CMDKEY キーワードを使用できる実行可能ファイルを示します。

表 A-18 CMDKEY キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwsgi	rwserver
×	×	×	×	○	×	×

説明 CMDKEY を使用して、cgicmd.dat ファイル内の、キーによって識別されるコマンドラインをコールします。次に例を示します。

`http://your_webserver/reports/rwervlet?cmdkey=key& ...`

構文 CMDKEY=key

値

key には、cgicmd.dat ファイルで指定されたコマンドラインに関連付けられたキーの名前を指定します。

デフォルト なし

使用上の注意

- CMDKEY を rwervlet で使用する際、コマンドラインでは順不同で使用できます (URL の場合は、疑問符の後に使用します)。rwervlet には、CMDKEY とともに他のコマンドライン・キーワードも使用できます。
- CMDKEY は、コマンドラインから JSP ベースの Web レポートを実行する場合に使用できません。

A.3.15 COLLATE

表 A-19 では、COLLATE キーワードを使用できる実行可能ファイルを示します。

表 A-19 COLLATE キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwsgi	rwserver
○	○	×	×	○	○	×

説明 COLLATE を使用して、レポートをプリンタに出力するときに部単位で印刷する動作を制御します。

たとえば、3 ページの文書を 3 部印刷する場合に、COLLATE を YES に設定すると、次のように出力されます。

1 2 3 | 1 2 3 | 1 2 3

文書はこのページ番号の順序で印刷されます。この動作は、「印刷」ダイアログ・ボックスの「部単位で印刷」チェック・ボックスを選択した場合と同じ動作です。

3 ページの文書を 3 部印刷する場合に、COLLATE を NO に設定すると、次のように出力されます。

1 1 1 | 2 2 2 | 3 3 3

構文 COLLATE={YES|NO}

値

- YES の場合、プリンタへの出力時に部単位で印刷されます。
- NO の場合、プリンタへの出力時に部単位で印刷されません。

デフォルト YES

A.3.16 CONTAINSHTMLTAGS

表 A-20 では、CONTAINSHTMLTAGS キーワードを使用できる実行可能ファイルを示します。

表 A-20 CONTAINSHTMLTAGS キーワードを使用する実行可能ファイル

rwclient	rwrund	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
○	○	○	×	○	○	×

説明 Oracle Reports 10g リリース 2 (10.1.2) では、テキスト書式拡張機能が導入されました。これにより、オブジェクトの「HTML タグを含む」プロパティを「はい」に設定した場合、HTML フォーマット・タグの定義済セットを使用して、テキスト・スタイル（太字、イタリック、下線および取消し線）とテキスト属性（フォント名、フォントの色およびフォント・サイズ）をフォーマットし、Oracle Reports でサポートされているすべてのビットマップ出力フォーマットで、フォーマット済テキスト・オブジェクトを生成できます。

CONTAINSHTMLTAGS を使用して、Oracle Reports で、サポートされているすべての出力フォーマットで、HTML フォーマット・タグが解釈されるかどうかを指定します。

構文 CONTAINSHTMLTAGS=YES|NO

値

- YES の場合、Oracle Reports では、「HTML タグを含む」プロパティが「はい」に設定されているすべてのオブジェクトで、HTML フォーマット・タグが解釈されます。
- NO の場合、Oracle Reports では、オブジェクトの「HTML タグを含む」プロパティの設定に関係なく、そのレポートに対して HTML フォーマット・タグは解釈されません。出力が HTML および HTMLCSS の場合は、ブラウザによって HTML フォーマット・タグが解釈されます。それ以外の出力フォーマットの場合は、HTML タグ自体がそのままの形でレポート出力に表示されます。

デフォルト YES

使用上の注意

- サポートされている出力フォーマットは、PDF、RTF、HTML、HTMLCSS、スプレッドシートおよび PostScript です。
- インライン HTML タグの解釈は、Oracle Reports とブラウザで異なる場合があります。したがって、Oracle Reports 6i、Oracle9i Reports または Oracle Reports 10g (9.0.4) でインライン HTML タグを使用して設計されたレポートは、Oracle Reports 10g リリース 2 (10.1.2) では異なる HTML または HTMLCSS 出力を生成することがあります。Oracle Reports で HTML フォーマット・タグが解釈されないようにして、以前のリリースの動作を保持する場合は、REPORTS_CONTAINSHTMLTAGS 環境変数を NO に設定します。
- REPORTS_CONTAINSHTMLTAGS 環境変数を NO に設定した場合でも、選択したレポートに対してコマンドラインで CONTAINSHTMLTAGS=YES を指定して、サポートされているすべての出力フォーマットに対して Oracle Reports が HTML フォーマット・タグを解釈するようにすることもできます。つまり、このコマンドライン・キーワードで指定する値は、REPORTS_CONTAINSHTMLTAGS 環境変数よりも優先されます。

A.3.17 CONTAINSOLE

注意： Oracle Reports 10g (9.0.4) では、OLE のサポートは廃止されました (OLE は、Web ベースの環境では適用できないクライアント / サーバー機能です)。かわりに、MIME タイプを、関連付けられたプラグインとハイパーリンクとともに使用します。

表 A-21 では、CONTAINSOLE キーワードを使用できる実行可能ファイルを示します。

表 A-21 CONTAINSOLE キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
○	○	○	×	○	○	×

説明 下位互換性のために、CONTAINSOLE を使用して、レポートのプログラム・ユニットまたは連結ライブラリに OLE (Object Linking and Embedding) コールが含まれているかどうかを指定します。CONTAINSOLE=YES の場合、OLE システムはレポート実行開始時に初期化され、レポート実行終了時に終了します。

構文 CONTAINSOLE=YES|NO

値

- YES の場合、レポートのプログラム・ユニットまたは連結ライブラリに OLE コールが含まれています。
- NO の場合、レポートのプログラム・ユニットまたは連結ライブラリに OLE コールは含まれていません。

デフォルト NO

A.3.18 CONTENTAREA

表 A-22 では、CONTENTAREA キーワードを使用できる実行可能ファイルを示します。

表 A-22 CONTENTAREA キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
○	○	×	×	○	○	×

説明 CONTENTAREA を使用して、レポート出力の配信先となる Oracle9iAS Portal リリース 1 のコンテンツ領域を指定します。このキーワードは、Oracle9iAS Portal リリース 1 に対する下位互換性を確保するために維持されています。Oracle WebDB リリース 2.2 に対する下位互換性は、「[SITENAME](#)」を参照してください。OracleAS Portal 10g (9.0.4) 以降では、[PAGEGROUP](#) を使用してください。

構文 CONTENTAREA=name

値

name には、有効な Oracle9iAS Portal リリース 1 コンテンツ領域の名前 (内部名) を指定します。

デフォルト なし

使用上の注意

- Oracle Reports 出力を Oracle9iAS Portal リリース 1 に配信する際、このキーワードの使用は必須です。
- CONTENTAREA 名には、表示名ではなく内部名を使用する必要があります。内部名は Oracle9iAS Portal リリース 1 でのコンポーネント・インスタンスの一意の識別に使用されます。
- 関連するキーワードは、[CONTENTAREA*](#)、[EXPIREDAYS](#)、[ITEMTITLE](#)、[OUTPUTFOLDER*](#)、[OUTPUTPAGE](#)、[PAGEGROUP](#)、[SITENAME*](#)、[STATUSFOLDER*](#)、[STATUSPAGE](#) です。

* Oracle9iAS Portal リリース 1 および Oracle WebDB リリース 2.2 に対する下位互換性を確保するために維持されています。

A.3.19 COPIES

表 A-23 では、COPIES キーワードを使用できる実行可能ファイルを示します。

表 A-23 COPIES キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
○	○	×	×	○	○	×

説明 COPIES を使用して、印刷するレポート出力の部数を指定します。

構文 COPIES=*n*

値

n には、1 ~ 9999 までの整数を使用します（位取りのカンマや小数点などは使用しないでください）。

デフォルト COPIES パラメータの初期値プロパティから取得します（初期値は、設計時に Reports Builder で定義されています）。

使用上の注意

- DESTYPE が PRINTER でない場合、このキーワードは無視されます。
- ランタイム・パラメータ・フォーム上で COPIES が空白のままになっている場合は、デフォルトの 1 が使用されます。

A.3.20 CUSTOMIZE

表 A-24 では、CUSTOMIZE キーワードを使用できる実行可能ファイルを示します。

表 A-24 CUSTOMIZE キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
○	○	×	○	○	○	×

説明 CUSTOMIZE を使用して、現在のレポートに対して実行する Oracle Reports XML ファイルを指定します。XML ファイルには、レポート定義をなんらかの形で変更する、カスタマイズされた設定（レイアウトやデータ・モデルの変更など）が入っています。

構文 CUSTOMIZE=*filename.xml* | (*filename1.xml*, *filename2.xml*, ...)

値

filenamen.xml には、有効な XML レポート定義を含むファイル名またはファイル名のリストを指定します。必要に応じて、ファイル名の前にパス情報を付加します (REPORTS_PATH レジストリまたは **engine** 要素の SourceDir プロパティで指定されているパスにファイルが置かれていない場合は、パスを付加する必要があります)。

注意： XML カスタマイズ・ファイルを使用してレポートを実行時にカスタマイズする方法の詳細は、第 16 章「XML によるレポートのカスタマイズ」を参照してください。

デフォルト なし

使用上の注意

- 通常、XML レポート定義のファイル拡張子には .xml を付けますが、CUSTOMIZE キーワードを使用するときは、その拡張子を付ける必要はありません。
- CUSTOMIZE は、コマンドラインから JSP ベースの Web レポートを実行する場合に使用できます。
- Microsoft Internet Explorer では、URL の戻りストリームの MIME タイプが無視され、URL を参照し MIME タイプが設定される場合があります。このため、URL で CUSTOMIZE を最後のキーワードとして指定した場合に問題が発生することがあります。

```
...REPORT=emp.rdf CUSTOMIZE=c:¥myreports¥emp.xml
```

この場合、URL が拡張子 .xml で終わるため、Internet Explorer では、URL の戻りストリームが実際は HTML であったとしても、XML として処理されます。結果として、ブラウザ・エラーが発生します。この問題を回避するには、URL の最後に一般的なファイル拡張子を使用しないようにします。前述の例では、URL で REPORT パラメータと CUSTOMIZE パラメータの位置を入れ替えることができます。

A.3.21 DATEFORMATMASK

表 A-25 では、DATEFORMATMASK キーワードを使用できる実行可能ファイルを示します。

表 A-25 DATEFORMATMASK キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
○	○	×	×	○	○	×

説明 DATEFORMATMASK を使用して、日付の値がデリミタ付きのレポート出力でどのように表示されるかを指定します。

構文 DATEFORMATMASK=*mask*

値

mask には、任意の日付書式マスクを指定します。

デフォルト なし

使用上の注意

- このキーワードを使用できるのは、DESFORMAT=DELIMITED または DESFORMAT=DELIMITEDDATA と指定しているときのみです。

注意: 有効な DATEFORMATMASK 値は、Oracle Reports のオンライン・ヘルプのトピック「DATE および TIME 書式マスク構文」を参照してください。

- DATEFORMATMASK は、コマンドラインから JSP ベースの Web レポートを実行する場合に使用できます。

A.3.22 DELAUTH

表 A-26 では、DELAUTH キーワードを使用できる実行可能ファイルを示します。

表 A-26 DELAUTH キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
×	×	×	×	○	○	×

説明 DELAUTH を使用して、rwervlet または rwcgi のユーザー ID Cookie を削除します。

構文 `http://your_webserver/reports/rwervlet/delauth[?] [server=server_name] [&authid=username/password]`

値 構文を参照してください。

デフォルト なし

使用上の注意

- このキーワードは値を必要としないコマンドです。つまり、対応する値を持たず、コマンドだけが入力されます。
- 関連するキーワードは、[SERVER](#) および [AUTHID](#) です。

A.3.23 DELIMITED_HDR

表 A-27 では、DELIMITED_HDR キーワードを使用できる実行可能ファイルを示します。

表 A-27 DELIMITED_HDR キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
○	○	×	×	○	○	×

説明 DELIMITED_HDR を使用して、DESFORMAT=DELIMITED または DESFORMAT=DELIMITEDDATA によるレポート実行の際に、レポート・ヘッダーなどのボーラプレート・テキストをオフにします。

構文 `DELIMITED_HDR={YES|NO}`

値

- YES の場合、デリミタ付き出力ファイルのボーラプレート・テキストをそのまま残します。
- NO の場合、デリミタ付き出力ファイルのボーラプレート・テキストをすべてオフにします。

デフォルト YES

使用上の注意 このキーワードを使用できるのは、DESFORMAT=DELIMITED または DESFORMAT=DELIMITEDDATA と指定しているときのみです。

A.3.24 DELIMITER

表 A-28 では、DELIMITER キーワードを使用できる実行可能ファイルを示します。

表 A-28 DELIMITER キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
○	○	×	×	○	○	×

説明 DELIMITER を使用して、レポート出力のセルを区切るために使用する文字を指定します。

構文 DELIMITER=*value*

値

value には、次のような任意の英数字または英数字の文字列を使用します。

表 A-29 一般的な有効値

値	説明
,	各セルをカンマで区切ります。
.	各セルをピリオドで区切ります。

予約値には次のようなものがあります。

表 A-30 予約済の有効値

値	説明
tab	各セルをタブで区切ります。
space	各セルをスペースで区切ります。
return	各セルを改行で区切ります。
none	デリミタを使用しません。

表 A-31 ASCII キャラクタ・セットに基づくエスケープ・シーケンスの有効値

値	説明
¥t	各セルをタブで区切ります。
¥n	各セルを改行で区切ります。

デフォルト Tab

使用上の注意 このキーワードを使用できるのは、DESFORMAT=DELIMITED または DESFORMAT=DELIMITEDDATA と指定しているときのみです。

A.3.25 DESFORMAT

表 A-32 では、DESFORMAT キーワードを使用できる実行可能ファイルを示します。

表 A-32 DESFORMAT キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
○	○	×	×	○	○	×

説明 DESFORMAT を使用して、DESTYPE=FILE および DESNAME=*filename* が指定されている場合にレポートのフォーマットに使用するレポートの出力フォーマットまたはプリンタ定義を指定します。

構文 DESFORMAT=*desformat*

値 長さが 1K を超えない任意の有効な宛先フォーマットです。このキーワードの有効値の例は、表 A-33 を参照してください。

表 A-33 DESFORMAT の有効値

値	説明
DFLT	レポート出力は、デフォルトのプリンタ・ドライバを使用してレポートをフォーマットするファイルに送信されます (たとえば、PostScript ドライバは PostScript 出力フォーマットを生成します)。
DELIMITED	このレポート出力は、標準のスプレッドシート・ユーティリティ (Microsoft Excel など) で読み取ることのできるファイルに出力されます。DELIMITER キーワードを使用して、デリミタを指定していない場合、デフォルトのデリミタとしてタブが使用されます。次の「 使用上の注意 」を参照してください。
DELIMITEDDATA	DELIMITED と同様の機能ですが、DESFORMAT=DELIMITED を指定して大量のレポートを実行すると問題が発生する場合に使用します。次の「 使用上の注意 」を参照してください。
HTML	このレポート出力は、HTML フォーマットのファイルに出力されます。次の「 使用上の注意 」を参照してください。
HTMLCSS	このレポート出力は、スタイル・シート拡張を含むファイルに出力されます。次の「 使用上の注意 」を参照してください。
PDF	このレポート出力は、PDF 形式のファイルに出力され、Adobe Acrobat などの PDF ビューアで読み取ることができます。PDF 出力は、現在システムに構成されているプリンタに基づきます。プリンタへの出力には、現在選択されているプリンタのドライバが使用されるため、レポートを実行しているマシンにプリンタを構成している必要があります。
<i>printer definition</i>	DESTYPE=FILE および DESNAME= <i>filename</i> でレポートをフォーマットする際に使用するプリンタ定義。 MODE=BITMAP の場合、これはプリンタ名です。DFLT の値は、デフォルトのプリンタ・ドライバを使用することを意味します。 MODE=CHARACTER の場合、これはプリンタ定義ファイル (.prt ファイル) の名前です。たとえば、hpl、hplwide、dec、decwide、decland、dec180、dflt、wide などがあります。有効なプリンタ定義のリストは、システム管理者におたずねください。
RTF	このレポート出力は、ワード・プロセッサ (Microsoft Word など) で読み取ることのできるファイルに出力されます。Microsoft Word でファイルをオープンする場合、レポート内のすべての図形およびオブジェクトを表示するには、「 表示 」→「 ページレイアウト 」を選択する必要があります。次の「 使用上の注意 」を参照してください。

表 A-33 DESFORMAT の有効値 (続き)

値	説明
SPREADSHEET	(コマンドラインのみ) レポート出力は HTML ファイルに送信され、それは Microsoft Excel 2000 で直接開くことができます。スプレッドシート出力は、どのフォーマット (.rdf、.jsp、.xml) で保存されたレポートのペーパー・レイアウトからでも生成できます。次の「 使用上の注意 」を参照してください。
XML	このレポート出力は、XML ファイルとして保存されます。このレポートは、XML をサポートしているブラウザ、あるいは任意の XML 表示アプリケーションでオープンしたり読み取ることができます。

デフォルト DESFORMAT システム・パラメータの初期値プロパティから取得します (設計時に Reports Builder で定義されています)。Reports Builder でレポートを実行するときに DESFORMAT が空白または DFLT である場合、現行のプリンタ・ドライバ (「[ファイル](#)」 → 「[プリンタ](#)」で指定) が使用されます。プリンタ名が選択されていない場合、デフォルトで PostScript 出力フォーマットが使用されます。

使用上の注意

- このキーワードの値は、使用しているオペレーティング・システムによっては、大文字と小文字の区別が必要な場合があります。
- DESFORMAT=HTML または DESFORMAT=HTMLCSS が指定されている場合、スペースは ` ` に置き換わります。このデフォルトの動作では、数値を右揃えにする位置合せの設定は解除されます。HTML および HTMLCSS 出力で空白を ` ` に置き換えない場合、REPORTS_NO_HTML_SPACE_REPLACE を YES に設定する必要があります。その場合、HTML および HTMLCSS 出力時の DELIMITER コマンドライン・キーワードの機能は排除されます (DESFORMAT=DELIMITED の場合は、DELIMITER は有効です)。

- DESFORMAT=DELIMITED は、.dst ファイルではサポートされません。その場合、Oracle Reports では、次のエラーが表示されます。

REP-34305: destid= 'DEST1' のキーワードの設定が無効です。

DELIMITED 機能では、[DELIMITER](#)、[CELLWRAPPER](#)、[NUMBERFORMATMASK](#) および [DATEFORMATMASK](#) コマンドライン・キーワードも使用できます。

- DESFORMAT=DELIMITEDDATA の場合、DelimitedData ドライバはレポート・データ・モデルで実行されず、XML ドライバとほぼ同じ動作をします。ドライバがデータ・モデルで実行されないため、レイアウトで定義されたフォーマットは DelimitedData 出力に反映されません。

次の列プロパティを設定して列名を変更し、列を DelimitedData 出力ファイルから除外できます。

- 「XML タグ」プロパティを使用して列の別名を入力できます。
- 「XML 出力から除外」プロパティを使用して列を DelimitedData 出力から除外できます。

DELIMITEDDATA 機能では、DELIMITED と同様に、[DELIMITER](#)、[CELLWRAPPER](#)、[NUMBERFORMATMASK](#) および [DATEFORMATMASK](#) コマンドライン・キーワードも使用できます。

デリミタ付き出力の詳細は、Oracle Reports のオンライン・ヘルプのデリミタ付き出力に関する項 (および『Oracle Reports レポート作成ガイド』の「[拡張概念](#)」の章) を参照してください。

- DESFORMAT=SPREADSHEET の場合、レポート出力では、色、フォント、条件付き書式、グラフ、イメージなどのリッチ・レイアウト・フォーマットが保持されます。DESFORMAT=SPREADSHEET で実行されるレポートで様々なレポート・オブジェクトがどのように生成されるかの詳細は、Oracle Reports のオンライン・ヘルプのスプレッドシート出力に関する項 (および『Oracle Reports レポート作成ガイド』の「[拡張概念](#)」の章) を参照してください。

- Oracle Reports で生成された RTF 出力を日本語版 Microsoft Word 95 で開く場合、ダッシュが正しく表示されないなどの出力の問題が生じる場合があります。この問題は Microsoft Word 95 固有の問題であり、日本語版 Microsoft Word 97 では発生しません。

A.3.26 DESNAME

表 A-34 では、DESNAME キーワードを使用できる実行可能ファイルを示します。

表 A-34 DESNAME キーワードを使用する実行可能ファイル

rwclient	rwrund	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
○	○	×	×	○	○	×

説明 DESNAME を使用して、レポート出力の送信先のキャッシュ、ファイル、プリンタ、WebDAV サーバー、または電子メール ID (または配布リスト) を指定します。

構文 DESNAME=*desname*

値

desname には、1KB を超えない長さの任意の有効なキャッシュ宛先、ファイル名、プリンタ名、電子メール ID または WebDAV サーバーを指定します。プリンタ名には、オプションでポートを指定できます。次に例を示します。

DESNAME=printer,LPT1:DESNAME=printer,FILE:

デフォルト DESNAME パラメータの初期値プロパティから取得します (初期値は、設計時に Reports Builder で定義されています)。DESTYPE=FILE で、かつ DESNAME が空の文字列である場合は、実行時にデフォルトの `reportname.lis` になります。

使用上の注意

- このキーワードの値は、使用しているオペレーティング・システムによっては、大文字と小文字の区別が必要な場合があります。
- レポート出力を電子メールで送信するには、電子メール ID を通常の電子メール・アプリケーション (任意の SMTP 互換アプリケーション) の場合と同様に指定します。複数のユーザー名を指定する際は、スペースを入れずにカンマで区切ります。次に例を示します。

name, name, name

- 場合によっては、このキーワードがオペレーティング・システムによって無効にされることもあります。

例

例 1: レポート出力をファイルに送信

```
rwrund report=test.rdf userid=scott/tiger@mydb desformat=pdf destype=file
desname=c:¥mydir¥test.pdf
```

```
http://myias.mycomp.com:7779/reports/rwservlet?server=myreperv+report=test.rdf+
userid=scott/tiger@mydb+desformat=pdf+destype=file+desname=c:¥mydir¥test.pdf
```

```
http://mywebserver.com:7779/cgi-bin/rwcgi.exe?server=myreperv+report=test.rdf+
userid=scott/tiger@mydb+desformat=pdf+destype=file+desname=c:¥mydir¥test.pdf
```

```
rwclient server=myreperv report=test.rdf userid=scott/tiger@mydb desformat=pdf
destype=file desname=c:¥mydir¥test.
```

例 2: レポート出力をプリンタに送信

```
rwrun report=test.rdf userid=scott/tiger@mydb desformat=pdf destype=printer
desname=myprinter
```

```
http://myias.mycomp.com:7779/reports/rwservlet?server=myrepserver+report=test.rdf+
userid=scott/tiger@mydb+desformat=pdf+destype=printer+desname=myprinter
```

```
http://mywebserver.com:7779/cgi-bin/rwcgi.exe?server=myrepserver+report=test.rdf+
userid=scott/tiger@mydb+desformat=pdf+destype=printer+desname=myprinter
```

```
rwclient server=myrepserver report=test.rdf userid=scott/tiger@mydb desformat=pdf
destype=printer desname=myprinter
```

例 3: レポート出力を電子メールで送信

```
rwrun report=test.rdf userid=scott/tiger@mydb desformat=pdf destype=mail
desname="emp1@comp.com, emp2@comp.com" cc="emp3@comp.com" bcc="mgr@comp.com"
replyto="me@comp.com" from="me@comp.com"
```

```
http://myias.mycomp.com:7779/reports/rwservlet?server=myrepserver+report=test.rdf+
userid=scott/tiger@mydb+desformat=pdf+destype=mail+
desname="emp1@comp.com, emp2@comp.com"+cc="emp3@comp.com"+bcc="mgr@comp.com"+
replyto="me@comp.com"+from="me@comp.com"
```

```
http://mywebserver.com:7779/cgi-bin/rwcgi.exe?server=myrepserver+report=test.rdf+
userid=scott/tiger@mydb+desformat=pdf+destype=mail+
desname="emp1@comp.com, emp2@comp.com"+cc="emp3@comp.com"+bcc="mgr@comp.com"+
replyto="me@comp.com"+from="me@comp.com"
```

```
rwclient server=myrepserver report=test.rdf userid=scott/tiger@mydb desformat=pdf
destype=mail desname="emp1@comp.com, emp2@comp.com" cc="emp3@comp.com"
bcc="mgr@comp.com" replyto="me@comp.com" from="me@comp.com"
```

例 4: レポート出力を WebDAV (いずれかの WebDAV サーバーまたは OracleAS Portal WebDAV) に送信

注意: 現在、Reports Builder 環境からの FTP および WebDAV 宛先へのサポートはありません。ただし、Reports Runtime および Reports Server 環境からはサポートされています。

```
rwrun report=test.rdf userid=scott/tiger@mydb desformat=htmlcss destype=webdav
desname="http://myusername:mypassword@mywebdavserv.com/mydir/test.html"
```

```
http://myias.mycomp.com:7779/reports/rwservlet?server=myrepserver+report=test.rdf+
userid=scott/tiger@mydb+desformat=htmlcss+destype=webdav+
desname="http://myusername:mypassword@mywebdavserv.com/mydir/test.html"
```

```
http://mywebserver.com:7779/cgi-bin/rwcgi.exe?server=myrepserver+report=test.rdf+
userid=scott/tiger@mydb+desformat=htmlcss+destype=webdav+
desname="http://myusername:mypassword@mywebdavserv.com/mydir/test.html"
```

```
rwclient server=myrepserver report=test.rdf userid=scott/tiger@mydbdesformat=htmlcss
destype=webdav
desname="http://myusername:mypassword@mywebdavserv.com/mydir/test.htm"
```


A.3.27 DEST

表 A-35 では、DEST キーワードを使用できる実行可能ファイルを示します。

表 A-35 DEST キーワードを使用する実行可能ファイル

rwclient	rwrund	rwbuilder	rwconverter	rservlet	rwsgi	rwserver
×	×	×	○	×	×	×

説明 DEST を使用して、変換後のレポートまたはライブラリの名前を指定します。

構文 DEST={dest_name| (dest_name1, dest_name2, ...)|pathname}

値

dest_name には、有効なレポート名、ライブラリ名またはファイル名、あるいは名前をカンマで区切り、括弧で囲んだそれらのリスト ((qanda, text, dmast) など) を指定します。

デフォルト DEST キーワードを指定しない場合、rwconverter は次のデフォルト名を使用します。

- DTYPE が PLDFILE のとき、DEST のデフォルト名は source.pld です。
- DTYPE が PLLFILE のとき、DEST のデフォルト名は source.pll です。
- DTYPE が RDIFFILE のとき、DEST のデフォルト名は source.rdf です。
- DTYPE が REPPFILE のとき、DEST のデフォルト名は source.rep です。
- DTYPE が REXFILE のとき、DEST のデフォルト名は source.rex です。
- DTYPE が XMLFILE のとき、DEST のデフォルト名は source.xml です。
- DTYPE が REGISTER のとき、DEST のデフォルト名は SQL*Plus スクリプト出力ファイルの名前 (output.sql など) です。

使用上の注意

- ファイル名のレポート / ライブラリ名のリストは、各エントリをカンマで区切り、括弧で囲む必要があります。次に例を示します。
(qanda, test, dmast) または (qanda, test, dmast)
- 存在するソース名よりも指定した宛先名が多い場合、余分な宛先名は無視されます。存在するソース名よりも指定した宛先名が少ない場合、宛先名がなくなるとデフォルト名が使用されます。
- DEST キーワードの値は、オペレーティング・システムによって異なります。
- DTYPE=REGISTER のとき、複数の宛先は不要です。DEST に対して複数の SQL*Plus スクリプト・ファイル名を指定した場合、最初のエントリのみが認識され、その他は無視されません。

A.3.28 DESTINATION

表 A-36 では、DESTINATION キーワードを使用できる実行可能ファイルを示します。

表 A-36 DESTINATION キーワードを使用する実行可能ファイル

rwclient	rwrund	rwbuilder	rwconverter	rservlet	rwsgi	rwserver
○	○	×	×	○	○	×

説明 DESTINATION キーワードを使用すると、レポートの現行の実行処理に対応する配布を定義する XML ファイルの名前を指定できます。

注意： XML ベースの配布ファイルには、.xml 拡張子を付ける必要があります。

構文 DESTINATION=*filename.xml*

値

filename.xml には、レポートあるいはレポートのある部分の配布を定義する XML ファイルの名前を指定します。

デフォルト なし

使用上の注意

- DESTINATION キーワードを使用可能にするには、コマンドラインで **DISTRIBUTE=YES** と指定する必要があります。これら両方のキーワードを指定した場合に、**DESTYPE**、**DESNAME** および **DESFORMAT** は指定しても無視されます。
- Microsoft Internet Explorer では、URL の戻りストリームの MIME タイプが無視され、URL を参照し MIME タイプが設定される場合があります。これは、レポートの配布を定義するときに問題となる場合があります。それは、URL の最後が DESTINATION キーワードになることがあるためです。次に例を示します。

```
...DISTRIBUTE=yes DESTINATION=c:¥oracle¥reports¥dist¥mydist.xml
```

この場合、URL が拡張子 .xml で終わるため、Internet Explorer では、URL の戻りストリームが実際は HTML であったとしても、XML として処理されます。結果として、ブラウザ・エラーが発生します。この問題を回避するには、URL の最後に一般的なファイル拡張子を使用しないようにします。前述の例では、URL で DISTRIBUTE パラメータと DESTINATION パラメータの位置を入れ替えることができます。

注意： 拡張配布の作成の詳細は、[第 15 章「拡張配布の作成」](#)を参照してください。

A.3.29 DESTYPE

[表 A-37](#) では、DESTYPE キーワードを使用できる実行可能ファイルを示します。

表 A-37 DESTYPE キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwsgi	rwserver
○	○	×	×	○	○	×

説明 DESTYPE を使用して、ペーパーベースのレポートのレポート出力を受け取るデバイス・タイプを指定できます。Oracle Reports Destination API で独自のプラグブル・デスティネーションを作成した場合は、作成した宛先をこのようにしてコールします。

構文

DESTYPE={CACHE|LOCALFILE|FILE|PRINTER|SYSOUT|MAIL|ORACLEPORTAL|FTP|WEBDAV|*name_of_pluggable_destination*}

値 [表 A-38](#) では、DESTYPE キーワードで使用できる値を説明します。

表 A-38 DESTYPE の有効値

値	説明
CACHE	rwclient、rwsgi および rwervlet に対してのみ使用できます。出力は Reports Server のキャッシュに直接送信されます。

表 A-38 DESTYPE の有効値 (続き)

値	説明
LOCALFILE	rwclient、rwcgi および rwservlet に対してのみ使用できます。出力はクライアント・マシン上のファイルに同期式または非同期式で送信されます。
FILE	出力は DESNAME で指定したサーバー上のファイルに出力されます。
PRINTER	出力は DESNAME で指定したサーバー上のプリンタに送信されます。OracleAS Reports Services が認識できるプリンタをインストールし、稼働させている必要があります。次の「使用上の注意」を参照してください。
SYSOUT	rwcgi にのみ有効です。出力はクライアント・マシンのデフォルトの出力デバイスに送信され、強制的に同期呼出しになります。
MAIL	出力は DESNAME で指定されたメール・ユーザーに送信されます。SMTP を使用する任意のメール・システムにメールを送信できます。 注意: 構成ファイル <i>server_name.conf</i> には、送信メール・サーバー名が含まれている必要があります。これは、Windows と UNIX のどちらの環境にも該当します。
ORACLEPORTAL	rwservlet、rwcgi および rwclient に対して有効です。出力は OracleAS Portal に送信されます。関連するキーワードは、CONTENTAREA*、EXPIREDAYS、ITEMTITLE、OUTPUTFOLDER*、OUTPUTPAGE、PAGEGROUP、SCHEDULE、SITENAME*、STATUSFOLDER*、STATUSPAGE です。 * Oracle9iAS Portal リリース 1 および Oracle WebDB リリース 2.2 に対する下位互換性を確保するために維持されています。 次の「使用上の注意」を参照してください。
FTP	出力は指定された FTP サーバーに送信されます。次の「使用上の注意」を参照してください。
WEBDAV	レポートを直接公開できるように、出力は指定された WebDAV サーバーに送信されます。次の「使用上の注意」を参照してください。
<i>name_of_pluggable_destination</i>	Oracle Reports Destination API で独自のプラグガブル・デスティネーションを作成した場合は、作成した宛先をこのようにしてコールします。

デフォルト DESTYPE システム・パラメータの初期値プロパティから取得します (設計時に Reports Builder で定義されています)。

使用上の注意

- Oracle Reports 10g リリース 2 (10.1.2) では、DESTYPE 値の SCREEN と PREVIEW は無効になりました。Reports Runtime (rwrun) ユーザー・インタフェースが廃止されたためです。Reports Builder では、依然として DESTYPE システム・パラメータを SCREEN に設定してレポートをフォーマットし、スクリーン・フォントを Reports Builder ユーザー・インタフェースのプレビューアに表示できます。
- DESTYPE=PRINTER: Windows では、ハードウェアベースの左マージンはデフォルトで無視されます。印刷は印刷可能領域ではなく、実際の用紙の左上隅 (0, 0) を起点として開始されます。そのため、プリンタ・ハードウェアベースのマージンに依存しないで、レポートを容易に設計できます。Windows でハードウェアベースの左マージンを入れずにレポートを印刷する際には、レポートのレイアウト時に、データが印刷可能領域に表示されるように十分なマージン・スペースを確保する必要があります。「ページ設定」ダイアログの「マージン」フィールドは、OracleAS Reports Services との一貫性を保つために無効になっています。ハードウェア・マージンを入れる従来の動作に戻すには、REPORTS_ADD_HWMARGIN 環境変数を YES に設定します。

- DESTYPE=ORACLEPORTAL: 配布の宛先に ORACLEPORTAL は使用できません。かわりに、DESTYPE=WEBDAV を使用して、XML ベースの拡張配布を OracleAS Portal に対し実行できます。OracleAS Portal インスタンスが WebDAV 対応になっている必要があります。WebDAV を有効にする方法の詳細は、OracleAS Portal のオンライン・ヘルプを参照してください。

OracleAS Portal への配布に WebDAV を使用する方法の詳細は、Oracle MetaLink (<http://metalink.oracle.com>) の「Note 241821.1」の「How to Send and Distribute Reports 9i Output to Oracle Portal?」を参照してください。

注意: DESTYPE=ORACLEPORTAL コマンドライン・キーワードは、rwrn 実行可能ファイルとともに使用できません。この宛先は、rwservlet、rwclient または rwcgi に対してのみ使用できます。

Oracle Reports 出力を OracleAS Portal に配信する前に、以下を作成しておく必要があります。

- 1つ以上の項目の領域が入った有効な [OUTPUTPAGE](#)
- 1つ以上の項目の領域が入った有効な [PAGEGROUP](#)

また、Reports Server 構成ファイルを次のように編集する必要があります。

1. destype=oraclePortal 要素のコメントを外します。

```
<destination
  destype="oraclePortal"
  class="oracle.reports.server.DesOraclePortal">
  <!--property name="portalUserId"
    value="%PORTAL_DB_USERNAME%/%PORTAL_DB_
      PASSWORD%@%PORTAL_DB_TNSNAME%"
    confidential="yes"
    encrypted="no"/-->
</destination>
```

注意: Oracle9i Reports リリース 2 (9.0.2) では、デフォルトで portalUserId プロパティのコメントが外され、プロパティの接続文字列がインフラストラクチャ・データベースを示します。レポート出力を OracleAS Portal に配信できるようにするには、portalUserId はコメントを外したままにする必要があります。

Oracle Reports 10g では、デフォルトで portalUserId はコメントアウトされています。Reports Server によって接続文字列が判別され、レポートが OracleAS Portal に配信されます。このコメントを外す必要があるのは、異なる OracleAS Portal インスタンスを使用しているときのみです。

2. Oracle Reports 出力をデフォルトの OracleAS Portal インスタンスに配信しない場合は、portalUserId プロパティの値を OracleAS Portal 接続情報と置き換えます。

注意: 値を置き換えない、または destype エントリのコメントを外さない場合は、次のようなエラー・メッセージが表示されます。

```
REP-56092: No class defined for destination type
oracleportal
```

リクエストの実行方法は他のデフォルトの宛先指定と同様です。次に例を示します。

```
http://your_server:port/reports/rwservlet?
report=test.rdf&userid=scott/tiger@reportal&authid=
pushportal/trial&destype=oracleportal&desformat=PDF&pagegroup
=PORTAL_REPORTS&outputpage=reports_output&itemtitle=
pushtportal&statuspage=result
```

- DESTYPE=FTP: リクエストの実行方法は他のデフォルトのプラグブル宛先指定と同様です。ファイル名とともに完全な FTP の URL 位置を指定する必要があります。FTP サーバーに認証が必要な場合は、それも含めて次のように URL で指定する必要があります。

```
http://your_
server:port/reports/rwservlet?report=rep.jsp&destype=FTP&desname=ft
p://user:pwd@ftpServer/dir/myreport.pdf&desformat=pdf
```

この例では、DESTYPE は FTP で、DESNAME 値は完全な FTP の URL 位置で、レポート名は myreport.pdf です。

プロキシ・サーバーを使用するには、サーバー構成ファイルの destination 要素（デフォルトの FTP 構成）を次のように編集します。

```
<destination destype="ftp"
class="oracle.reports.plugin.destination.ftp.DesFTP"> <!--property
name="proxy" value="proxyinfo.xml"/--> </destination>
```

プロキシ情報を指定するには、デフォルトの場所（ORACLE_HOME¥reports¥conf）にある proxyinfo.xml ファイルを編集します。次に、サーバー構成ファイルの proxy プロパティのコメントを外し、value には proxyinfo.xml ファイルへの完全なパスを指定します。

たとえば、ORACLE_HOME が D:¥oracle にある場合は、proxyinfo.xml のデフォルトの場所は次のように指定します。

```
<destination destype="ftp"
class="oracle.reports.plugin.destination.ftp.DesFTP"> <property
name="proxy" value="D:¥¥oracle¥¥reports¥¥conf¥¥proxyinfo.xml"/>
</destination>
```

注意： FTP 宛先に指定されたプロキシ・サーバーでは、SOCKS プロトコルがサポートされている必要があります。この確認は、初期化中に実行されます。プロキシ・サーバーで SOCKS プロトコルがサポートされていない場合、サーバーで次のエラーが発生します。

```
REP-62352: FTP Proxy Server specified is not responding
```

- DESTYPE=WEBDAV: リクエストの実行方法は他のデフォルトのプラグブル宛先指定と同様です。ファイル名とともに完全な WebDAV の URL 位置を指定する必要があります。WebDAV サーバーに認証が必要な場合は、それも含めて次のように URL で指定する必要があります。

```
http://your_
server:port/reports/rwservlet?report=rep.jsp&destype=webdav&desname
=http://user:pwd@webdavserver/myreport.pdf&desformat=pdf
```

この例では、DESTYPE は WEBDAV で、DESNAME 値は完全な WebDAV の URL 位置で、レポート名は myreport.pdf です。

プロキシ・サーバーを使用するには、サーバー構成ファイルの `destination` 要素（デフォルトの WebDAV 構成）を次のように編集します。

```
<destination destype="webdav"
class="oracle.reports.plugin.destination.webdav.DesWebDAV">
<!--property name="proxy" value="proxyinfo.xml"/--> </destination>
```

プロキシ情報を指定するには、デフォルトの場所（`ORACLE_HOME¥reports¥conf`）にある `proxyinfo.xml` ファイルを編集します。次に、サーバー構成ファイルの `proxy` プロパティのコメントを外し、`value` には `proxyinfo.xml` ファイルへの完全なパスを指定します。

たとえば、`ORACLE_HOME` が `D:¥reports` にある場合は、`proxyinfo.xml` のデフォルトの場所は次のように指定します。

```
<destination destype="webdav"
class="oracle.reports.plugin.destination.webdav.DesWebDAV">
<property name="proxy"
value="D:¥¥reports¥¥reports¥¥conf¥¥proxyinfo.xml"/>
</destination>
```

例

例 1: ペーパー・レポートをブラウザ（キャッシュ）で実行

```
http://myias.mycomp.com:7779/reports/rwservlet?server=myrepserve+report=test.rdf+
userid=scott/tiger@mydb+desformat=pdf+destype=cache
```

```
http://mywebserver.com:7779/cgi-bin/rwsgi.exe?server=myrepserve+report=test.rdf+
userid=scott/tiger@mydb+desformat=pdf+destype=cache
```

```
rwclient server=myrepserve report=test.rdf userid=scott/tiger@mydb desformat=pdf
destype=cache
```

例 2: レポート出力をファイルに送信

```
rwrun report=test.rdf userid=scott/tiger@mydb desformat=pdf destype=file
desname=c:¥mydir¥test.pdf
```

```
http://myias.mycomp.com:7779/reports/rwservlet?server=myrepserve+report=test.rdf+
userid=scott/tiger@mydb+desformat=pdf+destype=file+desname=c:¥mydir¥test.pdf
```

```
http://mywebserver.com:7779/cgi-bin/rwsgi.exe?server=myrepserve+report=test.rdf+
userid=scott/tiger@mydb+desformat=pdf+destype=file+desname=c:¥mydir¥test.pdf
```

```
rwclient server=myrepserve report=test.rdf userid=scott/tiger@mydb desformat=pdf
destype=file desname=c:¥mydir¥test.pdf
```

例 3: レポート出力をプリンタに送信

```
rwrun report=test.rdf userid=scott/tiger@mydb desformat=pdf destype=printer
desname=myprinter
```

```
http://myias.mycomp.com:7779/reports/rwservlet?server=myrepserve+report=test.rdf+
userid=scott/tiger@mydb+desformat=pdf+destype=printer+desname=myprinter
```

```
http://mywebserver.com:7779/cgi-bin/rwsgi.exe?server=myrepserve+report=test.rdf+
userid=scott/tiger@mydb+desformat=pdf+destype=printer+desname=myprinter
```

```
rwclient server=myrepserve report=test.rdf userid=scott/tiger@mydb desformat=pdf
destype=printer desname=myprinter
```

例 4: レポート出力を電子メールで送信

```
rwrun report=test.rdf userid=scott/tiger@mydb desformat=pdf destype=mail
desname="emp1@comp.com, emp2@comp.com" cc="emp3@comp.com" bcc="mgr@comp.com"
replyto=me@comp.com from="me@comp.com"
```

```
http://myias.mycomp.com:7779/reports/rwservlet?server=myreperv+report=test.rdf+
userid=scott/tiger@mydb+desformat=pdf+destype=mail+
desname="emp1@comp.com, emp2@comp.com"+cc="emp3@comp.com"+bcc="mgr@comp.com"+
replyto="me@comp.com"+from="me@comp.com"
```

```
http://mywebserver.com:7779/cgi-bin/rwcgi.exe?server=myreperv+report=test.rdf+
userid=scott/tiger@mydb+desformat=pdf+destype=mail+
desname="emp1@comp.com, emp2@comp.com"+cc="emp3@comp.com"+bcc="mgr@comp.com"+
replyto="me@comp.com"+from="me@comp.com"
```

```
rwclient server=myreperv report=test.rdf userid=scott/tiger@mydb desformat=pdf
destype=mail desname="emp1@comp.com, emp2@comp.com" cc="emp3@comp.com"
bcc="mgr@comp.com" replyto="me@comp.com" from="me@comp.com"
```

例 5: レポート出力を WebDAV (いずれかの WebDAV サーバーまたは OracleAS Portal WebDAV) に送信

```
rwrun report=test.rdf userid=scott/tiger@mydb desformat=htmlcss destype=webdav
desname="http://myusername:mypassword@mywebdavserv.com/mydir/test.html"
```

```
http://myias.mycomp.com:7779/reports/rwservlet?server=myreperv+report=test.rdf+
userid=scott/tiger@mydb+desformat=htmlcss+destype=webdav+
desname="http://myusername:mypassword@mywebdavserv.com/mydir/test.html"
```

```
http://mywebserver.com:7779/cgi-bin/rwcgi.exe?server=myreperv+report=test.rdf+
userid=scott/tiger@mydb+desformat=htmlcss+destype=webdav+
desname="http://myusername:mypassword@mywebdavserv.com/mydir/test.html"
```

```
rwclient server=myreperv report=test.rdf userid=scott/tiger@mydb
desformat=htmlcss destype=webdav
desname="http://myusername:mypassword@mywebdavserv.com/mydir/test.html"
```

例 6: レポート出力を OracleAS Portal に送信

```
rwrun report=test.rdf userid=scott/tiger@mydb destype=oracleportal desformat=PDF
pagegroup=mypagegrp outputpage=reports_output itemtitle=pushtportal
statuspage=result
```

```
http://myias.mycomp.com:7779/reports/rwservlet?server=myreperv+report=test.rdf+
userid=scott/tiger@mydb+destype=oracleportal+desformat=PDF+pagegroup=mypagegrp+
outputpage=reports_output+itemtitle=pushtportal+statuspage=result
```

```
http://mywebserver.com:7779/cgi-bin/rwcgi.exe?server=myreperv+report=test.rdf+
userid=scott/tiger@mydb+destype=oracleportal+desformat=PDF+pagegroup=mypagegrp+
outputpage=reports_output+itemtitle=pushtportal+statuspage=result
```

```
rwclient server=myreperv report=test.rdf userid=scott/tiger@mydb
destype=oracleportal desformat=PDF pagegroup=mypagegrp outputpage=reports_output
itemtitle=pushtportal statuspage=result
```

A.3.30 DISTRIBUTE

表 A-39 では、DISTRIBUTE キーワードを使用できる実行可能ファイルを示します。

表 A-39 DISTRIBUTE キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwsgi	rwserver
○	○	×	×	○	○	×

説明 DISTRIBUTE を使用すると、レポート出力の宛先としてレポート配布定義（設計時に Reports Builder で定義します）あるいは配布 XML ファイルで定義されている配布リスト上の複数の宛先への配布を有効化または無効化できます。

構文 DISTRIBUTE={YES|NO}

値

- YES の場合、レポートを配布リストにある宛先に配布します。
- NO の場合、配布リストを無視して、[DESNAME](#)、[DESTYPE](#) および [DESFORMAT](#) パラメータの指定に従ってレポートを出力します。これは基本的に、配布用のレポート設定を実行するデバッグ・モードで、配布を実際には実行しません。

デフォルト NO

使用上の注意 DISTRIBUTE キーワードは、[DESTINATION](#) キーワードと密接に関連して動作します。[DESTINATION](#) キーワードが効果を発揮するには、DISTRIBUTE に YES を指定する必要があります。これら両方のキーワードを指定した場合に、[DESTYPE](#)、[DESNAME](#) および [DESFORMAT](#) は指定しても無視されます。

注意： 拡張配布の作成の詳細は、[第 15 章「拡張配布の作成」](#)を参照してください。

A.3.31 DTYPE

表 A-40 では、DTYPE キーワードを使用できる実行可能ファイルを示します。

表 A-40 DTYPE キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwsgi	rwserver
×	×	×	○	×	×	×

説明 DTYPE を使用して、レポートまたはライブラリの変換後のフォーマットを指定します。

構文

DTYPE={PLDFILE|PLLFILE|RDFFILE|REPFIL|REXFILE|TDFFILE|XMLFILE|JSPFILE|REGISTER}

値 次の値が適用されます。

- PLDFILE の場合、変換された PL/SQL ライブラリは、ASCII 形式のファイルに格納されます。
- PLLFILE の場合、変換された PL/SQL ライブラリは、ソース・コードおよび P コード（コンパイル済 PL/SQL）を含むファイルに格納されます。
- RDFFILE の場合、変換されたレポートは、1 つ以上のレポート定義ファイル（.rdf 拡張子を持つファイル）に格納されます。

- REPFILFILE の場合、変換されたレポートは、1 つ以上のバイナリ実行ファイル (.rep 拡張子を持つファイル) に格納されます。
- REXFILE の場合、変換されたレポートは、1 つ以上のテキスト・ファイル (.rex 拡張子を持つファイル) に格納されます。
- TDFFILE の場合、レポートはテンプレート・ファイル (.tdf 拡張子を持つファイル) に変換されます。
- XMLFILE の場合、変換されたレポートは、XML ファイル (.xml 拡張子を持つファイル) に格納されます。
- JSPFILE の場合、変換されたレポートは、JSP ファイル (.jsp 拡張子を持つファイル) に格納されます。
- REGISTER の場合は、SOURCE によって指定された各レポートを RWWWVREG.REGISTER_REPORT 関数で OracleAS Portal にロードするためのスクリプト・ファイルが作成されます。各ロード関数には、OracleAS Portal にレポートを登録するために必要な情報が入力されます。作成された SQL*Plus のスクリプト・ファイルを Oracle Application Server DB プロバイダに対して実行すると、複数のレポートを OracleAS Portal にバッチ登録できます。詳細は、付録 C 「OracleAS Portal へのレポートのバッチ登録」を参照してください。

デフォルト REPFILFILE

使用上の注意

- `rwconverter` を使用して rep ファイルを作成する場合、ソース・レポートの PL/SQL は、自動的にコンパイルされます。コンパイルでエラーが発生すると、エラー・メッセージが表示され、.rep ファイルは作成されません。この問題を回避するために、.rep ファイルを作成する前に、必ず Reports Builder で「プログラム」→「コンパイル」を使用してソース・レポートの PL/SQL をコンパイルしてください。
- レポートをテンプレートに変換する場合、レポートのヘッダー・セクション、トレーラ・セクションおよびマージン領域にあるオブジェクトのみがテンプレートに使用されます。メイン・セクションにあるオブジェクトは無視されます。

A.3.32 DUNIT

表 A-41 では、DUNIT キーワードを使用できる実行可能ファイルを示します。

表 A-41 DUNIT キーワードを使用する実行可能ファイル

rwclient	rwrnrun	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
×	×	×	○	×	×	×

説明 DUNIT を使用して、レポートを変換する宛先のサイズの単位を指定できます。これを指定する場合、DUNIT は、SOURCE レポートのサイズの単位と異なっている必要があります。これを指定しない場合、SOURCE レポートのサイズの単位が使用されます。

構文 DUNIT={CENTIMETER|CHARACTER|INCH|POINT}

値

- CENTIMETER の場合、変換されたレポートは、サイズの単位として最初にセンチメートルを使用します。
- CHARACTER の場合、変換されたレポートは、サイズの単位として最初に文字を使用します。
- INCH の場合、変換されたレポートは、サイズの単位として最初にインチを使用します。
- POINT の場合、変換されたレポートは、サイズの単位として最初にポイントを使用します。

デフォルト Null (レポートのサイズの単位が使用されます)。

A.3.33 ENGINERESPONSETIMEOUT

表 A-42 では、ENGINERESPONSETIMEOUT キーワードを使用できるコマンドを示します。

表 A-42 ENGINERESPONSETIMEOUT キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwsgi	rwserver
○	×	×	×	○	○	×

説明 ENGINERESPONSETIMEOUT を使用すると、現行環境でのレポートの実行時に、エンジンがジョブのステータスを更新する最大時間（分）を指定できます。ジョブのステータスを更新する時間がなんらかの理由（エンジンがハングする、または SQL 問合せが長時間ブロックされるなどの理由）でこの最大時間を超えた場合、Reports Server はジョブを終了します。

このパラメータは、Reports Server 構成ファイルの engine 要素の engineResponseTimeOut 属性より優先されます。engine 要素の詳細は、第 3.2.1.4 項「engine」を参照してください。

構文 ENGINERESPONSETIMEOUT=*number*

値

number には、分数（5 など）を指定します。

デフォルト なし

A.3.34 ENVID

表 A-43 では、ENVID キーワードを使用できるコマンドを示します。

表 A-43 ENVID キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwsgi	rwserver
○	×	×	×	○	○	×

説明 ENVID を使用して、現行のジョブ・リクエストに必要な環境を指定します。第 3.2.2 項「動的な環境切替え」の説明のように、このキーワードによって環境の動的な切替えが可能になります。

構文 ENVID=*id*

値

id には、構成ファイルの environment 要素 *id* に対応する識別子を指定します。一致する environment 要素によって、現行のジョブ・リクエストに使用される環境変数が定義されます。例は、第 3.2.2 項「動的な環境切替え」を参照してください。

デフォルト なし

A.3.35 EXPIRATION

表 A-44 では、EXPIRATION キーワードを使用できるコマンドを示します。

表 A-44 EXPIRATION キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwsgi	rwserver
○	×	×	×	○	○	×

説明 EXPIRATION を使用して、レポート出力が削除されるまでキャッシュにとどまることのできる期間を定義します。

重複ジョブの検出の詳細は、第 13.11 項「キャッシュからのレポート出力の再使用」を参照してください。maxQueueSize および cacheSize 値に関連するチューニングの考慮事項は、第 20.2 項「Reports Server の構成のチューニング」と第 20.8 項「レポートの実行」を参照してください。

構文 EXPIRATION=*time_string*

値

time_string には、次のいずれかの形式を使用できます。

- *n{unit}* は、オプションの単位を持つ数値の場合に使用します。unit には、minute(s)、hour(s) または day(s) を指定できます。単位を指定しない場合は、デフォルトの minute(s) が使用されます。
- {*Mon DD, YYYY*} *hh:mi:ss* am|pm {*timezone*} が、日付 / 時刻の書式として使用されます。日付情報はオプションです。日付を指定しない場合は、本日の日付が使用されます。タイムゾーンもオプションです。タイムゾーンを指定しない場合は、Reports Server のタイムゾーンが使用されます。日付 / 時刻の書式には必ず米国のロケールを使用します。この形式は、Java DateFormat.MEDIUM タイプの定義と同じです。

デフォルト なし

A.3.36 EXPIREDAYS

表 A-45 では、EXPIREDAYS キーワードを使用できる実行可能ファイルを示します。

表 A-45 EXPIREDAYS キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rservlet	rwcgi	rwserver
×	×	×	×	○	○	×

説明 EXPIREDAYS を使用して、OracleAS Portal に送信された Oracle Reports 出力が期限切れになるまでの日数を指定できます。

構文 EXPIREDAYS={PERMANENT|1 day|2 days|3 days|7 days|14 days|31 days|60 days|90 days|120 days}

値

- PERMANENT は、期限がないことを示します。
- *n days* は、*n* 日後に期限が切れることを示します。

デフォルト なし

使用上の注意

- Oracle Reports 出力を OracleAS Portal に配信する際、このキーワードの使用はオプションです。
- 関連するキーワードは、CONTENTAREA*、EXPIREDAYS、ITEMTITLE、OUTPUTFOLDER*、OUTPUTPAGE、PAGEGROUP、SITENAME*、STATUSFOLDER*、STATUSPAGE です。

* Oracle9iAS Portal リリース 1 および Oracle WebDB リリース 2.2 に対する下位互換性を確保するために維持されています。

A.3.37 EXPRESS_SERVER

表 A-46 では、EXPRESS_SERVER キーワードを使用できる実行可能ファイルを示します。

表 A-46 EXPRESS_SERVER キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rservlet	rwcgi	rwserver
○	○	○	×	○	○	×

説明 EXPRESS_SERVER を使用して、接続する OLAP Server を指定できます。

構文 EXPRESS_SERVER="server=server/domain=domain/user=userid/password=passwd"

RAM とともに使用する場合の構文 EXPRESS_SERVER="server=server/domain=domain/user=userid/password=passwd/ramuser=ramuserid/rampassword=rampasswd/ramexpressid=ramexpid/ramserverscript=ramsscript/rammasterdb=ramdb/ramconnecttype=reamconn"

値 二重引用符 (") で囲まれた有効な接続文字列を指定します。

表 A-47 接続文字列値

値	説明
サーバー	OLAP Server 文字列 (たとえば ncacn_ip_tcp:olap2-pc/sl=x/st=x/ct=x/sv=x/)。サーバー文字列の詳細は次を参照してください。
domain	OLAP Server のドメイン。
userid	OLAP Server にログインするためのユーザー ID。
passwd	ユーザー ID に対するパスワード。
ramuserid	RDBMS にログインするためのユーザー ID。
rampasswd	RDBMS に対するパスワード。
ramexpid	Oracle Sales Analyzer データベースのユーザー ID。これが必要なのは Oracle Sales Analyzer データベースを使用するときのみです。
ramsscript	サーバー上のリモート・データベースの構成ファイル (RDC) の完全なファイル名 (フルパスを含む)。このファイルは、コードやデータのデータベースの場所などの情報を指定します。UNC (Universal Naming Convention) 構文を使用すると、同じドライバ文字をその場所にマップすることなく、複数のユーザーが同じ接続を使用してデータにアクセスすることを許可できます。UNC 構文は、 <code>¥¥ServerName¥ShareName¥</code> の後に任意のサブフォルダまたはファイルが続きます。
ramdb	最初に付加される Relational Access Manager データベースの名前。データベース・ファイル名以外は指定しないでください。このデータベースは OLAP Server の ServerDBPath にあるパス・リストに記載されているディレクトリに常駐している必要があります。ServerDBPath は、Express Configuration Manager のダイアログ・ボックスの「ファイル I/O」タブで確認できます。
ramconn	Express 接続の種類。直接接続の場合は必ず 0 を指定してください。

パラメータ サーバー値には、Oracle Express Connection Editor で設定され、接続 (XCF) ファイルに格納されている設定値に対応する 4 つのパラメータが含まれています。これら 4 つのパラメータはすべて必須で、任意の順序で指定できます。表 A-48 に、パラメータとその設定を示します。

表 A-48 EXPRESS_SERVER の server 値とともに使用するパラメータの設定

パラメータ	説明	設定
s1	サーバー・ログイン	-2: ホスト (ドメイン・ログイン) -1: ホスト (サーバー・ログイン) 0: 認証不要 1: ホスト (ドメイン・ログイン) と接続セキュリティ 2: ホスト (ドメイン・ログイン) とコール・セキュリティ 3: ホスト (ドメイン・ログイン) とパケット・セキュリティ 4: ホスト (ドメイン・ログイン) と整合性セキュリティ 5: ホスト (ドメイン・ログイン) とプライバシー・セキュリティ 注意: Windows ではすべての設定値を使用します。UNIX システムでは、設定値 0、-1 および -2 のみを使用します。これらの設定値の情報は、Express Connection Editor の Help システムを参照してください。
st	サーバーの種類	:1: OLAP Server
ct	接続の種類	0: Express 接続
sv	サーバー・バージョン	1: Express 6.2 以降

デフォルト なし

使用上の注意

- 文字列の全体が引用符で囲まれるので、必要に応じて (たとえば、ユーザー ID が John Smith である場合など) 文字列にスペースを含めることができます。
- 文字列にスラッシュ (/) が必要なときは、エスケープ文字としてスラッシュをもう 1 つ使用する必要があります。たとえば、ドメインがツールあるいはレポートである場合は、コマンドラインは次のようになります。

`EXPRESS_SERVER="server=ncacn_ip_tcp:olap2-pc/s1=0/
st=1/ct=0/sv=1/domain=tools//reports"`
- 文字列の中では一重引用符を使用できます。文字列全体が二重引用符で囲まれているため、特別な扱いは受けません。

A.3.38 FORMSIZE

表 A-49 では、FORMSIZE キーワードを使用できる実行可能ファイルを示します。

表 A-49 FORMSIZE キーワードを使用する実行可能ファイル

rwclient	rwrwn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
×	×	×	○	×	×	×

説明 FORMSIZE を使用して、変換するレポートのランタイム・パラメータ・フォームを、宛先のサイズの単位で (DUNIT を使用して) 指定できます。

構文 FORMSIZE=*width* x *height*

値

width/height には、指定されたサイズの単位で使用できる任意の値を指定します。

デフォルト なし

使用上の注意

- CHARACTER 以外の **DUNIT** では、端数を指定するのに小数を使用できます (8.5 x 11 など)。
- ランタイム・パラメータ・フォームの詳細は、**PARAMFORM** キーワードを参照してください。

A.3.39 FROM

表 A-50 では、FROM キーワードを使用できる実行可能ファイルを示します。

表 A-50 FROM キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwsgi	rwserver
○	○	×	×	○	○	×

説明 FROM を使用して、電子メール送信者の電子メール・アドレスを指定できます。

構文 FROM=*emailid*

値

emailid には、*someone@foo.com* の形式で有効な電子メール・アドレスを指定します。

デフォルト *loginid@machine_name*

使用上の注意 関連するキーワードは、**BCC**、**CC**、**FROM**、**REPLYTO** および **SUBJECT** です。電子メールの主な宛先を指定するには、**DESNAME** を使用します。

A.3.40 GETJOBID

表 A-51 では、GETJOBID キーワードを使用できる実行可能ファイルを示します。

表 A-51 GETJOBID キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwsgi	rwserver
×	×	×	×	○	○	×

説明 GETJOBID を使用して、ジョブ ID *n* の Reports Server ジョブの結果出力を取得できます。

構文 `http://your_webserver/reports/rwervlet/getjobid
n[?][server=server_name][&authid=username/password]`

値 構文を参照してください。

デフォルト なし

使用上の注意

- このキーワードは値を必要としないコマンドです。つまり、対応する値を持たず、コマンドだけが入力されます。
- ジョブが正常に終了し、Reports Server キャッシュに存在する必要があります。現在のジョブ・リストを参照するには、[SHOWJOBS](#) を使用します。
- 関連するキーワードは、[SERVER](#) および [AUTHID](#) です。

A.3.41 GETSERVERINFO

表 A-52 では、GETSERVERINFO キーワードを使用できる実行可能ファイルを示します。

表 A-52 GETSERVERINFO キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
×	×	×	×	○	×	×

説明 GETSERVERINFO を使用して、Reports Server 情報を表示できます。情報が返される形式 (HTML または XML) は、`statusformat` で選択できます。

構文 `http://your_webserver/reports/rwervlet/getserverinfo[?] [server=server_name] [&authid=username/password] [&statusformat={html|xml}]`

値 構文を参照してください。

デフォルト なし

使用上の注意

- このキーワードは値を必要としないコマンドです。つまり、対応する値を持たず、コマンドだけが入力されます。
- 関連するキーワードは、[SERVER](#) および [AUTHID](#) です。

A.3.42 HELP

表 A-53 では、HELP キーワードを使用できる実行可能ファイルを示します。

表 A-53 HELP キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
×	×	×	×	○	○	×

説明 HELP キーワードを使用して、ヘルプ・トピックを表示し、`rwervlet` コマンドとともに使用できるキーワードを示します。次に例を示します。

`http://your_webserver/reports/rwervlet/help?command=keyword`

構文 `http://yourwebserver/reports/rwervlet/help`

または

`http://your_webserver/reports/rwervlet/help?command=keyword`

値 構文を参照してください。

デフォルト なし

A.3.43 ITEMTITLE

表 A-54 では、ITEMTITLE キーワードを使用できる実行可能ファイルを示します。

表 A-54 ITEMTITLE キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwsgi	rwserver
○	○	×	×	○	○	×

説明 ITEMTITLE を使用して、OracleAS Portal が Oracle Reports 出力に使用する表示名を指定できます。OracleAS Portal にはこの名前が表示され、Oracle Reports 出力にリンクされません。

構文 ITEMTITLE=*title*

値

title には、任意のテキストを指定します。値に文字スペース単位が含まれる場合や、cgicmd.dat ファイルでオプションを指定する場合は、値の前後に引用符を付けます。

デフォルト レポートのファイル名

使用上の注意

- Oracle Reports 出力を OracleAS Portal に配信する際、このキーワードの使用はオプションです。
- 関連するキーワードは、CONTENTAREA*、EXPIREDAYS、ITEMTITLE、OUTPUTFOLDER*、OUTPUTPAGE、PAGEGROUP、SITENAME*、STATUSFOLDER*、STATUSPAGE です。

* Oracle9iAS Portal リリース 1 および Oracle WebDB リリース 2.2 に対する下位互換性を確保するために維持されています。

A.3.44 JOBNAME

表 A-55 では、JOBNAME キーワードを使用できる実行可能ファイルを示します。

表 A-55 JOBNAME キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwsgi	rwserver
○	×	×	×	○	○	×

説明 JOBNAME を使用して、Reports Queue Manager に表示されるジョブの名前を指定できます。これはコメントとして扱われ、ジョブの実行にはまったく関係しません。JOBNAME が指定されていないときは、Reports Queue Manager ではレポート名がジョブ名として表示されます。

構文 JOBNAME=*string*

値

string には、任意のジョブ名を指定します。

デフォルト なし

使用上の注意 JOBNAME は、コマンドラインから JSP ベースの Web レポートを実行する場合に使用できます。

A.3.45 JOBTYP

表 A-56 では、JOBTYP キーワードを使用できる実行可能ファイルを示します。

表 A-56 JOBTYP キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
○	×	×	×	○	○	×

説明 JOBTYP を使用して、サーバーによって実行されるジョブのタイプを記述できます。Reports Server に備わっているエンジンがそのジョブ・タイプを処理できれば、任意のジョブ・タイプを入力できます。

構文 JOBTYP=*jobtype*

値

jobtype には、Reports Server のエンジンが処理するジョブを指定します。たとえば、report (rwEng エンジンの場合) や rwurl (rwURLEng エンジンの場合) を指定します。

デフォルト report

使用上の注意 Oracle Reports で提供されるデータベース認証機能は、JOBTYP=report の場合にのみ使用できます。これは、OracleAS Reports Services に提供されるデフォルトのエンジン (rwEng) のジョブ・タイプです。データベース認証機能は、JOBTYP が別の値 (たとえば、独自で開発したカスタム・エンジンなど) を指定されている場合は使用できません。これは、カスタム・エンジンでは、接続文字列の形式が異なっていることが必要になる場合があるためです。Oracle Reports のデータベース認証機能では、接続文字列はデフォルトのエンジンで使用される Oracle Reports の形式 *user/password@dbname* に限定されます。

A.3.46 JVMOPTIONS

表 A-57 では、JVMOPTIONS キーワードを使用できる実行可能ファイルを示します。

表 A-57 JVMOPTIONS キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
×	○	○	○	×	×	○

説明 JVMOPTIONS を使用して、Java 仮想マシン (JVM) に対するオプションを設定します。

構文

JVMOPTIONS={*options in the Reports Runtime, Reports Builder, Reports Converter, or Reports Server's JVM*}

たとえば、次のようなコマンドラインを使用して、Reports Server (rwserver) を 512MB のヒープ領域で起動できます。

```
rwserver server=servername jvmoptions=-Xmx512M
```

また、次のようなコマンドラインを使用して、Reports Builder (rwbuilder) を 512MB のヒープ領域で起動することもできます。

```
rwbuilder jvmoptions=-Xmx512M
```

複数のオプションを指定する場合、それらを引用符で囲む必要があります。

```
rwserver server=servername jvmoptions="-Xmx256M -Xms=128M"
```

使用上の注意

- デフォルト値 `-Xmx256M` では、JVM ヒープ・サイズが 256MB に設定されており、大きなグラフ付きのレポートやサイズの大きいレポートを実行したときに、エラー・メッセージ `Out of memory.` が表示されないように考慮されています。
- Reports Engine の起動時に、`server_name.conf` ファイルの `engine` 要素にある `jvmoptions` 属性で指定されている JVM オプションがチェックされます。詳細は、[第 3.2.1.4 項「engine」](#) を参照してください。`server_name.conf` で JVM オプションが設定されている場合は、それが `REPORTS_JVM_OPTIONS` 環境変数の値よりも優先されます。`server_name.conf` で JVM オプションが指定されていない場合は、Oracle Reports では、`REPORTS_JVM_OPTIONS` 環境変数で指定された JVM オプションが使用されます。詳細は、[第 B.1.47 項「REPORTS_JVM_OPTIONS」](#) を参照してください。
- Reports Server でレポートを実行する場合、JVM オプションは `REPORTS_JVM_OPTIONS` 環境変数を使用して設定できません。Reports Server の場合、JVM オプションはコマンドラインで `JVMOPTIONS` コマンドライン・キーワードを使用して設定します。
- Reports Builder、Reports Runtime および Reports Converter でレポートを実行する場合、コマンドラインで `JVMOPTIONS` コマンドライン・キーワードを使用して指定された JVM オプションは、`REPORTS_JVM_OPTIONS` 環境変数で指定された JVM オプションよりも優先されます。

A.3.47 KILLENGINE

表 A-58 では、KILLENGINE キーワードを使用できる実行可能ファイルを示します。

表 A-58 KILLENGINE キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwsgi	rwserver
×	×	×	×	○	×	×

説明 KILLENGINE を使用して、指定されたエンジン ID およびエンジン・タイプを持つ Reports Server エンジン を停止します。セキュリティで保護された Reports Server の場合、管理者権限を持つユーザーのみがこのキーワードを使用できます。セキュリティで保護されていない Reports Server の場合は、[AUTHID](#) キーワードのユーザー ID とパスワード値は、`server.conf` 構成ファイルの `identifier` タグで指定されたユーザー ID とパスワードと一致している必要があります。

構文 `http://your_webserver/reports/rwervlet/killengine[?][server=server_name][&authid=username/password][&type=engine_type]`

たとえば、エンジン `rwEng-0` を強制終了するには、次のようにします。

```
http://yourwebserver/reports/rwervlet/killengine0?server=myserver&authid=mydb/password&type=rwEng
```

値 構文を参照してください。

デフォルト なし

使用上の注意

- エンジンが、現在 Reports Server に存在している必要があります。
- `GETSERVERINFO` を使用して、サーバーに存在している現行のエンジンを調べます。
- 関連するキーワードは、[GETSERVERINFO](#)、[SERVER](#) および [AUTHID](#) です。

A.3.48 KILLJOBID

表 A-59 では、KILLJOBID キーワードを使用できる実行可能ファイルを示します。

表 A-59 KILLJOBID キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
×	×	×	×	○	○	×

説明 KILLJOBID を使用して、ジョブ ID *n* の Reports Server ジョブを取り消すことができます。

構文 `http://your_webserver/reports/rwervlet/killjobidn[?] [server=server_name] [&authid=username/password] [&statusformat={html|xml|xmldtd}]`

値 構文を参照してください。

デフォルト なし

使用上の注意

- ジョブは現行である（キューに入っているかスケジュールされている）必要があります。
- 現在のジョブ・リストを参照するには、SHOWJOBS を使用します。STATUSFORMAT は html（デフォルト）、xml または xmldtd に設定できます。指定した形式でステータスが返されます。ステータス情報は、HTML、XML または XMLDTD（DTD の内部サブセット）に生成されます。
- 関連するキーワードは、SHOWJOBS、SERVER、AUTHID および STATUSFORMAT です。
- STATUSFORMAT パラメータは、rwervlet に対してのみ使用でき、rwcgi には使用できません。

A.3.49 LONGCHUNK

表 A-60 では、LONGCHUNK キーワードを使用できる実行可能ファイルを示します。

表 A-60 LONGCHUNK キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
○	○	○	×	○	○	×

説明 LONGCHUNK は、Reports Builder で LONG 列値を取り出す変更分のサイズ (KB) です。LONG の値を検索する際に、メモリー容量の制限から一度にすべてを検索するのではなく、少しずつ検索する方がよい場合があります。LONGCHUNK は、Oracle データベースにのみ適用されます。

構文 LONGCHUNK=*n*

値

n には 1 ~ 9999 までの数字を使用します（位取りのカンマや小数点などは使用しないでください）。オペレーティング・システムによっては、上限がこれより低い場合もあります。

デフォルト 10

使用上の注意 LONGCHUNK は、コマンドラインから JSP ベースの Web レポートを実行する場合に使用できます。

A.3.50 MIMETYPE

表 A-61 では、MIMETYPE キーワードを使用できる実行可能ファイルを示します。

表 A-61 MIMETYPE キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwsgi	rwserver
×	×	×	×	○	○	×

説明 MIMETYPE を使用して、Web の出力が返されるときに、Reports Server によって割り当てられた MIME タイプを上書きできます。通常はデフォルトの MIME タイプは有効ですが、誤っている場合には、このキーワードで無効にできます。次に例を示します。

mimetype=application/vnd.ms-excel

構文 MIMETYPE=*string*

値

string には、有効な MIME タイプを指定します。

デフォルト なし

使用上の注意 OracleAS Reports Services では、MIMETYPE に入力された文字列は検証されません。返されるレポート出力に対して、入力した文字列が有効であることを確認してください。

A.3.51 MODE

表 A-62 では、MODE キーワードを使用できる実行可能ファイルを示します。

表 A-62 MODE キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwsgi	rwserver
○	○	×	×	○	○	×

説明 MODE を使用して、レポートをキャラクタ・モードで実行するか、ビットマップで実行するかを指定できます。これにより、Reports Builder からキャラクタ・モードでレポートを実行できます。

構文 MODE={BITMAP|CHARACTER|DEFAULT}

値

- BITMAP は、レポートをビットマップ・モードで実行します。
- DEFAULT は、現在使用されている実行可能ファイルのモードでレポートを実行します。
- CHARACTER は、Windows で Reports Builder ASCII ドライバを使用し、編集可能な ASCII 出力を生成します。

デフォルト DEFAULT

A.3.52 MODULE|REPORT

表 A-63 では、MODULE|REPORT キーワードを使用できる実行可能ファイルを示します。

表 A-63 MODULE|REPORT キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwsgi	rwserver
○	○	○	×	○	○	×

説明 MODULE または REPORT を使用して、実行するレポートの名前を指定できます。

構文 REPORT|MODULE=*runfile*

値

runfile には、任意の有効な実行ファイル（つまり、拡張子が *.rep*、*.rdf*、*.jsp* または *.xml* のファイル）を指定します。

デフォルト なし

使用上の注意

- キャラクタ・モード・レポートを指定すると、Reports Builder では警告が表示され、8.5 x 11 インチのページ・サイズと 7 x 6 インチのフォーム・サイズを使用してレポートが開きます。
- レポートを実行する場合は（たとえば、ペーパー・デザイン・ビューで表示する場合は）、そのレポートは完全なレポート定義である（独自のデータ・モデルとレイアウト定義を持つ）必要があります。部分レポート定義は実行できません。
- XML レポート定義を MODULE|REPORT キーワードで指定する場合は、*.xml* のファイル拡張子を付ける必要があります。
- ファイルの拡張子を入力しないと、実行モジュールは、ファイルを検索するファイル・パス検索順序を使用して、*.rep* 拡張子、*.rdf* 拡張子、*.jsp* 拡張子、拡張子なしの順にファイルを検索します。

A.3.53 NAME

NAME キーワードは、*rwbridge* 実行ファイルでのみ使用できます。

説明 NAME を使用して、Oracle Reports ブリッジの名前を指定します。Oracle Reports ブリッジ実行可能ファイル (*rwbridge*) は、ブリッジ構成ファイル *repbrg_bridgename.conf* を *ORACLE_HOME/reports/conf* で検索します。見つからない場合は、新しい構成ファイルが *ORACLE_HOME/reports/conf* に作成されます。

構文 NAME=*bridgename*

値

bridgename には、有効な英数字の文字列を指定します。

デフォルト なし

A.3.54 NONBLOCKSQL

表 A-64 では、NONBLOCKSQL キーワードを使用できる実行可能ファイルを示します。

表 A-64 NONBLOCKSQL キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwsgi	rwserver
○	○	○	×	○	○	×

説明 NONBLOCKSQL を使用して、データベースからデータをフェッチしている間に、他のプログラムを実行できるかどうかを指定できます。

構文 NONBLOCKSQL={YES|NO}

値

- YES の場合、データをフェッチしているときに他のプログラムを実行できます。
- NO の場合、データをフェッチしているときは他のプログラムを実行できません。

デフォルト YES

使用上の注意 NONBLOCKSQL は、コマンドラインから JSP ベースの Web レポートを実行する場合に使用できます。

A.3.55 NOTIFYFAILURE

表 A-65 では、NOTIFYFAILURE キーワードを使用できる実行可能ファイルを示します。

表 A-65 NOTIFYFAILURE キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwsgi	rwserver
○	○	×	×	○	○	×

説明 NOTIFYFAILURE を使用して、レポート・リクエストが失敗した場合に電子メールで通知する宛先を指定できます。このキーワードは、通知クラスを使用するよう Reports Server を構成した場合使用できます。詳細は、「[OracleAS Reports Services の構成](#)」の通知に関する説明を参照してください。

構文 NOTIFYFAILURE=emailid|(emailid,emailid,...)

値

emailid には、someone@foo.com の形式で有効な電子メール・アドレスを指定します。

デフォルト なし

使用上の注意

- 通知電子メールの本文に使用されるデフォルトの通知電子メールのテンプレートが、Oracle Application Server のインストールに含まれています。NOTIFYFAILURE テンプレートは、failnote.txt という名前で、ORACLE_HOME¥reports¥template に格納されています。
- NOTIFYFAILURE は、コマンドラインから JSP ベースの Web レポートを実行する場合に使用できます。

A.3.56 NOTIFYSUCCESS

表 A-66 では、NOTIFYSUCCESS キーワードを使用できる実行可能ファイルを示します。

表 A-66 NOTIFYSUCCESS キーワードを使用する実行可能ファイル

rwclient	rwrund	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
○	○	×	×	○	○	×

説明 NOTIFYSUCCESS を使用して、レポート・リクエストが成功した場合に電子メールで通知する宛先を指定できます。このキーワードは、通知クラスを使用するよう Reports Server を構成した場合使用できます。詳細は、「[OracleAS Reports Services の構成](#)」の通知に関する説明を参照してください。

構文 NOTIFYSUCCESS=*emailid*|(*emailid*,*emailid*,...)

値

*emailid*には、*someone@foo.com* の形式で有効な電子メール・アドレスを指定します。

デフォルト なし

使用上の注意

- 通知電子メールの本文に使用されるデフォルトの通知電子メールのテンプレートが、Oracle Application Server のインストールに含まれています。NOTIFYSUCCESS テンプレートは、*succnote.txt* という名前で、*ORACLE_HOME\reports\template* に格納されています。
- NOTIFYSUCCESS は、コマンドラインから JSP ベースの Web レポートを実行する場合に使用できます。

A.3.57 NUMBERFORMATMASK

表 A-67 では、NUMBERFORMATMASK キーワードを使用できる実行可能ファイルを示します。

表 A-67 NUMBERFORMATMASK キーワードを使用する実行可能ファイル

rwclient	rwrund	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
○	○	×	×	○	○	×

説明 NUMBERFORMATMASK を使用して、数値がデリミタ付きのレポート出力でどのように表示されるかを指定します。

構文 NUMBERFORMATMASK=*mask*

値 任意の数値書式マスク

デフォルト なし

使用上の注意

- このキーワードを使用できるのは、DESFORMAT=DELIMITED または DESFORMAT=DELIMITEDDATA と指定しているときのみです。

注意： 有効な NUMBERFORMATMASK 値は、Oracle Reports のオンライン・ヘルプのトピック「Number Format Mask Syntax」を参照してください。

- NUMBERFORMATMASK は、コマンドラインから JSP ベースの Web レポートを実行する場合に使用できます。

A.3.58 OLAP_CON

表 A-68 では、OLAP_CON キーワードを使用できる実行可能ファイルを示します。

表 A-68 OLAP_CON キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwsgi	rwserver
○	○	○	×	○	○	×

説明 OLAP_CON を使用して、Oracle OLAP（オンライン分析処理）接続文字列を指定し、多次元 Oracle OLAP データが格納されたデータベースに接続します。

構文 OLAP_CON=*userid/password/hostname/SID/portnumber*

値 次のように有効な OLAP 接続文字列を指定します。

- *userid* には、Oracle OLAP に接続するためのユーザー ID を指定します。
- *password* には、ユーザー ID のパスワードを指定します。
- *hostname* には、データベースのホスト名を指定します。
- *SID* には、データベースに接続するためのシステム識別子（SID）を指定します。
- *portnumber* には、データベースに接続するためのポート番号を指定します。

次に例を示します。

OLAP_Con="user1/secret1/mypc.us.oracle.com/mySID/9201

A.3.59 ONFAILURE

表 A-69 では、ONFAILURE キーワードを使用できる実行可能ファイルを示します。

表 A-69 ONFAILURE キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwsgi	rwserver
○	○	○	×	○	○	×

説明 ONFAILURE を使用して、エラーが発生してレポートが完了されなかった場合に COMMIT または ROLLBACK を実行させるかどうかを指定できます。

構文 ONFAILURE={COMMIT|ROLLBACK|NOACTION}

値

- COMMIT の場合、レポートが完了しなかった場合に、COMMIT を実行します。
- ROLLBACK の場合、レポートが完了しなかった場合に、ROLLBACK を実行します。
- NOACTION の場合、レポートが失敗しても何も行いません。

デフォルト

- USERID が指定されている場合は、ROLLBACK です。
- USERID の指定がなく、外部ソース（OracleAS Forms Services など）からコールされる場合は、NOACTION です。

使用上の注意

- ONFAILURE での COMMIT または ROLLBACK は、レポートが失敗した後に実行されます。これに先だって、他の COMMIT および ROLLBACK が発生する場合があります。詳細は、[READONLY](#) コマンドを参照してください。
- ONFAILURE は、コマンドラインから JSP ベースの Web レポートを実行する場合に使用できます。

A.3.60 ONSUCCESS

表 A-70 では、ONSUCCESS キーワードを使用できる実行可能ファイルを示します。

表 A-70 ONSUCCESS キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rservlet	rwcgi	rwserver
○	○	○	×	○	○	×

説明 ONSUCCESS を使用して、レポートの実行が終了した時点で COMMIT または ROLLBACK のどちらを実行するかを指定できます。

構文 ONSUCCESS={COMMIT|ROLLBACK|NOACTION}

値

- COMMIT の場合、レポート完了時に COMMIT を実行します。
- ROLLBACK の場合、レポート完了時に ROLLBACK を実行します。
- NOACTION の場合、レポートが終了しても何も行われません。

デフォルト

- USERID が指定されている場合は COMMIT です。
- USERID の指定がなく、外部ソース (OracleAS Forms Services など) からコールされる場合は、NOACTION です。

使用上の注意

- ONSUCCESS での COMMIT または ROLLBACK は、After Report トリガーが発行されてから実行されます。これに先だって、他の COMMIT および ROLLBACK が発生する場合があります。詳細は、[READONLY](#) コマンドを参照してください。
- ONSUCCESS は、コマンドラインから JSP ベースの Web レポートを実行する場合に使用できます。

A.3.61 ORIENTATION

表 A-71 では、ORIENTATION キーワードを使用できる実行可能ファイルを示します。

表 A-71 ORIENTATION キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rservlet	rwcgi	rwserver
○	○	×	×	○	○	×

説明 ORIENTATION は、レポートの各ページのプリント方向を指定します。

構文 ORIENTATION={DEFAULT|LANDSCAPE|PORTRAIT}

値

- DEFAULT の場合、現行のプリンタの方向設定を使用します。
- LANDSCAPE は横方向です（長辺が上下になる配置）。
- PORTRAIT は縦方向です（短辺が上下になる配置）。

デフォルト DEFAULT

使用上の注意

- ORIENTATION コマンドライン・キーワードは、DESTTYPE=PRINTER の場合にのみ有効になります。PDF および RTF レポート出力の場合は、オブジェクト・ナビゲータの「ペーパー・レイアウト」ノードの「メイン」、「ヘッダー」、「トレーラ」の各セクションの「幅」プロパティ、「高さ」プロパティおよび「印刷方向」プロパティの設定でレポートの方向を変更します。
 - 横方向の場合、幅が高さよりも大きくなります（11 x 8.5 など）。
 - 縦方向の場合、高さが幅よりも大きくなります（8.5 x 11 など）。
- キャラクタ・モードのレポートに対して ORIENTATION=LANDSCAPE を指定する場合、プリンタ定義ファイルに横方向を指定する句が含まれていることが必要です。
- このキーワードは、UNIX 上の PCL プリンタに出力する場合はサポートされていません。

A.3.62 OUTPUTFOLDER

表 A-72 では、OUTPUTFOLDER キーワードを使用できる実行可能ファイルを示します。

表 A-72 OUTPUTFOLDER キーワードを使用する実行可能ファイル

rwclient	rwrwn	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
○	○	×	×	○	○	×

説明 OUTPUTFOLDER を使用して、レポート出力の配信先となる Oracle WebDB リリース 2.2 または Oracle9iAS Portal リリース 1 のフォルダの名前を指定できます。このキーワードは、Oracle WebDB リリース 2.2 および Oracle9iAS Portal リリース 1 に対する下位互換性を確保するために維持されています。OracleAS Portal 10g (9.0.4) 以降では、[OUTPUTPAGE](#) を使用してください。

構文 OUTPUTFOLDER=*folder*

値

folder には、Oracle WebDB リリース 2.2 または Oracle9iAS Portal リリース 1 で使用される有効なフォルダ名（内部名）を指定します。

デフォルト Oracle_Reports_Output

使用上の注意

- このキーワードの値は大文字と小文字を区別します。
- Oracle Reports 出力を Oracle WebDB リリース 2.2 または Oracle9iAS Portal リリース 1 に配信する際、このキーワードの使用はオプションです。
- 関連するキーワードは、[CONTENTAREA*](#)、[EXPIREDAYS](#)、[ITEMTITLE](#)、[OUTPUTFOLDER*](#)、[OUTPUTPAGE](#)、[PAGEGROUP](#)、[SITENAME*](#)、[STATUSFOLDER*](#)、[STATUSPAGE](#) です。

* Oracle9iAS Portal リリース 1 および Oracle WebDB リリース 2.2 に対する下位互換性を確保するために維持されています。

A.3.63 OUTPUTIMAGEFORMAT

表 A-73 では、OUTPUTIMAGEFORMAT キーワードを使用できる実行可能ファイルを示します。

表 A-73 OUTPUTIMAGEFORMAT キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
○	○	○	×	○	○	×

説明 OUTPUTIMAGEFORMAT を使用して、レポート出力のイメージの形式を指定します。

構文 OUTPUTIMAGEFORMAT={ PNG | JPEG | JPG | GIF | BMP }

値

- PNG、JPEG (デフォルト)、JPG
DESFORMAT が PDF、HTML、HTMLCSS、RTF または SPREADSHEET である場合にサポートされるイメージ形式。
- GIF
DESFORMAT 値が PDF、HTML、HTMLCSS または SPREADSHEET である場合にサポートされるイメージ形式。
- BMP
DESFORMAT 値が RTF の場合にサポートされるイメージ形式。

使用上の注意

- このコマンドライン・キーワードは、REPORTS_OUTPUTIMAGEFORMAT 環境変数の設定より優先されます。
- REPORTS_DEFAULT_DISPLAY 環境変数が明示的に NO (デフォルトは YES) に設定されている場合、このコマンドライン・キーワードはサポートされません。その場合、デフォルトのイメージ形式は、HTML、HTMLCSS および PDF 出力の場合 GIF 形式、RTF 出力の場合 BMP 形式になります。
- 指定した形式が出力タイプに一致するようにしてください。たとえば、BMP は RTF 出力にのみ適用します。HTML、HTMLCSS または PDF 出力には無効です。
- イメージを含むレポートおよびセキュリティで保護された Reports Server を使用した DESFORMAT=SPREADSHEET への実行はサポートされていません。これは、Microsoft Excel の Cookie サポートに対する制限が原因です。かわりに、WebDAV を使用して URL にセキュリティで保護された Reports Server からの Excel 出力を書き込むことができます。

例

例 1

次のコマンドラインでは、HTML 出力に PNG イメージが生成されます。

```
rwclient server=my_rep_server report=images.rdf destype=file
desformat=html desname=images.html userid=user_id outputimageformat=PNG

rwrn report=images.rdf destype=file desformat=html desname=images.html
userid=user_id outputimageformat=PNG
```

同様に、DESFORMAT=pdf を指定した場合、イメージは生成された PDF 文書に PNG 形式で埋め込まれます。

例 2

OUTPUTIMAGEFORMAT に無効な値を指定すると、エラー・メッセージが表示されます。次のコマンドラインでは、エラー・メッセージが表示されます。

```
rwclient server=my_rep_server report=images.rdf destype=file
desformat=html desname=images.html userid=user_id
outputimageformat=ABCD

rwr run report=images.rdf destype=file desformat=html desname=images.html
userid=user_id outputimageformat=ABCD
```

イメージ形式 ABCD は無効であるため、次のエラー・メッセージが表示されます。

REP-35000: The image output format specified is not supported.

A.3.64 OUTPUTPAGE

表 A-74 では、OUTPUTPAGE キーワードを使用できる実行可能ファイルを示します。

表 A-74 OUTPUTPAGE キーワードを使用する実行可能ファイル

rwclient	rwr run	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
○	○	×	×	○	○	×

説明 OUTPUTPAGE を使用して、レポート出力の配信先となる OracleAS Portal ページの名前を指定できます。旧バージョン（Oracle WebDB リリース 2.2 および Oracle9iAS Portal リリース 1）に対する下位互換性については、「[OUTPUTFOLDER](#)」を参照してください。

構文 OUTPUTPAGE=*page*

値

page には、OracleAS Portal で使用される有効なページ名（内部名）を指定します。

デフォルト Oracle_Reports_Output

使用上の注意

- Oracle Reports 出力を OracleAS Portal に配信する際、このキーワードの使用はオプションです。
 - 出力ページを指定しない場合、Oracle_Reports_Output という名前のデフォルト・ページが OracleAS Portal により作成されます。
 - 出力ページを指定する場合は、表示名ではなく内部名を使用します。内部名は OracleAS Portal でのコンポーネント・インスタンスの一意の識別に使用されます。
- このキーワードの値は大文字と小文字を区別します。
- DESTYPE=ORACLEPORTAL が指定されている場合、ページには 1 つ以上の項目の領域が含まれる必要があります。
- 関連するキーワードは、[CONTENTAREA*](#)、[EXPIREDAYS](#)、[ITEMTITLE](#)、[OUTPUTFOLDER*](#)、[OUTPUTPAGE](#)、[PAGEGROUP](#)、[SITENAME*](#)、[STATUSFOLDER*](#)、[STATUSPAGE](#) です。

* Oracle9iAS Portal リリース 1 および Oracle WebDB リリース 2.2 に対する下位互換性を確保するために維持されています。

A.3.65 OVERWRITE

表 A-75 では、OVERWRITE キーワードを使用できる実行可能ファイルを示します。

表 A-75 OVERWRITE キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwsgi	rwserver
×	×	×	○	×	×	×

説明 OVERWRITE を使用して、既存のファイルを、変換済のファイルで上書きするかどうかを指定できます。

構文 OVERWRITE={YES|NO|PROMPT}

値

- YES の場合、同名の既存ファイルを自動的にすべて上書きします。
- NO の場合、同名の既存ファイルがあっても、レポートを変換せず、警告メッセージを表示します。
- PROMPT の場合、既存ファイルを上書きする前に、確認を要求します。

デフォルト NO

A.3.66 P_AVAILABILITY

表 A-76 では、P_AVAILABILITY キーワードを使用できる実行可能ファイルを示します。

表 A-76 P_AVAILABILITY キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwsgi	rwserver
×	×	×	○	×	×	×

説明 P_AVAILABILITY を使用して、指定されたレポートがいつ処理できるようになるかを決定する使用可能日カレンダーの名前を指定します。このキーワードが使用されるのは、DTYPE=REGISTER の場合のみです。

構文 P_AVAILABILITY=*calendar_name*

値

calendar_name には、有効な使用可能日カレンダー名を指定します。

デフォルト なし

使用上の注意 使用可能日カレンダーは、この SQL*Plus スクリプトを実行する前に、OracleAS Portal に登録されている必要があります。登録されていない場合、無効なパッケージが作成されることがあります。

A.3.67 P_DESCRIPTION

表 A-77 では、P_DESCRIPTION キーワードを使用できる実行可能ファイルを示します。

表 A-77 P_DESCRIPTION キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwsgi	rwserver
×	×	×	○	×	×	×

説明 P_DESCRIPTION を使用して、レポートに関する追加情報を提供するテキストを指定できます。このキーワードが使用されるのは、DTYPE=REGISTER の場合のみです。

構文 P_DESCRIPTION=*description_text*

値

description_text には、任意のテキスト文字列を指定します。

デフォルト なし

A.3.68 P_FORMATS

表 A-78 では、P_FORMATS キーワードを使用できる実行可能ファイルを示します。

表 A-78 P_FORMATS キーワードを使用する実行可能ファイル

rwclient	rwrund	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
×	×	×	○	×	×	×

説明 P_FORMATS を使用して、指定されたレポートに許可された宛先フォーマットを指定できます。このキーワードが使用されるのは、DTYPE=REGISTER の場合のみです。

構文 P_FORMATS=*destination_format* / (*destination_format1, destination_format2, ...*)

値

destination_format には、有効な宛先の形式 (HTML など) または、名前をカンマで区切り、括弧で囲んだ有効な宛先の形式のリスト ((HTMLCSS,PDF,RTF) など) を指定します。

使用上の注意 レポートの宛先フォーマットが DELIMITEDDATA の場合は、レポートを OracleAS Portal にバッチ登録できないことがあります。回避策として、異なる宛先フォーマットを定義して、レポートをバッチ登録し、後で手動でレポートを DESFORMAT=DELIMITEDDATA に編集できます。バッチ登録の詳細は、[付録 C 「OracleAS Portal へのレポートのバッチ登録」](#) を参照してください。

デフォルト なし

A.3.69 P_JDBCPDS

表 A-79 では、P_JDBCPDS キーワードを使用できる実行可能ファイルを示します。

表 A-79 P_JDBCPDS キーワードを使用する実行可能ファイル

rwclient	rwrund	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
○	○	○	×	○	○	×

説明 P_JDBCPDS を使用して、JDBC 問合せを含むレポートを実行するためにデータベースに接続するための JDBC プラuggブル・データソース (PDS) 接続文字列を指定します。

関連項目: [第9章 「JDBC PDS の構成と使用」](#)

構文 P_JDBCPDS=*userid/password@database*

値 次のように有効な JDBC PDS 接続文字列を指定します。

- `userid` には、JDBC プラuggابل・データ・ソースに接続するためのユーザー ID を指定します。
- `password` には、ユーザー ID のパスワードを指定します。
- `database` には、第 9.2.1 項「接続情報の例」の説明のように、特定のデータソースに固有のデータベース接続情報を指定します。

たとえば、Sybase データソースに接続するには次のようにします。

```
P_JDBCPDS=sybuser/sybpwd@server1.mydomain.com:1300
```

使用上の注意 `P_JDBCPDS` は、JDBC PDS に接続するためのデフォルトのサインオン名です。レポート・ウィザードで JDBC 問合せを定義するときに、Reports Builder 内でこの名前を変更できます。新しいサインオン名は、`P_JDBCPDS` コマンドライン・キーワードの値としてコマンドラインで使用できます。Reports Builder での JDBC 問合せの定義の詳細は、第 9.2 項「JDBC 問合せの定義と実行」を参照してください。

A.3.70 P_NAME

表 A-80 では、`P_NAME` キーワードを使用できる実行可能ファイルを示します。

表 A-80 P_NAME キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
×	×	×	○	×	×	×

説明 `P_NAME` を使用して、OracleAS Portal に表示されるレポート名を指定します。このキーワードが使用されるのは、`DTYPE=REGISTER` の場合のみです。

構文 `P_NAME=report_name`

値

`report_name` には、任意のレポート名を指定します。

デフォルト `P_NAME` を指定しない場合、PL/SQL ファンクションにはレポート定義ファイル名が入力されます。

使用上の注意

- `P_NAME` を指定しない場合、PL/SQL ファンクションにはレポート定義ファイル名が入力されます。
- `P_NAME` は、OracleAS Portal に登録する各レポート定義ファイルに対して同じレポート名を使用する場合のみ指定します。このオプションは通常、空白にします。
- レポート名の先頭には数字は使用できません（たとえば、`401K_report` は無効なファイル名ですが、`my_401K_report` は有効です）。

A.3.71 P_OWNER

表 A-81 では、`P_OWNER` キーワードを使用できる実行可能ファイルを示します。

表 A-81 P_OWNER キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
×	×	×	○	×	×	×

説明 P_OWNER を使用して、レポート定義ファイルの登録時に作成された、レポートのパッケージを所有する OracleAS Portal DB プロバイダを指定できます。このキーワードが使用されるのは、DTYPE=REGISTER の場合のみです。

構文 P_OWNER=*portal_dbprovider*

値

portal_dbprovider には、有効な OracleAS Portal DB プロバイダ名を指定します。

デフォルト SQL*Plus スクリプト・ファイルの実行時に接続した OracleAS Portal DB プロバイダの名前。

A.3.72 P_PFORMTEMPLATE

表 A-82 では、P_PFORMTEMPLATE キーワードを使用する実行可能ファイルを示します。

表 A-82 P_PFORMTEMPLATE キーワードを使用する実行可能ファイル

rwclient	rwrund	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
×	×	×	○	×	×	×

説明 P_PFORMTEMPLATE を使用して、ランタイム・パラメータ・フォームのスタイルを決定する OracleAS Portal テンプレートの名前を指定できます。このキーワードが使用されるのは、DTYPE=REGISTER の場合のみです。

構文 P_PFORMTEMPLATE=*template_name*

値

template_name には、有効な OracleAS Portal テンプレート名を指定します。

デフォルト なし

A.3.73 P_PRINTERS

表 A-83 では、P_PRINTERS キーワードを使用する実行可能ファイルを示します。

表 A-83 P_PRINTERS キーワードを使用する実行可能ファイル

rwclient	rwrund	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
×	×	×	○	×	×	×

説明 P_PRINTERS を使用して、指定されたレポートに許可されたプリンタを指定できます。このキーワードが使用されるのは、DTYPE=REGISTER の場合のみです。

構文 P_PRINTERS=*printer_name*

値

printer_name には、有効なプリンタ (PRT1 など) または、名前をカンマで区切り、括弧で囲んだ有効なプリンタのリスト ((PRT1,PRT2,PRT3) など) を指定します。

デフォルト なし

使用上の注意 プリンタへのアクセスは、この SQL*Plus スクリプトを実行する前に、OracleAS Portal に登録されている必要があります。

A.3.74 P_PRIVILEGE

表 A-84 では、P_PRIVILEGE キーワードを使用できる実行可能ファイルを示します。

表 A-84 P_PRIVILEGE キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
×	×	×	○	×	×	×

説明 P_PRIVILEGE を使用して、選択したレポートを実行するアクセス権を持つユーザーまたはロールを指定できます。このキーワードが使用されるのは、DTYPE=REGISTER の場合のみです。

構文 P_PRIVILEGE=user_name

値

user_name には、OracleAS Portal が認識できるユーザー名またはロール (SCOTT など)、または、名前をカンマで区切り、括弧で囲んだユーザー名またはロールのリスト ((SCOTT,JABERS,PMARTIN) など) を指定します。

デフォルト なし

A.3.75 P_SERVERS

表 A-85 では、P_SERVERS キーワードを使用できる実行可能ファイルを示します。

表 A-85 P_SERVERS キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
×	×	×	○	×	×	×

説明 P_SERVERS を使用して、レポートを実行できる制限付き Reports Server の名前を指定できます。このキーワードが使用されるのは、DTYPE=REGISTER の場合のみです。

構文 P_SERVERS=tnsname

値

tnsname には、Reports Server の有効な TNS 名 (repserver など) または、名前をカンマで区切り、括弧で囲んだ Reports Server の有効な TNS 名のリスト ((repserver,acct_server,sales_server) など) を指定します。

デフォルト なし

使用上の注意 Reports Server へアクセスする前に OracleAS Portal に登録されている必要があります。

A.3.76 P_TRIGGER

表 A-86 では、P_TRIGGERS キーワードを使用する実行可能ファイルを示します。

表 A-86 P_TRIGGER キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
×	×	×	○	×	×	×

説明 P_TRIGGER を使用して、パラメータ値をコマンドラインで指定する際およびランタイム・パラメータ・フォームを確定する際に実行される PL/SQL ファンクションを指定できます。ファンクションは、ブール値 (TRUE または FALSE) を戻す必要があります。次に例を示します。

```
P_TRIGGER="Is begin IF UPPER(DESTYPE) = 'PRINTER' AND EMPNAME =
'SMITH' THEN RETURN(TRUE); ELSE RETURN(FALSE); END IF; end;"
```

このキーワードが使用されるのは、DTYPE=REGISTER の場合のみです。

構文 P_TRIGGER=*plsql_function*

値

plsql_function には、ブール値を戻す有効な PL/SQL ファンクションを指定します。

デフォルト なし

A.3.77 P_TYPES

表 A-87 では、P_TYPES キーワードを使用できる実行可能ファイルを示します。

表 A-87 P_TYPES キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwsgi	rwserver
×	×	×	○	×	×	×

説明 P_TYPES を使用して、指定されたレポートに許可された宛先タイプを指定できます。このキーワードが使用されるのは、DTYPE=REGISTER の場合のみです。

構文 P_TYPES=*destination_type*

値

destination_type には、有効な宛先タイプ (CACHE など) または、名前をカンマで区切り、括弧で囲んだ有効な宛先タイプのリスト ((CACHE,MAIL,PRINTER) など) を指定します。

デフォルト なし

A.3.78 PAGEGROUP

表 A-88 では、PAGEGROUP キーワードを使用できる実行可能ファイルを示します。

表 A-88 PAGEGROUP キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwsgi	rwserver
○	○	×	×	○	○	×

説明 PAGEGROUP を使用して、レポート出力の配信先となる OracleAS Portal ページ・グループの名前を指定できます。旧リリースとの下位互換性については、SITENAME (Oracle WebDB リリース 2.2 の場合) および「CONTENTAREA」(Oracle9iAS Portal リリース 1 の場合) を参照してください。このキーワードを使用するには、ページ・グループを事前に OracleAS Portal で作成しておく必要があります。

構文 PAGEGROUP=*pagegroup*

値

pagegroup には、OracleAS Portal で使用されている有効なページ・グループ名 (内部名) を指定します。

デフォルト なし

使用上の注意

- Oracle Reports 出力を OracleAS Portal に配信する際、このキーワードの使用は必須です。
- ページ・グループ名には、表示名ではなく内部名を使用する必要があります。内部名を使用して、OracleAS Portal ページ・インスタンスを一意に識別します。
- 関連するキーワードは、[CONTENTAREA*](#)、[EXPIREDAYS](#)、[ITEMTITLE](#)、[OUTPUTFOLDER*](#)、[OUTPUTPAGE](#)、[PAGEGROUP](#)、[SITENAME*](#)、[STATUSFOLDER*](#)、[STATUSPAGE](#) です。

* Oracle9iAS Portal リリース 1 および Oracle WebDB リリース 2.2 に対する下位互換性を確保するために維持されています。

A.3.79 PAGESIZE

表 A-89 では、PAGESIZE キーワードを使用できる実行可能ファイルを示します。

表 A-89 PAGESIZE キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
○	○	○	○	○	○	×

説明 PAGESIZE を使用して、物理ページの大きさ（プリンタが出力するページのサイズ）を設定できます。ページはレポートが入る大きさである必要があるので注意してください。たとえば、レポートのフレームがページ・ディメンションを超えるサイズになっている場合、レポートは実行されません。

構文 PAGESIZE=*width* x *height*

値 「ページ幅 x ページ高さ」の形式の有効なページ・サイズ。ページ幅とページ高さは 0（ゼロ）より大きくする必要があります。最大幅・高さは、Reports Builder に設定されている測定単位によって異なります（「[編集](#)」→「[プリファレンス](#)」→「[一般](#)」タブ）。インチでは、最大幅・高さは 512 インチです。cm では 1312cm です。パイカでは 36,864 パイカです。

デフォルト ビットマップの場合は 8.5 × 11 インチです。キャラクタ・モードの場合は 80 × 66 文字です。レポートがキャラクタ・モードに対して設計されていて、ビットマップで実行あるいは変換される場合、ページ・サイズが指定されていないときは次の式を使用して決定されます。（デフォルトのページ・サイズ * 文字ページのサイズ） / デフォルトの文字のページ・サイズ。たとえば、文字ページのサイズが 80 × 20 の場合、ビットマップのページは次のサイズになります。

$$((8.5 * 80) / 80) \times ((11 * 20) / 66) = (680 / 80) \times (220 / 66) = 8.5 \times 3.33$$

使用上の注意

- プリンタによっては、物理ページの印刷可能な面積に制限があります。たとえば、プリンタが受け入れる用紙は 8.5 × 11 インチであっても、実際にプリントできる面積は 8 × 10.5 インチである可能性があります。Reports Builder で、使用するプリンタが許容する印刷可能面積を超えるページ幅とページ高さを定義すると、レポートを出力した際にクリッピングが生じる可能性があります。クリッピングを防ぐには、プリンタの印刷可能領域を拡大するか（オペレーティング・システムで可能な場合）、ページ幅×ページの高さをページの印刷可能な領域のサイズに設定します。
- レター・サイズは、8.5 × 11 インチです。A4 サイズは、210mm × 297mm です。
- PAGESIZE キーワードを使用すると、その値が、レポート定義のページ・ディメンションよりも優先されます。
- ランタイム・パラメータ・フォームに入力された PAGESIZE 値は、コマンドラインに入力された PAGESIZE 値よりも優先されます。

A.3.80 PGESTREAM

表 A-90 では、PGESTREAM キーワードを使用できる実行可能ファイルを示します。

表 A-90 PGESTREAM キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
○	○	×	×	○	○	×

説明 PGESTREAM は、レポートが HTML あるいは HTMLCSS 出力としてフォーマットされている場合に、次のいずれかで設定されているナビゲーション・コントロールを使用して、そのレポートのページ・ストリーム（ページ区切り）を有効または無効にします。

- レポート・プロパティ・パレットのページ・ナビゲーション・コントロール値プロパティおよびページ・ナビゲーション・コントロール・タイプ・プロパティ
- BEFORE REPORT トリガーの PL/SQL (SRW.SET_PAGE_NAVIGATION_HTML)

構文 PGESTREAM={YES|NO}

値

- YES の場合、レポート出力のページ区切りを行います。
- NO の場合、ページ区切りなしでレポートを出力します。

デフォルト NO

A.3.81 PARAMFORM

表 A-91 では、PARAMFORM キーワードを使用できる実行可能ファイルを示します。

表 A-91 PARAMFORM キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
×	×	×	×	○	○	×

説明 PARAMFORM を使用して、レポートを CGI またはサーブレットを使用して実行するときランタイム・パラメータ・フォームを表示するかどうかを指定します。PARAMFORM は、JSP ベースの Web レポートではなく、ペーパー・レイアウト・レポートにパラメータを指定するためにのみ使用されます。

構文 PARAMFORM=YES|NO|HTML

値

- YES の場合、パラメータ・フォームが表示されます。
- NO の場合、パラメータ・フォームが表示されません。
- HTML の場合、パラメータ・フォームが HTML 形式で表示されます。

デフォルト NO

使用上の注意

- BATCH=YES の場合、PARAMFORM=YES を設定しても無効です。これは、バッチ・モードではランタイム・パラメータ・フォームを表示する意味がないためです。
- このキーワードは、OracleAS Portal 環境でレポートを実行する場合は使用しないでください。これは、OracleAS Portal 上でもレポートのランタイム・パラメータ・フォームの設定が可能であり、この設定が PARAMFORM キーワードで指定したパラメータ・フォームと競合する可能性があるためです。

例

```
http://myias.mycomp.com:7779/rwservlet?server=myrepserver+report=test.rdf
+userid=scott/tiger@mydb+destype=cache+desformat=htmlcss+paramform=html
```

```
http://mywebserver.com:7779/cgi-bin/rwcgi.exe?server=myrepserver+report=test.rdf
+userid=scott/tiger@mydb+destype=cache+desformathtmlcss+=paramform=yes
```

A.3.82 PARSEQUERY

表 A-92 では、PARSEQUERY キーワードを使用できる実行可能ファイルを示します。

表 A-92 PARSEQUERY キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
×	×	×	×	○	○	×

説明 PARSEQUERY を使用して、rwservlet に対するコマンド問合せを解析し、作成された Reports Server コマンドラインを表示できます。

構文 `http://your_webserver/reports/rwservlet/parsequery[?][server=server_name][&authid=username/password]`

値 構文を参照してください。

デフォルト なし

使用上の注意

- このキーワードは値を必要としないコマンドです。つまり、対応する値を持たず、コマンドだけが入力されます。
- 関連するキーワードは、[SERVER](#) および [AUTHID](#) です。

A.3.83 PDFCOMP

表 A-93 では、PDFCOMP キーワードを使用できる実行可能ファイルを示します。

表 A-93 PDFCOMP キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
○	○	×	×	○	○	×

説明 PDFCOMP を使用して、PDF 出力を圧縮するかどうかを指定できます。

構文 `PDFCOMP=value|{YES|NO}`

値

- value には、0～9の任意の値を指定します。0の場合、PDF出力は圧縮されません。1～9の場合、PDF出力が圧縮され、ユーザーが圧縮レベルを制御できます。
- YESの場合、出力は圧縮レベル6で圧縮されます。
- NOの場合、出力は圧縮レベル0（非圧縮）で圧縮されます。

デフォルト 6

A.3.84 PDFEMBED

表 A-94 では、PDFEMBED キーワードを使用できる実行可能ファイルを示します。

表 A-94 PDFEMBED キーワードを使用する実行可能ファイル

rwclient	rwruntime	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
○	○	×	×	○	○	×

説明 PDFEMBED を使用して、uifont.ali で指定した Type1 PostScript フォント・ファイルを Oracle Reports が PDF 出力に埋め込むかどうかを指定できます。

構文 PDFEMBED={YES|NO}

値

- YES の場合、uifont.ali ファイルの [PDF:Embed] ヘッダーで指定されたフォントが、PDF ドライバによって PDF 出力に埋め込まれます。
- NO の場合、フォントは PDF 出力に追加されません。

デフォルト YES

A.3.85 PFACTION

表 A-95 では、PFACTION キーワードを使用できる実行可能ファイルを示します。

表 A-95 PFACTION キーワードを使用する実行可能ファイル

rwclient	rwruntime	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
○	×	×	×	×	×	×

注意： PFACTION は、OracleAS Forms Services のキーワード RUN_REPORT_OBJECT とともに使用することもあります。これは、Web 上で Oracle Forms から Oracle Reports をコールする際の最も安全な方法です。RUN_REPORT_OBJECT を使用して Oracle Forms から Oracle Reports をコールする方法の詳細は、OTN にあるホワイト・ペーパー『Oracle Application Server 10g Integrating Oracle Reports in Oracle Forms Services Applications』(<http://otn.oracle.com/products/forms/pdf/10g/frm10gsrw10g.pdf>) を参照してください。また、『Oracle Application Server Forms Services 利用ガイド』も参照してください。

説明 PFACTION は、パラメータ・フォームのアクション文字列を指定するために使用します。rwclient または OracleAS Forms Services の RUN_REPORT_OBJECT キーワードを通じてリクエストを送信して、パラメータ・フォームを持つレポートを実行する場合、生成されたパラメータ・フォームは、コマンドラインに PFACTION が含まれていないと使用できません。パラメータ・フォームが使用できないという事象は、action 属性が空になっていることが原因です (rwclient と RUN_REPORT_OBJECT が、サーバー上で直接 Oracle Reports をコールするため、Oracle Reports は、HTML パラメータ・フォームを生成する際に、Web 環境にアクセスして action 属性への値移入に必要な情報を取得できません)。

action 属性は、エンド・ユーザーが「実行」をクリックしたときに実行されるアクションを定義する標準 HTML タグ form の一部です。Oracle Reports パラメータ・フォームの action 属性は、エンド・ユーザーが「実行」をクリックした後、Oracle Reports のリクエストを処理するために必要な非表示ランタイム・パラメータを含む必要があります。

構文 PFACTION=request_URL_to_rwervlet?_hidden_encoded_original_url_string

値

- `request_URL_to_rwservlet`: `http://host:port/reports/rwservlet` の URL。
- `encoded_original_url_string`: レポートを実行する問合せ文字列。

例 `rwclient.exe report=pform.rdf destype=cache desformat=pdf paramform=yes PFACTION="http://mymachine.mycompany.com:7777/reports/rwservlet?_hidden_report=pform.rdf%20destype=cache%20desformat=pdf"`

A.3.86 PRINTJOB

表 A-96 では、PRINTJOB キーワードを使用できる実行可能ファイルを示します。

表 A-96 PRINTJOB キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
×	×	○	×	×	×	×

説明 PRINTJOB を使用して、レポートを実行する前に「**ジョブの印刷**」ダイアログ・ボックスを表示させるかどうかを指定できます。

構文 PRINTJOB={YES|NO}

値

- YES の場合、レポートが実行される前に「**ジョブの印刷**」ダイアログ・ボックスが表示されます。
- NO の場合、「**ジョブの印刷**」ダイアログ・ボックスが表示されずに、レポートが実行されます。

デフォルト YES

使用上の注意

- レポートが派生プロセスで実行されるときは (rwrn などの実行ファイルが、rwbuilder などの別の実行ファイル内でコールされる場合)、PRINTJOB の値にかかわらず、「**ジョブの印刷**」ダイアログ・ボックスは表示されません。
- DESTYPE=MAIL のときは、PRINTJOB の値にかかわらず、「**ジョブの印刷**」ダイアログ・ボックスは表示されません。

A.3.87 READONLY

表 A-97 では、READONLY キーワードを使用できる実行可能ファイルを示します。

表 A-97 READONLY キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
○	○	○	×	○	○	×

説明 READONLY を使用して、レポートでの複数回にわたる問合せで読取り一貫性をリクエストできます。Oracle データベースからデータにアクセスする場合、読取り一貫性は、SET TRANSACTION READ ONLY 文で確保されます (SET TRANSACTION READ ONLY の詳細は、使用している Oracle データベースのドキュメントを参照してください)。

注意: SET TRANSACTION READ ONLY の詳細は、Oracle SQL のドキュメントを参照してください。このドキュメントは、Oracle Technology Network (<http://www.oracle.com/technology/index.html>) で入手できます。

構文 READONLY={YES|NO}

値

- YES の場合、読取り一貫性をリクエストします。
- NO の場合、読取り一貫性は提供されません。

デフォルト NO

使用上の注意

- READONLY は、複数の問合せを使用するレポートでのみ役立ちます。Oracle データベースでは、単一の問合せを使用するレポートに対しては、読取り一貫性がロックなしで自動的に提供されます。
- レポート・トリガーの実行順に関して、データ・フェッチが行われる前に SET TRANSACTION READ ONLY を設定しておく必要があります。
- READONLY は、コマンドラインから JSP ベースの Web レポートを実行する場合に使用できません。

A.3.88 RECURSIVE_LOAD

表 A-98 では、RECURSIVE_LOAD キーワードを使用できる実行可能ファイルを示します。

表 A-98 RECURSIVE_LOAD キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
○	○	×	○	○	○	×

説明 RECURSIVE_LOAD を使用して、レポートの実行時に、プログラム・ユニットの外部参照を検証するかどうかを指定できます。無効な参照があった場合、プログラム・ユニットは自動的に再コンパイルされます。RECURSIVE_LOAD を NO に設定することは、最初に PL/SQL がコンパイルされたデータベースとは異なるデータベースにレポートを実行する場合に役立ちます。

構文 RECURSIVE_LOAD={YES|NO}

値

- YES の場合、レポートの実行時に外部参照が検証されます。無効な参照があった場合、そのプログラム・ユニットは (.rdf または .p11 であっても) 再コンパイルされます。
- NO の場合、レポートの実行時に外部参照が検証されません。この設定は、最初に PL/SQL がコンパイルされたデータベースとは異なるデータベースにレポートを実行する場合に有効です。

デフォルト YES

A.3.89 REPLYTO

表 A-99 では、REPLYTO キーワードを使用できる実行可能ファイルを示します。

表 A-99 REPLYTO キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
○	○	×	×	○	○	×

説明 REPLYTO を使用して、送信者が送信者 (**FROM** キーワードで指定) 以外の宛先に返信を送信させる場合に、返信の送信先となる電子メール・アドレスを指定できます。

構文 REPLYTO=*emailid*

値

emailid には、*someone@foo.com* の形式で有効な電子メール・アドレスを指定します。

デフォルト なし

使用上の注意 関連するキーワードは、**BCC**、**CC**、**FROM**、**REPLYTO** および **SUBJECT** です。電子メールの主な宛先を指定するには、**DESNAME** を使用します。

A.3.90 REPORT|MODULE

「**MODULE|REPORT**」を参照してください。

A.3.91 ROLE

表 A-100 では、ROLE キーワードを使用できる実行可能ファイルを示します。

表 A-100 ROLE キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
○	○	×	×	○	○	×

説明 ROLE を使用して、レポートに対するデータベース・ロールを実行時にチェックするように指定できます。このキーワードは、通常アクセス権限を持たないデータベース表に対して、問合せを行うレポートを実行できるようにする場合に有効です。

構文 ROLE={*rolename* [/rolepassword]}

値

rolename には、有効なロールを指定します。

rolepassword には、(オプションで) ロール・パスワードを指定します。

デフォルト なし

使用上の注意 ROLE は、コマンドラインから JSP ベースの Web レポートを実行する場合に使用できます。

A.3.92 RUNDEBUB

表 A-101 では、RUNDEBUB キーワードを使用できる実行可能ファイルを示します。

表 A-101 RUNDEBUB キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
○	○	○	×	○	○	×

説明 RUNDEBUB を使用して、レポート中の論理エラーの実行時チェックを指定できます。RUNDEBUB では、エラーではなくても不測の出力がされる可能性のある内容がチェックされ、実行時のレポート出力が表示される前に、警告として表示されます。RUNDEBUB を使用するとレポートはデバッグ・モードで実行されますが、これは、PL/SQL インタプリタを使用したレポートのデバッグとは異なります。

RUNDEBUB では、次の内容がチェックされます。

- 枠または繰返し枠で、重なっているが別のオブジェクトを囲まないもの。これは、出力時にオブジェクトが別のオブジェクトに重なって表示されます。
- 固定サイズを持たないページ依存参照を持つレイアウト・オブジェクト。このようなオブジェクトは、「垂直拡張度」と「水平拡張度」のプロパティ設定にかかわらず、サイズが固定されます。
- PL/SQL の無効な頻度で参照されるバインド変数。

構文 RUNDEBUB={YES|NO}

値

- YES の場合、追加の実行時エラー・チェックが実行されます。
- NO の場合、追加の実行時エラー・チェックが実行されません。

デフォルト YES

使用上の注意 RUNDEBUB は、コマンドラインから JSP ベースの Web レポートを実行する場合に使用できます。

A.3.93 SAVE_RDF

表 A-102 では、SAVE_RDF キーワードを使用できる実行可能ファイルを示します。

表 A-102 SAVE_RDF キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
×	○	○	×	×	×	×

説明 SAVE_RDF を使用して、RDF ファイルおよび XML カスタマイズ・ファイルを複合したファイル名を指定します。このキーワードは、CUSTOMIZE キーワードを使用して既存の RDF ファイルと Oracle Reports XML カスタマイズ・ファイルを複合し、その複合したものを新しい RDF ファイルに保存する場合に使用します。

構文 SAVE_RDF=*filename.rdf*

値 任意の有効なファイル名

デフォルト なし

使用上の注意 SAVE_RDF は、JSP ファイルにも使用できますが、ペーパー・レイアウト部分のみで、Web ソースには使用できません。

A.3.94 SCHEDULE

表 A-103 では、SCHEDULE キーワードを使用できる実行可能ファイルを示します。

表 A-103 SCHEDULE キーワードを使用する実行可能ファイル

rwclient	rwrwn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
○	×	×	×	○	○	×

説明 SCHEDULE を使用して、レポートを実行する日付、時刻および頻度を設定できます。デフォルトは、1 回および即時のレポート実行です。時刻値は、24 時間方式で表します（例：1 時は 13:00）。スケジュール・コマンドは引用句 (") で囲むか、もしくはスペースのかわりにアンダースコア (_) を使用します。また、ある実行回数の後、または特定の日付 / 時刻にレポート・ジョブの期限切れを指定することもできます。次に例を示します。

```
SCHEDULE=every_first_fri_of_month_from_15:53_Oct_23,_1999_retry_3_
after_1_hour_expires_on_15:53_Oct_23,_2003
```

```
SCHEDULE=last_weekday_before_15_from_15:53_Oct_23,_1999_retry_after_1_
hour_expires_after_100
```

または

```
SCHEDULE="every first fri of month from 15:53 Oct 23, 1999 retry 3
after 1 hour expires on 15:53 Oct 23, 2003"
```

```
SCHEDULE="last weekday before 15 from 15:53 Oct 23, 1999 retry after 1
hour expires after 100"
```

構文 SCHEDULE=*string*

ここで、*string* は次のように指定します。

```
[FREQ from] TIME [retry {n} after LEN expires {on|after} time|n]
```

この文字列で使用される値は、表 A-104 で説明します。

表 A-104 SCHEDULE キーワードで使用できる文字列値

FREQ	hourly daily weekly monthly {every {LEN DAYREPEAT}} {last {WEEKDAYS weekday weekend} before {n}+}
LEN	{n}+ {minute[s] hour[s] day[s] week[s] month[s]}
DAYREPEAT	{first second third fourth fifth} WEEKDAYS of month
WEEKDAYS	mon tue wed thu fri sat sun
TIME	now CLOCK [DATE]
CLOCK	h:m h:mm hh:m hh:mm
DATE	today tomorrow {MONTHS {d dd} [,year]}
MONTHS	jan feb mar apr may jun jul aug sep oct nov dec
EXPIRES	on {today tomorrow {MONTHS {d dd} [,year]}} after n

デフォルト なし

A.3.95 SERVER

表 A-105 では、SERVER キーワードを使用できる実行可能ファイルを示します。

表 A-105 SERVER キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwsgi	rwserver
○	×	×	×	○	○	○

説明 SERVER を使用して、このレポートを実行する Reports Server の名前を指定できます。

構文 SERVER=*server_name*

値 構文を参照してください。

デフォルト rwsgi の REPORTS_SERVER 環境変数で指定したサーバー名。

使用上の注意

- rwsgi で実行されるジョブについては、Web サーバー・マシン上で REPORTS_SERVER 環境変数を設定して、SERVER キーワードを省略すると、リクエストがデフォルト・サーバーを使用して処理されます。SERVER キーワードを含めるとデフォルトよりも優先されます。rwervlet で実行されるか JSP として実行されるジョブについては、デフォルト・サーバーをサーブレット構成ファイル (rwervlet.properties) で指定している場合は、SERVER キーワードを省略できます。あるいは、SERVER キーワードを含めてデフォルトよりも優先させることができます。
- SERVER は、コマンドラインから JSP ベースの Web レポートを実行する場合に使用できません。

A.3.96 SHOWAUTH

表 A-106 では、SHOWAUTH キーワードを使用できる実行可能ファイルを示します。

表 A-106 SHOWAUTH キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwsgi	rwserver
×	×	×	×	○	○	×

説明 SHOWAUTH を使用して、Reports Server ログイン・ページを表示し、レポートを実行できます。

構文 `http://your_webserver/reports/rwervlet/showauth[?] [server=server_name] [&authid=username/password] [&nextpage=run_report_URL] [&authtype={s|d}]`

値 構文を参照してください。

デフォルト なし

使用上の注意

- このキーワードは値を必要としないコマンドです。つまり、対応する値を持たず、コマンドだけが入力されます。
- authtype=s の場合、Reports の「システム・ユーザー認証」ダイアログ・ボックスが表示されます。authtype=d の場合、Reports の「データベース・ユーザー認証」ダイアログ・ボックスが表示されます。
- 関連するキーワードは、SERVER および AUTHID です。

A.3.97 SHOWENV

表 A-107 では、SHOWENV キーワードを使用できる実行可能ファイルを示します。

表 A-107 SHOWENV キーワードを使用する実行可能ファイル

rwclient	rwrund	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
×	×	×	×	○	○	×

説明 SHOWENV を使用して、rwervlet 構成ファイル (rwervlet.properties) を表示できます。

構文 `http://your_webserver/reports/rwervlet/showenv[?] [server=server_name] [&authid=username/password]`

値 構文を参照してください。

デフォルト なし

使用上の注意

- このキーワードは値を必要としないコマンドです。つまり、対応する値を持たず、コマンドだけが入力されます。
- 関連するキーワードは、[SERVER](#) および [AUTHID](#) です。

A.3.98 SHOWJOBID

表 A-108 では、SHOWJOBID キーワードを使用できる実行可能ファイルを示します。

表 A-108 SHOWJOBID キーワードを使用する実行可能ファイル

rwclient	rwrund	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
×	×	×	×	○	×	×

説明 SHOWJOBID を使用して、Reports Server のジョブ・ステータスとジョブ ID *n* を表示できます。

構文 `http://your_webserver/reports/rwervlet/showjobidn[?] [server=server_name] [&authid=username/password] [&statusformat={html|xml|xmldtd}]`

値 構文を参照してください。

デフォルト なし

使用上の注意

- このキーワードは値を必要としないコマンドです。つまり、対応する値を持たず、コマンドだけが入力されます。
- ジョブは現行である (キューに入っているかスケジュールされている) 必要があります。
- 現在のジョブ・リストを参照するには、SHOWJOBS を使用します。STATUSFORMAT は html (デフォルト)、xml または xmldtd に設定できます。指定した形式でステータスが返されます。ステータス情報は、HTML、XML または XMLDTD (DTD の内部サブセット) に生成されます。
- 関連するキーワードは、[SHOWJOBS](#)、[SERVER](#)、[AUTHID](#) および [STATUSFORMAT](#) です。
- STATUSFORMAT パラメータは、rwervlet に対してのみ使用でき、rwcgi には使用できません。

A.3.99 SHOWJOBS

表 A-109 では、SHOWJOBS キーワードを使用できる実行可能ファイルを示します。

表 A-109 SHOWJOBS キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
×	×	×	×	○	○	×

説明 SHOWJOBS を使用して、Reports Server のキュー・ステータスを Web 形式で表示できます。

構文 `http://your_webserver/reports/rwervlet/showjobs[n][?][server=server_name][&authid=username/password][&queuetype={current|past|future}][&startrow=start_position_in_job_queue][&count=number_of_jobs_to_display][&statusformat={html|xml|xmldtd}]`

値 構文を参照してください。

デフォルト なし

使用上の注意

- このキーワードに値は必要ありません。つまり、対応する値を指定せず、キーワードのみで入力します。
- Reports Server の名前が、環境変数またはサーブレット構成ファイルによって暗黙的に指定されているか、URL リクエストで明示的に指定されている必要があります。リフレッシュ数 *n* はオプションです。この値を指定すると、レポートのキュー・ステータスが *n* 秒ごとに更新されます。
- STATUSFORMAT は html (デフォルト)、xml または xmldtd に設定できます。指定した形式でステータスが返されます。ステータス情報は、HTML、XML または XMLDTD (DTD の内部サブセット) に生成されます。
- QUEUETYPE、STARTROW、COUNT および STATUSFORMAT パラメータは、rwervlet に対してのみ使用でき、rwcgi には使用できません。
- 関連するキーワードは、[SERVER](#)、[AUTHID](#) および [STATUSFORMAT](#) です。

A.3.100 SHOWMAP

表 A-110 では、SHOWMAP キーワードを使用できる実行可能ファイルを示します。

表 A-110 SHOWMAP キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
×	×	×	×	○	○	×

説明 SHOWMAP を使用して、rwervlet のキー・マッピングを表示できます。

構文 `http://your_webserver/reports/rwervlet/showmap[?][server=server_name][&authid=username/password]`

値 構文を参照してください。

デフォルト なし

使用上の注意

- このキーワードは値を必要としないコマンドです。つまり、対応する値を持たず、コマンドだけが入力されます。
- 関連するキーワードは、[SERVER](#) および [AUTHID](#) です。

A.3.101 SHOWMYJOBS

表 A-111 では、SHOWMYJOBS キーワードを使用できる実行可能ファイルを示します。

表 A-111 SHOWMYJOBS キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
×	×	×	×	○	×	×

説明 SHOWMYJOBS を使用して、特定ユーザーに対する Reports Server のキュー・ステータスを表示できます。

構文 `http://your_webserver/reports/rwervlet/showmyjobs [?] [server=server_name] [&authid=username/password] [&statusformat={html|xml|xmldtd}]`

値 構文を参照してください。

デフォルト なし

使用上の注意

- このキーワードは値を必要としないコマンドです。つまり、対応する値を持たず、コマンドだけが入力されます。
- STATUSFORMAT は `html` (デフォルト)、`xml` または `xmldtd` に設定できます。指定した形式でステータスが返されます。ステータス情報は、`html`、`xml` または `xmldtd` (dtd の内部サブセット) に生成されます。
- 関連するキーワードは、[SERVER](#)、[AUTHID](#) および [STATUSFORMAT](#) です。
- STATUSFORMAT パラメータは、`rwervlet` に対してのみ使用でき、`rwcgi` には使用できません。

A.3.102 SHUTDOWN

表 A-112 では、SHUTDOWN キーワードを使用できる実行可能ファイルを示します。

表 A-112 SHUTDOWN キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver	rwbridge
×	×	×	×	×	×	○	○

説明 SHUTDOWN を使用して、すでに実行中のサーバーを停止したり、Oracle Reports ブリッジを停止します。また、`rwserver` とともに使用する場合は、[AUTHID](#) を使用してユーザー名とパスワードを指定する必要があります。ブリッジがセキュリティで保護されている場合 (`identifier` 要素がブリッジ構成ファイルで設定されている場合)、`rwbridge` とともに使用するときは、[AUTHID](#) も使用してユーザー名とパスワードを指定する必要があります。

構文 `SHUTDOWN={NORMAL|IMMEDIATE}`

値

- NORMAL の場合、通常の停止プロシージャを使用して、サーバーまたはブリッジを正常に停止します。
- IMMEDIATE の場合、他のプロセスが実行を完了するまで待機せずに、サーバーまたはブリッジをただちに停止します。

デフォルト NORMAL

使用上の注意 SHUTDOWN キーワードを使用するユーザーは、Oracle Reports 管理ユーザーである必要があります。サーバーのセキュリティが有効な場合、サーバーは security API に問合せを行い、ユーザーのロールが停止を実行するのに適切かどうか（つまり、ユーザーが Oracle Reports 管理ユーザーかどうか）を判別します。セキュリティが有効でない場合も、ユーザーは、そのサーバーで定義された Oracle Reports 管理ユーザーである必要があります。

A.3.103 SITENAME

表 A-113 では、SITENAME キーワードを使用できる実行可能ファイルを示します。

表 A-113 SITENAME キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwsgi	rwserver
○	○	×	×	○	○	×

説明 SITENAME を使用して、レポート出力の配信先となる Oracle WebDB リリース 2.2 サイトの名前を指定します。このキーワードは、Oracle WebDB リリース 2.2 に対する下位互換性を確保するために維持されています。Oracle9iAS Portal リリース 1 に対する下位互換性は、「CONTENTAREA」を参照してください。OracleAS Portal 10g (9.0.4) 以降については、PAGEGROUP を使用します。

構文 SITENAME=*name*

値

name には、Oracle WebDB リリース 2.2 で使用されている有効なサイト名を指定します。

デフォルト なし

使用上の注意

- Oracle Reports 出力を Oracle WebDB リリース 2.2 に配信する場合、このキーワードの使用は必須です。
- 関連するキーワードは、CONTENTAREA*、EXPIREDAYS、ITEMTITLE、OUTPUTFOLDER*、OUTPUTPAGE、PAGEGROUP、SITENAME*、STATUSFOLDER*、STATUSPAGE です。

* Oracle9iAS Portal リリース 1 および Oracle WebDB リリース 2.2 に対する下位互換性を確保するために維持されています。

A.3.104 SOURCE

表 A-114 では、SOURCE キーワードを使用できる実行可能ファイルを示します。

表 A-114 SOURCE キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwsgi	rwserver
×	×	×	○	×	×	×

説明 SOURCE を使用して、変換する 1 つのレポートまたはライブラリ、あるいはそれらのリストを指定できます。rwconverter コマンドを実行するには、ソース・レポートまたはソース・ライブラリを指定する必要があります。

構文 SOURCE={source_name | (source_name1, source_name2, ...)}

値 有効なレポート名、ライブラリ名またはファイル名、あるいは名前をカンマで区切り、括弧で囲んだそれらのリスト ((qanda, test, dmast) など) を指定します。

デフォルト なし

使用上の注意

- データベースに格納されているレポートまたはライブラリには、SQL のワイルドカード文字 (% および _) を使用できます。たとえば、R% では、データベースに格納されている R で始まるすべてのレポートがフェッチされます。一致するすべてのレポートが変換されます。
- レポート名、ライブラリ名、ファイル名のリストは、名前をカンマで区切り、括弧で囲む必要があります。次に例を示します。
(qanda, test, dmast) または (qanda, test, dmast)
- ワイルドカード文字は、ファイルに保存されているレポートまたはライブラリ (.rdf, .rep, .rex, .pld, .pll または .xml 拡張子を持つもの) に対しては無効です。
- SOURCE キーワードの値は、オペレーティング・システムによって異なります。
- ユーザー所有の Reports Builder 表を使用する場合、複数のユーザーからのレポートまたはライブラリは、各ユーザー用に個別に変換する必要があります。
- レポートまたはライブラリを変換するには、これらを自分で作成済であるか、または自分が作成したものでない場合は、アクセス権限が付与されている必要があります。レポートまたはライブラリ名の前に USERID 文字列が付いていない場合、USERID 文字列は現行のユーザーと想定されます。
- DTYPE=REGISTER である場合、宛先タイプや出力フォーマット、ユーザー・アクセス、および使用可能日カレンダーなどのパラメータが共通のレポート定義ファイルのみを指定する必要があります。

A.3.105 SQLTRACE

表 A-115 では、SQLTRACE キーワードを使用できる実行可能ファイルを示します。

表 A-115 SQLTRACE キーワードを使用する実行可能ファイル

rwclient	rwrund	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
○	○	○	×	○	○	×

説明 SQLTRACE を使用して、レポート定義を変更せずに、レポートに SQL トレースを実行するかどうかを指定できます。

構文 SQLTRACE= [YES | NO]

値

- YES の場合、レポートに SQL トレースが実行されます。
- NO の場合、レポートに SQL トレースは実行されません。

デフォルト NO

A.3.106 SSOCONN

表 A-116 では、SSOCONN キーワードを使用できる実行可能ファイルを示します。

表 A-116 SSOCONN キーワードを使用する実行可能ファイル

rwclient	rwrwn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
×	×	×	×	○	×	×

説明 SSOCONN を使用して、Single Sign-On 環境で 1 つ以上のデータソースへの接続に使用する 1 つ以上の接続文字列を指定できます。

構文 SSOCONN=key[/type[/conn_string_parameter]][,key[/type[/conn_string_parameter]]]

値 SSOCONN 構文で表される変数値に関する情報を次に示します。

- key は、Oracle Internet Directory に格納される接続文字列値を参照します。
- type は、接続するデータソースの種類であり、key と関連付けられた文字列の書式を識別します。type 値は、Oracle Internet Directory に格納されている有効なリソース・タイプである必要があります。Oracle Reports では、次のものに対してデフォルトのリソース・タイプを提供しています。
 - Oracle データベース (OracleDB)
 - JDBC (JDBCPS)
 - Oracle Express (EXPRESSPS)
- conn_string_parameter は、Oracle Reports システムの名前、または接続文字列を rwervlet に渡してレポートを実行するために使用されるユーザー・パラメータです。たとえば、OracleDB データソースの場合、Oracle Reports は USERID パラメータを介して接続文字列を受け取り、これを使用して、指定された Oracle データベースに接続します。同様に、EXPRESSPS の場合は EXPRESS_SERVER パラメータが使用され、JDBCPS の場合は P_JDBCPS が使用されます。独自のプラグابل・データ・ソースがある場合、Oracle Reports に接続文字列を渡すための独自のユーザー・パラメータを定義して、それを SSOCONN の conn_string_parameter として指定する必要があります。

次に例を示します。

```
SSOCONN=mykey/OracleDB/USERID
```

デフォルト なし

使用上の注意

- レポートに複数のデータソースが使用されている場合、データソース接続文字列をカンマで区切ります。次に例を示します。

```
ssconn=key1/type1/conn_str,key2/type2/conn_str2,key3/type3/conn_str3
```

- SSOCONN をコマンドラインで使用する場合は、次のことが行えません。
 - 同一のコマンドラインで AUTHID を指定する。
 - セキュアでない Reports Server に対して実行する。
 - rwervlet.properties で SINGLESIGNON=NO に設定する。
- 表 A-117 に示すように SSOCONN の簡易バージョンも使用できます。

表 A-117 SSOCONN オプションの簡易バージョン

オプション	説明
ssocnn=mkey	キー名のみを指定した場合、 <i>type</i> と <i>conn_string_parameter</i> のデフォルト値は、OracleDB と USERID です。
ssocnn=mkey/PDSApp	キー名とデータソース・タイプの両方を指定した場合、 <i>conn_string_parameter</i> のデフォルト値は USERID です。

- SSOCONN は、コマンドラインから JSP ベースの Web レポートを実行している場合にも使用できます。JSP として実行するレポートに SSOCONN キーワードを使用する場合、アンパサンド (&) を使用して、接続文字列を区切ります。次に例を示します。

http://...:8888/myjsp/foo.jsp?name1=value1&name2=value2 ...

注意： Oracle Reports と Single Sign-On (SSO) の詳細は、第 11 章「OracleAS Single Sign-On の構成と管理」を参照してください。

A.3.107 STATUSFOLDER

表 A-118 では、STATUSFOLDER キーワードを使用できる実行可能ファイルを示します。

表 A-118 STATUSFOLDER キーワードを使用する実行可能ファイル

rwclient	rwrwn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
○	○	×	×	○	○	×

説明 STATUSFOLDER を使用して、ジョブ・ステータス情報の配信先となる Oracle WebDB リリース 2.2 または Oracle9iAS Portal リリース 1 のフォルダの名前を指定できます。この指定を省略すると、Oracle_Reports_Status という新規フォルダが作成されます。このキーワードは、Oracle WebDB リリース 2.2 および Oracle9iAS Portal リリース 1 に対する下位互換性を確保するために維持されています。OracleAS Portal 10g (9.0.4) 以降では、STATUSPAGE を使用してください。

構文 STATUSFOLDER=*folder*

値

folder には、Oracle WebDB リリース 2.2 または Oracle9iAS Portal リリース 1 で使用される有効なフォルダ名（内部名）を指定します。

デフォルト Oracle_Reports_Status

使用上の注意

- Oracle Reports 出力を Oracle WebDB リリース 2.2 または Oracle9iAS Portal リリース 1 に配信する際、このキーワードの使用はオプションです。
- このキーワードの値は大文字と小文字を区別します。
- 関連するキーワードは、CONTENTAREA*、EXPIREDAYS、ITEMTITLE、OUTPUTFOLDER*、OUTPUTPAGE、PAGEGROUP、SITENAME*、STATUSFOLDER*、STATUSPAGE です。

* Oracle9iAS Portal リリース 1 および Oracle WebDB リリース 2.2 に対する下位互換性を確保するために維持されています。

A.3.108 STATUSFORMAT

表 A-119 では、STATUSFORMAT キーワードを使用できる実行可能ファイルを示します。

表 A-119 STATUSFORMAT キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwsgi	rwserver
×	×	×	×	○	×	×

説明 STATUSFORMAT を使用して、Reports Server のキュー・ステータスを Web で表示する際の形式を指定できます。

構文 `http://yourwebserver/rwervlet/showjobs?server=server_name&statusformat={html|xml|xmldtd}`

値

- html の場合、Reports Server のキュー・ステータスを HTML 形式で出力します。
- xml の場合、Reports Server のキュー・ステータスを XML 形式で出力します。
- xmldtd の場合、Reports Server のキュー・ステータスをインライン・データ型定義情報を持つ XML 形式で出力します。

デフォルト html

使用上の注意 STATUSFORMAT は、SHOWJOBS および SHOWMYJOBS キーワードとともに使用します。

A.3.109 STATUSPAGE

表 A-120 では、STATUSPAGE キーワードを使用できる実行可能ファイルを示します。

表 A-120 STATUSPAGE キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwsgi	rwserver
○	○	×	×	○	○	×

説明 STATUSPAGE を使用して、ジョブ・ステータス情報の配信先となる OracleAS Portal ページの名前を指定できます。この指定を省略すると、Oracle_Reports_Status という新規ページが作成されます。旧リリース (Oracle WebDB リリース 2.2 および Oracle9iAS Portal リリース 1) に対する下位互換性については、「STATUSFOLDER」を参照してください。

構文 STATUSPAGE=*page*

値

page には、OracleAS Portal で使用される有効なページ名 (内部名) を指定します。

デフォルト Oracle_Reports_Status

使用上の注意

- 出力を OracleAS Portal に配信する際、このキーワードの使用はオプションです。
- このキーワードの値は大文字と小文字を区別します。

- 関連するキーワードは、[CONTENTAREA*](#)、[EXPIREDAYS](#)、[ITEMTITLE](#)、[OUTPUTFOLDER*](#)、[OUTPUTPAGE](#)、[PAGEGROUP](#)、[SITENAME*](#)、[STATUSFOLDER*](#)、[STATUSPAGE](#) です。

* Oracle9iAS Portal リリース 1 および Oracle WebDB リリース 2.2 に対する下位互換性を確保するために維持されています。

A.3.110 STYPE

表 A-121 では、STYPE キーワードを使用できる実行可能ファイルを示します。

表 A-121 STYPE キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
×	×	×	○	×	×	×

説明 STYPE を使用して、変換するレポートまたはライブラリの形式を指定できます。

構文 STYPE={PLDFILE|PLLFILE|RDDFILE|REXFILE|XMLFILE|JSPFILE}

値 次の値をどれか 1 つ指定します。

- PLDFILE の場合、ソース PL/SQL ライブラリは ASCII 形式のファイルに格納されています。
- PLLFILE の場合、ソース PL/SQL ライブラリは、ソース・コードおよび P コード（コンパイル済 PL/SQL）を含むファイルに格納されています。
- RDDFILE の場合、ソース・レポートは 1 つ以上のレポート定義ファイル（拡張子 rdf を持つファイル）に格納されています。
- REXFILE の場合、ソース・レポートは 1 つ以上のテキスト・ファイル（拡張子 rex を持つファイル）に格納されています。
- XMLFILE の場合、ソース・レポートは 1 つ以上の XML ファイルに格納されています。
- JSPFILE の場合、ソース・レポートは 1 つ以上の JSP ファイルに格納されています。

デフォルト RDDFILE

使用上の注意 DTYPE=REGISTER である場合、STYPE には RDDFILE、REXFILE、XML または JSPFILE を選択します。

A.3.111 SUBJECT

表 A-122 では、SUBJECT キーワードを使用できる実行可能ファイルを示します。

表 A-122 SUBJECT キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
○	○	×	×	○	○	×

説明 SUBJECT を使用して、電子メールの件名を指定できます。

構文 SUBJECT=*string*

値 任意のテキスト文字列。

デフォルト なし

使用上の注意

- 文字スペースを含む件名は、引用符 (" ") で囲みます。単一語句の件名には引用符は不要です。
- 関連するキーワードは、**BCC**、**CC**、**FROM**、**REPLYTO** および **SUBJECT** です。電子メールの主な宛先を指定するには、**DESNAME** を使用します。

A.3.112 SUPPRESSLAYOUT

表 A-123 では、SUPPRESSLAYOUT キーワードを使用できる実行可能ファイルを示します。

表 A-123 SUPPRESSLAYOUT キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rservlet	rwsgi	rwserver
○	○	○	×	○	○	×

説明 SUPPRESSLAYOUT を使用して、実行時にペーパー・レイアウトのフォーマットを実行するかどうかを指定します。このキーワードにより、レポートの実行時にペーパー・レイアウトを実行させるかどうかを制御できます。このキーワードは通常、JSP レポートのパフォーマンスを向上させる場合に使用します。JSP レポートにペーパー・レイアウトがある場合、そのレイアウト内のオブジェクトは <rw:include> タグで参照されるため、Oracle Reports は、レポートの JSP セクションを実行する前に、ペーパー・レイアウトをフォーマットします。ペーパー・レイアウトと Web レイアウトの両方が保存されているが、ペーパー・レイアウト・オブジェクトが参照されない単一のソース JSP レポートのパフォーマンスを向上させる場合は、コマンドラインで SUPPRESSLAYOUT=YES に設定します。

注意： <rw:include> タグがある場合、そのタグの出力は作成されません。

構文 SUPPRESSLAYOUT= [YES | NO]

値

- YES の場合、実行時にペーパー・レイアウト・オブジェクトはフォーマットされません。
- NO の場合、実行時にペーパー・レイアウト・オブジェクトはフォーマットされます。

デフォルト NO

A.3.113 TOLERANCE

表 A-124 では、TOLERANCE キーワードを使用できる実行可能ファイルを示します。

表 A-124 TOLERANCE キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rservlet	rwsgi	rwserver
○	×	×	×	○	○	×

説明 TOLERANCE を使用して、重複したジョブが検出された場合の、レポートのキャッシュ出力の再利用に対する最大許容時間（分）を設定できます。レポートについて許容時間を設定すると、重複したジョブが見つかった場合の処理時間が短縮されます。

重複ジョブの検出の詳細は、「[キャッシュからのレポート出力の再利用](#)」を参照してください。

構文 TOLERANCE=*time_string*

値

`time_string` には、次のいずれかの形式を使用できます。

- `n{unit}` は、オプションの単位を持つ数値の場合に使用します。`unit` には、`minute(s)`、`hour(s)` または `day(s)` を指定できます。単位を指定しない場合は、デフォルトの `minute(s)` が使用されます。
- `{Mon DD, YYYY} hh:mi:ss am|pm {timezone}` が、日付 / 時刻の書式として使用されます。日付情報はオプションです。日付を指定しない場合は、本日の日付が使用されます。タイムゾーンもオプションです。タイムゾーンを指定しない場合は、`Reports Server` のタイムゾーンが使用されます。日付 / 時刻の書式には必ず米国のロケールを使用します。この形式は、Java の `DateFormat.MEDIUM` タイプの定義と同じです。

デフォルト なし

使用上の注意

- `TOLERANCE` が設定されていないときは、キャッシュに重複レポートが見つかった場合でも、`OracleAS Reports Services` はそのレポートを再実行します。
- 同一ジョブが送信されたときにレポートが処理中（つまり、現行ジョブ・キューにあるとき）であった場合は、`TOLERANCE` が設定されていない場合やゼロに設定されている場合でも、`OracleAS Reports Services` は現在実行中のジョブの出力を再利用します。

A.3.114 TRACEFILE

表 A-125 では、`TRACEFILE` キーワードを使用できる実行可能ファイルを示します。

表 A-125 TRACEFILE キーワードを使用する実行可能ファイル

<code>rwclient</code>	<code>rwrun</code>	<code>rwbuilder</code>	<code>rwconverter</code>	<code>rwervlet</code>	<code>rwcgi</code>	<code>rwserver</code>
×	○	○	×	×	×	○

説明 `TRACEFILE` は、トレース情報を書き込むファイルの名前です。

注意： レポート・トレースは、後述の「使用上の注意」および第 20.1.2 項「レポート・トレース」の説明のように様々な方法で指定できます。

構文 `TRACEFILE=tracefile`

値 任意の有効なファイル名

デフォルト `rwserver.trc` または `rwEng-x.trc`（「使用上の注意」を参照）

使用上の注意

- コマンドラインで指定したトレース・オプションは、構成ファイルの設定より優先されません。
- `Reports Builder (rwbuilder)` および `Reports Runtime (rwrun)` の場合、トレース・オプション（第 3.2.1.13 項「`trace`」で説明）は、ビルダー構成ファイル（`rwbuilder.conf`）で構成するか、コマンドラインで指定できます。

注意： トレース・ファイルの位置は、`Oracle Reports` ログ・ディレクトリ（`ORACLE_HOME\reports\logs\rep_machinename-rwbuilder¥`）に対する相対パスか、または絶対パス（フルパス名が指定されている場合）です。トレース・ファイル名を指定しない場合、デフォルトのビルダー・トレース・ファイル名は `rwserver.trc` です。

- Reports Server (rwservlet) の場合、トレース・オプション (第 3.2.1.13 項「trace」で説明) は、サーバー構成ファイル (server_name.conf) で構成するか、サーバーの起動時にコマンドラインで指定できます。

注意： トレース・ファイルの位置は、サーバー・ログ・ディレクトリ (ORACLE_HOME¥reports¥logs¥server_name) に対する相対パスか、または絶対パス (フルパス名が指定されている場合) です。トレース・ファイル名を指定しない場合、デフォルトのサーバー・トレース・ファイル名は rwservlet.trc、デフォルトのエンジン・トレース・ファイル名は rwEng-x.trc (x はエンジン ID) です。

- Reports Servlet (rwservlet) の場合、トレース・オプションは、第 3.4.5 項「Reports Servlet と JSP のトレース・オプションの設定」の説明のようにサーブレット構成ファイル (rwservlet.properties) で構成されます。
- 既存のファイルが上書きまたは追加されるかどうかは、TRACEMODE の設定によって決まります。

A.3.115 TRACEMODE

表 A-126 では、TRACEMODE キーワードを使用できる実行可能ファイルを示します。

表 A-126 TRACEMODE キーワードを使用する実行可能ファイル

rwclient	rwrun	rwbuilder	rwconverter	rwservlet	rwcgi	rwservlet
×	○	○	×	×	×	○

説明 TRACEMODE には、新規トレース情報を、TRACEFILE で指定されたファイルの既存情報に追加するか、ファイル全体を上書きするかを指定します。

注意： レポートのトレースを指定する方法はいくつかあります (第 20.1.2 項「レポート・トレース」を参照)。

構文 TRACEMODE={TRACE_APPEND|TRACE_REPLACE}

値

- TRACE_APPEND は、新しい情報をファイルの末尾に追加します。
- TRACE_REPLACE は、ファイルを上書きします。

デフォルト TRACE_APPEND

使用上の注意

- 第 A.3.114 項「TRACEFILE」の「使用上の注意」を参照してください。

A.3.116 TRACEOPTS

表 A-127 では、TRACEOPTS キーワードを使用できる実行可能ファイルを示します。

表 A-127 TRACEOPTS キーワードを使用する実行可能ファイル

rwclient	rwrun	rwbuilder	rwconverter	rwservlet	rwcgi	rwservlet
○	○	○	×	○	○	○

説明 TRACEOPTS は、TRACEFILE で指定されるファイルに記録するトレース情報を指定します。

注意: レポートのトレースを指定する方法はいくつかあります (第 20.1.2 項「レポート・トレース」を参照)。

構文 TRACEOPTS={TRACE_ALL|TRACE_APP|TRACE_BRK|TRACE_DBG|TRACE_DST|TRACE_ERR|TRACE_EXC|TRACE_INF|TRACE_LOG|TRACE_PLS|TRACE_PRF|TRACE_SQL|TRACE_STA|TRACE_TMS|TRACE_WRN}

値 次の値が適用されます。

- TRACE_ALL の場合、トレース・ファイルにすべてのトレース情報を記録します。
- TRACE_APP の場合、トレース・ファイルにすべてのレポート・オブジェクトのトレース情報を記録します。
- TRACE_BRK の場合、トレース・ファイルにすべてのブレーク・ポイントを記録します。
- TRACE_DBG の場合、デバッグ情報を記録します。
- TRACE_DST の場合、トレース・ファイルに宛先リストを記録します。この情報を使用して、どのセクションがどの宛先に送られたかを調べることができます。
- TRACE_ERR の場合、トレース・ファイルにエラー・メッセージと警告を記録します。
- TRACE_EXC の場合、Reports Server の例外を記録します。
- TRACE_INF の場合、他のオプションでトレース・ファイルに記録されないすべての情報を記録します。
- TRACE_LOG の場合、トレース・ファイルのログ情報を複製します。サーバー構成ファイルで trace 要素のほかに log 要素も指定した場合、ログ・ファイルに出力された情報がトレース・ファイルにも送信されます。
- TRACE_PLS の場合、トレース・ファイルにすべての PL/SQL オブジェクトのトレース情報を記録します。
- TRACE_PRF の場合、トレース・ファイルに、パフォーマンス統計を記録します。
- TRACE_SQL の場合、トレース・ファイルにすべての SQL に関するトレース情報を記録します。
- TRACE_STA の場合、初期化中、準備完了、実行中および停止など、サーバーおよびエンジンの状態に関する情報を記録します。
- TRACE_TMS の場合、トレース・ファイルの各エントリに対してタイムスタンプを入力します。
- TRACE_WRN の場合、サーバーの警告メッセージを記録します。

デフォルト TRACE_ALL

使用上の注意

- 第 A.3.114 項「TRACEFILE」の「使用上の注意」を参照してください。
- 複数のオプションを使用する場合は、オプションを括弧内に指定します。たとえば、TRACEOPTS=(TRACE_APP, TRACE_PRF) は、TRACE_APP と TRACE_PRF の両方を適用することを意味します。

A.3.117 UPGRADE_PLSQL

表 A-128 では、UPGRADE_PLSQL キーワードを使用できる実行可能ファイルを示します。

表 A-128 UPGRADE_PLSQL キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwsgi	rwserver
×	×	×	○	×	×	×

説明 UPGRADE_PLSQL を使用して、レポート内の PL/SQL コードを、Oracle Reports に必要な最新リリースにアップグレードするかどうかを指定できます。

構文 UPGRADE_PLSQL=[YES|NO]

値

- YES の場合、PL/SQL コードは必要に応じて自動的にアップグレードされます。
- NO の場合、PL/SQL コードは更新されません。

デフォルト YES

A.3.118 URLPARAMETER

表 A-129 では、URLPARAMETER キーワードを使用できる実行可能ファイルを示します。

表 A-129 URLPARAMETER キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwsgi	rwserver
○	×	×	×	○	○	×

説明 URLPARAMETER を使用して、URL エンジンでフェッチする URL を指定できます。

構文 URLPARAMETER=http://your_webserver/page_name.html

値 有効な URL

デフォルト なし

使用上の注意 このキーワードは、Reports Server 構成ファイルの **job** 要素で `jobType=rwurl` が指定されていて、かつ URL エンジンが存在する場合に使用できます。

A.3.119 USEJVM

表 A-130 では、USEJVM キーワードを使用できる実行可能ファイルを示します。

表 A-130 USEJVM キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwsgi	rwserver
○	×	×	×	×	×	×

説明 USEJVM を使用して、`rwclient` が Reports Server との通信に Java 仮想マシン (JVM) を使用するかどうかを指定します。

構文 USEJVM=YES|NO

値

- YES の場合、rwclient は JVM を起動し、CORBA を使用して Reports Server に接続を試みます。これに失敗した場合は、下位互換性のために維持されている SQLNet を使用して接続を試みます。
- NO の場合、rwclient は JVM を起動しません。かわりに、SQLNet を使用して Reports Server (rwproxy または Oracle Reports 6i サーバー) と通信します。

デフォルト YES

A.3.120 USERID

表 A-131 では、USERID キーワードを使用できる実行可能ファイルを示します。

表 A-131 USERID キーワードを使用する実行可能ファイル

rwclient	rwrwn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
○	○	○	○	○	○	×

説明 USERID は、Single Sign-On を使用していない場合にのみ、Oracle ユーザー名およびパスワードを、リモート・データベースにアクセスするためのオプションのデータベース名とともに指定するために使用します。パスワードを省略すると、ユーザーにレポートの実行が許可される前に、データベース・ログイン・フォームが自動的に表示されます。

ユーザーのデータベースへのログインを実施させるには、レポート・リクエストから USERID キーワードのパスワード部分を省いてください。レポート・リクエストを実行するたびにユーザーにログインさせるようにするには、Oracle Reports のキー・マップ・ファイル cgicmd.dat を使用してランタイム・コマンドを指定し、%D オプションを関連のキー・マップ・エントリに含めます。

注意: cgicmd.dat ファイルの使用方法の詳細は、第 13 章「レポート・リクエストの実行」を参照してください。

構文 `userid=username [/password] [@database]`

値

- `username` には、データベース管理者によって割り当てられているユーザー名を指定します。
- `password` には、ユーザー名に対するパスワードを指定します。後述の「使用上の注意」を参照してください。
- `database` には、アクセスするデータベースの名前を指定します。

デフォルト なし

使用上の注意

- ログイン定義は 512 バイトを超えることはできません。
- USERID は、コマンドラインから JSP ベースの Web レポートを実行する場合にも使用できます。
- USERID を rwbuilder、rwrwn、rwclient または rwconverter と使用する場合は、パスワードを含めないことを強くお勧めします。多くのオペレーティング・システムでは、この情報はどのようなユーザーでも入手できます (たとえば、UNIX では ps コマンドで可能)。かわりに、SSOCONN キーワードを使用してください。

A.3.121 USERSTYLES

表 A-132 では、USERSTYLES キーワードを使用できる実行可能ファイルを示します。

表 A-132 USERSTYLES キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
○	○	○	×	○	○	×

説明 USERSTYLES を使用して、HTMLCSS 出力を生成するときに、外部スタイル・シート・ファイル (.css) をレポートと関連付けるかどうかを指定します。レポートに適用するスタイル・シートは、レポートの「スタイル・シート」プロパティで指定します。この値は、デフォルトでは YES に設定されています。また、この値は、ペーパー・デザイン・レイアウトに組み込まれた設計時スタイルより優先されます。

関連項目： インライン HTML 形式の詳細は、『Oracle Reports レポート作成ガイド』を参照してください。

構文 USERSTYLES=YES|NO

値

- YES の場合、HTMLCSS 出力を生成するときに、レポートの「スタイル・シート」プロパティで指定された 1 つ以上の外部スタイル・シートとレポートを関連付けます。
- NO の場合、レポートの設計時に適用された形式がレポートと関連付けられます。外部スタイル・シートは無視されます。

デフォルト YES

使用上の注意

- YES と NO 以外の値を指定した場合、Oracle Reports ではデフォルトの YES になります。
- レポートが外部スタイル・シートと関連付けられていないと判明した場合、次の事項を確認します。
 - 「スタイル・シート」プロパティでスタイル・シートへの正しいパスを指定してある。
 - 「CSS クラス名」および「CSS ID」プロパティで指定されたスタイルが、指定されたスタイル・シートで定義されている。

関連項目： 「スタイル・シート」、「CSS クラス名」および「CSS ID」プロパティの詳細は、Oracle Reports のオンライン・ヘルプを参照してください。

- USERSTYLES が YES に設定されている。

例

```
http://myias.mycomp.com:7779/reports/rwervlet?server=myrepserver+report=test.jsp+userstyles=yes
```

A.3.122 VALIDATETAG

表 A-133 では、VALIDATETAG キーワードを使用できる実行可能ファイルを示します。

表 A-133 VALIDATETAG キーワードを使用する実行可能ファイル

rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
×	×	○	×	×	×	×

説明 VALIDATETAG は、JSP ベースの Web レポートの設計時または配布時に、JSP タグの検証を強制し、重複フィールドの ID や不正な属性などの項目をチェックするかどうかを指定します。

関連項目： VALIDATETAG を使用したレポートのパフォーマンスのチューニングの詳細は、[第 20.8 項「レポートの実行」](#) を参照してください。

構文 VALIDATETAG=YES|NO

値

- YES の場合、タグ検証が強制され、重複フィールドの ID や不正な属性などの項目がチェックされます。
- NO の場合、タグの検証は行われません。

デフォルト

- 設計時に、Reports Builder から JSP ベースの Web レポートを実行する場合は YES。
- 実行時に、JSP ベースの Web レポートを配布する場合は NO。

使用上の注意

- この機能は、本番環境ではなく、設計フェーズでのみ有用です。デフォルトでは、Reports Builder でレポートの設計中は VALIDATETAG=YES であり、OracleAS Reports Services でレポートの配布の場合は VALIDATETAG=NO です。レポートを配布するときこのオプションをチューニングするには、http リクエストで VALIDATETAG=YES を指定します (http://my.server.com/myreport.jsp?VALIDATETAG=YES など)。
- レポートを配布するとき VALIDATETAG=YES を使用すると、パフォーマンスが低下します。
- Reports Builder をコマンドラインから `rwbuilder VALIDATETAG=NO` を指定して起動すると、無効な JSP タグ構造を含むレポートが設計されるおそれがあります。

A.3.123 WEBSERVER_DEBUG

[表 A-134](#) では、WEBSERVER_DEBUG キーワードを使用できる実行可能ファイルを示します。

表 A-134 WEBSERVER_DEBUG キーワードを使用する実行可能ファイル

rwclient	rwrund	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
×	×	○	×	×	×	×

説明 WEBSERVER_DEBUG を使用して、JSP のデバッグを実行できます。このキーワードを使用すると、`stderr.log` ファイルおよび `stdout.log` ファイルが `docroot/port#` ディレクトリに作成され、JSP 一時ファイルは `docroot/port#/default` に、ログ・ファイルは `docroot/port#/log` に残されるので、ユーザーが調べることができます。

構文 WEBSERVER_DEBUG={YES|NO}

値

- YES の場合、デバッグ・ファイルが作成されます。
- NO の場合、デバッグ・ファイルは作成されません。

デフォルト NO

使用上の注意

- このキーワードは、ジョブを JSP として実行している場合のみ使用してください。
- 関連するキーワードは、[WEBSERVER_DEBUG](#)、[WEBSERVER_DOCROOT](#)、[WEBSERVER_PORT](#) です。

A.3.124 WEBSERVER_DOCROOT

表 A-135 では、WEBSERVER_DOCROOT キーワードを使用できる実行可能ファイルを示します。

表 A-135 WEBSERVER_DOCROOT キーワードを使用する実行可能ファイル

rwclient	rwrund	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
×	×	○	×	×	×	×

説明 WEBSERVER_DOCROOT を使用して、Reports Builder ドキュメント・ルート・ディレクトリを設定できます。イメージ、HTML など JSP で参照するすべてのファイルは、このディレクトリに対する相対パスで指定される必要があります。ドキュメント・ルートを作業ディレクトリに設定することで、これらのファイルをコピーする必要がなくなります。

構文 WEBSERVER_DOCROOT=*REPORTS_TMP/docroot*

次に例を示します。

WEBSERVER_DOCROOT=c:/temp/docroot

値 Oracle Reports 一時フォルダ内のドキュメント・ルート・フォルダへのディレクトリ

デフォルト なし

使用上の注意

- このキーワードは、ジョブを JSP として実行している場合のみ使用してください。
- 関連するキーワードは、[WEBSERVER_DEBUG](#)、[WEBSERVER_DOCROOT](#)、[WEBSERVER_PORT](#) です。

A.3.125 WEBSERVER_PORT

表 A-136 では、WEBSERVER_PORT キーワードを使用できる実行可能ファイルを示します。

表 A-136 WEBSERVER_PORT キーワードを使用する実行可能ファイル

rwclient	rwrund	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
×	×	○	×	×	×	×

説明 WEBSERVER_PORT を使用して、内部 Web サーバーがリスニングするポート番号を指定できます。単一のポート番号 (3002 など) またはポート番号の範囲 (3100-3200 など) を指定できます。単一のポート番号を指定した場合、Oracle Reports は内部 Web サーバーのリスニングをそのポートから開始します。そのポートが使用中の場合、次に利用可能なポートに移ります。ポート番号の範囲を指定した場合、Oracle Reports はその範囲内で空きポートを検索します。

構文 WEBSERVER_PORT=*port_num*

値

port_num には、有効なポート番号またはポート番号の範囲を指定します。

デフォルト

ポート番号: 3000

ポート番号の範囲: 3000-3010

使用上の注意

- このキーワードは、ジョブを JSP として実行している場合のみ使用してください。
- 関連するキーワードは、[WEBSERVER_DEBUG](#)、[WEBSERVER_DOCROOT](#)、[WEBSERVER_PORT](#) です。

環境変数

環境変数とは、Oracle Reports 環境を構成するパラメータです。関連する環境変数には、Oracle Application Server のインストーラによって自動的にデフォルト値が定義されます。環境変数の設定を編集してデフォルトの動作を変更します。

- Windows の場合は、レジストリ・エディタで環境変数を編集します（「スタート」 → 「ファイル名を指定して実行」 → 「Regedit」）。
- UNIX の場合は、初期のデフォルト値を定義するシェル・スクリプト（reports.sh）を修正して実行することで、環境変数を編集します。この場合、変更前の reports.sh ファイルのバックアップを作成してください。

注意： Oracle Reports 10g リリース 2 (10.1.2) の reports.sh ファイルには、そのリリースに必要な変更がいくつか含まれている可能性があります。したがって、以前のリリースで reports.sh ファイルに変更を行った場合、アップグレードを実行する前にバックアップを保存してください。アップグレードした後、その変更を Oracle Reports 10g リリース 2 (10.1.2) でインストールされた reports.sh ファイルとマージします。

この付録に記載されている情報は、Oracle Reports のオンライン・ヘルプにも記載されています。このオンライン・ヘルプは、「はじめに」の「[関連ドキュメント](#)」の説明のように、Reports Builder または Oracle Technology Network (OTN) から表示できます。

B.1 環境変数

表 B-1 では、すべての Oracle Reports 環境変数をアルファベット順に要約したリストと、その有効値およびデフォルト値を示しています。

- すべてのグローバリゼーション・サポート環境変数の詳細は、Oracle Technology Network (<http://www.oracle.com/technology/index.html>) の『Oracle Application Server グローバリゼーション・サポート・ガイド』を参照してください。また、第 18 章「グローバリゼーションと双方向のサポートの実装」も参照してください。
- イタリックで記載された環境変数は、Common Gateway Interface (CGI) レポートの下位互換性のため、または Single Sign-On が使用されていない場合のために、Oracle Reports でサポートされています。

注意： CGI のサポートにかわり、JavaServer Pages (JSP) またはサーブレットの機能が実装されています。

表 B-1 では、すべての Oracle Reports 環境変数をアルファベット順に要約したリストと、その有効値およびデフォルト値を示しています。

表 B-1 Oracle Reports 環境変数

キーワード	有効値	デフォルト
CA_GPREFS	任意のドライブのディレクトリ	ORACLE_HOME
CA_UPREFS	YES NO	YES
DELIMITED_LINE_END	YES NO	YES
DOC	任意のドライブのディレクトリ	ORACLE_HOME¥tools¥doc
DEVELOPER_NLS_LANG	第 18 章「グローバリゼーションと双方向のサポートの実装」を参照。	第 18 章「グローバリゼーションと双方向のサポートの実装」を参照。
NLS_CALENDAR		
NLS_CREDIT		
NLS_CURRENCY		
NLS_DATE_FORMAT		
NLS_DATE_LANGUAGE		
NLS_DEBIT		
NLS_ISO_CURRENCY		
NLS_LANG		AMERICAN_AMERICA.WE8ISO8859P1
NLS_LIST_SEPARATOR		
NLS_MONETARY_CHARACTERS		
NLS_NUMERIC_CHARACTERS		
NLS_SORT		
ORACLE_AFM	任意のドライブのディレクトリ	定義されていません
ORACLE_HOME	任意のドライブのディレクトリ	C:¥orawin
ORACLE_HPDP	任意のドライブのディレクトリ	定義されていません
ORACLE_PATH	任意のドライブのディレクトリ	定義されていません
ORACLE_PPD	任意のドライブのディレクトリ	定義されていません
ORACLE_TFM	任意のドライブのディレクトリ	定義されていません
ORAINFONAV_DOCPATH	任意のドライブのディレクトリ	定義されていません
PRINTER	デフォルトのプリンタ名	定義されていません

表 B-1 Oracle Reports 環境変数 (続き)

キーワード	有効値	デフォルト
REMOTE	任意の有効な SQL*Net ドライバ接頭辞とパラメータ	定義されていません
REPORTS_ADD_HWMARGIN	YES NO	NO
REPORTS_ARABIC_NUMERAL	ARABIC HINDI¥CONTEXT	ARABIC (インド-アラビア数字)
REPORTS_BIDI_ALGORITHM	ORACLE UNICODE	ORACLE
REPORTS_CGIDIAGBODYTAGS	<BODY> タグへの有効な HTML 属性	定義されていません
REPORTS_CGIDIAGHEADTAGS	<HEAD> と </HEAD> の間で有効な HTML タグ	定義されていません
REPORTS_CGIHELP	Web ページまたは HTML ファイルへの有効な URL	デフォルトのヘルプ画面がブラウザに表示されます
REPORTS_CGIMAP	マップ・ファイルへの有効なパス	ORACLE_HOME¥reports¥conf¥cgicmd.dat
REPORTS_CGINODIAG	YES NO	NO
REPORTS_CLASSPATH	デフォルト値は必須です。いずれかのエントリが削除されると、Oracle Reports 実行可能ファイルが正しく動作しなくなる場合があります。ユーザー定義ディレクトリまたは Java クラスが含まれた JAR ファイルをパスに追加することができます。	%ORACLE_HOME¥reports¥jlib¥rwbuilder.jar ;%ORACLE_HOME¥reports¥jlib¥rwrn.jar;%ORACLE_HOME¥jlib¥zrcclient.jar;%ORACLE_HOME¥j2ee¥home¥oc4j.jar;%ORACLE_HOME¥j2ee¥home¥lib¥ojjsp.jar
REPORTS_CONTAINSHTMLTAGS	YES NO	YES
REPORTS_COOKIE_EXPIRE	任意の分数	30
REPORTS_DB_AUTH	特別の認証アクションを含む HTML ファイル (デフォルト値の使用が推奨されます)	dbauth.htm
REPORTS_DEFAULT_DISPLAY	YES NO	YES
REPORTS_DEFAULT_PIXEL_SIZE	72 ~ 200 までの任意の値	Oracle Reports によって決定される画面解像度
REPORTS_ENABLE_RTF_SPACING	YES NO	NO
REPORTS_ENCRYPTION_KEY	任意の暗号化鍵	reports9.0
REPORTS_ENHANCED_SUBSET	YES NO	YES
REPORTS_GRAPH_IMAGE_DPI	72 ~ 300	72
REPORTS_IGNORE_IMAGE_TAG_RES	YES NO	NO
REPORTS_JPEG_QUALITY_FACTOR	1 ~ 100	100
REPORTS_JVM_OPTIONS	JVM コマンドライン構文の JVM オプションのリスト	-Xmx256M
REPORTS_NETWORK_CONFIG	ORACLE_HOME/reports/conf/ にある有効なカスタム・ネットワーク構成ファイル名	rwnetwork.conf
REPORTS-NLS_XML_CHARSETS	セミコロンで区切ったマッピング・ペアのセット。最初の値には生成されるエンコードを指定し、2 番目の値にはマッピング値を指定します。 <old_name>=<new_name>[;<old_name>=<new_name>] [<old_name>=<new_name>] ...	定義されていません
REPORTS_NO_DUMMY_PRINTER	TRUE 設定しません	TRUE

表 B-1 Oracle Reports 環境変数 (続き)

キーワード	有効値	デフォルト
REPORTS_NO_HTML_SPACE_REPLACE	YES 設定しません	設定しません
REPORTS_OUTPUTIMAGEFORMAT	GIF JPEG JPG PNG BMP	JPEG
REPORTS_PATH	任意のドライブのディレクトリ	%ORACLE_HOME%\REPORT\TEMPLATES; %ORACLE_ HOME%\REPORT\samples\demo; %ORACLE_ HOME%\REPORT\integ;%ORACLE_ HOME%\REPORT\Printers
REPORTS_RESOURCE	任意のドライブのディレクトリ	%ORACLE_HOME%\reports\res\US
REPORTS_SERVER	任意の Reports Server サービス・エントリ名	
REPORTS_SOLARIS_9	YES NO	Solaris 2.9 の場合は YES、他のプラットフォームの場合は NO
REPORTS_SPACE_BREAK	YES NO	YES
REPORTS_SRWRUN_TO_SERVER	YES 設定しません	設定しません
REPORTS_SSLPORT	有効なポート番号	443
REPORTS_SYS_AUTH	特別の認証アクションを含む HTML ファイル (デフォルト値の使用が推奨されます)	sysauth.htm
REPORTS_TAGLIB_URI	Oracle Reports タグ・ライブラリを参照する任意の "uri"	/WEB-INF/lib/reports_tld.jar
REPORTS_TMP	任意のドライブのディレクトリ	定義されていません
REPORTS_USEREXITS	任意のユーザー・イグジット Dynamic Link Library (絶対パス)	定義されていません
REPORTS_UTF8_XMLOUTPUT	YES NO	YES
RW	有効なディレクトリ名	%ORACLE_HOME%\reports (Windows) \$ORACLE_HOME/reports (UNIX)
TK_PRINT	PRINT コマンドと使用している UNIX に応じたすべての必要なキーワード。これには次の要素が含まれます。 <ul style="list-style-type: none"> ■ %n は、プリンタ名の文字列です。 ■ %c は、印刷部数です。 この文字列は、printf() 書式とほぼ同じです。この環境変数が設定されていない場合、Oracle Reports 6i は、プラットフォームの標準のデフォルト値を使用します。プラットフォームごとのデフォルト値の例を次に示します。 <p>System V: lp -s -d'%n' -n%c</p> <p>Solaris: lpr -P'%n' -#%c -s</p>	定義されていません
TK_PRINT_STATUS	プリンタ名の %n を含める必要があります (「TK_PRINT」も参照)。 <p>この環境変数が設定されていない場合、Oracle Reports では、組込みのデフォルト値が使用されます。</p> <p>System V: /usr/bin/lpstat -p'%n' 2>&1</p> <p>その他: /usr/etc/lpc status '%n' 2>&1</p>	定義されていません

表 B-1 Oracle Reports 環境変数 (続き)

キーワード	有効値	デフォルト
TK_PRINTER	デフォルトのプリンタ名	定義されていません
TK_AFM	任意のドライブのディレクトリ	定義されていません
TK_HPD	任意のドライブのディレクトリ	定義されていません
TK_PPD	任意のドライブのディレクトリ	定義されていません
TK_TFM	任意のドライブのディレクトリ	定義されていません
USERNAME	有効な Oracle ユーザー名 (OPS\$ 接頭辞なし)	定義されていません
USER_NLS_LANG		

B.1.1 CA_GPREFS

説明 この環境変数は、グローバル設定ファイル、CAGPREFS.ORA の位置を指定します。グローバル設定はネットワーク・ユーザー間で共有されます。Oracle Reports では、CA_GPREFS で指定されたディレクトリのみでなく、CAGPREFS.ORA ファイルの現行ディレクトリも検索されます。

CAGPREFS.ORA ファイルは、Oracle Installer によって自動的に作成されます。グローバル設定ファイルの内容を変更するには、メモ帳などのテキスト・エディタを使用して、ファイルを手動で編集します。この CAGPREFS.ORA ファイルのグローバル設定よりも、CA_UPREFS で定義されるローカル設定ファイル、CAUPREFS.ORA が優先します。

有効値 任意のドライブのディレクトリ

デフォルト ORACLE_HOME

例 CA_GPREFS=C:\orawin

B.1.2 CA_UPREFS

説明 この環境変数は、ユーザー設定ファイル、CAUPREFS.ORA の位置を指定します。CAUPREFS.ORA ファイルでは、製品内で「ツール」→「オプション」を選択して設定したプリファレンスが保持されます。Oracle Reports では、CA_UPREFS で指定されたディレクトリのみでなく、CAUPREFS.ORA ファイルの現行ディレクトリも検索されます。

Oracle 製品の中には、その設定情報が CAUPREFS.ORA ファイルに書き込まれるものもあります。ユーザー設定の内容を手動で変更するには、メモ帳などのテキスト・エディタを使用して、そのファイルを編集します。この CAUPREFS.ORA ファイルのユーザー設定は、CA_GPREFS で定義される CAGPREFS.ORA ファイルのグローバル設定よりも優先します。

有効値 任意のドライブのディレクトリ

デフォルト ORACLE_HOME

例 CA_UPREFS=C:\orawin

B.1.3 DELIMITED_LINE_END

説明 この環境変数は、デリミタ付き出力の行末にデリミタ付き文字を出力するかどうかを指定します。

有効値 YES | NO

デフォルト YES

使用上の注意

- この環境変数を NO に設定すると、デリミタ付き文字が行末に出力されません。

B.1.4 DOC

説明 この環境変数は、オンライン・ヘルプを含むオンライン・ドキュメント・ファイルの位置を指定します。

有効値 任意のドライブのディレクトリ

デフォルト `ORACLE_HOME¥tools¥doc`

例 `DOC=C:¥myreports_1012¥tools¥doc`

B.1.5 DEVELOPER_NLS_LANG

説明 この環境変数は、レポートの言語を指定します。この環境変数とその有効な値の詳細は、[第 18 章「グローバリゼーションと双方向のサポートの実装」](#)を参照してください。

B.1.6 NLS_CALENDAR

説明 この環境変数は、使用するカレンダー・システムを指定します。

B.1.7 NLS_CREDIT

説明 この環境変数は、正の通貨値を示す文字列を指定します。

B.1.8 NLS_CURRENCY

説明 この環境変数は、各国の通貨記号を指定します。

B.1.9 NLS_DATE_FORMAT

説明 この環境変数は、日付に使用するデフォルトの書式を指定します。

B.1.10 NLS_DATE_LANGUAGE

説明 この環境変数は、日付に使用するデフォルトの言語を指定します。

B.1.11 NLS_DEBIT

説明 この環境変数は、負の通貨値を示す文字列を指定します。

B.1.12 NLS_ISO_CURRENCY

説明 この環境変数は、ISO 通貨記号を指定します。

B.1.13 NLS_LANG

説明 この環境変数は、次のような言語と設定を指定します。

- 「しばらくお待ちください」など、ユーザーに表示されるメッセージの言語
- 日付や数値に使用されるデフォルトの書式マスク
- ソート基準
- キャラクタ・セットを構成する文字

この環境変数とその有効値の詳細は、[第 18 章「グローバル化と双方向のサポートの実装」](#)を参照してください。

構文 `NLS_LANG= language_territory_charset`

有効値

- `language` は、メッセージおよび月と曜日の名前を表示する言語とその表記規則を指定します。
- `territory` は、週番号と曜日番号の計算に使用される地域とその表記規則を指定します。
- `charset` は、UPPER、LOWER および INITCAP 関数に使用されるキャラクタ・セットと、ORDER BY 問合せによって使用されるソートのタイプを指定します。この引数は、メッセージの表示に使用するキャラクタ・セットも制御します。

Oracle Technology Network (<http://www.oracle.com/technology/index.html>) の『Oracle Application Server グローバリゼーション・サポート・ガイド』を参照してください。

デフォルト `AMERICAN_AMERICA.WE8ISO8859P1`

使用上の注意

- ロケールを変更するには、[REPORTS_RESOURCE](#) 環境変数以外に、この環境変数も変更する必要があります。

例 アプリケーションをフランス語で実行するとします。まず、アプリケーションをフランスで使用し、データを WE8ISO8895P1 キャラクタ・セットで表示するとします。この場合、NLS_LANG を次のように設定します。

```
NLS_LANG=French_France.WE8ISO8895P1
```

今度は、アプリケーションを同じフランス語で実行しますが、スイスで使用するとします。この場合、NLS_LANG を次のように設定します。

```
NLS_LANG=French_Switzerland.WE8ISO8895P1
```

その他の例

```
NLS_LANG=Norwegian_Norway.NDK7DEC
```

```
NLS_LANG=Norwegian_Norway.WE8ISO8895P1
```

```
NLS_LANG=Japanese_Japan.JA16SJIS
```

```
NLS_LANG=Arabic_Egypt.AR8MSWIN1256
```

```
NLS_LANG=American_America.AR8MSWIN1256
```

```
NLS_LANG=American_America.WE8ISO8859P1
```

B.1.14 NLS_LIST_SEPARATOR

説明 この環境変数は、リスト内の項目を区切るための文字を指定します。

B.1.15 NLS_MONETARY_CHARACTERS

説明 この環境変数は、通貨値用の小数点文字と 3 桁区切りを指定します。

B.1.16 NLS_NUMERIC_CHARACTERS

説明 この環境変数は、数値用の小数点文字と桁グループ・セパレータを指定します。

B.1.17 NLS_SORT

説明 この環境変数は、文字データに使用されるソートのタイプを指定します。

B.1.18 ORACLE_AFM

説明 この環境変数は、AFM ファイルの位置を指定します。[TK_AFM](#) の後に、[ORACLE_AFM](#) が参照されます。

有効値 任意のドライブのディレクトリ

デフォルト 定義されていません

使用上の注意

- いずれの変数にも値を指定しない場合、Oracle Reports では、次のディレクトリの AFM ファイルが検索されます。

`ORACLE_HOME/guicommon/tk/admin/AFM`

- UNIX 上で印刷する場合、適切な印刷環境の作成には、いくつかの設定および構成が必要になります。詳細は、[第 5 章「UNIX での Oracle Reports による印刷」](#)を参照してください。

B.1.19 ORACLE_HOME

説明 この環境変数は、Windows Oracle 製品がインストールされるホーム・ディレクトリを指定します。このディレクトリは、Oracle ディレクトリ階層の中の最上位ディレクトリです。

有効値 任意のドライブのディレクトリ

デフォルト `C:\orawin`

使用上の注意

- Reports Runtime (`rwrwin`) を使用する場合は、[ORACLE_HOME](#) と [ORACLE_PATH](#) を合わせた長さが 255 文字を超えないようにしてください。

例 `ORACLE_HOME=C:\orawin`

B.1.20 ORACLE_HPDP

説明 この環境変数は、HPD ファイルの位置を指定します。[TK_HPDP](#) の後に、ORACLE_HPDP が参照されます。

有効値 任意のドライブのディレクトリ

デフォルト 定義されていません

使用上の注意

- いずれの変数にも値を指定しない場合、Oracle Reports では、次のディレクトリの HPD ファイルが検索されます。

`ORACLE_HOME/guicommon/tk/admin/HPD`

- UNIX 上で印刷する場合、適切な印刷環境の作成には、いくつかの設定および構成が必要になります。詳細は、[第 5 章「UNIX での Oracle Reports による印刷」](#)を参照してください。

B.1.21 ORACLE_PATH

説明 この環境変数は、Reports Runtime で参照されるファイルの検索パスを指定します。ただし、ORACLE_PATH で指定されたディレクトリは、[REPORTS_PATH](#) で指定されたディレクトリの後に検索されます。

ORACLE_PATH では、複数のディレクトリを指定できます。パス・リスト内でディレクトリ名を区切るには、セミコロン (;) を使用します。

有効値 任意のドライブのディレクトリ

デフォルト 定義されていません

使用上の注意

- Reports Runtime (rwrun) を使用する場合は、[ORACLE_HOME](#) と ORACLE_PATH を合わせた長さが 255 文字を超えないようにしてください。

例 `ORACLE_PATH=C:\oracle\apps\forms;C:\oracle\apps\reports`

B.1.22 ORACLE_PPD

説明 この環境変数は、PPD ファイルの位置を指定します。[TK_PPD](#) の後に、ORACLE_PPD が参照されます。

有効値 任意のドライブのディレクトリ

デフォルト 定義されていません

使用上の注意

- いずれの変数にも値を指定しない場合、Oracle Reports では、次のディレクトリの PPD ファイルが検索されます。

`ORACLE_HOME/guicommon/tk/admin/PPD`

- UNIX 上で印刷する場合、適切な印刷環境の作成には、いくつかの設定および構成が必要になります。詳細は、[第 5 章「UNIX での Oracle Reports による印刷」](#)を参照してください。

B.1.23 ORACLE_TFM

説明 この環境変数は、TFM ファイルの位置を指定します。[TK_TFM](#) の後に、ORACLE_TFM が参照されます。

有効値 任意のドライブのディレクトリ

デフォルト 定義されていません

使用上の注意

- いずれの変数にも値を指定しない場合、Oracle Reports では、次のディレクトリの TFM ファイルが検索されます。

`ORACLE_HOME/guicommon/tk/admin/TFM`

- UNIX 上で印刷する場合、適切な印刷環境の作成には、いくつかの設定および構成が必要になります。UNIX 上で Oracle Reports を使用した印刷の詳細は、[第 5 章「UNIX での Oracle Reports による印刷」](#)を参照してください。

B.1.24 ORAINFONAV_DOCPATH

説明 この環境変数では、オンライン・ドキュメントの目次と索引の位置を指定します。

有効値 任意のドライブのディレクトリ

デフォルト 定義されていません

例 `ORAINFONAV_DOCPATH=C:\orawin\oin`

B.1.25 PRINTER

説明 この環境変数は、デフォルトのプリンタ名を指定します。

有効値 デフォルトのプリンタ名

デフォルト 定義されていません

使用上の注意

- [TK_PRINTER](#) は、PRINTER よりも優先します。つまり、両方の変数が設定されている場合は、TK_PRINTER が最初に参照されます。PRINTER が参照されるのは、TK_PRINTER に有効なプリンタが指定されていない場合のみです。TK_PRINTER と PRINTER の両方に、有効なプリンタが設定されていない場合は、Oracle Reports では `uiprint.txt` ファイルの最初のエントリが使用されます。REPORTS_NO_DUMMY_PRINTER が設定されており、`uiprint.txt` ファイルに有効なエントリが含まれていない場合は、`uiscreenprint.txt` に指定されている `screenprinter.ppd` が使用されます。

注意： REPORTS_NO_DUMMY_PRINTER はデフォルトで設定され、常に設定されている必要があります。これが設定されていない場合は（ユーザーが変更した結果）、REP-1800 エラー・メッセージが表示されます。

関連項目： PostScript プリンタ・ドライバ `screenprinter.ppd` の詳細は、[第 3.10.1 項「ScreenPrinter」](#)を参照してください。

- UNIX 上で印刷する場合、適切な印刷環境の作成には、いくつかの設定および構成が必要になります。詳細は、第 5 章「UNIX での Oracle Reports による印刷」を参照してください。

B.1.26 REMOTE

説明 この環境変数は、ローカル・データベースに接続する際に使用するデフォルトのリモート SQL*Net ドライバを指定します。パラメータには、デフォルトの SQL*Net パラメータ（完全なデータベース文字列）を設定できます。

ユーザーがログインし、REMOTE の指定と一致するドライバ接頭辞を明示的に使用して接続を指定し、SQL*Net パラメータを指定しなかった場合は、REMOTE で指定されているパラメータが使用されます。このパラメータにより DBA は、SQL*Net ユーザーが接続パラメータを指定する必要のない標準のネットワーク接続を定義できます。このパラメータは、コマンドラインからいつでもリセットできます。

注意： Windows 用 DOS SQL*Net ドライバを使用する場合、DOS Oracle ホーム・ディレクトリにある CONFIG.ORA ファイルに REMOTE パラメータが設定されているかどうかを確認してください。CONFIG.ORA に REMOTE が設定されている場合、同じ値をレジストリに設定する必要があります。

構文 REMOTE= *netPrefix:databaseName*

有効値

- *netPrefix* には、任意の有効な SQL*Net ドライバ接頭辞を指定します。
- *databaseName* には、ローカル・データベースの名前を指定します。

デフォルト 定義されていません

例 REMOTE=P:PIPER

ここでは

P は、Named Pipes のネットワーク接頭辞です。

PIPER は、データベース名です。

B.1.27 REPORTS_ADD_HWMARGIN

説明 (Windows のみ) この環境変数は、プリンタのハードウェアベースの左マージンを設定するかどうかを指定します。デフォルトでは、このマージンは無視されます。印刷は印刷可能領域ではなく、実際の用紙の左上隅 (0, 0) を起点として開始されます。そのため、プリンタ・ハードウェアのマージンに依存しないレポートを容易に設計できます。これらのレポートは、様々なプリンタに配布できます。

それまではプリンタの印刷可能領域が使用されていたため、プリンタの機種によって、レポート出力の位置が異なっていました。

必要に応じて、以前の動作に戻すことができます。その場合は、レジストリ変数 REPORTS_ADD_HWMARGIN を YES に設定します。

REPORTS_ADD_HWMARGIN レジストリ変数を設定するには、次の手順を実行します。

1. レジストリ・エディタ（たとえば、regedit.exe）を使用して Windows のレジストリを編集します。

注意： レジストリを編集する前に、レジストリのバックアップを作成してください。

- 次に示すキーにナビゲートします。

HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\HOME n

n は、このインストールが格納されている ORACLE_HOME の数です。

- REPORTS_ADD_HWMARGIN という新しい文字列値を追加し、この値を YES に設定します。

有効値 YES|NO

デフォルト NO

使用上の注意

- Windows でハードウェアベースの左マージンを入れずにレポートを印刷する際には、レポートのレイアウト時に、データが印刷可能領域に表示されるように十分なマージン・スペースを確保する必要があります。Reports Builder の「ページ設定」ダイアログ・ボックスの「マージン」フィールドは、OracleAS Reports Services との一貫性を保つために無効になっています。

B.1.28 REPORTS_ARABIC_NUMERAL

説明 この環境変数は、アラビア語の PDF 出力での数値の書式を指定します。この環境で有効な値は、ARABIC (アラビア数字)、HINDI (ヒンディー数字) または CONTEXT (コンテキスト) に応じてアラビア数字またはヒンディー数字) です。この環境変数では、大文字と小文字は区別されません。

有効値 ARABIC|HINDI|CONTEXT

デフォルト ARABIC (インド - アラビア数字)

B.1.29 REPORTS_BIDI_ALGORITHM

説明 この環境変数は、双方向 (BiDi) 言語 (アラビア語やヘブライ語など) の双方向レイアウト・アルゴリズムを切り替えます。この環境変数では、大文字と小文字は区別されません。

有効値

- ORACLE の場合、Oracle Reports は Oracle BiDi アルゴリズムに従います。
- UNICODE の場合、Oracle Reports は Unicode BiDi アルゴリズムに従います。

Unicode BiDi アルゴリズムの詳細は、<http://www.unicode.org/reports/tr9/> を参照してください。

デフォルト ORACLE

B.1.30 REPORTS_CGIDIAGBODYTAGS

説明 この環境変数は、rwcgi 診断およびデバッグ出力の <BODY> タグに追加する HTML 属性を指定します。たとえば、この環境変数を使用して、テキストと背景色または背景イメージを設定できます。

この環境変数には下位互換性があります。

有効値 <BODY> タグへの有効な HTML 属性

デフォルト 定義されていません

使用上の注意

この環境変数は、Common Gateway Interface (CGI) レポートの下位互換性のため、または Single Sign-On が使用されていない場合のために、Oracle Reports でサポートされています。

注意: CGI のサポートにかわり、JavaServer Pages (JSP) またはサーブレットの機能が実装されています。

例 REPORTS_CGIDIAGBODYTAGS="bgcolor="#CC3366""

B.1.31 REPORTS_CGIDIAGHEADTAGS

説明 この環境変数は、rwcgi 診断およびデバッグ出力の <HEAD> タグと </HEAD> タグの間に挿入する HTML タグを指定します。たとえば、この環境変数を使用して <TITLE> または <META> タグを設定できます。

有効値 <HEAD> タグと </HEAD> タグの間で有効な HTML タグ

デフォルト 定義されていません

使用上の注意

- この環境変数は、Common Gateway Interface (CGI) レポートの下位互換性のため、または Single Sign-On が使用されていない場合のために、Oracle Reports でサポートされています。

注意: CGI のサポートにかわり、JavaServer Pages (JSP) またはサーブレットの機能が実装されています。

例 REPORTS_CGIDIAGHEADTAGS="<title>Employee List</title>"

B.1.32 REPORTS_CGIHELP

説明 この環境変数は、rwcgi が次に示す空のリクエストで起動された場合に表示される rwcgi ヘルプ・ファイルの URL および URI を指定します。

この環境変数には下位互換性があります。

http://your_webserver/rwcgi?

有効値 Web ページまたは HTML ファイルへの有効な URL

デフォルト デフォルトのヘルプ画面がブラウザに表示されます

使用上の注意

- この環境変数は、Common Gateway Interface (CGI) レポートの下位互換性のため、または Single Sign-On が使用されていない場合のために、Oracle Reports でサポートされています。

注意: CGI のサポートにかわり、JavaServer Pages (JSP) またはサーブレットの機能が実装されています。

例

ブラウザに `www.yahoo.com` ページを表示するには、次のように指定します。

```
REPORTS_CGIHELP=http://www.yahoo.com
```

ブラウザに `myhelpfile.htm` という HTML ファイルを表示するには、次のように指定します。

```
REPORTS_CGIHELP=http://your_webserver/myhelpfile.htm
```

B.1.33 REPORTS_CGIMAP

説明 この環境変数は、マップ・ファイル構成を使用する場合に、`rwcgi` マップ・ファイルの完全修飾ファイル名および位置を指定します。

この環境変数には下位互換性があります。

有効値 マップ・ファイルへの有効なパス

デフォルト `ORACLE_HOME¥reports¥conf¥cgicmd.dat`

使用上の注意

- この環境変数は、Common Gateway Interface (CGI) レポートの下位互換性のため、または Single Sign-On が使用されていない場合のために、Oracle Reports でサポートされています。

注意： CGI のサポートにかわり、JavaServer Pages (JSP) またはサーブレットの機能が実装されています。

例 `REPORTS_CGIMAP=c:¥orawin¥reports¥conf¥cgicmd.dat`

B.1.34 REPORTS_CGINODIAG

説明 この環境変数は、`help` や `showmap` などの、`rwcgi` からのデバッグおよび診断出力をすべて無効にするかどうかを指定します。

この環境変数には下位互換性があります。

有効値 YES | NO

デフォルト NO

使用上の注意

- この環境変数は、Common Gateway Interface (CGI) レポートの下位互換性のため、または Single Sign-On が使用されていない場合のために、Oracle Reports でサポートされています。

注意： CGI のサポートにかわり、JavaServer Pages (JSP) またはサーブレットの機能が実装されています。

例 `REPORTS_CGINODIAG=YES` に設定されていると、次のリクエストが機能しくなくなります。

```
http://your_webserver/rwcgi/help?
```

B.1.35 REPORTS_CLASSPATH

説明 この環境変数は、Oracle Reports 実行可能ファイルで起動する場合に、Java Virtual Machine (JVM) が使用する JAR ファイルとディレクトリのリストを指定します。このリストを追加するのは通常、レポートの設計時に独自のクラスを追加する必要がある場合（たとえば、プラグgable・データソース (PDS) を追加したり、PL/SQL-Java ブリッジを使用する場合など）です。

注意： REPORTS_CLASSPATH 環境変数（レジストリ）の値が 511 文字を超えると、Reports Builder は正しく動作しません。この問題を解決するには、CLASSPATH 環境変数（システム）を使用して 511 文字を超える値を指定します。

有効値 デフォルト値は必須です。いずれかのエントリが削除されると、Oracle Reports 実行可能ファイルが正しく動作しなくなる場合があります。ユーザー定義ディレクトリまたは Java クラスが含まれた JAR ファイルをパスに追加することができます。

デフォルト %ORACLE_HOME%\reports\jlib\rwbuilder.jar;%ORACLE_HOME%\reports\jlib\rwrun.jar;%ORACLE_HOME%\jlib\zrclient.jar;%ORACLE_HOME%\j2ee\home\oc4j.jar;%ORACLE_HOME%\j2ee\home\lib\ojsp.jar

使用上の注意

- この環境変数のデフォルト値は、Oracle Reports 実行可能ファイルが正しく動作するうえで必要です。ユーザー・クラスを追加することはできますが、そのリストはプラットフォーム固有の Java の CLASSPATH 定義に従っている必要があります。

例

```
REPORTS_CLASSPATH=%ORACLE_HOME%\reports\jlib\rwbuilder.jar;
%ORACLE_HOME%\reports\jlib\rwrun.jar;%ORACLE_
HOME%\jlib\zrclient.jar;%ORACLE_HOME%\j2ee\home\oc4j.jar;%ORACLE_
HOME%\j2ee\home\lib\ojsp.jar
```

- REPORTS_CLASSPATH 環境変数（レジストリ）の値が 511 文字を超えると、Reports Builder は正しく動作しません。この問題を解決するには、CLASSPATH 環境変数（システム）を使用して 511 文字を超える値を指定します。

B.1.36 REPORTS_CONTAINSHTMLTAGS

説明 この環境変数は、Oracle Reports で、サポートされているすべての出力フォーマットで、HTML フォーマット・タグが解釈されるかどうかを指定します。

注意： インライン HTML タグの解釈は、Oracle Reports とブラウザで異なる場合があります。したがって、Oracle Reports 6i、Oracle9i Reports または Oracle Reports 10g (9.0.4) でインライン HTML タグを使用して設計されたレポートは、Oracle Reports 10g リリース 2 (10.1.2) では異なる HTML または HTMLCSS 出力を生成することがあります。

有効値

- YES の場合、Oracle Reports では、「HTML タグを含む」プロパティが「はい」に設定されているすべてのオブジェクトで、HTML タグが解釈されます。
- NO の場合、Oracle Reports では、オブジェクトの「HTML タグを含む」プロパティの設定に関係なく、そのレポートに対して HTML タグは解釈されません。出力が HTML および HTMLCSS の場合は、ブラウザによって HTML フォーマット・タグが解釈されます。それ以外の出力フォーマットの場合は、HTML タグ自体がそのままの形でレポート出力に表示されます。Oracle Reports で HTML フォーマット・タグが解釈されないようにして、以前のリリースの動作を保持する場合は、この環境変数を NO に設定します。

関連項目： インライン HTML フォーマット・タグの実装の詳細は、[第 A.3.16 項「CONTAINSHTMLTAGS」](#)を参照してください。

デフォルト YES

使用上の注意

コマンドライン・キーワード CONTAINSHTMLTAGS は、この環境変数の値よりも優先されます。

B.1.37 REPORTS_COOKIE_EXPIRE

説明 この環境変数は、指定された Reports Server セッション内の Cookie の存続期間を指定します。

Single Sign-On を使用せずに、Reports Server のセキュリティ保護されたインスタンスへユーザーがアクセスする場合、Oracle Reports 6i の場合と同様に、独自の認証メカニズムを使用して `rwServlet` により識別情報を提供する必要があります。HTTP 1.0 プロトコルはステートレスであるため（つまり、サーバーへの各コールは、事実上独立している）、Cookie が保持されていないかぎり、レポート・リクエストのたびにユーザーの認証が必要になる場合があります。

セッションごとに1度の認証で済むように、`rwServlet` には独自のクライアント側の Cookie である `authidCookie` が保存されます。ここには、現行セッションに必要な認証情報が格納されます。一度ユーザーが認証されると、暗号化された Cookie がブラウザで作成され、ユーザーはリクエストごとに認証を受けることなく、複数のレポート・ジョブを実行できます。ユーザーがブラウザ・セッションを終了すると `authidCookie` も終了しますが、この方法だけで Cookie を終了するのは十分とはいえません。REPORTS_COOKIE_EXPIRE 環境変数を使用して、セッション内の Cookie の存続期間を制限する必要があります。たとえば、ユーザーがログインし、昼食のためにブラウザ・セッションを起動したまま席を外したとします。管理者が Reports Server で REPORTS_COOKIE_EXPIRE 環境変数を定義しておけば、そのような潜在的セキュリティ侵害を最小限に抑えることができます。`rwServlet` でジョブ・リクエストが受信されると、Cookie に保存された時間と現在のシステム時間が比較されます。この時間が環境変数で定義された時間（たとえば、30 分）を超えていると、Cookie は否定され、ユーザーには認証情報を提供する必要が生じます。

注意： 特定のレポートに対してユーザーに認証を要求する場合は、SHOWAUTH コマンドライン・キーワードを使用できます。または、キー・マップ・ファイル内の対応するレポート・エントリに %S を使用することもできます。このファイルは通常 `cgicmd.dat` という名前前で、`ORACLE_HOME¥reports¥conf` に格納されています。%S を使用すると、ユーザーはレポートをコールするたびに、ユーザー名とパスワードの入力が要求されます。

有効値 任意の分数

デフォルト 30

使用上の注意

この環境変数は、Common Gateway Interface (CGI) レポートの下位互換性のため、または Single Sign-On が使用されていない場合のために、Oracle Reports でサポートされています。

注意： CGI のサポートにかわり、JavaServer Pages (JSP) またはサーブレットの機能が実装されています。

例 REPORTS_COOKIE_EXPIRE=30

B.1.38 REPORTS_DB_AUTH

説明 この環境変数は、データベースにログインするためのデータベース認証テンプレートを指定します。この環境変数には下位互換性があります。

有効値 特別の認証アクションを含む HTML ファイル (デフォルト値の使用が推奨されます)

デフォルト dbauth.htm

使用上の注意

この環境変数は、Common Gateway Interface (CGI) レポートの下位互換性のため、または Single Sign-On が使用されていない場合のために、Oracle Reports でサポートされています。

注意： CGI のサポートにかわり、JavaServer Pages (JSP) またはサーブレットの機能が実装されています。

例 REPORTS_DB_AUTH=dbauth.htm

B.1.39 REPORTS_DEFAULT_DISPLAY

説明 この環境変数は、Oracle Reports 10g (9.0.4) で導入された次の機能を実装するかどうかを指定します。

- DISPLAY 変数への依存性の排除 (UNIX のみ)
- イメージの画面解像度とフォント情報のための **ScreenPrinter** (screenprinter.ppd) の使用。これによって、Reports Runtime に対して有効なプリンタを定義すること (PRINTER および TK_PRINTER 環境変数を有効なプリンタに設定すること。または uiprint.txt の有効なエントリ) への依存性が排除されます (UNIX のみ)。
- 拡張イメージングのサポート (すべてのプラットフォーム)
詳細は、第 3.10.2 項「**拡張イメージングのサポート**」を参照してください。

有効値 YES | NO

デフォルト YES

使用上の注意

- UI を非表示にするには、Reports Server をバッチ・モードで起動する必要があります。
- REPORTS_DEFAULT_DISPLAY=YES に設定することにより、**REPORTS_OUTPUTIMAGEFORMAT** 環境変数と OUTPUTIMAGEFORMAT コマンドライン・キーワードで導入される拡張イメージング・サポートが有効になります。画面解像度は、screenprinter.ppd ファイルのエントリで制御できます。REPORTS_DEFAULT_DISPLAY=NO に設定すると、イメージング・サポートは、GIF 形式 (PDF 出力、HTML、HTMLCSS の場合) および BMP 形式 (RTF 出力の場合) に制限されます。

- UNIX では、`REPORTS_DEFAULT_DISPLAY=YES` は、`DISPLAY` 変数で設定された値よりも優先されます。`DISPLAY` 変数が定義されている場合でも X-Windows 表示画面はデフォルトでは使用されません。画面解像度は、`screenprinter.ppd` のエントリで制御できます。Oracle Reports 10g (9.0.4) より前のリリースからアップグレードする場合、この変更が、既存のレポートの外観、ページ数、出力ファイル・サイズまたはパフォーマンスに影響を与えることがあります。
- この機能は IBM JDK 1.4 (ヘッドレス・オプションはサポートしていません) においてサポートされていないので、IBM AIX プラットフォームでは使用できません。このため、AIX の場合は `DISPLAY` への依存が依然として存在します。
- `DISPLAY` に依存するように戻し、スクリーン・フォント (従来のフォント検索アルゴリズム) を使用する手順は次のとおりです。
 - `REPORTS_DEFAULT_DISPLAY=NO` に設定します。
 - `uiscreenprint.txt` ファイルの `screenprinter.ppd` エントリを削除します。
 - `DISPLAY` 変数をアクティブな X-Windows 表示画面に設定します。
- UNIX 上で印刷する場合、適切な印刷環境の作成には、いくつかの設定および構成が必要になります。詳細は、第 5 章「UNIX での Oracle Reports による印刷」を参照してください。

B.1.40 REPORTS_DEFAULT_PIXEL_SIZE

説明 この環境変数は、レポートを HTML 出力で生成する場合に、表示サーバーのデフォルトのピクセル・サイズより優先されるピクセル・サイズを指定します。通常、Oracle Reports では表示サーバーのピクセル・サイズが使用されます。以前のレポートで作業しており、そのレポートに表示サーバーと異なるピクセル・サイズ (たとえば、ピクセル・サイズ 80) が使用されている場合、この変数を使用すれば以前のレポートと同じ動作を再現できます。

有効値 72 ~ 200 までの任意の値

デフォルト Oracle Reports で決定される画面解像度

使用上の注意

- Windows の場合、`REPORTS_DEFAULT_PIXEL_SIZE` はレジストリで設定されます。UNIX の場合、コマンド・プロンプトまたはシェル・スクリプトから設定されます。
- `REPORTS_DEFAULT_DISPLAY = YES` (デフォルト) の場合、Oracle Reports の HTML 出力には、`REPORTS_DEFAULT_PIXEL_SIZE` で指定された値が使用されます。ただし、値が `REPORTS_DEFAULT_PIXEL_SIZE` で明示的に設定されていない場合は、第 5 章「UNIX での Oracle Reports による印刷」に記載されているとおり、画面解像度は `screenprinter.ppd` ファイルのエントリで制御できます。

B.1.41 REPORTS_ENABLE_RTF_SPACING

この環境変数は、RTF 出力で複数行テキストの切捨てを防止する機能を有効化するかどうかを指定します。この環境変数は、ほとんどの環境で必要ありませんが、RTF 出力で複数行テキストの切捨てが表示された場合のみ、YES を設定してください。

有効値

- YES の場合、RTF 出力で複数行テキストの切捨てを防止する機能が有効になります。
- NO の場合、RTF 出力で複数行テキストの切捨てを防止する機能が無効になります。切捨てが行われる可能性があります。

デフォルト NO

B.1.42 REPORTS_ENCRYPTION_KEY

説明 この環境変数は、ユーザー名とパスワードの暗号化に使用する暗号化鍵を指定します。

有効値 任意の暗号化鍵

デフォルト reports9i

使用上の注意

この環境変数は、Common Gateway Interface (CGI) レポートの下位互換性のため、または Single Sign-On が使用されていない場合のために、Oracle Reports でサポートされています。

注意： CGI のサポートにかわり、JavaServer Pages (JSP) またはサーブレットの機能が実装されています。

例 REPORTS_ENCRYPTION_KEY=oraclereports10g

B.1.43 REPORTS_ENHANCED_SUBSET

説明 この環境変数は、レポートを生成するときに拡張 TTF フォント・サブセット機能を含めるかどうかを指定します。この環境変数はデフォルトでは YES に設定されており、この設定で生成される PDF ファイルはアクセス可能で検索可能になります。

有効値 YES|NO

デフォルト YES

使用上の注意

Oracle Reports では、デフォルトでは拡張フォント・サブセットの実装が使用されます。REPORTS_ENHANCED_SUBSET=NO を設定した場合、Oracle Reports は、Oracle Reports 10g リリース 2 (10.1.2) よりも前のリリースで使用されている Type 3 フォント・サブセットの実装に戻ります。

PDF フォント・サブセットの詳細は、[第 6.1.2.2 項「フォント・サブセット」](#)を参照してください。

B.1.44 REPORTS_GRAPH_IMAGE_DPI

説明 この環境変数は、PDF ファイルまたはプリンタにグラフを出力する際のインチ当たりのドット数 (DPI) を指定します。この環境変数のデフォルト値は、72 DPI に設定されています。これは、レポートのファイル・サイズを小さくするとともに、レポートの生成に要する時間を最小化するためです。

72 DPI より高い値を指定すると、PDF ファイルやプリンタに出力されるグラフのイメージ解像度が向上します。ただし、これにより、ファイル・サイズが大きくなるだけでなく、レポート出力の生成に、より時間がかかります。

注意： REPORTS_GRAPH_IMAGE_DPI=250 の値の場合：

- グラフを含むレポートの生成にかかる時間が、同じレポートを 72 DPI で生成する場合の 5～6 倍になります。
 - PDF ファイルのサイズも 5～6 倍になります。
-
-

有効値 72～300

デフォルト 72**使用上の注意**

- Windows では、レジストリを使用して、値を指定します。UNIX または Linux では、`reports.sh` で環境変数を設定します。
- 高い DPI 値を設定する場合は、`REPORTS_JVM_OPTIONS` の JVM ヒープ・サイズ値も変更し、JVM にエラー・メッセージ「Out Of Memory」が表示されないようにする必要があります。
- この環境変数は、PDF 出力およびプリンタ出力専用になっているため、Oracle Reports 配布機能ではサポートされていません。

B.1.45 REPORTS_IGNORE_IMAGE_TAG_RES

説明 この環境変数は、レポートにイメージの物理サイズ（通常は解像度とピクセル・サイズ）を格納できる特定のイメージ形式が含まれている場合に役立ちます。イメージが物理サイズに拡大縮小されないようにするために、この環境変数を `YES` に設定し、Oracle Reports でイメージ解像度情報が無視され、イメージのピクセル・サイズのみが使用されるようになります。これによって、データベース列のこのタイプのイメージが、サムネイルで表示されるかわりに適切に表示されるようになります。

有効値

- `YES` の場合、Oracle Reports でイメージ解像度情報が無視され、そのイメージのピクセル・サイズのみが使用されます。
- `NO` の場合、Oracle Reports でそのイメージのイメージ解像度情報は無視されません。

デフォルト NO

B.1.46 REPORTS_JPEG_QUALITY_FACTOR

説明 この環境変数は、JPEG イメージに適した画質のレベルを指定します。JPEG イメージの画質とイメージのサイズのトレードオフが制御されます。通常は、イメージの画質がよいほど、イメージ・ファイル・サイズが大きくなります。

有効値 0 ~ 100**デフォルト** 100（最高画質）**使用上の注意**

- Windows では、レジストリを使用して、値を指定します。UNIX または Linux では、`reports.sh` で環境変数を設定します。
- `REPORTS_JPEG_QUALITY_FACTOR` が指定されていないか不適切な値（文字列や範囲外の値など）に設定されている場合、デフォルト値が使用されます。
- 75 の値を指定すると、イメージの画質もよく、圧縮率も適切になります。

B.1.47 REPORTS_JVM_OPTIONS

説明 この環境変数は、Reports Builder、Reports Runtime または Reports Converter が JVM を起動するときの JVM オプションを指定します。たとえば、この環境変数を使用して JVM の開始ヒープ・サイズと最大ヒープ・サイズ、追加の `CLASSPATH` エントリなどを指定できます。

有効値 JVM コマンドライン構文の JVM オプションのリスト

デフォルト -Xmx256M

使用上の注意

- デフォルト値 -Xmx256M では、JVM ヒープ・サイズが 256MB に設定されており、大きなグラフ付きのレポートやサイズの大きいレポートを実行したときに、エラー・メッセージ Out of memory が表示されないように考慮されています。
- Reports Engine の起動時に、`server_name.conf` ファイルの `engine` 要素にある `jvmoptions` 属性で指定されている JVM オプションがチェックされます。詳細は、[第 3.2.1.4 項「engine」](#) を参照してください。`server_name.conf` で JVM オプションが設定されている場合は、それが `REPORTS_JVM_OPTIONS` 環境変数の値よりも優先されます。`server_name.conf` で JVM オプションが指定されていない場合は、Oracle Reports では、`REPORTS_JVM_OPTIONS` 環境変数で指定された JVM オプションが使用されます。
- Reports Server でレポートを実行する場合、JVM オプションは `REPORTS_JVM_OPTIONS` 環境変数を使用して設定できません。Reports Server の場合、JVM オプションはコマンドラインで `JVMOPTIONS` コマンドライン・キーワードを使用して設定します。詳細は、[第 A.3.46 項「JVMOPTIONS」](#) を参照してください。
- Reports Builder、Reports Runtime および Reports Converter でレポートを実行する場合、コマンドラインで `JVMOPTIONS` コマンドライン・キーワードを使用して指定された JVM オプションが、`REPORTS_JVM_OPTIONS` 環境変数で指定された JVM オプションよりも優先されます。

B.1.48 REPORTS_NETWORK_CONFIG

説明 この環境変数は、`rwclient`、`rwrqm`、`rwcgi` または OracleAS Forms Services で、カスタム・ネットワーク構成ファイルを使用する場合にのみ設定します。この環境変数が設定されていない場合、これらの実行可能ファイルでは、デフォルトのネットワーク構成ファイル (`rwnetwork.conf`) が使用されます。`rwnetwork.conf` の詳細は、[第 3.3.1 項「ネットワークの構成要素 \(rwnetworkconf.dtd\)」](#) を参照してください。

有効値 `ORACLE_HOME\reports\conf` にある有効なカスタム・ネットワーク構成ファイル

デフォルト `rwnetwork.conf`

B.1.49 REPORTS-NLS_XML_CHARSETS

説明 この環境変数により、レポートを XML 形式で保存する場合に使用するキャラクタ・セットのエンコードを優先的に定義するオプションが提供されます。この設定が必要になるのは、`NLS_LANG` と XML IANA 定義のキャラクタ・セット間で使用されるキャラクタ・セット・マッピングで、思いどおりの結果が得られない場合のみです。

XML パーサーが XML ファイル内の文字を認識できるようにするために、Oracle Reports では次のことが実行されます。

1. `NLS_CHARACTERSET` (`NLS_LANG` 変数のキャラクタ・セット部分) の値に基づき、XML 宣言にエンコード属性が追加されます。
2. `NLS_LANG` キャラクタ・セットとして設定されている値 (たとえば、`JA16SJIS`) が、XML 仕様で想定されている値 (たとえば、`Shift_JIS`) に変換されます。

このマッピングは、`REPORTS-NLS_XML_CHARSET` にエントリを追加することによって無効にできます。

有効値 セミコロンで区切ったマッピング・ペアのセット。最初の値には生成されるエンコードを指定し、2 番目の値にはマッピング値を指定します。

```
<old_name>=<new_name>[;<old_name>=<new_name>] [<old_name>=<new_name>]...
```

デフォルト 定義されていません

例 WISO-8859-8=ISO-8859-8-1;CSEUCKR=EUC-KR;WINDOWS-949=EUC-KR;EUC-CN=GBK;WINDOWS-936=GBK

B.1.50 REPORTS_NO_DUMMY_PRINTER

説明 この環境変数は、他のプリンタおよび表示環境変数および設定とともに、プリンタのかわりにシステムの画面とフォントを使用するかどうかを指定します。

有効値 TRUE | 設定しません

デフォルト TRUE

使用上の注意

- REPORTS_NO_DUMMY_PRINTER はデフォルトで設定され、常に設定されている必要があります。これが設定されていない場合は（ユーザーが変更した結果）、有効なプリンタが検出されず、REP-1800 エラー・メッセージが表示されます。代替として、有効なプリンタがない場合は、[TK_PRINT_STATUS](#) を使用することもできます。有効なプリンタからのレスポンスは、ファイルに出力している場合でも、Oracle Reports が出力を生成するために必要です。

Oracle Reports 10g 以降、uiprint.txt ファイルに有効なエントリが含まれていない（有効なプリンタが定義されていない）が、REPORTS_NO_DUMMY_PRINTER が設定されている場合、Oracle Reports では、uiscreenprint.txt で指定された screenprinter.ppd が使用されます。この環境変数の設定を解除する必要があるのは、有効なプリンタが存在せず、Oracle Reports で screenprinter.ppd ドライバを使用しない場合のみです。

関連項目： PostScript プリンタ・ドライバ screenprinter.ppd の詳細は、[第 3.10.1 項「ScreenPrinter」](#) を参照してください。

- この方法には制約があり、レポートが設計されたシステムとは異なるシステムからレポートを表示した場合、そのレポートのフォーマットが失われることがあります。また、このレポートを印刷した場合、フォントやメトリックが異なるため、正しくフォーマットされないこともあります。
- UNIX 上で印刷する場合、適切な印刷環境の作成には、いくつかの設定および構成が必要になります。詳細は、[第 5 章「UNIX での Oracle Reports による印刷」](#) を参照してください。

B.1.51 REPORTS_NO_HTML_SPACE_REPLACE

説明

この環境変数は、HTML または HTMLCSS の出力時に、空白を で置き換えないかどうかを指定します。

Oracle Reports では、フィールドに取得したデータの HTML メタデータ・キャラクタが適切なエンコードにマッピングされます。つまり、Oracle Reports は、次のように自動的にマッピングします。

- < を < に。
- " を " に。

通常は、ブラウザで正しい結果が生成され、スペースも適切に処理されます。しかし、ブラウザのスペース処理で、思いどおりの結果が得られないこともあります。たとえば、ユーザーがデータの前にインデントを入れた場合などに起こります。ブラウザでは複数のスペースが 1 つのスペースとして処理されるため、インデントは失われます。

有効値 YES | 設定しません

デフォルト 設定しません

使用上の注意

- 値を設定しない場合、すべてのスペースが ` ` に置き換わります。この場合、ブラウザでスペースに改行を入れるとき、出力で問題が発生します。また、生成される HTML ファイルのサイズも増加します。
- フィールドの「HTML タグを含む」プロパティが「はい」に設定されている場合、Oracle Reports は、フィールドの値をそのまま出力に渡すため、エンコードは行われません。

例

REPORTS_NO_HTML_SPACE_REPLACE が YES に設定されている場合、[Typical data output] 文の出力は次のようになります。

[Typical data output]

そして、次のように表示されます（先頭のスペースは無視されます）。

[Typical data output]

この環境変数を設定しない場合は、次のように出力が変換されます。

[Typical data output]

そして、次のように表示されます（先頭のスペースは維持されます）。

[Typical data output]

注意： 前述の例にある括弧は先頭の空白を表示するために使用されており、文章の一部ではありません。

B.1.52 REPORTS_OUTPUTIMAGEFORMAT

説明 この環境変数は、レポートで使用するデフォルトのイメージ形式を指定します。

有効値 GIF|JPEG|JPG|PNG|BMP

デフォルト JPEG

使用上の注意

- 指定した形式が出力タイプに一致するようにしてください。たとえば、BMP は RTF およびスプレッドシート出力にのみ適用します。HTML、HTMLCSS または PDF 出力には無効です。
- この環境変数の設定よりも、**OUTPUTIMAGEFORMAT** コマンドライン・キーワードの値が優先します。

詳細は、[第 3.10.2 項「拡張イメージングのサポート」](#) を参照してください。

B.1.53 REPORTS_PATH

説明 この環境変数は、Reports Runtime で参照されるファイルの検索パスを指定します。REPORTS_PATH で指定されたディレクトリが最初に検索され、その後で ORACLE_PATH で指定されたディレクトリが検索されます。この環境変数を使用することにより、レポートと、レポートに使用する外部オブジェクト (PL/SQL ライブラリ、外部問合せ、外部ポイラプレートなど) を検索できます。そのため、ディレクトリ・パスをハードコーディングしなくても、オペレーティング・システム間で容易に移動可能なレポートを作成できます。

REPORTS_PATH は、オペレーティング・システムの他の環境変数と同じ方法で定義し、パス長などのプラットフォーム固有のルールに従ってください。

REPORTS_PATH を定義する際には、ディレクトリ・パス以外にキーワード DB も指定できます。これにより、Reports Builder に対して、現在接続しているデータベースを検索するように指示できます。

たとえば、UNIX プラットフォームで次のように指定したとします。

```
setenv REPORTS_PATH /home/tkostin/pay:/home/tkostin/receive
```

Reports Builder では、最初にディレクトリ /home/tkostin/pay が検索されます。そして、該当するファイルが見つからなかった場合は、/home/tkostin/receive が検索されます。それでもファイルが見つからなかった場合は、Reports Builder セッションの開始時に接続したデータベースが検索されます。

有効値 任意のドライブのディレクトリ

デフォルト %ORACLE_HOME%\REPORT\TEMPLATES; %ORACLE_HOME%\REPORT\samples\demo; %ORACLE_HOME%\REPORT\integ;%ORACLE_HOME%\REPORT\Printers

使用上の注意

- REPORTS_PATH の最大入力文字数は 256 文字です。
- Reports Server 構成ファイル (server_name.conf) に含まれる engine 要素の sourceDir 属性にパスを指定すると、ここに設定した値よりも sourceDir 値が優先されます。

例

```
REPORTS_PATH=C:\oracle\apps\reports;C:\myfiles
```

B.1.54 REPORTS_RESOURCE

この環境変数は、レポートに必要なリソース・ファイルの位置を指定します。リソース・ファイルの位置を指定する際には、このパスにグローバリゼーション・サポート・ディレクトリの展開先を含む必要があります。

有効値 任意のドライブのディレクトリ

デフォルト %ORACLE_HOME%\reports\res\US\

使用上の注意

ロケールを変更するには、NLS_LANG 以外に、この環境変数も変更する必要があります。

例 米語ファイルの場合：

```
REPORTS_RESOURCE = %ORACLE_HOME%\reports\res\US\
```

日本語ファイルの場合：

```
REPORTS_RESOURCE = %ORACLE_HOME%\reports\res\JA\
```

B.1.55 REPORTS_SERVER

説明 この環境変数は、Web Cartridge または Web CGI リクエストに対するデフォルトの Reports Server を指定します。この環境変数が設定されているときは、レポート・リクエストで SERVER コマンドライン・キーワードを指定せず、デフォルトの Reports Server を使用してリクエストを処理できます。または、SERVER コマンドライン・キーワードを使用してデフォルトより優先することもできます。

この環境変数には下位互換性があります。

有効値 任意の Reports Server サービス・エントリ名

使用上の注意

この環境変数は、Common Gateway Interface (CGI) レポートの低位互換性のため、または Single Sign-On が使用されていない場合のために、Oracle Reports でサポートされています。

注意： CGI のサポートにかわり、JavaServer Pages (JSP) またはサーブレットの機能が実装されています。デフォルトの Reports Server 名は、[第 3.4.11 項「インプロセス・サーバーの識別」](#)の説明のように、Reports Servlet (rwservlet) 構成ファイル (rwservlet.properties) で定義されます。

例 REPORTS_SERVER=my_rep_server

B.1.56 REPORTS_SOLARIS_9

説明 この環境変数は、Solaris 2.9 上のネイティブ Motif ライブラリと JDK UI パッケージとの間の同期の問題を解決します。REPORTS_SOLARIS_9=YES を設定した場合、Reports Builder は想定されたとおりに応答します。Solaris 2.9 環境で REPORTS_SOLARIS_9=NO を設定した場合、レポート・ウィザードまたはデータ・ウィザードを起動したときに Reports Builder が応答を停止することがあります。

有効値 YES|NO

デフォルト Solaris 2.9 の場合は YES、他のプラットフォームの場合は NO

B.1.57 REPORTS_SPACE_BREAK

説明 この環境変数は、空白をデリミタとして処理するかどうかを指定します。Oracle Reports には、単語が 1 行に収まらない場合に、その行を適切に折り返すアルゴリズムが採用されています。このワード・ラップ・アルゴリズムでは、デフォルトで空白をデリミタとして処理します。

有効値 YES|NO

デフォルト YES

使用上の注意

この環境変数を NO に設定するのは、日本語や中国語のようなマルチバイト・キャラクタ・セットを使用するアジア系言語の場合のみです。これにより、Oracle Reports では、空白がデリミタとして処理されず、マルチバイト・キャラクタ・セットの言語に必要なワード・ラップ機能が適切に実行されます。

B.1.58 REPORTS_SRWRUN_TO_SERVER**説明**

この環境変数は、SRW.RUN_REPORT 組込みプロシージャを使用してレポートを実行するときに、SERVER または USERID キーワードを許可するかどうかを指定します。

有効値 YES | 設定しません

デフォルト 設定しません

使用上の注意

- Oracle Reports 10g リリース 2 (10.1.2) では、キーワード SERVER および USERID を SRW.RUN_REPORT とともに使用することは薦められません。これらのキーワードを

SRW.RUN_REPORT とともに使用するレポートを以前のリリースで作成した場合は、REPORTS_SRWRUN_TO_SERVER=YES を設定して、これらのレポートを引き続き Oracle Reports 10g リリース 2 (10.1.2) で実行できます。

- 以前のリリースで作成したレポートを、非同期で実行しようとする問題が発生することがあります。このため、できるかぎり早くレポートを Oracle Reports 10g リリース 2 (10.1.2) に移行することが重要です。

注意： SRW.RUN_REPORT 組込みプロシージャを含む SRW 組込みパッケージの説明は、Oracle Reports のオンライン・ヘルプを参照してください。

B.1.59 REPORTS_SSLPORT

説明 この環境変数は、SSL を使用する場合のポート番号を指定します。

有効値 有効なポート番号

デフォルト 443

使用上の注意

この環境変数は、Common Gateway Interface (CGI) レポートの下位互換性のため、または Single Sign-On が使用されていない場合のために、Oracle Reports でサポートされています。

注意： CGI のサポートにかわり、JavaServer Pages (JSP) またはサーブレットの機能が実装されています。

例 REPORTS_SSLPORT=442

B.1.60 REPORTS_SYS_AUTH

説明 この環境変数は、制限付き Reports Server に対してレポート・リクエストを実行する場合に、ユーザー名とパスワードの認証に使用する認証テンプレートを指定します。

有効値 特別の認証アクションを含む HTML ファイル (デフォルト値の使用が推奨されます)

デフォルト sysauth.htm

使用上の注意

この環境変数は、Common Gateway Interface (CGI) レポートの下位互換性のため、または Single Sign-On が使用されていない場合のために、Oracle Reports でサポートされています。

注意： CGI のサポートにかわり、JavaServer Pages (JSP) またはサーブレットの機能が実装されています。

例 REPORTS_SYS_AUTH=sysauth.htm

B.1.61 REPORTS_TAGLIB_URI

説明 この環境変数は、JSP ベースのレポートの Web ソースに使用されるタグ接頭辞の位置を指定します。これにより、.jsp ファイルのタグ・ライブラリ (TAGLIB) 宣言の Reports URI が定義されます。これは、通常次のように定義されています。

```
<%@ taglib uri="/WEB-INF/lib/reports_tld.jar" prefix="rw" %>
```

Oracle Reports が環境変数と一致する "uri" を見つけると、対応する接頭辞属性を使用して、jsp ファイル内の Oracle Reports タグを識別します。

有効値 Oracle Reports タグ・ライブラリを参照する任意の "uri"

デフォルト /WEB-INF/lib/reports_tld.jar

使用上の注意

通常、デフォルト値は変更しません。これは、JDeveloper と Oracle Reports のどちらのレポート・ファイルでも同様です。接頭辞属性を変更して、"uri" 属性に依存しないように、ネーミングの競合を回避できます。

B.1.62 REPORTS_TMP

説明 この環境変数は、Reports Builder 一時ファイルの保存先ディレクトリを指定します。この目的において、Reports Builder が使用するディレクトリは 1 つのみであるため、2 つ以上定義しないでください。

REPORTS_TMP は、基本オペレーティング・システムの他の環境変数と同じ方法で定義し、パス長などのプラットフォーム固有のルールに従ってください。REPORTS_TMP を定義しない場合は、デフォルトで現行の作業ディレクトリが使用されます。

有効値 任意のドライブのディレクトリ

デフォルト 定義されていません

例 REPORTS_TMP=C:¥tmp

B.1.63 REPORTS_USEREXITS

説明 この環境変数は、Oracle Reports によって使用されるライブラリを指定します。これらのライブラリは、ユーザーが作成したプログラム・モジュールで、Oracle Reports によってコールされます。

REPORTS_USEREXITS では、複数のライブラリを指定できます。Windows では、パス内でディレクトリを区切るには円記号 (¥) を使用し、パスを区切るにはセミコロン (;) を使用します。UNIX では、パス内でディレクトリを区切るにはスラッシュ (/) を使用し、パスを区切るにはコロンの (:) を使用します。

この値が明示的に設定されない場合、Oracle Reports では、システムのパス変数に従って rwxtb.dll が検索されます。

注意: Oracle Reports 10g では、ORA_JAVA 組込みパッケージおよび Java Importer を使用して Java メソッドをコールできます。これによってレポートでのユーザー・イグジットの必要性が軽減され、さらにオープンでポータブルな配置が可能になります。ORA_FFI 組込みパッケージも使用できます。これにより、動的ライブラリの C 関数を起動するための外部関数インタフェースが提供されます。これらの組込みパッケージが使用可能になったことにより、ユーザー・イグジットの使用は Oracle Reports では推奨されなくなりました。ただし、makefile は、既存のユーザー・イグジットを引き続き使用できるようにするために提供されています。

下位互換性のために、この環境変数には以前の名前 REPORTS_USEREXIT を使用できます。

有効値 任意のユーザー・イグジット・ライブラリ (絶対パス)

デフォルト 定義されていません

例

Windows の場合：

```
REPORTS_USEREXITS=C:¥mydll.dll;d:¥mynew.dll;e:¥bin¥speed.dll
```

UNIX の場合：

```
REPORTS_USEREXITS=/usr/oracle/mylib.so:/usr/oracle/myfolder/speed.so
```

B.1.64 REPORTS_UTF8_XMLOUTPUT

説明 この環境変数は、NLS_LANG キャラクタ・セットのかわりに UTF8 キャラクタ・セットを使用するかどうかを指定します。この環境変数が有効になるのは、エンコード属性が「XML Prolog 値」プロパティで指定されていない場合のみです（「XML Prolog 値」プロパティの説明は、Oracle Reports のオンライン・ヘルプを参照してください）。

有効値

YES の場合、UTF8 キャラクタ・セットを割り当てます（「XML Prolog 値」プロパティが設定されていない場合）。

NO の場合、NLS_LANG（または IANA 定義）のキャラクタ・セットを割り当てます（「XML Prolog 値」プロパティが設定されていない場合）。

デフォルト YES

B.1.65 RW

この環境変数は、ORACLE_HOME 内のレポート固有のディレクトリを指定します。

有効値 有効なディレクトリ名

デフォルト

```
%ORACLE_HOME%¥reports (Windows)
```

```
$ORACLE_HOME/reports (UNIX)
```

B.1.66 TK_PRINT

説明 (UNIX のみ) この環境変数は、UNIX で動作する Oracle Reports 6i の印刷コマンドを指定します。Oracle Reports の後のリリースでは、TK_PRINT は廃止されました。同じ結果を得るには、印刷スクリプト・ファイル ORACLE_HOME/bin/rwlpr.sh を使用します。このスクリプトでは、デフォルトで lp と lpr コマンドがサポートされます。ご使用のマシンに他の印刷コマンドを使用している場合は、それに合せてこのファイルを変更する必要があります。

有効値 PRINT コマンドと使用している UNIX に応じたすべての必要なキーワード。これには次の要素が含まれます。

- %n は、プリンタ名の文字列です。
- %c は、印刷部数です。

この文字列は、printf() 書式とほぼ同じです。この環境変数が設定されていない場合、Oracle Reports 6i は、プラットフォームの標準のデフォルト値を使用します。プラットフォームごとのデフォルト値の例を次に示します。

System V: lp -s -d'%n' -n%c

Solaris: lpr -P'%n' -#%c -s

デフォルト 定義されていません

使用上の注意

- デフォルトの印刷コマンドで通常のニーズに対応できます。この環境変数を設定するのは、デフォルト値を変更する特別な必要性がある場合のみにしてください。たとえば、出力を多重化する場合は、TK_PRINT に設定する必要があります。
- UNIX 上で印刷する場合、適切な印刷環境の作成には、いくつかの設定および構成が必要になります。詳細は、第 5 章「UNIX での Oracle Reports による印刷」を参照してください。

B.1.67 TK_PRINT_STATUS

説明 (UNIX のみ) この環境変数は、プリンタを有効化するために実行するコマンドを指定します。プリンタを有効化するには、このコマンドを実行し、その出力から **unknown**、**non-existent** または **invalid** の文字列を検索します。このいずれかの文字列が出力に表示された場合、プリンタは無効で、選択できません。そうでない場合は、プリンタは Oracle Reports で使用できます。

有効値 プリンタ名の %n を含める必要があります (「TK_PRINT」も参照)。

この環境変数が設定されていない場合、Oracle Reports では、ビルトインのデフォルト値が使用されます。

System V: /usr/bin/lpstat -p'%n' 2>&1

その他: /usr/etc/lpc status '%n' 2>&1

デフォルト 定義されていません

使用上の注意

- この環境変数を使用するのは、使用しているプラットフォームのプリンタのステータス・コマンドがデフォルト値と異なる場合や、有効なプリンタが存在しない場合のみにしてください。有効なプリンタが存在しない場合、TK_PRINT_STATUS=echo を設定し、uiprint.txt ファイルにダミー・エントリを指定できます。これにより、Oracle Reports では、プリンタの確認時に有効なレスポンスを得ることができます。
- **REPORTS_NO_DUMMY_PRINTER** が設定されていても、uiprint.txt ファイルに有効なエントリが含まれていない場合は、uiscreenprint.txt に指定されている screenprinter.ppd が使用されます。

注意: REPORTS_NO_DUMMY_PRINTER はデフォルトで設定され、常に設定されている必要があります。これが設定されていない場合は (ユーザーが変更した結果)、REP-1800 エラー・メッセージが表示されます。

関連項目: PostScript プリンタ・ドライバ screenprinter.ppd の詳細は、第 3.10.1 項「ScreenPrinter」を参照してください。

- UNIX 上で印刷する場合、適切な印刷環境の作成には、いくつかの設定および構成が必要になります。詳細は、第 5 章「UNIX での Oracle Reports による印刷」を参照してください。

B.1.68 TK_PRINTER

説明 (UNIX のみ) この環境変数は、デフォルトのプリンタ名を指定します。

有効値 デフォルトのプリンタ名

デフォルト 定義されていません

使用上の注意

- TK_PRINTER は、PRINTER よりも優先します。つまり、両方の変数が設定されている場合は、TK_PRINTER が最初に参照されます。PRINTER が参照されるのは、TK_PRINTER に有効なプリンタが指定されていない場合のみです。TK_PRINTER と PRINTER の両方に、有効なプリンタが設定されていない場合は、Oracle Reports では uiprint.txt ファイルの最初のエントリが使用されます。REPORTS_NO_DUMMY_PRINTER が設定されており、uiprint.txt ファイルに有効なエントリが含まれていない場合は、uiscreenprint.txt に指定されている screenprinter.ppd が使用されます。

注意： REPORTS_NO_DUMMY_PRINTER はデフォルトで設定され、常に設定されている必要があります。これが設定されていない場合は（ユーザーが変更した結果）、REP-1800 エラー・メッセージが表示されます。

関連項目： PostScript プリンタ・ドライバ screenprinter.ppd の詳細は、第 3.10.1 項「ScreenPrinter」を参照してください。

- UNIX 上で印刷する場合、適切な印刷環境の作成には、いくつかの設定および構成が必要になります。詳細は、第 5 章「UNIX での Oracle Reports による印刷」を参照してください。

B.1.69 TK_AFM

説明 この環境変数は、AFM ファイルの位置を指定します。TK_AFM の後で、ORACLE_AFM が参照されます。

有効値 任意のドライブのディレクトリ

デフォルト 定義されていません

使用上の注意

- いずれの変数にも値を指定しない場合、Oracle Reports では、次のディレクトリの AFM ファイルが検索されます。
ORACLE_HOME/guicommon/tk/admin/AFM
- UNIX 上で印刷する場合、適切な印刷環境の作成には、いくつかの設定および構成が必要になります。詳細は、第 5 章「UNIX での Oracle Reports による印刷」を参照してください。

B.1.70 TK_HPD

説明 この環境変数は、HPD ファイルの位置を指定します。TK_HPD の後で、ORACLE_HPD が参照されます。

有効値 任意のドライブのディレクトリ

デフォルト 定義されていません

使用上の注意

- いずれの変数にも値を指定しない場合、Oracle Reports では、次のディレクトリの HPD ファイルが検索されます。
ORACLE_HOME/guicommon/tk/admin/HPD

- UNIX上で印刷する場合、適切な印刷環境の作成には、いくつかの設定および構成が必要になります。詳細は、第5章「UNIXでの Oracle Reportsによる印刷」を参照してください。

B.1.71 TK_PPD

説明 この環境変数は、PPDファイルの位置を指定します。TK_PPDの後で、ORACLE_PPDが参照されます。

有効値 任意のドライブのディレクトリ

デフォルト 定義されていません

使用上の注意

- いずれの変数にも値を指定しない場合、Oracle Reportsでは、次のディレクトリのPPDファイルが検索されます。

`ORACLE_HOME/guicommon/tk/admin/PPD`

- UNIX上で印刷する場合、適切な印刷環境の作成には、いくつかの設定および構成が必要になります。詳細は、第5章「UNIXでの Oracle Reportsによる印刷」を参照してください。

B.1.72 TK_TFM

説明 この環境変数は、TFMファイルの位置を指定します。TK_TFMの後で、ORACLE_TFMが参照されます。

有効値 任意のドライブのディレクトリ

デフォルト 定義されていません

使用上の注意

- いずれの変数にも値を指定しない場合、Oracle Reportsでは、次のディレクトリのTFMファイルが検索されます。

`ORACLE_HOME/guicommon/tk/admin/TFM`

- UNIX上で印刷する場合、適切な印刷環境の作成には、いくつかの設定および構成が必要になります。UNIX上でOracle Reportsを使用した印刷の詳細は、第5章「UNIXでの Oracle Reportsによる印刷」を参照してください。

B.1.73 USERNAME

説明 この環境変数は、デフォルトのログオン・アカウントを指定します。USERNAMEの設定の詳細は、ご使用のデータベースのドキュメントを参照してください。

有効値 有効な Oracle ユーザー名 (OP\$ 接頭辞なし)

デフォルト 定義されていません

例 USERNAME=dsanvita

B.1.74 USER_NLS_LANG

説明 この環境変数は、Oracle Reports Runtime コンポーネントの言語を指定します。この環境変数とその有効値の詳細は、[第 18 章「グローバル化と双方向のサポートの実装」](#)を参照してください。

OracleAS Portal へのレポートのバッチ登録

OracleAS Portal に多数のレポートを登録する場合は、OracleAS Portal ユーザー・インタフェースで個別に登録するのではなく、バッチ・スクリプトを使用してまとめて登録できます。同様に、多数のレポートを登録解除する場合にも、バッチ・スクリプトを使用すれば効率的です。

- [レポート定義ファイルのバッチ登録](#)
- [レポート・パッケージのバッチ削除](#)
- [PL/SQL バッチ登録ファンクション](#)

C.1 レポート定義ファイルのバッチ登録

OracleAS Portal でレポートをバッチ登録するには、次の手順を実行します。

1. [rwconverter](#) の実行による SQL スクリプトの生成
2. [SQL*Plus](#) でのスクリプトの実行

C.1.1 rwconverter の実行による SQL スクリプトの生成

SQL*Plus で実行可能なレポート登録用 SQL スクリプトを生成する手順は、次のとおりです。

1. オペレーティング・システム・プロンプト (DOS または UNIX) から、`rwconverter` コマンドをキーワードとともに入力して、レポート定義ファイルをバッチ登録します。

関連項目： `rwconverter` キーワードの詳細は [付録 A 「コマンドライン・キーワード」](#) を参照してください。

注意： 必要なロード関数を持つスクリプト・ファイルを作成するには、`DTYPE`、`STYPE`、`SOURCE` および `DEST` キーワードを指定します。
OracleAS Portal で機能パッケージを作成するには、スクリプト・ファイルの作成に使用したキーワードに加え、`P_SERVERS`、`P_PRIVILEGE`、`P_TYPES`、`P_FORMATS` キーワードを指定する必要があります。

次に、Microsoft Windows での `rwconverter` コマンドラインの例を示します。

```
rwconverter.exe dtype="register" stype="rdffile"
source="(security.rdf,earnings.rdf,acc_pay.rdf)" dest="(output.sql)"
p_owner="PORTAL_APP" p_servers="(repserver,acct_server)"
p_description="restricted report" p_privilege="(SCOTT,JABERS,ACCT)"
p_availability="production" p_types="(Cache,printer)"
p_formats="(HTMLCSS,PDF)" p_printers="(sales_printer,acct_printer)"
p_pformTemplate="public.finance_template"
p_trigger="Is begin IF UPPER(DESTYPE) = 'PRINTER' AND
EMPNAME = 'SMITH' THEN RETURN(TRUE); ELSE RETURN(FALSE); END IF; end;"
```

前述のコマンドラインを指定すると、`output.sql` という名前で、次の内容を含む SQL スクリプト・ファイルが生成されます。

```
SET SERVEROUTPUT ON
```

```
VAR STATUS NUMBER;
```

```
EXEC :STATUS := RWWWVREG.REGISTER_REPORT (P_NAME=>'Security',
P_OWNER=>'PORTAL_APP', P_SERVERS=>'repserver,acct_server',
P_FILENAME=>'security.rdf', P_DESCRIPTION=>'restricted report',
P_PRIVILEGE=>'SCOTT,JABERS,ACCT', P_AVAILABILITY=>'production'
P_TYPES=>'Cache,printer', P_FORMATS=>'HTMLCSS,PDF',
P_PRINTERS=>'sales_printer,acct_printer
P_PFORMTEMPLATE=>'public.finance_template' P_PARAMETERS=>'(P_LASTNAME)
(P_SSN)', P_TRIGGER=>'Is begin IF UPPER(DESTYPE) = 'PRINTER' AND
EMPNAME = 'SMITH' THEN RETURN(TRUE); ELSE RETURN(FALSE); END IF; end;');
```

```
EXEC :STATUS := RWWWVREG.REGISTER_REPORT (P_NAME=>'Earnings',
P_OWNER=>'PORTAL_APP', P_SERVERS=>'repserver,acct_server',
P_FILENAME=>'earnings.rdf', P_DESCRIPTION=>'restricted report',
P_PRIVILEGE=>'SCOTT,JABERS,ACCT', P_AVAILABILITY=>'production'
P_TYPES=>'Cache,printer)', P_FORMATS=>'HTMLCSS,PDF',
P_PRINTERS=>'sales_printer,acct_printer',
P_PFORMTEMPLATE=>'public.finance_template',
P_TRIGGER='Is begin IF UPPER(DESTYPE) = 'PRINTER' AND EMPNAME = 'JABERS'
THEN RETURN(TRUE); ELSE RETURN(FALSE); END IF; end;');
```

```
EXEC :STATUS := RWWWVREG.REGISTER_REPORT (P_NAME=>'Acc_pay',
P_OWNER=>'PORTAL_APP', P_SERVERS=>'repserver,acct_server',
P_FILENAME=>'acc_pay.rdf', P_DESCRIPTION=>'restricted report',
P_PRIVILEGE=>'SCOTT,JABERS,ACCT', P_AVAILABILITY=>'production'
P_TYPES=>'Cache,printer', P_FORMATS=>'HTMLCSS,PDF',
p_printers=>'sales_printer,acct_printer',
P_PFORMTEMPLATE=>'public.finance_template'
P_TRIGGER=>'Is begin IF UPPER(DESTYPE) = ''PRINTER'' AND
EMPNAME = ''JABERS'' THEN RETURN(TRUE); ELSE RETURN(FALSE); END IF; end;');
```

この SQL スクリプト・ファイルの内容の詳細は、[第 C.3 項「PL/SQL バッチ登録ファンクション」](#)を参照してください。

- reports.log ファイルで、変換処理中に発生したエラーがないかどうかを確認します。このファイルは通常、現在作業中のディレクトリに作成されます。reports.log ファイルが生成されなかった場合、rwconverter で発生したエラーがないことを示します。
- ここで、システム・パラメータおよびユーザー・パラメータの値を必要に応じて編集できます。たとえば、前述のサンプル・スクリプトで最初の RWWWVREG ファンクションにより、P_PARAMETERS という追加パラメータが生成されました。この処理が行われた理由は、security.rdf ファイルに、P_LASTNAME と P_SSN という 2 つのユーザー定義パラメータが次のように記述されているためです。

```
P_PARAMETERS=>' (P_LASTNAME) (P_SSN) ',
```

この場合、デフォルト、下限、上限の各値をオプションで定義したり、各ユーザー・パラメータの値リストを定義して、実行時にユーザーが入力できる値を制限できます。同様に、システム・パラメータを制限する場合には（たとえば、COPIES を使用してユーザーが作成できるコピーの数を制限する場合）、P_PARAMETERS キーワードを使用します。編集後の P_PARAMETERS キーワードは、次のようになります。

```
P_PARAMETERS=>' (P_LASTNAME, LOV=LASTNAME_LOV) (P_SSN) (COPIES,
DEFAULT=1,LOW=1,HIGH=2) '
```

この修正後のコード・セグメントでは、レポートは次のように制限されます。

- P_LASTNAME ユーザー・パラメータは、LASTNAME_LOV 値リストに含まれる値に制限されます。
 - P_SSN にユーザーが指定する値が必要です。
 - COPIES システム・パラメータのデフォルト値は 1、また印刷部数は 1 か 2 である必要があります。
- output.sql ファイルを保存して閉じます。

C.1.2 SQL*Plus でのスクリプトの実行

レポートを OracleAS Portal に実際に登録するには、rwconverter により生成したスクリプトを実行する必要があります。

- SQL*Plus を開始し、パッケージ・プロシージャを所有させる OracleAS Portal スキーマに接続します。
- SQL*Plus コマンド・プロンプトから、rwconverter で作成したスクリプトを実行します。

```
@ output.sql
```

このスクリプトにより、スクリプトに示された各レポートについて、指定されたパラメータを使用して OracleAS Portal でパッケージが実行および作成されます。

- RW_ADMINISTRATOR 権限を持つユーザーとして OracleAS Portal にログインします。
- 「企業ドキュメント」タブをクリックします。

5. 「ビルダー」をクリックします。
6. 「管理」タブをクリックします。
7. 「Oracle Reports セキュリティ」ポートレットで、「Oracle Reports セキュリティの設定」をクリックします。
8. 「レポート定義ファイル・アクセス」ポートレットで、SQL スクリプトでバッチ登録したいどれかのレポートの P_NAME を入力します。
9. 「編集」をクリックします。「コンポーネントの管理」ページが表示されます。
10. ページ下部の「編集」をクリックして、レポートのパラメータを編集します。
11. パラメータを必要に応じて編集します。
12. 「OK」をクリックします。
13. 「閉じる」をクリックします。
14. スクリプトでバッチ登録したレポートごとに手順 8 ～ 13 を繰り返します。

C.2 レポート・パッケージのバッチ削除

OracleAS Portal から多数のレポートをバッチ削除する手順は、次のとおりです。

1. テキスト・エディタで、削除するレポート定義ファイル・パッケージごとに、1 つの RWWWVREG.DEREGISTER_REPORT ファンクション・コールを含む SQL スクリプト・ファイル（例：rmv_rdfs.sql）を作成します。次に例を示します。

```
VAR STATUS NUMBER;  
EXEC :STATUS := RWWWVREG.DEREGISTER_REPORT (P_NAME=>'Security');  
EXEC :STATUS := RWWWVREG.DEREGISTER_REPORT (P_NAME=>'Earnings');  
EXEC :STATUS := RWWWVREG.DEREGISTER_REPORT (P_NAME=>'Acc_pay');
```

注意： P_NAME は、OracleAS Portal から削除するレポート定義ファイル・パッケージの名前です。

2. SQL*Plus を開始し、レポートのパッケージ・プロシージャを所有する OracleAS Portal スキーマにログインします。
3. SQL*Plus コマンド・プロンプトから、最初の手順で作成したスクリプトを実行します。

```
@ rmv_rdfs.sql
```

このスクリプトは、スクリプトに示された各レポートについて、OracleAS Portal からパッケージを削除します。

注意： このプロシージャを実行しても、レポート定義ファイルはファイル・システムから削除されません。レポートが登録解除され、OracleAS Portal から利用できなくなるだけです。レポート定義ファイルを削除するには、ユーザーがファイル・システムから削除する必要があります。

C.3 PL/SQL バッチ登録ファンクション

レポートを Oracle Application Server にバッチ登録する目的で `rwconverter` によって生成された SQL スクリプトは、主に `RWWWVREG.REGISTER_REPORT` ファンクションへのコールで構成されています。`RWWWVREG.REGISTER_REPORT` の構文を次に示します。

```
Function Rwwwvreg.register_report (
  p_owner varchar2,
  p_name varchar2,
  p_servers varchar2,
  p_filename varchar2,
  p_description varchar2,
  p_privileges varchar2,
  p_availability varchar2,
  p_types varchar2,
  p_formats varchar2,
  p_printers varchar2,
  p_pdfformTemplate varchar2,
  p_parameters varchar2,
  p_trigger varchar2)
return number;
  -- =0 : succeeded;
  -- !=0 : failed;
```

次の表に、`RWWWVREG.REGISTER_REPORT` によって取得されるパラメータを示します。

表 C-1 RWWWVREG.REGISTER_REPORT のパラメータ

パラメータ	説明
P_OWNER	DB プロバイダ名。デフォルトは、SQL*Plus スクリプト開始時に接続する現行の Oracle Application Server DB プロバイダです。 次に例を示します。 P_OWNER=>'PORTAL_APP'
P_NAME	OracleAS Portal でのレポートの識別に使用される名前。 P_NAME は、「レポート定義ファイルのアクセス権限を作成」ウィザードの「名前」フィールドに対応しています。 次に例を示します。 P_NAME=>'Earnings'
P_SERVERS	P_SERVERS パラメータで定義されたレポート定義ファイルがアクセス権を持つ Reports Server の名前。Reports Server のリストはカンマで区切ります。 P_SERVERS は、「レポート定義ファイルのアクセス権限を作成」ウィザードおよび「レポート定義ファイルの編集」ページの「Reports Server」フィールドに対応しています。 次に例を示します。 P_SERVERS=>'repserver,acct' 注意: P_SERVERS で示す Reports Server は、事前に OracleAS Portal に登録されている必要があります。詳細は、 第 12 章「OracleAS Portal でのレポートの配布」 を参照してください。
P_FILENAME	登録するレポート定義ファイルの名前。 P_FILENAME は、「レポート定義ファイルのアクセス権限を作成」ウィザードおよび「レポート定義ファイルの編集」ページの「Oracle Reports ファイル名」に対応しています。 次に例を示します。 P_FILENAME=>'earnings.rdf'

表 C-1 RWWWVREG.REGISTER_REPORT のパラメータ (続き)

パラメータ	説明
P_DESCRIPTION	<p>レポートの説明。</p> <p>P_DESCRIPTION は、「レポート定義ファイルのアクセス権限を作成」ウィザードおよび「レポート定義ファイルの編集」ページの「説明」フィールドに対応しています。</p> <p>次に例を示します。</p> <pre>P_DESCRIPTION=>'restricted report'</pre>
P_PRIVILEGE	<p>P_FILENAME で定義されたレポート定義ファイルの実行に必要な権限を持つユーザーまたはロール。このリストはカンマで区切ります。</p> <p>P_PRIVILEGE は、「コンポーネントの管理」ページの「アクセス」タブの、レポートに対する「ユーザー/ロール」リストに対応しています。「ユーザー/ロール」リストを表示するには、「Portal DB プロバイダから権限を継承」のチェックを外す必要があります。</p> <p>次に例を示します。</p> <pre>P_PRIVILEGE=>'SCOTT,JABERS,PORTAL90'</pre>
P_AVAILABILITY	<p>P_FILENAME パラメータで定義されたレポート定義ファイルがいつ処理できるようになるかを決定する使用可能日カレンダーの名前。</p> <p>P_AVAILABILITY は、「レポート定義ファイルのアクセス権限を作成」ウィザードおよび「レポート定義ファイルの編集」ページの「使用可能日カレンダー名」フィールドに対応しています。</p> <p>次に例を示します。</p> <pre>P_AVAILABILITY=>'production'</pre> <p>注意: 使用可能日カレンダーは、事前に OracleAS Portal に登録されている必要があります。使用可能日カレンダーの作成の詳細は、第 12 章「OracleAS Portal でのレポートの配布」を参照してください。</p>
P_TYPES	<p>P_FILENAME パラメータで定義されたレポート定義ファイルを送信できる宛先タイプ (キャッシュ、プリンタなど)。このリストはカンマで区切ります。</p> <p>P_TYPES は、「レポート定義ファイルのアクセス権限を作成」ウィザードおよび「レポート定義ファイルの編集」ページの「タイプ」複数選択ボックスに対応しています。</p> <p>次に例を示します。</p> <pre>P_TYPES=>'CACHE,printer'</pre>
P_FORMATS	<p>P_FILENAME パラメータで定義されたレポート定義ファイルを送信できる宛先フォーマット (HTML、PDF など)。このリストはカンマで区切ります。</p> <p>P_FORMATS は、「レポート定義ファイルのアクセス権限を作成」ウィザードおよび「レポート定義ファイルの編集」ページの「書式」複数選択ボックスに対応しています。</p> <p>次に例を示します。</p> <pre>P_FORMATS=>'HTMLCSS,PDF'</pre> <p>注意: レポートの宛先フォーマットが DELIMITEDDATA である場合は、レポートをバッチ登録できないことがあります。回避策として、異なる宛先フォーマットを定義して、レポートをバッチ登録し、後で手動でレポートを DESFORMAT=DELIMITEDDATA に編集できます。</p>

表 C-1 RWWWVREG.REGISTER_REPORT のパラメータ (続き)

パラメータ	説明
P_PRINTERS	<p>P_FILENAME パラメータで定義されたレポート定義ファイルを印刷できるプリンタ。このリストはカンマで区切ります。</p> <p>P_PRINTERS は、「レポート定義ファイルのアクセス権限を作成」ウィザードおよび「レポート定義ファイルの編集」ページの「プリンタ」複数選択ボックスに対応しています。</p> <p>次に例を示します。</p> <pre>P_PRINTERS=>'sales_printer,acct_printer'</pre> <p>注意: P_PRINTERS で示すプリンタは、事前に OracleAS Portal に登録されている必要があります。詳細は、第 12 章「OracleAS Portal でのレポートの配布」を参照してください。</p>
P_PFORMTEMPLATE	<p>ランタイム・パラメータ・フォームのページ・スタイルを決定するパラメータ・フォーム・テンプレート。</p> <p>P_PFORMTEMPLATE は、「レポート定義ファイルのアクセス権限を作成」ウィザードおよび「レポート定義ファイルの編集」ページの「パラメータ・フォーム・テンプレート」フィールドに対応しています。</p> <p>次に例を示します。</p> <pre>P_PFORMTEMPLATE=>'public.finance_template'</pre>
P_PARAMETERS	<p>ユーザー・パラメータおよびシステム・パラメータのデフォルト値、上限値および下限値、あるいは値リストの名前。</p> <p>注意: P_PARAMETERS パラメータには、対応する <code>rwconverter</code> オプションがありません。このため、ユーザー・パラメータの値、範囲、または値リストをバッチ・インポートする場合には、<code>rwconverter</code> によって生成された SQL スクリプトを手動で編集する必要があります。</p> <p>P_PARAMETERS は、「レポート定義ファイルのアクセス権限を作成」ウィザードおよび「レポート定義ファイルの編集」ページの「(パラメータの) 名前」、「LOV」、「下限」および「上限」フィールドに対応しています。</p> <p>デフォルトは、指定されたパラメータのランタイム・パラメータ・フォームに設定されている値に対応しています。</p> <p>次に例を示します。</p> <pre>P_PARAMETERS=>(P_LASTNAME, LOV=LASTNAME_LOV) (P_SSN) (COPIES, DEFAULT=1,LOW=1,HIGH=2)'</pre> <p>ここでは</p> <p>P_LASTNAME、P_SSN および COPIES は、パラメータ名です。</p> <p>LOV は、値リスト名です。</p> <p>DEFAULT は、デフォルト値です。</p> <p>LOW は、値範囲の下限です。</p> <p>HIGH は、値範囲の上限です。</p>

表 C-1 RWWWVREG.REGISTER_REPORT のパラメータ (続き)

パラメータ	説明
P_TRIGGER	<p>PL/SQL で記述された妥当性チェック・トリガーで、ブール文 (TRUE (成功した場合) または FALSE (失敗した場合)) を返します。</p> <p>P_TRIGGER は、「レポート定義ファイルのアクセス権限を作成」ウィザードおよび「レポート定義ファイルの編集」ページのテキスト・ボックスに対応しています。</p> <p>次に例を示します。</p> <pre>P_TRIGGER=>'Is begin IF UPPER(DESTYPE) = 'PRINTER' AND EMPNAME = 'SMITH' THEN RETURN(TRUE); ELSE RETURN(FALSE); END IF; end;'</pre>

OracleAS Reports Services の トラブルシューティング

この付録では、OracleAS Reports Services を使用してレポートを配布する際に発生する可能性のある一般的な問題とその解決方法について説明します。また、問題の診断手順についても説明します。この付録には、次の項があります。

- 問題と解決策
- パフォーマンスの問題の診断
- フォントの問題の診断
- 印刷の問題の診断
- JDBC PDS の問題の診断
- OracleAS Portal の問題の診断
- グローバリゼーションの問題の診断
- Oracle Reports ブリッジの問題の診断
- その他の解決策

D.1 問題と解決策

この項では、一般的な問題と解決策について説明します。この項は次のトピックで構成されています。

- レポート・リクエストのハング
- Reports Server のアクティビティによりエラー REP-50125 が生成される
- Reports Servlet を使用して長期レポートを実行すると失敗する
- プラットフォームによってフォントの表示が異なる
- UNIX プラットフォームでレポートを実行すると REP-56048 が生成される
- 過負荷状態で OPMN を使用するとインプロセス・サーバーに障害が発生する
- 右から左に表示する言語のフォントの問題
- RUN_REPORT_OBJECT を使用して Oracle Forms からレポートを実行する際のエラー
- Microsoft Excel でのレポート出力の表示
- ユーザー・イグジットが含まれるレポートの UNIX での障害
- インプロセス・サーバーを使用する際の印刷エラーとフォント・エラー

D.1.1 レポート・リクエストのハング

Reports Server でレポート・リクエストを実行すると、レポート・リクエストが様々な理由でハングすることがあります。このハングが適切なタイミングで検出されない場合は、安定性が損なわれる場合があります。ここでは、このようなシナリオに焦点をあて、問題を解説し、このようなパターンの特定方法と対処方法を示します。さらに、オラクル社カスタマ・サポート・センターに問題を報告する際に必要な情報の収集方法についても説明します。

まず第一に、Reports Server でどのように重複ジョブが識別されるかを理解することが重要です。ジョブが Reports Server に送信されると、ジョブ・キューに同様のジョブが存在していないかどうかチェックされます。送信されたジョブが現在実行中のジョブと同一の場合、Reports Server は送信されたジョブを重複ジョブ、現在実行中のジョブをマスター・ジョブと判断します。Reports Server は重複ジョブを実行することなく、かわりにマスター・ジョブの終了を待機してから、同じ出力を重複ジョブに渡します。アイドル・エンジンが利用可能であっても、重複ジョブはエンジンに送信されません。これは正常な動作であり、リクエストがハングしたわけではありません。

この項で説明する解決策以外の詳細は、第 20.2 項「Reports Server の構成のチューニング」を参照してください。

注意： Oracle Reports 10g リリース 2 (10.1.2) のスケラビリティの向上により、Reports Server の安定性も、レポート・リクエストが正常に実行されるように改善されています。

問題 1

マスター・ジョブが終了前にハングします。

解決策 1

マスター・ジョブがなんらかの理由でハングすると、マスター・ジョブの終了を待機している重複ジョブもハングします。1 つのマスター・ジョブに対して複数の重複ジョブが待機している場合は、問題に対処しないかぎり、そのマスター・ジョブがハングすると、サーバーが不安定になる場合があります。

`server_name.conf` ファイルの `engine` 要素にある `engineResponseTimeOut` 属性を確認してください (第 3.2.1.4 項「`engine`」を参照)。サーバーが不安定にならないよう、この属性を慎重に設定します。これにより、Reports Server はこのようなハング状況を自動的に検出し、回復できるようになります。また、`showjobs` コマンドを使用してハングしたジョブを終了し、Reports Server に他のリクエストを処理させることもできます。`showjobs` コマンドの詳細は、第 2.5 項「Reports Servlet および Reports Server が実行されていることの確認」および第 A.3.99 項「SHOWJOBS」を参照してください。

たとえば、一連のレポートを実行する必要があるとします。最も大きいレポートは、実行に最大 5 分かかります。このような場合は、`engineResponseTimeOut` を 5 分に設定することができます。

注意：

エンジンでジョブが実行されると、サーバーは最新のステータスに更新されます。たとえば、ページ 1、2 というようにフォーマットが行われます。5 分経過しても Reports Server がエンジンから更新情報を受信しない場合は、エンジンがハングしていると見なされ、Reports Server はエンジンを停止します。

実行に 1 分から 1 時間かかる複雑なレポートがある場合は、レポートの実行中に、コマンドラインで `ENGINERESPONSETIMEOUT` を指定します (第 A.3.33 項「`ENGINERESPONSETIMEOUT`」を参照)。

スケジュールされているジョブやバッチ・ジョブのほかに対話型のジョブがある場合は、前者と後者で 1 台ずつ個別のサーバーを起動することをお勧めします。パフォーマンスと安定性の理由から、対話型のジョブとバッチ・ジョブまたはスケジュールされたジョブの両方に同じサーバーを使用しないでください。

`engineResponseTimeOut` 属性またはコマンドラインの `ENGINERESPONSETIMEOUT` キーワードを慎重に設定したにもかかわらず、サーバーが不安定になったりクラッシュしたりする場合は、次の手順を実行して、オラクル社カスタマ・サポート・センターに報告してください。

1. サーバーのトレースとログを有効にします (第 20.1.2 項「レポート・トレース」を参照)。トレースを有効にできない場合、`server_name.conf` ファイルで `log` 要素の `option` 属性を `failedJobs` に設定することでログのみを有効にします (第 3.2.1.11 項「`log`」を参照)。ログを有効にすると、失敗したジョブ・レポートを `reports.log` ファイルで確認できます。失敗したレポートまたはエンジンのハングを引き起こしたレポートを特定します。
2. `server_name.conf` ファイルの `engine` 要素を編集し、`diagnosis` プロパティを追加してエンジンの診断ログを有効にします (第 3.2.1.4 項「`engine`」の「プロパティ」を参照)。その後、手順 1 で特定したレポートを実行し、ハングを再現します。
3. 次の情報を添えて、オラクル社カスタマ・サポート・センターにハングの問題を報告します。
 - `server_name.conf` ファイル
 - `reports.log` ファイル
 - ハングが再現されたときのエンジンの診断結果の出力
 - レポート定義ファイル (オラクル社カスタマ・サポート・センターにおける問題再現用)

問題 2

レポート・リクエストの実行中に、次のような現象で Reports Server がレスポンスを停止したかクラッシュします。

- ブラウザを通じてジョブが送信されると、ブラウザがハングします (レスポンスがない)。
- エンジンがアイドル状態であるのに、ジョブがエンジンに送信されません。
- Web コマンドが機能せず、しばらくしてブラウザがタイムアウトになります。
- スケジュールされているジョブが実行されません。

解決策 2

問題から回復するために、Reports Server を再起動します。それでも問題が解決しない場合は、次の情報を添えて、オラクル社カスタマ・サポート・センターに報告してください。

- `server_name.conf` ファイル
- ハング時の Reports Server のおおよその負荷率
- Reports Server のスレッド・ダンプ。これは、次の手順で出力できます。
 - Solaris の場合は、Reports Server がハングしたときに、`kill -3 server_pid` コマンドを使用します。このコマンドは、スレッド情報をコンソール出力に書き込みます。スレッド情報とエラー・ストリームの出力先をコンソールからファイルに変更するには、`ORACLE_HOME/bin` ディレクトリの `rserver.sh` ファイルを修正します。次に例を示します。

```
exec $ORACLE_HOME/bin/rserver "$@" > threaddump.txt 2>&1
```

注意： これは、UNIX k シェルの場合の例を示しています。他のシェルを使用している場合は、コードが多少異なることがあります。

インプロセス・サーバーを使用している場合は、OC4J_BI_Forms プロセス ID に `kill -3` コマンドを使用します。スレッド・ダンプは、`ORACLE_HOME/opmn/logs` ディレクトリの OC4J ログ (`OC4J~OC4J_BI_Forms~default_island~1`) ファイルに出力されます。

- Windows の場合は、`kill -3` コマンドは機能しません。かわりに、コマンド・プロンプトで表 D-1 に示すコマンドを入力し、Reports Server を起動します。問題が再現されたら、コマンド・プロンプトのウィンドウに移動し、[Ctrl] を押しながら [Break] キーを押してスレッド・ダンプを出力します。

表 D-1 Windows でスレッド・ダンプを出力するためのコマンド

Reports Server のバージョン	コマンド
10.1.2.0.2	<code>ORACLE_HOME/jdk/bin/java -Xmx256M -classpath %REPORTS_CLASSPATH% oracle.reports.server.RWServer oracle_home=ORACLE_HOME server=server_name showui=yes no batch=yes no</code>
9.0.4	<code>ORACLE_HOME/jdk/bin/java -Xbootclasspath/p:\$OH/vbroker4/lib/vbjboot.jar -Xmx256M -classpath %REPORTS_CLASSPATH% oracle.reports.server.RWServer oracle_home=ORACLE_HOME server=server_name showui=yes no batch=yes no</code>
9.0.2	<code>ORACLE_HOME/jdk/bin/java -Xmx256M -classpath %REPORTS_CLASSPATH% oracle.reports.server.RWServer oracle_home=ORACLE_HOME server=server_name showui=yes no batch=yes no</code>

問題 3

インプロセス・サーバーでレポートを実行しようとする、インプロセス・サーバーが起動に失敗し、ブラウザに次のメッセージが表示されます。

```
REP-52266: The in-process Reports Server failed to start.
```

スタンドアロン・サーバーを起動すると、すぐに停止します。

解決策 3

トレースを有効にして (第 20.1.2 項「レポート・トレース」を参照)、インプロセス・サーバーを起動してください。デフォルトでは、Reports Server の `rwserver.trc` ファイルに実際の問題の原因が記録されます。

- **Reports Server が、いずれかのプラグابل・データソースまたはプラグابل・デスティネーションの初期化に失敗しました。** プラグابل・データソース (PDS) またはプラグابل・デスティネーションの構成を修正して、Reports Server を再起動してください。PDS の詳細は、Oracle Reports のオンライン・ヘルプの「プラグابل・データソース」の項を参照してください。JDBC PDS の詳細は、第 9 章「JDBC PDS の構成と使用」を参照してください。
- **エンジンが起動に失敗しました。** `ORACLE_HOME/reports/logs/server_name` ディレクトリの `rwEng-enginenummer.trc` ファイルを確認してください。このファイルには、次の行が含まれている必要があります。

```
Debug 50103 (EngineImpl:EngineImpl): CInitEngine returns 0
Info 55003 (RWEEngine:init): Register this engine to Oracle Reports Server server_
name
```

`rwEng-enginenummer.trc` ファイルにこれらの行が存在しない場合は、エンジンが起動に失敗したことを示します。

このファイルの `CinitEngine` 戻り値が負の値の場合は、Reports Engine の初期化時にエラーが発生したことを示します。`CinitEngine` 戻り値がゼロ以外の場合は、PATH 環境変数 (Windows を使用している場合) または `LD_LIBRARY_PATH` 環境変数 (Solaris を使用している場合) を確認してください。インプロセス・サーバーでは、PATH および `LD_LIBRARY_PATH` の値は、`ORACLE_HOME/j2ee/OC4J_BI_Forms/config` ディレクトリの `oc4j.properties` ファイルから取得されます。

問題 4

レポート・リクエストを実行すると Reports Engine がクラッシュまたはハングします。

解決策 4

ケース 1 : Reports Server で毎日大量のレポートを実行し、レポートの印刷や Web への公開も行っている場合。この場合では、ブラウザがレスポンスを待機し、最終的にタイムアウトになる場合があります。ジョブ・キューを確認するための Web コマンドも、機能しないことがあります。

トレースを有効にして (第 20.1.2 項「レポート・トレース」を参照) この問題が発生したら、「解決策 2」で説明した `kill -3 server_pid` コマンドを Solaris で実行して、スレッド・ダンブを出力します。次に示すコード行は、`kill -3 server_pid` コマンドの実行結果です。これらの行は、Reports Server が、ネットワーク・ドライブにレポートを書き込もうとしたときにハングしたことを示しています。

```
"RequestProcessor[7]" daemon prio=5 tid=0x1835f210 nid=0x181c waiting on condition
[224cf000..224cfd88]
  at java.io.FileOutputStream.write (Native Code)
  at oracle.reports.utility.copyFile (Utility.java:424)
  at oracle.reports.server.DesFile.sendFile (DesFile.java:74)
  at oracle.reports.server.Destination.send (Destination.java:484)
  at oracle.reports.server.JobObject.distribute (JobObject.java:1582)
  at oracle.reports.server.JobManager.updateJobStatus (JobManager.java:2231)
  at oracle.reports.server.EngineCommImpl.updateEngineJobStatus (
    EngineCommImpl.java:134)
  at oracle.reports.server._EngineCommImplBase._invoke (
    _EngineCommImplBase.java:94)
  at com.sun.corba.se.internal.corba.ServerDelegate.dispatch
    (ServerDelegate.java:353)
  at com.sun.corba.se.internal.iiop.ORB.process (ORB.java:280)
  at com.sun.corba.se.internal.iiop.RequestProcessor.process
    (RequestProcessor.java:81)
  at com.sun.corba.se.internal.orbutil.ThreadPool$PooledThread.run
    (ThreadPool.java:106)
```

このシナリオのトレース・ファイルは次のとおりです。

```
[2005/5/31 6:26:47:321] Info 50132 (JobObject:reset): jobid = 15 Get command line:
server=vin report=c:\%backup%reps%emp.rdf destype=file desformat=html
desname=c:\%test.html userid=scott@ora9i authid=vnhegde
[2005/5/31 6:26:48:92] Debug 50103 (JobManager:firstToRun): job 15 is first to run
[2005/5/31 6:26:48:212] Debug 50103 (ConnectionImpl:runJob): Job queue for jobid =
15 is 0
[2005/5/31 6:26:48:212] Debug 50103 (ConnectionImpl:runJob): jobid = 15 is in
current queue
[2005/5/31 6:26:48:212] Debug 50103 (ConnectionImpl:runJob): Calling
findDuplicatedJob for jobid = 15
[2005/5/31 6:26:48:212] Debug 50103 (JobManager:findDuplicatedJob): Found no
duplicated job for job 15
[2005/5/31 6:26:48:212] Debug 50103 (ConnectionImpl:runJob): No Duplicate jobs for
jobid = 15
[2005/5/31 6:26:48:212] Debug 50103 (ConnectionImpl:runJob): Job 15 is Enqueued
[2005/5/31 6:26:48:212] Debug 50103 (JobManager:firstToRun): job 15 is first to
run
[2005/5/31 6:26:48:212] Debug 50103 (JobManager.runJobLocal): Trying to get engine
for Job 15
[2005/5/31 6:26:48:212] Debug 50103 (EngineManager:getIdleEngine): Target max
engines = 1
[2005/5/31 6:26:48:222] Debug 50103 (EngineManager:getIdleEngine): rwEng-0 is used
= true
[2005/5/31 6:26:48:222] Debug 50103 (EngineManager:getIdleEngine): rwEng-0 state
is 1
[2005/5/31 6:26:48:222] State 56004 (EngineInfo:setState): Engine rwEng-0 state
is: Reserved
[2005/5/31 6:26:48:222] Debug 50103 (JobManager.runJobLocal): Job 15 got Engine
rwEng-0
[2005/5/31 6:26:48:222] Debug 50103 (JobManager:runJobInEngine): Job 15 calling
setCommand on engine rwEng-0
[2005/5/31 6:26:48:222] Debug 50103 (EngineManager:updateEngineState): Engine
rwEng-0 status is 3
[2005/5/31 6:26:48:222] State 56004 (EngineInfo:setState): Engine rwEng-0 state
is: Running
[2005/5/31 6:26:48:222] Debug 50103 (EngineManager:updateEngineState): Engine
rwEng-0 status is 5
[2005/5/31 6:26:48:222] State 56004 (EngineInfo:setState): Engine rwEng-0 state
is: Idle
[2005/5/31 6:26:48:232] Debug 50103 (JobManager:runJobInEngine): Send job 15 to
engine rwEng-0
[2005/5/31 6:26:48:232] Debug 50103 (EngineManager:updateEngineState): Engine
rwEng-0 status is 3
[2005/5/31 6:26:48:232] State 56004 (EngineInfo:setState): Engine rwEng-0 state
is: Running
[2005/5/31 6:26:48:482] State 56016 (JobManager:updateJobStatus): Job 15 status
is: Running the report Initializing report
[2005/5/31 6:26:48:482] Debug 50103 (JobManager:updateJobStatus): Finished
updating job: 15
[2005/5/31 6:26:50:856] State 56016 (JobManager:updateJobStatus): Job 15 status
is: Running the report Formatting page 1
[2005/5/31 6:26:50:856] Debug 50103 (JobManager:updateJobStatus): Finished
updating job: 15
[2005/5/31 6:26:52:468] Debug 50103 (RWCACHEItem:addFile): add file
'test33347112.htm' for job 15
[2005/5/31 6:26:52:468] Debug 50103 (RWCACHE:updateCurrentCapacity): Current cache
capacity is 197239
```

このトレース・ファイルでは、次のイベントに注目してください。

- ID 15 のジョブが 6:26:47:321 に送信されています。
- 重複ジョブが 6:26:48:212 にチェックされています。
- rwEng-0 が 6:26:48:222 に取得されています。
- エンジンが 6:26:48:222 で実行を開始しています。
- 最初のページが 6:26:50:856 でフォーマットされています。

この後、このジョブに対する更新はありません。Finished successfully 行も存在しません。これは、ジョブに問題があることを示しています。

次の例は、正常に終了したジョブのトレース・ファイルです。

```
[2005/5/31 6:25:57:198] Info 50132 (JobObject:reset): jobid = 14 Get command line:
server=vin report=c:\%backup%reps%emp.rdf destype=file desformat=html
desname=c:\%test.html userid=scott@ora9i authid=vphegde
[2005/5/31 6:25:58:80] Debug 50103 (ConnectionImpl:runJob): Job queue for jobid =
  14 is 0
[2005/5/31 6:25:58:90] Debug 50103 (ConnectionImpl:runJob): jobid = 14 is in
  current queue
[2005/5/31 6:25:58:90] Debug 50103 (ConnectionImpl:runJob): Calling
  findDuplicatedJob for jobid = 14
[2005/5/31 6:25:58:90] Debug 50103 (JobManager:findDuplicatedJob): Found no
  duplicated job for job 14
[2005/5/31 6:25:58:90] Debug 50103 (ConnectionImpl:runJob): No Duplicate jobs for
  jobid = 14
[2005/5/31 6:25:58:90] Debug 50103 (ConnectionImpl:runJob): Job 14 is Enqueued
[2005/5/31 6:25:58:90] Debug 50103 (JobManager:firstToRun): job 14 is first to run
[2005/5/31 6:25:58:90] Debug 50103 (JobManager.runJobLocal): Trying to get engine
  for Job 14
[2005/5/31 6:25:58:90] Debug 50103 (EngineManager:getIdleEngine): Target max
  engines = 1
[2005/5/31 6:25:58:90] Debug 50103 (EngineManager:getIdleEngine): rwEng-0 is used
  = true
[2005/5/31 6:25:58:90] Debug 50103 (EngineManager:getIdleEngine): rwEng-0 state is
  1
[2005/5/31 6:25:58:90] State 56004 (EngineInfo:setState): Engine rwEng-0 state is:
  Reserved
[2005/5/31 6:25:58:90] Debug 50103 (JobManager.runJobLocal): Job 14 got Engine
  rwEng-0
[2005/5/31 6:25:58:90] Debug 50103 (JobManager:runJobInEngine): Job 14 calling
  setCommand on engine rwEng-0
[2005/5/31 6:25:58:100] Debug 50103 (EngineManager:updateEngineState): Engine
  rwEng-0 status is 3
[2005/5/31 6:25:58:100] State 56004 (EngineInfo:setState): Engine rwEng-0 state
  is: Running
[2005/5/31 6:25:58:100] Debug 50103 (EngineManager:updateEngineState): Engine
  rwEng-0 status is 5
[2005/5/31 6:25:58:100] State 56004 (EngineInfo:setState): Engine rwEng-0 state
  is: Idle
[2005/5/31 6:25:58:100] Debug 50103 (JobManager:runJobInEngine): Send job 14 to
  engine rwEng-0
[2005/5/31 6:25:58:110] Debug 50103 (EngineManager:updateEngineState): Engine
  rwEng-0 status is 3
[2005/5/31 6:25:58:110] State 56004 (EngineInfo:setState): Engine rwEng-0 state
  is: Running
[2005/5/31 6:25:58:350] State 56016 (JobManager:updateJobStatus): Job 14 status
  is: Running the report Initializing report
[2005/5/31 6:25:58:350] Debug 50103 (JobManager:updateJobStatus): Finished
  updating job: 14
[2005/5/31 6:26:0:663] State 56016 (JobManager:updateJobStatus): Job 14 status is:
  Running the report Formatting page 1
```

```
[2005/5/31 6:26:0:663] Debug 50103 (JobManager:updateJobStatus): Finished updating
job: 14
[2005/5/31 6:26:2:256] Debug 50103 (RWCACHEItem:addFile): add file
'test54106877.htm' for job 14
[2005/5/31 6:26:2:256] Debug 50103 (RWCACHE:updateCurrentCapacity): Current cache
capacity is 182329
[2005/5/31 6:26:2:286] State 56016 (JobManager:updateJobStatus): Job 14 status is:
Finished successfully
[2005/5/31 6:26:3:7] Debug 50103 (JobManager:notifyWaitingJobs): Master job 14
notify its duplicated jobs.
[2005/5/31 6:26:3:7] Debug 50103 (JobManager:updateJobStatus): Finished updating
job: 14
[2005/5/31 6:26:3:7] Debug 50103 (EngineManager:updateEngineState): Engine rwEng-0
status is 1
[2005/5/31 6:26:3:7] State 56004 (EngineInfo:setState): Engine rwEng-0 state is:
Ready
[2005/5/31 6:26:3:57] Info 56013 (ConnectionManager:release): Connection 1 is
released
```

このトレース・ファイルでは、ページ1のフォーマットの後に続く次のイベントに注目してください。

- ジョブが 6:26:2:286 で正常終了しています。
- 重複ジョブが 6:26:3:7 で通知されています。
- 接続が 6:26:3:57 で切断されています。

これらの行は、最初の例には存在しませんでした。すべてのジョブの Reports Server トレース・ファイルには、これらの行が存在する必要があります。イベントの欠落や強制終了は、ジョブが正常に終了していないことを示しており、ハングの原因と考えられます。

ケース 2: 次のエラーが表示される場合。

```
REP-56048: Engine rwEng-0 crashed, job Id: 17
```

このシナリオでは、Reports Server とエンジンのトレース・ファイルを確認します。通常のクラッシュでは、Reports Server トレース・ファイルは次のようになります。

```
[2005/6/1 3:38:35:156] Exception 50125 (org.omg.CORBA.COMM_FAILURE: vmcid: SUN
minor code: 208 completed: Maybe
at com.sun.corba.se.internal.iiop.IIOPConnection.purge_
calls (IIOPConnection.java:438)
at com.sun.corba.se.internal.iiop.ReaderThread.run (ReaderThread.java:70)
): Internal error org.omg.CORBA.COMM_FAILURE: vmcid: SUN minor code: 208
completed: Maybe
[2005/6/1 3:38:35:156] Info 56029 (EngineManager:shutdownEngine): Shutting down
engine rwEng-0
[2005/6/1 3:38:36:137] Exception 50125 (org.omg.CORBA.COMM_FAILURE: vmcid: SUN
minor code: 201 completed: No
at com.sun.corba.se.internal.iiop.ConnectionTable.getConnection
(ConnectionTable.java:148)
at com.sun.corba.se.internal.iiop.ConnectionTable.getConnection
(ConnectionTable.java:65)
at com.sun.corba.se.internal.iiop.GIOPImpl.getConnection (GIOPImpl.java:67)
at com.sun.corba.se.internal.corba.ClientDelegate.createRequest
(ClientDelegate.java:652)
at com.sun.corba.se.internal.corba.ClientDelegate.createRequest
(ClientDelegate.java:594)
at com.sun.corba.se.internal.corba.ClientDelegate.request
(ClientDelegate.java:886)
at org.omg.CORBA.portable.ObjectImpl._request (ObjectImpl.java:431)
at oracle.reports.engine._EngineClassStub.shutdown (_EngineClassStub.java:173)
at oracle.reports.server.EngineManager.shutdownEngine (EngineManager.java:1354)
at oracle.reports.server.JobManager.runJobInEngine (JobManager.java:974)
at oracle.reports.server.JobManager.runJobLocal (JobManager.java:1779)
```



```

at oracle.reports.server.JobManager.dispatch(JobManager.java:1045)
at oracle.reports.server.ConnectionImpl.runJob(ConnectionImpl.java:1274)
at oracle.reports.server._ConnectionImplBase._invoke
  (_ConnectionImplBase.java:401)
at com.sun.corba.se.internal.corba.ServerDelegate.dispatch
  (ServerDelegate.java:353)
at com.sun.corba.se.internal.iiop.ORB.process
  (ORB.java:280)
at com.sun.corba.se.internal.iiop.RequestProcessor.process
  (RequestProcessor.java:81)
at com.sun.corba.se.internal.orbutil.ThreadPool$PooledThread.run
  (ThreadPool.java:106)
): Internal error org.omg.CORBA.COMM_FAILURE: vmcid: SUN minor code: 201
completed: No
[2005/6/1 3:38:36:147] State 56004 (EngineInfo:setState): Engine rwEng-0 state is:
Shutdown
[2005/6/1 3:38:36:147] Info 56047 (EngineManager:remove): Reports Server shut down
engine rwEng-0
[2005/6/1 3:38:36:147] State 56016 (JobManager:updateJobStatus): Job 17 status is:
Terminated with error:
REP-56048: Engine rwEng-0 crashed, job Id: 17
[2005/6/1 3:38:36:157] Debug 50103 (JobManager:notifyWaitingJobs): Master job 17
notify its duplicated jobs.
[2005/6/1 3:38:36:157] Debug 50103 (JobManager:updateJobStatus): Finished updating
job: 17
[2005/6/1 3:38:36:157] Exception 56048 (): Engine rwEng-0 crashed, job Id: 17
oracle.reports.RWException: IDL:oracle/reports/RWException:1.0
at oracle.reports.server.JobManager.runJobInEngine(JobManager.java:1009)
at oracle.reports.server.JobManager.runJobLocal(JobManager.java:1779)
at oracle.reports.server.JobManager.dispatch(JobManager.java:1045)
at oracle.reports.server.ConnectionImpl.runJob(ConnectionImpl.java:1274)
at oracle.reports.server._ConnectionImplBase._invoke
  (_ConnectionImplBase.java:401)
at com.sun.corba.se.internal.corba.ServerDelegate.dispatch
  (ServerDelegate.java:353)
at com.sun.corba.se.internal.iiop.ORB.process
  (ORB.java:280)
at com.sun.corba.se.internal.iiop.RequestProcessor.process
  (RequestProcessor.java:81)
at com.sun.corba.se.internal.orbutil.ThreadPool$PooledThread.run
  (ThreadPool.java:106)

```

エンジンのトレース・ファイルでは、クラッシュのトレースを示す最後の数行が次のようになります。

```

[2005/6/1 3:38:34:575] (rwfdt:rwfdtprint) Distributing the report
[2005/6/1 3:38:34:585] (rwfdt:rwfdtpredo) running
[2005/6/1 3:38:34:585] (rwfdt:rwfdtpredo) no preformat of pages requested, quit
[2005/6/1 3:38:34:585] (rwfdt:rwfdtni_NextInstance) running
[2005/6/1 3:38:34:595] (rwfdt:rwfdtni_NextInstance) quit
[2005/6/1 3:38:34:595] (rwfdt:rwfdtgcf_GenCachefile) running
[2005/6/1 3:38:34:615] (rwfdt:rwfdtgcf_GenCachefile) Cache file is
D:¥orawin¥reports¥cache¥03564661.htm
[2005/6/1 3:38:34:615] (rwfdt:rwfdtgcf_GenCachefile) quit
[2005/6/1 3:38:34:615] (rwfdt:rwfdtprint) caching output from backend drivers
[2005/6/1 3:38:34:755] (C Engine)

```

注意： エンジンのトレース・ファイルは、エンジンがクラッシュすると中断します。

処置：エンジンのクラッシュを引き起こしているレポートを特定します。これには、ジョブ ID を特定します。前述の例の場合、エンジンは jobid 17 の実行中にクラッシュしています。サーバーのトレース・ファイルで、jobid = 17 Get command line 文字列を検索します。この行には、レポート名も含む完全なコマンド行が含まれています。トレースとエンジンの診断を有効にします。問題のあるレポートを何度か実行し、クラッシュを再現します。クラッシュが再現されたら、トレース・ファイルと診断の出力をオラクル社カスタマ・サポート・センターに送り、分析を依頼します。

D.1.2 Reports Server のアクティビティによりエラー REP-50125 が生成される

REP-50125 は、Reports Server が関係する様々な状況で生成される一般的なエラー・メッセージです。

REP-50125: Caught exception: {0}

原因：Oracle Reports で内部例外が発生しました。

処置：オラクル社カスタマ・サポート・センターにサポートを依頼してください。

問題 1

REP-50125 のヘルプ・トピックの原因および処置に関する項には、問題を効率的に特定し診断するための十分な情報が記載されていません。

解決策 1

Oracle Reports 10g リリース 2 (10.1.2) では、次の新しいエラー・メッセージを特定のシナリオに適用し、焦点を絞ったトラブルシューティング・サポート情報がヘルプ・トピックの原因および処置に関する項で用意されています。

- REP-56126: Failed to parse server config file {0}
原因：サーバー構成ファイルの解析が失敗しました。XML 構文に誤りがあります。
処置：サーバー構成ファイルを修正して、サーバーを起動してください。
- REP-56127: Failed to decrypt <{0}> element
原因：要素に対する復号化コールが失敗しました。
処置：encrypted 属性が要素に正しく設定されていることを確認してください。
- REP-56128: Failed to initialize {0} destination. Nested Exception: {1}
原因：宛先の初期化が失敗しました。
処置：宛先の構成を確認し、修正してください。

問題 2

Reports Server の起動時に REP-50125 が表示されます。

解決策 2

Oracle MetaLink (<http://metalink.oracle.com>) で、Reports Server の起動時の問題に関するトラブルシューティングのノート 289748.1 を参照してください。

問題 3

レポート・リクエストを実行すると REP-50125 が表示されます。

解決策 3

Oracle MetaLink (<http://metalink.oracle.com>) で、Reports Server に発行されたレポート・リクエストの失敗に関するトラブルシューティングのノート 290827.1 を参照してください。

問題 4

SLES-8/UnitedLinux 1.0 で Reports Server を起動すると、セグメンテーション違反とともに REP-50125 が表示されます。

解決策 4

Oracle Reports 10g リリース 2 (10.1.2) では、SLES8 と SLES9 がサポートされています。しかし、Oracle Reports 10g リリース 2 (10.1.2) では UnitedLinux 1.0 はサポートされていないため、このプラットフォームを使用してレポート・リクエストを実行することはできません。

問題 5

Linux で openmotif を使用してレポート・リクエストを実行すると、REP-50125 が表示されます。

解決策 5

Linux 上の Oracle Reports 6i、9i および 10g では、openmotif 2.1.30 のみがサポートされます。

D.1.3 Reports Servlet を使用して長期レポートを実行すると失敗する

Reports Servlet (rwservlet) から送信された長期レポート・リクエストは、AS コンポーネントによっては、失敗したりエンジンのクラッシュやハングを引き起こす場合があります。

問題

rwservlet を使用して長期にわたって実行しているレポートが終了しません。

解決策

次のことを確認してください。

- Web ブラウザから同期式にレポートを実行している場合は、この障害が HTTP サーバーのタイムアウトによるものかどうかを確認します。同じジョブを非同期で送信してみます。正常に終了する場合は、アプリケーション・サーバー構成の HTTP サーバーのタイムアウトを変更するか、長期レポートを非同期で実行すること（この方法をお勧めします）を検討してください。Reports Server の通知機能を利用すると、ジョブが終了したことをユーザーに通知できます。
- Enterprise Manager で Reports Server の統計を調べ、サーバーの全般的なレスポンス時間が大幅に変化していないかどうかを確認します。
- データベース・サーバーが正常にレスポンスしているかどうかを確認します。長期にわたって実行されるレポートでは特に、データベースの負荷がパフォーマンスに重大な影響を及ぼす場合があります。

D.1.4 プラットフォームによってフォントの表示が異なる

複数のプラットフォームにレポートを配布すると、フォントの問題が生じる場合があります。

問題

複数のプラットフォームにレポートを配布した場合に、プラットフォーム間のフォントの表示およびマッピングが一致しません。

解決策

Oracle Reports 10g リリース 2 (10.1.2) では、この問題について、新しいマニュアルの次の各章で取り上げています。

- [第 4 章「Oracle Reports でのフォントの管理」](#)（特に、Oracle Reports におけるフォントの処理メカニズムの詳細は、[第 4.1 項「フォントの使用」](#)を参照）。
- [第 6 章「Oracle Reports での PDF の使用」](#)（特に、[第 6.1.2 項「フォント関連の機能」](#)および [第 6.2 項「プラットフォーム間の配布時における PDF フォントの問題への対応」](#)を参照）。

- 第7章「クロス・プラットフォームの移植問題の解決」(特に、第7.1.1項「プラットフォームによって異なるフォント」および第7.1.2項「フォントに関連する問題の解決」を参照)。

D.1.5 UNIX プラットフォームでレポートを実行すると REP-56048 が生成される

REP-56048 は、UNIX で `rwsvrlet`、`rwclient`、Oracle Forms などを使用してレポートを実行したときに発行される、一般的なエラー・メッセージです。Reports Server は、レポートを実行する Report Engine にジョブを渡します。その Report Engine がクラッシュすると、このエラーが発生します。

REP-56048: Engine {0} crashed

原因 : Reports Server で、指定したエンジンのクラッシュが検出されました。

処置 : Reports Server で別のエンジンを再起動します。エンジンがクラッシュするテスト・ケースとともに、問題をオラクル社カスタマ・サポート・センターに報告してください。

後述の解決策および第 D.1.1 項「レポート・リクエストのハング」の「解決策 4」を参照し、問題に対処します。それでも REP-56048 が発生する場合は、次の手順を実行して、問題をオラクル社カスタマ・サポート・センターに報告してください。

1. サーバーのトレースとログを有効にします (第 20.1.2 項「レポート・トレース」を参照)。トレースを有効にできない場合、`server_name.conf` ファイルで log 要素の `option` 属性を `failedJobs` に設定することでログのみを有効にします (第 3.2.1.11 項「log」を参照)。ログを有効にすると、失敗したジョブ・レポートを `reports.log` ファイルで確認できます。失敗したレポートまたはエンジンのクラッシュを引き起こしたレポートを特定します。
2. `server_name.conf` ファイルの `engine` 要素を編集し、`diagnosis` プロパティを追加してエンジンの診断ログを有効にします (第 3.2.1.4 項「engine」の「プロパティ」を参照)。その後、手順 1 で特定したレポートを実行し、クラッシュを再現します。
3. 次の情報を添えて、オラクル社カスタマ・サポート・センターにクラッシュの問題を報告します。
 - `server_name.conf` ファイル
 - `reports.log` ファイル
 - クラッシュが再現されたときのエンジンの診断結果の出力
 - レポート定義ファイル (オラクル社カスタマ・サポート・センターにおける問題再現用)

問題 1

UNIX プラットフォームでレポートを実行したときに、`NLS_LANG` に `WE8ISO8859P1` または `IW8ISO8859P8` 以外のキャラクタ・セットが定義されている場合、REP-56048 が表示されません。

解決策 1

Oracle のキャラクタ・セット名および XLFD の `CHARSET_REGISTRY` と `CHARSET_ENCODING` とのマッピングに使用される `Tk2Motif.rgb` 内のエントリを修正します (最後の 2 つのフィールド。 `iso8859-1` など)。詳細は、次を参照してください。

- 第 4.3 項「フォント構成ファイル」
- 第 13.3.6 項「UNIX で WE8MSWIN1252 キャラクタ・セットを使用したレポートの実行」
- 第 18.7 項「グローバリゼーションに関する問題のトラブルシューティング」
 - グローバリゼーション・サポートの環境変数の設定
 - HP-UX の日本語環境での Oracle Reports の実行

また、Report Engine でレポートを実行する前に、Reports Runtime (`rwrun`) を使用してこのレポートを実行し、環境設定を確認することもできます。

問題 2

UNIX プラットフォームでレポートを実行したときに、DISPLAY 環境変数が設定されていない場合、REP-56048 が表示されます。

解決策 2

Oracle Reports 10g (9.0.4) では、REPORTS_DEFAULT_DISPLAY 環境変数により、DISPLAY 環境変数への依存性が解消されます。デフォルトでは REPORTS_DEFAULT_DISPLAY=YES です。REPORTS_DEFAULT_DISPLAY が NO に設定されていないことを確認してください。

注意： この機能は IBM JDK 1.4 (ヘッドレス・オプションはサポートしていません) においてサポートされていないので、IBM AIX プラットフォームでは使用できません。AIX では、REPORTS_DEFAULT_DISPLAY=NO に設定し、DISPLAY をアクティブな X-Window 表示画面に設定してください。

詳細は、[第 B.1.39 項「REPORTS_DEFAULT_DISPLAY」](#) を参照してください。

問題 3

巨大なレポートを実行しているときにトレースを有効にすると、REP-56048 が表示されます。

解決策 3

トレースを有効にした場合、膨大な出力のレポートを実行したときに、Reports Engine がクラッシュすることがあります。これは、トレース・ファイルのサイズの問題と、トレースの作成に使用できるディスク領域、メモリーまたはプロセッサ処理能力が不十分であることが原因と考えられます。このエラーに対処するには、診断情報を使用してレポートの問題をトラブルシューティングする必要がある場合に、エンジンの診断ログのみを有効にします。これには、server_name.conf ファイルの engine 要素を編集して、diagnosis プロパティを追加します ([第 3.2.1.4 項「engine」の「プロパティ」](#) を参照)。server_name.conf 構成ファイルの trace 要素にある traceModule 属性を使用して、生成されるトレース・ファイルを制限することもできます。

traceModule 属性と他のトレース・オプションの詳細は、[第 3.2.1.13 項「trace」](#) を参照してください。トレースの詳細は、[第 20.1.2 項「レポート・トレース」](#) を参照してください。

問題 4

UNIX で DISTRIBUTE=YES に設定すると REP-56048 が表示されます。

解決策 4

配布ファイルに指定されている宛先の 1 つがプリンタの場合、PRINTER 環境変数が有効なプリンタに設定されていないと、UNIX で配布エラーが発生します。次の環境変数を設定してください。

```
PRINTER=printer_name; export PRINTER
TK_PRINTER=printer_name; export TK_PRINTER
TK_PRINT_STATUS="echo %n is valid"; export TK_PRINT_STATUS
TK_PRINT=echo; export TK_PRINT
```

\$ORACLE_HOME/guicommon/tk/admin/uiprint.txt に、次の行を追加します。

```
printer_name:PostScript:1:test:default.ppd:
```

レポートの配布の詳細は、[第 15 章「拡張配布の作成」](#) を参照してください。

問題 5

UNIX でレポートを印刷すると REP-56048 が表示されます。

解決策 5

Oracle Reports は、UNIX での印刷にシェル・スクリプト `rw1pr.sh` を使用します。このファイルは直接変更することはできません。オラクル社カスタマ・サポート・センターにサポートを依頼してください。

UNIX での印刷の詳細は、第 5 章「UNIX での Oracle Reports による印刷」を参照してください。

問題 6

UNIX でグラフィックが含まれるレポートを実行すると REP-56048 が表示されます。

解決策 6

このエラーは、Oracle Reports が、オペレーティング・システムのデフォルト以外の Motif バージョンにリンクされている場合に発生することがあります。リンク先となる正しい Motif のバージョンの詳細は、Oracle Application Server のリリース・ノートの Oracle Reports の章を参照してください。

問題 7

マトリックス・レポートのデリミタ付きレポート出力を生成すると REP-56048 が表示されません。

解決策 7

レポート出力を、デリミタ付きではなく、`DelimitedData` (`DESFORMAT=DELIMITEDDATA`) または `スプレッドシート` (`DESFORMAT=SPREADSHEET`) 出力に生成してください。`DelimitedData` を使用すると、大量のレポートをサポートできますが、Microsoft Excel の出力には、レポート・データ・モデルで定義されているデータのみが表示され、レイアウト情報は表示されません。レポート・レイアウトに定義されているフォーマットを維持したレポート出力を生成するには、Oracle Reports 10g リリース 2 (10.1.2) の新しい出力フォーマットである `DESFORMAT=SPREADSHEET` を使用します。

デリミタ付き出力およびスプレッドシート出力の詳細は、Oracle Reports のオンライン・ヘルプのデリミタ付き出力とスプレッドシート出力に関する項、および『Oracle Reports レポート作成ガイド』の「拡張概念」の章を参照してください。第 A.3.25 項「DESFORMAT」も参照してください。

問題 8

前述の解決策で問題を解決できなかった場合に、Reports Engine からレポートを実行すると、REP-56048 が表示されます。

解決策 8

このエラーは、環境設定に関連しているか、レポート自体の問題によって発生する場合があります。Reports Server で Reports Servlet (`rwervlet`) の `showjobs` ページを調べると、エラーを引き起こしているジョブを特定できます。UNIX マシンを使用している場合は、実行環境にコア・ダンプが作成されます。関連するバグを簡単に検索できるよう、このコア・ダンプからスタック・トレースを抽出します。実行可能ファイルは、`rwrun` ではなく、`java` になることに注意してください。

以前のリリースでは、コア・ファイルからスタック・トレースを抽出するのに、ランタイム実行可能ファイルでデバッグを実行していました。この方法は、`rwrun` 実行可能ファイルのみで問題を再現できる場合は、現在でも有効です。しかし、Reports Server を使用した場合のみクラッシュが発生する場合は、エンジンが Java ラッパーを使用してコールされるため、Java 実行可能ファイルでデバッグを実行する必要があります。これにより、すべての Oracle Reports ライブラリが自動的にロードされます。

次に例を示します。

```
dbx java core
```

この例は、dbx の場合を示しています。スタック・トレースを抽出したら、Oracle MetaLink (<http://metalink.oracle.com>) で、スタックにある最後の数コールを使用して、関連する問題を検索できます。

特定のジョブが問題の原因と思われる場合は、次の手順として、コマンドラインの実行可能ファイル `rwclient` および `rwrn` を使用して、そのレポートを実行します。`rwclient` を使用して実行すると、Web コンポーネントが環境から削除されます。`rwrn` を使用して実行すると、Reports Server をバイパスしてエンジンのみで実行することと同じこととなります。

D.1.6 過負荷状態で OPMN を使用するとインプロセス・サーバーに障害が発生する

OPMN の実装に関連する各種の問題により、インプロセス・サーバーは、過負荷状態の低速コンピュータで実行した場合に、タイムアウトになったり停止したりします。

問題 1

インプロセス・サーバーが低速コンピュータでの起動に失敗すると、それが `OC4J_BI_FORMS` `OC4J` インスタンスの起動シーケンスのトリガーとなります。

現時点では、`pingserver?start=auto` URL が渡されると、インプロセス・サーバーの状態が CORBA コールを通じて取得されます。この CORBA コールは、1 秒でタイムアウトになります。したがって、サーバーを実行しているコンピュータが低速の場合、CORBA コールは失敗する可能性があります。このような障害が発生すると、インプロセス・サーバーは稼働しているにもかかわらず、稼働していないものと見なされるため、起動シーケンスが開始します。

インプロセス・サーバーを再起動すると、ネットワークに同じ名前の別のサーバーが存在するかどうかチェックされます。同じ名前の別のサーバーが存在する場合、インプロセス・サーバーは停止します。その結果、CORBA コールが失敗し、`OC4J` が再起動されます。

解決策 1

Oracle Reports 10g リリース 2 (10.1.2) では、この問題に対応するために、次の変更が行われています。

- CORBA コールは、スタンドアロン・サーバーのみの状態を取得するために実行されます。インプロセス・サーバーの状態のチェックには CORBA コールは使用されません。
- インプロセス・サーバーと同じ名前のスタンドアロン・サーバーが稼働している場合、Reports Servlet (`rwervlet`) はそのスタンドアロン・サーバーとの接続を確立します。
- Reports Servlet は、最初の CORBA コールにかぎり、`start=auto` を使用してインプロセス・サーバーを起動します。後からインプロセス・サーバーが手動で停止された場合、Reports Servlet はインプロセス・サーバーを起動できません。

問題 2

インプロセス・サーバーが終了する前に `ping` がタイムアウトになります。

解決策 2

OPMN は、インプロセス・サーバー (`OC4J_BI_FORMS` 内) からのコールバックを待機する必要がありますが、第 3.9 項「レポート配布の最適化」で説明しているように、`ping` のタイムアウトは、そのコールバックがタイムアウトと見なされるまでの待ち時間を特定するための目安となります。

ping のタイムアウトを構成するには、opmn.xml の次の要素に ping エントリを追加して、マシンの負荷にあった十分なタイムアウト値を指定します。

```
<ias-component id="OC4J">
...
<process-type id="OC4J_BI_Forms" module-id="OC4J">
...
<restart timeout="720" retry="2" />
...
<ping timeout="110" interval="30" />
...
```

また、opmn.xml の次の要素を削除またはコメントアウトして、URL ping をオフにすることもできます。

```
<category id="urlping-parameters">
  <data id="/reports/rwservlet/pingserver?start=auto" value="200" />
</category>
```

その後で OC4J_BI_Forms を再起動します。

Reports Server での OPMN の使用方法の詳細は、[第 2.1.2 項「Oracle Process Manager and Notification Server からの Reports Server の起動、停止および再起動」](#) および [第 3.8 項「Oracle Reports と Oracle Workflow の通信の構成」](#) を参照してください。

D.1.7 右から左に表示する言語のフォントの問題

双方向サポートは、利用対象者の要件に応じて、左から右または右から左の方向にレポート出力を表示できるようにします。右から左に表示する言語にはフォントの問題があるため、レポート出力の生成が完全ではありません。

問題

右揃えのテキストが正しく表示されず、固定幅フォントを必要とする制限が生じます。

解決策

Oracle Reports 10g リリース 2 (10.1.2) では、右から左に表示する言語（ヘブライ語、アラビア語など）にフォント・サブセットを適用して、テキストが正しく右揃えされるように PDF 出力を改善することで、この問題に対応しています。しかし、UNIX プラットフォームでは、右揃えのテキストが正しく表示されない場合があります。

右から左に表示するテキストの問題に対処するには、次の各項を参照してください。

- [第 18.4 項「双方向サポート」](#) では、右から左に表示する言語のレポートを設計する際に使用できるオプションについて説明します。
- [第 6.4 項「双方向 \(BiDi\) PDF ファイルの生成」](#) では、双方向 (BiDi) 言語の PDF ファイルを生成する手順について説明します。
- [第 B.1.28 項「REPORTS_ARABIC_NUMERAL」](#) では、アラビア語の PDF 出力での数値の書式を指定する環境変数の使用方法について説明します。この環境で有効な値は、ARABIC (アラビア数字)、HINDI (ヒンディー数字) または CONTEXT (コンテキストに応じてアラビア数字またはヒンディー数字) です。
- [第 B.1.29 項「REPORTS_BIDI_ALGORITHM」](#) では、双方向 (BiDi) 言語 (アラビア語やヘブライ語など) の双方向レイアウト・アルゴリズムを切り替える環境変数について説明します。
- [第 7.4 項「マルチバイト PDF 出力の生成」](#) では、UNIX で固定幅フォントを対処する方法の例を示します。

D.1.8 RUN_REPORT_OBJECT を使用して Oracle Forms からレポートを実行する際のエラー

Web 上で Oracle Forms から Oracle Reports をコールする際の最も安全な方法は、Oracle Application Server Reports Services を RUN_REPORT_OBJECT とともに使用することです。RUN_REPORT_OBJECT を使用して Oracle Forms から Oracle Reports をコールする方法の詳細は、OTN にあるホワイト・ペーパー『Oracle Application Server 10g Integrating Oracle Reports in Oracle Forms Services Applications』(<http://otn.oracle.com/products/forms/pdf/10g/frm10gsrw10g.pdf>) を参照してください。

『Oracle Application Server Forms Services 利用ガイド』も参照してください。

問題 1

RUN_REPORT_OBJECT で次のエラーが生成されます。

```
FRM-41214: Unable to run report.
```

解決策 1

Web でレポートを配布し、ブラウザ・ウィンドウに出力を表示する場合は、DESTYPE を、SCREEN または PREVIEW ではなく、CACHE に設定します。ブラウザにレポート出力を表示するには、RUN_REPORT_OBJECT ではなく、WEB.SHOW_DOCUMENT を使用します。

問題 2

RUN_REPORT_OBJECT で次のエラーが生成されます。

```
FRM-41213: Unable to connect to the report server server_name.
```

解決策 2

次を確認してください。

- RUN_REPORT_OBJECT コードで参照されている Reports Server が起動していることを確認します。
- Oracle Reports の OC4J インスタンスが起動していることを確認します。
- RUN_REPORT_OBJECT に渡されるパラメータの値にスペースが使用されていないこと、およびパラメータが一重引用符で囲まれていることを確認します。

問題 3

RUN_REPORT_OBJECT で次のエラーが生成されます。

```
REP-503 You did not specify the name of a report.
```

解決策 3

Oracle Forms オブジェクト・ナビゲータのレポート・オブジェクトのプロパティ・インスペクタに、レポート名が指定されていることを確認します。

問題 4

フォームからレポートを実行して、フォームからレポートにパラメータを渡すことができません。

解決策 4

次を確認してください。

- Oracle Forms から渡すパラメータが正しいフォーマットであることを確認します。パラメータの初期値は、Reports Builder のパラメータのプロパティ・インスペクタで指定されます。
- パラメータの初期値が、Reports Builder のパラメータのプロパティ・インスペクタで指定されていることを確認します。

- Before Report トリガーまたはレポート・パラメータ・フォームを使用して、コマンドラインを渡します。

それでも問題が解決しない場合は、レポートのトレースを有効にして（第 20.1.2 項「レポート・トレース」を参照）、問題の原因を絞り込みます。

問題 5

レポート・パラメータ・フォーム（PARAMFORM=YES）を RUN_REPORT_OBJECT とともに使用するとエラーになり、Internal Server Error が発生します。

解決策 5

OTN にあるホワイト・ペーパー『Oracle Forms Services - Using Run_Report_Object() to call Reports with a parameter form』

(<http://otn.oracle.com/products/forms/pdf/10g/frmrepparamform.pdf>) を参照してください。

D.1.9 Microsoft Excel でのレポート出力の表示

Microsoft Excel に表示するためにデリミタ付きのレポート出力を生成することは、一般的な要件であり、様々な方法で実現できます。しかし、どの方法を選択したらよいかわからない場合があります。

問題

要件に最も適したデリミタ付き出力のソリューションがわかりません。

解決策

レポートの定義および出力ディスプレイの要件に応じて、Microsoft Excel 用のデリミタ付きレポート出力を生成する適切な方法を選択します。

- 要件：ペーパー・レイアウト・レポートがあり、Microsoft Excel に出力する必要があるが、レポート・レイアウトをリッチ・フォーマットにする必要はない。

出力ソリューション：デリミタ付きのレポート出力を生成します。

- DESFORMAT=DELIMITED
- DESFORMAT=DELIMITEDDATA（DELIMITED を使用すると、大量のレポートの実行に問題が生じる場合）

- 要件：ペーパー・レポートがあり、レポート・レイアウトをリッチ・フォーマットにすることを含めて、Microsoft Excel に出力する必要がある。

出力ソリューション：スプレッドシート形式のレポート出力を生成します（Oracle Reports 10g リリース 2 (10.1.2) の新機能）。

- DESFORMAT=SPREADSHEET

- 要件：ペーパー・レポートがあり、レポート・レイアウトをリッチ・フォーマットにすることを含めて、Microsoft Excel に出力する必要がある。さらに、レポートを JSP として配布する必要がある。

出力ソリューション：JSP レポートであるため、直接 .xls ファイル（DESTYPE=FILE）に生成することはできませんが、ブラウザに .xls ファイルとして表示される出力を保存することはできます。<rw:include> を使用してこのソリューションを実行するには、『Oracle Reports レポート作成ガイド』にある、第 29 章「スプレッドシート出力のレポートの作成」の章を参照してください。

- 出力要件：JSP ベースの Web レポートがあり、Microsoft Excel に出力する必要がある。
ソリューション：JSP レポートであるため、直接 .xls ファイル (DESTYPE=FILE) に生成することはできませんが、ブラウザに .xls ファイルとして表示される出力を保存することはできます。Excel テンプレートと JSP タグを使用してこのソリューションを実行する方法の詳細は、OTN の Oracle Reports 10g のページで「**Getting Started with Oracle Reports**」をクリックし、「**Generating Excel Output with Oracle Reports**」というタイトルのデモを参照してください。

スプレッドシート出力およびデリミタ付き出力の詳細は、Oracle Reports のオンライン・ヘルプおよび『Oracle Reports レポート作成ガイド』を参照してください。

D.1.10 ユーザー・イグジットが含まれるレポートの UNIX での障害

ユーザー・イグジットは、Oracle Reports の以前のリリースで開発されたレポートに存在する場合があります。

注意： Oracle Reports 10g では、ORA_JAVA ビルトイン・パッケージと Java Importer を使用して Java メソッドをコールできます。これによってレポートでのユーザー・イグジットの必要性が軽減され、さらにオープンでポータブルな配置が可能になります。また、ダイナミック・ライブラリの C 関数を呼び出すための外部関数インタフェースを提供する、ORA_FFI ビルトイン・パッケージを使用することもできます。これらのビルトイン・パッケージが利用できるようになったため、Oracle Reports では、ユーザー・イグジットの使用はお薦めしません。ただし、既存のユーザー・イグジットを引き続き使用できるように、makefile が提供されています。

問題

UNIX でユーザー・イグジットが含まれるレポートを実行すると失敗します。

解決策

UNIX では、Reports Builder (rwbuilder) と Reports Runtime (rwruntime) によってユーザー・イグジット・ライブラリが動的にロードされ、ユーザー・イグジットが含まれるレポートが正常に実行されます。Reports Server (rwservlet) を使用してレポートを実行する場合は、rwservlet.sh に次の環境変数を追加して、ユーザー・イグジット・ライブラリをロードする必要があります。

```
LD_PRELOAD==librw.so:user_exit_library; export LD_PRELOAD
```

D.1.11 インプロセス・サーバーを使用する際の印刷エラーとフォント・エラー

インプロセス・サーバーでは、現在 Windows に直接ログオンしているユーザーのデフォルトのプリンタが認識されません。これは、インプロセス・サーバーを実行しているサービスが Local System としてログオンされているためです。

問題

次のような問題があります。

- デフォルトのプリンタに出力すると、REP_3002 エラー・メッセージが出力されます。たとえば、次のようなコマンドを入力したとします。

```
http://myrepsrvr.us.oracle.com:7777/reports/rwservlet?report=myrep.rdf&destype=printer&desformat=html
```

この場合、次のエラー・メッセージが表示されます。

```
Error:"REP-3002: Error initializing printer. Please make sure a printer is installed."
```

- Oracle6i Graphics (OGD) グラフィックが含まれるレポートを配布すると、Reports Server がレスポンスを停止します。
- インプロセス・サーバーから PDF ファイルを出力すると、フォントの位置に問題が発生します。

解決策

これらの問題をすべて対処する手順は、次のとおりです。

1. レジストリ・エディタ (regedit.exe など) を使用して Windows のレジストリを開きます。編集する前に、レジストリのバックアップを作成します。

2. 次に示すキーにナビゲートします。

```
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Windows
```

3. このキーの Device の文字列値をコピーします。次に例を示します。

```
\MOWGLI\sierra,winspool,Ne02:
```

4. 次に示すキーにナビゲートします。

```
HKEY_USERS\DEFAULT\Software\Microsoft\Windows\CurrentVersion\Windows
```

5. HKEY_CURRENT_USER からコピーした Device の値を貼り付けます (このキーの Device の文字列値は空になります)。

注意： この対処方法は、Default Printer の値を変更するたびに実行する必要があります。

この対処方法は、OPMN-managed Reports Server では機能しません。

インプロセス・サーバーの実行の長所と短所は、第 1 章「OracleAS Reports Services のアーキテクチャについて」を参照してください。その他の詳細は、第 3.4.10 項「インプロセス・サーバーの指定」および第 3.4.11 項「インプロセス・サーバーの識別」を参照してください。

D.2 パフォーマンスの問題の診断

レポートの実行時間を改善する方法および全体的なパフォーマンスを効率化する方法の詳細は、第 20 章「Oracle Reports のチューニング」を参照してください。

Oracle Reports のトレース・オプションを使用して、Oracle Reports のパフォーマンス関連の問題などをトレースおよび診断する方法の詳細は、第 20.1.2 項「レポート・トレース」を参照してください。

D.3 フォントの問題の診断

UNIX マシンでフォントを使用する際の一般的な問題および解決策の詳細は、第 7 章「クロス・プラットフォームの移植問題の解決」を参照してください。

D.4 印刷の問題の診断

UNIX でレポートを印刷する際の一般的な問題および解決策の詳細は、第 5 章「UNIX での Oracle Reports による印刷」の第 5.7 項「よくある質問」を参照してください。

D.5 JDBC PDS の問題の診断

JDBC PDS を使用する際の一般的な問題および解決策の詳細は、第 9 章「JDBC PDS の構成と使用」の第 9.4 項「トラブルシューティング情報」を参照してください。

D.6 OracleAS Portal の問題の診断

レポートをポートレットまたはアイテム・リンクとして OracleAS Portal に公開する際の一般的な問題およびソリューションの詳細は、第 12 章「OracleAS Portal でのレポートの配布」の第 12.4 項「トラブルシューティング情報」を参照してください。

D.7 グローバリゼーションの問題の診断

Oracle Reports の一般的なグローバリゼーションの問題および解決策の詳細は、第 18 章「グローバリゼーションと双方向のサポートの実装」の第 18.7 項「グローバリゼーションに関する問題のトラブルシューティング」を参照してください。

D.8 Oracle Reports ブリッジの問題の診断

ビルトインのブロードキャスト・メカニズムを使用中に、サブネット間で通信の問題が発生した場合、Oracle Reports ブリッジに関連している可能性があります。この問題を診断するには、第 3.3.2.6 項「trace」の説明に従って、Oracle Reports ブリッジのトレースを有効にします。

問題 1

Oracle Reports ブリッジのトレース・ファイルには、次の情報が含まれます。

```
[2005/12/8 1:45:7:339] Info 50103 (BridgeConnection:getResponsePacket):
Getting response object from remote bridge usunrao06.us.oracle.com/130.35.37.76:14011
[2005/12/8 1:45:8:340] Debug 50103 (BridgePacketHandler:handleRequestPacket): Got
response from remote subnet null
[2005/12/8 1:45:8:991] Error 64013 (BridgeConnection:getResponsePacket):
Bridge failed to serve the requestjava.net.ConnectException: Connection refused:
connect
```

解決策 1

リモートの Oracle Reports ブリッジが実行されていません。第 2.2 項「Oracle Reports ブリッジの起動と停止」の説明に従って、Oracle Reports ブリッジを起動します。

問題 2

ローカル・サブネット（クライアントが配置されているサブネット）にある Oracle Reports ブリッジのトレース・ファイルには、次の情報が含まれます。

```
[2005/12/8 1:50:34:219] Info 50103 (BridgeConnection:getResponsePacket):
Getting response object from remote bridge usunrao06.us.oracle.com/130.35.37.76:14011
[2005/12/8 1:50:34:469] Debug 50103 (NetworkUtility:write): Writing .....
[2005/12/8 1:50:34:469] Debug 50103 (NetworkUtility:read): Reading .....
[2005/12/8 1:50:35:220] Debug 50103 (BridgePacketHandler:handleRequestPacket): Got
response from remote subnet null
```

さらに、リモート・サブネット (Reports Server が配置されているサブネット) にある Oracle Reports ブリッジのトレース・ファイルには、次の情報が含まれます。

```
[2005/12/8 2:36:59:997] Debug 50103 (Multicast:registerReceiver): Packet handler
registered
[2005/12/8 2:37:1:16] Info 65003 (NetworkUtility:getIOR): Request timed out
[2005/12/8 2:37:2:19] Info 65003 (NetworkUtility:getIOR): Request timed out
[2005/12/8 2:37:3:30] Info 65003 (NetworkUtility:getIOR): Request timed out
[2005/12/8 2:37:3:31] Debug 50103 (NetworkUtility:getIOR): No response from server
retuning null ior
[2005/12/8 2:37:3:32] Info 50103 (Multicast:registerReceiver): Packet handler
unregistered
[2005/12/8 2:37:3:32] Debug 50103 (NetworkUtility:write): Writing .....
```

解決策 2

Reports Server がリモート・サブネットで実行されていません。第 2.1 項「Reports Server の起動と停止」の説明に従って、Reports Server を起動します。

比較のために、次のサンプル出力で、Oracle Reports ブリッジが Reports Server の検出に成功した場合の Oracle Reports ブリッジのトレースを示します。

```
[2005/12/8 4:4:6:700] Info 50103 (BridgeConnection:getResponsePacket):
Getting response object from remote bridge usunrao06.us.oracle.com/130.35.37.76:14011
[2005/12/8 4:4:6:950] Debug 50103 (NetworkUtility:write): Writing .....
```

```
[2005/12/8 4:4:6:950] Debug 50103 (NetworkUtility:read): Reading .....
```

```
[2005/12/8 4:4:7:932] Error 50103 (BridgeConnection:run): Got response
[2005/12/8 4:4:7:942] Debug 50103 (BridgePacketHandler:handleRequestPacket):
Got response from remote subnet Response Packet -
ServerName      = vin
CorrelationID   = 1134056309492
SenderID        = ServerName: vinVMID: 7e444dbc56c79b06:f9c40:10809ddbdaaf:-8000
Duplicate       = false
Type            = FULL
Add. Info       = Type = server : Host = usunrao06.us.oracle.com
```

問題 3

Reports Server はリモート・サブネット上で実行されており、Oracle Reports ブリッジは両方のサブネット上で実行され、正しく構成されていますが、Oracle Reports クライアントはリモート Reports Server に接続できません。

これは、ブリッジ構成ファイルのタイムアウトの設定値が不適切であることが原因となっている場合があります。このような場合、ローカル・サブネット (クライアントが配置されているサブネット) にある Oracle Reports ブリッジは、リモート・サブネットにある Oracle Reports ブリッジが応答する前に、タイムアウトになる場合があります。

解決策 3

Oracle Reports ブリッジ構成ファイル (rwbridge_bridgename.conf) のタイムアウト値を増やします。次に例を示します。

```
<bridge version="10.1.2" port="14011" timeout="2000">
```

ローカル・サブネットにある Oracle Reports ブリッジは、リモート・サブネットにある Oracle Reports ブリッジからのレスポンスを、タイムアウトになるまで待ちます。ネットワークの接続スピードが低いと、Oracle Reports ブリッジがタイムアウトする場合があります。

さらに、Oracle Reports ブリッジが応答する前に Oracle Reports クライアントがタイムアウトしないように、ネットワーク構成ファイル (rwnetwork.conf) のタイムアウト値を増やします。次に例を示します。

```
<multicast channel="228.5.6.7" port="14021" timeout="1000" retry="3"/>
```

一般的には、Oracle Reports ブリッジのタイムアウトを設定する場合は、次の手順を実行します。両方のブリッジが正しく構成されていることを前提とします。

1. ローカル・サブネットにある Oracle Reports ブリッジに、非常に大きなタイムアウト値を設定します。
2. 両方の Oracle Reports ブリッジを起動します。
3. (リモート・サブネットにある) Reports Server を起動します。
4. ローカル・サブネットで、次のコマンドを使用して `rwdiag` ユーティリティを実行します。

```
rwdiag -find server_name
```

このコマンドによって、Oracle Reports ブリッジによるリモート Reports Server の検出所要時間が出力されます。次に例を示します。

```
D:\orawin\reports\conf>rwdiag -find vin
```

```
Broadcast mechanism used to locate servers
```

```
-----  
Channel address = 228.5.6.7
```

```
Channel port      = 14021
```

```
'vin' found in the network
```

```
Time taken - 1181 milliseconds
```

```
Name = vin : Type = server : Host = usunrao06.us.oracle.com
```

この例では、Oracle Reports ブリッジはリモート Reports Server の検出に 1181 ミリ秒かかっています。

5. 次のように、ローカル・サブネットにある Oracle Reports ブリッジのタイムアウト値を予測します。

```
timeout = time taken to discover the remote Reports Server * safety factor
```

次に例を示します。

```
timeout = 1181 * 1.3
```

これは、約 1500 ミリ秒になります。

6. `repbrg_bridgename.conf` ファイルでタイムアウト値を設定します。次に例を示します。

```
<bridge version="10.1.2" port="14011" timeout="1500">
```

7. `rwnetwork.conf` ファイルのタイムアウト値が、次の条件に従っているか確認します。

```
timeout (in rwnetwork.conf) * retry (in rwnetwork.conf) > timeout (in repbrg_bridgename.conf)
```

次に例を示します。

```
timeout (in rwnetwork.conf) * 3 > 1500
```

したがって、`rwnetwork.conf` の `timeout` の値は、500 以上にする必要があります。

問題 4

どの Oracle Reports ブリッジが構成され、サブネットで実行されているか確認することが必要になる場合もあります。

解決策 4

rwdiag ユーティリティを使用して、すべての Reports Server と Oracle Reports ブリッジを検出します。たとえば、次のようなコマンドを実行します。

```
rwdiag -findall
```

このコマンドにより、次の例のような出力が生成されます。

```
C:¥>rwdiag -findall
Broadcast mechanism used to locate servers
-----
Channel address = 228.5.6.7
Channel port    = 14021

(1) Name = bugupdate : Type = server : Host = strep15.idc.oracle.com
(2) Name = rep_supadhya-pc_frhome1 : Type = server : Host = supadhya-pc.idc.oracle.com
(3) Name = vinod : Type = server : Host = strep10.idc.oracle.com
(4) Name = rep_strep10 : Type = server : Host = strep10.idc.oracle.com
(5) Name = abc : Type = bridge : Host = strep12.idc.oracle.com
(6) Name = rep_stfrm08_frhome1 : Type = server : Host = stfrm08.idc.oracle.com
(7) Name = rep_stport79_as101202mid : Type = server : Host = stport79.idc.oracle.com
(8) Name = rep_iwinreb20_0508041930_bif : Type = server : Host =
iwinreb20.us.oracle.com
C:¥>
```

rwdiag ユーティリティの詳細は、付録 E 「Reports Server とブリッジの診断ユーティリティ」を参照してください。

D.9 その他の解決策

Oracle MetaLink (<http://metalink.oracle.com>) には、さらに詳しい解決策が掲載されています。問題の解決策が見つからない場合は、オラクル社カスタマ・サポート・センターでサービス・リクエストを受け付けています。

オラクル社カスタマ・サポート・センターに問題のトラブルシューティングを依頼する場合は、次の手順を実行してください。

1. 第 20.1.2 項「レポート・トレース」の説明に従って、レポートの実行をトレースします。
2. オラクル社カスタマ・サポート・センターに問い合わせます。オラクル社カスタマ・サポート・センターに問題のトラブルシューティングを依頼するには、必要に応じて、トレース情報、配布のシナリオ、レポート出力のサンプルなどが含まれる Zip ファイルを提供してください。

関連項目：

- Oracle Technology Network (OTN) の Oracle Application Server のリリース・ノート

Reports Server とブリッジの 診断ユーティリティ

この付録では、Reports Server とブリッジの診断ユーティリティ `rwdiag` の引数および使用方法について説明します。

- [rwdiag の概要](#)
- [コマンドライン構文](#)

トラブルシューティングの例と解決策については、[第 D.8 項「Oracle Reports ブリッジの問題の診断」](#)を参照してください。

E.1 rwdiag の概要

rwdiag は、ネットワーク上の Reports Server とブリッジを検出し、ネットワーク上のパケットのブロードキャストを Reports Server とそのクライアント単位で監視するためのユーティリティです。このユーティリティは、`ORACLE_HOME/reports/conf/rwnetwork.conf` とブリッジのタイムアウト値の最適な設定を選択する場合にも役立ちます。rwdiag は、Borland 社の VisiBroker ORB に付属する `osfind` ユーティリティと似ています。このユーティリティは、Oracle Reports 10g リリース 2 (10.1.2) で JDK ORB に置き換わっています。

注意： Oracle Reports 10g リリース 2 (10.1.2) では、Borland 社の VisiBroker が、業界標準である Sun 社の Java Developer's Kit Object Request Broker (JDK ORB) に置き換わっています。JDK ORB は、異なるサブネットのクライアントからの Reports Server リクエストをサポートし、サブネット内とサブネット間の両方で、動的に Reports Server を検出するためのブロードキャスト・メカニズムを有効にします。

rwdiag は、使用しているプラットフォームに応じて、次の 2 つのスキ립トのいずれかで起動できます。

Microsoft Windows の場合：

```
ORACLE_HOME/bin/rwdiag.bat
```

UNIX の場合：

```
ORACLE_HOME/bin/rwdiag.sh
```

E.1.1 例

次の各項では、rwdiag の使用例を示します。

E.1.1.1 例 1

このコマンドラインは、デフォルトの検出タイムアウト値 10 秒を使用して、ネットワークから abc という名前のブリッジまたは Reports Server を検出します。

```
rwdiag.bat -find abc
```

ネットワークで abc が検出されると、検出成功を示すメッセージ、名前、タイプ、ホスト名および所要時間が返されます。Oracle Reports の検出メカニズムとしてネーミング・サービスが使用されている場合は、このユーティリティにホスト名が使用できないため、検出成功を示すメッセージのみが返されます。

E.1.1.2 例 2

このコマンドは、検出タイムアウトを 5 秒に指定して、ネットワークで abc という名前のブリッジまたは Reports Server を検出します。

```
rwdiag.bat -find abc -timeout 5
```

E.1.1.3 例 3

このコマンドは、構成ファイル xyz.conf 内の設定を使用して、ネットワークで abc という名前のブリッジまたは Reports Server を検出します。

```
rwdiag.bat -find abc -conf xyz.conf
```

xyz.conf の内容は次のとおりです。

```
<?xml version = '1.0' encoding = 'ISO-8859-1'?>
<!DOCTYPE discoveryService SYSTEM "file:c:\orawin\reports\dt\rdnetworkconf.dtd">
<discoveryService>
  <multicast channel="105.2.3.8" port="35078" timeout="1000" retry="3"/>
  <!--namingService name="Cos" host="localhost" port="9999"/-->
</discoveryService>
```

構成ファイルからどのようにチャンネル・アドレスとポート番号が選択されるかに注目してください。abc がなんらかの理由で別のポートで実行されていると、検出されません。

E.1.1.4 例 4

このコマンドは、ネットワーク上のすべての Reports Server とブリッジの検出を試みます。

```
rwdiag.bat -findAll
```

ブロードキャスト・メカニズムにより、すべての情報が用意されます。Oracle Reports の検出メカニズムとしてネーミング・サービスが使用されている場合は、ホスト情報は使用できません。

E.1.1.5 例 5

このコマンドは、すべてのブロードキャスト・パケットを Reports Server とそのクライアント単位で監視し、画面にパケット情報を表示します。[q] を押してから [Enter] を押すと、監視が停止します。

```
rwdiag.bat -monitor
```

E.1.1.6 例 6

このコマンドは、すべてのブロードキャスト・パケットを Reports Server とそのクライアント単位で監視し、ログ・ファイル c:\log.txt にパケット情報を保存します。[q] を押してから [Enter] を押すと、監視が停止します。

```
rwdiag.bat -monitor -log c:\log.txt
```

E.2 コマンドライン構文

rwdiag には、次の操作を実行するためのキーワードを使用します。

- ネットワーク上で稼動する Reports Server またはブリッジを検索する。
- ネットワークで稼動しているすべての Reports Server またはブリッジをリストする。
- ネットワークのブロードキャスト・パケットを Reports Server またはクライアント単位で監視する。

E.2.1 構文

```
rwdiag.bat | rwdiag.sh {-find server_or_bridge_name | -findAll |
  -monitor [-log log_file_name]} [-conf config_file_name] [-timeout seconds]
```

ここでは

rwdiag.bat は、Microsoft Windows 用のスクリプトです。

rwdiag.sh は、UNIX 用のスクリプトです。

-find server_or_bridge_name は、server_or_bridge_name という名前のブリッジまたは Reports Server を検出します。

-findAll は、ネットワーク上のすべての Reports Server とブリッジを検出し、リストします。

-monitor は、ネットワーク上のブロードキャスト・パケットをリストします。監視を終了するには、[q]を押してから [Enter] を押します。このオプションは、ネットワーク構成ファイルに指定されている検出メカニズムがネーミング・サービスの場合、使用できません。

-log *log_file_name* は、監視結果を書き込むログ・ファイルを指定します。指定しない場合、監視結果は画面に表示されます。ファイル名には絶対パスを使用できます。ファイル名のみを指定した場合、ログ・ファイルは現在使用されているフォルダに作成されます。

-conf *config_file_name* は、カスタムの構成ファイルを指定します。指定しない場合、`rwnetwork.conf` がデフォルトのファイル名になります。検出メカニズム（ブロードキャストまたはネーミング・サービス）やポート番号などの設定は、このファイルから取得されます。このユーティリティでは、構成ファイルは `ORACLE_HOME\reports\conf` にあるものと見なされます。存在しないファイルを指定すると、そのファイルが `rwnetwork.template` 内のデフォルト設定を使用して作成されます。

-timeout *seconds* は、タイムアウト値を秒単位で指定します。指定しない場合、デフォルト値は 10 秒になります。タイムアウトは、クライアントがリクエスト・パケットをブロードキャストしてから、サーバーのレスポンスを待機する時間です。このオプションは、ネットワーク構成ファイルに指定されている検出メカニズムがネーミング・サービスの場合、無視されます。

E.2.2 使用上の注意

- 検出メカニズムにネーミング・サービスを使用している場合、ホスト情報は使用できません。
- 検出メカニズムにネーミング・サービスを使用している場合、検出はネーミング・サービス内の **Reports Server** 名に基づいて行われるため、サーバーの検出に要した時間は表示されません。このユーティリティがサーバーのレスポンスを待機する必要がないため、ネーミング・サービスの場合は、所要時間は意味がありません。
- ブリッジは、ネーミング・サービスにバインドしていないため、ネーミング・サービスを使用して検出できません。Reports Server の実装のみがネーミング・サービスにバインドしています。
- 構成ファイルにあるタイムアウト値は無視されます。コマンドラインで指定した値のみが適用されます。コマンドラインに値を指定しない場合、デフォルト値は 10 秒になります。
- 検出対象となる **Reports Server** が検出されなかった場合、サーバーが見つからなかったことを示す REP-50504 メッセージが生成されます。

用語集

AFM

Adobe Font Metrics の略称。AFM ファイルおよび PPD ファイルは、Adobe およびプリンタのベンダーによって提供される。これらのファイルには、プリンタに関する情報が含まれる。Oracle Reports では、これらのファイルから、プリンタで使用可能なフォントに関する情報が他のパラメータとともに読み取られる。PPD ファイルに記述されているすべてのフォントについて、フォント名を基に対応する AFM ファイルが検索され、有効な AFM が存在するすべてのフォントがロードされる。

CGI

Common Gateway Interface の略称。Web サーバーと CGI プログラム間の情報の転送に使用される標準規格。CGI は、HTTP リクエストの一部として引数をプログラムに渡す方法を指定し、プログラムで使用可能になる一連の環境変数を定義する。プログラムはその後、ブラウザに返す出力を生成する。CGI は、Web サーバーがユーザーと動的に対話できるよう、サーバー側の処理を提供する。

HTML

Hypertext Markup Language の略称。インターネットの Web サーバー上のコンテンツおよび他のドキュメントへのリンクの指定に使用される、タグベースの ASCII 言語。エンド・ユーザーは、Web ブラウザから HTML ドキュメントを表示し、リンクを追って他のドキュメントを表示する。

HTTP

HyperText Transfer Protocol の略称。Web ブラウザ・コンピュータとアクセス先の Web サーバー間で Web 通信を行うためのプロトコル。

IANA

Internet Assigned Numbers Authority の略称。インターネット全体の新しい IP アドレスの割当てを管理する、Internet Architecture Board (IAB) の下位組織。IANA 定義のキャラクタ・セットとは、charset タグに対して定義でき、インターネットで使用可能なキャラクタ・セットのことを指す。

J2EE

Java 2 Platform, Enterprise Edition の略称。Java で企業アプリケーションを開発、デプロイするための環境。複数層からなる Web ベースのアプリケーションの開発を可能にする、一連のサービス、Application Programming Interface (API)、プロトコルを提供する。

JAR

Java ARchive の略称。複数のファイル (Java クラス・ファイル、イメージなど) を 1 つのファイルに統合する際に使用するファイル。

Java

プラットフォームに依存しないサーブレットまたはアプレットの形式でインターネット向けのプログラミングをサポートするコンピュータ言語。

JSP

JavaServer Page の略称。JSP テクノロジーは、Sun 社の Java サブレット・テクノロジーの拡張機能で、Web ページ上に動的コンテンツを表示するための簡単なプログラミング手段を提供する。JSP はサーバー側テクノロジーである。また、JSP は、Web またはアプリケーション・サーバーで実行される埋込みの Java ソース・コードを持つ HTML ページである。HTML は Web ブラウザに表示されるページ・レイアウトを提供し、Java はビジネス・ロジックを提供する。

Oracle Application Server (OracleAS)

ネットワーク・アプリケーションを配置するための戦略的プラットフォーム。アプリケーション・ロジックをアプリケーション・サーバーに移動し、ネットワーク・クライアントを配置することで、複雑さが緩和され、管理が容易になり、開発と配布が簡素化されるため、組織は大幅な省力化を実現できる。OracleAS のみが、ビジネスに不可欠なプラットフォームを提供する。このプラットフォームによって、データベース Web パブリッシングが容易になるほか、従来のクライアント / サーバー・アーキテクチャからネットワーク・アプリケーション・アーキテクチャに移行しつつ、過去の資産を完全に統合できる。

Oracle Developer Suite

Oracle の優れたアプリケーション開発ツールとビジネス・インテリジェンス・ツールを結合して 1 つの統合製品としたもの。Java や XML などのインターネット標準に基づくこの製品は、Oracle Application Server および Oracle Database 向けのアプリケーションを作成するための、高い完成度と生産性を持つ開発環境を提供する。

OracleAS Portal

スケーラブルかつセキュアで拡張可能な HTML アプリケーションおよび Web サイトを作成するためのブラウザベースの開発ツール。OracleAS Reports Services では、OracleAS Portal を使用して様々な情報を保存することにより、Web 上で発行されたレポートに対するエンド・ユーザーのアクセスを制御する。この保存する情報とは、レポート・リクエスト、保護されているサーバー、およびレポート出力の印刷に使用する OracleAS Reports Services プリンタに関するものである。

OracleAS Reports Services

「[Reports Services](#)」を参照。

ORACLE_HOME

ディレクトリベースのオペレーティング・システムにおける、Oracle ディレクトリ階層の最上位ディレクトリの別名。Oracle 製品のルート・ディレクトリを示す環境変数。

ORACLE_HOME に指定されたディレクトリは、次の構文で参照できる。

UNIX の場合 : \$ORACLE_HOME

Windows の場合 : %ORACLE_HOME%

PDF

Portable Document Format の略称。ドキュメントを、その作成に使用されたアプリケーション・ソフトウェア、ハードウェアおよびオペレーティング・システムに依存することなく表示できるようにする Adobe Acrobat 専用のファイル形式。PDF ファイルでは、テキスト、グラフィックおよびイメージのあらゆる組合せを含むドキュメントを、デバイスや解像度に依存しない形式で記述できる。

PL/SQL

SQL 言語に対する Oracle 独自の拡張。アプリケーションの作成に適したプロシージャ構造やその他の構造を SQL に追加する。

PPD

PostScript Printer Definition の略称。PPD ファイルおよび AFM ファイルは、Adobe およびプリンタのベンダーによって提供される。これらのファイルには、プリンタに関する情報が含まれる。Oracle Reports では、これらのファイルから、プリンタで使用可能なフォントに関する情報が他のパラメータとともに読み取られる。PPD ファイルに記述されているすべてのフォントについて、フォント名を基に対応する AFM ファイルが検索され、有効な AFM が存在するすべてのフォントがロードされる。

RDF ファイル (RDF file)

単一のレポート定義をバイナリ形式で保存したファイル。RDF ファイルは、レポートを実行および編集するために使用される。

Reports Builder (rwbuilder)

Oracle Reports の実行可能ファイル。これにより、レポート開発者はレポート定義を作成および保守できる。

Reports CGI (rwcgi)

注意： Oracle Reports 10g では、Reports CGI (rwcgi) は使用不可になり（下位互換性を確保するためにのみ維持されている）、かわりに、Reports JSP、rwservlet (Reports Servlet) または Reports Web サービスを使用することができる。

Common Gateway Interface (CGI) または Reports Web Cartridge と呼ばれる Oracle Reports の実行可能ファイル。Web サーバーまたは J2EE コンテナ (OC4J など) と Reports Server との間で情報の変換と配布を行い、Web ブラウザからレポートを動的に実行する。

Reports Engine

OracleAS Reports Services のコンポーネント。データソースからのデータの取得、レポートのフォーマット、キャッシュへの出力の送信を行い、ジョブの準備が完了したことを Reports Server に通知する。

Reports Queue Manager (rwrqm)

(Windows のみ) Reports Server (rwservlet) で管理されているレポート・ジョブのタイムスタンプやステータス情報を保持する。

Reports Runtime (rwrn)

OracleAS Reports Services インプロセス・サーバーを使用してレポートを実行する、Oracle Reports の実行可能ファイル。

Reports Server (rwservlet)

Oracle Reports の実行可能ファイル。これにより、レポート・サービスは全社規模でレポートを実行、配布および公開できる。OracleAS Reports Services のコンポーネントで、ユーザー認証、スケジューリング、キャッシュ、レポート配布などのクライアント・リクエストを処理する。レポートを Reports Server に送信するには、rwservlet、Reports JSP、CGI、rwclient などの Oracle Reports クライアントを使用する。

Reports Services

Reports Developer アプリケーションのランタイム環境。OracleAS Reports Services では、各レポートを全社規模で実行、配布および公開できる。OracleAS Reports Services を使用してレポートを配置することで、柔軟性が増し、時間が節約され、処理能力も高まる。

Reports Servlet (rwservlet)

OracleAS Reports Services のコンポーネント。Web サーバーまたは J2EE コンテナ (OC4J など) と Reports Server との間で情報の変換と配布を行い、Web ブラウザからレポートを動的に実行できる。

Reports キャッシュ (Reports cache)

完了したジョブの出力を格納する OracleAS Reports Services のコンポーネント。

Reports クライアント (rwclient)

Oracle Reports の実行可能ファイル。コマンドライン・インタフェースを使用してレポートをリモートの Reports Server (rwserver) に送信できる。

REP ファイル (REP file)

単一のレポート定義をバイナリ形式で保存したファイル。.REP ファイルはレポートの実行用のみ使用され、編集はできない。

SELECT 文 (SELECT statement)

1 つ以上の表またはビューからフェッチする行と列を指定する SQL 文。

SQL

リレーショナル・データベースに対して、情報を格納または取り出すための標準インタフェース。SQL は、Structured Query Language の略称。

SQL スクリプト (SQL script)

SQL 文を含むファイル。これを実行すると、データベースの管理をすばやく容易に実施できる。Oracle 製品には SQL スクリプトがいくつか付属している。

SQL ファイル (SQL file)

テキスト形式 (ASCII や EBCDIC など) で保存された問合せを含むファイル。

SQL 文 (SQL statement)

Oracle に対する SQL 命令。SELECT 文は SQL 文の一種である。

URL

Uniform Resource Locator の略称。インターネットを通じて利用できるリソースの場所を簡潔な文字列で表したもの。また、クライアントが OracleAS へのリクエストをエンコードするために使用するテキスト文字列の書式でもある。

Web サーバー (Web server)

Web サイトで実行されるサーバー・プロセス (HTTP デーモン) で、リモートの Web ブラウザからの HTTP リクエストに応じて Web ページを送信する。

Web ソース・ビュー (Web Source view)

レポート・エディタのビューの 1 つ。レポートの HTML または JSP ソースを表示する。このビューでレポート・ブロック・ウィザードおよびグラフ・ウィザードを使用することにより、Web ページに動的なコンテンツを追加できる。熟練した Java 開発者であれば、このビューで直接 Web ソースを編集できる。

Web ブラウザ (Web browser)

エンド・ユーザーが、コンピュータに格納されている HTML ドキュメントおよびプログラム (Web サーバーを通じて提供される) を読み取るために使用するプログラム。

XML

Extensible Markup Language の略称。SGML を使用してデータの定義と構造化を行うメタ言語。Reports Builder では、XML の出力をサポートすることで、Web パブリッシングおよびサードパーティ製アプリケーションとの電子データ交換を実現している。XML を使用して、実行時に他のレポート定義とマージできるレポート定義や、個別に実行できるレポート定義を作成することもできる。

アイコン (icon)

ウィンドウやツールのグラフィック表示。

イメージ (image)

アプリケーションに格納およびロードできるビットマップ・オブジェクト。クライアントは、インポートされたイメージを変更できない。

イントラネット (intranet)

内部的な TCP/IP ネットワークであり、これに対するアクセスは（ファイアウォールにより）企業や組織内の人物に制限されている。イントラネットは、インターネットと同様のサービスを組織内で提供するが、必ずしもインターネットに接続されているとはかぎらない。企業内で情報やアプリケーションの配布のため、内部ネットワーク上に 1 つ以上の Web サーバーをセットアップするのはイントラネットの一般的な例である。

ウィザード (wizard)

一般的に実行されるタスクを、順を追って実行するためのインタフェース。Reports Builder でのウィザードは、次のとおり。

- レポート・ウィザード：基本的なペーパー・レポートまたは Web レポートを作成するための手順を示す。ウィザードの各ページでは、最初のレポートを作成するための情報の入力が必要とされる。
- データ・ウィザード：複数問合せデータ・モデルで使用する問合せを簡単に定義または変更できるようにする。
- グラフ・ウィザード：真の 3 次元グラフなどの様々なチャートやグラフを追加する。Reports Builder では、Oracle BI Graph Bean で実装される。
- レポート・ブロック・ウィザード：静的 HTML ページにデータを追加できるようにする。

エディタ (editor)

「[ビュー](#)」を参照。

オブジェクト (object)

1. レイアウト上に配置できるアイテム。オブジェクトの例としては、四角形、線、楕円、円弧、多角形、折れ線、丸い四角形、手書き、チャート、テキスト、記号、テキスト・フィールドなどがある。
2. Oracle のデータベースにおけるオブジェクト・タイプのインスタンス。オブジェクト表の行、またはリレーショナル表の列オブジェクトに含まれる行の一部がオブジェクトとなる。

オブジェクト・ナビゲータ (Object Navigator)

アプリケーション・オブジェクトの検索と操作を迅速かつ容易に実行できる、階層的な表示および編集インタフェース。次のような機能がある。

- インデントと開閉可能ノードによる階層表現（トップ・レベルのノードに、モジュール・タイプ、データベース・オブジェクトおよびビルトイン・パッケージを表示）。オブジェクトの作成、編集、削除、名前の変更などのタスクを実行できる。
- 検索のフィールドとアイコン。あらゆるレベルのノードやノード内の個別項目の順方向 / 逆方向検索を実行できる。
- 水平ツールバーの各種アイコン。「ファイル」メニューの一般的な機能と同じもの。

外部キー (foreign key)

表内の値または列で、別の表の主キーを参照するもの。

行 (row)

表内のフィールド値の 1 セット。たとえば、サンプル表 EMP 内の 1 人の従業員を表すフィールドなど。

繰返し枠 (repeating frame)

グループに対してフェッチされたデータの行の表示に使用されるレイアウト・オブジェクト。

グループ (group)

1. Reports Builder では、問合せによって選択されたすべての列を格納するために自動的に作成されるか、レポートに表示するデータの階層を変更するためにユーザーが作成するデータ・モデル・オブジェクト。グループは主にレポートでのブレイクの作成や、計算のリセットを目的として使用される。
2. 他の複数のオブジェクトで構成されるオブジェクト。

サーブレット

Web サーバーまたはアプリケーション・サーバーで実行される Java アプリケーション。通常はデータベースへのアクセス、E-Commerce プロセスなどのサーバー側の処理を行う。サーブレットは Java で記述されているため、サーバー間やオペレーティング・システム間での移植性が高い。

Reports Servlet (rwservlet) と JSP は、カスタム (JSP) レポート・タグを処理し、Oracle HTTP Server と Reports Server との間で情報を配布する OracleAS Reports Services のコンポーネントである。

式列 (formula column)

PL/SQL ファンクションまたは式、SQL 文、あるいはこれらの組合せからデータを取得するユーザー作成の列。

書式マスク (format mask)

フィールドの値の表示方法を定義する設定。たとえば、書式マスクは通貨金額や日付の表示の指定に使用される。

スキーマ (schema)

関連するデータベース・オブジェクトの集合。通常、データベース・ユーザー ID によってグループ化される。スキーマ・オブジェクトには、表、ビュー、順序、ストアド・プログラム・ユニット、シノニム、索引、クラスタ、データベース・リンクなどがある。

スタイル・シート (style sheet)

HTML ドキュメントで強力なフォーマットの柔軟性を提供する、HTML の拡張書式。スタイル・シートを利用する HTML ドキュメントは、スタイル・シートをサポートしているブラウザで表示する必要がある。

ダイアログ・ボックス (dialog box)

操作を完了するために必要な情報の入力を求める、部分的な画面またはウィンドウ。

ツール (tool)

アプリケーションでオブジェクトの作成や操作に使用されるアイコン・ボタン。

ツールバー (toolbar)

製品のコマンドを実行するアイコン・ボタンの集合。通常、ウィンドウ上部に水平に表示されるか、ウィンドウの右側または左側に垂直に表示される。

ツール・パレット (tool palette)

ユーザー・インタフェースのアイコン・ボタンによって表されるツールの集合。これにより、レポート開発者は、ペーパー・レイアウト・ビューでの四角形の描画やデータ・モデル・ビューでの問合せの作成などのタスクを実行できる。

データソース (data source)

問合せから返されるデータのソース。表、ビュー、シノニム、スナップショット、ビューとして格納された問合せなどのデータベース・オブジェクトがある。OracleAS Reports Services では、様々なデータソースにアクセスできる。

Oracle Open Client Adapter (OCA) は新しいプラグgable・データソース (PDS) アーキテクチャに置き換えられ、Open Database Connectivity (ODBC) ドライバは、Oracle Reports 10g ではサポートされなくなった。ただし、Java Database Connectivity (JDBC) は、JDBC-ODBC Bridge を使用できるプラグgable・データソースの1つであり、これを利用することにより他のデータソースへのアクセスが可能になる。

データベース (database)

1. ユニットとして扱われる、1組のディクショナリ表とユーザー表。
2. (Oracle OLAP Server) データを編成、格納および操作するオブジェクトを含む単一ファイル (通常は拡張ファイルを伴う)。Oracle OLAP Server には、このようなオブジェクトの例として、変数、ディメンション、式、モデル、プログラムなどがある。

データ・モデル (data model)

データソースからフェッチするデータ、計算する値、およびレポートでのデータの順序を定義するリレーショナル・モデル。データ・モデルを定義する Reports Builder のオブジェクトとして、問合せ、グループ、列、パラメータおよびリンクがある。

データ・モデル・ビュー (Data Model view)

レポート・エディタのビューの1つ。レポート内のデータを構造的に表示する。レポート出力にオブジェクトは表示されないが、構造によってレイアウト・スタイルが決まり、データ・オブジェクトによってレイアウト・オブジェクトに表示される値が提供される。

ディテール問合せ (detail query)

マスター / ディテール・レポートの定義時には、マスター (親) 問合せによって取得された各レコードのすべての関連レコードがディテール問合せによって取得される。

テンプレート (template)

共通のスタイルと標準を含むフレームワークを定義したもの。グラフィックを含む場合もある。テンプレートによって提供される標準書式を基に、専門的な水準のルック・アンド・フィールを備えたレポートを迅速かつ簡単に開発できる。

テンプレート・エディタ (Template Editor)

テンプレートのオブジェクトおよび書式設定プロパティを定義できる作業領域。レポート・エディタのペーパー・レイアウト・ビューに似ている。このマージン領域で、オブジェクト (ページ番号、テキスト、グラフィックなど) を作成、削除および変更できる。本体領域ではオブジェクトの作成も削除も行えないが、本体オブジェクトのプロパティはプロパティ・インスペクタで変更できる。

問合せ (query)

データベースの1つ以上の表またはビューから取り出すデータを指定する SQL SELECT 文。

ハイパーリンク (hyperlink)

ドキュメント内の特定の場所から別のドキュメント (内の特定の場所) または同じドキュメント内の別の場所への参照 (リンク)。Web ブラウザでは通常、ハイパーリンクが強調表示 (異なる色、フォントまたはスタイルの使用により) される。ユーザーがハイパーリンクをアクティブにすると (マウスでクリックすることによって)、ブラウザにリンク先が表示される。

ビュー (view)

1. Reports Builder では、レポートのデータ・モデル、レイアウトまたはパラメータ・フォームの定義など、一連の特定のタスクを実行する作業領域。
2. 実際にはデータベースに存在しないデータ行から構成されるが、データベースに物理的に格納されている表に基づく仮想表。

表 (table)

関連する情報の集合を行と列で構成される 2 次元のグリッド形式にして名前を付けたもの。リレーショナル・データベースまたはサーバーに格納される。

表形式 (tabular)

ページの最上部にラベルを表示し、その下にデータ列を表示したデフォルト・レイアウト。

フィールド (field)

1. その中でデータの入力、編集および削除が可能なインタフェース要素。
2. 特定の問合せ列のデータ表示方法を定義するレイアウト・オブジェクト。

フォーマット・トリガー (format trigger)

オブジェクトのフォーマット属性を動的に変更するときに使用できる PL/SQL ファンクション。

プロパティ・インスペクタ (Property Inspector)

オブジェクト・ナビゲータ、レポート・エディタおよびテンプレート・エディタで現在選択されているオブジェクトのプロパティの表示、検索および設定が可能なウィンドウ。Reports Builder の各オブジェクト（問合せ、グループ、枠、パラメータなど）にはプロパティが関連付けられており、プロパティ・インスペクタを使用して表示できる。プロパティ・インスペクタには、次のような機能がある。

- 開閉可能ノード
- プロパティのインプレイス編集
- 検索機能
- 複数選択
- 複合プロパティ・ダイアログ
- プロパティ・インスペクタの複数のインスタンスの起動機能

プロパティに関するヘルプを参照するには、プロパティ・インスペクタでプロパティをクリックして [F1] を押す。

ペーパー・デザイン・ビュー (Paper Design view)

レポート・エディタのビューの 1 つ。ペーパー・レポートの出力を表示し、レイアウトに対する間隔設定、フィールドの書式設定、カラー設定、テキスト編集などの一般的で簡単な変更の多くを、ペーパー・レイアウト・ビューを開くことなく実行できるようにする。

ペーパー・パラメータ・フォーム・ビュー (Paper Parameter Form view)

パラメータ・フォームのレイアウトを表示し、ユーザーは実行時にランタイム・パラメータ・フォームでパラメータ値を入力することができる。

ペーパー・レイアウト・ビュー (Paper Layout view)

レポート・エディタのビューの 1 つ。ペーパー・レポート内のレイアウト・オブジェクトを表示し、あらゆるレイアウト・オブジェクトに対して多くの変更を実行できるようにする。すべてのレイアウト・オブジェクトには、プロパティ・インスペクタで変更可能なプロパティが設定されている。レイアウト・オブジェクトの階層はデータ・モデルによって決まる。

マージン (margin)

レポート・セクション（ヘッダー、メインまたはトレーラ）の各論理ページの上部および下部に表示されるオプションのレポート・リージョン。マージンにはあらゆるレイアウト・オブジェクトを挿入できるが、通常は、ボイラープレートやフィールド（ページ番号、ページ合計、総計、現在の日付と時刻など）が組み込まれる。

無効化 (disabled)

インタフェース要素の状態の1つ。メニュー項目やボタンなどが現行コンテキストで使用できない（つまり、キーボードやマウスによる入力に反応しない）ことを表す。

有効化 (enabled)

インタフェース要素の状態の1つ。メニュー項目やボタンなどが現行コンテキストで使用できる（つまり、キーボードやカーソル/マウスによる入力に反応する）ことを表す。

ランタイム・パラメータ・フォーム (Runtime Parameter Form)

実行時に必要に応じて表示される画面またはウィンドウ。ユーザーはレポートの実行前に、この画面で印刷オプションやパラメータを変更できる。

レイアウト (layout)

「[ペーパー・レイアウト・ビュー](#)」を参照。

レコード (record)

SQL SELECT 文でフェッチされる1つの行。

列 (column)

1. データの特定のドメインを表す、データベース表内の垂直方向の領域。列には列名 (ENAME など) があり、特定のデータ型 (CHAR など) を持っている。たとえば、従業員情報の表で従業員名すべてが1つの列を構成するなど。レコード・グループ列は、データベースの列を表す。
2. 問合せの SELECT リスト内の列のそれぞれの式に対して自動的に作成されるか、集計や式を実行するため、またはプレースホルダとして使用するために手動で作成されるデータ・モデル・オブジェクト。
3. エンティティの属性を表したものの。

レポート・エディタ (Report Editor)

Reports Builder のウィンドウ。Web レポートおよびペーパー・レポートのデータ・オブジェクトやレイアウト・オブジェクトの処理に役立つ様々なビューを備える。提供されるビューは、次のとおり。

- [データ・モデル・ビュー](#)
- [ペーパー・レイアウト・ビュー](#)
- [ペーパー・デザイン・ビュー](#)
- [ペーパー・パラメータ・フォーム・ビュー](#)
- [Web ソース・ビュー](#)

枠 (frame)

他のレイアウト・オブジェクトを囲むために使用されるレイアウト・オブジェクト。複数のオブジェクトの書式、表示頻度および位置を同時に制御できる。

索引

A

accessible コマンド・キーワード, A-18
Adobe Font Metrics (AFM) ファイル, 4-10
AFM ファイル, 4-10, 4-23
ALLOWHTMLTAGS、rwservlet.properties, 3-50
API
 cache, 3-9
 destinations, 3-16
 engine, 3-10
 notification, 3-19
 Pluggable destinations, 8-2
 repository, 3-21
 security, 3-14
 イベント, 3-18, 17-1 ~ 17-10
 デバッグ・イベント, 17-6
arraysize コマンド・キーワード, 20-24, A-18
attach 配布要素, 15-9
 format 属性, 15-9
 instance 属性, 15-9
 name 属性, 15-9
 srcType 属性, 15-9
authid コマンド・キーワード, 2-5, A-19
authid パラメータ、イベント, 17-4
autocommit コマンド・キーワード, A-20
auxDatFiles 属性, 3-29

B

background コマンド・キーワード, A-20
batch コマンド・キーワード, 2-4, 2-5, A-21
bcc コマンド・キーワード, A-21
bcc 属性
 mail, 15-7
blankpages コマンド・キーワード, A-22
body 配布要素, 15-7
 format 属性, 15-8
 instance 属性, 15-8
 srcType 属性, 15-8
Borland VisiBroker, 1-8, 3-36
bridgeconf.dtd, 3-40
 bridge 要素, 3-41
 identifier 要素, 3-42
 networkConfig 要素, 3-42
 remoteBridges 要素, 3-43
 remoteBridge 要素, 3-43
 trace 要素, 3-44
bridge 構成要素, 3-41

buffers コマンド・キーワード, A-22

C

CA_GPREFS, B-5
CA_UPREFS, B-5
cache, 3-9
 cacheDir プロパティ, 3-9
 cachelob コマンド・キーワード, A-23
 cacheSize プロパティ, 3-9
 Reports Server 構成のチューニング, 20-12, 20-24
 cache 宛先タイプ, 8-5
 cache 構成要素, 3-9, 13-15
 cacheSize プロパティ, 20-12, 20-24
 class 属性, 3-9
 callBackTimeOut 属性, 3-11
cc コマンド・キーワード, A-23
cc 属性
 mail, 15-6
cellwrapper コマンド・キーワード, A-24
CGI, 1-2, 1-5
 URL 構文, 13-4
 下位互換性, 3-8
cgicmd.dat, 3-47, 13-16 ~ 13-17
 エントリの追加, 13-16
 使用方法, 13-17
CID フォント, 4-25
class、JDBC 構成ファイル, 9-4
classPath 属性、engine, 3-11
class 属性
 cache, 3-9
 destination, 3-17
 engine, 3-10
 jobStatusRepository, 3-21
 notification, 3-19
 security, 3-14
cmdfile コマンド・キーワード, A-25
cmdkey コマンド・キーワード, A-26
cmdkey パラメータ、イベント, 17-7
collate コマンド・キーワード, A-26
Common Object Service (COS) ネーミング・サービス
 , 1-8, 3-36
compatible 構成要素, 3-3, 3-8
 version 属性, 3-8
confidential 属性, 3-15, 8-5
CONNECTION_POOLSIZE、
 rwservlet.properties, 3-51
connection、JDBC 構成ファイル, 9-5

- connection 構成要素, 3-24
 - idleTimeout 属性, 3-24
 - maxConnect 属性, 3-24
- connectString, 9-4
- containshtmltags コマンド・キーワード, A-27
- containsole コマンド・キーワード, A-28
- contentarea コマンド・キーワード, A-28
- Cookie, 11-2
- copies コマンド・キーワード, 20-24, A-29
- copies 属性
 - printer, 15-15
- CORBA, A-95
- COS ネーミング・サービス, 1-8
 - OPMN による制御, 3-38, 3-63
- Custom Tag Handler, 1-5
- customize コマンド・キーワード, 16-1, A-29

D

- DAS、「Delegated Administration Services」を参照
- dateformatmask コマンド・キーワード, A-30
- DB2 ドライバ, 9-8
- DBAUTH、rwservlet.properties, 3-49
- dbms_AQadm パッケージ, 17-8
- dbms_aq.dequeue, 17-9
- DEFAULTCHARSET、rwservlet.properties, 3-50
- delauth コマンド・キーワード, A-31
- Delegated Administration Services, 10-12, 11-2, 12-4
- delimited_hdr コマンド・キーワード, A-31
- DELIMITED_LINE_END, B-5
- delimiter コマンド・キーワード, A-32
- DEQUEUE, 17-7
- desformat コマンド・キーワード, A-33
- desname コマンド・キーワード, A-35
- destination
 - 宛先タイプ, 8-4
 - クラス, 8-4
 - 有効な値, 8-4
- destinations 配布要素, 15-4
- destination 構成要素, 3-16, 3-17, 8-4
 - class 属性, 3-17
 - destype 属性, 3-17
- destination コマンド・キーワード, A-37
- destype コマンド・キーワード, A-38
- destype 属性, 3-17, 8-4
 - cache, 8-5
 - file, 8-5
 - mail, 8-5
 - oraclePortal, 8-5
 - printer, 8-5
- destype 配布要素, 15-15
 - id 属性, 15-16
 - instance 属性, 15-17
 - name 属性, 15-16
- dest コマンド・キーワード, A-37
- DEVELOPER_NLS_LANG, 18-6, B-6
- DIAGHEADTAGS、rwservlet.properties, 3-53
- diagnosis プロパティ, D-3, D-12, D-13
- DIAGNOSTIC、rwservlet.properties, 3-48
- discoveryService 構成要素, 3-38
- DISPLAY
 - UNIX での印刷, 3-67, 5-25
- distribute コマンド・キーワード, A-44

- distribution.dtd, 15-2
- dist パラメータ、イベント, 17-7
- DOC, B-6
- DTD、内部, 9-2
- DTD、「データ型ディクショナリ」を参照
- dtype コマンド・キーワード, A-44
- dunit コマンド・キーワード, A-45

E

- EM、「Oracle Enterprise Manager」を参照
- encrypted 属性, 3-15, 8-5
- engineId 属性, 3-19
- engineerestimeout コマンド・キーワード, A-46
- engineResponseTimeout 属性
 - Reports Server 構成のチューニング, 20-13
- engine 構成要素, 3-10
 - callBackTimeout 属性, 3-11
 - classPath 属性, 3-11
 - class 属性, 3-10
 - diagnosis プロパティ, 3-13
 - engineResponseTimeout 属性, 20-13
 - engLife 属性, 3-11
 - id 属性, 3-10
 - initEngine 属性, 3-11, 20-12
 - maxConnect 属性, 20-13
 - maxEngine 属性, 3-11, 20-12
 - maxIdle 属性, 3-11
 - minEngine 属性, 3-11, 20-12
- engLife 属性, 3-11
- ENQUEUE, 17-7
- envid コマンド・キーワード, A-46
- environment 構成要素, 3-31
 - id 属性, 3-32
- envVariable 構成要素
 - envVariable の属性, 3-32
 - name 属性, 3-32
- ERRORTEMPLATE、rwservlet.properties, 3-51
- expiration コマンド・キーワード, 1-6, A-46
 - Reports Server 構成のチューニング, 20-12, 20-24
- expiredays コマンド・キーワード, A-47
- EXPLAIN PLAN、SQL トレース, 20-8
- express_server コマンド・キーワード, A-48

F

- FAX
 - 配布の例, 15-26
- fileName 属性, 3-28
- file 宛先タイプ, 8-5
- file 配布要素, 15-12
 - format 属性, 15-13
 - id 属性, 15-13
 - instance 属性, 15-13
 - name 属性, 15-13
- fnchk.exe, 4-16
- foreach 配布要素, 15-4
- format 属性
 - attach, 15-9
 - body, 15-8
 - file, 15-13
 - 配布, 15-28
- Forms Services

セキュリティに関する考慮事項, 11-14
formsize コマンド・キーワード, A-49
from コマンド・キーワード, A-50
from 属性
 mail, 15-7
FTP
 配布の例, 15-25
FTP 宛先, 3-16, 8-2, A-39

G

gateway パラメータ、イベント, 17-4
getjobid コマンド・キーワード, A-50
getserverinfo コマンド・キーワード, A-51

H

HELP、rwservlet.properties, 3-54
help コマンド・キーワード, A-51
HPD ファイル, 4-10
HTML
 診断の出力, 3-53
 診断の出力 DIAGBODYTAGS、
 rwservlet.properties, 3-53
HTTP Secure Sockets Layer, 1-4
HTTP Server, 1-4, 10-11
HTTPS, 1-4
HTTP タイムアウト値
 Reports Server 構成のチューニング, 20-13

I

identifier 構成要素, 3-29, 3-42
idleTimeOut 属性, 3-24
id 属性, 3-32
 destype, 15-16
 engine, 3-10
 file, 15-13
 mail, 15-6
 notification, 3-19
 orbClient, 3-25
 printer, 15-14
 security, 3-14
IMAGE_URL、rwservlet.properties, 3-53
include 配布要素, 15-10
 src 属性, 15-12
increment 属性
 oidconnection, 3-15
Informix ドライバ, 9-10
initEngine 属性, 3-11
 Reports Server 構成のチューニング, 20-12
init 属性
 oidconnection, 3-15
instance 属性
 attach, 15-9
 body, 15-8
 destype, 15-17
 file, 15-13
 printer, 15-15
itemtitle コマンド・キーワード, A-52
IX ユーティリティ, 5-16

J

Java 2 Enterprise Edition, 1-4
Java Developer's Kit Object Request Broker, 1-8
Java Importer
 パフォーマンス分析, 20-11
Java 仮想マシン (JVM), A-94
Java サブレット, 1-2
Java ストアド・プロシージャ
 パフォーマンス分析, 20-11
JDBC, D-21
JDBC PDS
 トラブルシューティング, D-21
JDBC-ODBC ドライバ, 9-6
JDBC 構成ファイル, 9-2
JDBC 問合せ, 9-12
JDBC プラuggابل・データソース (PDS), 9-1
JDBC レポート, 9-16
JDK ORB, 1-8, 3-36, E-2
jobname コマンド・キーワード, A-52
jobRecovery 構成要素, 3-28
 auxDatFiles 属性, 3-29
jobStatusRepository 構成要素, 3-21
 class 属性, 3-21
 repositoryconn プロパティ, 20-8
jobtype コマンド・キーワード, A-53
jobType 属性, 3-18
job 構成要素, 3-18
 engineId 属性, 3-19
 jobType 属性, 3-18
 securityId 属性, 3-19
JSP, 1-2, 1-5
 URL 構文, 13-3
 VALIDATETAG, 20-23, A-97
 イメージ, 3-53
 キー・マッピングの使用方法, 13-17
 キャラクタ・セットの指定, 18-6 ~ 18-8
 グローバル化・サポート, 18-6 ~ 18-8
JSP レポート、Web またはペーパーへの配布, 13-7
JVM, A-94
jvmoptions コマンド・キーワード, A-53

K

KEYMAPFILE, 3-47
killengine コマンド・キーワード, A-54
killjobid コマンド・キーワード, A-55

L

LDAP, 12-2
ldapmodify コマンド, 11-13
LDIF ファイル, 11-13
loginTimeout, 9-5
log 構成要素, 3-20
 option 属性, 3-21
longchunk コマンド・キーワード, 20-24, A-55

M

mail 宛先タイプ, 8-5
mail 配布要素, 15-6
 bcc 属性, 15-7

- cc 属性, 15-6
- from 属性, 15-7
- id 属性, 15-6
- organization 属性, 15-7
- priority 属性, 15-7
- replyTo 属性, 15-7
- returnReceipt 属性, 15-7
- subject 属性, 15-7
- to 属性, 15-6
- maxConnect 属性, 3-24
 - Reports Server 構成のチューニング, 20-13
- maxEngine 属性, 3-11
 - Reports Server 構成のチューニング, 20-12
- maxIdle 属性, 3-11
- maxQueueSize 属性, 3-27
 - Reports Server 構成のチューニング, 20-12, 20-24
- Merant ドライバ, 9-6
- MESSAGES, 17-7
- mfontchk, 4-16
- mimetype コマンド・キーワード, A-56
- minEngine 属性, 3-11
 - Reports Server 構成のチューニング, 20-12
- mod_oc4j, 1-4
- mode コマンド・キーワード, A-56
- module コマンド・キーワード, A-57
- multicast 構成要素, 3-38
- MyIdent.AuthID, 17-4
- MyIdent.GatewayURL, 17-4
- MyIdent.JobID, 17-4
- MyIdent.ServerName, 17-4

N

- name コマンド・キーワード, A-57
- name 属性, 3-31
 - attach, 15-9
 - destype, 15-16
 - envVariable, 3-32
 - file, 15-13
 - printer, 15-15
- namingService 構成要素, 3-39
- networkConfig 構成要素, 3-42
- NLS
 - 「グローバル化・サポート」を参照, 18-1
 - NLS_CALENDAR, B-6
 - NLS_CREDIT, B-6
 - NLS_CURRENCY, B-6
 - NLS_DATE_FORMAT, B-6
 - NLS_DATE_LANGUAGE, B-6
 - NLS_DEBIT, B-6
 - NLS_ISO_CURRENCY, B-6
 - NLS_LANG, 18-3, B-7
 - NLS_LIST_SEPARATOR, B-8
 - NLS_MONETARY_CHARACTERS, B-8
 - NLS_NUMERIC_CHARACTERS, B-8
 - NLS_SORT, B-8
- nonblocksql コマンド・キーワード, A-58
- notification 構成要素, 3-19
 - class 属性, 3-19
 - id 属性, 3-19
- notifyfailure コマンド・キーワード, A-58
- notifysuccess コマンド・キーワード, A-59
- numberformatmask コマンド・キーワード, A-59

O

- Object Request Broker, 1-8
- OC4J, 10-11
- OIDCON_INCREMENT、rwservlet.properties, 3-52
- OIDCON_INIT、rwservlet.properties, 3-52
- OIDCON_TIMEOUT、rwservlet.properties, 3-52
- oidconnection 構成要素, 3-15
 - increment 属性, 3-15
 - init 属性, 3-15
 - timeout 属性, 3-15
- oidEntity プロパティ, 3-14
- OID、「Oracle Internet Directory」を参照
- olap_con コマンド・キーワード, A-60
- onfailure コマンド・キーワード, A-60
- onsuccess コマンド・キーワード, A-61
- OPMN
 - COS ネーミング・サービスの指定, 3-38, 3-63
 - インプロセス Reports Server の指定, 3-61
 - スタンドアロンの Reports Server の指定, 3-58
 - ブリッジの指定, 3-62
 - ロードするモジュールの指定, 3-58
- OPMN (Oracle Process Management and Notification)
 - , 2-2
 - Reports Server の構成, 3-57
- opmn.xml, 3-58
- option 属性, 3-21
- ORA_PROF, 20-10
- Oracle Advanced Queuing, 17-2, 17-7, 17-8, 17-9
 - dbms_AQadm パッケージ, 17-8
 - dbms_aq.dequeue, 17-9
 - DEQUEUE, 17-7
 - ENQUEUE, 17-7
 - MESSAGES, 17-7
- Oracle Application Server Containers for J2EE, 1-4
- Oracle Application Server Containers for Java 2
 - Enterprise Edition, 10-11
- Oracle Enterprise Manager, 19-1 ~ 19-7
 - Reports Server の起動, 2-2
 - Reports Server の構成, 3-57
 - Reports Server の再起動, 2-2
 - Reports Server の停止, 2-2
 - Reports Services へのナビゲート, 19-2
 - 起動, 19-2
 - パフォーマンス分析, 20-2
- Oracle HTTP Server, 1-4, 10-11
- Oracle Internet Directory, 10-11, 12-2
 - rwservlet の接続パラメータの指定, 3-52
 - 接続エンティティの選択, 11-13
 - 接続パラメータの指定
 - rwserver, 3-15
- Oracle JDBC OCI (Thick) ドライバ, 9-6
- Oracle JDBC Thin ドライバ, 9-6
- Oracle Login Server, 10-11
- Oracle Process Manager and Notification Server,
 - 「OPMN」を参照
- Oracle Reports ブリッジ
 - OPMN による制御, 3-62
 - rwbridge, A-17
 - 構成, 3-40
 - トラブルシューティング, D-21
- Oracle Technology Network, 12-1
- Oracle Workflow、統合, 3-65

- ORACLE_AFM, B-8
- ORACLE_HOME, B-8
- ORACLE_HPDP, B-9
- ORACLE_PATH, B-9
- ORACLE_PPD, B-9
- ORACLE_TFM, B-10
- OracleAS Forms Services
 - クラスタ名の移行, 3-54
 - セキュリティに関する考慮事項, 11-14
 - レポートのコール, 20-23
- OracleAS Portal, 12-1 ~ 12-25
 - destination, 3-16, 8-2
 - destination 要素、例, 8-6
 - oraclePortal 宛先タイプ, 8-5
 - RW_ADMINISTRATOR, 12-2
 - RW_BASIC_USER, 12-2
 - RW_DEVELOPER, 12-2
 - RW_POWER_USER, 12-2
 - WebDAV, A-36
 - 概要, 10-12
 - サーバー・アクセス, 12-4
 - 使用可能日カレンダー
 - 結合, 12-15
 - シンプル, 12-14
 - トラブルシューティング, D-21
 - レポートのバッチ登録, C-1 ~ C-8
 - 配布の制限, 15-27
 - 配布の例, 15-25
 - プリンタ・アクセス, 12-11
 - ユーザーとグループ, 12-2
 - ランタイム・パラメータ・フォーム, 12-10
 - レポート・アクセス, 12-7
 - レポートの実行, 12-22
 - レポート・ポートレットの公開, 13-13
 - レポート・リクエスト, 13-5
- OracleAS Portal へのレポートのバッチ登録, C-1 ~ C-8
- OracleAS Reports Services
 - Single Sign-On, 3-54, 10-13, 11-1 ~ 11-15
 - 永続性, 1-2
 - 概要, 1-2
 - コンポーネント, 1-4
- OracleAS Single Sign-On
 - rwservlet.properties, 3-54
 - 機能, 10-13
 - 構成, 11-1 ~ 11-15
- ORACLEPORTAL 宛先, 15-27, A-39
- oraclePortal 宛先タイプ, 8-5
- ORAINFONAV_DOCPATH, B-10
- ORB
 - JDK, E-2
 - VisiBroker, E-2
- orbClient 構成要素, 3-25
 - id 属性, 3-25
 - publicKeyFile 属性, 3-25
- ORBPorts 構成要素, 3-26
 - value 属性, 3-26
- ORDER BY
 - ブレーク・グループ, 20-17
- organization 属性
 - mail, 15-7
- orientation コマンド・キーワード, A-61
- OTN、「Oracle Technology Network」を参照

- outputfolder コマンド・キーワード, A-62
- outputimageformat コマンド・キーワード, A-63
- outputpage コマンド・キーワード, A-64
- overwrite コマンド・キーワード, A-65

P

- p_availability コマンド・キーワード, A-65
- p_description コマンド・キーワード, A-65
- p_formats コマンド・キーワード, A-66
- p_jdbcpsds コマンド・キーワード, A-66
- p_name コマンド・キーワード, A-67
- p_owner コマンド・キーワード, A-67
- p_pformtemplate コマンド・キーワード, A-68
- p_printers コマンド・キーワード, A-68
- p_privilege コマンド・キーワード, A-69
- p_servers コマンド・キーワード, A-69
- p_trigger コマンド・キーワード, A-69
- p_types コマンド・キーワード, A-70
- pagegroup コマンド・キーワード, A-70
- pagesize コマンド・キーワード, A-71
- pagestream コマンド・キーワード, A-72
- paramform コマンド・キーワード, 20-24, A-72
- parsequery コマンド・キーワード, A-73
- PASTA ユーティリティ, 5-16
- PATH, 2-4
- PCL フォント
 - 追加, 4-8
- pdfcomp コマンド・キーワード, 20-24, A-73
- pdfembed コマンド・キーワード, A-74
- PDF 出力
 - アクセシビリティ, 6-12
 - 圧縮, 6-2
 - パフォーマンスのためのチューニング, 20-24
 - フォント関連の機能, 6-2
 - フォント機能の優先順位, 6-11
 - 分類, 6-12
- PDF 出力でのフォント埋込み, 6-9
- PDF 出力でのフォント・エイリアシング, 6-3
- PDF 出力でのフォント・サブセット, 6-5
- persistFile 構成要素, 3-3, 3-27
 - fileName 属性, 3-28
- pfaction コマンド・キーワード, A-74
- PFA ファイル, 4-23
- PFB ファイル, 4-23
- PFM ファイル, 4-23
- ping interval, 3-65
- ping timeout, 3-65
- PL/SQL
 - SRW_PARAMETER, 17-2
 - SRW_PARAMLIST, 17-2, 17-4, 17-8
 - srw_test.sql, 17-6
 - SRW.ADD_PARAMETER, 17-2
 - srwAPIIdrop.sql, 17-2
 - srwAPIgrant.sql, 17-2
 - srwAPIlins.sql, 17-2
 - SRW.APPLY_DEFINITION, 16-2
 - SRW.CANCEL_REPORT, 17-5
 - SRW.CLEAR_PARAMETER_LIST, 17-3
 - SRW.DO_SQL, 20-10
 - SRW.JOB_IDENT, 17-4
 - SRW.MAXROW_UNSET, 20-17
 - SRW.REMOVE_PARAMETER, 17-3

- SRW.REPORT_STATUS, 17-4
- SRW.SET_MAXROW(), 20-17
- SRW.SET_TEXT_COLOR, 20-20
- SRW.START_DEBUGGING, 17-6
- SRW.STATUS_RECORD, 17-4
- SRW.STOP_DEBUGGING, 17-6
- SRW パッケージ, 17-2
- ストアド・プロシージャ, 20-10
- パフォーマンス分析, 20-10
- ブロックの翻訳, 18-11
- PL/SQL、拡張配布, 15-3
- Pluggable
 - cache, 3-9
 - destinations, 3-16, 8-2, 15-15
 - engine, 3-10
 - notification, 3-19
 - repository, 3-21
 - security, 3-14
 - イベント, 3-18
- pluginParam
 - notification とともに使用, 3-20
- pluginParam 構成要素, 3-30, 15-7
 - destination とともに使用, 3-17
 - name 属性, 3-31
 - type 属性, 3-31
- Portal
 - OracleAS Portal を参照, 3-16
- PostScript
 - プリンタ・ドライバ, 3-67
- PostScript フォント
 - 追加, 4-7
- PostScript プリンタ定義 (PPD) ファイル, 4-10
- PPD ファイル, 4-10
- PRINTER, B-10
- printer 宛先タイプ, 8-5
- printer 配布要素, 15-14
 - copies 属性, 15-15
 - id 属性, 15-14
 - instance 属性, 15-15
 - name 属性, 15-15
- printjob コマンド・キーワード, A-75
- priority 属性
 - mail, 15-7
- property、JDBC 構成ファイル, 9-5
- property 配布要素, 15-17
- publicKeyFile 属性, 3-25

Q

- Queue Manager, 1-2, 13-14
- queue 構成要素
 - maxQueueSize 属性, 3-27, 20-12, 20-24
- queue 要素, 3-27

R

- readonly コマンド・キーワード, A-75
- recursive_load コマンド・キーワード, 20-24, A-76
- RELOAD_KEYMAP、rwservlet.properties, 3-48
- REMOTE, B-11
- remoteBridges 構成要素, 3-43
- remoteBridge 構成要素, 3-43
- REP-50125, D-10

- REP-52266, D-4
- REP-56048, D-12
- replyto コマンド・キーワード, A-77
- replyTo 属性
 - mail, 15-7
- Reports CGI, 1-5
- Reports Engine, 1-5, 1-7
 - diagnosis プロパティ, 3-13
- Reports JSP, 1-5
- Reports Queue Manager, 1-2, 13-14
- Reports Queue Viewer, 1-2
- Reports Server
 - OID 接続パラメータの指定, 3-15
 - Oracle Enterprise Manager からの起動, 2-2
 - Oracle Enterprise Manager からの再起動, 2-2
 - Oracle Enterprise Manager からの停止, 2-2
 - Windows サービスとして実行, 2-2, 2-3
 - アクセス制御, 12-4
 - 宛先処理, 8-3
 - 宛先タイプの登録, 8-4
 - インプロセス, 1-2, 1-5, 3-52
 - インプロセス、OPMN による制御, 3-61
 - 永続性, 3-27, 13-15
 - クラスタリング (非推奨), 3-25, 3-54
 - 検出, 1-8
 - 検出メカニズム, 3-36
 - 検出ユーティリティ, E-1
 - 構成のチューニング, 20-11
 - 構成ファイル, 1-7, 3-2
 - コマンドラインからの起動, 2-4
 - サブレットとしての起動, 2-4
 - スタンドアロン、OPMN による制御, 3-58
 - ステータス・レコード, 17-5
- Reports Services
 - cache API, 3-9
 - Destinations API, 3-16, 8-2
 - engine API, 3-10
 - notification API, 3-19
 - repository API, 3-21
 - security API, 3-14
 - Single Sign-On, 3-54, 10-13, 11-1 ~ 11-15
 - イベント API, 3-18
 - 永続性, 1-2
 - 監視, 19-1 ~ 19-7
 - 管理, 19-1 ~ 19-7
 - 概要, 1-2
 - 起動と停止, 2-1
 - コンポーネント, 1-4
- Reports Services の監視, 19-1 ~ 19-7
- Reports Services の管理, 19-1 ~ 19-7
- Reports Servlet, 1-5, 3-3, 10-11
 - OID 接続パラメータの指定, 3-52
 - URL 構文, 13-2
 - カスタム・ヘルプの追加, 3-54
 - 構成, 3-45
- REPORTS_ADD_HWMARGIN, B-11
- REPORTS_ARABIC_NUMERAL, 6-19, B-12
- REPORTS_BIDI_ALGORITHM, 6-19, B-12
- REPORTS_CGIDIAGBODYTAGS, B-12
- REPORTS_CGIDIAGHEADTAGS, B-13
- REPORTS_CGIHELP, B-13
- REPORTS_CGIMAP, B-14
- REPORTS_CGINODIAG, B-14

REPORTS_CLASSPATH, B-15
 REPORTS_CONTAINSHTMLTAGS, B-15
 REPORTS_COOKIE_EXPIRE, B-16
 REPORTS_DB_AUTH, B-17
 REPORTS_DEFAULT_DISPLAY, B-17
 REPORTS_DEFAULT_PIXEL_SIZE, B-18
 REPORTS_ENABLE_RTF_SPACING, B-18
 REPORTS_ENCRYPTION_KEY, B-19
 REPORTS_ENHANCED_SUBSET, 6-7, B-19
 REPORTS_GRAPH_IMAGE_DPI, B-19
 REPORTS_IGNORE_IMAGE_TAG_RES, B-20
 REPORTS_JPEG_QUALITY_FACTOR, B-20
 REPORTS_JVM_OPTIONS, B-20
 REPORTS_NETWORK_CONFIG, B-21
 REPORTS_NETWORK_CONFIG、
 rwservlet.properties, 3-54
 REPORTS-NLS_XML_CHARSETS, B-21
 REPORTS_NO_DUMMY_PRINTER, B-22
 REPORTS_NO_HTML_SPACE_REPLACE, B-22
 REPORTS_OUTPUTIMAGEFORMA, B-23
 REPORTS_PATH, B-23
 REPORTS_RESOURCE, B-24
 REPORTS_SERVER, B-24
 REPORTS_SERVERMAP、rwservlet.properties, 3-54
 REPORTS_SOLARIS_9, B-25
 REPORTS_SPACE_BREAK, B-25
 REPORTS_SRWRUN_TO_SERVER, B-25
 REPORTS_SSLPORT, B-26
 REPORTS_SYS_AUTH, B-26
 REPORTS_TAGLIB_URI, B-26
 REPORTS_TMP, B-27
 REPORTS_USEREXITS, B-27
 REPORTS_UTF8_XMLOUTPUT, B-28
 report コマンド・キーワード, A-57
 report パラメータ、イベント, 17-4, 17-7
 repositoryconn プロパティ
 confidential 属性, 20-8
 returnReceipt 属性
 mail, 15-7
 role コマンド・キーワード, A-77
 rundebug コマンド・キーワード, 20-23, A-78
 RW_ADMINISTRATOR, 12-2
 RW_BASIC_USER, 12-2
 RW_DEVELOPER, 12-2
 RW_POWER_USER, 12-2
 RW_SERVER_JOB_QUEUE 表, 20-5
 RW_SERVER_QUEUE ビュー, 20-8
 rw_server.sql, 20-8
 rwbridge コマンド, A-17
 rwbuilder.conf, 3-2
 rwbuilder コマンド, 16-18, A-10
 rwcgi コマンド, A-14
 rwclient, 13-5
 rwclient コマンド, A-7
 rwconverter
 バッチ登録用 SQL スクリプトの生成, C-2
 rwconverter コマンド, 16-13, A-10
 rwdiag, 20-13, E-2
 構文, E-3
 使用上の注意, E-4
 例, E-2
 rwnetworkconf.dtd, 3-37
 discoveryService 要素, 3-38
 multicast 要素, 3-38
 namingService 要素, 3-39
 rwproxy, 3-8
 rwrund コマンド, A-8
 rwserver, 2-5
 server, 2-4
 rwserverconf.dtd, 3-2, 3-4, 8-4, 8-5
 cache 要素, 3-9
 compatible 構成要素, 3-8
 connection 要素, 3-24
 destination 要素, 3-16, 3-17
 engine 要素, 3-10
 environment 要素, 3-31
 identifier 要素, 3-29
 jobRecovery 要素, 3-28
 jobStatusRepository 要素, 3-21
 job 要素, 3-18
 log 要素, 3-20
 notification 要素, 3-19
 oidconnection 要素, 3-15
 orbClient 要素, 3-25
 ORBPorts 要素, 3-26
 persistFile 要素, 3-27
 pluginParam 要素, 3-30
 queue 要素, 3-27
 security 要素, 3-13
 server 要素, 3-7
 trace 要素, 3-22
 rwserver.template, 3-3
 rwserver コマンド, A-17
 rwservlet, 1-5, 2-4
 Web サービスとして公開, 13-14
 rwservlet.properties, 3-3
 ALLOWHTMLTAGS, 3-50
 CONNECTION_POOLSIZE, 3-51
 COOKIEEXPIRE, 3-53
 DBAUTH, 3-49
 DEFAULTCHARSET, 3-50
 DIAGBODYTAGS, 3-53
 DIAGHEADTAGS, 3-53
 DIAGNOSTIC, 3-48
 ENCRYPTIONKEY, 3-53
 ERRORTEMPLATE, 3-51
 HELP, 3-54
 IMAGE_URL, 3-53
 KEYMAPFILE, 3-47
 OIDCON_INCREMENT, 3-52
 OIDCON_INIT, 3-52
 OIDCON_TIMEOUT, 3-52
 RELOAD_KEYMAP, 3-48
 REPORTS_NETWORK_CONFIG, 3-54
 REPORTS_SERVERMAP, 3-54
 SERVER, 3-52
 SERVER_IN_PROCESS, 3-52
 SINGLESIGNON, 3-54, 11-3
 SYSAUTH, 3-49
 TRACEFILE, 3-49
 TRACEMODE, 3-49
 TRACEOPTS, 3-49
 rwservlet コマンド, A-11
 RW 環境変数, B-28

S

save_rdf コマンド・キーワード, A-78
schedule コマンド・キーワード, 13-14, A-79
screenprinter.ppd, 3-67, 4-9
Secure Sockets Layer, 1-4
securityId 属性, 3-19, 11-4
security 構成要素, 3-13, 11-4
 class 属性, 3-14
 id 属性, 3-14
SERVER、rwservlet.properties, 3-52
SERVER_IN_PROCESS、rwservlet.properties, 3-52
server_name.conf, 3-2
SERVEROUT, 17-6
server 構成要素, 3-7
server コマンド・キーワード, 2-4, 2-5, A-80
server パラメータ、イベント, 17-4
showauth コマンド・キーワード, A-80
showenv コマンド・キーワード, A-81
showjobid コマンド・キーワード, A-81
showjobs コマンド・キーワード, 20-8, A-82
showmap コマンド・キーワード, A-82
showmyjobs コマンド・キーワード, A-83
shutdown コマンド・キーワード, 2-5, A-83
Single Sign-On
 rwservlet.properties, 3-54
 機能, 10-13
 構成, 11-1 ~ 11-15
SINGLESIGNON, 11-3
SINGLESIGNON、rwservlet.properties, 3-54
sitename コマンド・キーワード, A-84
SMTP, 3-20, 15-6, 15-7
sourceDatabase, 9-4
source コマンド・キーワード, A-84
SQL
 パフォーマンス分析, 20-8
SQL Server ドライバ, 9-9
SQL*PLUS, 17-6
SQLNet, A-95
SQL トレース, 20-8
srcType 属性
 attach, 15-9
 body, 15-8
src 属性
 include, 15-12
SRW_PARAMETER, 17-2
SRW_PARAMLIST, 17-2, 17-4, 17-8
SRW_ParamList, 17-2
SRW_ParamList_Object, 17-2
srw_test.sql, 17-6
SRW.ADD_PARAMETER, 17-2
srwAPIdrop.sql, 17-2
srwAPIgrant.sql, 17-2
srwAPIlins.sql, 17-2
SRW.APPLY_DEFINITION, 16-2
SRW.CANCEL_REPORT, 17-5
SRW.CLEAR_PARAMETER_LIST, 17-3
SRW.DO_SQL, 20-10
SRW.JOB_IDENT, 17-4
SRW.REMOVE_PARAMETER, 17-3
SRW.REPORT_STATUS, 17-4
SRW.RUN_REPORT, 13-5
SRW.SET_MAXROW, 20-17

SRW.SET_PRINTER_TRAY, 5-23
SRW.SET_TEXT_COLOR, 20-20
SRW.START_DEBUGGING, 17-6
SRW.STATUS_RECORD, 17-4
SRW.STOP_DEBUGGING, 17-6
SRW パッケージ, 17-2
SSL, 1-4
SSO
 rwservlet.properties, 3-54
 機能, 10-13
 構成, 11-1 ~ 11-15
ssoconn コマンド・キーワード, 11-5, A-86
statusfolder コマンド・キーワード, A-87
statusformat コマンド・キーワード, A-88
statuspage コマンド・キーワード, A-88
stype コマンド・キーワード, A-89
subject コマンド・キーワード, A-89
subject 属性
 mail, 15-7
subProtocol, 9-4
Sybase ドライバ, 9-7
SYSAUTH、rwservlet.properties, 3-49
Sysout 宛先, A-39

T

TFM ファイル, 4-10
timeout 属性
 oidconnection, 3-15
TK_AFM, B-30
TK_HPDP, B-30
TK_PPD, B-31
TK_PRINT, B-28
TK_PRINT_STATUS, B-29
TK_PRINTER, B-29
TK_TFM, B-31
Tk2Motif.rgb, 13-12
Tk2Motif.rgb ファイル, 4-11
TKPROF、SQL トレース, 20-8
tnsnames.ora, 3-8
tolerance コマンド・キーワード, 1-6, 13-15, A-90
to 属性
 mail, 15-6
trace_all, 3-24
trace_all (ブリッジ), 3-45
trace_app, 3-23
trace_brk, 3-23
trace_dbg, 3-24
trace_dbg (ブリッジ), 3-45
trace_dst, 3-24
trace_err, 3-24
trace_err (ブリッジ), 3-45
trace_exc, 3-24
trace_inf, 3-24
trace_inf (ブリッジ), 3-45
trace_log, 3-24
trace_pls, 3-23
trace_prf, 3-23
trace_sql, 3-23
trace_sta, 3-24
trace_tms, 3-23
trace_wrn, 3-24
TRACEFILE、rwservlet.properties, 3-49

tracefile コマンド・キーワード, A-91
traceFile 属性, 3-23, 20-3
traceFile 属性 (ブリッジ), 3-45
TRACEMODE、rwservlet.properties, 3-49
tracemode コマンド・キーワード, A-92
traceMode 属性, 3-23, 20-3
traceMode 属性 (ブリッジ), 3-45
TRACEOPTS、rwservlet.properties, 3-49
traceopts コマンド・キーワード, A-92
traceOpts 属性, 3-23, 20-3
traceOpts 属性 (ブリッジ), 3-45
trace 構成要素, 3-22
 traceFile 属性, 3-23, 20-3
 traceFile 属性 (ブリッジ), 3-45
 traceMode 属性, 3-23, 20-3
 traceMode 属性 (ブリッジ), 3-45
 traceOpts 属性, 3-23, 20-3
 traceOpts 属性 (ブリッジ), 3-45
trace 構成要素 (ブリッジ), 3-44
TrueType Big Font, 4-19, 18-10
TrueType Collection (TTC), 4-24
TrueType フォント, 4-24
TTC (TrueType Collection), 4-24
Type1 フォント, 4-23
type 属性, 3-31

U

uifont.ali, 4-9, 5-11
 検証, 4-16
 セクション, 4-14
 場所, 4-14
uiprint.txt, 4-9, 5-5
Unicode, 4-23, 18-9 ~ 18-11
UNIX、印刷, 3-67, 5-1, 5-25
UNIX での印刷, 3-67, 5-1, 5-25
upgrade_plsql コマンド・キーワード, A-94
urlparameter コマンド・キーワード, A-94
URL エンジン
 構成, 3-55
 要素, 3-55
 レポート・リクエスト, 13-13
URL ジョブ・リクエスト, 13-1 ~ 13-17
URL、ランタイム構文, 13-2
usejvm コマンド・キーワード, A-94
USER_NLS_LANG, 18-6, B-32
userid コマンド・キーワード, A-95
userid パラメータ、イベント, 17-4, 17-7
USERNAME, B-31
userstyles コマンド・キーワード, A-96
UTF8, 18-10

V

VALIDATETAG, 20-23, A-97
validatetag コマンド・キーワード, 20-23, A-96
value 属性, 3-26, 3-32
version 属性, 3-8
VisiBroker, 3-36, E-2

W

WE8MSWIN1252 キャラクタ・セット, 13-12

WebDAV
 DESNAME, A-36
 配布の例, 15-25
WebDAV 宛先, A-39
webserver_debug コマンド・キーワード, A-97
webserver_docroot コマンド・キーワード, A-98
webserver_port コマンド・キーワード, A-98
Web サービス, 14-1
 rwservlet の公開, 13-14
 WSDL の表示, 14-3
 操作, 14-5
Web ブラウザ
 レポート・リクエスト, 13-13
Web リスナー, 1-4, 10-11
Windows サービス, 2-2, 2-3
Windows のレジストリ, 18-4

X

XML
 1 つの定義の適用, 16-13
 distribution.dtd, 15-2
 JDBC 構成ファイル, 9-2
 PL/SQL による適用, 16-14
 Reports Builder でのオープン, 16-18
 解析, 16-2
 拡張配布, 15-1 ~ 15-28
 カスタマイズの作成, 16-2, 16-3 ~ 16-7
 カスタマイズの適用, 16-2
 カスタマイズのデバッグ, 16-17 ~ 16-19
 カスタマイズのトレース・オプション, 16-18
 クロス積グループの作成, 16-10
 グループ階層の作成, 16-9
 サマリーの作成, 16-11
 式の作成, 16-11
 書式マスクの変更, 16-4
 書式例外の追加, 16-5
 新規問合せの追加, 16-7
 実行時の適用, 16-13
 スタイルの変更, 16-4
 単独での実行, 16-17
 適用, 16-13 ~ 16-17
 データソースのリンク, 16-9
 データの検証, 20-15
 データ・モデルの作成, 16-8 ~ 16-12
 デバッグ, 16-19
 ハイパーリンクの追加, 16-6
 配布 XML ファイルの使用, 15-26
 配布の制限, 15-27
 バッチ変更, 16-17
 パーサーのエラー・メッセージ, 16-18
 パラメータの作成, 16-12
 必須カスタマイズ・タグ, 16-3
 複数の定義の適用, 16-14
 複数のデータソースの作成, 16-8
 ブレースホルダの作成, 16-11
 プログラム・ユニットの追加, 16-6
 マトリックス・グループの作成, 16-10
 レポートのカスタマイズ, 16-1 ~ 16-19
XML 属性、変数の使用, 15-3

あ

アーキテクチャ
 グローバル化・サポート, 18-2
アクセス制御, 12-4
 サーバー, 12-4
 使用可能日カレンダー、結合, 12-15
 使用可能日カレンダー、シンプル, 12-14
 プリンタ, 12-11
 レポート, 12-7
宛先タイプ, 3-16, 8-4
アドバンスド・キューイング, 17-2, 17-7, 17-8, 17-9
 dbms_AQadm パッケージ, 17-8
 dbms_aq.dequeue, 17-9
 DEQUEUE, 17-7
 ENQUEUE, 17-7
 MESSAGES, 17-7
アプリケーション・エンティティのマージ, 11-13
アプリケーションの翻訳, 18-11 ~ 18-12
暗号化, 8-5
一括処理, 15-4
 配布, 20-22
イベント
 authid パラメータ, 17-4
 cmdkey パラメータ, 17-7
 dist パラメータ, 17-7
 gateway パラメータ, 17-4
 MyIdent.AuthID, 17-4
 MyIdent.GatewayURL, 17-4
 MyIdent.JobID, 17-4
 MyIdent.ServerName, 17-4
 report パラメータ, 17-4, 17-7
 server パラメータ, 17-4
 SRW_PARAMETER, 17-2
 SRW_PARAMLIST, 17-2, 17-4, 17-8
 SRW_ParamList, 17-2
 SRW_ParamList_Object, 17-2
 srw_test.sql, 17-6
 SRW.ADD_PARAMETER, 17-2
 srwAPIdrop.sql, 17-2
 srwAPIgrant.sql, 17-2
 srwAPIlins.sql, 17-2
 SRW.CANCEL_REPORT, 17-5
 SRW.CLEAR_PARAMETER_LIST, 17-3
 SRW.JOB_IDENT, 17-4
 SRW.REMOVE_PARAMETER, 17-3
 SRW.REPORT_STATUS, 17-4
 SRW.START_DEBUGGING, 17-6
 SRW.STATUS_RECORD, 17-4
 SRW.STOP_DEBUGGING, 17-6
 SRW パッケージ, 17-2
 userid パラメータ, 17-4, 17-7
 エンキュー・プロシージャの作成, 17-8
 ジョブの取消し, 17-5
 デキュー・プロシージャの作成, 17-9
 デバッグ, 17-6
 メッセージ・キューの作成, 17-8
 レポートの起動, 17-6
イベント・ドリブンの公開, 1-7, 17-1 ~ 17-10
イメージ、サポート, 3-68
印刷
 トラブルシューティング, D-21
インプロセス Reports Server, 3-52

OPMN による制御, 3-61
インプロセス・サーバー, 1-2, 1-5
 デフォルト・プリンタの設定, D-19
ウィザード
 用語集, 用語集-5
永続性, 1-2, 3-27, 13-15
エラー・メッセージ, 3-48
エラー・メッセージ、XML, 16-18
エンキュー、プロシージャの作成, 17-8
エンジンの診断, 3-13
大文字と小文字の区別, 2-9

か

カスタム・ドライバ, 9-11
環境切替え, 3-32
環境変数
 CA_GPPREFS, B-5
 CA_UPREFS, B-5
 DELIMITED_LINE_END, B-5
 DEVELOPER-NLS_LANG, 18-6, B-6
 DOC, B-6
 NLS_CALENDAR, B-6
 NLS_CREDIT, B-6
 NLS_CURRENCY, B-6
 NLS_DATE_FORMAT, B-6
 NLS_DATE_LANGUAGE, B-6
 NLS_DEBIT, B-6
 NLS_ISO_CURRENCY, B-6
 NLS_LANG, 18-3, B-7
 NLS_LIST_SEPARATOR, B-8
 NLS_MONETARY_CHARACTERS, B-8
 NLS_NUMERIC_CHARACTERS, B-8
 NLS_SORT, B-8
 ORACLE_AFM, B-8
 ORACLE_HOME, B-8
 ORACLE_HPDI, B-9
 ORACLE_PATH, B-9
 ORACLE_PPD, B-9
 ORACLE_TFM, B-10
 ORAINFONAV_DOC_PATH, B-10
 PATH, 2-4
 PRINTER, B-10
 REMOTE, B-11
 REPORTS_ADD_HWMARGIN, B-11
 REPORTS_ARABIC_NUMERAL, 6-19, B-12
 REPORTS_BIDI_ALGORITHM, 6-19, B-12
 REPORTS_CGDIAGBODYTAGS, B-12
 REPORTS_CGDIAGHEADTAGS, B-13
 REPORTS_CGIIHELP, B-13
 REPORTS_CGIMAP, B-14
 REPORTS_CGINODIAG, B-14
 REPORTS_CLASSPATH, B-15
 REPORTS_CONTAINSHTMLTAGS, B-15
 REPORTS_COOKIE_EXPIRE, B-16
 REPORTS_DB_AUTH, B-17
 REPORTS_DEFAULT_DISPLAY, B-17
 REPORTS_DEFAULT_PIXEL_SIZE, B-18
 REPORTS_ENABLE_RTF_SPACING, B-18
 REPORTS_ENCRYPTION_KEY, B-19
 REPORTS_ENHANCED_SUBSET, 6-7, B-19
 REPORTS_GRAPH_IMAGE_DPI, B-19
 REPORTS_IGNORE_IMAGE_TAG_RES, B-20

REPORTS_JPEG_QUALITY_FACTOR, B-20
 REPORTS_JVM_OPTIONS, B-20
 REPORTS_NETWORK_CONFIG, B-21
 REPORTS-NLS_XML_CHARSETS, B-21
 REPORTS_NO_DUMMY_PRINTER, B-22
 REPORTS_NO_HTML_SPACE_REPLACE, B-22
 REPORTS_OUTPUTIMAGEFORMAT, B-23
 REPORTS_PATH, B-23
 REPORTS_RESOURCE, B-24
 REPORTS_SERVER, B-24
 REPORTS_SOLARIS_9, B-25
 REPORTS_SPACE_BREAK, B-25
 REPORTS_SRWRUN_TO_SERVER, B-25
 REPORTS_SSLPORT, B-26
 REPORTS_SYS_AUTH, B-26
 REPORTS_TAGLIB_URI, B-26
 REPORTS_TMP, B-27
 REPORTS_USEREXITS, B-27
 REPORTS_UTF8_XMLOUTPUT, B-28
 RW, B-28
 TK_AFM, B-30
 TK_HPDP, B-30
 TK_PPD, B-31
 TK_PRINT, B-28
 TK_PRINT_STATUS, B-29
 TK_PRINTER, B-29
 TK_TFM, B-31
 USER-NLS_LANG, 18-6, B-32
 USERNAME, B-31
 環境変数、グローバル化・サポート, 18-2 ~ 18-6
 環境変数、編集, B-1
 キー・マップ・ファイル, 13-16 ~ 13-17
 URL パラメータのマッピング, 13-16
 位置の指定, 3-47
 エントリの追加, 13-16
 再ロード, 3-48
 使用可能にする方法, 13-16
 使用方法, 13-17
 パラメータ・フォームを使用した制限付きの実行, 13-17
 利点, 13-16
 キーワード
 accessible, A-18
 arraysize, A-18
 authid, A-19
 autocommit, A-20
 background, A-20
 batch, A-21
 bcc, A-21
 blankpages, A-22
 buffers, A-22
 cachelob, A-23
 cc, A-23
 cellwrapper, A-24
 cmdfile, A-25
 cmdkey, A-26
 collate, A-26
 containshtmltags, A-27
 containsole, A-28
 contentarea, A-28
 copies, A-29
 customize, 16-1, A-29
 dateformatmask, A-30
 delauth, A-31
 delimited_hdr, A-31
 delimiter, A-32
 desformat, A-33
 desname, A-35
 dest, A-37
 destination, A-37
 destype, A-38
 distribute, A-44
 dtype, A-44
 dunit, A-45
 engine_responsetimeout, A-46
 envid, A-46
 expiration, A-46
 expiredays, A-47
 express_server, A-48
 formsize, A-49
 from, A-50
 getjobid, A-50
 getserverinfo, A-51
 help, A-51
 itemtitle, A-52
 jobname, A-52
 jobtype, A-53
 jvmoptions, A-53
 killengine, A-54
 killjobid, A-55
 longchunk, A-55
 mimetype, A-56
 mode, A-56
 module, A-57
 name, A-57
 nonblocksql, A-58
 notifyfailure, A-58
 notifysuccess, A-59
 numberformatmask, A-59
 olap_con, A-60
 onfailure, A-60
 onsuccess, A-61
 orientation, A-61
 outputfolder, A-62
 outputimageformat, A-63
 outputpage, A-64
 overwrite, A-65
 p_availability, A-65
 p_description, A-65
 p_formats, A-66
 p_jdbcpsds, A-66
 p_name, A-67
 p_owner, A-67
 p_pformtemplate, A-68
 p_printers, A-68
 p_privilege, A-69
 p_servers, A-69
 p_trigger, A-69
 p_types, A-70
 pagegroup, A-70
 pagesize, A-71
 pagestream, A-72
 paramform, A-72
 parsequery, A-73
 pdfcomp, A-73

- pdfembed, A-74
- pfaction, A-74
- printjob, A-75
- readonly, A-75
- recursive_load, A-76
- replyto, A-77
- report, A-57
- role, A-77
- rundebug, A-78
- save_rdf, A-78
- schedule, 13-14, A-79
- server, A-80
- showauth, A-80
- showenv, A-81
- showjobid, A-81
- showjobs, A-82
- showmap, A-82
- showmyjobs, A-83
- shutdown, A-83
- sitename, A-84
- source, A-84
- ssoconn, A-86
- statusfolder, A-87
- statusformat, A-88
- statuspage, A-88
- stype, A-89
- subject, A-89
- tolerance, 13-15, A-90
- tracefile, A-91
- tracemode, A-92
- traceopts, A-92
- upgrade_plsql, A-94
- urlparameter, A-94
- usejvm, A-94
- userid, A-95
- userstyles, A-96
- validatetag, A-96
- webserver_debug, A-97
- webserver_docroot, A-98
- webserver_port, A-98
- 北アフリカ圏の言語, 18-8
- キャッシュ, 13-15
- キャッシュ宛先, A-38
- キャッシュ・キー, 13-15
- キャラクタ・セット, 4-23, 18-1, 18-4, 18-6 ~ 18-8
 - Unicode, 18-9 ~ 18-11
 - UTF8, 18-10
 - 設計上の考慮事項, 18-4
 - フォントのエイリアシング, 18-5
- キュー・ビューア, 1-2
- クラスタリング、Reports Server (非推奨), 3-25, 3-54
- 「繰返し」プロパティ, 20-22
- クロス・プラットフォームの移植, 7-1 ~ 7-23
- グラフ・ウィザード
 - 用語集, 用語集-5
- グループ・フィルタ, 20-18
 - WHERE 句, 20-18
- グローバリゼーション
 - トラブルシューティング, D-21
- グローバリゼーション・サポート, 18-1 ~ 18-15
 - DEVELOPER-NLS_LANG, 18-6
 - JSP, 18-6 ~ 18-8
 - NLS_LANG, 18-3
 - Unicode, 18-9 ~ 18-11
 - USER-NLS_LANG, 18-6
 - アーキテクチャ, 18-2
 - アプリケーションの翻訳, 18-11 ~ 18-12
 - 環境変数, 18-2 ~ 18-6
 - キャラクタ・セット, 18-4
 - 言語依存データ, 18-2
 - 言語に依存しない機能, 18-2
 - 双方向サポート, 18-8 ~ 18-9
 - トラブルシューティング, 18-12 ~ 18-15
 - フォントのサポート, 18-10
- 検索、ファイル, 4-11
- 検出
 - Reports Server のユーティリティ, E-1
- 検出メカニズム, 3-36, 3-54
- 言語
 - 北アフリカ圏, 18-8
 - 中東圏, 18-8
- 構成, 3-1 ~ 3-69
 - URL エンジン, 3-55
 - 考慮事項, 1-13
 - チューニング, 20-1 ~ 20-24
 - プロキシ情報, 3-56
- 構成、Reports Server, 1-7
- 構成ファイル
 - Reports Server, 3-2
 - rwserverconf.dtd, 3-4
 - rwserver.template, 3-3
 - rwervlet.properties, 3-3
 - フォント, 4-8
- 構成要素
 - bridge, 3-41
 - cache, 3-9, 13-15
 - compatible, 3-3, 3-8
 - connection, 3-24
 - destination, 3-16, 3-17, 8-4
 - discoveryService, 3-38
 - engine, 3-10
 - environment, 3-31
 - identifier, 3-29, 3-42
 - job, 3-18
 - jobRecovery, 3-28
 - jobStatusRepository, 3-21
 - log, 3-20
 - multicast, 3-38
 - namingService, 3-39
 - networkConfig, 3-42
 - notification, 3-19
 - oidconnection, 3-15
 - orbClient, 3-25
 - ORBPorts, 3-26
 - persistFile, 3-3, 3-27
 - pluginParam, 3-30, 15-7
 - queue, 3-27
 - remoteBridge, 3-43
 - remoteBridges, 3-43
 - security, 3-13
 - server, 3-7
 - trace, 3-22
 - trace (ブリッジ), 3-44
- 構文、Reports の URL, 13-2
- 構文、コマンド, A-2
- コールバックのタイムアウト, 3-66

コマンド

rwbridge, A-17
rwbuilder, 16-18, A-10
rwcgi, A-14
rwclient, A-7
rwconverter, 16-13, A-10
rwdiag, E-2
rwrund, A-8
rwservlet, A-17
rwservlet, A-11
概要, A-3
構文, A-2

コマンド・キーワード

accessible, A-18
arraysize, 20-24, A-18
authid, 2-5, A-19
autocommit, A-20
background, A-20
batch, 2-4, 2-5, A-21
bcc, A-21
blankpages, A-22
buffers, A-22
cachelob, A-23
cc, A-23
cellwrapper, A-24
cmdfile, A-25
cmdkey, A-26
collate, A-26
containshtmltags, A-27
containsole, A-28
contentarea, A-28
copies, 20-24, A-29
customize, 16-1, A-29
dateformatmask, A-30
delauth, A-31
delimited_hdr, A-31
delimiter, A-32
desformat, A-33
desname, A-35
dest, A-37
destination, A-37
destype, A-38
distribute, A-44
dtype, A-44
dunit, A-45
engineerresponsetimeout, A-46
envid, A-46
expiration, 1-6, 20-24, A-46
expiredays, A-47
express_server, A-48
formsize, A-49
from, A-50
getjobid, A-50
getserverinfo, A-51
help, A-51
itemtitle, A-52
jobname, A-52
jobtype, A-53
jvmoptions, A-53
killengine, A-54
killjobid, A-55
longchunk, 20-24, A-55
mimetype, A-56

mode, A-56
module, A-57
name, A-57
nonblocksq, A-58
notifyfailure, A-58
notifysuccess, A-59
numberformatmask, A-59
olap_con, A-60
onfailure, A-60
onsuccess, A-61
orientation, A-61
outputfolder, A-62
outputimageformat, A-63
outputpage, A-64
overwrite, A-65
p_availability, A-65
p_description, A-65
p_formats, A-66
p_jdbcpsds, A-66
p_name, A-67
p_owner, A-67
p_pformtemplate, A-68
p_printers, A-68
p_privilege, A-69
p_servers, A-69
p_trigger, A-69
p_types, A-70
pagegroup, A-70
pagesize, A-71
pagestream, A-72
paramform, 20-24, A-72
parsequery, A-73
pdfcomp, 20-24, A-73
pdfembed, A-74
pfaction, A-74
printjob, A-75
readonly, A-75
recursive_load, 20-24, A-76
replyto, A-77
report, A-57
role, A-77
rundebug, 20-23, A-78
save_rdf, A-78
schedule, 13-14, A-79
server, 2-4, 2-5, A-80
showauth, A-80
showenv, A-81
showjobid, A-81
showjobs, 20-8, A-82
showmap, A-82
showmyjobs, A-83
shutdown, 2-5, A-83
sitename, A-84
source, A-84
ssoconn, A-86
statusfolder, A-87
statusformat, A-88
statuspage, A-88
stype, A-89
subject, A-89
tolerance, 1-6, 13-15, A-90
tracefile, A-91
tracemode, A-92

traceopts, A-92
upgrade_plsql, A-94
urlparameter, A-94
usejvm, A-94
userid, A-95
userstyles, A-96
validatetag, 20-23, A-96
webserver_debug, A-97
webserver_docroot, A-98
webserver_port, A-98
コマンドライン、指定, 13-5

な

サーバー, 10-11
 rwbridge コマンド, A-17
 rwserver コマンド, A-17
 インプロセス, 1-2, 1-5, 3-52
 インプロセス、OPMN による制御, 3-61
サーバー・アクセス制御, 12-4
サブレット, 1-2, 1-5, 2-4, 3-3, 10-11
 rwservlet, A-11
 URL 構文, 13-2
出力
 UNIX, 3-67, 5-25
出力処理, 8-1 ~ 8-3
使用可能日カレンダー, 12-13 ~ 12-17
 結合, 12-15
 シンプル, 12-14
シングルバイト言語, 18-10
診断ログ
 Reports Engine, 3-13
事前フェッチ, 20-22
実行可能ファイル
 概要, A-3
ジョブ、実行, 13-1 ~ 13-17
ジョブの取消し, 17-5
スクリプト
 srw_test.sql, 17-6
 srwAPIdrop.sql, 17-2
 srwAPIgrant.sql, 17-2
 srwAPIins.sql, 17-2
スタンドアロンの Reports Server
 OPMN による制御, 3-58
ステータス・レコード, 17-5
スプレッドシート出力
 配布の制限, 15-27
セキュリティに関する考慮事項
 OracleAS Forms Services, 11-14
セッション Cookie, 11-2
セルの囲み文字, 20-15
 テキスト・データソース, 20-15
双方向サポート, 18-8, 18-9
ソート基準、言語, 18-5
属性、変数の使用, 15-3

た

タイムアウト設定
 rwdiag, 20-13
多言語テキストの表示, 18-10
中東圏の言語, 18-8
チューニング

レポートのパフォーマンス, 20-1 ~ 20-24
月名、言語, 18-5
テキストの表示、多言語, 18-10
テキストの読取り順序, 18-1
テンプレート、rwserver.template, 3-3
データ・ウィザード
 用語集, 用語集-5
データ型
 BLOB, 20-24
 CLOB, 20-24
 DATE, 18-3
 LONG, 20-24
 LONG RAW, 20-24
 NUMBER, 18-3
データ型ディクショナリ
 distribution.dtd, 15-2
 DTD, 20-15
 rwserverconf.dtd, 3-2, 3-3, 3-4
データソース
 XML によるグループ階層, 16-9
 XML による作成, 16-8
 XML によるリンク, 16-9
データ定義
 DTD スキーマ, 20-15
 XML スキーマ, 20-15
データベース・トリガー, 17-6
データベース認証 Cookie COOKIEEXPIRE、
 rwservlet.properties, 3-53
データベース認証 Cookie ENCRYPTIONKEY、
 rwservlet.properties, 3-53
データベースの索引
 SQL WHERE 句, 20-16
データ・モデル、作成, 16-8 ~ 16-12
デキュー、プロシージャの作成, 17-9
デバッグ・イベント, 17-6
デリミタ付き出力
 配布の制限, 15-27
統合、Oracle Workflow, 3-65
トラブルシューティング
 JDBC PDS, D-21
 Oracle Reports ブリッジ, D-21
 OracleAS Portal, D-21
 OracleAS Reports Services, D-1 ~ D-24
 印刷, D-21
 クロス・プラットフォームの問題, 7-6, 7-12, 7-16,
 7-20
 グローバル化、D-21
 グローバル化に関する問題, 18-12 ~ 18-15
 パフォーマンス, D-20
 フォント, D-20
 フォントの問題, 4-16 ~ 4-22
トリガー、データベース, 17-6
トレース
 Reports Servlet と JSP, 3-49
 XML レポート定義, 16-13, 16-18
 診断オプションとの比較, 3-13
 レポート, 20-3
トレーラ, 20-22
動的な環境切替え, 3-32
動的な検出, 1-8
同等の記号, 18-5
ドライバ, 9-6
ドライバ、カスタム, 9-11

な

名前 / 値のペア、宛先, 8-5
認証 Cookie, 3-53
ネーミング・サービス, 3-36
ネーミング・サービス (COS), 1-8
ネットワーク
 タイムアウトのチューニング, 20-13

は

ハイパーリンク、XML による追加, 16-6
配布, 15-1 ~ 15-28
 OracleAS Portal, 15-27
 XML 出力, 15-27
 一括処理, 15-4, 20-22
 スプレッドシート出力, 15-27
 制限, 15-27
 デリミタ付き出力, 15-27
 動的フォーマット属性値, 15-28
配布、XML ファイルの使用, 15-26
配布の概要, 15-2
配布の例, 15-17 ~ 15-26
 destype, 15-25
 file, 15-21
 foreach, 15-17
 mail, 15-19
 printer, 15-23
「配布」プロパティ, 20-22
配布要素
 attach, 15-9
 body, 15-7
 destinations, 15-4
 destype, 15-15
 file, 15-12
 foreach, 15-4
 include, 15-10
 mail, 15-6
 printer, 15-14
 property, 15-17
バーコード・フォント, 4-25
バッチ変更、XML, 16-17
パフォーマンス
 向上, 20-1 ~ 20-24
 トラブルシューティング, D-20
 配布の最適化, 3-65
パラメータ・フォーム, 12-10
 キー・マップ・ファイル, 13-17
パラメータ・リスト (イベント)
 作成, 17-2 ~ 17-3
 操作, 17-2 ~ 17-3
表記の方向、言語, 18-5
ファイアウォール
 プロキシ情報, 3-56
ファイルの検索, 4-11
ファイル宛先, A-39
フォーマット
 グラフィカル・レイアウト・オブジェクト, 20-19
 垂直拡張度, 20-19
 水平拡張度, 20-19
 非グラフィカル・レイアウト・オブジェクト, 20-19
フォーマット・トリガー, 20-20
フォント

 構成ファイル, 4-8
 使用と管理, 4-1
 追加, 4-6
 トラブルシューティング, 4-16, D-20
フォント、TrueType Big, 4-19, 18-10
フォント検索アルゴリズム, 4-3
フォント検証ユーティリティ, 4-16
フォントのエイリアシング, 4-12, 18-5
 検証, 4-16
フォントのサポート、グローバリゼーション, 18-10
ブリッジ, 20-13
 タイムアウトのチューニング, 20-13
 「Oracle Reports ブリッジ」を参照, A-17
ブリッジおよびネットワークのタイムアウトのチューニング, 20-13
ブリッジ構成 (OPMN), 3-62
ブレイク・グループ
 ORDER BY, 20-17
ブロードキャスト・メカニズム, 1-8, 3-36
プリンタ・アクセス制御, 12-11
プリンタ宛先, A-39
プロキシ情報
 構成, 3-56
プログラム・ユニット、XML による追加, 16-6
プロバイダ
 レポートに対して作成, 12-21
ヘッダー, 20-22
変数、XML 属性での使用, 15-3
変数、グローバリゼーション・サポートの環境, 18-2 ~ 18-6
方向、言語, 18-5
本体, 20-22
翻訳
 PL/SQL ブロック, 18-11
 TranslationHub ツール, 18-11
ポートレット
 ページへの追加, 12-23
 レポートに対するプロバイダの作成, 12-21

ま

マップ
 URL パラメータ、キー・マップ・ファイル, 13-16
マルチバイト・キャラクタ・セット, 18-1, 18-9
メール宛先, A-39
メッセージ・キュー、作成, 17-8
メッセージ、言語, 18-5

や

要素、「配布」または「構成」を参照
曜日名、言語, 18-5
読取り順序, 18-1, 18-8

ら

ランタイム URL, 13-1 ~ 13-17
ランタイム URL 構文, 13-2
ランタイム・パラメータ・フォーム, 12-10
リスナー, 1-4
例
 rwdiag, E-2
例、配布, 15-17 ~ 15-26

- destype, 15-25
- file, 15-21
- foreach, 15-17
- mail, 15-19
- printer, 15-23
- レジストリ、Windows, 18-4
- レジストリ、編集, B-1
- レポート
 - OracleAS Portal からのバッチ削除, C-4
 - Portal を使用した要求, 13-5
 - URL 構文, 13-2
 - URL リクエスト, 13-5
 - XML カスタマイズ, 16-3 ~ 16-7
 - XML データ・モデル, 16-8 ~ 16-12
 - 一括処理, 15-4
 - イベントによる起動, 17-6
 - カスタム XML の適用, 16-13 ~ 16-17
 - カスタム XML のデバッグ, 16-17 ~ 16-19
 - キャッシュ, 13-15
 - コマンドライン・リクエスト, 13-5
 - 処理, 1-6
 - 自動実行, 13-14
 - スケジューリング, 13-14
 - パッケージ・プロシージャを使用した要求, 13-5
 - 要求の方法, 13-5
- レポート・アクセス制御, 12-7
- レポート・ウィザード
 - 用語集, 用語集-5
- レポート、実行, 13-1 ~ 13-17
- レポート定義、XML, 16-1 ~ 16-19
- レポートでの計算, 20-16
- レポート・トレース, 20-3
- レポートのカスタマイズ、XML, 16-1 ~ 16-19
- レポートの実行, 13-1
- レポートの自動実行, 13-14
 - OracleAS Portal の使用, 12-22
- レポートの配布, 12-1 ~ 12-25
- レポートの配布、最適化, 3-65
- レポート・ブロック・ウィザード
 - 用語集, 用語集-5
- レポート・ポートレットの公開, 13-13
- レポート・リクエスト
 - URL エンジン, 13-13
 - Web ブラウザから, 13-13
 - 自動実行のスケジューリング, 13-14
 - レポート・ポートレットの公開, 13-13
- ローカルファイル宛先, A-39

わ

- ワークフロー, 3-65