

## **Oracle® Application Server Reports Services**

レポート Web 公開ガイド

10g (9.0.4)

**部品番号 : B12355-02**

2004 年 6 月

Oracle Application Server Reports Services レポート Web 公開ガイド, 10g (9.0.4)

部品番号 : B12355-02

原本名 : Oracle Application Server Reports Services Publishing Reports to the Web, 10g (9.0.4)

原本部品番号 : B13673-01

原著者 : Frank Rovitto

原本協力者 : Mick Andrew, Robin J. Fisher, Parineetha Henry, Shaun Lin, Avnish Malik, Rohit Marwaha, Vinay Pamadi, Rajesh Ramachandran, Sripathy P Rao, Danny Richardson, Jim Safcik, Anil Sharma, Vishal Sharma, J. Toby Shimizu, Navneet Singh, Jeff Tang, Vanessa Wang, Philipp Weckerle

Copyright © 2003, 2004 Oracle Corporation. All rights reserved.

#### 制限付権利の説明

このプログラム（ソフトウェアおよびドキュメントを含む）には、オラクル社およびその関連会社に所有権のある情報が含まれています。このプログラムの使用または開示は、オラクル社およびその関連会社との契約に記された制約条件に従うものとし、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 は Oracle Corporation およびその関連会社の登録商標です。その他の名称は、Oracle Corporation または各社が所有する商標または登録商標です。

---

---

# 目次

はじめに .....	xix
対象読者 .....	xx
このマニュアルの構成 .....	xx
関連ドキュメント .....	xxii
表記規則 .....	xxiii

## 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 のコンポーネント .....	1-4
1.3	OracleAS Reports Services のランタイム・プロセス .....	1-6
1.4	システム・セットアップ時の考慮事項 .....	1-8
1.4.1	使用される要求タイプの選択 .....	1-8
1.4.2	サーブレット、JSP、または CGI の選択 .....	1-9
1.4.3	シングルマシン構成とマルチマシン構成の選択 .....	1-9
1.4.4	複数サーバーのクラスタ化の選択 .....	1-10
1.5	高可用性の維持 .....	1-11

### 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
2.2	Oracle HTTP Server が実行されていることの確認 .....	2-8
2.3	Reports Servlet および Reports Server が実行されていることの確認 .....	2-8

### 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
3.2.2	動的な環境切替え .....	3-34
3.2.3	OracleAS Portal への接続 .....	3-39
3.3	Reports Servlet の構成 .....	3-39
3.3.1	キー・マップ・ファイルの位置の指定 .....	3-40
3.3.2	キー・マップ・ファイルの再ロード .....	3-41
3.3.3	Web コマンド出力の非表示 .....	3-41
3.3.4	ログイン・ダイアログ・ボックスの選択 .....	3-43
3.3.5	Reports Servlet と JSP のトレース・オプションの設定 .....	3-43
3.3.6	rwervlet を使用したレポート出力のキャラクタ・エンコードの指定 .....	3-44
3.3.7	URL に指定された HTML コードのブラウザでの実行の禁止 .....	3-44
3.3.8	rwervlet に対する同時接続のプール・サイズの指定 .....	3-45
3.3.9	サーバー・エラー・メッセージの外観のカスタマイズ .....	3-46
3.3.10	インプロセス・サーバーの指定 .....	3-46
3.3.11	デフォルトの Reports Server の識別 .....	3-46
3.3.12	動的に生成されたイメージの指定 .....	3-47
3.3.13	データベース認証とシステム認証 Cookie の期限切れの設定 .....	3-47
3.3.14	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 .....	3-48
3.3.15	診断 / デバッグ出力のフォーマットの追加 .....	3-48
3.3.16	rwervlet ヘルプ・ファイルの定義 .....	3-48
3.3.17	OracleAS Single Sign-On 使用の指定 .....	3-49
3.4	URL エンジンの構成 .....	3-49
3.5	プロキシ情報の入力 .....	3-50
3.6	Oracle Process Manager and Notification Server および Oracle Enterprise Manager 10g による Reports Server の構成 .....	3-51
3.6.1	opmn.xml .....	3-51
3.7	UNIX での DISPLAY およびプリンタ依存性 .....	3-56
3.7.1	ScreenPrinter .....	3-57
3.7.2	拡張イメージングのサポート .....	3-58



3.8	インプロセス・サーバーのデフォルトのプリンタの設定 .....	3-59
-----	---------------------------------	------

## 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
4.1.3	ユーザー・インタフェースのフォント .....	4-7
4.2	フォントの追加 .....	4-8
4.2.1	Reports Builder へのフォントの追加 .....	4-8
4.2.2	レポート出力用の新しいフォントの追加 .....	4-9
4.3	フォント構成ファイル .....	4-11
4.3.1	ファイルの検索 .....	4-15
4.4	フォント・エイリアシング .....	4-15
4.4.1	エイリアシング情報の指定 .....	4-16
4.4.2	フォント・エイリアシングのメカニズム .....	4-18
4.4.3	フォント・エイリアス・ファイルのセクション .....	4-19
4.4.4	フォント・エイリアシング・ファイルの検証 .....	4-21
4.5	フォントに関する問題のトラブルシューティング .....	4-21
4.6	フォントのタイプ .....	4-29
4.6.1	キャラクタ・セット .....	4-30
4.6.2	Unicode .....	4-30
4.6.3	Type1 フォント .....	4-32
4.6.4	TrueType フォント .....	4-32
4.6.5	バーコード・フォント .....	4-33
4.6.6	CID フォント .....	4-34

## 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 .....	5-4
5.2	UNIX でのプリンタの設定 .....	5-5
5.2.1	UNIX でのプリンタのインストール .....	5-5
5.2.2	Oracle Reports で使用するプリンタの設定の確認 .....	5-5
5.3	印刷環境の構成 .....	5-6

5.3.1	uiprint.txt ファイルの編集 .....	5-6
5.3.2	環境変数 .....	5-9
5.3.3	印刷プロパティのダイアログ・ボックス .....	5-9
5.4	プリンタ関連のファイル .....	5-9
5.4.1	ファイルの概要 .....	5-10
5.4.2	PPD ファイル .....	5-10
5.4.3	HPD ファイル .....	5-13
5.4.4	フォント・メトリック・ファイル .....	5-14
5.4.5	uifont.ali .....	5-15
5.4.6	uiprint.txt .....	5-15
5.4.7	プリンタ関連ファイルの編集 .....	5-16
5.5	NLS のサポート .....	5-20
5.5.1	マルチバイト・キャラクタ・セットの印刷 .....	5-20
5.5.2	IX および PASTA の概要 .....	5-21
5.6	デバッグ・オプション .....	5-21
5.6.1	DEBUG_SLFIND .....	5-22
5.6.2	TK_DEBUG_POSTSCRIPT .....	5-23
5.7	よくある質問 .....	5-23
5.7.1	一般的な印刷エラー・メッセージ .....	5-24
5.7.2	PCL の印刷の問題 .....	5-27
5.7.3	PostScript の印刷の問題 .....	5-28
5.7.4	フォントに関連する印刷の問題 .....	5-30
5.7.5	印刷された出力の問題 .....	5-31

## 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
6.1.2	フォント関連の機能 .....	6-3
6.1.3	実行の優先順位 .....	6-12
6.1.4	アクセシビリティ .....	6-14
6.1.5	分類 .....	6-15
6.1.6	グラフのサポートの強化 .....	6-15
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 .....	6-16
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 .....	6-16
6.2.2	レポートを Windows で設計し、UNIX に配布する場合 .....	6-16

6.3	Unicode PDF ファイルの生成 .....	6-21
6.3.1	フォント・サブセット .....	6-21
6.4	BiDi PDF ファイルの生成 .....	6-22
6.4.1	フォント・サブセット .....	6-22
6.5	マルチバイト PDF ファイルの生成 .....	6-23
6.5.1	フォント・エイリアシング .....	6-24
6.6	バーコード PDF ファイルの生成 .....	6-25
6.6.1	フォントの埋込み .....	6-25

## 7 OracleAS Reports Services の宛先の構成

7.1	出力処理の概要 .....	7-2
7.2	サーバーへの宛先タイプの登録 .....	7-4
7.2.1	サーバー構成ファイルの destination セクションの設定 .....	7-4
7.2.2	宛先として有効な値の入力 .....	7-5
7.2.3	宛先の例 .....	7-7

## 8 JDBC PDS の構成と使用

8.1	JDBC 構成ファイル .....	8-2
8.1.1	プリインストールされているドライバ・エントリの確認 .....	8-7
8.1.2	Merant DataDirect ドライバのインストールと構成 .....	8-7
8.2	JDBC 問合せの定義と実行 .....	8-14
8.2.1	接続情報の例 .....	8-18
8.3	OracleAS Reports Services を使用した JDBC レポートの実行 .....	8-20
8.4	トラブルシューティング情報 .....	8-21
8.4.1	エラー・メッセージ .....	8-21
8.4.2	トレース情報 .....	8-23
8.5	ユーザー独自の PDS の追加 .....	8-26
8.5.1	PDS の登録 .....	8-27
8.5.2	jdbcpds.conf ファイルの構成 .....	8-28
8.5.3	PDS JAR ファイルのインストール .....	8-28
8.5.4	ドライバの JAR ファイルのインストール .....	8-28

## 9 OracleAS Reports Services のセキュリティ保護

9.1	OracleAS Reports Services のセキュリティについて .....	9-2
9.1.1	リソースの保護 .....	9-2

9.1.2	認証とアクセス権の適用 .....	9-4
9.1.3	Oracle Identity Management インフラストラクチャの利用 .....	9-9
9.2	OracleAS Reports Services セキュリティの構成 .....	9-11
9.2.1	OracleAS Reports Services セキュリティ・オプションの構成 .....	9-11

## 10 OracleAS Single Sign-On の構成と管理

10.1	前提条件 .....	10-2
10.2	OracleAS Single Sign-On のデフォルトの構成 .....	10-3
10.3	OracleAS Single Sign-On の管理 .....	10-4
10.3.1	OracleAS Single Sign-On の有効化と無効化 .....	10-4
10.3.2	Reports Server のセキュリティの有効化と無効化 .....	10-4
10.3.3	データソースのセキュリティの有効化と無効化 .....	10-5
10.3.4	Oracle Internet Directory への接続 .....	10-15
10.4	Oracle Internet Directory の接続エンティティの選択 .....	10-16
10.5	OracleAS Forms Services のセキュリティに関する考慮事項 .....	10-17

## 11 OracleAS Portal でのレポートの配布

11.1	OracleAS Portal での Reports ユーザーおよび名前付きグループの作成 .....	11-2
11.1.1	デフォルトの Reports 関連グループ .....	11-3
11.1.2	ユーザーとグループの作成 .....	11-4
11.2	OracleAS Portal への Oracle Reports コンポーネントの登録 .....	11-5
11.2.1	使用可能日カレンダーの作成 .....	11-5
11.2.2	プリンタの登録 .....	11-10
11.2.3	Reports Server の登録 .....	11-13
11.2.4	レポートの登録 .....	11-16
11.2.5	「管理」ポートレット .....	11-23

## 12 Reports Server のクラスタ化

12.1	クラスタの概要 .....	12-2
12.2	クラスタのセットアップ .....	12-3
12.2.1	Reports Server 名の変更 .....	12-3
12.2.2	公開鍵および秘密鍵の新規生成 .....	12-5
12.2.3	公開鍵および秘密鍵のサーバー構成ファイルへの入力 .....	12-5
12.2.4	Reports Server の再起動 .....	12-6
12.2.5	クラスタへの要求送信 .....	12-6

## 第 II 部 サーバーへの要求送信

### 13 レポート要求の実行

13.1	Reports の URL 構文 .....	13-2
13.1.1	サーブレット .....	13-2
13.1.2	JSP .....	13-3
13.1.3	CGI .....	13-5
13.2	レポート要求の方法 .....	13-6
13.3	レポートの配布 .....	13-7
13.3.1	ペーパー・レイアウトを使用したレポートの配布 .....	13-8
13.3.2	ペーパー・レイアウトを使用したレポートの実行 .....	13-8
13.3.3	Web およびペーパーへの JSP レポートの配布 .....	13-9
13.3.4	JSP ベースの Web レポートのブラウザからの実行 .....	13-14
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 .....	13-14
13.3.6	Solaris で WE8MSWIN1252 キャラクタ・セットを使用したレポートの実行 .....	13-15
13.4	OracleAS Portal でのレポートの公開 .....	13-16
13.4.1	レポートに対するプロバイダの作成 .....	13-17
13.4.2	レポート定義ファイル・アクセスの作成 .....	13-17
13.4.3	ページへのレポート・ポートレットの追加 .....	13-17
13.4.4	レポート・コンポーネントのアイテムとしてのページへの追加 .....	13-20
13.5	Web ブラウザからのレポート要求の指定 .....	13-21
13.6	URL エンジンへの要求送信 .....	13-21
13.7	Web サービスを介したレポートの実行 .....	13-22
13.8	自動実行のためのレポートのスケジューリング .....	13-22
13.9	追加のパラメータ .....	13-23
13.10	キャッシュからのレポート出力の再使用 .....	13-23
13.10.1	使用上の注意 .....	13-24
13.11	キー・マップ・ファイルの使用方法 .....	13-24
13.11.1	キー・マッピングを使用可能にする方法 .....	13-25
13.11.2	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 .....	13-25
13.11.3	JSP 以外のすべてでキーを使用する場合 .....	13-26
13.11.4	JSP としてのレポート実行でのキーの使用方法 .....	13-26

### 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
14.2.3	Oracle Reports Web サービスの操作 .....	14-6
14.3	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 .....	14-11

## 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 .....	15-4
15.3.1	destinations .....	15-4
15.3.2	foreach .....	15-5
15.3.3	mail .....	15-7
15.3.4	body .....	15-10
15.3.5	attach .....	15-11
15.3.6	include .....	15-13
15.3.7	file .....	15-15
15.3.8	printer .....	15-17
15.3.9	destype .....	15-19
15.3.10	property .....	15-22
15.4	配布 XML ファイルの例 .....	15-22
15.4.1	foreach の例 .....	15-22
15.4.2	mail の例 .....	15-25
15.4.3	file の例 .....	15-28
15.4.4	printer の例 .....	15-29
15.5	実行時の配布 XML ファイルの使用方法 .....	15-32
15.6	カスタム / プラッグアブル・デスティネーションの定義 .....	15-33
15.7	配布の使用に関する制限 .....	15-36
15.7.1	destype=oraclePortal .....	15-36
15.7.2	XML とデリミタ付き出力 .....	15-36
15.7.3	配布と動的フォーマット値 .....	15-37

## 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 .....	16-3
------	-----------------	------

16.2	XML カスタマイズの作成 .....	16-3
16.2.1	必須 XML タグ .....	16-4
16.2.2	スタイルの変更 .....	16-4
16.2.3	書式マスクの変更 .....	16-5
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 .....	16-8
16.2.7	URL のエンコード .....	16-9
16.3	XML データ・モデルの作成 .....	16-9
16.3.1	複数のデータソースの作成 .....	16-9
16.3.2	データソース間のリンク .....	16-10
16.3.3	各データソース内でのグループ階層の作成 .....	16-11
16.3.4	クロス積（マトリックス）グループの作成 .....	16-12
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 .....	16-13
16.3.6	パラメータの作成 .....	16-14
16.4	実行時の XML ファイルの使用方法 .....	16-15
16.4.1	実行時の XML レポート定義の適用 .....	16-16
16.4.2	XML レポート定義の単独での実行 .....	16-20
16.4.3	バッチ変更の実行 .....	16-21
16.5	XML レポート定義のデバッグ .....	16-21
16.5.1	XML パーサーのエラー・メッセージ .....	16-22
16.5.2	トレース・オプション .....	16-22
16.5.3	rwbuilder .....	16-23
16.5.4	デバッグを目的とした、ファイルへの XML の書込み .....	16-23

## 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
17.1.3	ジョブの送信方法 .....	17-4
17.1.4	ステータスの確認方法 .....	17-5
17.1.5	サーバーのステータス・レコードの使用方法 .....	17-6
17.2	Event-Driven Publishing API を使用するアプリケーションのデバッグ .....	17-7
17.3	データベース・イベントからのレポートの起動 .....	17-7
17.4	Oracle Advanced Queuing との統合 .....	17-9

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 .....	17-9
17.4.2	エンキュー・プロシージャの作成 .....	17-10
17.4.3	デキュー・プロシージャの作成 .....	17-11

## 第 III 部 各国語サポートおよび双方向サポート

### 18 NLS と双方向サポートの実装

18.1	NLS 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	NLS の環境変数 .....	18-2
18.2.1	NLS_LANG 環境変数 .....	18-3
18.2.2	DEVELOPER-NLS_LANG 環境変数と USER-NLS_LANG 環境変数 .....	18-7
18.3	JSP または XML ファイルでのキャラクタ・セットの指定 .....	18-8
18.4	双方向サポート .....	18-11
18.5	Unicode .....	18-12
18.5.1	Unicode のサポート .....	18-12
18.5.2	Unicode フォントのサポート .....	18-13
18.5.3	Unicode サポートの有効化 .....	18-13
18.5.4	ALTER SESSION の使用方法 .....	18-14
18.6	アプリケーションの翻訳 .....	18-14

## 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 ページへのナビゲート .....	19-2
19.2.1	Grid Control コンソールでの Reports Server ページへのナビゲート .....	19-3
19.2.2	Application Server Control コンソールでの Reports Server ページへのナビゲート .....	19-5
19.3	Reports Server の起動、停止および再起動 .....	19-7
19.4	ジョブ・キューの表示と管理 .....	19-8
19.4.1	Application Server Control コンソールでのジョブ・キューの表示と管理 .....	19-8
19.5	サーバー・パフォーマンスの監視 .....	19-16
19.5.1	Oracle Enterprise Manager 10g Grid Control コンソール .....	19-16
19.5.2	Oracle Enterprise Manager 10g Application Server Control コンソール .....	19-20



19.6	Reports Server 構成ファイルの表示と変更 .....	19-23
19.6.1	選択したパラメータの構成 .....	19-23
19.6.2	server_name.conf ファイルの構成 .....	19-25
19.7	サーバー・クラスター・メンバーの表示とメンバーへのリンク .....	19-26
19.8	ポート番号の表示 .....	19-27
19.9	中間層の変更 .....	19-28

## 20 Oracle Reports のチューニング

20.1	方法論 .....	20-2
20.1.1	パフォーマンス分析ツール .....	20-3
20.1.2	データへのアクセス .....	20-11
20.1.3	データのフォーマット .....	20-16
20.1.4	レイアウトに関する一般的なガイドライン .....	20-19
20.1.5	Forms からの Oracle Reports のコール .....	20-20
20.1.6	レポートの実行 .....	20-21

## 第 V 部 付録

### A コマンドライン・オプション

A.1	コマンドの概要 .....	A-2
A.1.1	キーワードと使用可能なコマンド .....	A-2
A.1.2	rwclient .....	A-6
A.1.3	rwrn .....	A-8
A.1.4	rwbuilder .....	A-10
A.1.5	rwconverter .....	A-10
A.1.6	rwservlet .....	A-12
A.1.7	rwcgi .....	A-15
A.1.8	rwserver .....	A-18
A.2	コマンドライン構文 .....	A-18
A.3	全般的な使用上の注意 .....	A-18
A.4	コマンドライン・キーワード .....	A-18
A.4.1	ACCESSIBLE .....	A-18
A.4.2	ARRAYSIZE .....	A-19
A.4.3	AUTHID .....	A-20
A.4.4	AUTOCOMMIT .....	A-20
A.4.5	AUTOSTART .....	A-21

A.4.6	BACKGROUND .....	A-22
A.4.7	BATCH .....	A-22
A.4.8	BCC .....	A-23
A.4.9	BLANKPAGES .....	A-24
A.4.10	BUFFERS .....	A-25
A.4.11	CACHELOB .....	A-25
A.4.12	CC .....	A-26
A.4.13	CELLWRAPPER .....	A-27
A.4.14	CMDFILE .....	A-28
A.4.15	CMDKEY .....	A-29
A.4.16	COLLATE .....	A-30
A.4.17	CONTENTAREA .....	A-30
A.4.18	COPIES .....	A-31
A.4.19	CUSTOMIZE .....	A-32
A.4.20	DATEFORMATMASK .....	A-33
A.4.21	DELAUTH .....	A-33
A.4.22	DELIMITED_HDR .....	A-34
A.4.23	DELIMITER .....	A-35
A.4.24	DESFORMAT .....	A-36
A.4.25	DESNAME .....	A-38
A.4.26	DEST .....	A-39
A.4.27	DESTINATION .....	A-40
A.4.28	DESTYPE .....	A-41
A.4.29	DISTRIBUTE .....	A-46
A.4.30	DTYPE .....	A-46
A.4.31	DUNIT .....	A-48
A.4.32	ENGINERESPONSETIMEOUT .....	A-48
A.4.33	EXPIRATION .....	A-49
A.4.34	EXPIREDAYS .....	A-50
A.4.35	EXPRESS_SERVER .....	A-50
A.4.36	FORMSIZE .....	A-53
A.4.37	FROM .....	A-53
A.4.38	GETJOBID .....	A-54
A.4.39	GETSERVERINFO .....	A-54
A.4.40	HELP .....	A-55
A.4.41	INSTALL .....	A-56
A.4.42	ITEMTITLE .....	A-56
A.4.43	JOBNAME .....	A-57

A.4.44	JOBTYPE .....	A-57
A.4.45	JVMOPTIONS .....	A-58
A.4.46	KILLJOBID .....	A-59
A.4.47	LONGCHUNK .....	A-59
A.4.48	MIMETYPE .....	A-60
A.4.49	MODE .....	A-61
A.4.50	MODULE REPORT .....	A-61
A.4.51	NONBLOCKSQL .....	A-62
A.4.52	NOTIFYFAILURE .....	A-62
A.4.53	NOTIFYSUCCESS .....	A-63
A.4.54	NUMBERFORMATMASK .....	A-64
A.4.55	OLAP_CON .....	A-65
A.4.56	ONFAILURE .....	A-65
A.4.57	ONSUCCESS .....	A-66
A.4.58	ORIENTATION .....	A-67
A.4.59	OUTPUTFOLDER .....	A-68
A.4.60	OUTPUTIMAGEFORMAT .....	A-69
A.4.61	OUTPUTPAGE .....	A-70
A.4.62	OVERWRITE .....	A-71
A.4.63	P_AVAILABILITY .....	A-71
A.4.64	P_DESCRIPTION .....	A-72
A.4.65	P_FORMATS .....	A-73
A.4.66	P_NAME .....	A-73
A.4.67	P_OWNER .....	A-74
A.4.68	P_PFORMTEMPLATE .....	A-74
A.4.69	P_PRINTERS .....	A-75
A.4.70	P_PRIVILEGE .....	A-75
A.4.71	P_SERVERS .....	A-76
A.4.72	P_TRIGGER .....	A-76
A.4.73	P_TYPES .....	A-77
A.4.74	PAGEGROUP .....	A-77
A.4.75	PAGESIZE .....	A-78
A.4.76	PAGESTREAM .....	A-79
A.4.77	PARAMFORM .....	A-80
A.4.78	PARSEQUERY .....	A-81
A.4.79	PDFCOMP .....	A-81
A.4.80	PDFEMBED .....	A-82
A.4.81	PRINTJOB .....	A-82

A.4.82	READONLY .....	A-83
A.4.83	RECURSIVE_LOAD .....	A-84
A.4.84	REPLACEITEM .....	A-85
A.4.85	REPLYTO .....	A-86
A.4.86	REPORT   MODULE .....	A-86
A.4.87	ROLE .....	A-86
A.4.88	RUNDEBUG .....	A-87
A.4.89	SAVE_RDF .....	A-88
A.4.90	SCHEDULE .....	A-88
A.4.91	SERVER .....	A-89
A.4.92	SHOWAUTH .....	A-90
A.4.93	SHOWENV .....	A-91
A.4.94	SHOWJOBID .....	A-91
A.4.95	SHOWJOBS .....	A-92
A.4.96	SHOWMAP .....	A-93
A.4.97	SHOWMYJOBS .....	A-94
A.4.98	SHUTDOWN .....	A-94
A.4.99	SITENAME .....	A-95
A.4.100	SOURCE .....	A-96
A.4.101	SQLTRACE .....	A-97
A.4.102	SSOCONN .....	A-97
A.4.103	STATUSFOLDER .....	A-99
A.4.104	STATUSFORMAT .....	A-100
A.4.105	STATUSPAGE .....	A-100
A.4.106	STYPE .....	A-101
A.4.107	SUBJECT .....	A-102
A.4.108	SUPPRESSLAYOUT .....	A-102
A.4.109	TOLERANCE .....	A-103
A.4.110	TRACEFILE .....	A-104
A.4.111	TRACEMODE .....	A-105
A.4.112	TRACEOPTS .....	A-106
A.4.113	UNINSTALL .....	A-108
A.4.114	UPGRADE_PLSQL .....	A-108
A.4.115	URLPARAMETER .....	A-109
A.4.116	USERID .....	A-109
A.4.117	WEBSERVER_DEBUG .....	A-110
A.4.118	WEBSERVER_DOCROOT .....	A-111
A.4.119	WEBSERVER_PORT .....	A-111

## B Reports 関連の環境変数

B.1	環境変数 .....	B-2
B.1.1	CA_GPREFS .....	B-2
B.1.2	CA_UPREFS .....	B-2
B.1.3	DELIMITED_LINE_END .....	B-3
B.1.4	DOC .....	B-3
B.1.5	DEVELOPER-NLS_LANG .....	B-3
B.1.6	NLS_CALENDAR .....	B-4
B.1.7	NLS_CREDIT .....	B-4
B.1.8	NLS_CURRENCY .....	B-4
B.1.9	NLS_DATE_FORMAT .....	B-4
B.1.10	NLS_DATE_LANGUAGE .....	B-4
B.1.11	NLS_DEBIT .....	B-4
B.1.12	NLS_ISO_CURRENCY .....	B-4
B.1.13	NLS_LANG .....	B-4
B.1.14	NLS_LIST_SEPARATOR .....	B-5
B.1.15	NLS_MONETARY_CHARACTERS .....	B-5
B.1.16	NLS_NUMERIC_CHARACTERS .....	B-6
B.1.17	NLS_SORT .....	B-6
B.1.18	ORACLE_AFM .....	B-6
B.1.19	ORACLE_HOME .....	B-6
B.1.20	ORACLE_HPD .....	B-7
B.1.21	ORACLE_PATH .....	B-7
B.1.22	ORACLE_PPD .....	B-7
B.1.23	ORACLE_TFM .....	B-8
B.1.24	ORAINFONAV_DOCPATH .....	B-8
B.1.25	PRINTER .....	B-9
B.1.26	REPORTS_ADD_HWMARGIN .....	B-9
B.1.27	REPORTS_ARABIC_NUMERAL .....	B-10
B.1.28	REPORTS_BIDI_ALGORITHM .....	B-11
B.1.29	REPORTS_CGIDIAGBODYTAGS * .....	B-11
B.1.30	REPORTS_CGIDIAGHEADTAGS * .....	B-12
B.1.31	REPORTS_CGIHELP * .....	B-12
B.1.32	REPORTS_CGIMAP * .....	B-13
B.1.33	REPORTS_CGINODIAG * .....	B-13
B.1.34	REPORTS_CLASSPATH .....	B-14
B.1.35	REPORTS_COOKIE_EXPIRE * .....	B-15
B.1.36	REPORTS_DB_AUTH * .....	B-16

B.1.37	REPORTS_DEFAULT_DISPLAY .....	B-16
B.1.38	REPORTS_DEFAULT_PIXEL_SIZE .....	B-17
B.1.39	REPORTS_ENCRYPTION_KEY * .....	B-18
B.1.40	REPORTS_GRAPH_IMAGE_DPI .....	B-18
B.1.41	REPORTS_JVM_OPTIONS .....	B-19
B.1.42	REPORTS-NLS_XML_CHARSETS .....	B-20
B.1.43	REPORTS_NO_DUMMY_PRINTER .....	B-20
B.1.44	REPORTS_NO_HTML_SPACE_REPLACE .....	B-21
B.1.45	REPORTS_OUTPUTIMAGEFORMAT .....	B-22
B.1.46	REPORTS_PATH .....	B-23
B.1.47	REPORTS_RESOURCE .....	B-23
B.1.48	REPORTS_SERVER * .....	B-24
B.1.49	REPORTS_SPACE_BREAK .....	B-24
B.1.50	REPORTS_SSLPORT * .....	B-25
B.1.51	REPORTS_SYS_AUTH .....	B-25
B.1.52	REPORTS_TAGLIB_URI .....	B-26
B.1.53	REPORTS_TMP .....	B-26
B.1.54	REPORTS_USEREXIT .....	B-27
B.1.55	RW .....	B-27
B.1.56	TK_PRINT .....	B-27
B.1.57	TK_PRINT_STATUS .....	B-28
B.1.58	TK_PRINTER .....	B-29
B.1.59	TK90_AFM .....	B-30
B.1.60	TK90_HPD .....	B-30
B.1.61	TK90_PPD .....	B-31
B.1.62	TK90_TFM .....	B-31
B.1.63	USERNAME .....	B-31
B.1.64	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-4
C.2	レポート・パッケージのバッチ削除 .....	C-5
C.3	PL/SQL バッチ登録ファンクション .....	C-5

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 OracleAS 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：** このマニュアルの最新版を参照するには、OTN-J (Oracle Technology Network Japan) (<http://otn.oracle.co.jp/products/reports/>) にアクセスし、「ドキュメント」をクリックして、『Oracle Application Server Reports Services レポート Web 公開ガイド』にナビゲートします。

---

---

## 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す。

## このマニュアルの構成

このマニュアルには、次の章が含まれています。

### **第 1 章「OracleAS Reports Services のアーキテクチャについて」**

OracleAS Reports Services アーキテクチャの概要について説明します。

### **第 2 章「OracleAS Reports Services の起動と停止」**

OracleAS Reports Services の起動および停止方法について説明します。

### **第 3 章「OracleAS Reports Services の構成」**

OracleAS Reports Services の構成方法について説明します。

### **第 4 章「Oracle Reports でのフォントの管理」**

Oracle Reports でのフォントの構成方法、フォント・エイリアシングおよびフォントのトラブルシューティングについて説明します。

### **第 5 章「UNIX での Oracle Reports による印刷」**

UNIX での印刷の構成方法および印刷問題のトラブルシューティングについて説明します。

## **第 6 章「Oracle Reports での PDF の使用」**

Oracle Reports での PDF 機能、およびマルチバイト・フォント、双方向 (BiDi) 言語、パーコード、Unicode キャラクタ・セットを使用した PDF 出力の生成方法について説明します。

## **第 7 章「OracleAS Reports Services の宛先の構成」**

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。

## **第 8 章「JDBC PDS の構成と使用」**

OracleAS Reports Services での使用を目的とした、JDBC PDS の構成方法について説明します。

## **第 9 章「OracleAS Reports Services のセキュリティ保護」**

OracleAS Reports Services のセキュリティ機能の概念的な説明を提供します。

## **第 10 章「OracleAS Single Sign-On の構成と管理」**

OracleAS Reports Services を構成して、OracleAS Single Sign-On とともに使用する方法について説明します。

## **第 11 章「OracleAS Portal でのレポートの配布」**

OracleAS Reports Services で作成したレポートを、OracleAS Portal を使用して配布する方法について説明します。

## **第 12 章「Reports Server のクラスタ化」**

Reports Server をクラスタ化して、パフォーマンスおよび信頼性を強化する方法について説明します。

## **第 13 章「レポート要求の実行」**

レポートの様々な実行方法、特にランタイム URL の構成方法について説明します。

## **第 14 章「Oracle Reports Web サービスの使用方法」**

Oracle Reports の Web サービスによってサポートされている様々な操作について説明します。

## **第 15 章「拡張配布の作成」**

配布 XML ファイルを使用して拡張配布を設定する方法について説明します。

## **第 16 章「XML によるレポートのカスタマイズ」**

XML を使用して、実行時のレポートのカスタマイズを行う方法について説明します。

## 第 17 章「イベント・ドリブンによる公開の使用方法」

イベント・ドリブンの公開を使用して、データベース・トリガーに応じてレポートを自動的に起動する方法について説明します。

## 第 18 章「NLS と双方向サポートの実装」

OracleAS Reports Services に関する NLS の情報を提供します。

## 第 19 章「OracleAS Reports Services の管理と監視」

OracleAS Reports Services と Oracle Enterprise Manager 10g (OEM) との統合、および OEM で Reports Server を監視する方法について説明します。

## 第 20 章「Oracle Reports のチューニング」

チューニングおよびパフォーマンス強化のヒントを提供します。

## 付録 A「コマンドライン・オプション」

Reports に関連するコマンドライン引数の情報を提供します。

## 付録 B「Reports 関連の環境変数」

Oracle Reports の環境変数に関する情報を提供します。

## 付録 C「OracleAS Portal へのレポートのバッチ登録」

OracleAS Portal でバッチ・スクリプトを使用して、レポートを登録する方法について説明します。

## 関連ドキュメント

レポートの作成、OracleAS Portal、または OracleAS Reports Services の詳細は、次のマニュアル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 Technology Network (<http://portalcenter.oracle.com>) の「Getting Started with OracleAS Portal」
- OTN-J (Oracle Technology Network Japan) (<http://otn.oracle.co.jp/products/reports/>) の「ドキュメント」

# 表記規則

このマニュアルでは、次のような表記規則を使用しています。

規則	意味
.	垂直の省略記号は、例に直接関連しない複数の行が省略されていることを示します。
...	水平の省略記号は、文やコマンド内の、例に直接関連しない文またはコマンドの一部が省略されていることを示します。
<b>太字</b>	太字は、本文中で定義されている用語および用語集に記載されている用語を示します。
<i>変数</i>	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、カッコ内の項目を任意に選択することを表します。大カッコは、入力しないでください。



# 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。これには、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と Reports Server クラスタの情報などが含まれます。

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「OracleAS Reports Services の宛先の構成」
- 第8章「JDBC PDS の構成と使用」
- 第9章「OracleAS Reports Services のセキュリティ保護」
- 第10章「OracleAS Single Sign-On の構成と管理」
- 第11章「OracleAS Portal でのレポートの配布」
- 第12章「Reports Server のクラスタ化」





---

---

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

レポートを公開する準備が整ったら、必要な Web サーバーおよびアプリケーション・サーバー・ツールはすべて Oracle Application Server で利用できます。この章では、関連する OracleAS Reports Services 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、サーバー環境セットアップ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 のコンポーネント](#)
- [OracleAS Reports Services のランタイム・プロセス](#)
- [システム・セットアップ時の考慮事項](#)
- [高可用性の維持](#)

## 1.1 OracleAS Reports Services の概要

Oracle Application Server は、各種 Web サイト、ポータルまたはインターネット・アプリケーションを実行する包括的な統合アプリケーション・サーバーです。このサーバーにより、アプリケーションを Web ブラウザ、モバイル・デバイスやコマンドラインから利用できます。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の公開が可能です。

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、同様の要求が発行された際に再利用できるように、レポート出力をキャッシュします。このサービスは、JSP、Java サブレットおよび CGI 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、サブレット、JSP または CGI（下位互換性を確保するためにのみ維持）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、すべてのジョブ要求を 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要求を操作します。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。Reports Queue Manager (Windows)、Reports Queue Viewer (UNIX)、showjobs コマンド (Web)、および Oracle Enterprise Manager 10g (OEM) の OracleAS Reports Services ページを使用すると、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・オブジェクトは永続的です。したがって、サーバーを停止した後には再起動すると、<sup>1</sup> 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。

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。

<sup>1</sup> 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。

---

---

**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)

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す)。

- 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gable・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。

Web サーバーを複数のマシンにインストールすると、Oracle HTTP Server ではロード・バランシング機能が提供され、Web サーバーの負荷を複数のマシンに分散できます。アプリケーション・サーバーを複数のマシンにインストールすると、OracleAS Reports Services でピアレベルのクラスタ化が可能になり、Reports Server の負荷を複数のマシンに分散できます。

ロード・バランシングとピアレベルのクラスタ化の違いは、ロード・バランシングでは 1 台のマシンですべてのマシンの通信量を管理するのに対し、ピアレベルのクラスタ化ではすべてのマシンが各マシン上の通信量を監視し、各マシンで監視タスクおよび要求への応答タスクを共有することです。ピアレベルのクラスタ化の利点は、単一の障害箇所を排除し、フェイルセーフ環境をさらに強化することです。

---

---

**注意：** Reports Server のクラスタ化の詳細は、[第 12 章「Reports Server のクラスタ化」](#)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位標を上回った (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 のコンポーネント

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

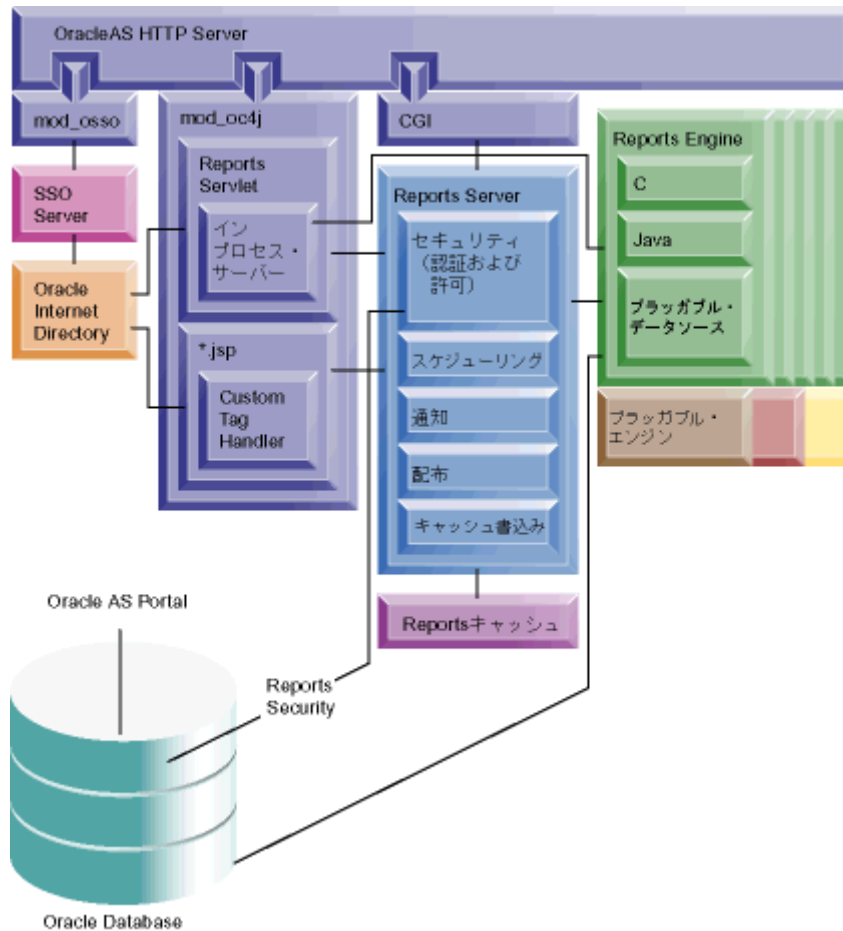


図 1-1 は、稼動中の OracleAS Reports Services 環境のコンポーネントを示しています。これには次のものが含まれます。

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 サブレット・アプリケーションの実行をサポートするサブレット・エンジンも備えています。
2. モジュール **mod\_oc4j** は、Oracle HTTP Server が、サブレットや JSP からの要求を 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** は、Web サーバーのサブレット・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要求を 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。
  - **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。
4. **Reports CGI** 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：** Reports CGI は、下位互換性を確保するためにのみ維持されています。

---

5. **Reports Server** はクライアント要求を処理します。この処理では、クライアント要求を、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、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。
6. **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。
7. **Reports Engine** は SQL ベースおよびプラグブル・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。

8. **プラグgable・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gable・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。

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

ここでは、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について説明します。

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 を使用した要求は、JSP、rwservlet または CGI 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 の要求は、mod\_oc4j にリダイレクトされます (ジョブは JSP として実行されるので、mod\_oc4j は OJSP を使用して JSP をサーブレットに変換します)。CGI 要求は、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 Reports CGI (rwcgi は下位互換性を確保するためにのみ維持) 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。
  - JSP または rwservlet を使用したサーバー要求は、インプロセス Reports Server で実行されるか、スタンドアロンの Reports Server プロセス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 rwservlet.properties (ORACLE\_HOME¥reports¥conf¥) 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・サーバーと異なり、クライアントからの要求に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。また、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るため、パフォーマンスが向上します。
  - rwcgi を使用したサーバー要求は、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 が次のように要求を処理します。

要求に TOLERANCE 引数が含まれる場合は、Reports Server がキャッシュをチェックして、その要求を満たす既存の出力があるかどうかを判別します。キャッシュ内に要求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 に送るどのジョブ要求にも、TOLERANCE 引数を指定できます。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 分以内に作成された最新の複製レポート出力を探します。expiration 引数には、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、付録 A 「コマンドライン・オプション」を参照してください。

---

要求が現在実行中のジョブと同じ場合、その要求は、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、Reports Server は要求を処理します。

- a. Reports Server は、Oracle HTTP Server の一部である mod\_osso を介して認証および許可の確認を開始します（構成されている場合）。
- 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・キューに要求を挿入し、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、要求は現在のジョブ・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 Report 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要求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、要求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、実行対象の要求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 要求のみ) 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、アドバンスト・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・ドリブによる公開の使用方法」を参照してください。

## 1.4 システム・セットアップ時の考慮事項

OracleAS Reports Services のセットアップ方法は、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 をセットアップする前に、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、セットアップ・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。

- 使用される要求タイプの選択
- サーブレット、JSP、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
- 複数サーバーのクラスタ化の選択

### 1.4.1 使用される要求タイプの選択

OracleAS Reports Services は、Web ベースの要求と非 Web ベースの要求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 1 つと OracleAS Reports Services が通信し、Web クライアントからのレポート要求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、[サーブレット、JSP、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、ジョブ要求を送信できます。

構成の観点から見た、Web 要求と非 Web 要求の有効化の重要な相違点は、次のとおりです。

- Web 要求を有効にする場合、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。
- 非 Web 要求を有効にする場合は、要求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 要求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 要求と非 Web 要求の両方の実装をサポートします。

## 1.4.2 サブレット、JSP、または CGI の選択

OracleAS Reports Services を Web 環境で使用するには、サブレット、JSP、または CGI を実装する必要があります。サブレットまたは JSP の選択をお勧めします。OracleAS Reports Services での CGI の実装は、下位互換性を確保するためにのみ維持されています。

サブレットと JSP のどちらを選択するかについては、さらに考慮事項が必要です。JSP のみを実装すると、Web 配信に適したレイアウトを公開できます。サ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、サブレットを使用しても 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、サブレットを URL で指定し、JSP をコマンドライン引数でコールします。 `report=myreport.jsp`

レポート実行の詳細は、[第 13 章「レポート要求の実行」](#) を参照してください。

## 1.4.3 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要求がネットワークを経由せず、受信要求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（サブレット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。

レポートを複数の言語で配布する場合は、言語ごとに1つ以上の Reports Server をセットアップするようお勧めします。

### 1.4.4 複数サーバーのクラスタ化の選択

クラスタとは、複数のサーバーを1つのコミュニティに仮想的にグループ化したものです。このコミュニティの目的は、要求処理をクラスタの各メンバーに効率的に分散することです。以前のバージョンと異なり、OracleAS Reports Services でのクラスタ化は、マスター/スレーブ型ではなく、ピアレベルで行われます。ピアレベルのクラスタ化では、受信要求の分散と処理に対して、クラスタの全メンバーが同等の役割を担います。受信要求は、クラスタ内の特定の Reports Server ではなく、クラスタ全体に対して送信されます。このため、あるメンバーが停止しても、残りのメンバーが要求負荷の管理を続行します。1台のマシンの障害が原因でシステム全体がダウンする単一の障害箇所はありません。

各クラスタ・メンバー・マシンは、各サーバー・メンバー上でのレポートの実行が可能になるよう、ほぼ同様に構成する必要があります。つまり、構成ファイルの内容はほぼ同じになります。同じ設定にする必要があるかどうかは、ジョブ関連の設定か、マシン関連の設定かで区別します。ジョブ関連の設定は、すべてのクラスタ・メンバーで同じにする必要があります。ジョブ関連の設定には、セキュリティ、データソースおよび宛先タイプに関する設定などがあります。マシン関連の設定には、maxEngine、minEngine、maxIdle、initEngine などの属性があり、これらはメンバーごとに異なる値を設定できます。

また、クラスタ・メンバーは、次の事項に従う必要があります。

- サーバー関連の環境変数には、同じ値を設定します。
- TNS 設定では、同じデータベースを同じ方法で指定します。

同じクラスタのメンバーとなるサーバーは、クラスタ名を共有し（各サーバー名に追加されます）、同じ公開鍵および秘密鍵を設定する必要があります。

ジョブ関連の構成設定が異なるマシンがあると、クラスタ化の利点を享受できません。

多言語対応のために複数のサーバーをセットアップする場合は、言語ごとに1つ以上のクラスタをセットアップすることをお勧めします。

## 1.5 高可用性の維持

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、その中間層での稼働を継続することができます。それら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何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 接続が何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。

**関連項目：** 高可用性の詳細は、次のマニュアルを参照してください。

- 『Oracle Application Server 10g 高可用性ガイド』



---

---

# 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 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。

---

---

Web 上でレポートを実行する場合は、事前に Oracle HTTP Server を起動しておく必要があります。この操作の詳細は、Oracle Application Server のドキュメントを参照してください。この章の手順はすべて、Oracle HTTP Server が事前に起動されていることを前提としています。

## 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 では、Oracle HTTP Server、Oracle Application Server Containers for J2EE 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 章「OracleAS Reports Services の構成」](#)を参照してください。

---

---

**注意：** OPMN を使用して Reports Server を構成することのもう 1 つの利点は、Reports Server が何らかの原因でクラッシュすると、OPMN によって自動的に再起動されることです。

---

---

### 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、デフォルトで Oracle Enterprise Manager 10g から Reports Server を起動、停止および再起動できます。

---

---

**注意：** スタンドアロンの Reports Server は、Oracle Application Server をインストールする際、自動的に 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、Oracle Enterprise Manager 10g の `targets.xml` ファイルおよび Oracle Process Manager and Notification Server の `opmn.xml` ファイルに手動で登録する必要があります。詳細は、[3.6 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。インプロセス Reports Server は、OPMN では構成されません。

---

---

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. Oracle Enterprise Manager 10g Application Server Control コンソールで、「ターゲット」→「ホスト名」→「Application Server」→「Reports Server」を選択して、管理対象の Reports Server に移動します。
2. 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。
- サーバーが稼働している場合は、「再起動」ボタンと「停止」ボタンが表示されます。

## 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 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 を使用した Reports Server の構成の詳細は、第 3 章「OracleAS Reports Services 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 を実行することができます。

- サービスとしての Reports Server のインストールと起動 (Windows)
- 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

次の各項では、これらのオプションを設定する方法について説明します。

### 2.1.3.1 サービスとしての Reports Server のインストールと起動 (Windows)

Reports Server は、デフォルトではインプロセス・サーバーとしてインストールされますが、Windows マシンでは、必要に応じてサービスとしてインストールできます。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。

```
rwserver -install server_name [batch=yes/no] [autostart=yes/no]
```

batch のデフォルトは、no です。インストール中に確認のためのプロンプトを表示する必要がない場合は、yes と入力します。autostart のデフォルトは、no です。再起動時に、サービスを自動的に開始して、ユーザーが手動で Reports Server を起動する手間を省く場合は、yes と入力します。

このサーバーがクラスタのメンバーである場合は、このコマンドにクラスタ名を追加します。次に例を示します。

```
rwserver -install server_name.cluster_name [batch=yes/no] [autostart=yes/no]
```

サーバーのクラスタ化の詳細は、[第 12 章「Reports Server のクラスタ化」](#)を参照してください。

---

---

**注意：** Reports Server サービスを削除するには、コマンド・プロンプトで `rwserver -uninstall server_name` と入力します。サーバーがクラスタのメンバーである場合は、クラスタ名を追加します。たとえば、`rwserver -uninstall server_name.cluster_name` と入力します。

---

---

Windows で Reports Server を起動する手順は、次のとおりです。

1. Reports Server をホスティングするマシンで、「スタート」→「設定」→「コントロールパネル」を選択し、「コントロールパネル」フォルダの「サービス」をダブルクリックします。
2. 「サービス」ダイアログ・ボックスで Oracle ORACLE\_HOME Reports [repserver] を選択します。ORACLE\_HOME は Reports Server のホーム・ディレクトリ、**repserver** は Reports Server インスタンスの名前です。
3. 「スタートアップ」をクリックします。「サービス」ダイアログ・ウィンドウが表示されます。
4. 「サービス」ダイアログ・ウィンドウで、「ログオン」セクションの「アカウント」を選択し、オペレーティング・システムのユーザー名とパスワードを選択します。これで、サーバー・プロセスを実行するユーザー・アカウントが指定されます。

---

---

**注意：** PostScript またはプリンタに出力する場合は、Reports Server サービスを実行しているユーザーが、確実にデフォルト・プリンタへのアクセス権限を保持するようにしてください。そのためには、Windows サービスの「ログオン」セクションを設定する際に、プリンタへのアクセス権限を持つ特定のユーザーを指定します。システム・アカウントは、通常、プリンタへのアクセス権限を持っていません。

さらに、Reports Server サービスを実行しているユーザーは、サーバーが必要とする可能性のあるすべてのサービスに対してアクセス権限を持っている必要があります。たとえば、サーバーが別のドライブへの書き込みアクセス権限を必要とする場合があります。

---

---

5. システムが起動されたら、サービスの「スタートアップの種類」を「自動」に設定します。



6. 「OK」をクリックします。
7. 「開始」をクリックします。

「サービス コントロール」メッセージ・ボックスに、Reports Server を起動した日時が表示されます。

### 2.1.3.2 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、レポート実行要求を送信するとインプロセス Reports Server が起動されます。ただし、コマンドラインを介して要求を送信する場合は、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 Reports Server の起動も成功したことになります。

URL からインプロセス Reports Server 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/rwservlet/startserver
```

### 2.1.3.3 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。

```
rwsver 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。

```
rwsver 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。

```
rwsver.sh server=server_name
```

または

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

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。

### 2.1.3.4 Reports Server の停止

この項では、Windows および UNIX での Reports Server の停止方法について説明します。

- Windows で Reports Server をサービスとして実行している場合は、「サービス」コントロール・パネルを使用して停止します。

- 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。
- Oracle Enterprise Manager 10g を起動し、停止する Reports Server にナビゲートします。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、コマンド・プロンプトで次のコマンドを入力します。

Windows および UNIX の場合 (UNIX では `rwserver` のかわりに `rwserver.sh` を使用) :

次のコマンドラインで、サーバーが正常に停止します (つまり保留中のジョブを終了した後に停止)。

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

次のコマンドラインで、サーバーが直ちに停止します (つまり保留中のジョブを終了せずに停止)。

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

次のコマンドラインで、関連メッセージを表示せずにサーバーが停止します。

```
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・オプション」](#) 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。`identifier` 要素は、Oracle Application Server 10g をインストールする際の Reports の構成時に設定され、Reports Server によって暗号化されています。この `identifier` 要素は、任意の値に変更することができます。その後、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)。

```
./rwserver.sh server=server_name shutdown=normal authid=scott/tiger
./rwrqv.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_repserv"` 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 HTTP Server が実行されていることの確認

Oracle HTTP Server が実行されていることを Oracle Enterprise Manager 10g から確認することができます。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/`

## 2.3 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/rwervlet/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、`rwervlet` 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/rwervlet/showjobs?server=server_name`

デフォルトの Reports Server 名 (`rep_machine_name`) を使用している場合、またはサーブレット構成ファイル `rwervlet.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：** サーブレット構成ファイルの詳細は、第 3 章「[OracleAS Reports Services 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 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](#)
- [UNIX での DISPLAY およびプリンタ依存性](#)
- [インプロセス・サーバーのデフォルトのプリンタの設定](#)

---

---

**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せん。環境変数の詳細は、[付録 B 「Reports 関連の環境変数」](#) 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 の各種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 および Reports Servlet 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><code>ORACLE_HOME¥reports¥conf¥server_name.conf</code></p> <p>この XML ファイルを使用して、Reports Server キャッシュ、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。</p> <p>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>このファイルの詳細は、<a href="#">Reports Server の構成</a>を参照してください。</p>
Reports Server	<p><code>ORACLE_HOME¥reports¥dtd¥rwsrverconf.dtd</code></p> <p>このファイルには、<code>server_name.conf</code> および <code>report_server_name.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。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。</p> <p>このファイルの詳細は、<a href="#">Reports Server の構成要素 (rwsrverconf.dtd)</a>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 Server Reports Builder	<p><code>ORACLE_HOME\reports\conf\rwbuilder.conf</code></p> <p>この XML ファイルを使用して、Reports Builder および Reports Runtime に埋め込まれている Reports Server を構成します。要求はすべて 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、要求を処理する Reports Server を自動的に起動します。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・サーバー・インスタンスに構成を提供します。<code>server_name.conf</code> ファイルと同様に、このファイルのデータ型定義は <code>rwserverconf.dtd</code> ファイルに依存します。ただし、<b>compatible</b>、<b>persistFile</b> および <b>security</b> の要素など、いくつかの要素を除きます。</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、<a href="#">Reports Server の構成</a> を参照してください。</p>
Reports Servlet	<p><code>ORACLE_HOME\reports\conf\rwservlet.properties</code></p> <p>このファイルでは、Reports Servlet のキー・マップ・ファイル (<code>cgicmd.dat</code>) の位置とファイル名、および Reports Servlet のインプロセス・サーバーを使用するかどうかなどを指定します。</p> <p>このファイルの詳細は、<a href="#">Reports Servlet の構成</a> 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 `report_server_name.conf` の各 XML ファイルを介して構成できます。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\server_name` or `rwbuilder.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` ファイルによってサポートされています。

---

**注意：** `rwserver.template` ファイルは、`ORACLE_HOME\reports\conf` ディレクトリにあります。

---

`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。`report_server_name.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 `report_server_name.conf` ファイルは、ほぼ同じです。唯一の違いは、`report_server_name.conf` は [compatible](#)、[persistFile](#) または [security](#) の構成要素（この項で後述）を使用しませんが、`server_name.conf` ではこれらを使用することです。

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。
- `report_server_name.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
```

### 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 2003 Oracle Corp.  
500 Oracle Parkway, Redwood Shores, CA 94065, U.S.A. All rights reserved.  
  
This is the DTD defining the Reports Server Version 9i Configuration file  
(XML) format/syntax.  
-->  
  
<!ELEMENT server (compatible?,  
                 cache?,  
                 engine+,  
                 security*,  
                 destination*,  
                 job+,
```



```

        notification*,
        log?,
        jobStatusRepository?,
        trace?,
        connection?,
        queue?,
        persistFile?,
        identifier?,
        environment*,
        pluginParam*)>
<!ATTLIST server
  version      CDATA      #IMPLIED>

<!ELEMENT cache (property*)>
<!-- class specifies fully 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 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 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      "10000">

<!ELEMENT connection (orbClient*, cluster?)>
<!ATTLIST connection
  maxConnect   CDATA      "20"
  idleTimeOut  CDATA      "10">

<!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_wrn|trace_sta|trace_exc|trace_all) "trace_all"
  traceMode      (trace_replace|trace_append) "trace_replace"
  traceModule    (all|server|engine) "all">

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

<!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
- destination
- job
- notification
- log
- jobStatusRepository
- trace
- connection
- queue
- persistFile
- identifier
- pluginParam
- environment

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サーバー構成ファイル (`server_name.conf`) に追加する場合、`rwserverconf.dtd` (`ORACLE_HOME¥reports¥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。

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。

### 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

#### 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、Reports 6i クライアント (RWCLI60、RWCGI60、RWQMU60.EXE、RWRQM60.EXE、RWRQV60.EXE、Forms 6i) 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要求をリスニングし、それを 9i サーバーに転送します。

`compatible` には、`version` という1つの属性があります (表 3-2 を参照)。

**表 3-2 compatible 要素の属性**

属性	有効値	説明
<code>version</code>	6i	<code>version</code> を 6i に設定すると、Oracle Reports で Reports 6i クライアントを実行できます。 <code>version</code> を 6i に設定する場合、6i より前のバージョンも Oracle Reports で実行できますが、保証された動作ではなく、Oracle ではサポートしていません。

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

---

**注意：** `tnsnames.ora` ファイルは、使用する Oracle Application Server の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。

---

---

**注意：** `compatible` 要素に `6i` が設定され、Reports Server に対する TNS エントリが `tnsnames.ora` ファイルに含まれている場合、サーバーがクラスタのメンバーであれば、クラスタ名を追加する必要があります。クラスタ名を使用する場合、`sqlnet.ora` ファイルで指定されたデフォルト・ドメインは除外します。次に例を示します。

```
myserver.world (standalone server with default domain world)
myserver.cluster1 (server part of cluster1)
myserver.cluster1.world (invalid entry)
```

---

---

### 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 を使用して独自の実装を定義できます。

---

---

**注意：** OracleAS Reports Services API の最新情報は、Oracle Technology Network (<http://otn.oracle.com/> 英語サイト) または Oracle Technology Network Japan (<http://otn.oracle.co.jp/> 日本語サイト) を参照してください。

---

---

`cache` 要素には、`class` という 1 つの属性があります (表 3-3 を参照)。

表 3-3 cache 要素の属性

属性	有効値	説明
class	「説明」を参照してください。	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、受信したジョブ要求がキャッシュ内の既存出力に一致するかどうか判断されます)。

```
<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="60000">
<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 (oracle.reports.engine.EngineImpl) 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：** Oracle Reports API の最新情報は、Oracle Technology Network (<http://otn.oracle.com/> 英語サイト) または Oracle Technology Network Japan (<http://otn.oracle.co.jp/> 日本語サイト) を参照してください。

---

engine には、複数の属性があります (表 3-4 を参照)。

**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」を参照してください。	<b>デフォルト：</b> 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。
classPath	文字列	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。  <b>Windows:</b>  classPath="%ORACLE_HOME%\myEngine.jar"  <b>UNIX:</b>  classPath="\$ORACLE_HOME/myEngine.jar"



表 3-4 engine 要素の属性 (続き)

属性	有効値	説明
initEngine	数値	デフォルト: 1 初期化時に Reports Server で起動するエンジンの数。
maxEngine	数値	デフォルト: 1 サーバーで実行できるこのタイプのエンジンの最大数。
minEngine	数値	デフォルト: 0 サーバーで維持されるこのタイプのエンジンの最小数。
engLife	数値	デフォルト: 50 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。
maxIdle	数値	デフォルト: 30 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)。 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。
callbackTimeOut	数値	デフォルト: 60000 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)。 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。

表 3-4 engine 要素の属性 (続き)

属性	有効値	説明
defaultEnvId	文字列	<p>オプションの属性。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、<a href="#">動的な環境切替え</a>を参照してください。</p>
engineResponseTimeOut	数値	<p><b>デフォルト:</b> 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の理由) 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>サーバー環境で実行されている Reports Engine によって、REPORTS_JVM_OPTIONS 環境変数の値は無視されます。Reports Engine による JVM オプションの選択を実行するには、jvmoptions 属性に環境変数の値を追加する必要があります。</p>

また、任意数のプロパティを engine 要素に入力できます。唯一の要件は、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。たとえば、OracleAS Reports Services で提供されるデフォルトの engine Java クラスを使用する場合、engine の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="60000">
<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・ディレクトリを使用します。

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

### 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：** 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。10g (9.0.4) 以降のリリース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：** Oracle Reports API の詳細は、Oracle Technology Network (<http://otn.oracle.com/> 英語サイト) または Oracle Technology Network Japan (<http://otn.oracle.co.jp/> 日本語サイト) を参照してください。

---

security の属性は、表 3-5 を参照してください。

**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。  id="rwSec"
class	「説明」を参照してください。	<b>デフォルト：</b> oracle.reports.server.RWSecurity  Reports Server Security Java インタフェース (oracle.reports.server.Security) 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 要素に入力することもできます。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 の構成エントリは次のようになります。

```
<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>
```

この例では、oidEntity プロパティは、10g (9.0.4) 以降の構成を示しています。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 の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_BBEFDCDAC2343600E0340800020C7BCC
```

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せん。

---

クラスタ環境でセキュリティを設定する場合、予期しない動作を防ぐため、各クラスタ・メンバーに同じセキュリティ・ポリシーを使用する必要があります。

### 3.2.1.6 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、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。OracleAS Portal を除き、プリンタ、電子メール、ファイル、キャッシュなど、提供されている（デフォルト）宛先の登録は不要です。OracleAS Reports Services Destinations API で作成した宛先タイプは登録する必要があります。

---

**注意：** Oracle Reports API の詳細は、Oracle Technology Network (<http://otn.oracle.com/> 英語サイト) または Oracle Technology Network Japan (<http://otn.oracle.co.jp/> 日本語サイト) を参照してください。

宛先の構成は、第 7 章「OracleAS Reports Services の宛先の構成」を参照してください。

---

destination の属性は、表 3-6 で説明します。

**表 3-6 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」を参照してください。	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。

### 3.2.1.7 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 要素の属性は、表 3-7 で説明します。

表 3-7 job 要素の属性

属性	有効値	説明
jobType	文字列	デフォルト: report 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。
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 <b>engine</b> 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。

### 3.2.1.8 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：** RSDK には、Oracle Reports に使用できる最新の API が含まれています。

---

notification 要素の属性は、表 3-8 で説明します。



表 3-8 notification 要素の属性

属性	有効値	説明
id	文字列	<p><b>デフォルト:</b> mailNotify</p> <p>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。</p> <pre>id="tellMe01"</pre>
class	「説明」を参照してください。	<p><b>デフォルト:</b> oracle.reports.server.MailNotify</p> <p>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。</p>

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・オプション」](#)を参照）。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。

### 3.2.1.9 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き込みを行います。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\rwserver.log
```

log 要素には、option という1つの属性があります（表 3-9 を参照）。

表 3-9 log 要素の属性

属性	有効値	説明
option	allJobs	デフォルト: noJob
	succeededJobs	デフォルトでログを記録するサーバー・アクティビティの他に、ログを記録するジョブのタイプを記述します。次のオプションがあります。
	failedJobs	
	noJob	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。

### 3.2.1.10 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。

---

**注意：** OracleAS Reports Services API の詳細は、Oracle Technology Network (<http://otn.oracle.com/> 英語サイト) または Oracle Technology Network Japan (<http://otn.oracle.co.jp/> 日本語サイト) を参照してください。

---

`jobStatusRepository` 要素には、`class` という 1 つの属性があります (表 3-10 を参照)。

**表 3-10 jobStatusRepository 要素の属性**

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

`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@ordb`) の両方の JDBC フォーマットがサポートされます。

---



---

### 3.2.1.11 trace

#### 例

```
<trace traceFile="neptune.trc" traceOpts="trace_prf|trace_dbg|trace_wrn"
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：** サーブレットと JSP のトレースは、サーブレット構成ファイル (`rwServlet.properties`) で構成します ([Reports Servlet の構成](#)を参照)。個別レポートのトレースは、レポートのランタイム・コマンドラインで指定できます ([付録 A 「コマンドライン・オプション」](#)を参照)。

---



---

`trace` の属性は、[表 3-11](#) で説明します。

**表 3-11 trace 要素の属性**

属性	有効値	説明
<code>traceFile</code>	<code>*.trc</code>	<p><b>デフォルト:</b> <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/server_name/</code></p>

表 3-11 trace 要素の属性 (続き)

属性	有効値	説明
traceOpts	表 3-12 を参照してください。	<p>デフォルト: trace_all</p> <p>この属性は、トレースするアクティビティを定義します。1 つ以上の traceOpt 値を指定できます。次に例を示します。</p> <pre>&lt;traceOpts="trace_prf trace_brk"&gt;</pre> <p>値は縦線 ( ) で区切ります。有効値は、表 3-12 を参照してください。</p>
traceMode	trace_replace trace_append	<p>デフォルト: trace_replace</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 (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。</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。traceModule を使用して、サーバー・イベントのみを追跡するか (server)、エンジン・イベントのみを追跡するか (engine)、両方を追跡するか (all) を指定できます。次に例を示します。</p> <pre>&lt;trace traceOpts="trace_all" traceModule="server"/&gt;</pre> <p>traceModule 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>

表 3-12 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 3-12 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 要素の他に 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。

複数のトレース要素を指定するときは、縦線で区切ります。次に例を示します。

```
traceOpts="trace_prf|trace_dbg|trace_wrn"
```

### 3.2.1.12 connection

#### 例

```
<connection maxConnect="50" idleTimeOut="20">
<orbClient id="RWClient" publicKeyFile="clientpub.key"/>
<cluster publicKeyFile="%ORACLE_HOME%\reports\server\serverpub.key"
privateKeyFile="%ORACLE_HOME%\reports\server\serverpri.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 3-13 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 の属性は、表 3-13 で説明します。

**表 3-13 connection 要素の属性**

属性	有効値	説明
maxConnect	数値	<b>デフォルト: 20</b>  サーバーが同時にサービスを提供できる要求の最大数。要求数が maxConnect 値を超えると、Java 例外が返されません。
idleTimeOut	数値	<b>デフォルト: 15</b>  接続のアイドル状態が許可される時間 (分)。

connection には、orbClient と cluster という 2 つのサブ要素があります。

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 要素の属性は、表 3-14 で説明します。

**表 3-14 orbClient 要素の属性**

属性	有効値	説明
id	文字列	<b>デフォルト: RWClient</b>  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	<i>filename.key</i>	<b>デフォルト: clientpub.key</b>  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。

### Microsoft Windows

```
rwgenkey.bat path_and_client_public_key_file_name
              path_and_client_private_key_file_name
```

### UNIX

```
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。

`cluster` サブ要素は、すべてのクラスタ・メンバーが使用する公開鍵ファイルおよび秘密鍵ファイルを指定します。サーバー構成ファイルには、`cluster` 要素を1つ指定するか、または指定しません。

---

---

**注意：** サーバー・クラスタの詳細は、[第12章「Reports Server のクラスタ化」](#)を参照してください。

---

---

同じクラスタのメンバーとなるサーバーは、同じ拡張クラスタ名と公開鍵および秘密鍵を持つ必要があります。つまり、拡張クラスタ名 (`serverA.clus`、`serverB.clus` の場合、`.clus` が拡張クラスタ名) が等しく、同じ公開鍵ファイルおよび秘密鍵ファイルが各クラスタ・メンバーのサーバー構成ファイル (`server_name.conf`) で指定されている必要があります。

デフォルトのサーバーの公開鍵および秘密鍵は、Windows と UNIX の両方で、`ORACLE_HOME¥reports¥jlib¥rwrwn.jar` に格納されています。ただし、クラスタで使用するように新たに生成した公開鍵および秘密鍵を `jar` 形式に変換する必要はありません。これらの鍵は任意の場所に置き、それらの絶対パスとファイル名をサーバー構成ファイルで指定します。

### 3.2.1.13 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。

---

**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 要素には、`maxQueueSize` という 1 つの属性があります (表 3-15 を参照)。

**表 3-15 queue 要素の属性**

属性	有効値	説明
<code>maxQueueSize</code>	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。

**3.2.1.14 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、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` 要素には、`fileName` という 1 つの属性があります (表 3-16 を参照)。

**表 3-16 `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>

### 3.2.1.15 `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、管理者として Reports Queue Manager に最初にログインしたときに、構成ファイルに自動的に書き込まれます。初回ログイン時、Reports Queue Manager の管理者ユーザー ID とパスワードが設定されます。この情報は暗号化された後サーバー構成ファイルに書き込まれ、その後すべての Queue Manager のログイン情報の認証に使用されます。

Queue Manager のログイン情報を忘れた場合、または変更する場合は、Reports Server 構成ファイルでこの情報を削除して次のフォーマットで再入力します。

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

この変更を行ったら、ただちに Reports Server を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。次のようになります。

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

---

**注意：** `rwservlet.properties` ファイルで `DIAGNOSTIC=NO` が設定されていれば、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、「[Reports Servlet の構成](#)」を参照してください。

### 3.2.1.16 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、OracleAS Reports Services のすべてのプラグガブル・コンポーネントと連携して動作します。プラグガブル・コンポーネントには、エンジン、セキュリティ、キャッシュ、宛先、および `jobstatusRepository` があります。これらには、メール・サーバー、FTP URL またはその他のタイプのプラグイン・パラメータへのアクセスが必要な場合もあります。`pluginParam` 要素は、すべてのプラグガブル・コンポーネントで使用できるプラグインを指定する手段を提供します。この要素を使用すると、この情報をプラグガブル・コンポーネントのクラス定義に含める必要がなく、プラグインのソースをすばやく簡単に変更できます。

たとえば、カスタムのプラグガブル・エンジンおよび宛先 Java クラスの両方にプロキシ・サーバーへのアクセスが必要であるとします。これら両方のクラスでプロキシ・サーバーへのアクセスをハードコーディングするかわりに、`proxy` など必要なプラグイン・パラメータのタイプを呼び出して、その位置をサーバー構成ファイルの `pluginParam` で示せばよいだけです。

プラグガブル・コンポーネントでサポートおよび必要とされるプラグインであり、プラグガブル・コンポーネントが認識する名前であれば、どのようなプラグイン・パラメータでも指定でき、また自由に命名できます。

pluginParam の属性は、表 3-17 で説明します。

**表 3-17 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	<b>デフォルト:</b> 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。

### 3.2.1.17 environment

#### 例

```
<environment id="JP">
  <envVariable name="NLS_LANG" value="Japanese_Japan.JA16SJIS"/>
</environment>
```

```
<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、「[engine](#)」を参照してください。
- レポートのジョブ要求のコマンドライン・オプション ENVID。特定のレポートのジョブ要求についてのみ環境設定を有効にします。

environment の属性は、[表 3-18](#) で説明します。

**表 3-18 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 要素の属性は、[表 3-19](#) で説明します。

**表 3-19 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

以前の Reports Server では、Reports Server の起動時に設定してある環境と互換性のあるレポートのみ処理できました。たとえば、レポートには、Reports Server 起動時の NLS\_LANG パラメータの値との互換性が必要でした。この制限があったため、処理する各言語についてそれぞれ Reports Server を実行しておく必要がありました。新しい環境切替え機能ではこの制限が解消され、言語など、任意の環境設定であっても Reports Server の 1 つのインスタンスでレポートの処理ができるようになり、この制限が解消されています。

この機能を使用するには、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、次のいずれかの方法で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` 属性によって、エンジン起動時に、指定の `environment` 要素がエンジンに適用されます。詳細は、「[engine](#)」を参照してください。
- レポートのジョブ要求のコマンドライン・オプション `ENVID`。特定のレポートのジョブ要求についてのみ環境設定を有効にします。詳細は、「[例](#)」を参照してください。

`environment` 要素の例を次に示します。

```
<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>
```

この `environment` 要素は、`engine` 要素から次のように呼び出すことができます。

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

値 `JP` によって、Reports Server 構成ファイルに事前に作成された `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。`environment` 要素の詳細は、「[engine](#)」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法を示しています。

#### 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 URL を使用して日本語出力のレポートを要求することができます。

```
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` パラメータが検出され、指定された ID（この例では JP）と、構成ファイルの中の対応する 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` はオプションのパラメータです。このパラメータ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要求が処理されます。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。

Reports Server がその時点で要求を処理できない場合、その要求をクラスタ上の別のサーバーに転送することがあります。ジョブがクラスタ上の別のサーバーにルーティングされた場合でも、ターゲット・サーバーは、要求された環境でエンジンを再利用または起動して、このジョブを処理します。

## 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 (Reports Servlet と組み合わせずに) 使用方法を示します。

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、アプリケーションは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、特定の環境の要求を取得するときに、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ابل・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ابل・エンジンを起動します。
- インプロセス・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、10g (9.0.4) 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 構成ファイルの中の 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、10g (9.0.4) 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 構成ファイルの中の 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` から取得しません。

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、要求を渡したときに、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` のみが使われます。

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. `environment id` タグを使用して、`reports.sh` ファイル内の環境変数の値を、`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` の値を `engine` タグに追加し、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="60000"
defaultEnvId="default">
```

5. これでレポートを実行します。

- Solaris 用 Oracle Reports の従来のリリースでは、`reports.sh` ファイルに次の行を追加する必要がありました。

```
NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P1; export NLS_LANG
```

この回避策が使用されたのは、OPMN では NLS\_LANG に AMERICAN\_AMERICA.WE8MSWIN1252 が暗黙的に設定され、NLS\_LANG が設定されていない場合に、この暗黙的な設定がエンジンに渡されるためです。これが、エンコードが存在しないために、Solaris に問題を引き起こしていました。しかし、reports.sh の NLS\_LANG 設定により、環境切替え機能が起動して、次のようなエラー・メッセージが表示されます。

```
REP-3000: Internal error starting Oracle Toolkit
```

または

```
REP-56048: Engine rwEng-0 crashed
```

この問題を回避するには、reports.sh の NLS\_LANG エントリを削除する必要があります。

### 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@orcl）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 Servlet の構成

次のパス（Windows と UNIX で共通）にある rwservlet.properties というファイルを使用して、Reports Servlet を構成します。

```
ORACLE_HOME\reports\conf\rwservlet.properties
```

サブレットでは、CGI 実装を使用していたユーザーになじみのあるコンポーネントが使用されます。

Reports Servlet 構成ファイルを使用して、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
- `rwsvrlet` を使用したレポート出力のキャラクタ・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
- `rwsvrlet` 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
- デフォルトの Reports Server の識別
- 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 / 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

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、サブレット構成ファイルでは、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 Reports Servlet ファイルで、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
```

### 3.3.1 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「レポート要求の実行」を参照してください。Reports Servlet 構成ファイルを使用して、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サーブレット構成ファイルに表示されません。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 パラメータをサーブレット構成ファイルで指定する必要があります。

### 3.3.2 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、サーブレットが要求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。

### 3.3.3 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. `rwserver.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、`rwserver` 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/rwserver/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、「`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、またはその両方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。

### 3.3.4 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、およびその他の 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
```

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

トレースは、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るよう、OracleAS Reports Services 環境にあらかじめ追加されています。

---

---

**注意：** Reports Server のトレースは、サーバー構成ファイル (`server_name.conf`) で構成します (3.2.1.11 項を参照)。個別レポートのトレースは、レポートのランタイム・コマンドラインで指定できます (コマンドライン・オプションを参照)。

---

---

Reports Servlet および JSP のランタイム情報を追跡および記録するには、サーブレット構成ファイルの 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 3-12 で説明します。

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=myrwservlet.trc
```

デフォルトの名前は `rwservlet.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
```

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、`rwservlet` 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。

### 3.3.6 rwservlet を使用したレポート出力のキャラクタ・エンコードの指定

Oracle Reports 10g (9.0.4) からは、要求 URL またはパラメータ・フォーム入力に非 ASCII エスケープ・キャラクタを使用できます。使用する場合は、あらかじめ `rwservlet.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、パラメータ値を解析するときに、`rwservlet` で必要なエンコードが使用されます。

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

- NLS キャラクタ・セット (JA16EUC など)
- IANA キャラクタ・セット (EUC-JP など)

#### 例

```
DEFAULTCHARSET=JA16EUC
```

### 3.3.7 URL に指定された HTML コードのブラウザでの実行の禁止

レポート要求 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恐れがあります。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。Oracle Reports 10g (9.0.4) からは、HTML コードを 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ため、`rwsvrlet.properties` ファイルに新しいプロパティ `ALLOWHTMLTAGS` が追加されています。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

### 3.3.8 `rwsvrlet` に対する同時接続のプール・サイズの指定

`rwsvrlet` に対して同時に接続し、ジョブ要求を送信できるユーザー数を定義することができます。その場合は、`rwsvrlet.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、同時にジョブ要求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、同時にジョブ要求を送信するアクティブ・ユーザーの数です。要求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。

---

---

### 3.3.9 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。

エラー・メッセージ・テンプレートの名前と位置を示すエントリをサブレット構成ファイルに入力します。

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
```

### 3.3.10 インプロセス・サーバーの指定

Reports Server を Reports Servlet と同じプロセスで実行することを選択した場合、`SERVER_IN_PROCESS` パラメータを適切に設定します。Reports Server をインプロセス・サーバーとして実行するには、サブレット構成ファイル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、`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。

---

### 3.3.11 デフォルトの Reports Server の識別

Reports Servlet は、`SERVER` パラメータを使用してデフォルトの Reports Server を識別します。ランタイム URL などサーバー名が指定されない場合は、ここで指定したデフォルトのサーバーが使用されます。デフォルトの Reports Server 名をサブレット構成ファイルに入力します。

次に例を示します。

```
SERVER=server_name
```

デフォルトの Reports Server がサーバー・クラスタのメンバーである場合は、次のようにクラスタ名を使用します。

```
SERVER=server_name.cluster_name
```

サーバー名とクラスタ名の組合せを URL に使用した場合、デフォルトのサーバーに送信された要求はこの特定のマシンに渡されます。このマシンが停止した場合、エラー・メッセージが返され、レポートは実行されません。クラスタ名のみを URL で指定すると、要求は任意のクラスタ・メンバーに送信され、送信先のマシンにアイドル状態のエンジンがない場合は、別のマシンに転送されます。

`rwervlet.properties` で SERVER パラメータを指定しない場合、デフォルトのサーバー名は `rep_machine_name` です。

### 3.3.12 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/rwervlet
```

このパラメータは、Reports Servlet 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。

### 3.3.13 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。

### 3.3.14 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
```

### 3.3.15 診断 / 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>/</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>
```

### 3.3.16 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、[付録 A 「コマンドライン・オプション」](#)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、サーブレット構成ファイルの 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
```

### 3.3.17 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、サーブレット構成ファイルの `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` は `YES` に設定されています。

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

## 3.4 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す。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、次のようにジョブ要素を入力します。

```
<job jobType="urlEngine"
     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 エンジンへの要求送信の詳細は、第 13 章「レポート要求の実行」を参照してください。

## 3.5 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。

---

---

2. 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。

---

---

## 3.6 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クラッシュした場合、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
```

`addNewServerTarget.sh` (UNIX の場合) または `addNewServerTarget.bat` (Windows の場合) を使用して Reports Server を OPMN に追加することができます。ただし、Reports Server を Enterprise Manager ページで表示するには、次のコマンドを使用して OPMN を再起動する必要があります。

```
opmnctl stopall  
opmnctl startall
```

### 3.6.1 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。OPMN を使用して Reports Server を構成するためには、次のものが `opmn.xml` 内に必要です。

- プロセス・モジュール。

- インプロセス Reports Server の場合、所定の Oracle Application Server Containers for J2EE コンポーネント内に Reports Server の指定が必要。
- スタンドアロン Reports Server の場合、Reports Server コンポーネント。

**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 10g 管理者ガイド』

### 3.6.1.1 プロセス・モジュール

モジュール・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が必要であることを 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>
```

### 3.6.1.2 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、Reports Server には独立した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"/>
      </dependencies>
    </process-set>
  </process-type>
</ias-component>
```



```

    <managed-process ias-component="WebCache" process type="WebCache"
      process_set="WebCache" autostart="true"/>
  </dependencies>
  <start timeout="120" retry="3"/>
  <stop timeout="120"/>
  <restart timeout="120"/>
  <ping timeout="30" interval="30"/>
</process-set>
</process-type>
</ias-component>

```

---

**注意：** 上記の例のタイムアウト値は、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：** [第 19 章「OracleAS Reports Services の管理と監視」](#)

targets.xml の詳細に関する記述があります。

```
<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 のクラッシュ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 10g 管理者ガイド』

### 3.6.1.3 インプロセス Reports Server の指定

インプロセス Reports Server 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 Reports Server 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-set .../>
  </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 id="urlping-parameters">
```

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 id="/reports/rwservlet/pingserver?start=auto" value="200"/>
```

/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 Reports Server 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。

value=200 は、ping 要求に対する応答で期待される有効な HTTP コード (200) を指定します。応答 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 UNIX での DISPLAY およびプリンタ依存性

UNIX の場合、OracleAS Reports Services の以前のリリースでは、Reports Server に対して DISPLAY 環境変数を設定することが必要でした。以前のリリースでは、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 が必要でした。この依存性は、OracleAS Reports Services 10g (9.0.4) で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。リリース 10g (9.0.4) では、OracleAS Reports Services に組み込まれているデフォルトのスクリーン・プリンタ・サーフェス (ScreenPrinter) は、プリンタが使用できない場合に、スクリーン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、OracleAS Reports Services 10g (9.0.4) では、OracleAS Reports Services を実行する場合に UNIX に有効なプリンタが存在している必要はありません。

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)

以前のリリースの従来の動作に戻すには、REPORTS\_DEFAULT\_DISPLAY=NO に設定します。

**関連項目：** [付録 B 「Reports 関連の環境変数」](#)

Oracle Reports の環境変数の詳細に関する記述があります。

### 3.7.1 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/guicommon9/tk90/admin
screenprinter.ppd : ORACLE_HOME/guicommon9/tk90/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 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。

- 有効なプリンタが存在するかどうか。
  - (YES) → ・プリンタからフォント情報を取得
  - (NO) → ・ScreenPrinter サーフェスの生成として ScreenPrinter からフォント情報を取得
    - ・ScreenPrinter の生成に失敗した場合 → REP-1800 エラー

---



---

**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：**

- 「Oracle Reports でのフォントの管理」
- 「UNIX での Oracle Reports による印刷」

UNIX でのフォントと印刷の詳細に関する記述があります。

## 3.7.2 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 3-20 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

- 拡張イメージングを有効にすることによって、HTML や HTMLCSS のタイプの出力で従来の CGM 形式や 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 を 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 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：

- [付録 B 「Reports 関連の環境変数」](#)

環境変数の詳細に関する記述があります。

- [付録 A 「コマンドライン・オプション」](#)

コマンドライン・キーワードの詳細に関する記述があります。

## 3.8 インプロセス・サーバーのデフォルトのプリンタの設定

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。これにより、次のような問題が生じます。

- 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."
```

- 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。

この問題を回避する手順は、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。

- 次に示すキーにナビゲートします。

```
HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\Windows
```

- 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,winpool,Ne02:
```

- 次に示すキーにナビゲートします。

```
HKEY_USERS\.\DEFAULT\Software\Microsoft\Windows NT\CurrentVersion\Windows
```

- 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：** この回避策は、Default Printer の値を変更するたびに実行する必要があります。

---

---



---

## 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行のデフォルト・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、「[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（「[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

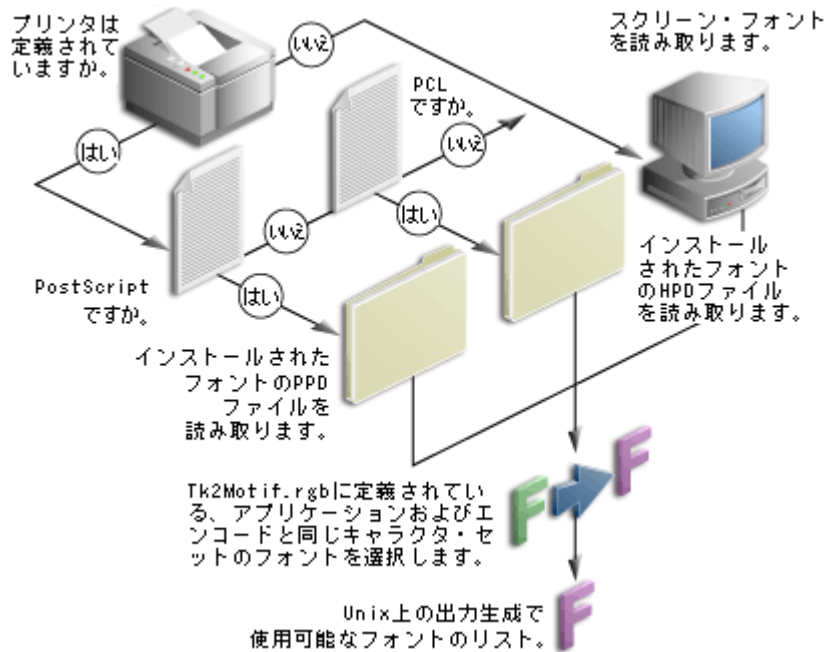
#### 関連項目：

- [「フォント構成ファイル」](#)
- [「ScreenPrinter」](#)

uiprint.txt および ScreenPrinter (screenprinter.ppd) の詳細に関する記述があります。

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：**

- 「[フォント構成ファイル](#)」
- 「[ScreenPrinter](#)」

`uiprint.txt` および ScreenPrinter (`screenprinter.ppd`) の詳細に関する記述があります。

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。何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、あるレポートが韓国語の 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。

2. 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"
```

3. 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、「[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： 第 6 章「Oracle Reports での PDF の使用」

OracleAS Reports Services での PDF 機能および拡張機能の詳細に関する記述があります。

PDF の全般的な情報は、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 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、「[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 :

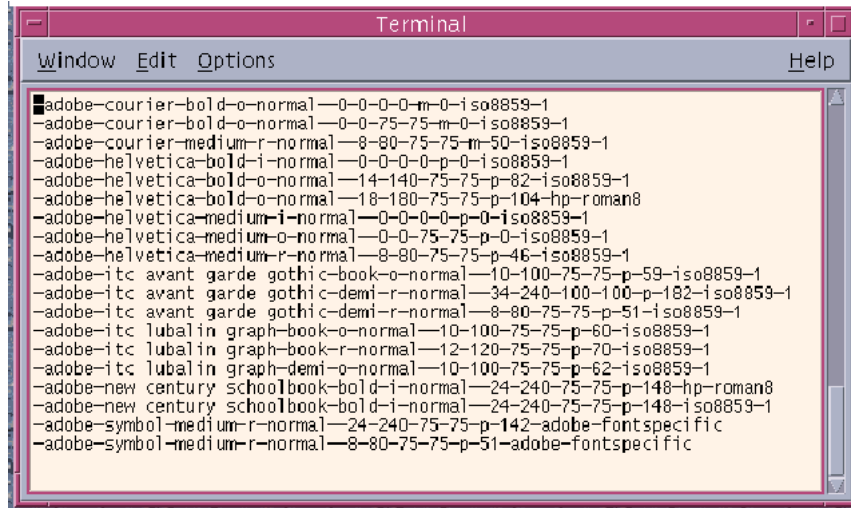
1. フォント関連のファイルをベンダーから入手します。これらのファイルには、.pfb ファイル、.pfa ファイル、AFM ファイルなどがあります。
2. 使用可能な変換プログラムを使用して、バイナリの .pfb ファイルを ASCII の .pfa ファイルに変換します。一般に、このような変換プログラムは、t1ascii などのシェアウェアとして入手できます。
3. プラットフォームごとの指示に従って、フォントのインストール先のディレクトリに .pfa ファイルをコピーします。



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、エミュレータの設定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



1. 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、「[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、「[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 Information セクションに追加します。

```
*Font new_font_name Standard '(00.1001)" Standard ROM
*Font ...
```

2. 新しいフォントの AFM ファイルを、`ORACLE_HOME/guicommon9/tk90/admin/AFM` にコピーします。AFM ファイルの名前が、PPD ファイルで指定されている `new_font_name` 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。

たとえば、PPD ファイルでの指定が `*Font CodedreineunBold:Standard '(00.1001)" Standard ROM` である場合、AFM ファイルの名前は拡張子のない `CodedreineunBold` となります。

3. 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、「[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 のインストールで使えない場合は、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。

1. 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
```

2. 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/guicommon9/tk90/admin/TFM
```

3. 必要であれば、「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

ここでは、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)

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：**

- 「[uiprint.txt ファイルの編集](#)」
- 「[uiprint.txt](#)」

uiprint.txt ファイルの詳細に関する記述があります。

**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、それぞれコロンで区切られています。

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/guicommon9/tk90/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、「[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（\*FM 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、「[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、「[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

PCL5 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。

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 である必要があり、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、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

フォント構成ファイルの検索条件は、ファイルのタイプおよびそれぞれに対応する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、TK90\_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、TK90\_PPD で指定されるディレクトリ、ORACLE\_PPD、ORACLE\_HOME/guicommon9/tk90/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 TK90\_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、MS Windows から Motif へ移行するときフォント・エイリアシングを使用すれば、MS 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。

## 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。

```
Font name = 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
Inverted	32
Blink	64



**表 4-3 太さに対応する数値**

太さ	対応する数値
Ultralight	1
Extralight	2
Light	3
Demilight	4
Medium	5
Demibold	6

**表 4-4 幅に対応する数値**

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。

PDF の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 (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、「[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、「[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\common90` (Windows)

`ORACLE_HOME/guicommon9/tk90/admin` (UNIX)

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。
Printer:PostScript t1	PostScript レベル 1 のプリンタにのみ適用されます。
Printer:PostScript t2	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	PDF のフォント・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）PDF の Type1 フォントの埋込みに使用されます。
PDF:Subset	（Oracle Reports のみ）PDF のフォント・サブセットに使用されます。

**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
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。
<i>printer_name</i>	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]

**注意：** レポート・エディタの Web ソース・ビューでは、日本語、タイ語、アラビア語およびヘブライ語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]
.....ja16sjis="MS Gothic"
.....ar8mswin1256="Courier New"
```

## 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、一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¥common90 ディレクトリ (Windows) および ORACLE\_HOME/guicommon9/tk90/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：** [第6章「Oracle Reports での PDF の使用」](#)

uifont.ali ファイルの各セクションの詳細に関する記述があります。

#### 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ためのユーティリティが `ORACLE_HOME/bin` ディレクトリにあります。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。

```
fontchk90 [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/guicommon9/tk90/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

**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を確認する方法：**

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 ファイルを見つけた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**

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....
```

## 問題と解決策

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、**Reports Builder** 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

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、量が多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。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"
.....TH8TISASCTII="Andale Duospace WT"
.....TR8MSWIN1254="Courier New"
```



## 制約事項

### Windows の場合：

- Oracle Reports では、Unicode を使用する場合の各国語のフォントと入力方式について、Windows オペレーティング・システムに依存しています。特定の言語でテキストを入力および表示するには、その言語をサポートするバージョンの Windows を実行している必要があります。フォントのサポートは限られていますが、Windows オペレーティング・システムのフォント・サポートの範囲内に制限されているわけではありません。
- 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。
- Windows NT 4.0 および以降のバージョンは、TrueType Big Font を提供しています。これらのフォントには、多言語テキストの表示や印刷に必要なすべての文字が含まれています。多言語テキストを入力、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 を使用していないことが考えられます。Microsoft 社が Windows NT 4.0 以降で提供している Big Font は、Arial、Courier New、Lucida Console および Lucida Sans Unicode です。Arial Unicode MS は適正な Big Font の 1 つで、Microsoft 社のサイトからダウンロードできます。
- Wingdings フォントは、NLS\_LANG が UTF8 の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

**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。

水平拡張度の設定を「拡張」または「可変」に変更してみてください。

**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にありません。

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。

HTMLCSS 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。

HTMLCSS 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、HTMLCSS 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た場合も、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。Solaris 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、Solaris 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（ラスターライズ済）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
- 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があります。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。**Oracle Reports** では Unicode をサポートしています。Unicode を使用すると、複数の言語を同じアプリケーションで表示できるようになります。これには、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、韓国語などのアジア系マルチバイト言語の両方が含まれます。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=JAPAN_JAPANESE.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、画面上に日本語とドイツ語を同時に表示するには、NLS\_LANG 変数を次のいずれかに設定する必要があります。

```
NLS_LANG=JAPAN_JAPANESE.UTF8  
NLS_LANG=GERMAN_GERMANY.UTF8
```

Oracle Reports では、様々な言語のフォントと入力方式について、オペレーティング・システムに依存しています。特定の言語でテキストを入力および表示するには、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する必要があります。フォントのサポートは限られています。オペレーティング・システムのフォントのみに制限されているわけではありません。Windows NT 4.0 および以降のバージョンは、TrueType Big Font を提供しています。これらのフォントには、多言語テキストの表示や印刷に必要なすべての文字が含まれています。多言語テキストを入力、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 社が Windows NT 4.0 以降で提供している Big Font は、次のとおりです。

- Arial
- Courier New
- Lucida Console
- Lucida Sans Unicode
- Times New Roman

サードパーティの Unicode フォントも利用できます。

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=character_set.UTF8
```

### 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・フォントに顕著なぎざぎざが発生しません。

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ー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、ヒンティング・コードも含まれています。ヒンティングとは、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 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、この他にバーコード用の 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。

## 4.6.6 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
- NLS のサポート
- 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

次の図は、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

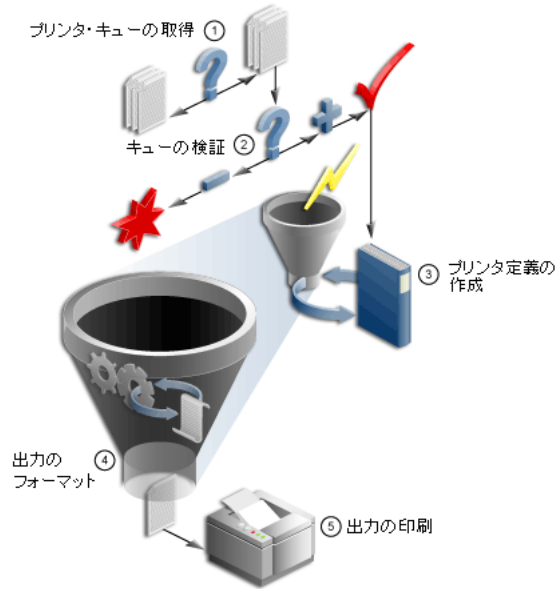
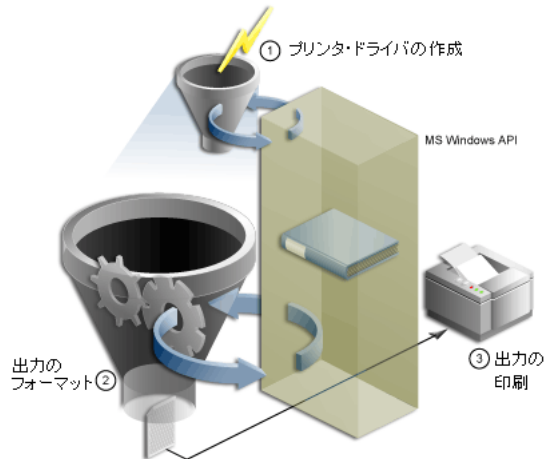


図 5-2 Microsoft Windows での Oracle Reports の印刷



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。

- 印刷要求のス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が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/guicommon9/tk90/admin/PPD
ORACLE_HOME/guicommon9/tk90/admin/HPD
```

2. プリンタにインストールされているフォントのメトリックである 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/guicommon9/tk90/admin/AFM`

`ORACLE_HOME/guicommon9/tk90/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

「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/guicommon9/tk90/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、それぞれコロンの区切られています。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><b>Solaris</b></p> <p><code>lpstat -p printername</code></p> <p><b>Linux</b></p> <p><code>lpstat -p printername</code></p> <p><b>HP-UX</b></p> <p><code>lpstat -d printername</code></p> <p><b>HP Tru64</b></p> <p><code>lpstat -p printername</code></p> <p><b>IBM AIX</b></p> <p><code>lpstat -pprintername</code></p> <p>IBM AIX では、<code>-p</code> の後に空白を入れないでください。</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"> <li>■ 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/guicommon9/tk90/admin/PPD</li> <li>■ 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/guicommon9/tk90/admin/HPD</li> <li>■ 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。</li> </ul>

### 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：「環境変数」

プリンタ関連の環境変数の詳細に関する記述があります。

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：** 付録 B 「Reports 関連の環境変数」

Oracle Reports で設定可能な環境変数の詳細に関する記述があります。

- [TK\\_PRINTER / PRINTER](#)
- [TK\\_PRINTER](#)
- [TK\\_PRINT\\_STATUS](#)
- [REPORTS\\_NO\\_DUMMY\\_PRINTER](#)
- [TK90\\_HPD](#) および [ORACLE\\_HPD](#)
- [TK90\\_PPD](#) および [ORACLE\\_PPD](#)
- [TK90\\_TFM](#) および [ORACLE\\_TFM](#)
- [TK90\\_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あ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、この他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트리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ー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 付属の一般的な PPD ファイル Oracle Reports**

PPD ファイル名	対応するプリンタ
<code>appl230.ppd</code>	Apple LaserWriter v23.0
<code>datap462.ppd</code>	Dataproducts LZR-2665
<code>declps32.ppd</code>	Digital PrintServer 40
<code>default.ppd</code>	デフォルトのレベル 1 PostScript プリンタ
<code>hpljet41.ppd</code>	HP LaserJet 4/4M PostScript 600DPI
<code>lwntx470.ppd</code>	Apple LaserWriter II NTX
<code>nccps801.ppd</code>	NEC Colormate PS/80
<code>tkphzr33.ppd</code>	Tektronix Phaser III PXi v2011.108
<code>1530_523.ppd</code>	Linotronic 530
<code>screenprinter.ppd</code>	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している必要があります。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・ファイルを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あ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せん。

- 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、「[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

PCL5 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、「[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/guicommon9/tk90/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：** [第 4 章「Oracle Reports でのフォントの管理」](#)

フォントの詳細に関する記述があります。

## 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 ファイルの編集」](#)

uiprint.txt の詳細に関する記述があります。

## 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。

### 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。

**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。Motif 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、メモリーの大きさ、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えます。

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。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、それに従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。

**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 NLS のサポート

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のみ Oracle Reports でサポートされる、新しい PASTA コンポーネント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：** [第 4 章「Oracle Reports でのフォントの管理」](#)

フォントの詳細に関する記述があります。

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 E-Business Suite とともに使用する場合、Oracle Reports では PostScript 出力にフォント埋込みユーティリティが追加できます。

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、Oracle Reports に付属している IX という新しいコンポーネントを使用します。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す。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、Solaris 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 を使用するときのセットアップおよび使用方法は、『Oracle Applications System Administrator's Guide』を参照してください。

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、セットアップ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。

- TK90\_PPD や TK90\_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 TK90_PPD(8).
slsfindfile(): environment variable not set
slsfindfile(): checking environment variable ORACLE_PPD(10).
slsfindfile(): environment variable not set
slfpath():    looking up path
              /s/rw/994/solohome/guicommon9/tk90/admin/PPD/
slfexist():   testing /s/rw/994/solohome/guicommon9/tk90/admin/PPD
slfexist():   testing /s/rw/994/solohome/guicommon9/tk90/admin/PPD/default.ppd
slsfindfile():returned
/s/rw/994/solohome/guicommon9/tk90/admin/PPD/default.ppd
slfindfile(): type = 39 (AFM)slfindfile(): name = Courier-Bold
slsfindfile(): checking environment variable TK90_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 /s/rw/994/solohome (len=18)
slfpath():    looking up path/s/rw/994/solohome/guicommon9/tk90/admin/AFM/
slfexist():   testing /s/rw/994/solohome/guicommon9/tk90/admin/AFM
slfexist():   testing /s/rw/994/solohome/guicommon9/tk90/admin/AFM/Courier-Bold
slsfindfile():returned /s/rw/994/solohome/guicommon9/tk90/admin/AFM/Courier-Bold
slfindfile(): name = uiprint.txt
slsfindfile(): checking ORACLE_HOME environment variable.
slfpath():    looking up path/s/rw/994/solohome/guicommon9/tk90/admin/
slfexist():   testing /s/rw/994/solohome/guicommon9/tk90/admin
slfexist():   testing /s/rw/994/solohome/guicommon9/tk90/admin/uiprint.txt
slsfindfile(): returned /s/rw/994/solohome/guicommon9/tk90/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](#)

## 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-03002: 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/guicommon9/tk90/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、「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。

**解決策:**

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	ファイル名 .ps	プリンタ名
印刷	Printer	プリンタ名	
Distribute=Yes			プリンタ名
Mode=Character			ファイル名 .prt

## 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セットアップする必要があります。

#### 解決策:

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、「[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、「[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。

#### 解決策

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

```

## 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、「[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・ファンクションを使用します。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。

SRW.SET\_PRINTER\_TRAY は、プリンタ・トレイの設定を行う Oracle Reports のビルトイン・ファンクションです。このファンクションの構文の説明は、Reports Builder のオンライン・ヘルプ・システム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

### 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せん。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。

**関連項目：** [第 4 章「Oracle Reports でのフォントの管理」](#)

フォントの詳細に関する記述があります。



## 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ためです。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何らかの問題が発生しますか。

OracleAS Reports Services の以前のバージョンを実行する場合、フォントを入手するために、UNIX 上に有効なプリンタが存在している必要がありました。有効なプリンタがない場合、OracleAS Reports Services では、前述の DISPLAY 変数を必要とする画面フォントが使用されていました。OracleAS Reports Services 10g (9.0.4) 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に、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、OracleAS Reports Services 10g (9.0.4) では、OracleAS Reports Services を実行する場合に UNIX 上に有効なプリンタが存在している必要はありません。

### 関連項目： 第 5 章「UNIX での Oracle Reports による印刷」

UNIX での DISPLAY およびプリンタへの依存の詳細に関する記述があります。

## 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：** [第4章「Oracle Reports でのフォントの管理」](#)

フォントの詳細に関する記述があります。

---

---

**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、プリンタにあ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あ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。

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。

- [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 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質の 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
- アクセシビリティ
- 分類
- グラフのサポートの強化

### 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

Oracle Reports では、PDF 出力の圧縮レベルを制御するためのコマンドライン・オプションを実装しています。Oracle Reports の PDF 出力はデフォルトでは圧縮されており、コマンドライン・オプションの PDFCOMP を使用して、圧縮をオフにしたり、圧縮率を変えることができます。

表 6-1 では、PDFCOMP キーワードを使用できるコマンドを示します。

**表 6-1 PDFCOMP キーワードを使用できるコマンド**

rwclient	rwrwn	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ファイルは圧縮されません。1～9の場合、PDFファイルが圧縮され、ユーザーが圧縮レベルを制御できます。

---

**注意：** レベル0～9は、圧縮の段階的なレベルです。0は圧縮がされないことを、9は圧縮の最大レベルを表します。適用する圧縮レベルが大きいほど、レポートのPDFファイル生成に要する時間が長くなります。



---

- 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

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**

Name ▲	Size	Type	Modified
 pdf_no_comp.pdf	725 KB	Adobe Acrobat Doc...	3/31/2003 5:02 PM
 pdf_yes_comp.pdf	119 KB	Adobe Acrobat Doc...	4/3/2003 3:06 PM

---

**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 でサポートされる各種のフォント関連の機能について説明します。

### 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、キャラクター・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。

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せん。

- 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 (マルチバイト・フォント、Unicode フォントの両方)。これらは、Adobe のサイトから無料ダウンロードできます。
- Type 1 Postscript フォント。
- TrueType フォント。

#### 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・オプションはありません。

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\common90` (Windows の場合)
- `ORACLE_HOME/guicommon9/tk90/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、「[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 Type 1 font */
"Kino MT" = UtopiaBold
/*Alias multibyte to available CID font */
.....SJIS = "HeiseiKakuGo-W5-Acro"
```

各項目の内容は次のとおりです。

- "Kino MT" = UtopiaBold では、Oracle Reports で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、Oracle Reports で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 で、「ファイル」→「文書のプロパティ」→「フォント」をクリックします。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

PDF のフォント・サブセットは、フォントの埋込みの変形です。このオプションには、文書で実際に使用されている絵文字と文字情報だけが含まれます。

フォント・サブセットは、そのフォントがインストールされているマシンに依存しません。PDF のフォント・サブセットは、シングルバイト・フォントとマルチバイト・フォントのどちらでも使用可能で、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。

---

---

**注意：** フォント・サブセットは、絵文字とフォント・データが PDF ファイルに含まれている場合にのみ機能します。

---

---

#### 6.1.2.2.1 設定

フォント・サブセットには、コマンドライン・オプションはありません。

フォント・サブセット機能を使用する前に、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。

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] です。

uifont.ali ファイルのエントリは次のとおりです。

```
[PDF:Subset]
Font Name = "Font File Name"
```

ファイル名のエントリは、引用符で囲みます。一方、フォント名は複数の単語を含む場合にのみ、引用符で囲みます。

フォント・サブセットが正常に動作するには、uifont.ali ファイル内で参照されるフォント・ファイル名が、既存のフォント・ファイル名と一致している必要があります。

---



---

**注意：** このフォント・ファイル名は、Reports Builder で表示されるフォント名ではありません。

---



---

### 例

```
[PDF:Subset]
Arial = "Arial.ttf"
```

---



---

**注意：** Oracle Reports では、TTC だけでなく、TTF もサポートします。

---



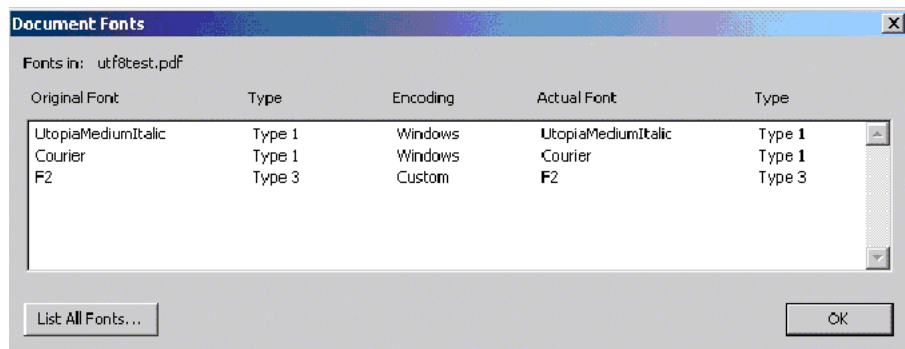
---

PDF ファイル内のフォントのサブセットを作成すると、そのフォントは、レポート出力に必要な文字だけを含むので、カスタム・フォントになります。

Adobe Acrobat 3.0 以降では、レポートで使用されているフォントを、次の方法で表示できます。

- 「ファイル」→「文書のプロパティ」→「フォント」。
- 「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「カスタム」と表示されます。
- 「実際のフォント」列に「F2」と表示されます。
- 「種類」列に「Type3」と表示されます。

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。

---

---

### 推奨

最適な表示を得るには、Adobe Acrobat Reader 5.0 を使用し、Adobe Acrobat Reader のパラメータを次のように設定することをお勧めします。

1. 「編集」→「環境設定」。
2. 「表示」→「スムージング」。
3. 「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。
4. (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。
5. 「OK」をクリックします。

Oracle Reports の PDF フォント・サブセットは、TrueType フォントでのみ機能します。Type1 フォントでは機能しません。特定の TrueType 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、Type1 フォントを TrueType フォントに変換します。

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。この問題を回避する手順は、次のとおりです。

1. TrueType フォントを UNIX マシンにコピーします。
2. REPORTS\_PATH 環境変数で、正しいフォント・パスが指定されていることを確認します。
3. 関連する TrueType フォントを含むレポートを実行します。UNIX マシンにおけるレポートの実行の詳細は、「[レポートを Windows で設計し、UNIX に配布する場合](#)」を参照してください。

**6.1.2.2.2 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。

- Acrobat Reader で、「ファイル」→「文書のプロパティ」→「フォント」をクリックします。「実際のフォント」値が「カスタム」、「種類」が「Type3」であることを確認します。そのフォントが指定されていない場合は、フォント・サブセットが作成されていません。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・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表示するには、最低でもこのバージョンが必要です。
  - フォント・サブセットが太字で表示されていること。これは、**Type3** フォントに当てはまります。**Type3** フォントは、フォント・サブセットとともに作成されます。**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、「[推奨](#)」を参照してください。
- UNIX マシンにおけるレポートの実行の詳細は、「[レポートを Windows で設計し、UNIX に配布する場合](#)」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（シングルバイトおよびマルチバイトのフォント）だけが対象であり、**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・オプション: **PDFEMBED**
- **uifont.ali** ファイル・エントリのエントリ、[PDF:Embed]

コマンドライン・オプション 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。

---



---

**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にのみ機能します。

---



---

### PDFEMBED

表 6-2 では、PDFEMBED キーワードを使用できるコマンドを示します。

**表 6-2 PDFEMBED キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	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

### 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"

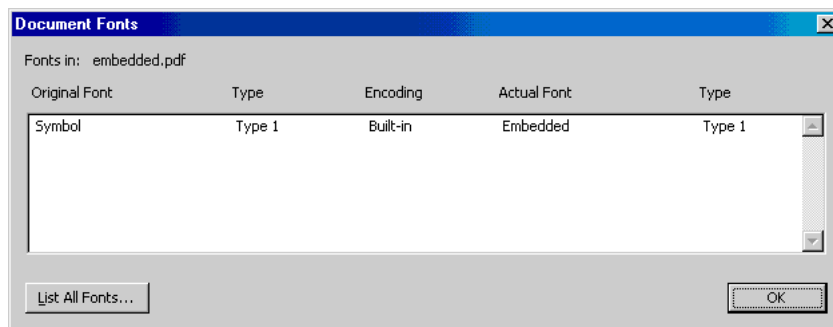
### 例 6-1 フォントの埋込み

```
[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 で、「ファイル」→「文書のプロパティ」→「フォント」をクリックします。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。

### 6.1.3 実行の優先順位

uifont.ali ファイル内の複数の箇所に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. フォントの埋込みがフォント・サブセットより優先される (中間の優先順位)
3. フォント・サブセットは優先されない (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・エンタリを含めた場合は、フォントの埋込みが優先されます。これには、コマンドライン・オプションの PDFEMBED=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)、特殊なエンタリを最初に指定してから、汎用エンタリを指定します。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、エンタ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" エンタリが最初にあると、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 エンタリにも、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"
```

表 6-3 各種 PDF タイプの比較

PDF タイプ	長所	短所
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 キャラクター・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・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。	表示品質が不十分 (*)。 スタイル（斜体と太字）をサポートしていない。 生成した PDF ファイルを Acrobat Reader で編集できない。

(\*) 修正方法の詳細は、「[推奨](#)」を参照してください。

## 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。出力にアクセシビリティ機能を含めるコマンドライン・オプションは ACCESSIBLE です。

表 6-4 では、ACCESSIBLE キーワードを使用できるコマンドを示します。

**表 6-4 ACCESSIBLE キーワードを使用できるコマンド**

rwclient	rwrwn	rwbuilder	rwconverter	rwservlet	rwcgi	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 ファイルで有効化されるか (YES) または無効化されるか (NO) を指定します。

**構文** ACCESSIBLE={YES|NO}

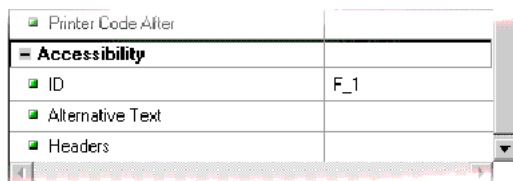
### 値

- YES の場合、アクセシビリティの機能が PDF ファイルで有効化されます。
- NO の場合、アクセシビリティの機能が PDF ファイルで無効化されます。

**デフォルト** NO

追加プロパティ (図 6-4) を使用すると、ユーザー補助が利用可能な PDF 文書の生成に必要な情報を提供することができます。

**図 6-4 アクセシビリティ**



## 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 6-5 分類プロパティ**

プロパティ名	タイプ	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
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

---

**注意：** 分類プロパティは、レポート・レベルのプロパティです。

---

分類プロパティの詳細は、Reports Builder のオンライン・ヘルプを参照してください。

## 6.1.6 グラフのサポートの強化

Oracle Reports では、PDF のグラフに対するサポートを強化しました。これには、グラフのイメージ解像度を上げるために、より大きなインチあたりのドット数（DPI）が指定可能になったことが含まれます。グラフのサポートが強化されたことにより、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るようになりました。

そのための環境変数は、REPORTS\_GRAPH\_IMAGE\_DPI です。

**説明** REPORTS\_GRAPH\_IMAGE\_DPI 環境変数は、プリンタにグラフを出力（PDF、PostScript または PCL）する際の DPI 値を指定します。この環境変数を使用すると、初期のデフォルトの固定値である 72 DPI を大きくすることができます。これにより、印刷されるグラフ（PDF、PostScript または PCL）のイメージ解像度を上げることができます。

**有効値** 72 ~ 300

**デフォルト** 250

#### 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。
- 高解像度のグラフを生成すると、PDF ファイルのサイズが相対的に大きくなり、PDF ファイルやプリンタ出力の生成に要する時間も長くなります。
- DPI 値を 250 以上に設定し、グラフの大きさを 5"x5" 以上にした場合は、JVM の Out of Memory エラーを回避するために、`REPORTS_JVM_OPTIONS` で JVM ヒープ・サイズの値を変更することも必要になります。JVM オプションの設定の詳細は、「[REPORTS\\_JVM\\_OPTIONS](#)」を参照してください。
- 元の設定に戻すには、この環境変数の値を 72 DPI に設定します。
- この変数は、PDF 出力、PS 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。

## 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 レポートを Windows で設計し、UNIX に配布する場合

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。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 Solaris などの 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・フォントの場合の手順例

この項では、Windows 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、Solaris などの UNIX プラットフォームで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。

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/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. 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。
4. 作成した AFM ファイル (`arial.afm` および `tahoma.afm`) を `ORACLE_HOME/guicommon9/tk90/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. TK\_PRINTER 環境変数または PRINTER 環境変数に hrprinter など、プリンタの名前が設定されていることを確認します。

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：** 追加対象となる ppd ファイルは uiprint.txt ファイルで指定されていて、フォント情報を格納しているものです。デフォルトでは、このような ppd ファイルとして datap462.ppd が設定されています。

---

---

```
*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 で、「ファイル」→「文書のプロパティ」→「フォント」または「ファイル」→「文書情報」→「フォント」をクリックすると、次のようになります。

- 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 で、「ファイル」→「文書のプロパティ」→「フォント」をクリックします。
- b. 「オリジナルフォント」列に「F2」、「エンコーディング」列に「カスタム」、「種類」列に「Type3」がそれぞれ表示されているはずです。

### 6.2.2.2 マルチバイト・フォントまたは Unicode フォントの場合の手順例

マルチバイト・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。マルチバイトのレポートを UNIX プラットフォームに配布する場合の、PDF での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 h2mj<sub>sm</sub>.ttf と KO16KSC5601 キャラクタ・セットです。
2. 韓国語フォント (h2mj<sub>sm</sub>.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/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 h2mj<sub>sm</sub>.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/guicommon9/tk90/admin/AFM/.
% cp h2mjsm.afm ORACLE_HOME/guicommon9/tk90/admin/AFM/h2mjsm
```

5. TK\_PRINTER 環境変数または PRINTER 環境変数に hrprinter など、プリンタの名前が設定されていることを確認します。
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：** 追加対象となる ppd ファイルは uiprint.txt ファイルで指定されていて、フォント情報を格納しているものです。デフォルトでは、このような ppd ファイルとして datap462.ppd が設定されています。

---



---

```
*DefaultFont: h2mjsm
*Font h2mjsm: 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¥common90 (Windows の場合)
- ORACLE\_HOME/guicommon9/tk90/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 ]
.....k016ksc5601 ="h2mjsm"
[ Printer:PostScript2 ]
.....k016ksc5601 ="h2mjsm"
```

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 で、「ファイル」→「文書のプロパティ」→「フォント」または「ファイル」→「文書情報」→「フォント」をクリックすると、次のようになります。
- a. 「オリジナルフォント」列に h2mjsm 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 ]
"h2mjsm"="h2mjsm.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 で、「ファイル」→「文書のプロパティ」→「フォント」または「ファイル」→「文書情報」→「フォント」をクリックします。
  - b. 「オリジナルフォント」列に「F2」、「エンコーディング」列に「カスタム」、「種類」列に「Type3」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。この問題は回避できません。

フォント・サブセットを有効にして生成した PDF ファイルは、**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）。これは、**Type3** フォントを表示する際の **Acrobat Reader** の制限が原因で発生します。**Type3** フォントの表示をスムージングする方法の詳細は、「[推奨](#)」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。

### 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 `C:\WINNT\Fonts` など、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。
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\common90` (Windows の場合)
  - `ORACLE_HOME/gui\common90\tk90\admin` (UNIX の場合)
- 
- 

#### 例

```
[ PDF:Subset ]
"Andale Duospace WT J" = "Aduoj.ttf"
"Albany WT J"="AlbanWTJ.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 `DESTTYPE=FILE DESFORMAT=PDF` で実行します。

## 6.4 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。

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:** [「REPORTS\\_BIDI\\_ALGORITHM」](#)

環境変数の詳細に関する記述があります。

### 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:** [「REPORTS\\_ARABIC\\_NUMERAL」](#)

環境変数の詳細に関する記述があります。

### 6.4.1 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 `C:¥WINNT¥Fonts` など、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。
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¥common90` (Windows の場合)
  - `ORACLE_HOME/guicommon9/tk90/admin` (UNIX の場合)
-

## 例

```
[PDF:Subset]
"Andale Duospace WT J" = "Aduoj.ttf"
"Albany WT J"="AlbanWTJ.ttf"
```

- 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。
- 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。

図 6-5 は、PDF の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表を示しています。

図 6-5 PDF の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 キャラクター セット名	CMap 名	CID フォント名
日本語	JA16SJIS	90ms-RKSJ-H	"KozMinPro-Regular-Acro" (*) "HeiseiKakuGo-W5-Acro" (**) "HeiseiMin-W3-Acro" (**)
	JA16EUC	EUC-H	
韓国語	KO16KSC5601	KSC-EUC-H	"HYSMyeongJoStd-Medium-Acro" (*) "HYGoThic-Medium-Acro" (**) "HYSMyeongJo-Medium-Acro" (**)
	KO16MSWIN949	KSCms-UHC-H	
繁体字中国語	ZHT32EUC	CNS-EUC-H	"MSungStd-Light-Acro" (*) "MHei-Medium-Acro" (**) "MSung-Light-Acro" (**)
	ZHT16BIG5, ZHT16MSWIN950	ETen-B5-H	
	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 バージョン 5 で利用できます。

(\*\*) これらのフォントは Adobe Acrobat Reader バージョン 4 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、Acrobat Reader 5.0 には Version 5 CIDFonts (\*) の使用をお勧めします。

## 6.5.1 フォント・エイリアシング

フォント・エイリアシング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. 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¥common90 (Windows の場合)
  - ORACLE\_HOME/guicommon9/tk90/admin (UNIX の場合)
- 
- 

### 例

```
[ PDF ]
.....JA16SJIS = "KozMinPro-Regular-Acro"
"MS UI Gothic".....JA16SJIS = "KozMinPro-Regular-Acro"
```

1. 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。
2. DESTYPE=FILE DESFORMAT=PDF を指定してレポートを実行します。
3. 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。

### 6.5.1.1 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 C:¥WINNT¥Fonts など、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。
3. ORACLE\_HOME¥tools¥common90 ディレクトリの 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。

### 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。

### 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\common90 (Windows の場合)
  - ORACLE\_HOME/guicommon9/tk90/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。

#### 6.6.1.1 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. ORACLE\_HOME\tools\common90 ディレクトリの uifont.ali ファイルを開き、[PDF:Subset] セクションで、次のエントリを追加して TrueType フォントの名前を指定します。

**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。

---

## OracleAS Reports Services の宛先の構成

レポートを実行する際には、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](#)

## 7.1 出力処理の概要

レポートの出力方法は、実行時に指定する `destype` (宛先タイプ) によって制御されます。指定可能な宛先タイプは、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。ユーザーは出力タイプを登録できず、OracleAS Reports Services から提供される次のデフォルト・タイプを使用します。

- キャッシュ (つまりブラウザ)
- SMTP 対応電子メール
- ファイル
- プリンタ
- OracleAS Portal (ポータルへアクセスするには、サーバー構成ファイルにユーザー ID とパスワードを指定する必要があるため、このタイプは例外です)

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、必要な任意の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：** カスタムの宛先タイプは、OracleAS Reports Services Destinations API で作成します。Oracle Reports API とダウンロード用宛先タイプの最新情報は、Oracle Technology Network (<http://otn.oracle.com/> 英語サイト) または Oracle Technology Network Japan (<http://otn.oracle.co.jp/> 日本語サイト) を参照してください。

---

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 Reports Engine 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。

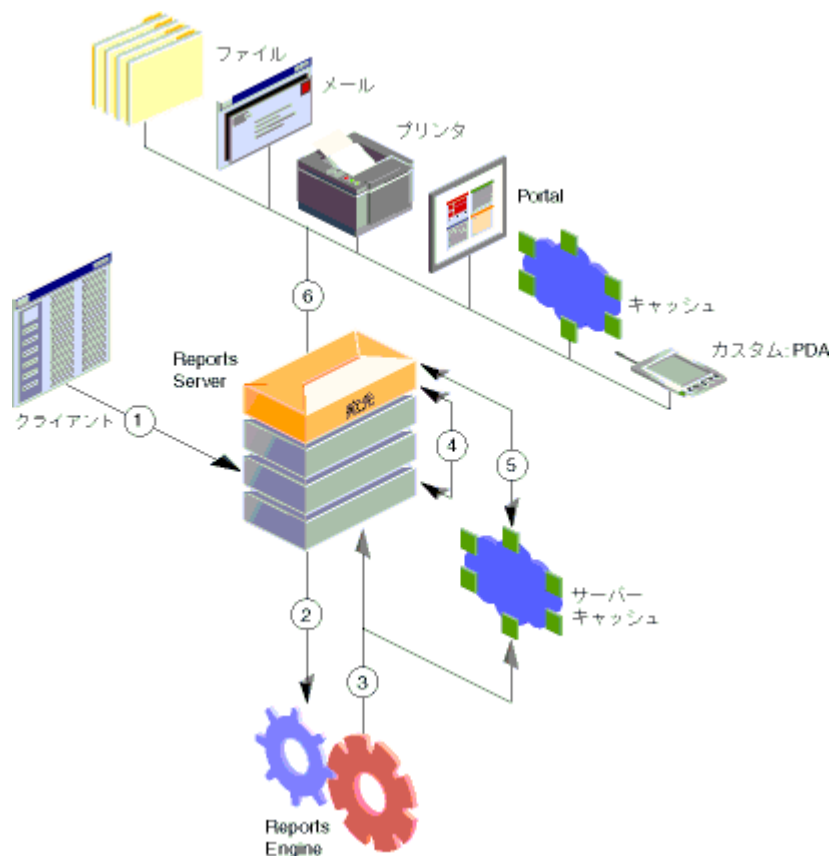
以前は、Reports Runtime エンジンが出力の配信を全面的に処理していました。このため、出力先との通信方法を Runtime エンジンで認識する必要がありました。この結果、エンジンとサポートされている宛先は密接に結び付いていました。

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 エンジンは、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 7-1 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 7-1 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、要求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 に要求を送信します。
2. サーバーがこの要求を 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。

## 7.2 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。

### 7.2.1 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。

## 7.2.2 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。

### 7.2.2.1 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 7-1 に示します。

表 7-1 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

デフォルト・タイプのうち、oraclePortal 宛先タイプだけは登録する必要があります。これは、oraclePortal 宛先タイプがポータルにアクセスするためのユーザー ID とパスワードの指定を必要とするためです。

**関連項目：** 付録 A 「コマンドライン・オプション」

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 API の最新情報は、Oracle Technology Network (<http://otn.oracle.com/> 英語サイト) または Oracle Technology Network Japan (<http://otn.oracle.co.jp/> 日本語サイト) を参照してください。

---

### 7.2.2.2 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ของผู้ใช้ 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\dttd\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がきます。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、出力タイプの設定のみに関係します。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。

### 7.2.3 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 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](#)
- [トラブルシューティング情報](#)
- [ユーザー独自の PDS の追加](#)

## 8.1 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 Japan (<http://otn.oracle.co.jp>) 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.jdbcpds.
        JDBCConnectionHandling">
</driver>

<driver name = "sqlserver-merant"
    sourceDatabase = "sqlserver"
    subProtocol = "merant:sqlserver"
    connectionString = "mainProtocol:subProtocol://databaseName"
    class = "com.oracle.ias.jdbc.sqlserver.SQLServerDriver"
    connection = "oracle.reports.plugin.datasource.jdbcpds.
    JDBCConnectionHandling">
</driver>

<driver name = "sybase-merant"
    sourceDatabase = "sybase"
    subProtocol = "merant:sybase"
    connectionString = "mainProtocol:subProtocol://databaseName"
    class = "com.oracle.ias.jdbc.sybase.SybaseDriver"
    connection = "oracle.reports.plugin.datasource.jdbcpds.
    JDBCConnectionHandling"
    loginTimeout = "0">
</driver>

<driver name = "db2-merant"
    sourceDatabase = "db2"
    subProtocol = "merant:db2"
    connectionString = "mainProtocol:subProtocol://databaseName"
    class = "com.oracle.ias.jdbc.db2.DB2Driver"
    connection = "oracle.reports.plugin.datasource.jdbcpds.
    JDBCConnectionHandling"
    loginTimeout = "0">
</driver>

<driver name = "informix-merant"
    sourceDatabase = "informix"
    subProtocol = "merant:informix"
    connectionString = "mainProtocol:subProtocol://databaseName"
    class = "com.oracle.ias.jdbc.informix.InformixDriver"
    connection = "oracle.reports.plugin.datasource.jdbcpds.
    JDBCConnectionHandling">
</driver>

</driverInfo>
</jdbcpds>
```

表 8-1 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 8-1 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、 <i>jdbc:subProtocol://databaseName</i> 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. c.informix.InformixDriver

表 8-1 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.JDBCConnectionHandling</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 ドキュメントを参照してください。</p>	<code>oracle.reports.plugin.datasource.jdbcpds.JDBCConnectionHandling</code>
loginTimeout (オプション)	<p>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。</p>	0
property	<p>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。</p>	-

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わせられます。この組み合わせた接続情報が、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 8-3、表 8-4、表 8-5、表 8-6、表 8-7、表 8-8 および表 8-9 を参照してください。

図 8-1 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 8-1 Reports Builder の JDBC 接続ダイアログ



## 8.1.1 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。

---

---

## 8.1.2 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-J (Oracle Technology Network Japan) (<http://otn.oracle.co.jp>) 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 (表 8-1) 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。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. **report\_server\_name.conf:** report\_server\_name.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 8-1 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。

### 8.1.2.1 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、YMutil.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。
  - 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¥YMutil.jar;D:¥sybase\_installed¥YMsybase.jar;D:¥sybase\_installed¥YMbase.jar;existing classpath entries
  - b. **report\_server\_name.conf:** report\_server\_name.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¥Mutil.jar;D:¥sybase_installed¥Msybase.jar;D:¥sybase_
installed¥Mbase.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¥Mutil.jar;D:¥sybase_installed¥Msybase.jar;D:¥sybase_
installed¥Mbase.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 8-1 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>
```

### 8.1.2.2 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、`YMutil.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。

- 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¥YMutil.jar;D:¥db2_installed¥YMdb2.jar;D:¥db2_
installed¥YMbase.jar;existing classpath entries
```

- b. **report\_server\_name.conf:** report\_server\_name.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¥YMutil.jar;D:¥db2_
installed¥YMdb2.jar;D:¥db2_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"
maxEngLife="1" minEngine="0" engLife="50" maxIdle="30"
callbackTimeOut="60000" classPath="D:¥db2_installed¥YMutil.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 8-1 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>
```

### 8.1.2.3 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、YMutil.jar、YMsqserver.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。
- 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¥Mutil.jar;D:¥sqlserver_
installed¥Msqlserver.jar;D:¥sqlserver_installed¥Mbase.jar;existing
classpath entries
```
  - report\_server\_name.conf:** `report_server_name.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¥Mutil.jar;D:¥sqlserver_installed¥Msqlserver.jar;D:¥sqlserver_
installed¥Mbase.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¥Mutil.jar;D:¥sqlserver_installed¥Msqlserver.jar;D:¥sqlserver_
installed¥Mbase.jar;">
...
</engine>
```
  - 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 8-1](#) を参照してください。
 

**例:**

```
<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.jdbcpds.
JDBCConnectionHandling">
</driver>
```

### 8.1.2.4 Informix 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、YMutil.jar、YMinformix.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。

- 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:¥informix_installed¥YMutil.jar;D:¥informix_
installed¥YMinformix.jar;D:¥informix_installed¥YMbase.jar;existing classpath
entries
```

- b. **report\_server\_name.conf:** rwbuilder.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>
```

- 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:¥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 8-1 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>

```

### 8.1.2.5 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。

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. report\_server\_name.conf: report\_server\_name.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 8-1](#) を参照してください。

例：

```
<driver name = "<driver name>"
      sourceDatabase = "<sourceDatabase>"
      subProtocol = "<subProtocol>"
      connectString = "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。

---

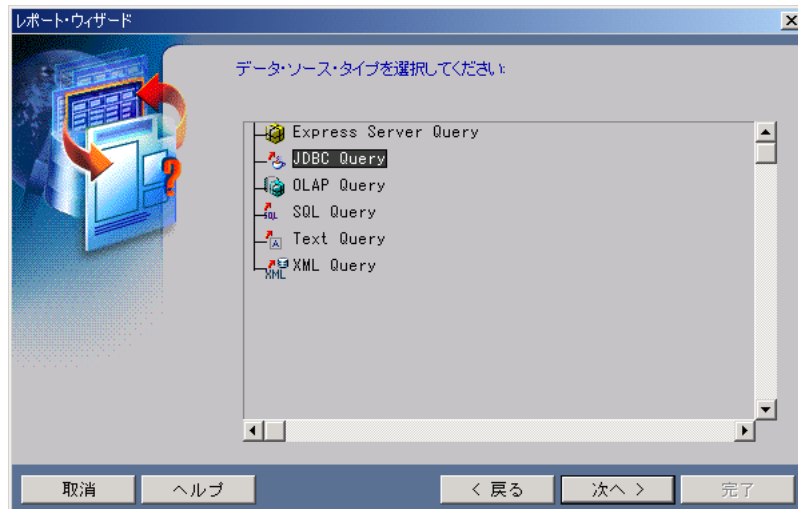
## 8.2 JDBC 問合せの定義と実行

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、Reports Builder オンライン・ヘルプを参照してください。

図 8-2 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:

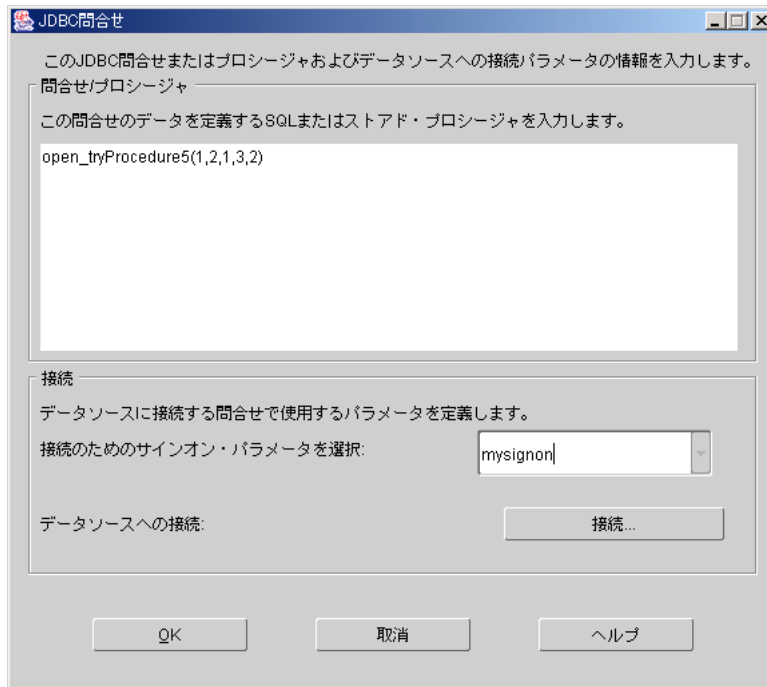
使用するデータベースのストアド・プロシージャの完全なコール構文を入力します。次に例を示します。

```
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 8-3 ストアド・プロシージャのコール



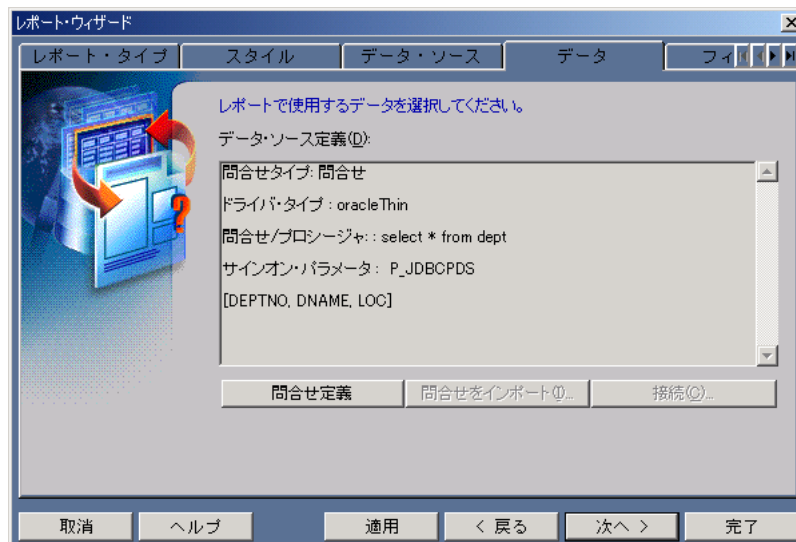
- 表 8-2

表 8-2 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)
<p>SELECT * FROM [SHEET1\$] または SELECT COL1, COL2, ...COLn FROM [SHEET1\$]</p> <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><b>注意:</b>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>	<p>SELECT * FROM [WORKSHEETNAME\$]</p> <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><b>注意:</b>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・パラメータ値は `p_jdbcpds` です。
  - 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 8-3、表 8-4、表 8-5、表 8-6 および表 8-7 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 8-1)
    - \* `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 8-4)。

図 8-4 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。

## 8.2.1 接続情報の例

表 8-3、表 8-4、表 8-5、表 8-6、表 8-7、表 8-8 および表 8-9 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 Japan (<http://otn.oracle.co.jp>) から入手可能な DataDirect Merant ドライバ。

**表 8-3 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 8-4 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 8-5 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。



表 8-5 JDBC-ODBC ドライバ (続き)

プロパティ	値
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 8-6 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 8-7 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 8-8 SQL Server

プロパティ	値
ユーザー名	Reports
パスワード	Welcome

表 8-8 SQL Server (続き)

プロパティ	値
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 8-9 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

## 8.3 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際には、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、JDBC 問合せに対して設計時に定義されます。

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
```

各項目の内容は次のとおりです。

- 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。
- `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。

レポートの設計時に「JDBC 問合せ」ダイアログ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_jdbcpds` が使用されます。

## 8.4 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)

### 8.4.1 エラー・メッセージ

表 8-10、表 8-11 および表 8-12 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 8-10 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 8-11 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。
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 8-12 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트리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트리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。

## 8.4.2 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 8-1 は、設計時のトレース出力の例です。

例 8-2 は、実行時のトレース出力の例です。

## トレース出力の例

### 例 8-1 「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.JDBCConnectionH
andling</connectionClass><SignOnParameter>P
JDBC PDS</SignOnParameter><jdbcElements><element name = "EMPNO" type = "2" type Name
= "NUMBER" columnSize = "4" columnScale = "0" /><element name = "ENAME" type =
"12" type Name = "VARCHAR2" columnSize = "10" columnScale = "0" /><element name =
"JOB" type = "12" type Name = "VARCHAR2" columnSize = "9" columnScale = "0"
/><element name = "MGR" type = "2" type Name = "NUMBER" columnSize = "4"
columnScale = "0" /><element name = "HIREDATE" type = "93"
type Name = "DATE" columnSize = "16" columnScale = "0" /><element name = "SAL"
type = "2" type Name = "NUMBER" columnSize = "7" columnScale = "2" /><element name
= "COMM" type = "2" type Name = "NUMBER" columnSize = "7" columnScale = "2"
/><element name = "DEPTNO" type = "2" type Name = "NUMBER" columnSize = "2"
columnScale = "0" /></jdbcElements><referenceColumns></referenceColumns></jdbc pds>
[2003/3/31 6:35:46:383] Debug 50103 (jdbcpds): JDBCUIEventHandler :handleOKEvent END
```

### 例 8-2 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><
onnectionClass>oracle.reports.plugin.datasources.jdbcps.JDBCConnectionHandling</conn
ectionClass>...
```

**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 andresultset 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 with actual 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 string to 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 and result set object is com.oracle.ias.jdbc.base.BaseResultSet@56c3cf
[2003/4/7 5:26:7:767] Debug 50103 (jdbcpds): JDBCDataSource : execute : END
```

## 8.5 ユーザー独自の PDS の追加

Oracle Reports では、PDS API を公開しており、Reports SDK チュートリアルも用意しています (<http://otn.oracle.com/products/reports/apis/index.html>)。このチュートリアルでは、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。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。

PDS を追加するには、次の作業が必要です。

- PDS の登録
- `jdbcpds.conf` ファイルの構成
- PDS 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



## 8.5.1 PDS の登録

PDS を設定ファイルに登録して、Reports Builder および OracleAS Reports Services から使用できるようにします。

**表 8-13 設定ファイル**

ファイル	格納場所	オペレーティング・システム
cauprefs.ora	ORACLE_HOME	Windows
prefs.ora	ORACLE_HOME	UNIX

設定ファイルは、Oracle Reports またはカスタム・クラス・ファイルの、ファクトリ・クラスの実装を参照する必要があります。Reports Builder では、ファクトリ・クラスが次のように登録されている場合にのみ、関連する PDS アイコンが表示されます。

```
Reports.PluggableDataSourceFactories =
("oracle.reports.plugin.datasource.xmlpds.XMLDataSourceFactory",
"oracle.reports.plugin.datasource.jdbcpds.JDBCDataSourceFactory",
"oracle.reports.plugin.datasource.textpds.TextDataSourceFactory",
"oracle.reports.plugin.datasource.myPDS.myPDSDataSourceFactory",
"oracle.dss.pds.bibbeans.xrpds.XRPDSFactory",
"oracle.dss.pds.snapi.expresspds.ExpressPDSFactory")
```

PDS を追加する手順は次のとおりです。

1. 設定ファイル (表 8-13) を任意のテキスト・エディタで開きます。
2. Reports.PluggableDataSourceFactories を見つけます。
3. 次の構文を使用して、任意 w の JDBC ファクトリ・クラスの名前を指定します。

```
Reports.PluggableDataSourceFactories = ("pluginClassname"[,
"pluginClassname"]...)
```

pluginClassname は、必要な実装を含むファクトリ・クラスの名前です。

---

**注意：** カスタム PDS クラスを実装する方法の詳細は、OTN (<http://otn.oracle.com>) の「Oracle Reports SDK」を参照してください。

---

4. 変更を有効にするために、Reports Builder および OracleAS Reports Services を再起動します。

## 8.5.2 jdbcpds.conf ファイルの構成

jdbcpds.conf ファイルの構成方法の詳細は、[JDBC 構成ファイル](#)の項を参照してください。

## 8.5.3 PDS JAR ファイルのインストール

PDS JAR ファイルのパスを指定します。この指定により、すべての関連クラスが Reports Builder および Reports Server から使用できるようになります。

### 8.5.3.1 Reports Builder

Oracle Reports の環境変数 `REPORTS_CLASSPATH` の前に、依存するすべての JAR ファイルのパスを挿入します。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:¥mypds.jar;existing classpath entries`

### 8.5.3.2 report\_server\_name.conf

`report_server_name.conf` 構成ファイル内にあるエンジンの `classPath` 属性に、JAR ファイル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:¥mypds.jar;">
...
</engine>
```

### 8.5.3.3 Reports Server

Reports Server 構成ファイル内にある `classPath` 属性に、JAR ファイル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:¥mypds.jar;">
...
</engine>
```

## 8.5.4 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、[カスタム・ドライバ](#)の項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](#)

## 9.1 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](#)

### 9.1.1 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要求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）

#### 9.1.1.1 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：**「[OracleAS Reports Services セキュリティの構成](#)」

インタフェースの詳細に関する記述があります。

**9.1.1.2 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。

ある特定のレポートを指定の 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：**「[OracleAS Reports Services セキュリティの構成](#)」

インタフェースの詳細に関する記述があります。

**9.1.1.3 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を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、「[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、「[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：**「[OracleAS Reports Services セキュリティの構成](#)」

インタフェースの詳細に関する記述があります。

## 9.1.2 認証とアクセス権の適用

レポート要求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要求処理](#)
- [OracleAS Single Sign-On を使用しないレポート要求処理](#)

### 9.1.2.1 OracleAS Single Sign-On を使用するレポート要求処理

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要求を受け取ると、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：** レポート要求が 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。

---

---

**9.1.2.1.1 Single Sign-On を使用するレポート要求フロー** このシナリオでは、レポート要求が 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。

1. レポートが OracleAS Portal 内から実行される場合、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。

レポートが OracleAS Portal 内から実行されない場合、ユーザーは、Web ページ上のリンクやブックマークなどを介して、レポート要求を送信する URL にアクセスする必要があります。

2. レポート要求は、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。
  - 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。

---

---

3. 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。
4. 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。
5. 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。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 9-1 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。

---

**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 9-1 Oracle Delegated Administration Services の「リソースの作成」

リソースのアクセス情報

Username

Password

Database

取消 送信



6. Reports Servlet が URL (SSOCONN を使用した場合は Oracle Internet Directory 情報からも) からコマンドラインを作成し、これを Reports Server に渡します。
7. 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。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。
8. Reports Server がこのジョブを Reports Engine 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。
9. 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、この出力はプリンタに送信されるか、ブラウザに返されます。または、後で参照するためにファイルに保存されます。

### 9.1.2.2 OracleAS Single Sign-On を使用しないレポート要求処理

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 Reports6i 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、レポート要求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。1 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、ユーザーは要求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・オプションを使用できます。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 を含めることができます。このファイルは通常 cgicmd.dat という名前で、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。

---

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、rwservlet.properties ファイル内で COOKIEXPIRE パラメータを指定できます。Reports Servlet でジョブ要求が受信されると、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、ユーザには認証情報を提供する必要が生じます。

**関連項目：**「[Reports Servlet の構成](#)」

COOKIEEXPIRE パラメータと `rwervlet.properties` ファイルの詳細に関する記述があります。

**9.1.2.2.1 OracleAS Single Sign-On を使用しないレポート要求フロー** このシナリオでは、レポート要求が 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。

1. 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、レポート要求を送信する URL にアクセスし、そこで URL を選択する必要があります。
2. 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 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 と Reports Authid Cookie が両方とも見つからない場合、Reports Servlet は、「システム認証」画面を表示します。ここに、ユーザーが各自の Single Sign-On ユーザー名およびパスワードを入力する必要があります。この情報は、Reports Authid Cookie に格納されます。
3. 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、その一部の証明書をフィールドに示した「データベース認証」ページが表示されます。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、`rwervlet.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まったく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：**「[Reports Servlet の構成](#)」

`DBAUTH` パラメータと `rwervlet.properties` ファイルの詳細に関する記述があります。

4. 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。
5. 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。また、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。
6. データソース証明書は、後で参照するために Reports Userid Cookie に保存されます。プラグابل・データソース証明書は、Reports Userid Cookie には保存されないことに注意してください。

7. Reports Server がこのジョブを Reports Engine 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。
8. 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、この出力はプリンタに送信されるか、ブラウザに返されます。または、後で参照するためにファイルに保存されます。

### 9.1.3 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。

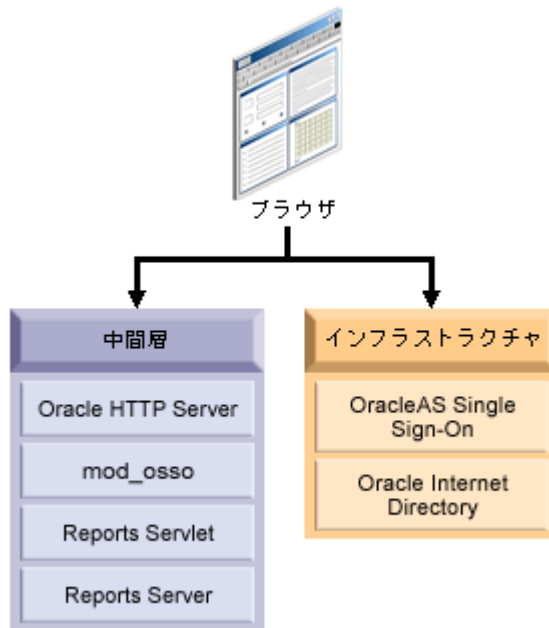
#### 9.1.3.1 OracleAS Single Sign-On

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。

**9.1.3.1.1 Single Sign-On コンポーネント** [図 9-2](#) 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 9-2 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・ブラウザからの要求を処理します。

- **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 がレポート要求を受信すると、Reports Servlet がジョブ要求を Reports Server に渡します。

- **Reports Server**

Reports Server は、クライアント要求を処理します。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 10 章「OracleAS Single Sign-On の構成と管理」を参照してください。

## 9.2 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。

### 9.2.1 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（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 10 章「OracleAS Single Sign-On の構成と管理」および第 11 章「OracleAS Portal 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、Oracle Reports6i 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。

---

---

### 9.2.1.1 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タル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。

OracleAS Single Sign-On および OracleAS Portal の詳細は、『Oracle Application Server 10g 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-J (Oracle Technology Network Japan) (<http://otn.oracle.co.jp>) に掲載されています。

詳細は、第 11 章「OracleAS Portal でのレポートの配布」を参照してください。

### 9.2.1.2 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、次のことができるようになります。

- 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

このチュートリアルと RSDK の詳細情報は、Oracle Technology Network にあります。





---

## 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](#)
- [OracleAS Forms Services のセキュリティに関する考慮事項](#)

## 10.1 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11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、付録 A「コマンドライン・オプション」では、SSOCONN コマンドライン・オプション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り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。

---

---

## 10.2 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、「[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、[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ます。Reports Servlet で利用可能なコマンドの詳細は、[rwservlet](#) を参照してください。SSOCONN の詳細は、[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 でのレポートの配布」](#)

OracleAS Portal と OracleAS Reports Services との統合の詳細に関する記述があります。

## 10.3 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の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

### 10.3.1 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 (`rwervlet.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 (Oracle Reports6i で使用される認証メカニズム)。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。

---

---

### 10.3.2 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:** 10g (9.0.4) 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。10g (9.0.4) 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 つのデフォルトのジョブ要素から参照されます。

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

### 10.3.3 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。

独自のプラグガブル・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。

#### 10.3.3.1 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・ファイルを通じて参照する) 必要があります。

*key\_name/data\_source\_type/parameter\_name*

`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 への入力」

Oracle Internet Directory へのリソースの入力の詳細に関する記述があります。

`data_source_type` は、接続するデータソースの種類です。Oracle Reports では、`data_source_type` を使用して、`key_name` に関連付けられている文字列の形式が認識されません。`data_source_type` 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 (`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：**「新しいリソース・タイプの追加」

リソース・タイプの追加の詳細に関する記述があります。

`parameter_name` 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 `parameter_name` として指定する必要があります。

**10.3.3.1.1 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
```

**10.3.3.1.2 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=exppds.rdf&destype=cache&ssocnn=exptest1/EXPRESSPDS/express_server&desformat=html
```

**10.3.3.1.3 JDBC プラグابل・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 PDS の構成と使用](#)」

JDBC データソースの構成方法の詳細に関する記述があります。

### 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。
  - `rwsvlet.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。

### 10.3.3.2 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](#)

**10.3.3.2.1 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。

---

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、「設定」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 10-1 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「設定」タブが表示されないことに注意してください。

図 10-1 Delegated Administration Services の「設定」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。

**10.3.3.2 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 10-2 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 10-2 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。

---

---

**10.3.3.2.3 バッチ・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
```

**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:\ora9ias\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:\ora9ias\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:\ora9ias\BIN>ldapadd -D cn=orcladmin -w orcladmin's_password -h host_name
-p port_num -f createresource.ldif
```

<sup>1</sup> `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。

**10.3.3.2.4 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. 「設定」サブ・タブをクリックすると、[図 10-3](#) のようなページが表示されます。

**図 10-3 Oracle Internet Directory の「構成」タブの「設定」ページ**



1. 「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。
2. 「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、「JDBC PDS」を選択します。

3. 「次へ」をクリックします。
4. 接続情報を入力します。たとえば、`scott/tiger@mydb` と入力します。
5. 「実行」をクリックします。
6. 「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。

### 10.3.3.3 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. 「設定」サブ・タブをクリックすると、[図 10-3](#) 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 10-4](#) のようなページが表示されます。

図 10-4 「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 10-1 では、これらのフィールドについて説明します。

表 10-1 「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 data_source_type 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	このフィールドは、空白のままにします。

表 10-1 「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 Express 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 10-2) 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 10-2) 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>

表 10-1 「リソース・タイプの作成」プロパティ (続き)

プロパティ	説明
追加フィールド 1 ~ 3	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 10-2) 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。

- 「実行」をクリックします。リソース・タイプが作成され、SSOCONN 引数の data\_source\_type 部分で参照できるようになります。

**関連項目：** [「SSOCONN」](#)

SSOCONN の詳細に関する記述があります。

## 10.3.4 Oracle Internet Directory への接続

第 9 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](#)

### 10.3.4.1 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。

### 10.3.4.2 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 10g 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。

## 10.4 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 which represents the logical grouping of all report instances
changetype: modify
add: uniquemember
uniquemember: dn of the Reports Application Entity
```

各項目の内容は次のとおりです。

```
dn of the group which represents the logical grouping of all report
instances=cn=Virtual Application Group, orclApplicationCommonName=reports_
application_entity_name, cn=Reports, cn=Products, cn=OracleContext
```

および

```
dn of the Reports Application Entity =orclApplicationCommonName=2, cn=Reports,
cn=Products, cn=OracleContext
```



## 10.5 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。

```
rwserver server=Reports 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、レポート要求は、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 10-2 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。

<sup>2</sup> `reports_application_entity_name` は、`reportsApp_hostname GUID` の形式です。たとえば、`reportsApp_serv1.us.oracle.com_C7543D42A9E26726E034080020A46EE2` のように使います。

表 10-2 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。

- OracleAS Portal での Reports ユーザーおよび名前付きグループの作成
- OracleAS Portal への Oracle Reports コンポーネントの登録

レポートを配布するには、事前に OracleAS Portal と Oracle Application Server Reports Services の両方がインストールされ、構成されていることが必要です。

**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」。Oracle Application Server 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 10g 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 Technology Network Japan (<http://otn.oracle.co.jp>)

## 11.1 OracleAS Portal での 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 (OID) に登録し、それら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済みである場合は、Oracle Internet Directory (OID) の機能を使用してディレクトリ間の同期をとることで、ユーザーを 1 人ずつ入力する手間が省けます。OID 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、Oracle Application Server 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、`report.sh` に NLS\_LANG を明示的に設定することでこの問題を回避できます。新しい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もできます。詳細は、「[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](#)

## 11.1.1 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。

### 11.1.1.1 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。

### 11.1.1.2 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。

### 11.1.1.3 RW\_DEVELOPER

RW\_DEVELOPER グループには、RW\_POWE\_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。

### 11.1.1.4 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、Oracle Application Server Reports Services から完全なエラー・メッセージを受け取ることができます。

---

## 11.1.2 ユーザーとグループの作成

OracleAS Portal は、Oracle Internet Directory (OID) の Delegated Administration Service (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 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。

## 11.2 OracleAS Portal への 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必要があります。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。

### 関連項目：「レポート要求の実行」

OracleAS Portal でのレポートの実行の詳細に関する記述があります。

この項では、次の操作に必要な手順について説明します。

- [使用可能日カレンダーの作成](#)
- [プリンタの登録](#)
- [Reports Server の登録](#)
- [レポートの登録](#)

### 11.2.1 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。

### 11.2.1.1 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、「Reports カレンダー・アクセス」 ポートレットの「Reports のシンプル・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「完了」 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 Oracle Application Server 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。

### 11.2.1.2 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. 「Reports カレンダー・アクセス」ポートレットの「Reports の結合されたカレンダー・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するか (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] を押しながらかlick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プレビューします。

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. 「完了」 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：**「[管理](#)」ポータルレット

「開発」タブ、「管理」タブおよび「アクセス」タブなど、「管理」ポータルレットに表示されるフィールドや説明に関する記述があります。

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 Oracle Application Server 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。

## 11.2.2 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。レポート要求を実行する 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 11-1 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. 「Reports プリンタ・アクセス」 ポートレットの「Reports 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:** 「使用可能日カレンダーの作成」

使用可能日カレンダーの作成の詳細に関する記述があります。

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:** 「管理」ポートレット」

「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に関する記述があります。

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。

## 11.2.3 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 11-2 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
Reports サーバー名	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/

表 11-2 サンプル値 (続き)

プロパティ	サンプル値
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

この項では、Reports Server を OracleAS Portal に登録する方法を説明します。

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. 「Reports Server アクセス」 ポートレットの「Reports Server 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。
  - 「Reports Server 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` などで、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：** [第 3 章「OracleAS Reports Services の構成」](#)

仮想パスの指定の詳細に関する記述があります。

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、レポート要求を送信するユーザーが、この Reports Server に対するアクセス権限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] を押しながらかlickするか（Windows の場合）、click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：** [第7章「OracleAS Reports Services の宛先の構成」](#)

カスタムの宛先タイプの詳細に関する記述があります。

11. 「次へ」をクリックします。
12. (オプション) この Reports Server でレポート要求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：** [「使用可能日カレンダーの作成」](#)

使用可能日カレンダーの作成の詳細に関する記述があります。

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：** [「管理」ポートレット](#)

「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に関する記述があります。

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。

## 11.2.4 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、レポート要求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 11-3 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 11-3 サンプル値 (続き)

プロパティ	サンプル値
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. 「Reports 定義ファイル・アクセス」 ポートレットの「Reports 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不可)。
    - \* 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]** を押しながらかlick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：

- [付録 B 「Reports 関連の環境変数」](#)

Oracle Reports 関連の環境変数の詳細に関する記述があります。

- [第 3 章 「OracleAS Reports Services の構成」](#)

仮想パスの指定の詳細に関する記述があります。

- (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。
    - 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。「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。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。

- 「書式」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、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・フォームのルック・アンド・フィール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：**「[使用可能日カレンダーの作成](#)」

使用可能日カレンダーの作成の詳細に関する記述があります。

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. 「完了」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 でのレポートの公開](#)」

OracleAS Portal からレポートを実行する方法の詳細に関する記述があります。

- 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 11-4 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 11-4 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	このレポート要求を受け取る 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。

表 11-4 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
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。
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、「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要求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、コマンドライン要求または 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。



## 11.2.5 「管理」ポートレット

既存の Oracle 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 11-5 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 11-5 「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。  <b>注意:</b> 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。
実行リンク (一部のポートレットには適用不可)	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。  <b>注意:</b> 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。
PL/SQL ソース (一部のポートレットには適用不可)	「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。  <b>パッケージ仕様:</b> ポートレットの PL/SQL 仕様を表示します。 <b>パッケージ本体:</b> ポートレットの PL/SQL 本体を表示します。

表 11-5 「開発」タブ (続き)

フィールド	説明
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 スタアド・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><b>注意:</b> 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)	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。
編集	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。
新規として編集	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。
データの編集 (一部のポートレットには適用不可)	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。
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><b>注意:</b> 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	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)。
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><b>注意:</b> 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「参照」が表示されます。</p>
お気に入り追加	Oracle 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。
バージョン情報	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。
削除	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。

表 11-6 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 11-6 「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要求とその要求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 11-7、表 11-8、表 11-9、表 11-10、表 11-11、表 11-12 および表 11-13 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 11-7 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。  <b>注意:</b> 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 11-8 権限モード

フィールド	説明
プロバイダから権限を継承	<p>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。</p> <p>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、さまざま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。</p> <p><b>注意:</b> 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。</p>

表 11-9 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 11-10 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 Oracle 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 Oracle 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 Oracle Portal ユーザーまたは Oracle 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 11-11 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、さまざま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><b>注意:</b> 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 11-12 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 11-13 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。



---

## Reports Server のクラスタ化

クラスタとは、複数のサーバーを1つのコミュニティに仮想的にグループ化したものです。このコミュニティの目的は、要求処理をクラスタの各メンバーに効率的に分散することです。OracleAS Reports Services では、ピアレベルのクラスタ化を採用しています。つまり、受信要求の分散と処理に対して、クラスタの全メンバーが同等の役割を担います。このため、あるメンバーが停止しても、残りのメンバーが要求負荷の管理を続行します。またあるメンバーのキャッシュに存在する出力を、他のメンバーが使用することもできます。1台のマシンの障害が原因でシステム全体がダウンする単一の障害箇所はありません。

この章では、クラスタへのサーバーの登録に関する情報、およびサーバーのクラスタ化による利点について説明します。この章には、次の項があります。

- [クラスタの概要](#)
- [クラスタのセットアップ](#)

## 12.1 クラスタの概要

たとえば、次のようなサーバーがあるとします。

```
serverA.cluster1
serverB
serverC.cluster1
```

`serverA.cluster1` と `serverC.cluster1` は、`cluster1` と呼ばれる同じクラスタのメンバーです。これらのサーバーは、クライアントからの要求を協同して処理します。クライアントが同期要求を `serverA.cluster1` に送信したときに、このサーバーにそのジョブ・タイプのアイドル・エンジンがない場合、このサーバーは、`serverC.cluster1` で処理できるかどうかをチェックします。`serverC.cluster1` にアイドル・エンジンがあれば、`serverA.cluster1` は、処理のために要求を `serverC.cluster1` に渡します。

この例では、スタンドアロン・サーバーである `serverB` は、処理要求を他のサーバーから受け取ることも、他のサーバーに送信することもできません。

1つのクラスタにいくつでもサーバーを追加できます。あるクラスタ・メンバーが停止すると、未処理の同期ジョブはクラスタ内の他のサーバーにリダイレクトされます。クラスタ内で1つでもサーバーが稼働していれば、クラスタは処理を続けます。

クラスタが次のスケジュール済要求または即時要求を処理するメンバーを決定する際、次の基準に従って優先順位を付けます。

1. クラスタ内のいずれかのサーバーに、要求と一致する情報がキャッシュされているか。
2. 現在、同様のジョブがキューにあるか。
3. そのジョブ・タイプのアイドル・エンジンが利用可能か。
4. 現在アクティブなエンジンの数は、そのジョブ・タイプに対してサーバーで指定されている `MAXENGINE` の数に達していないか。

スタンドアロンのサーバーとクラスタ化されているサーバーの基本構成は、どちらも同じです。クラスタでは、共通のクラスタ名と共通の公開鍵および秘密鍵を共有すればよいだけで、特別な構成要件はありません。使用するプラットフォーム、クラスタ内のサーバー数、あるいはサーバーの位置には何の制約もありません。クラスタ・サーバー間でリソースを共有する必要もありません。

エンジン出力はクラスタ内の特定の **Reports Server** でローカルにキャッシュされますが、クラスタ全体でその存在を認識しており、それを使用できます。あるサーバーがダウンすると、そのサーバーのキャッシュ・ファイルは再利用できなくなります。つまり、出力を取得するには、クラスタ内の別のサーバーが要求に応答する必要があります。キャッシュは永続的に格納されるので、サーバーが再開すると、すべてのキャッシュ・ファイルが利用可能になります。



## 12.2 クラスタのセットアップ

OracleAS Reports Services でのクラスタ化は簡単に行うことができます。つまり、`server_name.cluster` または `server_name.xyz` のように、すべてのメンバー・サーバーに同じドット拡張子を持つ名前を付け、すべてのクラスタ・メンバーが同じ公開鍵および秘密鍵を共有すればよいだけです。

この項では、Reports Server 名の変更、公開鍵および秘密鍵の作成と指定、およびクラスタへの要求送信について説明します。ここには、次の項があります。

- Reports Server 名の変更
- 公開鍵および秘密鍵の新規生成
- 公開鍵および秘密鍵のサーバー構成ファイルへの入力
- Reports Server の再起動
- クラスタへの要求送信

### 12.2.1 Reports Server 名の変更

この項では、少なくとも 1 つの Reports Server がすでにセットアップされているものとします。この場合、クラスタ名をサーバー名に追加するには、サーバー名を変更する必要があります。

---

**注意：** サーバーがインストールされていない場合は、インストール時に、同じクラスタ名を持つ別々の名前を各サーバーに付けます。たとえば、`servernameA.cluster1`、`servernameB.cluster1` などです。

---

Reports Server 名を変更する手順は次のとおりです。

1. サーバーが実行中の場合は、次のように停止します。
  - Windows のサービスとして実行中の場合は、「サービス」コントロール・パネルで停止します。
  - Windows または UNIX のコマンドラインから実行中の場合は、コマンド・プロンプトで次の Windows または UNIX のコマンドを入力します。

次のコマンドによって、サーバーが正常に停止します。

```
rwserver server=server shutdown=normal authid=admin/pword
```

次のコマンドによって、サーバーが直ちに停止します。

```
rwserver server=server shutdown=immediate authid=admin/pword
```

次のコマンドで、関連メッセージを表示せずにサーバーが停止します。

```
rwserver server=server shutdown=normal authid=admin/pword 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・オプション」](#)で説明します。

1. Reports Server 構成ファイル (`server_name.conf`) にカスタムの構成設定が含まれている場合、このファイル名を新しいクラスタ・サーバー名 (`server_name.cluster_name.conf`) に変更します。

構成ファイルは、UNIX および Windows 上の次のパスにあります。

```
ORACLE_HOME\reports\conf\server_name.conf
```

Reports Server 構成ファイルにカスタムの構成設定が含まれていない場合は、名前を変更したサーバーを再起動すると、新しい名前を持つ新しい構成ファイルが自動的に生成されます。

2. 影響を受けるすべてのファイルで Reports Server 名を変更し、各クラスタ・メンバーに同じクラスタ名を指定します。

- サブレット構成ファイル (`rwervlet.properties`) を開き、クラスタ名を含むようサーバー名を変更します。次に例を示します。

```
SERVER=server_name.cluster_name
```

サブレット構成ファイルは、Windows および UNIX 上の次のパスにあります。

```
ORACLE_HOME\reports\conf\rwervlet.properties
```

- Windows サービスとして実行しているサーバー名を変更する場合は、サービスをアンインストールしてから再びインストールする必要があります。

Windows サービスをアンインストールするには、コマンド・プロンプトで次のように入力します。

```
rwserver -uninstall server_name
```

Windows サービスを再びインストールするには、コマンド・プロンプトで次のように入力します。

```
rwserver -install server_name.cluster_name
```

---

**注意：** 再インストール後、サーバーは自動的に起動します。すべてのサーバー・クラスタ・メンバー名の変更が終わるまでサーバーを停止させておき、クラスタのセットアップが完了した時点ですべて一緒に起動することもできます。サーバーの停止の詳細は、[第 2 章「OracleAS Reports Services の起動と停止」](#)を参照してください。

---

Reports Server を再起動する前に、サーバーの公開鍵および秘密鍵を生成し、その情報を各メンバー・サーバーの構成ファイルに入力できます。この手順は、次の各項で説明します。

## 12.2.2 公開鍵および秘密鍵の新規生成

サーバーの公開鍵および秘密鍵のファイルは、クラスタ・メンバー間のメッセージ暗号化および認証に使用されます。デフォルトのファイルは、次のパスにある `rwrun.jar` ファイルに格納されています (UNIX と Windows の両方で共通)。

```
ORACLE_HOME¥reports¥jlib¥rwrun.jar
```

クラスタの各メンバーの構成ファイル (`server_name.cluster_name.conf`) には、同じ公開鍵および秘密鍵のファイルを指定する必要があります。クラスタ・メンバーが排他的な公開鍵および秘密鍵のファイルを共有できるようにするには、クラスタをセットアップする際に新しいバージョンのファイルを生成します。クラスタ・メンバー以外のサーバーは、OracleAS Reports Services で提供されるデフォルトの鍵を使用して処理を続行できます。

新しい公開鍵および秘密鍵のファイルを生成するには、コマンド・プロンプトで次のコマンドを入力します。

```
rwgenkey.sh public_key_file_name private_key_file_name (UNIX)
```

```
rwgenkey.bat public_key_file_name private_key_file_name (Windows)
```

これらのファイルを特定のディレクトリに生成するには、コマンドラインで新しい公開鍵および秘密鍵のファイルの名前とともに必要なパスを指定します。コマンドラインでファイル名のみを指定すると、鍵ファイルは現行ディレクトリに生成されます。

## 12.2.3 公開鍵および秘密鍵のサーバー構成ファイルへの入力

新しい公開鍵および秘密鍵のファイルを生成した後、その情報をクラスタの全メンバーの Reports Server 構成ファイルに入力する必要があります。このファイルの各クラスタ・メンバーのバージョンは、UNIX と Windows の両方とも、各サーバーのホスト・マシン上の次のパスに置かれています。

```
ORACLE_HOME¥reports¥conf¥server_name.cluster_name.conf
```

公開鍵および秘密鍵のファイルを変更するには、サーバー構成ファイルの `connection` 要素で、次のように `cluster` サブ要素のエントリを変更または追加します。

```
<cluster publicKeyFile="path and filename of new public key" privateKeyFile="path and filename of new private key">
```

`connection` 要素の詳細は、第 3 章「OracleAS Reports Services の構成」を参照してください。

## 12.2.4 Reports Server の再起動

クラスタ・メンバー名を変更し、共通の公開鍵および秘密鍵をそれぞれに再指定した後、Reports Server を再起動してクラスタを有効化できます。

Reports Server を起動する手順は次のとおりです。

- Reports Server を Windows サービスとして起動する場合は、「サービス」コントロール・パネルでサービスを起動します。
- Reports Server をコマンドラインから起動する場合は、コマンド・プロンプトで次のように入力します。

Windows の場合：

```
rwserver server=server_name.cluster_name
```

UNIX の場合：

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

クラスタ・メンバー名を変更し、公開鍵および秘密鍵を再指定し、Reports Server を再起動したら、クラスタのセットアップ・プロセスは完了です。

## 12.2.5 クラスタへの要求送信

クラスタに要求を送信する手順は次のとおりです。

Reports Servlet または JSP で、次のように指定します。

```
server=cluster_name
```

たとえば、mercury.cluster1、venus.cluster1 という名前の 2 台のクラスタ・マシンがある場合、server を次のように指定します。

```
server=cluster1
```

Reports Servlet または JSP は、クラスタ内で実行中の Reports Server を検出し、その Reports Server に要求を送信します。キャッシュとの一致状況やサーバーの負荷状況によって、その Reports Server は要求を処理するか、クラスタ内の別のサーバーにリダイレクトします。

# 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 「レポート要求の実行」](#)
- [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」](#)



# 13

---

## レポート要求の実行

この章では、**Reports Server** にレポート要求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](#)
- [レポート要求の方法](#)
- [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要求の指定](#)
- [URL エンジンへの要求送信](#)
- [Web サービスを介したレポートの実行](#)
- [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/rwervlet?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

構成要素	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 が要求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>rwervlet</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/rwervlet?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・マッピングの使用方法の詳細は、「[キー・マップ・ファイルの使用方法](#)」を参照してください。



JSP ファイルに Web レイアウトとペーパー・レイアウトの両方が含まれている場合に、サブレットの使用によって JSP レポート・ファイルが使用できなくなることはありません。レポートを実行する際に、URL にサブレットを指定し、次のコマンドライン・オプションで JSP をコールします。report=myreport.jsp

次に例を示します。

```
http://neptune.world.com:80/reports/rwervlet?report=myreport.jsp&destype=cache&desformat=html
```

コマンドラインの詳細は、付録 A 「コマンドライン・オプション」を参照してください。

---

**注意：** これらのパラメータは、JSP ファイルの内部に指定することもできます。

---



---

**注意：** Web Cache を停止または無効にし、Reports Servlet からブラウザに対して pagestream を有効にしたレポートを実行した場合、レポート出力が正しく表示されない場合があります。httpd.conf (Oracle HTTP Server 構成ファイル) 内の port および listener プロパティは同じポートを参照していないため、予測される動作としてこのようなことが起こります。

Oracle Reports に要求を送信するときに、WebCache を迂回、停止または無効にする場合は、httpd.conf ファイルで Port の値を Listen と同じ値に変更する必要があります。

---

## 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**

構成要素	説明
web_server	Oracle HTTP Server をインストールしたときに各自で付けた名前。
domain_name	所属する組織のドメイン名。
port	Oracle HTTP Server が要求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。
alias	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。

表 13-2 JSP ベース・レポートの URL の構成要素 (続き)

構成要素	説明
<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 (あるいはその両方)。

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 を使用する場合に、Reports Servlet を同時に使用できないわけではありません。レポートを実行する際に、URL にサーブレットを指定し、次のコマンドライン・オプションで JSP をコールします。 `report=myreport.jsp`

次に例を示します。

```
http://neptune.world.com:80/reports/rwervlet?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・オプション」](#) を参照してください。 `cgicmd.dat` ファイルの詳細は、「[キー・マップ・ファイルの使用方法](#)」を参照してください。

### 13.1.3 CGI

---

**注意：** Reports CGI は、下位互換性を確保する目的で Oracle Application Server Reports Services に用意されている機能です。レポートを配布する場合は、サーブレット実装または JSP 実装のいずれかの方法を強くお勧めします。

---

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**

構成要素	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 が要求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 レポート要求の方法

レポート要求は、様々な方法で実行することができます。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要求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` コマンドライン要求は、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、「[コマンドライン・オプション](#)」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` コマンドライン要求に変換され、その後 Oracle Application Server 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要求を行うのに必要な 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。

### ■ 経由 OracleAS Portal

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、レポート要求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、「[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、Reports Builder 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。

## 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、「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、「[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。

---

---

#### 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。

---

---

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/rwservlet?server=server_
name&report=myreport.rdf&userid=username/password@my_db&desformat=pdf
```

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、「[Web ブラウザからのレポート要求の指定](#)」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、Reports Engine を介して実行されません。
- Web レイアウトを使用した JSP レポートは、J2EE コンテナを介して実行されます。

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 Reports 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。
3. 手動または Oracle JDeveloper などのツールを使用して、新規 WAR 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. [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、「OC4J\_BI\_Forms」をクリックします。
4. 「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、[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。
5. 「J2EE アプリケーション」に前項で作成した EAR ファイルの位置を入力します。
6. 「アプリケーション名」に、MyReportApp などのアプリケーション名を入力し、「続行」をクリックします。
7. 「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。
8. 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
```

9. 「完了」をクリックします。
10. 次のページで、「デプロイ」をクリックします。
11. 表示される「OC4J\_BI\_Forms」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。
12. アプリケーション名 (MyReportApp) をクリックします。
13. 「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。
14. 「ライブラリ・パス」で「行の追加」をクリックし、rwrun.jar ライブラリへの次のパスを追加します。

```
ORACLE_HOME¥reports¥jlib¥rwrun.jar
```

15. zrclient.jar ライブラリへの次のパスを指定する行をさらに追加します。

```
ORACLE_HOME¥jlib¥zrclient.jar
```

16. 「適用」をクリックし、「OK」をクリックします。

### 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

1. [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 で、中間層の詳細ページを表示します。
3. 「OC4J インスタンスの作成」をクリックします。
4. OC4J インスタンスの名前を入力します。
5. 「作成」をクリックします。
6. 確認ページで「OK」をクリックします。
7. 「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 を手動で構成します。

8. 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`
  - `java.rmi.server.randomIDs`
  - `oracle.display`
  - `oracle.path`
  - `org.omg.CORBA.ORBClass`
  - `org.omg.CORBA.ORBSingletonClass`
9. `ORACLE_HOME` の `opmn.xml` ファイルで、`PATH` プロパティと `DISPLAY` プロパティを新規 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。これは `<process-type id="<your OC4J>" module-id="OC4J">` で開始されています。

```
<oc4j instanceName="<your application>" gid="<your application>">
  <config-file path="D:\oracle\ora9iasR2App\j2ee\<your
application>\config\server.xml"/>
  <oc4j-option value="-properties"/>
  <port ajp="3001-3100" rmi="3101-3200" jms="3201-3300"/>
</oc4j>
```

- b. 同じ `opmn.xml` ファイルの `OC4J_BI_FORMS` インスタンスに関する記述から、「`PATH`」変数（HP-UX では `SHLIB_PATH`、その他の UNIX では `LD_LIBRARY_PATH`）と「`DISPLAY`」変数に関する記述を含む `<environment>` タグをコピーして追加します。

```
<oc4j instanceName="<your application>" gid="<your application>"
  <config-file path="D:\oracle\ora9iasR2App\j2ee\<your
application>\config\server.xml"/>
  <oc4j-option value="-properties"/>
  <port ajp="3001-3100" rmi="3101-3200" jms="3201-3300"/>
  <environment> <!-- entry to copy from the other instan -->
    <prop name="PATH" value="values are dependant of the system"/>
    <prop name="DISPLAY" value="localhost:0"/>
  </environment>
</oc4j>
```

- c. OC4J インスタンスを再起動します。

### 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、「[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、「[Web ブラウザからのレポート要求の指定](#)」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、「[Web ブラウザからのレポート要求の指定](#)」を参照してください。

### 13.3.6 Solaris 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/guicommon9/tk90/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/Reports Runtime です。Reports Server/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 フォントがシステムに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。

**関連項目：**

[「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」](#)

OracleAS Portal を実行する前に必要なコンポーネントの登録方法に関する記述があります。

---

---

**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、report.sh に NLS\_LANG を明示的に設定することでこの問題を回避できます。新しい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もできます。詳細は、「[動的な環境切替え](#)」を参照してください。

---

---

ポータル内でレポートを公開するには、次の作業が必要です。

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。
2. レポート定義ファイル・アクセスを作成します。この手順では、レポートのプロパティ、特にレポートを収納するプロバイダを定義することで、ページ設計者がレポートをポートレットとして利用できるようにします。
3. レポートをアイテム・リンク<sup>1</sup>またはポートレット<sup>2</sup>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。この手順では、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。

---

<sup>1</sup> 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）。

<sup>2</sup> 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。

### 13.4.1 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。

---

---

### 13.4.2 レポート定義ファイル・アクセスの作成

レポートをポートレットとして提供するには、次の作業が必要です。

---

---

**注意：** 多数のレポートに対してレポート定義ファイル・アクセスを作成する必要がある場合は、それらをバッチ登録する方法が効率的です。詳細は、「[OracleAS Portal へのレポートのバッチ登録](#)」を参照してください。

---

---

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. 「管理」タブをクリックします。
3. 「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。
4. 「Reports 定義ファイル・アクセス」ポートレットで、「Reports 定義ファイル・アクセスの作成」をクリックします。
5. 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。ウィザードの最後に、「完了」をクリックします。
6. 「アクセス」タブをクリックします。
7. 「ポータルに公開」をクリックします。
8. 「適用」をクリックします。これでレポートがポートレット・リポジトリに追加されたので、レポートをページに追加することができます。

### 13.4.3 ページへのレポート・ポートレットの追加

ポートレット・リポジトリに追加されたレポートのポートレットは、他のポートレットと同様に任意のページに追加できます。

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・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。
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。

**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、  
[「レポート・コンポーネントのアイテムとしてのページへの追加」](#)を参照してください。

6. 既存のページを編集する場合は、次の手順に進みます。
7. 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せると、そのツールに関するヒントが表示されます。

8. 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。
9. 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。
10. 「OK」をクリックします。
11. 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。
12. 「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リングします。
13. 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、Oracle Application Server 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。

14. 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、「Reports 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。

- a. レポートの「コンポーネントの管理」ページの最下部にある「カスタマイズ」をクリックします。
- b. 表示するパラメータごとに「ユーザーに表示」をクリックします。

---

---

**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す。

---

---

「コンポーネントの管理」ページの値は、常にポートレットの「カスタマイズ」ページのデフォルト値として使用されます。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。
- 
-

## 13.4.4 レポート・コンポーネントのアイテム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。

---

---

**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。
6. 既存のページを編集する場合は、次の手順に進みます。
7. 「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。
8. 「非表示のアイテム・タイプ」リストで「Oracle Reports」を選択して、「>」リンクをクリックし、「表示されるアイテム・タイプ」リストに移動します。または、「>>」リンクをクリックし、「非表示のアイテム・タイプ」リスト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。
9. 「OK」をクリックします。
10. 「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」をクリックします。Oracle Reports の追加ページが表示されます。
11. 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。
12. 使用可能なデフォルトの Oracle Reports コンポーネントから選択します。
13. レポートが表示される前にユーザー入力を求める場合は、「パラメータ・フォームの表示」を選択します。
14. 「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。

15. 「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。
16. 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。

## 13.5 Web ブラウザからのレポート要求の指定

レポート要求に必要な 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 要求。次に例を示します。

```
http://your_webserver.domain_
name:port/alias/rwservlet?report=myreport.rdf&userid=username/password@my_
db&server=server_name&desformat=html&destype=cache
```

他のコマンドライン・オプションが必要な場合は、`rwclient` の有効なコマンドライン・オプションのリストが掲載されている「[コマンドライン・オプション](#)」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 要求。次に例を示します。

```
http://your_webserver.domain_name:port/alias/rwservlet?key1
```

## 13.6 URL エンジンへの要求送信

Reports Server の URL エンジンを実態にアクティブにした場合は、次のコマンドライン・オプションを使用してジョブ要求を 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 (`urlEngine` など) です。

---

**注意：** URL エンジンのアクティブ化の詳細は、「[OracleAS Reports Services の構成](#)」を参照してください。

---

たとえば、`urlParameter` に外部 URL を指定する要求は、次のようになります。

```
http://your_
webserver:portnum/reports/rwservlet?server=ReportsServer+jobType=urlEngine+urlParame
ter="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要求は、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%3Fser
ver%3DreportsServer"
```

---

---

**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要求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 バ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 サービスの使用方法」](#)

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に関する記述があります。

## 13.8 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 または OracleAS Portal から、あるいは SCHEDULE コマンドライン・オプションによって、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、レポート要求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 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、「[コマンドライン・オプション](#)」の「[SCHEDULE](#)」を参照してください。

## 13.9 追加のパラメータ

Reports Server に要求を送信すると、次に示す追加のパラメータが要求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 レポート要求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	要求の送信元であるリモート IP アドレス。
REMOTE_HOST	要求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	要求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。

## 13.10 キャッシュからのレポート出力の再利用

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、現行要求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。

- 新規のジョブ要求 "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要求 "A" で `TOLERANCE=n` ( $n$  は分単位) を指定しても、そのジョブが実際には発生しないのは、Reports Server により、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が試行されている状態です。Reports Server によってこのようなジョブが検出された場合は、新規のジョブ要求 "A" ではジョブ "B" の出力が返されます。

---

---

**注意：** TOLERANCE コマンドライン・オプションの詳細は、「[コマンドライン・オプション](#)」を参照してください。

---

---

- クラスタ環境では、クラスタの全メンバーを対象に重複ジョブ（キャッシュ・キーが同じであるジョブ）のチェックが実行されます。同じクラスタ内の別のサーバーで重複ジョブが検出された場合は、ジョブ要求がそのサーバーに転送され、キャッシュされている結果が取得されます。

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。

### 13.10.1 使用上の注意

- キャッシュのサイズは、Reports Queue Manager 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「[OracleAS Reports Services の構成](#)」を参照してください。

## 13.11 キー・マップ・ファイルの使用方法

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 要求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ようにできます。このキー値はサーバーまたはサ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要求 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、レポート要求 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:** cmdkey キーワードの詳細は、「[コマンドライン・オプション](#)」を参照してください。

---

### 13.11.1 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、「[Reports 関連の環境変数](#)」を参照してください。

### 13.11.2 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.cluster_name (if present)> 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、「[コマンドライン・オプション](#)」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、「[Web ブラウザからのレポート要求の指定](#)」を参照してください。

### 13.11.3 JSP 以外のすべてでキーを使用する場合

レポート要求 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/rwsgi.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要求に相当する要求が生成されます。

```
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。

### 13.11.4 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。



---

## 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](#)
- [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、ジョブ要求をサーバー・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。

- `rwervlet`

`rwervlet` は、HTTP と Reports Server 間で、ジョブ要求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、ジョブ要求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、`rwervlet`、`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://host: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 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 の最後のエンタ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す。

```
<soap:address location="http://localhost:8888/reports/rwwebservice" />
```

図 14-1 WSDL の表示

## RWWebService endpoint

WSDL for Service: RWWebService, generated by O

リンクをクリックして、  
WSDLを表示します。

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://hostname:portnumber/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://hostname:portnumber/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>
...
- <service name="RWWebService">
- <port name="RWWebServicePort" binding="tns:RWWebServiceBinding">
  <soap:address location="http://localhost:8888/reports/rwwebservice" />
</port>
</service>
</definitions>
```

## 14.2.3 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](#)

### 14.2.3.1 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">9.0.4.0.9</return>
</ns1:getAPIVersionResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```



### 14.2.3.2 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働していない場合、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="repssrv" version="9.0.4.0.7">
  <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="responseTime" value="2124"/>
  </performance>
</serverInfo>
```

### 14.2.3.3 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働していない場合、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>repserv</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>
```

### 14.2.3.4 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働していない場合、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>
```

### 14.2.3.5 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@v815
```

---

---

**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>
```

## 14.3 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。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://otn.oracle.com/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 version 10g (9.0.4) , 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();
    }
}
}
```



## 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。このファイルには、レポートのどのセクションがどの出力フォーマットによってどの宛先に送信されるかを指定できます。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。

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](#)
- [カスタム / プラグガブル・デスティネーションの定義](#)
- [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うえで、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、配布に関連する各種 XML ファイルについて説明します。

### 15.2.1 distribution.dtd ファイル

配布 XML ファイルを作成する際は、次のディレクトリ（Windows と UNIX で共通のパス）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す。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 `&amp;column_name` または `&&lt;column_name&>` を入力します。

### 使用上の注意

アンパサンド (&) と小なり記号 (<) は、XML で特別な意味を持つ一方で、特定の Oracle Reports Developer 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 (`&amp;`) および小なり記号と大なり記号 (`&&lt;` と `&&gt;`) を指定します。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 (`&&gt;`) 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を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](#)
- [include](#)
- [file](#)
- [printer](#)
- [destype](#)
- [attach](#)
- [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。[配布 XML ファイルの例](#)では、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。

`destinations` 要素には、次のサブ要素があります。

- [foreach](#)
- [mail](#)

- [file](#)
- [printer](#)
- [destype](#)

これらの要素について、次の各項で説明します。

## 15.3.2 foreach

### 例

```
<foreach>
<mail id="a1" to="my_addressee@mycompany.com" subject="Fourth Quarter Results">
<attach format="pdf" name="dept_&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-0069: 内部エラー

REP-57054: インプロセス・ジョブ終了済: エラーで終了しました。

REP-594: レポート出力は生成されていません。

foreach 要素には、次のサブ要素があります。

- [mail](#)
- [file](#)
- [printer](#)
- [destype](#)
- [attach](#)

これらの要素について、次の各項で説明します。

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す。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;.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;>.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 要素には、次の 3 つのサブ要素があります。

- [body](#)
- [attach](#)
- [foreach](#)

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、 <a href="#">第 15.2.2 項</a> 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。
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。



表 15-1 mail 要素の属性 (続き)

属性	有効値	説明
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	<b>デフォルト:</b> 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	<b>デフォルト:</b> normal 電子メールの配信優先度。
returnReceipt	true false	<b>デフォルト:</b> 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、「[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 タグの開始と終了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="&lt;first_name&gt;. &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	<b>デフォルト:</b> 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	<b>デフォルト:</b> 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	<b>デフォルト:</b> 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 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。attach 属性は、表 15-3 で説明します。

**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	<b>デフォルト:</b> 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	<b>デフォルト:</b> 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	<b>デフォルト:</b> all  <b>foreach</b> 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。ただし、include には 1 つだけ属性があります (src)。表 15-4 で、この属性について説明します。

**表 15-4 mail の 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に呼び出されるため、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>先行する body 要素または attach 要素の srcType 属性が file である場合は、ディレクトリ・パスとファイル名を指定するか、または単にファイル名を指定します (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)。</p> <p>先行する body 要素または attach 要素の srcType 属性が report である場合は、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 (headerSection、mainSection、trailerSection のいずれか、またはその組合せ)。</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;&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、サブ要素 **include** があります。file 要素の開始と終了の間には、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_&lt;&lt;department_id&gt;.pdf" format="pdf"
instance="this">
<include="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。

file 要素には、関連する属性もあります。これらの属性は、file タグ内に記述されます。たとえば、file の "id" 属性と "name" 属性は、次のように表されます。

**Windows:** <file id="a7" name="d:¥reports¥2001¥q4sales.pdf">

**UNIX:** <file id="a7" name="/reports/2001/q4sales.pdf">



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。 <b>Windows:</b> name="d:¥reports¥file.pdf" <b>UNIX:</b> name="reports/file.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、 <a href="#">第 15.2.2 項</a> を参照してください。
format	pdf html htmlcss rtf ascii xml bitmap	<b>デフォルト:</b> 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	<b>デフォルト:</b> all <b>foreach</b> 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、サブ要素 `include` があります。printer 要素の開始と終了の間には、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="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、<a href="#">第 15.2.2 項</a>を参照してください。</p>
copies	文字列	<p><b>デフォルト:</b> 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><b>デフォルト:</b> all</p> <p><b>foreach</b> 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、<b>instance</b> 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 (<b>this</b>)、またはすべてを同じ印刷レポートに収めるか (<b>all</b>) 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>
```

### 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 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。

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。

---

**注意：** カスタムの宛先タイプは、OracleAS Reports Services Destinations API で作成します。Oracle Reports API の最新情報は、Oracle Technology Network (<http://otn.oracle.com/> 英語サイト) または Oracle Technology Network Japan (<http://otn.oracle.co.jp/> 日本語サイト) を参照してください。

---

`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、`foreach/printer/instance="this"` を組み合わせて、それぞれに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="report"/>
<property name="number" value="&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 (`department_id` 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、`destype` の `"id"` 属性、`"name"` 属性および `"instance"` 属性は、次のように表されます。

```
<foreach>
<destype id="a1" name="name_of_destination_type" instance="all">
</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
<code>id</code>	文字列	必須です。特定の <code>file</code> 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。
<code>name</code>	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。  <code>name="fax"</code>  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。  <code>name="oraclePortal"</code>
<code>instance</code>	<code>this all</code>	<b>デフォルト:</b> <code>all</code>  <code>foreach</code> 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、 <code>instance</code> で、グループをグループ・インスタンスごとに別々の <code>destype</code> インスタンスに分割するか ( <code>this</code> )、すべてを同じ <code>destype</code> インスタンスに収めるか ( <code>all</code> ) を指定します。  たとえば、カスタムの宛先タイプが FAX である場合、 <code>instance="this"</code> 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、 <code>instance="all"</code> 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。

OracleAS Reports Services の環境では、カスタムの宛先タイプ（プラグガブル・デスティネーション）の作成と使用がサポートされています。その一環として、配布 XML ファイル内でカスタムの宛先を呼び出すことができます。配布 XML ファイルでは、`destype` 要素とともに使用されるプロパティの名前 / 値のペアを使用して、カスタムの宛先を定義することができます。

## 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。**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

### 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_&department_id&.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ことが必要です。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>@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\_&&lt;department\_id&&gt;.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="headerSection"/>
<include="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;&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;&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。

## 15.5 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かで使用する場合も基本的に同じです。次のコマンドを使用します。

```
distribute=yes destination=filename.xml
```

*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。

```
distribute=yes destination=c:¥ORACLE_HOME¥reports¥distribution¥filename.xml
```

UNIX の場合は次のように指定します。

```
distribute=yes destination=ORACLE_HOME/reports/distribution/filename.xml
```

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宛先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 で配布パラメータと宛先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「レポート要求の実行」](#)を参照してください。



## 15.6 カスタム / プラッグابل・デスティネーションの定義

配布時に、Oracle Reports で使用可能なカスタムのプラッグابل・デスティネーションを定義できます。次の例では、宛先のタイプ別に定義方法を示します。

- FAX 宛先
- oraclePortal 宛先

### FAX 宛先

汎用のタグ構造に従って、次のように配布 XML ファイルに宛先を指定できます。

#### 例 15-1 汎用的な FAX 宛先

```
<destype id="faxdest" name="fax">
<property name="number" value="123456789"/>
<include src="report"/>
</destype>
```

### oraclePortal 宛先

汎用のタグ構造に従って、配布 XML ファイルに宛先を指定できます。例 15-2 を参照してください。

例 15-2 は、DESTYPE=ORACLEPORTAL の構造の例です。DESTYPE=ORACLEPORTAL を使用してレポート出力を Portal に配信するときは、レポート出力は PAGEGROUP で指定される OracleAS Portal 内のフォルダに作成されます。

#### 関連項目：「コマンドライン・オプション」

この例にあるプロパティの詳細に関する記述があります。

#### 例 15-2 簡単な Portal 宛先

```
<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>
```

プラッグابل・デスティネーションの汎用の destype タグ構造に従ってカスタムの宛先を指定できます。これとは別に、使いやすさを考慮して、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。

### 例 15-3 カスタムの FAX 宛先

```
<fax id="faxdest" number="123456789">
<include src="report"/>
</fax>
```

distribution.dtd では、このようなタグを認識できないため、distribution.dtd に指定されている汎用の destype タグ構造に、これらのタグを対応付ける必要があります。

Reports Runtime では、ここに示した <fax> タグ構造を処理できません。これは、<fax> タグが distribution.dtd ファイルに指定される標準の宛先ではないためです。したがって、カスタム・タグ構造を、例 15-1 に示した汎用フォーマットに変換する必要があります。

distribution.xsl ファイルは、配布 XML ファイル内のユーザー定義カスタム・タグを、Reports Runtime で必要なフォーマットに変換するために使用します。Reports で認識されるのは、すべてのプラグブル・デスティネーションに使用される汎用の destype タグ構造だけです。

distribution.xsl ファイルは、XML スタイル・シートであり、Windows と UNIX どちらの場合も、ORACLE\_HOME¥reports¥conf¥distribution.xsl にあります。このファイルを変更するには、destype タグ・フォーマットを、distribution.dtd ファイルで定義されている必要なフォーマットに変換するためのテンプレートを追加します。

そのためには、FAX 宛先用のテンプレートを distribution.xsl ファイルに指定する必要があります。このテンプレートを使用して、<fax> タグ構造が汎用の destype 構造に変換されます。distribution.xsl のエントリは次のようになります。

```
<xsl:output doctype-system="distribution.dtd"/>

<xsl:template match = "/">
<xsl:apply-templates match = "destinations" />
</xsl:template>

<xsl:template match="destinations">
<destinations>

<!--
The Standard mail/file/printer/destype and foreach must be copied to the transformed
xml. The foreach tag must be copied only if it is specified with
file/mail/printer/destype tags.
-->
<xsl:copy-of select="mail"/>
<xsl:copy-of select="file"/>
<xsl:copy-of select="printer"/>
<xsl:copy-of select="destype"/>
<xsl:copy-of select="foreach"/>

<!-- apply template for the sample FAX destination -->
<xsl:apply-templates match = "fax" />
```

```
</destinations>
</xsl:template>

<!--
Sample Transformation Template for a FAX destination specified in the
distribution.xml file

<fax id="FAXDEST" number="123456789">
<include src="report"/>
</fax>
-->

<xsl:template match="fax">
<!-- create a new destype element -->
<xsl:element name="destype">
<!--
create an ID attribute and copy the value from the ID given for the fax destination
-->
<xsl:attribute name="id">
<xsl:value-of select="@id"/>
</xsl:attribute>

<!-- create a Name attribute with a fax as it's value -->
<xsl:attribute name="name">fax</xsl:attribute>
<!--
create a Property Attribute with name / value attribute pairs property tag is
created for number attribute. Similarly create more property tags for any other
attribute you add to the FAX destination
-->
<xsl:element name="property">
<xsl:attribute name="name">number</xsl:attribute>
<xsl:attribute name="value">
<xsl:value-of select="@number"/>
</xsl:attribute>
</xsl:element>

<!-- copy the include tag as it is -->
<xsl:copy-of select="include"/>
</xsl:element>
<!-- end of template -->
</xsl:template>

</xsl:stylesheet>
```

---

---

**注意：** XSL ファイルを変更した後は、そのファイルを同じ位置に同じファイル名で保存します。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。

---

---

## 15.7 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。

### 15.7.1 destype=oraclePortal

Oracle9i Reports リリース 2 (9.0.2) 以降では、Reports Server 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：** 「[コマンドライン・オプション](#)」 および Reports Builder のオンライン・ヘルプ (DESTYPE コマンドライン・キーワードの詳細)

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。

---

---

- DESTYPE=ORACLEPORTAL は、rwruntime 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。

### 15.7.2 XML とデリミタ付き出力

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、DESFORMAT=XML および DESFORMAT=DELIMITED 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 「XML 宛先ファイルの配布リストでは参照パラメータを使用できません。」

ただし、参照パラメータを指定しない配布であれば、これらのフォーマットを使用できます。

この制限は、今後のリリースで解決する予定です。

### 15.7.3 配布と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。

この制限は、今後のリリースで解決する予定です。



## 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 コマンドを追加し、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、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・モデルのサポートによつ

---

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。

- Reports Builder 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」を参照します。
- 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ぎられます。

選択した Oracle Reports XML タグの詳細は、Reports Builder 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。

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：** [「コマンドライン・オプション」](#)

コマンドライン・キーワードの使用方法の詳細に関する記述があります。

## 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。

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**

演算子	使用方法
eq	等しい
lt	より小さい
lteq	以下
neq	等しくない
gt	より大きい
gteq	以上
btw	範囲内
notBtw	範囲外
like	類似
notLike	非類似
null	NULL
notNull	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" DTDVersion="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が設定されません。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" DTDVersion="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" DTDVersion="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_EMPLOYEES">
  <dataItem="EMPLOYEE_ID"/>
  <dataItem="FIRST_NAME"/>
  <dataItem="LAST_NAME"/>
  <dataItem="JOB_ID"/>
  <dataItem="MANAGER_ID"/>
  <dataItem="HIRE_DATE"/>
  <dataItem="SALARY"/>
  <dataItem="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"/>
  <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" DTDVersion="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、「[実行時の XML レポート定義の適用](#)」を参照してください。
- 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、「[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、「[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、rwrun、rwbuilder、rwconverter および URL レポート要求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的詳細は、「[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、ジョブ要求を 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 コマンドが 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 RWRUN を使用する場合のコマンドラインは、次のとおりです。

```
RWRUN 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要求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 TRACEOPT=trace_app
```

---

---

**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 に適用するジョブ要求を、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 コマンド・エントリに、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、BeforeForm または AfterForm トリガーで SRW.APPLY\_DEFINITION および SRW.ADD\_DEFINITION ビルトインを使用します。この後の項では、これらのビルトインの例を示します。

**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 BeforeForm または AfterForm 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 AfterParameterForm トリガーで使用されます。

videosales\_custom.rdf ファイルの AfterParameterForm 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='$$NNNN.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="LNNNNNNNNNNNOD00"/>');
SRW.ADD_DEFINITION('<field name="F_SumTOTAL_SALESPerCITY" source="SumTOTAL_
SALESPerCITY"
formatMask="LNNNNNNNNNNNOD00"/>');
SRW.ADD_DEFINITION('<field name="F_SumTOTAL_COSTPerCITY" source="SumTOTAL_
COSTPerCITY"
formatMask="LNNNNNNNNNNNOD00"/>');
SRW.ADD_DEFINITION('</section>');
SRW.ADD_DEFINITION('</layout>');
ELSIF :MONEY_FORMAT='$NNNN' THEN SRW.ADD_DEFINITION('<layout>');
SRW.ADD_DEFINITION('<section name="main">');
SRW.ADD_DEFINITION('<field name="F_TOTAL_PROFIT" source="TOTAL_PROFIT"
formatMask="LNNNNNNNNNNNO"/>');
SRW.ADD_DEFINITION('<field name="F_TOTAL_SALES" source="TOTAL_SALES"
formatMask="LNNNNNNNNNNNO"/>');
SRW.ADD_DEFINITION('<field name="F_TOTAL_COST" source="TOTAL_COST"
formatMask="LNNNNNNNNNNNO"/>');
SRW.ADD_DEFINITION('<field name="F_SumTOTAL_PROFITPerCITY" source="SumTOTAL_
PROFITPerCITY" formatMask="LNNNNNNNNNNNO"/>');
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要求を送信します。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要求を OracleAS Reports Services に送信します。

```

rwclient userid=username/password@my_db
report=c:¥corp¥myreports¥emp.xml
destype=file desname=emp.pdf desformat=pdf
server=server_name

```

Reports Runtime のコマンドである rwrun を使用する場合のコマンドラインは、次のとおりです。

```

rwrun userid=username/password@my_db
report=c:¥corp¥myreports¥emp.xml
destype=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 オプションを使用して、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
stype=rdf
source="(inven1.rdf, inven2.rdf, manu.rdf)"
dtype=rdf
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 コマンド・エントリに、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](#)

次の各項では、これらの機能を説明します。

### 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://otn.oracle.com/> 英語サイト) または Oracle Technology Network Japan (<http://otn.oracle.co.jp/> 日本語サイト) 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  
TRACEOPT=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 コマンド・エントリおよびトレース・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どう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 コマンド・エントリに、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)。

---

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手法

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起動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、Oracle では OracleAS Reports Services にスケジューリング・メカニズムを導入し、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できるようにしました。ただし、これでも 1 つだけ解決されない要求事項が残りました。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 について説明するとともに、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。Oracle 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](#)



これらのプロシーダを使用して、パラメータ・リストを操作できます。この項では、これらのプロシーダを簡単に説明します。詳細は、Oracle Reports API のドキュメントを参照してください。

---

---

**注意：** Oracle Reports API の最新情報は、Oracle Technology Network (<http://otn.oracle.com/> 英語サイト) または Oracle Technology Network Japan (<http://otn.oracle.co.jp/> 日本語サイト) を参照してください。

---

---

### 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。このプロシーダを呼び出し、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 ジョブの送信方法

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` で要求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、要求の処理に使用する `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。

各要求からは、`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。この情報はパッケージ変数であり、`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・リストはオブジェクト型であり、`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)、要求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。

## 17.1.4 ステータスの確認方法

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、Reports Servlet の SHOWJOBS コマンドを使用したときにジョブ・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。

## 17.1.5 サーバーのステータス・レコードの使用方法

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系統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。これらの例外の詳細は、Oracle Reports API のリファレンス・ドキュメントを参照してください。

---

**注意：** Oracle Reports API の最新情報は、Oracle Technology Network (<http://otn.oracle.com/> 英語サイト) または Oracle Technology Network Japan (<http://otn.oracle.co.jp/> 日本語サイト) を参照してください。

---

## 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、レポート要求が実行されます。

cgicmd.dat ファイルに指定されているキーを使用して要求を実行する場合は、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 ファイルを使用した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。

アドバンスド・キューイング付属の `dbms_AQadm` パッケージには、アドバンスド・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の、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だけのままとなります。

---

**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。

Oracle Reports API の最新情報は、Oracle Technology Network (<http://otn.oracle.com/> 英語サイト) または Oracle Technology Network Japan (<http://otn.oracle.co.jp/> 日本語サイト) を参照してください。

---

### 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ず。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・プロセス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・プロセス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

---

## 各国語サポートおよび双方向サポート

第 III 部では、Reports 関連の各国語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「NLS と双方向サポートの実装」](#)



## NLS と双方向サポートの実装

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・セットに対する各国語サポート (NLS) と、テキストの読取り順序に対する双方向サポートです。

Oracle NLS 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。NLS 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、NLS アーキテクチャを、Reports に関連する NLS 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、Oracle でサポートされている双方向、Unicode および翻訳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。

- [NLS アーキテクチャ](#)
- [NLS 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](#)

## 18.1 NLS アーキテクチャ

Oracle NLS 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、NLS 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 NLS の環境変数

NLS 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：** OracleAS Reports Services ホスト・マシン 1 つに対して、言語を 1 つしか指定できません。使用する言語がそれぞれに異なる利用対象者にアプリケーション・サーバー・サービスを提供する場合は、言語ごとにホスト・マシンを用意する必要があります。

---

---

表 18-1 では、特に OracleAS Reports Services に関する NLS 関連の環境変数を説明します。

---

---

**注意：** NLS のすべての環境変数の詳細は、OTN-J (Oracle Technology Network Japan) (<http://otn.oracle.co.jp>) の『Oracle Application Server 10g グローバリゼーション・ガイド』を参照してください。

---

---

表 18-1 OracleAS Reports Services に特に関連する環境変数

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
USER_NLS_LANG	Reports Runtime コンポーネントで使用される言語

## 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 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、report.sh に NLS\_LANG を明示的に設定することでこの問題を回避できます。新しい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もできます。詳細は、「[動的な環境切替え](#)」を参照してください。

---

---

NLS\_LANG の language、territory、character に一般に使用される値の詳細は、『Oracle Application Server 10g グローバリゼーション・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：** help、showjobs、showenv などの Reports Servlet コマンド・ページおよびエラー・メッセージの言語は、NLS\_LANG ではなく、マシンのロケール (Solaris 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せん。

---

---



### 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」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

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。

**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アプリケーション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せん。

**18.2.1.2.2 Windows プラットフォームでのフォントのエイリアシング** 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
```

uifont.ali ファイル内の各行の構文は、次のとおりです。

```
Face.Size.Style.Weight.Width.CharSet=Face.Size.Style.Weight.Width.CharSet
```

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。

言語環境変数の詳細は、「[NLS の環境変数](#)」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。

- サーバー・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 機能の 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）
- 肯定応答と否定応答の文字列

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  
FRM-10043: Cannot open file.

French:  
ORA-0092: table ou vue inexistante  
FRM-10043: Ouverture de fichier impossible

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 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せん。

- Reports Builder を英語で使用する一方で、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。

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。

### 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 で HTML を生成している場合は、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場合があります（ここでは日本語のキャラクタ・セットを指定）。

```
<META http-equiv="Content-Type" content="text/html; charset=shift_jis">
```

さらに、レポートを XML に出力する予定がある場合は、次の構文に従って、レポートの XML Prolog Value プロパティにキャラクタ・セットを含めることができます。

```
<?xml version="1.0" encoding="shift_jis" ?>
```

どちらの場合も、キャラクタ・セットを表す値によって、ホスト環境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、IANA charset または encoding パラメータで一般に使用される値を示します。

---

---

**注意：** charset と encoding の値には、大文字と小文字の区別がありません。小文字と大文字のどちらでも入力できます。

---

---

---

---

**注意：** HTML の生成に使用する予定のキャラクタ・セットを .rdf ファイルに設定するには、Before Report Escape プロパティに次の記述が必要です。

```
charset="text/html; charset=&encoding"
```

&encoding は、実行時に適切な設定で置き換えられます。

---

---

表 18-2 IANA charset の有効値

言語	IANA キャラクタ・セット名
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

表 18-2 IANA charset の有効値 (続き)

言語	IANA キャラクタ・セット名
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
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

表 18-2 IANA charset の有効値 (続き)

言語	IANA キャラクタ・セット名
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、通常の文字表記方向が右から左である中東圏および北アフリカ圏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。
- 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 の環境変数 `DEVELOPER_NLS_LANG` および `USER_NLS_LANG` を `NLS_LANG` のかわりに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。環境変数の詳細は、「[NLS の環境変数](#)」を参照してください。

---

**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、`report.sh` に `NLS_LANG` を明示的に設定することでこの問題を回避できます。新しい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もできます。詳細は、「[動的な環境切替え](#)」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。
- インターネットを介した全世界的なデータ交換を可能にする。

### 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。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、画面上に日本語とドイツ語を同時に表示するには、NLS\_LANG 変数を次のいずれかに設定する必要があります。

```
NLS_LANG=JAPANESE_JAPAN.UTF8
```

```
NLS_LANG=GERMAN_GERMANY.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 を含むデータベースのテキスト、多言語テキスト、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。OracleAS Reports Services は、現在 Windows 上で Unicode をサポートしています。

---

---

**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

OracleAS Reports Services では、様々な言語のフォントと入力方式について、オペレーティング・システムに依存しています。特定の言語でテキストを入力および表示するには、その言語をサポートするオペレーティング・システムの実行している必要があります。フォントのサポートは限られていますが、オペレーティング・システムのフォントのみに制限されているわけではありません。

Windows NT 4.0 および以降のバージョンは、TrueType Big Font を提供しています。これらのフォントには、多言語テキストの表示や印刷に必要なすべての文字が含まれています。多言語テキストを入力、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 社が Windows NT 4.0 以降で提供している Big Font は、次のとおりです。

- Arial
- Courier New
- Lucida Console
- Lucida Sans Unicode
- Times New Roman

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、「[NLS の環境変数](#)」を参照してください。

## 18.5.4 ALTER SESSION の使用方法

USERID キーワードを通じて Oracle データベースに接続されているか、または Reports Builder の「接続」ダイアログから接続した場合は、SQL コマンド ALTER SESSION を使用して、NLS のデフォルト値の一部を上書きすることができます。たとえば、様々な地域にデータを公開するレポートを、Oracle データベースで作成するとします。この場合、Oracle データベースが通貨フィールドを書式設定する際に使用する通貨記号、3桁の区切りおよび小数点記号を、ユーザー・パラメータに応じて変える必要があります。これにはいくつかの方法がありますが、その1つに、Before Report トリガーで Oracle データベース・セッションの NLS\_LANG 変数を変更する方法があります。

---

---

**注意：** ALTER SESSION は、プラグابل・データソース問合せ (JDBC や XML など) には適用されません。

---

---

---

---

**注意：** ALTER SESSION コマンドの詳細は、OTN-J (Oracle Technology Network Japan) (<http://otn.oracle.co.jp>) で入手可能な『Oracle9i SQL リファレンス』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 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 Technology Network (<http://otn.oracle.com/> 英語サイト) 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・ルーチン `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));
```

アプリケーションを別の言語に変更するには、`nls_appl_mes` ファンクションを含む `PL/SQL` ライブラリを、翻訳済みテキストを指定した `nls_appl_mesg` ファンクションを含む同名のライブラリに置き換えるだけです。



# 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 コンソールは、Reports Server に対して管理機能と監視機能の両方を提供します。
- Oracle Enterprise Manager 10g Grid Control コンソール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。

この章では、OracleAS Reports Services に関連する Oracle Enterprise Manager 10g の管理機能と監視機能について説明するとともに、Oracle Enterprise Manager 10g を使用して Reports Server を構成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 ページへのナビゲート](#)
- [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](#)
- [サーバー・パフォーマンスの監視](#)
- [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](#)

---

**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。Oracle Enterprise Manager 10g で「ヘルプ」リンクをクリックすると、ヘルプが表示されます。

---

**関連項目：** Oracle Enterprise Manager 10g の詳細は、『Oracle Application Server 10g 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 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、「[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。
- Oracle Process Manager and Notification Server の `opmn.xml` ファイル。`opmn.xml` ファイルとその使用方法の詳細は、「[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。

---

**関連項目：** 詳細は、次の項を参照してください。

- 「[Oracle Process Manager and Notification Server および Oracle Enterprise Manager 10g による Reports Server の構成](#)」

## 19.2 Reports Server ページへのナビゲート

この項では、Reports Server ページにナビゲートする方法について説明します。

- [Grid Control コンソールでの Reports Server ページへのナビゲート](#)
- [Application Server Control コンソールでの Reports Server ページへのナビゲート](#)

---

**注意：** Oracle Enterprise Manager 10g を起動する前に、管理サーバーをインストールおよび構成する必要があります。その方法の詳細は、『Oracle Enterprise Manager 構成ガイド』を参照してください。このマニュアルは、Oracle Technology Network Japan (<http://otn.oracle.co.jp/>) から入手できます。

---



## 19.2.1 Grid Control コンソールでの Reports Server ページへのナビゲート

Reports Server ページへのナビゲートは、様々な方法で行うことができます。次の手順は、可能な方法の 1 つです。

1. Grid Control コンソールを起動します。Grid Control 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。Grid Control コンソールは、図 19-1 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 19-1 Grid Control コンソール

Page Refreshed May 20, 2003 11:30:46 P

**Status**  
View: All Targets [Go]

Total Monitored Targets: 20

**Availability**

Down (6)	30%
Unknown (1)	5%
Up (13)	65%

**Alerts**

Critical	8
Warning	16
Errors	4

**Target Search**  
Search: All Targets [Go]

**Enterprise Configuration**  
View: Databases & Application Servers [Go]

Targets Not Included 2 of 3

Databases & Application Servers	Targets	Installations	Interim Patches
Oracle9i 9.0.1.4.0	0	1	Not available
Oracle9i Application Server Instance 9.0.4.0.0	1	2	Not available

**Resource Center**

- [Documentation](#)
- [Release Notes](#)
- [Support](#)
- [Oracle Technology Network](#)

Home | Targets | Configuration | Alerts | Management System | Setup | Preferences | Help | Logout

Copyright © 1996, 2003, Oracle. All rights reserved.  
About Oracle Enterprise Manager Version 4.0.1.0.0

2. 「ターゲット」タブをクリックします。
3. 「Application Server」サブ・タブをクリックします。これは、図 19-2 のように表示されます。

図 19-2 Grid Control コンソールの「ターゲット」の「Application Server」タブ

Application Servers

Page Refreshed May 20, 2003 11:36:46 PM

Search  (Go) Add

Select	Name	Availability	Alerts	CPU Usage (%)	Memory Usage (MB)
<input checked="" type="radio"/>	EnterpriseManager.cpeddama-ultra.us.oracle.com	↑	0 0	4.72 ✓	441.87
<input type="radio"/>	ias0312core897_reports-int.us.oracle.com	↑	0 0	29.38 ✓	272.68

Availability Legend: ↑ Up ↓ Down 🚫 Under blackout 🚫 Agent Unreachable 🚫 Agent Down ⚙️ Status Pending 🚫 Metric Collection Error

Home | Targets | Configuration | Alerts | Management System | Setup | Preferences | Help | Logout

Copyright © 1996, 2003, Oracle. All rights reserved.  
About Oracle Enterprise Manager Version 4.0.1.0.0

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 19-3 のように表示されます。

図 19-3 Grid Control コンソールの Application Server ページ

Home | J2EE Applications | Web Applications | Performance

Page Refreshed May 21, 2003 12:21:31 AM

**General**

Status **Up**  
Availability (%) **Not Applicable**  
(Last 24 Hours)  
Availability URL <http://cpeddama-ultra.us.oracle.com:7778>  
Version **9.0.4.0.0**  
Installation Type **J2EE and Web Cache**  
Oracle Home `/private1/iasw/DMS_re13`  
Host [cpeddama-ultra.us.oracle.com](http://cpeddama-ultra.us.oracle.com)

**Availability URL Response (seconds)**

**Components**

Name	Current Status
WebCache	↑
HTTP_Server	↑
home	↑
OC4J_EM	↑
Reports Server temp	↓

**Host Alerts**

Metric	Severity	Alert Triggered	Last Value	Last Checked
Alert Severity for Tue May 20 02:08:31 200334	⚠️	May 19, 2003 7:08:28 PM		2 May 19, 2003 7:08:28 PM
Memory Utilization for Top Processes (%) for 11140	⚠️	May 19, 2003 11:38:24 AM	29.4	May 21, 2003 12:18:26 AM
Filesystem Space Available (%) for /	⚠️	May 7, 2003 4:18:58 PM	12.13	May 20, 2003 4:52:20 PM

**Related Links**

<a href="#">Patch</a>	<a href="#">Change Availability URL</a>	<a href="#">Administer</a>
<a href="#">All Metrics</a>	<a href="#">Edit Metric Thresholds</a>	<a href="#">Alert History</a>
<a href="#">Blackouts</a>		

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 ページが表示されます。これは、図 19-7 のように表示されます。

## 19.2.2 Application Server Control コンソールでの Reports Server ページへのナビゲート

Reports Server ページへのナビゲートは、様々な方法で行うことができます。次の手順は、可能な方法の 1 つです。

1. Application Server Control コンソールを起動します。Application Server Control 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 コンソールは、[図 19-4](#) に示すように「ファーム」ページに起動されます。ここには、使用可能な Oracle Application Server インスタンスが表示されます。

**図 19-4 Oracle Enterprise Manager 10g の「ファーム」ページ**

The screenshot displays the Oracle Enterprise Manager 10g 'Farm' page. It features a 'Clusters' section with a table listing a cluster named 'cluster1'. Below this is a 'Standalone Instances' section with a table listing two instances. The page includes various control buttons and a footer with copyright information and navigation links.

Select Name	Status	Instances
cluster1	↓	0

Select Name	Host	Oracle Home
ias904m@binfra.abihari-ultra.us.oracle.com	abihari-ultra.us.oracle.com	/private2/AS904M8/iAS904M8binfra
ias904m@bpw.abihari-ultra.us.oracle.com	abihari-ultra.us.oracle.com	/private2/AS904M8/iAS904M8bPW

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 19-5](#) に示すような該当インスタンスのページが表示されます。

図 19-5 Oracle Enterprise Manager 10g の Oracle Application Server ページ

ホーム | J2EEアプリケーション ポート Infrastructure ページ更新時刻 2004/01/06 13:55:33

---

**一般**

すべてを停止(D)    すべてを再起動(R)

**稼働中**

ステータス: 稼働中  
 ホスト: ipsun2ee3.jp.oracle.com  
 インストールタイプ: Business Intelligence and Forms  
 Oracleホーム: /disk01/oracle/product/ora9iashi  
 ファーム: ssdb.jp.oracle.com

**CPU使用率**

Application Server (3%)  
 アイドル(68%)

**メモリー使用量**

Application Server (43% 872MB)  
 空き(3% 69Mバイト)

---

システム・コンポーネント

すべて選択 | 選択解除

選択	名前	ステータス	開始時間	CPU使用率(%)	メモリー使用量(Mバイト)
<input type="checkbox"/>	Discoverer	↑	2003/12/29 13:34:50	0.00	19.50
<input type="checkbox"/>	Forms	↑	2003/12/29 13:45:39	0.00	0.00
<input type="checkbox"/>	home	↑	2004/01/05 11:51:52	0.10	53.59
<input type="checkbox"/>	HTTP_Server	↑	2003/12/29 13:45:06	0.23	74.37
<input type="checkbox"/>	OC4J_BI_Forms	↑	2003/12/29 13:45:39	0.12	84.77
<input type="checkbox"/>	OC4J_Portal	↑	2003/12/29 13:46:31	0.26	122.55
<input type="checkbox"/>	OC4J_Wireless	↑	2003/12/29 13:49:22	0.98	139.71
<input type="checkbox"/>	Portal.portal	↑	N/A	N/A	N/A
<input type="checkbox"/>	Reports Server.rep_ipsun2ee3	↑	N/A	N/A	N/A
<input type="checkbox"/>	Web_Cache	↑	2003/12/29 13:52:33	0.03	31.33

- 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 19-6 に示すような Reports Server のメイン・ページが表示されます。

## 図 19-6 Oracle Enterprise Manager 10g の Reports Server メイン・ページ


Page Retrieved Monday, June 2, 2003 4:58:32 AM PDT

---

Home [Engines](#)

---

**General**



Current Status **Up**  
 Start Time **Jun 2, 2003 4:51:32 AM**  
 Version **9.0.4.0.10**

[Stop](#) [Restart](#)

**Configuration**

Reports Server Cluster **Not Clustered**  
 Trace Option **trace\_all**  
 Trace Mode **trace\_replace**  
 Maximum Queue Size **1000**

---

**Response and Load**

CPU Usage (%) **18.56**  
 Memory Usage (MB) **58.56**  
 Average Response Time (ms)

**Jobs**

Current Jobs **0**  
 Failed Jobs **7**  
 Finished Jobs **0**  
 Scheduled Jobs **0**  
 Number of Jobs Transferred **0**

---

**Dependent Components**

Component Name	Status	Start Time
HTTP_Server	↑	Monday, June 2, 2003 4:50:21 AM PDT
Web_Cache	↓	
Portal:portal	↓	
Single_Sign-On:orasso	↓	
OC4J_BI_Forms	↑	Monday, June 2, 2003 4:50:38 AM PDT

---

**Administration**

[Configuration](#) [Edit Configuration File](#) [Server Trace](#) [Server Log](#)

---

Home [Engines](#)

---

[Logs](#) | [Preferences](#) | [Help](#)

Copyright © 1996, 2003, Oracle. All rights reserved.  
 About Oracle Enterprise Manager Version 9.0.4

## 19.3 Reports Server の起動、停止および再起動

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. 起動、停止または再起動する Reports Server について、Application Server Control コンソールの Reports Server メイン・ページまでナビゲートします。

---

**注意：** Reports Server を起動、停止および再起動できるのは、Application Server Control コンソールからのみです。

---

**関連項目：** 詳細は、「[Application Server Control コンソールでの Reports Server ページへのナビゲート](#)」を参照してください。

2. Reports Server のメイン・ページでは、次の操作を実行できます。
  - サーバーを起動する場合は、「起動」ボタンをクリックします。
  - サーバーを停止する場合は、「停止」ボタンをクリックします。
  - サーバーを再起動する場合は、「再起動」ボタンをクリックします。

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、「停止」（無効化） ボタンと「起動」 ボタンが表示されます。
- サーバーが稼働している場合は、「停止」 ボタンと「再起動」 ボタンが表示されません。

---

---

**注意：** Reports Server のホームページにある各タブの詳細は、「[Oracle Enterprise Manager 10g Application Server Control コンソール](#)」を参照してください。

---

---

## 19.4 ジョブ・キューの表示と管理

ジョブ・キューは、Application Server Control コンソールを使用して表示および管理できます。

- [Application Server Control コンソールでのジョブ・キューの表示と管理](#)

---

---

**注意：** ジョブ・キューの管理（たとえば、ジョブの取消しや再実行）は、Application Server Control コンソールから行います。Application Server Control コンソールにアクセスするには、Reports Server メイン・ページに戻り、「関連リンク」の下にある「管理」をクリックします。そこで、「[Application Server Control コンソールでのジョブ・キューの表示と管理](#)」の手順に従います。

---

---

### 19.4.1 Application Server Control コンソールでのジョブ・キューの表示と管理

この後の項では、次の操作について説明します。

- [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](#)

#### 19.4.1.1 現行ジョブ・キューの表示と管理

Reports の現行ジョブ・キューには、特定の Reports Server で現在実行されているすべてのジョブが示されます。

Oracle Enterprise Manager 10g では、次の操作が可能です。

- [Reports Server](#) の現行ジョブ・キューの表示
- 現行ジョブの取消し

**19.4.1.1.1 Reports Server の現行ジョブ・キューの表示** 現行ジョブ・キューを表示する手順は、次のとおりです。

1. 情報を表示する Reports Server について、Oracle Enterprise Manager 10g の Reports Server メイン・ページにナビゲートします。

**関連項目：** 詳細は、「[Reports Server ページへのナビゲート](#)」を参照してください。

2. Reports Server のメイン・ページで、「ジョブ」セクションまでスクロールし、「現行ジョブ」の横の数字をクリックします。

現行ジョブがない場合は、「現行ジョブ」の横にゼロが表示され、キューが空であるためリンクはありません。

表 19-1 では、現行ジョブ・キューに表示される情報を説明します。

**表 19-1 現行ジョブ・キューに表示される情報**

項目	説明
選択	特定のジョブを選択するために使用するラジオ・ボタン。現行ジョブ・キューのページでは、ジョブを取り消す際にこの機能が最も役立ちます。取り消すジョブの横にある「選択」ラジオ・ボタンをクリックし、ページ上部の「取消」ボタンをクリックします。
ID	Reports Server がこのジョブに割り当てた一意のジョブ識別子。この番号はサーバーで管理されるため、ユーザーはリセットできません。
ジョブ名	このジョブを実行する際に使用したコマンドラインにジョブ名を指定した場合はその名前が表示され、ジョブ名を指定しなかった場合は、ジョブ要求の "report=" または "module=" パラメータに指定されたジョブの名前が表示されます。
所有者	このジョブを実行しているユーザーの ID。
出力タイプ	実行時にこのジョブに指定された宛先タイプ (destype)。
出力フォーマット	実行時にこのジョブに指定された出力フォーマット (desformat)。
受付時間	この要求がジョブ・キューに登録された日付と時刻。
開始時間	このジョブが実行された日付と時刻。
ステータス	現行ジョブのステータス。

**19.4.1.1.2 現行ジョブの取消し** 現行ジョブを取り消す手順は、次のとおりです。

1. 「現行ジョブ」 ページで、取り消すジョブの横にある「選択」 ラジオ・ボタンをクリックします。
2. 「取消」 ボタンをクリックします。現在実行されているジョブがない場合は、Oracle Enterprise Manager 10g にこのボタンが表示されません。

ジョブを再実行する場合は、「終了したジョブ」 ページからそれを実行できます。

### 19.4.1.2 スケジュールされたジョブ・キューの表示と管理

スケジュールされたジョブ・キューには、特定の Reports Server で実行がスケジュールされているすべてのジョブが表示されます。

Oracle Enterprise Manager 10g では、次の操作が可能です。

- [Reports Server のスケジュールされたジョブ・キューの表示](#)
- [スケジュールされたジョブの取消し](#)

**19.4.1.2.1 Reports Server のスケジュールされたジョブ・キューの表示** スケジュールされたジョブ・キューを表示する手順は、次のとおりです。

1. ジョブ・キューを表示する Reports Server について、Oracle Enterprise Manager 10g の Reports Server メイン・ページにナビゲートします。

**関連項目：** 詳細は、「[Reports Server ページへのナビゲート](#)」を参照してください。

2. Reports Server のメイン・ページで、「ジョブ」 セクションまでスクロールし、「スケジュールされたジョブ」の横の数字をクリックします。

スケジュールされたジョブがない場合は、「スケジュールされたジョブ」の横にゼロが表示され、キューが空であるためリンクはありません。

[表 19-2](#) では、スケジュールされたジョブ・キューに表示される情報を説明します。

**表 19-2 スケジュールされたジョブ・キューに表示される情報**

項目	説明
選択	特定のジョブを選択するために使用するラジオ・ボタン。スケジュールされたジョブ・キューのページでは、ジョブを取り消す際にこの機能が最も役立ちます。取り消すジョブの横にある「選択」ラジオ・ボタンをクリックし、ページ上部の「取消」ボタンをクリックします。
ID	Reports Server がこのジョブに割り当てた一意のジョブ識別子。この番号はサーバーで管理されるため、ユーザーはリセットできません。



**表 19-2 スケジュールされたジョブ・キューに表示される情報 (続き)**

項目	説明
ジョブ名	このジョブを実行する際に使用したコマンドラインにジョブ名を指定した場合はその名前が表示され、ジョブ名を指定しなかった場合は、ジョブ要求の "report=" または "module=" パラメータに指定されたジョブの名前が表示されます。
所有者	このジョブの実行をスケジュールしたユーザーの ID。
出力タイプ	このジョブに指定された宛先タイプ (destype)。
出力フォーマット	このジョブに指定された出力フォーマット (desformat)。
繰り返し間隔	ジョブが実行される頻度 (毎日、毎月など)。
前回実行時間	スケジュールされたジョブが前に実行されたときのタイムスタンプ。
次回実行時間	スケジュールされたジョブがその次に実行されたときのタイムスタンプ。

**19.4.1.2.2 スケジュールされたジョブの取消し** スケジュールされたジョブを取り消す手順は、次のとおりです。

1. 「スケジュールされたジョブ」 ページで、取り消すジョブの横にある「選択」ラジオ・ボタンをクリックします。
2. 「取消」 ボタンをクリックします。  
スケジュールされたジョブがない場合は、Oracle Enterprise Manager 10g にこのボタンが表示されません。

ジョブを再実行する場合は、「終了したジョブ」 ページからそれを実行できます。

### 19.4.1.3 終了したジョブ・キューの表示と管理

終了したジョブ・キューには、特定の Reports Server で正常に終了したすべてのジョブが表示されます。

Oracle Enterprise Manager 10g では、次の操作が可能です。

- [Reports Server の終了したジョブ・キューの表示](#)
- [ジョブのトレース・ファイルの表示](#)
- [キャッシュからの結果の表示](#)
- [終了したジョブの再実行](#)

**19.4.1.3.1 Reports Server の終了したジョブ・キューの表示** 終了したジョブ・キューを表示する手順は、次のとおりです。

1. 情報を表示する Reports Server について、Oracle Enterprise Manager 10g の Reports Server メイン・ページにナビゲートします。

**関連項目：** 詳細は、「[Reports Server ページへのナビゲート](#)」を参照してください。

2. Reports Server のメイン・ページで、「ジョブ」セクションまでスクロールし、「終了したジョブ」の横の数字をクリックします。

終了したジョブがない場合は、「終了したジョブ」の横にゼロが表示され、キューが空であるためリンクはありません。

表 19-3 では、終了したジョブ・キューに表示される情報を説明します。

**表 19-3 終了したジョブ・キューに表示される情報**

項目	説明
選択	<p>特定のジョブを選択するために使用するラジオ・ボタン。終了したジョブ・キューのページでは、ジョブを選択して次の操作を行う際にこの機能が最も役立ちます。</p> <p>ジョブ出力の表示</p> <p>表示するジョブの横にある「選択」ラジオ・ボタンをクリックし、ページ上部の「結果の表示」ボタンをクリックします。</p> <p>ジョブのトレース結果の表示（ランタイム・コマンドにトレース・コマンドを指定した場合）</p> <p>トレース結果を表示するジョブの横にある「選択」ラジオ・ボタンをクリックし、ページ上部の「トレースの表示」ボタンをクリックします。トレース・ファイルが見つからない場合、Reports Server によって「ログ・ファイルがありません」エラー・メッセージが表示されます。ジョブの <code>trace</code> オプションを指定する方法の詳細は、「<a href="#">trace</a>」を参照してください。</p> <p>ジョブの再実行</p> <p>再実行するジョブの横にある「選択」ラジオ・ボタンをクリックし、ページ上部の「レポートの再実行」ボタンをクリックします。</p>
ID	<p>このリンクを使用して、ジョブのトレース・ファイルを表示します。Reports Server がこのジョブに割り当てた一意のジョブ識別子。この番号はサーバーで管理されるため、ユーザーはリセットできません。</p>

表 19-3 終了したジョブ・キューに表示される情報 (続き)

項目	説明
ジョブ名	このリンクを使用して、ジョブの結果を表示します。このジョブを実行する際に使用したコマンドラインにジョブ名を指定した場合はその名前が表示され、ジョブ名を指定しなかった場合は、ジョブ要求の "report=" または "module=" パラメータに指定されたジョブの名前が表示されます。
所有者	このジョブを実行したユーザーの ID。
出力タイプ	このジョブに指定された宛先タイプ (destype)。
出力フォーマット	このジョブに指定された出力フォーマット (desformat)。
受付時間	この要求がジョブ・キューに登録された日付と時刻。
開始時間	このジョブの実行が開始された日付と時刻。
終了時間	このジョブが完了した日付と時刻。
ステータス	ジョブの終了状態。終了したジョブ・キューでは、ステータスが常に「正常に終了しました」と表示されます。

**19.4.1.3.2 ジョブのトレース・ファイルの表示** トレース・ファイルには、レポートの実行に関するトレース情報が記録されます。Reports Server で使用するトレース・オプションは、ジョブのコマンドラインで `traceopts` 引数を指定して設定します。

**関連項目：** 使用可能なトレース・オプションの詳細は、「[trace](#)」を参照してください。

ジョブのトレース・ファイルを表示するには：

1. 「終了したジョブ」ページにある「選択」列で、トレース・ファイルを表示するジョブの横にあるラジオ・ボタンをクリックします。
2. ページ上部の「トレースの表示」ボタンをクリックします。または、ID リンクをクリックします。

---

**注意：** トレース・ファイルが見つからない場合、Reports Server によって「ログ・ファイルがありません」エラー・メッセージが表示されます。ジョブのトレース・オプションを指定する方法の詳細は、「[trace](#)」を参照してください。

---

**19.4.1.3.3 キャッシュからの結果の表示** Reports Server の監視時に、適切な出力が生成されていることを確認するために、レポートの出力をすばやくチェックできれば便利な場合があります。Reports Server のキャッシュからジョブ結果を表示する手順は、次のとおりです。

1. 「終了したジョブ」 ページにある「選択」列で、表示するジョブの横にあるラジオ・ボタンをクリックします。
2. ページ上部の「結果の表示」 ボタンをクリックします。または、名前のリンクをクリックします。

新しいブラウザ・ウィンドウに結果が表示されます。

**19.4.1.3.4 終了したジョブの再実行** レポートをすばやく再送信して実行することが必要な場合があります（たとえば、キャッシュの内容をリフレッシュする場合など）。「終了したジョブ」 ページからジョブを再実行する手順は、次のとおりです。

1. 「選択」列で、再実行するジョブの横にあるラジオ・ボタンをクリックします。
2. ページ上部の「レポートの再実行」 ボタンをクリックします。

### 19.4.1.4 失敗したジョブ・キューの表示と管理

失敗したジョブ・キューには、特定の Reports Server で取り消された、またはエラーで終了したすべてのジョブが示されます。

Oracle Enterprise Manager 10g では、次の操作が可能です。

- [Reports Server の失敗したジョブ・キューの表示](#)
- [失敗したジョブのトレース・ファイルの表示](#)
- [失敗したジョブの再実行](#)

**19.4.1.4.1 Reports Server の失敗したジョブ・キューの表示** 失敗したジョブ・キューを表示する手順は、次のとおりです。

1. 情報を表示する Reports Server について、Oracle Enterprise Manager 10g の Reports Server メイン・ページにナビゲートします。

**関連項目：** 詳細は、「[Reports Server ページへのナビゲート](#)」を参照してください。

2. Reports Server のメイン・ページで、「ジョブ」セクションまでスクロールし、「失敗したジョブ」の横の数字をクリックします。

失敗したジョブがない場合は、「失敗したジョブ」の横にゼロが表示され、キューが空であるためリンクはありません。

表 19-4 では、失敗したジョブ・キューに表示される情報を説明します。

表 19-4 失敗したジョブ・キューに表示される情報

項目	説明
選択	<p>特定のジョブを選択するために使用するラジオ・ボタン。失敗したジョブ・キューのページでは、ジョブを選択して次の操作を行う際にこの機能が最も役立ちます。</p> <p>ジョブのトレース結果の表示（ランタイム・コマンドにトレース・コマンドを指定した場合）</p> <p>トレース結果を表示するジョブの横にある「選択」ラジオ・ボタンをクリックし、ページ上部の「トレースの表示」ボタンをクリックします。</p> <p>ジョブの再実行</p> <p>再実行するジョブの横にある「選択」ラジオ・ボタンをクリックし、ページ上部の「レポートの再実行」ボタンをクリックします。</p>
ID	<p>Reports Server がこのジョブに割り当てた一意のジョブ識別子。この番号はサーバーで管理されるため、ユーザーはリセットできません。</p>
ジョブ名	<p>このジョブを実行する際に使用したコマンドラインにジョブ名を指定した場合はその名前が表示され、ジョブ名を指定しなかった場合は、ジョブ要求の "report=" または "module=" パラメータに指定されたジョブの名前が表示されます。</p>
所有者	<p>このジョブを実行したユーザーの ID。</p>
出力タイプ	<p>このジョブに指定された宛先タイプ (destype)。</p>
出力フォーマット	<p>このジョブに指定された出力フォーマット (desformat)。</p>
受付時間	<p>この要求がジョブ・キューに登録された日付と時刻。</p>
開始時間	<p>このジョブの実行が開始された日付と時刻。</p>
終了時間	<p>このジョブが取り消されたか、またはエラーで終了した日付と時刻。</p>
ステータス	<p>ジョブの終了状態。このステータスで、ジョブがユーザーによって取り消されたか、またはエラーで終了したかを確認できます。ジョブがエラーで終了した場合は、その原因を示す簡単なエラー・メッセージが表示されます。</p>

**19.4.1.4.2 失敗したジョブのトレース・ファイルの表示** ジョブが失敗した原因の究明に役立てるために、失敗したジョブのトレース・ファイルを表示できます。失敗したジョブのトレース・ファイルを表示する手順は、次のとおりです。

1. 「失敗したジョブ」ページにある「選択」列で、トレース・ファイルを表示するジョブの横にあるラジオ・ボタンをクリックします。

2. ページ上部の「トレースの表示」ボタンをクリックします。または、IDリンクをクリックします。

---

---

**注意：** トレース・ファイルが見つからない場合、Reports Server によって「ログ・ファイルがありません」エラー・メッセージが表示されます。ジョブの trace オプションを指定する方法の詳細は、「[trace](#)」を参照してください。

---

---

**19.4.1.4.3 失敗したジョブの再実行** 失敗したジョブの問題が解決されたかどうかを簡単に確認するには、「失敗したジョブ」ページからそのジョブを再実行します。失敗したジョブを再実行する手順は、次のとおりです。

1. 「失敗したジョブ」ページにある「選択」列で、再実行するジョブの横にあるラジオ・ボタンをクリックします。
2. ページ上部の「レポートの再実行」ボタンをクリックします。

## 19.5 サーバー・パフォーマンスの監視

Oracle Enterprise Manager 10g に登録されている各 Reports Server には、サーバーのステータスとパフォーマンスに関する一般情報を要約するメイン・ページが個別に用意されます。Reports Server のパフォーマンスは、次のコンポーネントから監視できます。

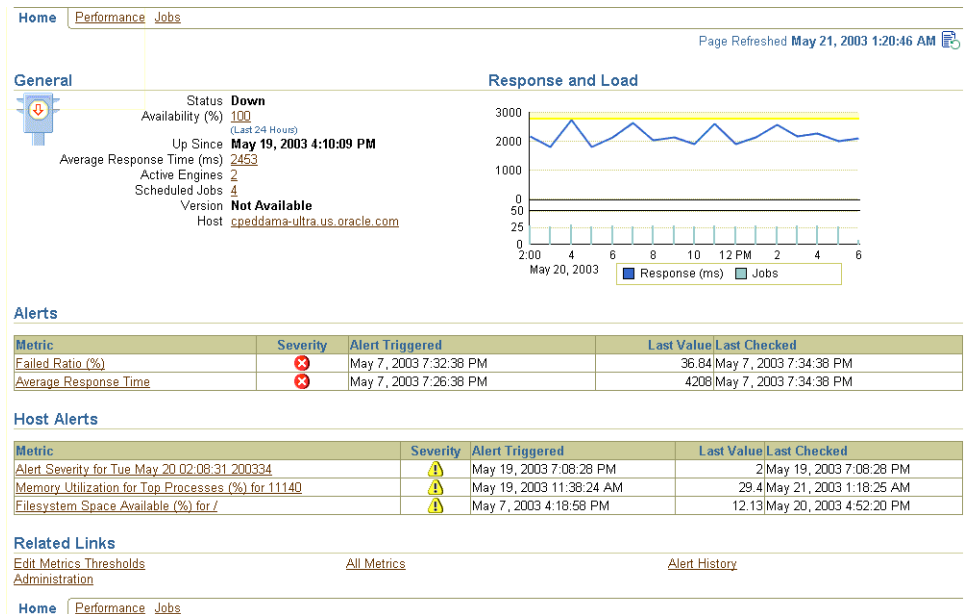
- [Oracle Enterprise Manager 10g Grid Control コンソール](#)
- [Oracle Enterprise Manager 10g Application Server Control コンソール](#)

### 19.5.1 Oracle Enterprise Manager 10g Grid Control コンソール

Grid Control コンソールの Reports Server メイン・ページには、Reports Server の監視を支援する各種の統計およびグラフが表示されます。Grid Control コンソールの Reports Server メイン・ページには、次の3つのタブがあります。

- ホーム ([図 19-7](#) を参照)
- パフォーマンス ([図 19-8](#) を参照)
- ジョブ ([図 19-9](#) を参照)

図 19-7 Reports Server メイン・ページの「ホーム」タブ

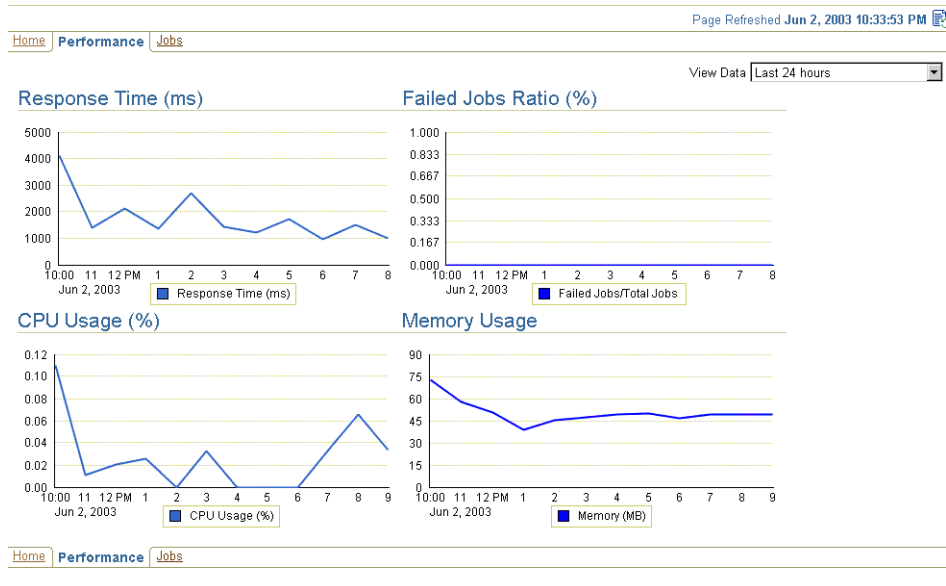


「ホーム」タブには、次のセクションがあります。

- 「一般」には、Reports Server の現行ステータスが表示されます。稼動中である場合は、サーバーが起動された日付と時刻も表示されます。また、実行中のエンジンの数、スケジュールされたジョブ、使用可能な割合、平均レスポンス時間、製品バージョン、この特定の Reports Server を所有するホストなどもリストされます。使用可能な割合、平均レスポンス時間または実行中のエンジンの数をクリックすると、さらに詳細へドリルダウンできます。すべてのドリルダウン・ページで、表示された統計の時間間隔を変更できます（たとえば、最近 24 時間、最近 7 日間、最近 1 か月の数値を表示することができます）。ホスト名をクリックすると、そのホストのメイン・ページが表示されます。
- 「レスポンスおよびロード」には、平均レスポンス時間と処理中のジョブの数との関係を示すグラフが表示されます。そのグラフをクリックすると、レスポンス時間に関する詳細ページにドリルダウンできます。このドリルダウン・ページには、データの追加統計と履歴グラフが含まれます。
- 「警告」には、メトリックがしきい値を超えたために受け取った最近のメッセージが表示されます。たとえば、平均レスポンス時間が一定以上の長さになると警告が発生する場合があります。メトリックの名前をクリックすると、そのメトリックの詳細ページが表示されます。

- 「ホスト・アラート」には、**Reports Server** を所有するホストに関して受け取った最近のメッセージが表示されます。たとえば、システム使用率が特定のしきい値を超えるとホスト・アラートが発生する場合があります。
- 「関連リンク」からは、他の領域にジャンプできます。
  - 「メトリックしきい値の編集」をクリックすると、各種メトリックの警告対象となるしきい値を変更できます。
  - 「すべてのメトリック」をクリックすると、使用可能なすべてのメトリックの展開および縮小可能なリストが表示されます。リスト内のメトリックの詳細にドリルインすることもできます。
  - 「アラート履歴」をクリックすると、メトリックごとの警告の発生数を集計した棒グラフが表示されます。
  - 「管理」をクリックすると、**Application Server Control** の **Reports Server** ページが表示されます。このページでは、**Reports Server** の起動や停止などのアクションを実行できます。

図 19-8 Reports Server メイン・ページの「パフォーマンス」タブ

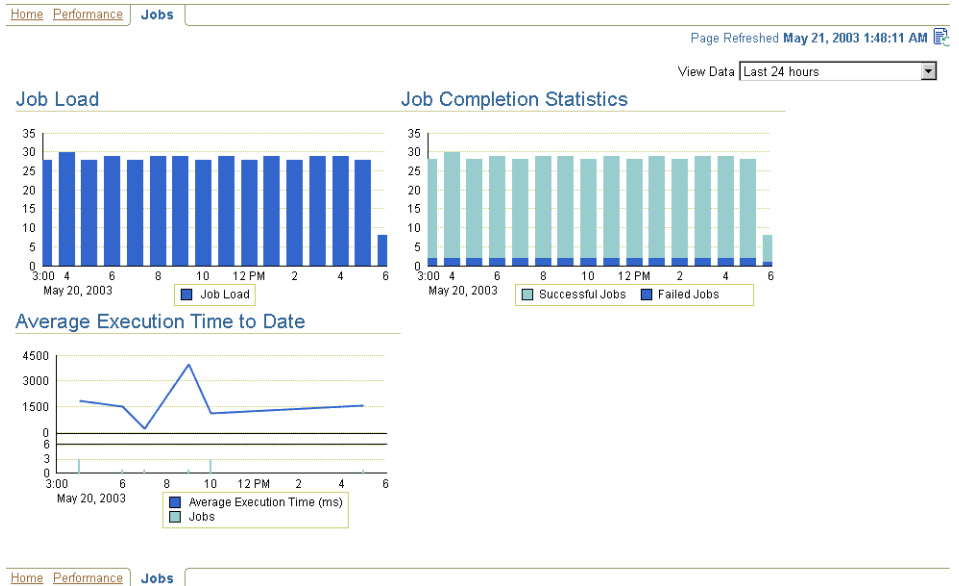


「パフォーマンス」タブには、次のセクションがあります。



- 「データの表示」値リストを使用すると、「パフォーマンス」タブにある各グラフの時間間隔を変更できます。
- 「レスポンス時間」では、特定の時間間隔（たとえば、1時間）にわたるレスポンス時間がグラフ化されます。
- 「失敗したジョブの割合」では、特定の時間間隔（たとえば、1時間）にわたる、全ジョブに対する失敗したジョブの割合がグラフ化されます。
- 「CPU 使用率 (%)」では、特定の時間間隔（たとえば、1時間）にわたる CPU 使用率がグラフ化されます。
- 「メモリー使用率」では、特定の時間間隔（たとえば、1時間）にわたるメモリー使用率がグラフ化されます。

図 19-9 Reports Server メイン・ページの「ジョブ」タブ



「ジョブ」タブには、次のセクションがあります。

- 「データの表示」値リストを使用すると、「ジョブ」タブにある各グラフの時間間隔を変更できます。
- 「ジョブのロード」では、特定の時間間隔（たとえば、1時間）の間に、Reports Server に送信されたジョブの数がグラフ化されます。

- 「ジョブ完了の統計」では、特定の時間間隔（たとえば、1時間）にわたる失敗ジョブと成功ジョブの数がグラフ化されます。
- 「経過時間」では、グラフ起動時に実行されたジョブの所要時間がグラフ化されます。現行ジョブについては、平均実行時間がグラフ化されます。このグラフで表示される統計によって次のことがわかります。
  - 迅速に実行されないジョブ
  - レスポンスを停止するジョブ

## 19.5.2 Oracle Enterprise Manager 10g Application Server Control コンソール

Application Server Control コンソールの Reports Server メイン・ページには、次の2つのタブがあります。

- ホーム (図 19-10 を参照)
- エンジン (図 19-11 を参照)

図 19-10 Reports Server メイン・ページの「ホーム」タブ

ホーム エンジン ページ取得時刻 2004/01/06 14:16:57

---

**一般**

現行ステータス **稼働中**  
開始時間 **2004/01/06 14:15:21**  
バージョン **9.0.4.0.21**

**停止** **再起動**

**レスポンスおよびロード**

CPU使用率(%) N/A  
メモリ使用量(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 **trace\_all**  
トレース・モード **trace\_replace**  
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	↑	2003/12/29 13:45:06
Web Cache	↑	2003/12/29 13:52:33
Portal.portal	↑	N/A
OC4J_BI_Forms	↑	2003/12/29 13:45:39

---

**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](#)

---

ホーム エンジン

---

[ログ](#) | [設定\(環境\)](#) | [ヘルプ](#)

Copyright (C) 1996, 2003, Oracle. All rights reserved.  
Oracle Enterprise Manager 10g Application Server Controlのバージョン情報

「ホーム」タブには、次のセクションがあります。

- 「一般」には、Reports Server の現行ステータスが表示されます。稼動中である場合は、サーバーが起動された日付と時刻も表示されます。このセクションには、停止、起動または再起動するためのボタンも表示されます。

---

**注意：** Reports Server の現行ステータスは、Oracle Enterprise Manager 10g または OPMN を使用して Reports Server を起動および停止する場合のみ、正確に反映されます。

---

- 「構成」には、選択した Reports Server のクラスタ名（そのサーバーがクラスタのメンバーである場合）、その構成ファイルに指定されているトレース・オプションとモード、および Reports Server のキューに一度に保持できる最大ジョブ数（現行ジョブ、スケジュールされたジョブおよび過去のジョブの3つのキューのすべてに対して適用される）が表示されます。これらの値は、すべて Reports Server 構成ファイルで設定可能です。

**関連項目：** 構成ファイルの詳細は、次の項を参照してください。

- 「Reports Server の構成要素 (rwserverconf.dtd)」
- 「レスポンスおよびロード」には、CPU 使用率、メモリー使用率および Reports Server の平均レスポンス時間について、現行の統計値が表示されます。
  - 「CPU 使用率 (%)」には、選択した Reports Server で現在使用されているホスト・マシンの CPU の割合が表示されます。
  - 「メモリー使用量 (MB)」には、選択した Reports Server で現在使用されているホスト・マシンの RAM の割合が表示されます。このメトリックは、インプロセス・サーバーには適用されません。

---

**注意：** インプロセス・サーバーは OC4J\_BI\_Forms インスタンス内で実行され、「CPU 使用率 (%)」および「メモリー使用量 (MB)」メトリックは OC4J\_BI\_Forms インスタンスの一部になります。したがって、インプロセス・サーバーのメトリックは、常に「N/A」と表示されます。

---

- 「平均レスポンス時間 (ミリ秒)」には、Reports Server がクライアントからの要求を処理するために要した平均時間がミリ秒単位で表示されます。
- 「ジョブ」には、現行（現在実行中の）ジョブ、終了したジョブ、スケジュールされたジョブ、失敗したジョブのそれぞれの数が表示されます。また、同一のクラスタ（クラスタ環境内）の現行サーバーから別のサーバーに転送されたジョブの数が表示されます。ゼロよりも大きな数値が表示されている場合は、それをクリックして、特定のキューに含まれる各ジョブの完全な一覧（失敗したすべてのジョブなど）にドリルダウンできます。値がゼロの場合は、そのキューは空であるため、ドリルダウンは必要ありません。

- 「このクラスタで実行されている他のサーバー」：この項の詳細は、「[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](#)」を参照してください。
- 「依存コンポーネント」リストには、Reports Server が依存しているすべてのコンポーネント（Oracle HTTP Server や Oracle Application Server Web Cache など）のステータスが一覧表示されます。このリストのコンポーネント名をクリックすることで、各依存コンポーネントのより詳細なステータス情報にドリルインできます。
- 「管理」には、各種管理機能へのリンクがあります。
  - 「構成」をクリックすると、最も一般的に使用される Reports Server またはエンジンのパラメータを変更するページが表示されます。このページにある「構成ファイルの編集」ボタンを使用して、Reports Server 構成ファイルを直接編集できます。
  - 「構成ファイルの編集」をクリックすると、Reports Server 構成ファイルを直接編集するページが表示されます。このページには便利な「構文チェック」機能があり、変更内容を検証した後で、それらの変更を構成ファイルにコミットできます。
  - 「サーバー・トレース」をクリックすると、Reports Server の最新のレポート実行トレース情報を参照したり、分析用にダウンロードすることができるページが表示されます。
  - 「サーバー・ログ」をクリックすると、Reports Server の最新のログ・ファイルを参照したり、分析用にダウンロードすることができるページが表示されます。

### 関連項目： Oracle Enterprise Manager 10g オンライン・ヘルプ

ログ・ファイルおよびトレース・ファイルの検索に使用可能なオプションに関する詳細があります。

### 図 19-11 Reports Server メイン・ページの「エンジン」タブ

エンジン

ページ取得時刻 2004/01/06 14:39:41

[ホーム](#) [エンジン](#)

各エンジンはReportsジョブの処理に使用できる実行プロセスです。このプロセスは、長時間アイドルになると終了します。

エンジン合計 2

エンジンID	エンジン	実行中
rwEng	1	0
rwURLEng	1	0

[エンジンのパラメータ](#)

[ホーム](#) [エンジン](#)

[ログ](#) | [設定\(環境\)](#) | [ヘルプ](#)

「エンジン」タブには、次のセクションがあります。

- 「エンジン合計」には、Reports Server で現在実行されているエンジンの数が示されません。このフィールドの下の表には、ID 別にエンジンが一覧表示され、現行ステータスも表示されます。
- 「エンジン ID」には、選択した Reports Server で、ジョブの処理に使用可能なエンジンのタイプが表示されます。
- 「エンジン」には、指定のエンジン・タイプについて、選択した Reports Server で現在アクティブになっているエンジンの合計数が表示されます。
- 「実行中」には、該当するエンジン・タイプについて、選択した Reports Server で現在実行されているエンジンの合計数が表示されます。
- 「エンジンのパラメータ」ボタンをクリックすると、「構成」ページが直接表示されます。このページでは、エンジンのパラメータと Report Server のパラメータを参照および変更できます。このページにあるリンクを使用して、Report Server 構成ファイルを直接編集できます。

## 19.6 Reports Server 構成ファイルの表示と変更

Reports Server の構成は、次のいずれかの方法で変更および表示できます。

- [選択したパラメータの構成](#)
- [server\\_name.conf ファイルの構成](#)

### 19.6.1 選択したパラメータの構成

Oracle Enterprise Manager 10g で最も一般的に使用されるエンジンのパラメータとサーバーのパラメータを表示および変更する手順は、次のとおりです。

1. 情報を表示する Reports Server について、Application Server Control コンソールの Reports Server メイン・ページにナビゲートします。
2. 「管理」の下の「構成」リンクをクリックします。「構成」画面に表示されるパラメータは、Reports Server の構成ファイル (`server_name.conf`) から取得されます。
  - a. **レポート・エンジンのパラメータ：**

- \* エンジン ID: 選択したサーバーで有効なエンジン ID。

ドロップダウン・メニューからエンジン ID を選択し、選択したエンジンのエンジン・パラメータを編集します。サーバーの構成ファイルにあるエンジンのみが、ドロップダウン・リストに表示されます。

---

---

**注意：** 新しいエンジンを追加する場合は、「構成ファイルの編集」ボタンをクリックします。

---

---

- \* クラスパス (オプション) : class 属性で指定される Java クラスへのディレクトリ・パス。Reports Server で使用できる各種エンジン・プロパティの詳細は、「[engine](#)」を参照してください。
- \* 初期エンジン数 (initEngine) : 初期化時に Reports Server で起動するエンジンの数。Reports Server で使用できる各種エンジン・プロパティの詳細は、「[engine](#)」を参照してください。
- \* 最大エンジン数 (maxEngine) : サーバーで実行できるこのタイプのエンジンの最大数。Reports Server で使用できる各種エンジン・プロパティの詳細は、「[engine](#)」を参照してください。
- \* 最小エンジン数 (minEngine) : サーバーで維持されるこのタイプのエンジンの最小数。Reports Server で使用できる各種エンジン・プロパティの詳細は、「[engine](#)」を参照してください。
- \* シャットダウン前の最大ジョブ数 (engLife) : エンジンをシャットダウンする前に完了する必要があるジョブの最大数。Reports Server で使用できる各種エンジン・プロパティの詳細は、「[engine](#)」を参照してください。
- \* シャットダウン前の最大アイドル時間 (maxIdle) (分) : 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)。Reports Server で使用できる各種エンジン・プロパティの詳細は、「[engine](#)」を参照してください。
- \* JVM オプション (オプション) : Java 仮想マシン (JVM) でエンジンを起動するときに、Reports Server で使用する JVM のオプション。Reports Server で使用できる各種エンジン・プロパティの詳細は、「[engine](#)」を参照してください。

### b. Reports Server のパラメータ :

- \* キャッシュ・サイズ (MB) : Reports Server のキャッシュのファイル・サイズ。単位はメガバイト (MB) です。cache プロパティの詳細は、「[cache](#)」を参照してください。
- \* キャッシュ・ディレクトリ (オプション) : Reports Server のキャッシュの場所。cache プロパティの詳細は、「[cache](#)」を参照してください。
- \* 最大キャッシュ・ファイル数 (オプション) : キャッシュで許可されるファイルの最大数。cache プロパティの詳細は、「[cache](#)」を参照してください。
- \* 無視するパラメータ (オプション) : Reports Server でキャッシュ・キーを作成するときに無視するレポート・パラメータ (Reports Server では、キャッシュ・キーを使用して、受信したジョブ要求がキャッシュ内の既存出力に一致するかどうか判断されます)。cache プロパティの詳細は、「[cache](#)」を参照してください。
- \* アイドル・タイムアウト (分) (オプション) : 接続のアイドル状態が許可される時間 (分)。各種 connection プロパティの詳細は、「[connection](#)」を参照してください。

- \* 最大接続数 (オプション) : サーバーが同時にサービスを提供できる要求の最大数。要求数が maxConnect 値を超えると、Java 例外が返されます。各種接続プロパティの詳細は、「[connection](#)」を参照してください。
  - \* 最大キュー・サイズ : Reports Server のジョブ・キューに保持できるジョブの最大数。各種キュー・プロパティの詳細は、「[queue](#)」を参照してください。
  - \* トレース・オプション : 使用可能な trace オプションの詳細は、「[trace](#)」を参照してください。
- c. 「OK」をクリックして、エンジンおよびサーバーのパラメータに加えた変更を Reports Server 構成ファイルに保存します。「確認」ページが表示されます。「はい」をクリックして、サーバーを再起動します。

---

---

**注意 :** 変更を有効にするには、サーバーを再起動する必要があります。

---

---

- d. 変更を破棄する場合は、「取消」をクリックします。

## 19.6.2 server\_name.conf ファイルの構成

Oracle Enterprise Manager 10g で Reports Server の構成ファイルを表示および変更する手順は、次のとおりです。

1. 情報を表示する Reports Server について、Application Server Control コンソールの Reports Server メイン・ページにナビゲートします。

**関連項目 :** 詳細は、「[Application Server Control コンソールでの Reports Server ページへのナビゲート](#)」を参照してください。

2. 「管理」の下の「構成ファイルの編集」リンクをクリックします。
3. 表示ウィンドウで必要な変更を加えます。
4. 「構文チェック」ボタンをクリックして、XML の構文をチェックします。

---

---

**注意 :** このボタンをクリックしても、構成要素に対して入力した値は検証されません。たとえば、ある要素がディレクトリ・パスの指定を必要とする場合でも、構文チェックでは指定したパスが正しいかどうかは検証されません。このチェックでは、XML の構文のみが検証されます。

---

---

5. 「OK」ボタンをクリックして変更を保存するか、「取消」ボタンをクリックして変更を破棄します。「取消」をクリックすると、Reports Server メイン・ページに戻ります。

6. 前の手順で「OK」をクリックした場合は、Reports Server を再起動するように求められます。Reports Server 構成ファイルの変更は、Reports Server を再起動するまで有効になりません。

変更内容は、次にサーバーを起動または再起動した後に有効になります。

**関連項目：** [第3章「OracleAS Reports Services の構成」](#)

Reports Server 構成ファイルの属性と値の詳細に関する記述があります。

## 19.7 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

複数の Reports Server をクラスタ化すると、Oracle Enterprise Manager 10g で各クラスタ・メンバーのページに表示される「このクラスタで実行されている他のサーバー」の下にクラスタ化の内容が反映されます。リスト内の各クラスタ・メンバーから、そのメンバーのページにアクセスできます。

表 19-5 では、「このクラスタで実行されている他のサーバー」セクションでクラスタ・メンバーごとに表示される情報を説明します。

**表 19-5 「このクラスタで実行されている他のサーバー」の下に表示される情報**

行	説明
サーバー名	選択した Reports Server が属しているクラスタのメンバーである他の Reports Server の名前。サーバー名をクリックすると、そのクラスタ・メンバーのホームページにジャンプします。
終了したジョブ	リスト内の Reports Server のジョブ・キューに現在入っている終了したジョブの合計数。
現行ジョブ	リスト内の Reports Server のジョブ・キューに入っている現在実行中のジョブの合計数。
スケジュールされたジョブ	リスト内の Reports Server のジョブ・キューに現在入っているスケジュールされたジョブの合計数。
失敗したジョブ	リスト内の Reports Server のジョブのうち、完了前に停止したジョブの合計数。これには、ユーザーが強制終了したジョブや、エラーで終了したジョブが含まれます。
平均レスポンス時間	リスト内の Reports Server がクライアントからの要求を処理するために要する平均時間（ミリ秒）。



## 19.8 ポート番号の表示

多くの場合、Reports Server により使用されているポート番号を簡単に調べる方法が必要になります。Application Server Control コンソールでは、次の手順によりポート番号をすばやく確認できます。

1. Application Server Control コンソールを起動します。Application Server Control 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 コンソールは、[図 19-4](#) に示すように「ファーム」ページに起動されます。ここには、使用可能な Oracle Application Server インスタンスが表示されます。
3. 情報を表示する Reports Server を含む Oracle Application Server インスタンスを見つけます。そのインスタンスをクリックして、メイン・ページを表示します。
4. Oracle Application Server インスタンスのメイン・ページで、「ポート」をクリックして「ポート」タブを表示します。「ポート」タブには、該当の Oracle Application Server インスタンスにより現在使用されているすべてのポートが一覧表示され、その中に Reports Server で使用されているポートも含まれます。

**図 19-12 Application Server Control コンソール、「ポート」タブ**

ホーム [12EEアプリケーション](#) ポート [infrastructure](#) ページ更新時刻2004/01/06 15:11:48 JST

「使用中のポート」列は、ポートが定義されていない、またはコンポーネントが稼働していない場合は空です。Enterprise Managerを使用してポートを構成できる場合は、「構成」列にアイコンが表示されます。それ以外の場合は、コンポーネントドキュメントを参照してください。ポートの変更方法にかかわらず、ポート値を変更する前にポート依存性を考慮する必要があります。詳細情報: [OracleASのポート依存性情報](#)

◁ 行前へ | 1-25 / 32 | 7行後へ ▷

コンポーネント名	タイプ	使用中のポート	推奨されるポート範囲	構成
DCM Object Cache	Cache Discovery Port		7100-7199	
home	JMS	3706	3701-3800	
home	RMI	3206	3201-3300	
home	AJP	3306	3301-3400	
Log Loader	Management	44001	44000-44099	
OC4J_BI_Forms	AJP	3303	3301-3400	
OC4J_BI_Forms	JMS	3703	3701-3800	
OC4J_BI_Forms	RMI	3203	3201-3300	
OC4J_Portal	JMS	3704	3701-3800	
OC4J_Portal	RMI	3204	3201-3300	
OC4J_Portal	AJP	3304	3301-3400	
OC4J_Wireless	JMS	3705	3701-3800	
OC4J_Wireless	RMI	3205	3201-3300	
OC4J_Wireless	AJP	3305	3301-3400	
OPMN	ONS Local	6101	6100-6199	
OPMN	ONS Request	6004	6003-6099	
OPMN	ONS Remote	6201	6200-6299	

## 19.9 中間層の変更

場合によっては、中間層のインスタンスが使用するインフラストラクチャ・サービスの変更が必要になることがあります。たとえば、インストール時に選択したものとは異なる **Oracle Internet Directory** や **Metadata Repository** を使用する場合があります。中間層のインスタンスが使用するインフラストラクチャ・サービスの変更方法の手順は、『Oracle Application Server 10g 管理者ガイド』を参照してください。

---

## Oracle Reports のチューニング

レポート要求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。Oracle Reports のチューニングを開始する前に、次の要点を慎重に検討してください。

- 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（背後で実行される）のことをいいます。体感パフォーマンスの例は、「[事前フェッチ](#)」を参照してください。

この章では、個々の Oracle Reports 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-J (Oracle Technology Network Japan) (<http://otn.oracle.co.jp>) にあるホワイトペーパー『OracleAS Reports Services のスケーラビリティ』を参照してください。

---

この章の主な項は、「[方法論](#)」です。

## 20.1 方法論

チューニングでは、レポートのみを切り離して捉えるのではなく、次のものとの関連で考える必要があります。

- 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有無など）
- ユーザー操作の必要性の程度

これらが明確になったら、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2つ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ます。

### 1. 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などを行います。

次の項を参照してください。

- [「パフォーマンス分析ツール」](#)
- [「データへのアクセス」](#)

レポートの実行時間を最小化する方法に関する記述があります。

### 2. 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。

次の項を参照してください。

- [「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」](#)

レポートのフォーマットとレイアウトを最適化して生産性を最大化する方法に関する記述があります。

### 3. 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、「一括処理と配布」を参照してください。

次の項を参照してください。

- 「Forms からの Oracle Reports のコール」
- 「レポートの実行」

適切なランタイム・パラメータを設定および削除する方法に関する説明があります。

次は、パフォーマンスの向上を実現する最も効果的な方法です。

- 変更を徐々に行います。
- パフォーマンスのボトルネックを切り離します。
- データ・アクセスを最適化します。
- フォーマットを効率化します。
- 実行時の効率を最大化します。

ある分野で行った変更は、他の分野のパフォーマンスにも影響を与えます。

## 20.1.1 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。

### 20.1.1.1 レポート・トレース

レポート・トレース・オプションを設定すると、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にも役立ちます。

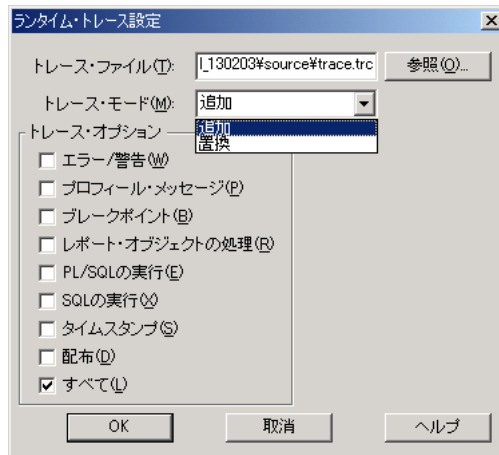
トレースは、様々なレポート実行方法に対して設定できます。たとえば、.rdf および .jsp の両方のレポート定義に、SQL または非 SQL のデータソース (XML およびテキスト・ブラッグブル・データソースなど)、あるいはその両方が含まれる場合などもトレース可能です。

## レポート・トレース・ファイルの生成

トレースをオンにするには、次のいずれかを実行します。

- Reports Builder では：
  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。トレース・オプションの詳細は、第3章「OracleAS Reports Services の構成」を参照してください。

図 20-1 Reports Builder



- report\_server\_name.conf 構成ファイルで、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、Oracle Reports ログ・ディレクトリ (ORACLE\_HOME¥reports¥log) 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、デフォルト名は *hostname-rwbuilder.trc* です。

---

- Reports Builder ビルトイン・パッケージ (SRW) で、SRW.TRACE\_START、SRW.TRACE\_END、SRW.TRACE\_ADD\_OPTION、SRW.TRACE\_REM\_OPTION を使用して、トレース・オプションを指定します。
- rwbuilder および rwrwn 実行可能ファイルの場合は、コマンドラインで次のようにトレース・オプションを指定します。

```
traceFile=trace_file_name traceOpts=trace_all traceMode=trace_replace
```

---

**注意：** トレース・ファイルの位置は、現在作業中のディレクトリ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。コマンドラインのトレース・オプションは、report\_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せん。

---

- rwservlet 実行可能ファイルの場合は、servername.conf ファイルでトレース・オプションを設定します。サーバー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。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・ログ・ディレクトリ (ORACLE\_HOME¥reports¥log) に対する相対パスです。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 serverName.trc、デフォルトのエンジン・トレース・ファイル名は serverName-engineName-engineNo.trc です。

---

- rwservlet の場合は、rwservlet.properties 構成ファイルでトレース・オプションを設定します。

各オプションについて、それぞれ 1 行を使用します。

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

**関連項目：**

- 第3章「OracleAS Reports Services の構成」

OracleAS Reports Services トレース・オプションの詳細に関する記述があります。

- 付録 A 「コマンドライン・オプション」

レポートのトレース・オプションの詳細に関する記述があります。

コマンドラインで各オプションを組み合わせで使用できます。たとえば、TRACEOPTS=(TRACE\_APP,TRACE\_PRF)とした場合、すべてのレポート・オブジェクトとパフォーマンス統計に関する情報がログ・ファイルに記録されます。

生成されたトレース・ファイルを参照することで、データのフェッチにかかった実時間とレポートのフォーマットにかかった実時間を確認できます。この情報は、パフォーマンスのボトルネックの特定に役立ちます。

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 module=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 の Profiler の統計**

```
+-----+
| 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。



表 20-1 Reports Builder (続き)

フィールド	説明
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ابل・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。

### 20.1.1.2 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 Query Builder を使用してください。ただし、データ・ウィザードでは、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。たとえば、使用可能な索引があっても、それらを使用しない SQL などがあります。

レポートの SQL をチューニングするには、Oracle データベースの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どうかわかります。

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 Form 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、『SQL 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・オプ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・オプ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・オプ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。

### 20.1.1.3 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。

ORA\_PROF ビルトイン・パッケージの詳細は、Reports Builder 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・ストアド・プロシージャとして実装するようにします。ストアド・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、可能なかぎ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。しかしこの操作は、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。

## 例

```
SRW.DO_SQL ('CREATE TABLE' || :tname ||
` (ACCOUNT NUMBER
NOT NULL PRIMARY KEY, COMP NUMBER (10,2))');
```

### 20.1.1.4 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・レベルでのビジネス・ロジックの実装が可能になり、結果として、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。Oracle9i 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 プログラマは 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 プログラマは 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、『Oracle9i Java Stored Procedures Developer's Guide』を参照してください。

### 20.1.1.5 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 OBJECT 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。

## 20.1.2 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、データの使用法を決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。

### 20.1.2.1 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 のプラグガブル・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 のプラグガブル・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・プラグガブル・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 プラグガブル・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。実行時の XML データ検証をアクティブにするには、「XML Query」ウィザードで「有効データ・ソース」チェック・ボックスを選択します。「有効データ・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、レポートの開発時のみ有用です。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。DTD スキーマの場合は、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。

プラグガブル・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 プラグガブル・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。

### 20.1.2.2 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。

---

---

### 20.1.2.3 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、Oracle9i 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、Oracle9i データベース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
```

### 20.1.2.4 冗長なデータ

パフォーマンスへの影響を考えると、レポートの問合せでは、必要な列のみを選択し、不要な列（冗長な問合せ）を選択しないことが理想的です。問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。単一問合せによるデータ・モデルは、複数問合せによるデータ・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、異なる問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、冗長な問合せを無効にします。

## 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（問合せの解析後）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、問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・プラグブル・データソースのいずれかに基づいて問合せを定義するには、問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問合せレベルでサブセットを使用します。

### 20.1.2.5 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 問合せでは、この設定が追加の列として問合せの 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て済みます。ブレーク・グループの作成は、問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。問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。

### 20.1.2.6 グループ・フィルタ

グループ・フィルタは、データソースから取り出された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 Reports Builder 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、レポートに返される結果セットには、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。

### 20.1.2.7 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決める必要があります。

## 20.1.3 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、次の操作は実行されません。

- Web ページの所有
- レンダリング・メカニズムの制御

データが、通常の JSP ページに送られるだけです。

### 20.1.3.1 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、オブジェクトがレポートの実行時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。

オブジェクトに未定義のサイズ（可変で、水平または垂直あるいはその両方に拡張または縮小可能）があるときは、追加の処理が必要になります。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ルライズされており、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。

**20.1.3.1.1 ペーパー・レイアウトでのフォーマット・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。

フォーマット・トリガーの使用について、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。

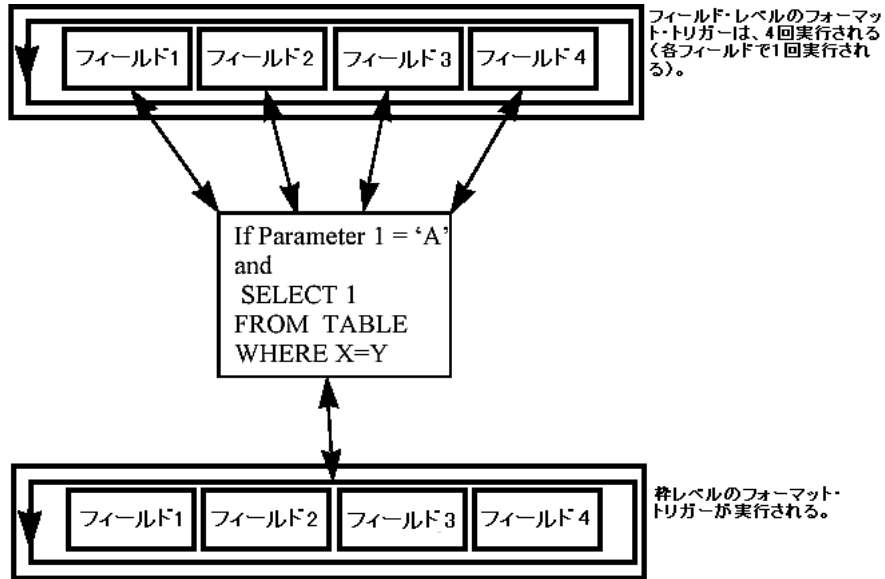
繰り返し枠には、データをフィルタリングするフォーマット・トリガーを使用することもできます。ただし、フィルタリングは適切なレベルで使用することで、レポートのパフォーマンスを向上させるだけでなく、この種のレポートで必要とされる複雑さを軽減することもできます。

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、『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。

### 20.1.3.2 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実装してから、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 Reports 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るためです。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・オプションを使用して、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。

JSP テクノロジーのパフォーマンスについてのヒントは、OTN-J (Oracle Technology Network Japan) (<http://otn.oracle.co.jp>) にあるホワイトペーパー『Oracle9i Application Server Best Practices』を参照してください。

## 20.1.4 レイアウトに関する一般的なガイドライン

この項では、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。

### 20.1.4.1 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。

#### 20.1.4.2 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、ボディ、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 では次のメッセージが表示されます。

REP-0069: 内部エラー

REP-57054: インプロセス・ジョブ終了: エラーで終了しました。

REP-594: レポート出力は生成されません。

### 20.1.5 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 Oracle Application Server Reports 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 Oracle Application Server Reports 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。

Oracle Application Server Reports 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。これが、OracleAS Forms Services から Oracle Reports をコールできる唯一の方法です。

Oracle Application Server Reports Services アプリケーションからのレポートのコールに関する詳細は、OTN-J (Oracle Technology Network Japan) (<http://otn.oracle.co.jp>) にあるホワイトペーパー『Oracle9iAS Forms Service と Oracle9iAS Reports の統合』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。

---

---

## 20.1.6 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用です。デフォルトでは、OracleAS Reports Services のタグ検証はオフです。このオプションを有効にするには、http リクエストで VALIDATETAG=YES オプションを使用します。
- デフォルトでは、rwrun コマンドと rwservlet コマンドの両方で使用される RECURSIVE\_LOAD コマンドライン・オプションが 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。このオプションは開発環境でのみ使用されるため、本番環境では、RECURSIVE\_LOAD オプションを NO に設定します。
- 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・パラメータの値を大きな値に設定します (KB 単位)。

- 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 パラメータ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。Oracle8i または Oracle9i 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・レイアウトでパラメータ・フォームが必要ない場合は、PARAMFORM オプションを NO に設定します。

PostScript に出力するときは、COPIES パラメータ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。



# 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・オプション」](#)
- [付録 B 「Reports 関連の環境変数」](#)
- [付録 C 「OracleAS Portal へのレポートのバッチ登録」](#)



---

---

# コマンドライン・オプション

この付録では、`rwclient`、`rwr`、`rwbuilder`、`rwconverter`、`rwserver`、`rwserver` などのコマンドとともに使用できるコマンドライン・オプションについて説明し、例を示します。各オプションの説明には、どのコマンドでどのオプション・キーワードを使用できるか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・オプションの使用例は、[レポート要求の実行](#)を参照してください。

---

---

この付録には、次の項があります。

- [コマンドの概要](#)
- [コマンドライン構文](#)
- [全般的な使用上の注意](#)
- [コマンドライン・キーワード](#)

## A.1 コマンドの概要

この項では、この付録で説明するキーワードとオプションを使用するコマンドについて簡単に説明します。これには、次のようなコマンドがあります。

- [rwclient](#)
- [rwrn](#)
- [rwbuilder](#)
- [rwconverter](#)
- [rwservlet](#)
- [rwcgi](#)
- [rwserver](#)

各コマンドの説明では、使用できるキーワードのリストを記載します。キーワードは、オプションとともにコマンドラインに入力する必要があります。コマンドラインに入力する際、キーワードは順不同で使用できます。

### A.1.1 キーワードと使用可能なコマンド

この項では、各種キーワードをリストし、Oracle Reports の7つのコマンドで使用できるかどうかを示します。

表 A-1 各種キーワードおよび使用できるコマンド

キーワード	<a href="#">rwclient</a>	<a href="#">rwrn</a>	<a href="#">rwbuilder</a>	<a href="#">rwconverter</a>	<a href="#">rwservlet</a>	<a href="#">rwcgi</a>	<a href="#">rwserver</a>
ACCESSIBLE	○	○	○	×	○	○	×
ARRAYSIZE	○	○	○	×	○	○	×
AUTHID	○	○	×	×	○	○	○
AUTOCOMMIT	○	○	○	×	○	○	×
AUTOSTART	×	×	×	×	×	×	○
BACKGROUND	○	×	×	×	○	○	×
BATCH	×	×	×	○	×	×	○
BCC	○	○	×	×	○	○	×
BLANKPAGES	○	○	○	×	○	○	×
BUFFERS	○	○	○	×	○	○	×
CACHELOB	○	○	○	×	○	○	×
CC	○	○	×	×	○	○	×

表 A-1 各種キーワードおよび使用できるコマンド (続き)

キーワード	rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
CELLWRAPPER	○	○	×	×	○	○	×
CMDFILE	○	○	○	○	×	×	×
CMDKEY	×	×	×	×	○	×	×
COLLATE	○	○	×	×	○	○	×
CONTENTAREA	○	○	×	×	○	○	×
COPIES	○	○	×	×	○	○	×
CUSTOMIZE	○	○	×	○	○	○	×
DATEFORMATMASK	○	○	×	×	○	○	×
DELAUTH	×	×	×	×	○	○	×
DELIMITED_HDR	○	○	×	×	○	○	×
DELIMITER	○	○	×	×	○	○	×
DESFORMAT	○	○	×	×	○	○	×
DESNAME	○	○	×	×	○	○	×
DEST	×	×	×	○	×	×	×
DESTINATION	○	○	×	×	○	○	×
DESTYPE	○	○	×	×	○	○	×
DISTRIBUTE	○	○	×	×	○	○	×
DTYPE	×	×	×	○	×	×	×
DUNIT	×	×	×	○	×	×	×
ENGINERESPONSE TIME OUT	○	×	×	×	○	○	×
EXPIRATION	○	×	×	×	○	○	×
EXPIREDAYS	×	×	×	×	○	○	×
EXPRESS_SERVER	○	○	○	×	○	○	×
FORMSIZE	×	×	×	○	×	×	×
FROM	○	○	×	×	○	○	×
GETJOBID	×	×	×	×	○	○	×
GETSERVERINFO	×	×	×	×	○	×	×
HELP	×	×	×	×	○	○	×

表 A-1 各種キーワードおよび使用できるコマンド (続き)

キーワード	rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwsgi	rwserver
INSTALL	×	×	×	×	×	×	○
ITEMTITLE	○	○	×	×	○	○	×
JOBNAME	○	×	×	×	○	○	×
JOBTYPE	○	×	×	×	○	○	×
JVMOPTIONS	×	○	○	○	×	×	○
KILLJOBID	×	×	×	×	○	○	×
LONGCHUNK	○	○	○	×	○	○	×
MIMETYPE	×	×	×	×	○	○	×
MODE	○	○	×	×	○	○	×
MODULE   REPORT	○	○	○	×	○	○	×
NONBLOCKSQL	○	○	○	×	○	○	×
NOTIFYFAILURE	○	○	×	×	○	○	×
NOTIFYSUCCESS	○	○	×	×	○	○	×
NUMBERFORMATMASK	○	○	×	×	○	○	×
OLAP_CON	○	○	○	×	○	○	×
ONFAILURE	○	○	○	×	○	○	×
ONSUCCESS	○	○	○	×	○	○	×
ORIENTATION	○	○	×	×	○	○	×
OUTPUTFOLDER	○	○	×	×	○	○	×
OUTPUTIMAGEFORMAT	○	○	○	×	○	○	×
OUTPUTPAGE	○	○	×	×	○	○	×
OVERWRITE	×	×	×	○	×	×	×
P_AVAILABILITY	×	×	×	○	×	×	×
P_DESCRIPTION	×	×	×	○	×	×	×
P_FORMATS	×	×	×	○	×	×	×
P_NAME	×	×	×	○	×	×	×
P_OWNER	×	×	×	○	×	×	×
P_PFORMTEMPLATE	×	×	×	○	×	×	×
P_PRINTERS	×	×	×	○	×	×	×

表 A-1 各種キーワードおよび使用できるコマンド (続き)

キーワード	rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
P_PRIVILEGE	×	×	×	○	×	×	×
P_SERVERS	×	×	×	○	×	×	×
P_TRIGGER	×	×	×	○	×	×	×
P_TYPES	×	×	×	○	×	×	×
PAGEGROUP	○	○	×	×	○	○	×
PAGESIZE	○	○	○	○	○	○	×
PAGESTREAM	○	○	×	×	○	○	×
PARAMFORM	×	×	×	×	○	○	×
PARSEQUERY	×	×	×	×	○	○	×
PDFCOMP	○	○	×	×	○	○	×
PDFEMBED	○	○	×	×	○	○	×
PRINTJOB	×	×	○	×	×	×	×
READONLY	○	○	○	×	○	○	×
RECURSIVE_LOAD	○	○	×	○	○	○	×
REPLACEITEM	○	○	×	×	○	○	×
REPLYTO	○	○	×	×	○	○	×
REPORT   MODULE	○	○	○	×	○	○	×
ROLE	○	○	×	×	○	○	×
RUNDEBUG	○	○	○	×	○	○	×
SAVE_RDF	×	○	○	×	×	×	×
SCHEDULE	○	×	×	×	○	○	×
SERVER	○	×	×	×	○	○	○
SHOWAUTH	×	×	×	×	○	○	×
SHOWENV	×	×	×	×	○	○	×
SHOWJOBID	×	×	×	×	○	×	×
SHOWJOBS	×	×	×	×	○	○	×
SHOWMAP	×	×	×	×	○	○	×
SHOWMYJOBS	×	×	×	×	○	×	×
SHUTDOWN	×	×	×	×	×	×	○

表 A-1 各種キーワードおよび使用できるコマンド (続き)

キーワード	rwclient	rwrn	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
SITENAME	○	○	×	×	○	○	×
SOURCE	×	×	×	○	×	×	×
SQLTRACE	○	○	○	×	○	○	×
SSOCONN	×	×	×	×	○	×	×
STATUSFOLDER	○	○	×	×	○	○	×
STATUSFORMAT	×	×	×	×	○	×	×
STATUSPAGE	○	○	×	×	○	○	×
STYPE	×	×	×	○	×	×	×
SUBJECT	○	○	×	×	○	○	×
SUPPRESSLAYOUT	○	○	○	×	○	○	×
TOLERANCE	○	×	×	×	○	○	×
TRACEFILE	○	○	○	×	×	×	×
TRACEMODE	○	○	○	×	○	○	×
TRACEOPTS	○	○	○	×	○	○	○
UNINSTALL	×	×	×	×	×	×	○
UPGRADE_PLSQL	×	×	×	○	×	×	×
URLPARAMETER	○	×	×	×	○	○	×
USERID	○	○	○	○	○	○	×
WEBSERVER_DEBUG	×	×	○	×	×	×	×
WEBSERVER_DOCROOT	×	×	○	×	×	×	×
WEBSERVER_PORT	×	×	○	×	×	×	×

## A.1.2 rwclient

rwclient コマンドは、コマンドラインを解析し、指定された（またはデフォルトの）Reports Server に転送します。rwclient 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、CMDFILE キーワードで指定されたファイルはこれらを参照しません。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 CMDFILE が再び読み取られることはありません。



## 例

**例 1:** ペーパー・レポートをブラウザ（キャッシュ）で実行

```
rwclient server=myrepserv 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=myrepserv 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=myrepserv 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=myrepserv 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=myrepserv 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=myrepserv 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=myrepserv 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=myrepserv report=myjdbcpdstest.rdf destype=file  
desformat=PDF desname=c:¥mydir¥myxml.pdf p_  
jdbcpds=syuser/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

```
AUTHIDrwclient server=myrepserver report=test.rdf  
userid=scott/tiger@mydb desformat=pdf destype=file desname=test.pdf  
=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 JOBTYPE=rwurl  
URLPARAMETER="http://www.oracle.com"
```

**例 14:** traceopts を有効化してレポートを実行

```
rwclient server=myrepserver report=test.rdf userid=scott/tiger@mydb  
destype=file desformat=pdf desname=test.pdf TRACEOPTS=trace_prf  
TRACEMODE=trace_replace
```

## A.1.3 rwrn

rwrn コマンドは、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。rep ファイルを実行する場合、PL/SQL はすでにコンパイルされているので、再コンパイルはされません。rdf ファイルを実行する場合、PL/SQL は必要に応じて自動的に再コンパイルされます。再コンパイルが必要になるのは、レポートが Reports Builder によってコンパイルおよび保存されていない場合や、レポートを実行したプラットフォームまたはバージョンが、前回レポートがコンパイルおよび保存されたプラットフォームと互換性がない場合です。

## 例

### 例 1: レポートのカスタマイズ

```
rwr run userid=scott/tiger@mydb  
report=emp.rdf CUSTOMIZE=empcustomize.xml destype=file desformat=pdf  
desname=emp.pdf
```

### 例 2: ペーパー・レポートをブラウザ（キャッシュ）で実行

```
rwr run report=test.rdf userid=scott/tiger@mydb desformat=pdf  
destype=cache
```

### 例 3: レポート出力をファイルに送信

```
rwr run report=test.rdf userid=scott/tiger@mydb desformat=pdf DESTYPE=file  
DESNAME=c:¥mydir¥test.pdf
```

### 例 4: 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 5: 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 6: 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 7: レポート出力を OracleAS Portal に送信

```
rwr run report=test.rdf userid=scott/tiger@mydb DESTYPE=oracleportal  
desformat=PDF PAGEGROUP=mypagegrp OUTPUTPAGE=reports_output  
ITEMTITLE=pushtoportal STATUSPAGE=result
```

### 例 8: 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 9:** 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=sybuser/sybpwd@server1.mydomain.com:1300
```

**例 10:** 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 11:** 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 12:** 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 13:** traceopts 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.1.4 rwbuilder

rwbuilder コマンドは、Reports Builder を起動します。コマンド・プロンプトで rwbuilder コマンドに REPORT|MODULE キーワードを追加して [Enter] を押すと、指定されたレポートがオブジェクト・ナビゲータで選択された状態で、Reports Builder が起動します。指定されたレポートがない場合、Reports Builder は、既存のレポートをオープンするか新規レポートを作成するかを選択する「Welcome ダイアログ」とともに起動します。

**例**

```
rwbuilder report=myrep.rdf userid=scott/tiger@mydb
```

## A.1.5 rwconverter

rwconverter コマンドによって、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 スクリプトのバッチ作成

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。
- ソースとして別のプラットフォーム上に作成されたレポートを使用する場合。

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 でレポートをオープンしたときにマップされます。

---

---

**例：**

```
rwconverter scott/tiger@mydb stype=rdffile source=inven1.rdf
dtype=xmlfile dest=inven1_new.xml
```

## A.1.6 rwservlet

rwservlet は、Web サーバーまたは J2EE コンテナ (OC4J など) と Reports Server 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、クライアントからの最初の要求を受信するたびに、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 コマンドを使用して JSP を実行する場合、rwservlet の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、「[レポート要求の実行](#)」を参照してください。

---

---

---

---

**注意：** キーワード [SHOWENV](#)、[SHOWJOBSSHOWMAP](#)、[SHOWMYJOBS](#)、[KILLJOBID](#)、[PARSEQUERYDELAUTH](#)、[GETJOBID](#) および [GETSERVERINFO](#) は、キーワードと値のペアではなくコマンドです。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。

---

---

**例**

```
http://myias1.mycomp.com:77792/reports/rwservlet?server=myrepserver+report=myrepo.rdf+desname=sample.pdf+desformat=pdf+destype=filehttp://myias.mycomp.com:7779/reports/rwservlet?server=myrepserver+authid3=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://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

---

<sup>1</sup> Oracle Application Server インスタンス

<sup>2</sup> Reports Servlet が実行されているポート

<sup>3</sup> 保護されたサーバーのみ

`http://myias.mycomp.com:7779/reports/rwservlet?server=myrepsserv+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=myrepsserv+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=myrepsserv+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://myias.mycomp.com:7779/reports/rwservlet?server=myrepsserv+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://myias.mycomp.com:7779/reports/rwservlet?server=myrepsserv+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

`http://myias.mycomp.com:7779/reports/rwservlet?server=myrepsserv+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=myrepsserv+report=myjdbcpdstest.rdf+destype=file+desformat=PDF+desname=c:¥mydir¥myxml.pdf+p_jdbcpds=syuser/sybpwd@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:\mydistribute.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+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://myias.mycomp.com:7779/reports/rwservlet?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://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`

\*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://myias.mycomp.com:7779/rwservlet?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://myias.mycomp.com:7779/rwservlet?server=myrepserver+report=test.rdf+userid=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+userid=scott/tiger@mydb+destype=file+desformat=pdf+desname=test.pdf+JOBTYPE=rwurl+URLPARAMETER="http://www.oracle.com"`



**例 16:** `traceopts` 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+TRACEMODE=trace_replace
```

## A.1.7 rwcgi

`rwservlet` と同様に、`rwcgi` (Common Gateway Interface: CGI) は、Web サーバーまたは J2EE コンテナ (OC4J など) と Reports Server 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、`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)。かわりに、JSP またはサーブレット (`rwservlet`) を使用することができます。

---

### 例

```
http://mywebserver.com4:77795/cgi-bin/rwcgi.exe6?server=myrepserver+report=myrepo.rdf+desname=sample.pdf+desformat=pdf+destype=file
```

```
http://mywebserver.com:7779/cgi-bin/rwcgi.sh7?server=myrepserver+authid8=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

---

<sup>4</sup> CGI スクリプトを実行している Web サーバー

<sup>5</sup> Web サーバー・リスナー・ポート

<sup>6</sup> 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

<sup>7</sup> 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

<sup>8</sup> 保護されたサーバーのみ

`http://mywebserver.com:7779/cgi-bin/rwcgi.exe?server=myrepserv+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=myrepserv+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=myrepserv+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=myrepserv+report=test.rdf+userid=scott/tiger@mydb+DESTYPE=oracleportal+desformat=PDF+PAGEGROUP=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://mywebserver.com:7779/cgi-bin/rwcgi.exe?server=myrepserv+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=myrepserv+report=myjdbcpdstest.rdf+destype=file+desformat=PDF+desname=c:¥mydir¥myxml.pdf+p_jdbcps=sybusersybpwd@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=myrepserv+report=test.rdf+userid=scott/tiger@mydb+DISTRIBUTE=yes+DESTINATION=c:¥mydistributetribute.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=myrepserv+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+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.rdf+userid=scott/tiger@mydb+destype=file+desformat=pdf+desname=test.pdf+JOBTYPE=rwurl+URLPARAMETER="http://www.oracle.com"
```

**例 16:** traceopts 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=test.pdf+TRACEOPTS=trace_prf+TRACEMODE=trace_replace
```

## A.1.8 rwservlet

rwservlet コマンドは、Reports Server を起動します。Reports Server は、クライアント要求を処理します。この処理では、クライアント要求を、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、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 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。

例:rwservlet -install myrepserver batch=yes autostart=yes

## A.2 コマンドライン構文

コマンドラインの構文は、次のとおりです。keyword=value は、有効なコマンドライン・オプションです。

```
rwclient REPORT|MODULE=runfile USERID=username/password@database[  
[keyword=value|value1,value2,...] SERVER=server_name
```

キーワードは必ず指定する必要がありますが、コマンドの後ろに順不同で使用できます。

## A.3 全般的な使用上の注意

- 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、コマンド・ファイル・キーワード (CMDFILE=) に対して指定されたファイルはこれらを参照しません。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 CMDFILE が再び読み取られることはありません。
- 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。

## A.4 コマンドライン・キーワード

この項では、Oracle Reports で使用できる各種コマンドライン・キーワードについて説明します。

### A.4.1 ACCESSIBLE

表 A-2 では、ACCESSIBLE キーワードを使用できるコマンドを示します。

**表 A-2 ACCESSIBLE キーワードを使用できるコマンド**

rwclient	rwruntime	rwbuilder	rwconverter	rwservlet	rwcgi	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.4.2 ARRAYSIZE

表 A-3 では、ARRAYSIZE キーワードを使用できるコマンドを示します。

**表 A-3 ARRAYSIZE キーワードを使用できるコマンド**

rwclient	rwruntime	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
○	○	○	×	○	○	×

**説明** ARRAYSIZE を使用して、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す。

**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 レポートを実行している場合に使用できます。

### A.4.3 AUTHID

表 A-4 では、AUTHID キーワードを使用できるコマンドを示します。

**表 A-4 AUTHID キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
○	○	×	×	○	○	○

**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、レポート要求を行っているユーザーが要求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。

**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せてください。

**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 レポートを実行している場合に使用できます。
- シングル・サインオン環境では、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.4.4 AUTOCOMMIT

表 A-5 では、AUTOCOMMIT キーワードを使用できるコマンドを示します。

**表 A-5 AUTOCOMMIT キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
○	○	○	×	○	○	×

**説明** AUTOCOMMIT を使用して、データベースへの変更 (INSERT 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、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使用できます。

## A.4.5 AUTOSTART

表 A-6 では、AUTOSTART キーワードを使用できるコマンドを示します。

**表 A-6 AUTOSTART キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
×	×	×	×	×	×	○

**説明** AUTOSTART を使用して、Reports Server が初回インストール後にリブートした後、ユーザー・ログインなしで自動的に起動することを指定します。

**構文** AUTOSTART={YES|NO}

**値**

- YES の場合、サービスがリブート後にユーザー・ログインなしで自動的に起動します。
- NO の場合、サービスがリブート後に自動的に起動せず、ユーザー・ログインを要求します。

**デフォルト** NO

**使用上の注意** AUTOSTART キーワードは、Microsoft Windows プラットフォームのみで認識されます。

## A.4.6 BACKGROUND

表 A-7 では、BACKGROUND キーワードを使用できるコマンドを示します。

**表 A-7 BACKGROUND キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	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 (YES) 実行するかを指定できます。

**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 または rwrn 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.4.7 BATCH

表 A-8 では、BATCH キーワードを使用できるコマンドを示します。

**表 A-8 BATCH キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
×	×	×	○	×	×	○

**説明** すべての関連コマンドで、BATCH オプションは、非 UI モードでの実行をサーバーに指示します。コマンドでの使用方法は、アプリケーションによって UI が生成されないという点で、BATCH=YES を含むコマンドラインからの実行と同じです。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することができ、実行中にサーバー・ダイアログが表示されません。

rwconverter で BATCH 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非表示にします。rwserver で BATCH を使用すると、サーバー・ウィンドウ・ダイアログがオフ (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)。
- 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、「[DESNAME](#)」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.4.8 BCC

表 A-9 では、BCC キーワードを使用できるコマンドを示します。

**表 A-9 BCC キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	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 レポートを実行している場合に使用できません。

## A.4.9 BLANKPAGES

表 A-10 では、BLANKPAGES キーワードを使用できるコマンドを示します。

**表 A-10 BLANKPAGES キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	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.4.10 BUFFERS

表 A-11 では、BUFFERS キーワードを使用できるコマンドを示します。

表 A-11 BUFFERS キーワードを使用できるコマンド

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	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 レポートを実行している場合に使用できます。

## A.4.11 CACHELOB

表 A-12 では、CACHELOB キーワードを使用できるコマンドを示します。

表 A-12 CACHELOB キーワードを使用できるコマンド

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	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・ディレクトリの位置に十分なディスク領域がない場合、この値はNOに設定することをお勧めします。ただし、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生じるため、パフォーマンスが低下する可能性があります。
- CACHELOB は、コマンドラインから JSP ベースの Web レポートを実行している場合に使用できます。

## A.4.12 CC

表 A-13 では、CC キーワードを使用できるコマンドを示します。

**表 A-13 CC キーワードを使用できるコマンド**

rwclient	rwrwn	rwbuilder	rwconverter	rwservlet	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.4.13 CELLWRAPPER

表 A-14 では、CELLWRAPPER キーワードを使用できるコマンドを示します。

**表 A-14 CELLWRAPPER キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	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-15 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-16 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-17 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。

**表 A-17 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 (続き)**

値	説明
¥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.4.14 CMDFILE

表 A-18 では、CMDFILE キーワードを使用できるコマンドを示します。

**表 A-18 CMDFILE キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	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 `rwcgi` 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) を上書きします。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.4.15 CMDKEY**

表 A-19 では、`CMDKEY` キーワードを使用できるコマンドを示します。

**表 A-19 CMDKEY キーワードを使用できるコマンド**

<code>rwclient</code>	<code>rwrund</code>	<code>rwbuilder</code>	<code>rwconverter</code>	<code>rwervlet</code>	<code>rwcgi</code>	<code>rwserver</code>
×	×	×	×	○	×	×

**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://yourwebserver/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せん (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 レポートを実行している場合に使用できます。

## A.4.16 COLLATE

表 A-20 では、COLLATE キーワードを使用できるコマンドを示します。

**表 A-20 COLLATE キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	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.4.17 CONTENTAREA

表 A-21 では、CONTENTAREA キーワードを使用できるコマンドを示します。

**表 A-21 CONTENTAREA キーワードを使用できるコマンド**

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 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。[SITENAME](#) を参照してください。OracleAS Portal リリース 2 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](#)、[REPLACEITEM](#)、[SITENAME\\*](#)、[STATUSFOLDER\\*](#)、[STATUSPAGE](#) です。

\*Oracle9iAS Portal リリース 1 または Oracle WebDB リリース 2.2（あるいはその両方）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.4.18 COPIES

表 A-22 では、COPIES キーワードを使用できるコマンドを示します。

表 A-22 COPIES キーワードを使用できるコマンド

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	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.4.19 CUSTOMIZE

表 A-23 では、CUSTOMIZE キーワードを使用できるコマンドを示します。

**表 A-23 CUSTOMIZE キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
○	○	×	○	○	○	×

**説明** CUSTOMIZE を使用して、現在のレポートに対して Oracle Reports XML ファイルが実行されるよう指定します。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、「[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 レポートを実行している場合に使用できます。

## A.4.20 DATEFORMATMASK

表 A-24 では、DATEFORMATMASK キーワードを使用できるコマンドを示します。

表 A-24 DATEFORMATMASK キーワードを使用できるコマンド

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、Reports Builder 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 レポートを実行している場合に使用できます。

## A.4.21 DELAUTH

表 A-25 では、DELAUTH キーワードを使用できるコマンドを示します。

表 A-25 DELAUTH キーワードを使用できるコマンド

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
×	×	×	×	○	○	×

**説明** DELAUTH を使用して、rwservlet または rwcgi のユーザー ID Cookie を削除します。

**構文**

`http://yourwebserver/reports/rwservlet/delauth[?] [server=servername] [&auth id=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.4.22 DELIMITED\_HDR

表 A-26 では、DELIMITED\_HDR キーワードを使用できるコマンドを示します。

表 A-26 DELIMITED\_HDR キーワードを使用できるコマンド

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	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・ヘッダーなどのボイラープレート・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イラープレート・テキストをすべてオフにします。
- NO の場合、デリミタ付き出力ファイルのボイラープレート・テキストをそのまま残します。

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.4.23 DELIMITER

表 A-27 では、DELIMITER キーワードを使用できるコマンドを示します。

表 A-27 DELIMITER キーワードを使用できるコマンド

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwsgi	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-28 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-29 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-30 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.4.24 DESFORMAT

表 A-31 では、DESFORMAT キーワードを使用できるコマンドを示します。

**表 A-31 DESFORMAT キーワードを使用できるコマンド**

rwclient	rwrun	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
○	○	×	×	○	○	×

**説明** DESFORMAT を使用して、ジョブ出力のフォーマットを指定します。DESFORMAT を使用すると、DESTYPE が FILE で、DESNAME=*filename* であるときには使用されるプリンタ・ドライバを指定できます。

**構文** DESFORMAT=*desformat*

**値** 長さが 1KB 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-32 を参照してください。

**表 A-32 DESFORMAT の有効値**

値	説明
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 TAB が使用されます。「使用上の注意」を参照してください。
DELIMITEDDATA	DELIMITED と同じ機能ですが、DESFORMAT=DELIMITED を指定して大量のレポートを実行する際に問題がある場合に使用します。「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 形式のファイルに出力されます。

表 A-32 DESFORMAT の有効値 (続き)

値	説明
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せん。
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する必要があります。
POSTSCRIPT	このレポート出力は、ポストスクリプト形式のファイルに出力されます。
PRINTER DEFINITION	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。 <b>MODE=BITMAP</b> の場合、これはプリンタ名です。 <b>MODE=CHARACTER</b> の場合、これは、hpl、hplwide、dec、decwide、decland、dec180、dflt、wide などのキャラクタモードのプリンタ定義ファイル (.prt ファイル) です。有効な <i>printer definition</i> 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 でファイルを開く場合、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。
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開いたり読み取ることができます。

**デフォルト** DESFORMAT 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)。Reports Builder でレポートを実行する際に 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。ドロップダウン・リストからプリンタ名が選択されていない場合、デフォルトで PostScript が使用されます。

### 使用上の注意

- 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 `&nbsp;` 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 出力でスペースを `&nbsp;` に置き換えない場合は、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・メッセージが表示されます。

REP-34305:destid='DEST1' のキーワードの設定が無効です。

- DESFORMAT=DELIMITEDDATA の場合、DelimitedData ドライバはデータ・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変更は 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。

また、DelimitedData 機能では DELIMITED と同様に、DELIMITER、CELLWRAPPER、NUMBERFORMATMASK および DATEFORMATMASK のコマンドライン・オプションを使用できます。

- 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.4.25 DESNAME

表 A-33 では、DESNAME キーワードを使用できるコマンドを示します。

表 A-33 DESNAME キーワードを使用できるコマンド

rwclient	rwrn	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、OracleAS Portal、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 OraclePortal 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。

## A.4.26 DEST

表 A-34 では、DEST キーワードを使用できるコマンドを示します。

**表 A-34 DEST キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	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 RDFFILE のとき、DEST のデフォルト名は source.rdf です。
- DTYPE が REPFIL 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す。

## A.4.27 DESTINATION

表 A-35 では、DESTINATION キーワードを使用できるコマンドを示します。

**表 A-35 DESTINATION キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	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** も指定すると、これらは無視されます。

---



---

**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.4.28 DESTYPE

表 A-36 では、DESTYPE キーワードを使用できるコマンドを示します。

**表 A-36 DESTYPE キーワードを使用できるコマンド**

rwclient	rwrwn	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
○	○	×	×	○	○	×

**説明** DESTYPE を使用して、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。

### 構文

```
DESTYPE={CACHE|LOCALFILE|FILE|PRINTER|SYSOUT|MAIL|ORACLEPORTAL|FTP|W  
EBDAV|name_of_pluggable_destination}
```

**値** 表 A-37 では、DESTYPE キーワードで使用できる値を説明します。

**表 A-37 DESTYPE キーワードの有効値**

値	説明
CACHE	出力は Reports Server のキャッシュに直接送信されます。
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。

表 A-37 DESTYPE キーワードの有効値 (続き)

値	説明
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せん。 <b>注意:</b> 構成ファイル <code>report_server_name.conf</code> には、 <code>&lt;pluginParam name="mailServer"&gt;</code> で送信先メール・サーバー名を指定しておく必要があります。これは、Windows と UNIX のどちらの環境にも該当します。
ORACLEPORTAL	rwervlet、rwcgi および rwclient に対して使用できます。 出力は OracleAS Portal に送信されます。関連するキーワードは、 <a href="#">CONTENTAREA*</a> 、 <a href="#">EXPIREDAYS</a> 、 <a href="#">ITEMTITLE</a> 、 <a href="#">OUTPUTFOLDER*</a> 、 <a href="#">OUTPUTPAGE</a> 、 <a href="#">PAGEGROUP</a> 、 <a href="#">REPLACEITEM</a> 、 <a href="#">SCHEDULE</a> 、 <a href="#">SITENAME*</a> 、 <a href="#">STATUSFOLDER*</a> 、 <a href="#">STATUSPAGE</a> です。  *Oracle9iAS Portal リリース 1 または Oracle WebDB リリース 2.2 (あるいはその両方) 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。
FTP	rwervlet、rwclient、rwcgi および rwrn に対して使用できます。出力は指定された FTP サーバーに送信されます。
WEBDAV	rwervlet、rwclient、rwcgi および rwrn に対して使用できます。レポートを直接公開できるように、出力は指定された WebDAV URL に送信されます。
<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 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

- Oracle Reports では、DESTYPE 値の SCREEN と PREVIEW は有効ではありません。Reports Runtime (rwrn) ユーザー・インタフェースは廃止されました。Reports Builder (rwbuilder) では、DESTYPE システム・パラメータを SCREEN または PREVIEW に設定してレポートをフォーマットし、スクリーン・フォントまたはプリンタ・フォント (preview) を Reports Builder ユーザー・インタフェースに表示できます。

- **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 を有効にする方法は、OracleAS Portal オンライン・ヘルプを参照してください。

---

DESTYPE=ORACLEPORTAL は、rwruntime 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。

Oracle Reports 出力を Oracle9iAS Portal リリース 1 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 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 Oracle9iAS Portal リリース 1 に配信するには、portalUserid のコメントを外したままにしておく必要があります。

Oracle Reports 10g (9.0.4) では、デフォルトで portalUserid がコメント化されています。Reports Server は、接続文字列を判断し、レポートを Oracle9iAS Portal リリース 1 に配信します。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 Oracle9iAS Portal リリース 1 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: 「宛先タイプ oracleportal へのクラスが定義されていません。」

---



---

要求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@repportal&authid=pushportal/trial&destype=oracleportal&desformat=PDF&pagegroup=PORTAL_
REPORTS&outputpage=reports_
output&itemtitle=pushtoportal&statuspage=result
```

- DESTYPE=FTP: 要求の実行方法は他のデフォルトの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=ftp://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、Reports Server 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。このプロパティ・エントリのコメントを外し、`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:¥¥9iAS¥reports¥¥conf¥¥proxyinfo.xml"/>
</destination>
```

- **DESTYPE=WEBDAV:** 要求の実行方法は他のデフォルトの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、`Reports Server` 構成ファイルの `destination` 要素 (デフォルトの `WebDAV` 構成) を編集します。

```
<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。このプロパティ・エントリのコメントを外し、`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="WebDav"
class="oracle.reports.plugin.destination.webdav.DesWebDAV">
<property name="proxy" value="D:¥¥Oracle¥reports¥¥conf¥¥proxyinfo.xml"/>
</destination>
```

## A.4.29 DISTRIBUTE

表 A-38 では、DISTRIBUTE キーワードを使用できるコマンドを示します。

**表 A-38 DISTRIBUTE キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	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](#) も指定すると、これらは無視されます。

---

**注意：** 拡張配布の作成の詳細は、「[拡張配布の作成](#)」を参照してください。

---

## A.4.30 DTYPE

表 A-39 では、DTYPE キーワードを使用できるコマンドを示します。

**表 A-39 DTYPE キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	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できます。

### 構文

```
DTYPE={ PLDFILE | PLLFILE | RDIFFILE | REPFIL | REXFILE | TDDFILE | 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。
- RDIFFILE 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。
- REPFIL 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。
- TDDFILE 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。

**デフォルト** REPFIL

### 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。この事態を回避するには、.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」→「すべて」を使用して、ソース・レポートの PL/SQL をコンパイルします。

- 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.4.31 DUNIT

表 A-40 では、DUNIT キーワードを使用できるコマンドを示します。

**表 A-40 DUNIT キーワードを使用できるコマンド**

rwclient	rwrn	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.4.32 ENGINERESPONSETIMEOUT

表 A-41 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-41 ENGINERESPONSETIMEOUT キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	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、「[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.4.33 EXPIRATION

表 A-42 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-42 EXPIRATION キーワードを使用できるコマンド**

rwclient	rwruntime	rwbuilder	rwconverter	rwservlet	rwcgi	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、「[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。

**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 の DateFormat.MEDIUM タイプの定義と同じです。

デフォルト なし

## A.4.34 EXPIREDDAYS

表 A-43 では、EXPIREDDAYS キーワードを使用できるコマンドを示します。

表 A-43 EXPIREDDAYS キーワードを使用できるコマンド

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
×	×	×	×	○	○	×

**説明** EXPIREDDAYS を使用して、Oracle9iAS Portal リリース 1 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** EXPIREDDAYS={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 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\\*](#)、[EXPIREDDAYS](#)、[ITEMTITLE](#)、[OUTPUTFOLDER\\*](#)、[OUTPUTPAGE](#)、[PAGEGROUP](#)、[REPLACEITEM](#)、[SITENAME\\*](#)、[STATUSFOLDER\\*](#)、[STATUSPAGE](#) です。

\*Oracle9iAS Portal リリース 1 または Oracle WebDB リリース 2.2 (あるいはその両方) 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.4.35 EXPRESS\_SERVER

表 A-44 では、EXPRESS\_SERVER キーワードを使用できるコマンドを示します。

表 A-44 EXPRESS\_SERVER キーワードを使用できるコマンド

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	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=ramconn"

**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-45 接続文字列値**

値	説明
server	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。
user	OLAP Server にログインするためのユーザー ID。
password	ユーザー ID に対するパスワード。
ramuser	RDBMS にログインするためのユーザー ID。
rampassword	RDBMS に対するパスワード。
ramexpressid	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。
ramserverscript	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、同じドライブ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、¥¥ServerName¥¥ShareName¥¥の後に任意のサブフォルダまたはファイルが続きます。
rammasterdb	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。

表 A-45 接続文字列値 (続き)

値	説明
ramconnecttype	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-46 に、パラメータとその設定を示します。

表 A-46 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
sl	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  <b>注意:</b> Windows NT 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/sl=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.4.36 FORMSIZE

表 A-47 では、FORMSIZE キーワードを使用できるコマンドを示しています。

表 A-47 FORMSIZE キーワードを使用できるコマンド

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	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.4.37 FROM

表 A-48 では、FROM キーワードを使用できるコマンドを示します。

表 A-48 FROM キーワードを使用できるコマンド

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	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.4.38 GETJOBID

表 A-49 では、GETJOBID キーワードを使用できるコマンドを示します。

表 A-49 GETJOBID キーワードを使用できるコマンド

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	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://yourwebserver/reports/rwservlet/getjobidn[?][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.4.39 GETSERVERINFO

表 A-50 では、GETSERVERINFO キーワードを使用できるコマンドを示します。

表 A-50 GETSERVERINFO キーワードを使用できるコマンド

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	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://yourwebserver/reports/rwservlet/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.4.40 HELP

表 A-51 では、HELP キーワードを使用できるコマンドを示します。

表 A-51 HELP キーワードを使用できるコマンド

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	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、`rwservlet` 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://yourwebserver/reports/rwservlet/help?command=keyword
```

**構文** `http://yourwebserver/reports/rwservlet/help`

または

`http://yourwebserver/reports/rwservlet/help?command=keyword`

**値** 構文を参照してください。

**デフォルト** なし

## A.4.41 INSTALL

表 A-52 では、INSTALL キーワードを使用できるコマンドを示します。

**表 A-52 INSTALL キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
×	×	×	×	×	×	○

**説明** INSTALL を使用して、Microsoft Windows 上の Reports Server インスタンスをサービスとして構成できます。このキーワードは UNIX プラットフォームでは無効です。

**構文** `INSTALLreports_server_name`

**値**

`reports_server_name` には、Reports Server インスタンスの有効な名前を指定します。

**デフォルト** なし

### 使用上の注意

- INSTALL とともに AUTOSTART キーワードを使用すると、インストール後およびシステム再起動後に Reports Server サービスが自動的に開始されます。
- INSTALL とともに BATCH=YES を使用すると、通常インストール時に表示されるプロンプトおよびダイアログが表示されなくなります。

## A.4.42 ITEMTITLE

表 A-53 では、ITEMTITLE キーワードを使用できるコマンドを示します。

**表 A-53 ITEMTITLE キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
○	○	×	×	○	○	×

**説明** ITEMTITLE を使用して、Oracle9iAS Portal リリース 1 が Oracle Reports 出力に使用する表示名を指定できます。Oracle9iAS Portal リリース 1 にはこの名前が表示され、Oracle Reports 出力にリンクされます。

**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 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、REPLACEITEM、SITENAME\*、STATUSFOLDER\*、STATUSPAGE です。

\*Oracle9iAS Portal リリース 1 または Oracle WebDB リリース 2.2 (あるいはその両方) 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.4.43 JOBNAME

表 A-54 では、JOBNAME キーワードを使用できるコマンドを示します。

表 A-54 JOBNAME キーワードを使用できるコマンド

rwclient	rwrwn	rwbuilder	rwconverter	rwservlet	rwcgi	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** 任意のジョブ名

**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 レポートを実行している場合に使用できます。

## A.4.44 JOBTYP

表 A-55 では、JOBTYP キーワードを使用できるコマンドを示します。

**表 A-55 JOBTYP キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	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。

**デフォルト** REPORT

## A.4.45 JVMOPTI

表 A-56 では、JVMOPTI キーワードを使用できるコマンドを示します。

**表 A-56 JVMOPTI キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
×	○	○	○	×	×	○

**説明** JVMOPTI を使用して、Reports Runtime、Reports Builder、Reports Converter または Reports Server の Java Virtual Machine (JVM) にオプションを設定できます。たとえば、次のようなコマンドラインを使用して、Reports Server を 512MB のヒープ領域で起動できます。

```
rwserver server= servername jvmoptions=-Xmx512M
```

複数のオプションを指定できますが、その場合はオプションを二重引用符で囲む必要があります。

```
rwserver server=servername jvmoptions="-Xmx256M -Xms=128M"
```

また、次のようなコマンドラインを使用して、Reports Builder を 512MB のヒープ領域で起動することもできます。

```
rwbuilder jvmoptions=-Xmx512M
```

**構文** JVMOPTIONS={options in the server's JVM}  
 rwbuilder jvmoptions="-Xmx256M -Xms=128M"

## A.4.46 KILLJOBID

表 A-57 では、KILLJOBID キーワードを使用できるコマンドを示します。

**表 A-57 KILLJOBID キーワードを使用できるコマンド**

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://yourwebserver/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

- 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.4.47 LONGCHUNK

表 A-58 では、LONGCHUNK キーワードを使用できるコマンドを示します。

表 A-58 LONGCHUNK キーワードを使用できるコマンド

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	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 データベースのみに適用されます。

**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 レポートを実行している場合に使用できます。

## A.4.48 MIMETYPE

表 A-59 では、MIMETYPE キーワードを使用できるコマンドを示します。

表 A-59 MIMETYPE キーワードを使用できるコマンド

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
×	×	×	×	○	○	×

**説明** MIMETYPE を使用して、Web の出力が返されるときに、Reports Server で割り当てられた MIME タイプを無効にできます。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.4.49 MODE

表 A-60 では、MODE キーワードを使用できるコマンドを示します。

**表 A-60 MODE キーワードを使用できるコマンド**

rwclient	rwruntime	rwbuilder	rwconverter	rwservlet	rwcgi	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** 次の値が適用されます。

- 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.4.50 MODULE|REPORT

表 A-61 では、MODULE|REPORT キーワードを使用できるコマンドを示します。

**表 A-61 MODULE|REPORT キーワードを使用できるコマンド**

rwclient	rwruntime	rwbuilder	rwconverter	rwservlet	rwcgi	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.4.51 NONBLOCKSQL

表 A-62 では、NONBLOCKSQL キーワードを使用できるコマンドを示します。

**表 A-62 NONBLOCKSQL キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	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 レポートを実行している場合に使用できます。

## A.4.52 NOTIFYFAILURE

表 A-63 では、NOTIFYFAILURE キーワードを使用できるコマンドを示します。

**表 A-63 NOTIFYFAILURE キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
○	○	×	×	○	○	×

**説明** NOTIFYFAILURE を使用して、レポート要求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。

---

**注意：** sender プロパティの詳細は、「[notification](#)」を参照してください。このプロパティは、ドメイン名の確認に際し、SMTP サーバーで発生した問題を修正するものです。

---

**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 レポートを実行している場合に使用できます。

## A.4.53 NOTIFYSUCCESS

表 A-64 では、NOTIFYSUCCESS キーワードを使用できるコマンドを示します。

表 A-64 NOTIFYSUCCESS キーワードを使用できるコマンド

rwclient	rwrwn	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
○	○	×	×	○	○	×

**説明** NOTIFYSUCCESS を使用して、レポート要求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。

**注意：** sender プロパティの詳細は、「notification」を参照してください。このプロパティは、ドメイン名の確認に際し、SMTP サーバーで発生した問題を修正するものです。

**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 レポートを実行している場合に使用できます。

## A.4.54 NUMBERFORMATMASK

表 A-65 では、NUMBERFORMATMASK キーワードを使用できるコマンドを示します。

表 A-65 NUMBERFORMATMASK キーワードを使用できるコマンド

rwclient	rwrwn	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、Reports Builder のオンライン・ヘルプのトピック「数値書式マスク構文」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 レポートを実行している場合に使用できます。

## A.4.55 OLAP\_CON

表 A-66 では、OLAP\_CON キーワードを使用できるコマンドを示します。

**表 A-66 OLAP\_CON キーワードを使用できるコマンド**

rwclient	rwrund	rwbuilder	rwconverter	rwservlet	rwcgi	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できます。

**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.4.56 ONFAILURE

表 A-67 では、ONFAILURE キーワードを使用できるコマンドを示します。

表 A-67 ONFAILURE キーワードを使用できるコマンド

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	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 レポートを実行している場合に使用できます。

## A.4.57 UNSUCCESS

表 A-68 では、UNSUCCESS キーワードを使用できるコマンドを示します。

表 A-68 UNSUCCESS キーワードを使用できるコマンド

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	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 レポートを実行している場合に使用できます。

## A.4.58 ORIENTATION

表 A-69 では、ORIENTATION キーワードを使用できるコマンドを示します。

**表 A-69 ORIENTATION キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
○	○	×	×	○	○	×

**説明** ORIENTATION は、レポートの各ページのプリント方向をコントロールします。

**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**

- 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。
- 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.4.59 OUTPUTFOLDER

表 A-70 では、OUTPUTFOLDER キーワードを使用できるコマンドを示します。

**表 A-70 OUTPUTFOLDER キーワードを使用できるコマンド**

rwclient	rwrwn	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
○	○	×	×	○	○	×

**説明** OUTPUTFOLDER を使用して、Oracle Reports 出力の配信先となる Oracle WebDB リリース 2.2 または OracleAS Portal 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](#)、[REPLACEITEM](#)、[SITENAME\\*](#)、[STATUSFOLDER\\*](#)、[STATUSPAGE](#) です。

\*Oracle9iAS Portal リリース 1 または Oracle WebDB リリース 2.2 (あるいはその両方) 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.4.60 OUTPUTIMAGEFORMAT

表 A-71 では、OUTPUTIMAGEFORMAT キーワードを使用できるコマンドを示します。

**表 A-71 OUTPUTIMAGEFORMAT キーワードを使用できるコマンド**

rwclient	rwrund	rwbuilder	rwconverter	rwservlet	rwsgi	rwserver
○	○	○	×	○	○	×

**説明** OUTPUTIMAGEFORMAT を使用して、レポート出力のイメージの形式を選択します。

**構文** OUTPUTIMAGEFORMAT={PNG|JPEG|JPG|GIF|BMP}

### 値

- PNG、JPEG、JPG は、DESFORMAT 値が PDF、HTML、HTMLCSS または RTF の場合にサポートされているイメージ形式です。
- GIF は、DESFORMAT 値が PDF、HTML または HTMLCSS の場合にサポートされているイメージ形式です。
- BMP は、DESFORMAT 値が RTF の場合にサポートされているイメージ形式です。

### 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。

### 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
```

```
rwruntime 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
```

```
rwruntime 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: 指定された出力イメージ形式はサポートされていません。

## A.4.61 OUTPUTPAGE

表 A-72 では、OUTPUTPAGE キーワードを使用できるコマンドを示します。

表 A-72 OUTPUTPAGE キーワードを使用できるコマンド

rwclient	rwruntime	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
○	○	×	×	○	○	×

**説明** OUTPUTPAGE を使用して、レポート出力情報の配信先となる Oracle Reports 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 Oracle9iAS Portal リリース 1 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 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。
  - 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、REPLACEITEM、SITENAME\*、STATUSFOLDER\*、STATUSPAGE です。
 

\*Oracle9iAS Portal リリース 1 または Oracle WebDB リリース 2.2 (あるいはその両方) 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.4.62 OVERWRITE

表 A-73 では、OVERWRITE キーワードを使用できるコマンドを示します。

表 A-73 OVERWRITE キーワードを使用できるコマンド

rwclient	rwrwn	rwbuilder	rwconverter	rwservlet	rwcgi	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.4.63 P\_AVAILABILITY

表 A-74 では、P\_AVAILABILITY キーワードを使用できるコマンドを示します。

表 A-74 P\_AVAILABILITY キーワードを使用できるコマンド

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
×	×	×	○	×	×	×

**説明** P\_AVAILABILITY は使用可能日カレンダーの名前で、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.4.64 P\_DESCRIPTION

表 A-75 では、P\_DESCRIPTION キーワードを使用できるコマンドを示します。

表 A-75 P\_DESCRIPTION キーワードを使用できるコマンド

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	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.4.65 P\_FORMATS

表 A-76 では、P\_FORMATS キーワードを使用できるコマンドを示します。

表 A-76 P\_FORMATS キーワードを使用できるコマンド

rwclient	rwruntime	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せん。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。

**デフォルト** なし

## A.4.66 P\_NAME

表 A-77 では、P\_NAME キーワードを使用できるコマンドを示します。

表 A-77 P\_NAME キーワードを使用できるコマンド

rwclient	rwruntime	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、OracleAS Portal に登録するために呼びだされる 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、OracleAS Portal に登録するために呼びだされる 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値を使用できません（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.4.67 P\_OWNER

表 A-78 では、P\_OWNER キーワードを使用できるコマンドを示します。

**表 A-78 P\_OWNER キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	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.4.68 P\_PFORMTEMPLATE

表 A-79 では、P\_PFORMTEMPLATE キーワードを使用できるコマンドを示します。

**表 A-79 P\_PFORMTEMPLATE キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	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.4.69 P\_PRINTERS

表 A-80 では、P\_PRINTERS キーワードを使用できるコマンドを示します。

**表 A-80 P\_PRINTERS キーワードを使用できるコマンド**

rwclient	rwrund	rwbuilder	rwconverter	rwservlet	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.4.70 P\_PRIVILEGE

表 A-81 では、P\_PRIVILEGE キーワードを使用できるコマンドを示します。

**表 A-81 P\_PRIVILEGE キーワードを使用できるコマンド**

rwclient	rwrund	rwbuilder	rwconverter	rwservlet	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限を持つユーザ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.4.71 P\_SERVERS

表 A-82 では、P\_SERVERS キーワードを使用できるコマンドを示します。

**表 A-82 P\_SERVERS キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	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.4.72 P\_TRIGGER

表 A-83 では、P\_TRIGGERS キーワードを使用できるコマンドを示します。

**表 A-83 P\_TRIGGER キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	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.4.73 P\_TYPES

表 A-84 では、P\_TYPES キーワードを使用できるコマンドを示します。

**表 A-84 P\_TYPES キーワードを使用できるコマンド**

rwclient	rwrwn	rwbuilder	rwconverter	rwservlet	rwcgi	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.4.74 PAGEGROUP

表 A-85 では、PAGEGROUP キーワードを使用できるコマンドを示します。

表 A-85 PAGEGROUP キーワードを使用できるコマンド

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
○	○	×	×	○	○	×

**説明** PAGEGROUP 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 ページ・グループの名前を指定できます。旧バージョン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 Oracle9iAS Portal リリース 1 で作成しておく必要があります。

**構文** PAGEGROUP=*pagegroup*

### 値

*pagegroup* には、Oracle9iAS Portal リリース 1 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 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。
- 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、Oracle9iAS Portal リリース 1 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](#)、[REPLACEITEM](#)、[SITENAME\\*](#)、[STATUSFOLDER\\*](#)、[STATUSPAGE](#) です。

\*Oracle9iAS Portal リリース 1 または Oracle WebDB リリース 2.2 (あるいはその両方) 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.4.75 PAGESIZE

表 A-86 では、PAGESIZE キーワードを使用できるコマンドを示します。

表 A-86 PAGESIZE キーワードを使用できるコマンド

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ります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** フォームに対する任意の有効なページ・ディメンションを指定します（ページ幅×ページ高さ：それぞれ 0 より大きい数）。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」→「環境設定」→「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ている場合、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もあります。Reports Builder で、使用するプリンタが許容する印刷可能な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、レポート定義のページ・ディメンションはその値によって上書きされます。
- コマンドラインに入力された PAGESIZE 値は、ランタイム・パラメータ・フォームに入力された PAGESIZE 値によって上書きされます。

## A.4.76 PAGESTREAM

表 A-87 では、PAGESTREAM キーワードを使用できるコマンドを示します。

**表 A-87 PAGESTREAM キーワードを使用できるコマンド**

rwclient	rwruntime	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
○	○	×	×	○	○	×

**説明** PAGESSTREAM 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** PAGESSTREAM={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.4.77 PARAMFORM

表 A-88 では、PARAMFORM キーワードを使用できるコマンドを示します。

**表 A-88 PARAMFORM キーワードを使用できるコマンド**

rwclient	rwrwn	rwbuilder	rwconverter	rwervlet	rwsgi	rwserver
×	×	×	×	○	○	×

**説明** PARAMFORM を使用して、レポートを CGI またはサーブレット経由で実行するときにランタイム・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、Web ソース・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、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。
- レポートを OracleAS Portal 環境で実行するときは、このキーワードを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フォームと競合する可能性があるためです。

## A.4.78 PARSEQUERY

表 A-89 では、PARSEQUERY キーワードを使用できるコマンドを示します。

表 A-89 PARSEQUERY キーワードを使用できるコマンド

rwclient	rwrund	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
×	×	×	×	○	○	×

**説明** PARSEQUERY を使用して、rwervlet 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://yourwebserver/reports/rwervlet/parsequery[?] [server=servername] [&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.4.79 PDFCOMP

表 A-90 では、PDFCOMP キーワードを使用できるコマンドを示します。

表 A-90 PDFCOMP キーワードを使用できるコマンド

rwclient	rwrund	rwbuilder	rwconverter	rwervlet	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、任意の値 0～9 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.4.80 PDFEMBED

表 A-90 では、PDFEMBED キーワードを使用できるコマンドを示します。

**表 A-91 PDFEMBED キーワードを使用できるコマンド**

rwclient	rwrund	rwbuilder	rwconverter	rwservlet	rwsgi	rwserver
○	○	×	×	○	○	×

**説明** PDFEMBED を使用して、uifont.ali ファイルで指定した 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.4.81 PRINTJOB

表 A-92 では、PRINTJOB キーワードを使用できるコマンドを示します。

**表 A-92 PRINTJOB キーワードを使用できるコマンド**

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** NO

**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.4.82 READONLY

表 A-93 では、READONLY キーワードを使用できるコマンドを示します。

**表 A-93 READONLY キーワードを使用できるコマンド**

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要求できます。Oracle データベースへのアクセス時に、SET TRANSACTION READ ONLY 文によって読取り一貫性を持たせることができます。

---



---

**注意：** SET TRANSACTION READ ONLY の詳細は、SQL のドキュメントを参照してください（Oracle Technology Network Japan (<http://otn.oracle.co.jp/>) にアクセスしてください）。

---



---

**構文** READONLY={YES|NO}

**値**

YES の場合、読取り一貫性を要求します。

NO の場合、読取り一貫性を要求しません。

**デフォルト** NO

**使用上の注意**

- このキーワードは、複数の問合せを使用するレポートでのみ使用します。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。
- Report トリガーの実行順に関して、データ・フェッチが発生する前に 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 レポートを実行している場合に使用できます。

## A.4.83 RECURSIVE\_LOAD

表 A-94 では、RECURSIVE\_LOAD キーワードを使用できるコマンドを示します。

**表 A-94 RECURSIVE\_LOAD キーワードを使用できるコマンド**

rwclient	rwrund	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有効です。

**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 .pl1 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.4.84 REPLACEITEM

表 A-95 では、REPLACEITEM キーワードを使用できるコマンドを示します。

**表 A-95 REPLACEITEM キーワードを使用できるコマンド**

rwclient	rwrund	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
○	○	×	×	○	○	×

**説明** REPLACEITEM を使用して、Oracle9iAS Portal リリース 1 に配信される現行のレポート出力による、同じ出力先に格納されている同じアイテムの旧バージョンの置換を指定できます。

**構文** REPLACEITEM={YES|NO}

### 値

- YES の場合、旧レポート出力は置換されます。
- NO の場合、旧バージョンは置換されません。この場合、新しい出力へのリンクが、同じレポートの旧バージョンへのリンクのリストに追加されます。

**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オプションです。
- 関連するキーワードは、CONTENTAREA\*、EXPIREDAYS、ITEMTITLE、OUTPUTFOLDER\*、OUTPUTPAGE、PAGEGROUP、REPLACEITEM、SITENAME\*、STATUSFOLDER\*、STATUSPAGE です。

\*Oracle9iAS Portal リリース 1 または Oracle WebDB リリース 2.2 (あるいはその両方) 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.4.85 REPLYTO

表 A-96 では、REPLYTO キーワードを使用できるコマンドを示します。

**表 A-96 REPLYTO キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	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.4.86 REPORT|MODULE

「**MODULE|REPORT**」を参照してください。

## A.4.87 ROLE

表 A-97 では、ROLE キーワードを使用できるコマンドを示します。

**表 A-97 ROLE キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	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 レポートを実行している場合に使用できます。

## A.4.88 RUNDEBUG

表 A-98 では、`RUNDEBUG` キーワードを使用できるコマンドを示します。

**表 A-98 RUNDEBUG キーワードを使用できるコマンド**

<code>rwclient</code>	<code>rwrn</code>	<code>rwbuilder</code>	<code>rwconverter</code>	<code>rwservlet</code>	<code>rwcgi</code>	<code>rwserver</code>
○	○	○	×	○	○	×

**説明** `RUNDEBUG` 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。`RUNDEBUG` 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。`RUNDEBUG` 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。

`RUNDEBUG` では、次の内容がチェックされます。

- 枠または繰り返し枠で、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** `RUNDEBUG={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** `RUNDEBUG` 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使用できます。

## A.4.89 SAVE\_RDF

表 A-99 では、SAVE\_RDF キーワードを使用できるコマンドを示します。

**表 A-99 SAVE\_RDF キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	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

*filename.rdf* には、有効なファイル名を指定します。

**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.4.90 SCHEDULE

表 A-100 では、SCHEDULE キーワードを使用できるコマンドを示します。

**表 A-100 SCHEDULE キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	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-101 で説明します。

**表 A-101 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 { <i>n</i> }+}
LEN	{ <i>n</i> }+ {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 { <i>d</i>   <i>dd</i> } [,year]}
MONTHS	jan   feb   mar   apr   may   jun   jul   aug   sep   oct   nov   dec
EXPIRES	on {today   tomorrow   {MONTHS { <i>d</i>   <i>dd</i> } [,year]}}   after <i>n</i>

**デフォルト** なし

## A.4.91 SERVER

表 A-102 では、SERVER キーワードを使用できるコマンドを示します。

**表 A-102 SERVER キーワードを使用できるコマンド**

rwclient	rwrwn	rwbuilder	rwconverter	rwservlet	rwcgi	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=*servername*

**値** サーバー名、または TNS サービス・エントリ名 (*6i* 互換サーバーを使用している場合)

**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、要求がデフォルト・サーバーを使用して処理されます。あるいは、SERVER キーワードを含めるとデフォルトが上書きされます。*rwsvlet* 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 (*rwsvlet.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 キーワードを含めると、デフォルトが無効になります。
- SERVER は、コマンドラインから JSP ベースの Web レポートを実行している場合に使用できます。

## A.4.92 SHOWAUTH

表 A-103 では、SHOWAUTH キーワードを使用できるコマンドを示します。

**表 A-103 SHOWAUTH キーワードを使用できるコマンド**

<i>rwclient</i>	<i>rwrn</i>	<i>rwbuilder</i>	<i>rwconverter</i>	<i>rwsvlet</i>	<i>rwsgi</i>	<i>rwserver</i>
×	×	×	×	○	○	×

**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://yourwebserverserver/reports/rwsvlet/showauth[?] [server=servername] [&authhid=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.4.93 SHOWENV

表 A-104 では、SHOWENV キーワードを使用できるコマンドを示します。

表 A-104 SHOWENV キーワードを使用できるコマンド

rwclient	rwrund	rwbuilder	rwconverter	rwervlet	rwsgi	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://yourwebserver/reports/rwervlet/showenv[?] [server=servername] [&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.4.94 SHOWJOBID

表 A-105 では、SHOWJOBID キーワードを使用できるコマンドを示します。

表 A-105 SHOWJOBID キーワードを使用できるコマンド

rwclient	rwrund	rwbuilder	rwconverter	rwervlet	rwsgi	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://yourwebserver/reports/rwservlet/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、rwservlet に対してのみ使用でき、rwcgi には使用できません。

## A.4.95 SHOWJOBS

表 A-106 では、SHOWJOBS キーワードを使用できるコマンドを示します。

**表 A-106 SHOWJOBS キーワードを使用できるコマンド**

rwclient	rwrund	rwbuilder	rwconverter	rwservlet	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://yourwebserver/reports/rwservlet/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。
- Reports Server の名前が、環境変数またはサ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 要求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.4.96 SHOWMAP

表 A-107 では、SHOWMAP キーワードを使用できるコマンドを示します。

表 A-107 SHOWMAP キーワードを使用できるコマンド

rwclient	rwruntime	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://yourwebserver/reports/rwervlet/showmap[?] [server=servername] [&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.4.97 SHOWMYJOBS

表 A-108 では、SHOWMYJOBS キーワードを使用できるコマンドを示します。

**表 A-108 SHOWMYJOBS キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	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://yourwebservice/reports/rwservlet/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、rwservlet に対してのみ使用でき、rwcgi には使用できません。

## A.4.98 SHUTDOWN

表 A-109 では、SHUTDOWN キーワードを使用できるコマンドを示します。

**表 A-109 SHUTDOWN キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
×	×	×	×	×	×	○



**説明** SHUTDOWN を使用して、すでに実行中のサーバーを停止できます。また、[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。

**構文** SHUTDOWN={NORMAL|IMMEDIATE}

**値** NORMAL または IMMEDIATE

**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.4.99 SITENAME

表 A-110 では、SITENAME キーワードを使用できるコマンドを示します。

**表 A-110 SITENAME キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
○	○	×	×	○	○	×

**説明** SITENAME を使用して、レポート出力の配信先となる WebDB サイト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。[CONTENTAREA](#) を参照してください。OracleAS Portal リリース 2 以降では [PAGEGROUP](#) を使用してください。

**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 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。

- 関連するキーワードは、[CONTENTAREA\\*](#)、[EXPIREDAYS](#)、[ITEMTITLE](#)、[OUTPUTFOLDER\\*](#)、[OUTPUTPAGE](#)、[PAGEGROUP](#)、[REPLACEITEM](#)、[SITENAME\\*](#)、[STATUSFOLDER\\*](#)、[STATUSPAGE](#) です。

\*Oracle9iAS Portal リリース 1 または Oracle WebDB リリース 2.2 (あるいはその両方) 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.4.100 SOURCE

表 A-111 では、SOURCE キーワードを使用できるコマンドを示します。

**表 A-111 SOURCE キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	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

- 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)
- 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。
- 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ユーザー ID が付いていない場合、ユーザー ID は現行のユーザーになります。
- 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指定しなければなりません。

## A.4.101 SQLTRACE

表 A-112 では、SQLTRACE キーワードを使用できるコマンドを示します。

**表 A-112 SQL\_TRACE キーワードを使用できるコマンド**

rwclient	rwrn	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.4.102 SSOCONN

表 A-113 では、SSOCONN キーワードを使用できるコマンドを示します。

**表 A-113 SSOCONN キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
×	×	×	×	○	×	×

**説明** SSOCONN を使用して、シングル・サインオン環境で 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\_str]] [,key [/type [/conn\_str]]]

次に例を示します。

```
ssoconn=mykey/OracleDB/userid
```

**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 (OID) に格納される接続文字列値を参照します。

- `type` は、レポートのデータソース・タイプを指定する事前定義の文字列です。Oracle Application Server のインストール時に提供されるタイプには、OracleDB、JDBC PDS (Java Database Connectivity Pluggable Data Source) および Express PDS (Oracle Express Pluggable Data Source) があります。
- `conn_str` は、レポート実行のために `rwervlet` によって構成される接続文字列パラメータの名前です。

**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。

```
ssoconn=key1/type1/conn_str,key2/type2/conn_str2,key3/type3/conn_str3
```

- SSOCONN オプションの簡易バージョンもあります。表 A-114 に例を示します。
- 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。
  - 同じコマンドラインでの `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` 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。

**表 A-114 SSOCONN オプションの簡易バージョン**

オプション	説明
<code>ssoconn=mkey</code>	キー名のみ指定すると、デフォルトのタイプ (Oracle DB) が使用され、接続パラメータは <code>USERID</code> キーワードになります。
<code>ssoconn=mkey/PDSApp</code>	キー名とアプリケーション・タイプの両方を指定すると、接続パラメータは <code>USERID</code> キーワードになります。
<code>ssoconn=mkey/PDSApp/P_1</code>	すべてを指定すると、指定された値が使用されます。

- 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 として実行しているレポートに SSOCONN キーワードを使用している場合、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 10 章「OracleAS Single Sign-On の構成と管理」を参照してください。

## A.4.103 STATUSFOLDER

表 A-115 では、STATUSFOLDER キーワードを使用できるコマンドを示します。

**表 A-115 STATUSFOLDER キーワードを使用できるコマンド**

rwclient	rwrwn	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
○	○	×	×	○	○	×

**説明** STATUSFOLDER を使用して、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 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 リリース 2 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、REPLACEITEM、SITENAME\*、STATUSFOLDER\*、STATUSPAGE です。

\*Oracle9iAS Portal リリース 1 または Oracle WebDB リリース 2.2（あるいはその両方）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.4.104 STATUSFORMAT

表 A-116 では、STATUSFORMAT キーワードを使用できるコマンドを示します。

表 A-116 STATUSFORMAT キーワードを使用できるコマンド

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	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/rwservlet/showjobs?server=server_name&statusformat={html|xml|xmldtd}`

### 値

- HTML は、レポートのキュー・ステータスを HTML 形式で出力することを指定します。
- XML は、XML 形式で出力することを指定します。
- XMLDTD は、インライン・データ型定義情報を持つ XML 形式で出力することを指定します。

**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.4.105 STATUSPAGE

表 A-117 では、STATUSPAGE キーワードを使用できるコマンドを示します。

表 A-117 STATUSPAGE キーワードを使用できるコマンド

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	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 Oracle9iAS Portal リリース 1 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、STATUSFOLDER を参照してください。

**構文** STATUSPAGE=*page*

**値**

page には Oracle9iAS Portal リリース 1 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**

- 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、REPLACEITEM、SITENAME\*、STATUSFOLDER\*、STATUSPAGE です。

\*Oracle9iAS Portal リリース 1 または Oracle WebDB リリース 2.2（あるいはその両方）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.4.106 STYPE

表 A-118 では、STYPE キーワードを使用できるコマンドを示します。

**表 A-118 STYPE キーワードを使用できるコマンド**

rwclient	rwrund	rwbuilder	rwconverter	rwservlet	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|RDFFILE|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。
- RDFFILE 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.4.107 SUBJECT

表 A-119 では、SUBJECT キーワードを使用できるコマンドを示します。

**表 A-119 SUBJECT キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	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=*any text*

**値** 任意のテキスト

**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.4.108 SUPPRESSLAYOUT

表 A-120 では、SUPPRESSLAYOUT キーワードを使用できるコマンドを示します。

**表 A-120 SUPPRESSLAYOUT キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	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.4.109 TOLERANCE

表 A-121 では、TOLERANCE キーワードを使用できるコマンドを示します。

**表 A-121 TOLERANCE キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	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使用](#)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。

- $n\{\text{unit}\}$  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。
- $\{\text{Mon DD, YYYY}\} \text{hh:mi:ss am|pm} \{\text{timezone}\}$  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.4.110 TRACEFILE

表 A-122 では、TRACEFILE キーワードを使用できるコマンドを示します。

**表 A-122 TRACEFILE キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
○	○	○	×	×	×	×

**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：** Oracle Reports でのトレース実行方法の詳細は、「[レポート・トレース](#)」を参照してください。

---

**構文** TRACEFILE=*tracefile*

**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**

- ランタイム環境で、または Reports Services コンポーネントを監視するとき、3つのすべての TRACE キーワードを使用できます。ただしサーバー・セキュリティの場合、サーバー環境では TRACEOPTS のみが使用できます。これにより、ファイルが Reports Server のファイル・システムに任意に書き込まれることを防止できます。
- Reports Server のトレースは、サーバー構成ファイル (server\_name.conf) で構成します (「OracleAS Reports Services の構成」を参照)。Reports Servlet のトレースは、サーブレット構成ファイル (rwservlet.properties) で構成します (「OracleAS Reports Services の構成」を参照)。個別ジョブのトレースは、ランタイム・コマンドラインで TRACEOPTS コマンドライン・オプションを使用して指定します。
- 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。
- トレース情報は .rdf ファイルの実行時にのみ生成できます。.rep ファイルの実行中には、トレース情報の書き込みを指定することはできません。
- TRACEFILE とともに TRACEOPTS=TRACE\_LOG または TRACEOPTS=TRACE\_ERR を指定した場合、これらのトレース情報はすべて直前に指定されたファイルに格納されます。たとえば、次のような場合では、指定されたトレース情報はすべて err.log に格納されます。これは、そのファイルが RWRUN コマンドで最後に指定されているためです。

```
RWRUN MODULE=order_entry USERID=scott/tiger
TRACEFILE=trace.log LOGFILE=mylog.log
ERRFILE=err.log
```

**A.4.111 TRACEMODE**

表 A-123 では、TRACEMODE キーワードを使用できるコマンドを示します。

**表 A-123 TRACEMODE キーワードを使用できるコマンド**

rwclient	rwrun	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
○	○	○	×	○	○	×

**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。

**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**

- ランタイム環境で、または Reports Services コンポーネントを監視するとき、3つのすべての TRACE キーワードを使用できます。ただしサーバー・セキュリティの場合、サーバー環境では TRACEOPTS のみを使用できます。これによりファイルがサーバーに書き込まれることを防止できます。
- Reports Server のトレースは、サーバー構成ファイル (server\_name.conf) で構成します。Reports Servlet のトレースは、サーブレット構成ファイル (rwservlet.properties) で構成します。個別ジョブのトレースは、ランタイム・コマンドラインで TRACE コマンドライン引数を使用して指定します。Oracle Reports でのトレース実行方法の詳細は、「レポート・トレース」を参照してください。
- トレース情報は、.rdf ファイルの実行時にのみ生成できます。.rep ファイルの実行中には、トレース情報の書込みを指定できません。

## A.4.112 TRACEOPTS

表 A-124 では、TRACEOPTS キーワードを使用できるコマンドを示します。

**表 A-124 TRACEOPTS キーワードを使用できるコマンド**

rwclient	rwrwn	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
○	○	○	×	○	○	○

**説明** TRACEOPTS は、レポート実行中にトレース・ファイルに記録するトレース情報を指示します。

**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イク・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* 要素の他に *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

- ランタイム環境で、または Reports Services コンポーネントを監視するとき、3つのすべての TRACE キーワードを使用できます。ただしサーバー・セキュリティの場合、サーバー環境では TRACEOPTS のみが使用できます。これによりファイルがサーバーに書き込まれることを防止できます。
- Reports Server のトレースは、サーバー構成ファイル (*server\_name.conf*) で構成します。Reports Servlet のトレースは、サーブレット構成ファイル (*rwervlet.properties*) で構成します。個別ジョブのトレースは、ランタイム・コマンドラインで TRACExxx コマンドライン引数を使用して指定します。Oracle Reports でのトレース実行方法の詳細は、「レポート・トレース」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、TRACE\_APP と TRACE\_PRF を使用する場合は、TRACEOPTS=(TRACE\_APP, TRACE\_PRF) のように指定します。
- トレース情報は .rdf ファイルの実行時にのみ生成できます。.rep ファイルの実行中には、トレース情報の書き込みを指定することはできません。

- TRACEOPTS 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使用できます。

## A.4.113 UNINSTALL

表 A-125 では、UNINSTALL キーワードを使用できるコマンドを示します。

**表 A-125 UNINSTALL キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	rwsgi	rwserver
×	×	×	×	×	×	○

**説明** UNINSTALL を使用して、Microsoft Windows から Reports Server インスタンスを削除できます。このプロセスは、システムから Reports Server サービスを削除します。このキーワードは、UNIX では無視されます。

**構文** UNINSTALL *reports\_server\_name*

**値**

*reports\_server\_name* には、既存の Reports Server インスタンス名を指定します。

**デフォルト** なし

**使用上の注意** INSTALL とともに BATCH=YES を使用すると、通常は削除プロセス時に表示されるプロンプトおよびダイアログが表示されなくなります。

## A.4.114 UPGRADE\_PLSQL

表 A-126 では、UPGRADE\_PLSQL キーワードを使用できるコマンドを示します。

**表 A-126 UPGRADE\_PLSQL キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	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 Developer に必要な最新バージョン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.4.115 URLPARAMETER

表 A-127 では、URLPARAMETER キーワードを使用できるコマンドを示します。

**表 A-127 URLPARAMETER キーワードを使用できるコマンド**

rwclient	rwrund	rwbuilder	rwconverter	rwservlet	rwcgi	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://webserver\_name/pagename.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 構成ファイルで **job** 要素の *jobType* パラメータに *rwurl* が指定されており、URL エンジンが存在する場合に使用できます。

## A.4.116 USERID

表 A-128 では、USERID キーワードを使用できるコマンドを示します。

**表 A-128 USERID キーワードを使用できるコマンド**

rwclient	rwrund	rwbuilder	rwconverter	rwservlet	rwcgi	rwserver
○	○	○	○	○	○	×

**説明** USERID は、シングル・サインオンを使用しない場合にのみ使用します。USERID を使用して、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できます。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、レポート要求から USERID キーワードのパスワード部分を省いてください。レポート要求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 ファイルの使用の詳細は、「[レポート要求の実行](#)」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。

`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 レポートを実行している場合にも使用できます。

## A.4.117 WEBSERVER\_DEBUG

表 A-129 では、WEBSERVER\_DEBUG キーワードを使用できるコマンドを示します。

**表 A-129 WEBSERVER\_DEBUG キーワードを使用できるコマンド**

rwclient	rwrn	rwbuilder	rwconverter	rwservlet	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.4.118 WEBSERVER\_DOCROOT

表 A-130 では、WEBSERVER\_DOCROOT キーワードを使用できるコマンドを示します。

**表 A-130 WEBSERVER\_DOCROOT キーワードを使用できるコマンド**

rwclient	rwrund	rwbuilder	rwconverter	rwservlet	rwsgi	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.4.119 WEBSERVER\_PORT

表 A-131 では、WEBSERVER\_PORT キーワードを使用できるコマンドを示します。

**表 A-131 WEBSERVER\_PORT キーワードを使用できるコマンド**

rwclient	rwrwn	rwbuilder	rwconverter	rwservlet	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。

---

---

## Reports 関連の環境変数

環境変数とは、Oracle Reports をホスティングする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。デフォルトの環境を一部変更する場合は、環境変数の設定を編集できません。

- 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、必ず変更前の 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 (9.0.4) にアップグレードする場合は、その reports.sh ファイルの内容をいくつか変更する必要があります。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 10g (9.0.4) をインストールした後で、10g (9.0.4) とともにインストールされた reports.sh に、保存しておいたファイルをマージします。

---

---

この付録の情報は、Reports Builder オンライン・ヘルプにも記載されています。オンライン・ヘルプには、次の方法でアクセスできます。

- Reports Builder で、「ヘルプ」 → 「ヘルプ・コンテンツ」を選択します。

## B.1 環境変数

この項では、Oracle Reports で設定可能なすべての環境変数を説明します。

- NLS のすべての環境変数の詳細は、OTN-J (Oracle Technology Network Japan) (<http://otn.oracle.co.jp>) の『Oracle Application Server 10g グローバリゼーション・ガイド』を参照してください。
- アスタリスク (\*) の付いた環境変数は、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。

---

---

### 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:¥oracle

### 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、Oracle Reports で「ツール」→「オプション」を選択して指定した設定が保持されます。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:¥oracle`

### 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:¥oracle¥tools¥doc`

### B.1.5 DEVELOPER\_NLS\_LANG

**説明** この環境変数は、Reports Builder 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「NLS と双方向サポートの実装」](#)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「NLS と双方向サポートの実装」](#)を参照してください。

**有効値** 現在有効な値のリストは、『Oracle9i グローバリゼーション・サポート・ガイド』で確認できます。または、OTN-J (Oracle Technology Network Japan) (<http://otn.oracle.co.jp>) から『Oracle Application Server 10g グローバリゼーション・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。
- 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、NLS\_LANG 環境変数を reports.sh に明示的に設定することで、この問題を解決できます。また、新しい環境切替え機能を使用して、Oracle Reports の環境を動的に設定できます。詳細は、「[動的な環境切替え](#)」を参照してください。

**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
```

## 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。[TK90\\_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/guicommon9/tk90/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、「[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:¥oracle`

### 使用上の注意

- Reports Runtime (rwruntime) 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:¥oracle`



## 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。TK90\_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/guicommon9/tk90/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、「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 (rwrn) を使用する場合は、ORACLE\_HOME と ORACLE\_PATH を合せた長さが 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。TK90\_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 TFM ファイルが検索されます。

`ORACLE_HOME/guicommon9/tk90/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、「[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。TK90\_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**

- TK90\_TFM または ORACLE\_TFM のいずれの値も指定しない場合、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/guicommon9/tk90/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、「[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:¥oracle¥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： [ScreenPrinter](#)

PostScript プリンタ・ドライバの詳細は、「`screenprinter.ppd`」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、「[UNIX での Oracle Reports による印刷](#)」を参照してください。

## B.1.26 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、プリンタに依存しないレポートを容易に設計できます。**Oracle Reports** は、実際の用紙の左上隅を起点としてレポート出力を生成するよう変更されています。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。

---

---

2. 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。

3. 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.1.27 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.1.28 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|UNICODE

ここでは

ORACLE の場合、Oracle Reports は Oracle BiDi アルゴリズムに従います。Oracle アルゴリズムは、Oracle Application Server Forms Services で使用されます。

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://unicode.org/unicode/reports/tr9/index.html> を参照してください。

**デフォルト** ORACLE

## B.1.29 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。

**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。

---

---

**例** `REPORTS_CGIDIAGBODYTAGS="bgcolor="#CC3366"`

### B.1.30 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.1.31 REPORTS\_CGIHELP \*

**説明** この環境変数は、`rwcgi` が次に示す空の要求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。

`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.1.32 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、`rwsgi` 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

**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:¥oracle¥reports¥conf¥cgicmd.dat`

### B.1.33 REPORTS\_CGINODIAG \*

**説明** この環境変数は、`help` や `showmap` などの、`rwsgi` 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。

**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なくなります。

http://your\_webserver/rwcgi/help?

## B.1.34 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、プラグガブル・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.1.35 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 Reports6i 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、レポート要求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。1 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、ユーザーは要求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 `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。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。Reports Servlet でジョブ要求が受信されると、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` という名前です。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.1.36 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。

**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.1.37 REPORTS\_DEFAULT\_DISPLAY

**説明** この環境変数は、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 環境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用の **ScreenPrinter** (screenprinter.ppd) (UNIX のみ)。
- **拡張イメージングのサポート** (全プラットフォーム共通)。

**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、PDF 出力の GIF 形式と、HTML、HTMLCSS および RTF 出力の BMP 形式に制限されます。
- 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、「[UNIX での Oracle Reports による印刷](#)」を参照してください。
- UNIX では、REPORTS\_DEFAULT\_DISPLAY = YES は、DISPLAY 環境変数で設定された値よりも優先します。
- DISPLAY への依存性を復元し、画面フォントを使用するには、次の処理を実行します。
  - 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。

## B.1.38 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、オペレーティング・システムのデフォルトのピクセル・サイズを無効にするピクセル・サイズを指定します。通常、Oracle Reports ではオペレーティング・システムのピクセル・サイズが使用されます。以前のレポートで作業しており、そのレポートにオペレーティング・システムと異なるピクセル・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、「UNIX での Oracle Reports による印刷」に記載されるとおり、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.1.39 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** `reports9.0`

**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=reports9.0`

## B.1.40 REPORTS\_GRAPH\_IMAGE\_DPI

**説明** 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。

---

---

**注意:** 値が 250 の場合

- 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 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「メモリーが足りません」が表示されないようにする必要があります。
- 元の設定に戻すには、この環境変数の値を 72 DPI に設定します。

## B.1.41 REPORTS\_JVM\_OPTIONS

**説明** この環境変数は、Reports Builder、Reports Runtime または OracleAS Reports Services 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 256 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「メモリーが足りません」が表示されないように考慮されています。
- サーバー・モードでは、`servername.conf` ファイルの `engine` 属性で JVM オプションを指定することもできます。その場合、サーバー構成ファイルで設定された JVM オプションは、`REPORTS_JVM_OPTIONS` 環境変数より優先します。

## B.1.42 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、Reports の NLS\_LANG と XML IANA 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

**使用上の注意** この環境変数は、通常は必要ありません。現行では、次の Bug 2224413 に対する回避策としてのみ使用されています。

```
NLS:NEED STABLE MAPPING FROM ORACLE'S NLS_CHARACTER_SET TO IANA  
ENCODING
```

**例** WISO-8859-8=ISO-8859-8-1;CSEUCKR=EUC-KR;WINDOWS-949=EUC-KR;EUC-CN=GBK;WINDOWS-936=GBK

## B.1.43 REPORTS\_NO\_DUMMY\_PRINTER

**説明** この環境変数は、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 (9.0.4) からは、`uiprint.txt` ファイルに有効なエントリが含まれておらず（つまり、有効なプリンタが定義されていない）、`REPORTS_NO_DUMMY_PRINTER` が設定されている場合、Oracle Reports では、`uiscreenprint.txt` に指定されている `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： [ScreenPrinter](#)

PostScript プリンタ・ドライバの詳細は、`screenprinter.ppd` 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、「[UNIX での Oracle Reports による印刷](#)」を参照してください。

## B.1.44 REPORTS\_NO\_HTML\_SPACE\_REPLACE

### 説明

この環境変数は、HTML または HTMLCSS の出力時に、スペースを `&nbsp;` で置き換えないように指定します。

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、`<` を `&lt;` に、`"` を `&quot;` に自動的にマッピングします。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 `&nbsp;` 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。

```
[&nbsp;&nbsp;&nbsp;Typical&nbsp;&nbsp;&nbsp;data&nbsp;&nbsp;&nbsp;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スペース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.1.45 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 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、「[拡張イメージングのサポート](#)」を参照してください。



## B.1.46 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、外部ボイラプレートなど）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、使用しているオペレーティング・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。

```
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。

**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、ここに設定した値は sourceDir 値で上書きされます。

### 例

```
REPORTS_PATH=C:¥oracle¥apps¥reports;C:¥myfiles
```

## B.1.47 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 NLS 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.1.48 REPORTS\_SERVER \*

**説明** この環境変数は、Web Cartridge または Web CGI 要求に対するデフォルトの Reports Server を指定します。このパラメータが設定されているときは、レポート要求で `SERVER` コマンドライン・キーワードを指定せず、デフォルト・サーバーを使用して要求を処理できます。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を無効にすることもできます。

**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_SERVER=my_rep_server`

## B.1.49 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.1.50 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.1.51 REPORTS\_SYS\_AUTH**

**説明** この環境変数は、制限付き Reports Server に対してレポート要求を実行する場合に、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せん)

**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.1.52 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 Reports のどちらの Reports 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.1.53 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.1.54 REPORTS\_USEREXIT

**説明** この環境変数は、Reports Builder と、その関連する実行可能ファイル (Reports Runtime など) が使用する DLL を指定します。この DLL は、ユーザーが作成したプログラム・モジュールで、Reports Builder 実行可能ファイルによってコールされます。

REPORTS\_USEREXIT では、複数の DLL を指定できます。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。

この値が明示的に設定されない場合、Reports Builder では、システムのパス変数に従って `rwxtb90.dll` が検索されます。

**有効値** 任意のユーザー・イグジット Dynamic Link Library (絶対パス)

**デフォルト** 定義されていません

**例** `REPORTS_USEREXIT=C:¥mydll.dll;d:¥mynew.dll;e:¥bin¥speed.dll`

## B.1.55 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.1.56 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 では、`TK_PRINT` は廃止されました。同じ結果を得るには、次の印刷スクリプト・ファイルを使用します。`ORACLE_HOME/bin/rwlpr.sh`。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、それに合わ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`

**UnixWare:** `lp -s -Tpostscript -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、「[UNIX での Oracle Reports による印刷](#)」を参照してください。

## B.1.57 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：** [ScreenPrinter](#)

PostScript プリンタ・ドライバの詳細は、「`screenprinter.ppd`」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、「[UNIX での Oracle Reports による印刷](#)」を参照してください。

## B.1.58 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:** [ScreenPrinter](#)

PostScript プリンタ・ドライバの詳細は、「[screenprinter.ppd](#)」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、「[UNIX での Oracle Reports による印刷](#)」を参照してください。

## B.1.59 TK90\_AFM

**説明** この環境変数は、AFM ファイルの位置を指定します。TK90\_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/guicommon9/tk90/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、「[UNIX での Oracle Reports による印刷](#)」を参照してください。

## B.1.60 TK90\_HPD

**説明** この環境変数は、HPD ファイルの位置を指定します。TK90\_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/guicommon9/tk90/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、「[UNIX での Oracle Reports による印刷](#)」を参照してください。



## B.1.61 TK90\_PPD

**説明** この環境変数は、PPD ファイルの位置を指定します。TK90\_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/guicommon9/tk90/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、「[UNIX での Oracle Reports による印刷](#)」を参照してください。

## B.1.62 TK90\_TFM

**説明** この環境変数は、TFM ファイルの位置を指定します。TK90\_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

- TK90\_TFM または ORACLE\_TFM のいずれの値も指定しない場合、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/guicommon9/tk90/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、「[UNIX での Oracle Reports による印刷](#)」を参照してください。

## B.1.63 USERNAME

**説明** この環境変数は、デフォルトのログイン・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 (OPS\$ 接頭辞なし)

**デフォルト** 定義されていません

**例** USERNAME=dsanvita

## B.1.64 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「NLS と双方向サポートの実装」](#)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:** [付録 A 「コマンドライン・オプション」](#)

[rwconverter](#) キーワードの詳細に関する記述があります。

---

---

**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](#) オプションを指定します。  
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オプションに加え、[P\\_SERVERS](#)、[P\\_PRIVILEGE](#)、[P\\_TYPES](#)、[P\\_FORMATS](#) 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="rdf" file="
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_formTemplate="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、[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 パラメータを使用します。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。
4. 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。

1. 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。
2. 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。

3. RW\_ADMINISTRATOR 権限を持つユーザーとして OracleAS Portal にログインします。
4. 「ビルダー」をクリックします。
5. 「管理」タブをクリックします。
6. 「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。
7. 「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いずれかのレポートの P\_NAME を入力します。
8. 「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。
9. 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。
10. パラメータを必要に応じて編集します。
11. 「OK」をクリックします。
12. 「閉じる」をクリックします。
13. スクリプトでバッチ登録したレポートごとにステップ 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、最初のステップ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 11 章「OracleAS Portal でのレポートの配布」](#)を参照してください。



表 C-1 RWWWVREG.REGISTER\_REPORT のパラメータ (続き)

パラメータ	説明
P_FILENAME	<p>登録するレポート定義ファイルの名前。</p> <p>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。</p> <p>次に例を示します。</p> <p>P_FILENAME=&gt;'earnings.rdf'</p>
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> <p>P_DESCRIPTION=&gt;'restricted report'</p>
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> <p>P_PRIVILEGE=&gt;'SCOTT,JABERS,PORTAL90'</p>
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> <p>P_AVAILABILITY=&gt;'production'</p> <p><b>注意:</b> 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、<a href="#">第 11 章「OracleAS Portal でのレポートの配布」</a>を参照してください。</p>

表 C-1 RWWWVREG.REGISTER\_REPORT のパラメータ (続き)

パラメータ	説明
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=&gt;'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=&gt;'HTMLCSS,PDF'</pre>
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=&gt;'sales_printer,acct_printer'</pre> <p><b>注意:</b> 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、<a href="#">第 11 章「OracleAS Portal でのレポートの配布」</a>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=&gt;'public.finance_template'</pre>

表 C-1 RWWWVREG.REGISTER\_REPORT のパラメータ (続き)

パラメータ	説明
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><b>注意:</b> 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=&gt;'(P_LASTNAME, LOV=LASTNAME_LOV) (P_SSN) (COPIES, DEFAULT=1, LOW=1, HIGH=2)'</pre> <p>各項目の内容は次のとおりです。</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=&gt;'Is begin IF UPPER(DESTYPE) = 'PRINTER' AND   EMPNAME = 'SMITH' THEN   RETURN(TRUE); ELSE   RETURN(FALSE); END IF; end;'</pre>

---

# 用語集

## HTML (HyperText Markup Language)

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)

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。

## J2EE (Java 2 Platform, Enterprise Edition)

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)

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 Pages)

JavaServer Pages (JSP) テクノロジは、Sun 社の Java サーブレット・テクノロジの拡張機能で、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。

## 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%

## 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 HTML 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、レポート要求、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。

## PDF (Portable Document Format)

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。

## 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 CGI (rwcgi)

Common Gateway Interface (CGI) または Reports Web Cartridge とも呼ばれる 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できるようにする。

---

---

**注意：** Oracle Reports 10g では、rwcgi の使用は薦められていない (下位互換性を確保するために維持されている)。かわりに、JSP またはサーブレット (rwservlet) を使用することができる。

---

---

## Reports Builder (rwbuilder)

Reports Builder を起動する、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 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)

リモート・サーバーでのレポートの実行を可能にする、Oracle Reports の実行可能ファイル。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要求を処理する。レポートを Reports Server に送信するには、rwclient 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)

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できる。

## Reports キャッシュ (reports cache)

完了したジョブを格納する 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 (rwservlet) 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にのみ使用され、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)

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、クライアントがアプリケーション・サーバーへの要求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/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、Common Gateway Interface (CGI) スクリプトにかわるプログラムである。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 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 ドキュメントの書式をきわめて柔軟に設定できるようにする、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) は新しいプラグガブル・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ガブル・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領域。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。

## 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  
ALTER\_SESSION, 18-14  
API  
  cache, 3-10  
  destinations, 3-18  
  engine, 3-12  
  notification, 3-20  
  repository, 3-23  
  security, 3-16  
  イベント, 3-19, 17-1 ~ 17-12  
  デバッグ・イベント, 17-7  
  プラグガブル・デスティネーション, 7-2, 15-1, 15-34  
arraysize コマンド・キーワード, 20-21, A-19  
attach 配布要素, 15-11  
  format 属性, 15-12  
  instance 属性, 15-12  
  name 属性, 15-12  
  srcType 属性, 15-12  
authid コマンド・キーワード, 2-6, A-20  
authid パラメータ, イベント, 17-4  
autocommit コマンド・キーワード, A-20  
autostart コマンド・キーワード, 2-3, A-21

## B

background コマンド・キーワード, A-22  
batch コマンド・キーワード, 2-3, 2-5, 2-6, A-22  
bcc コマンド・キーワード, A-23  
bcc 属性, mail, 15-8  
blankpages コマンド・キーワード, A-24  
body 配布要素, 15-10  
  format 属性, 15-11

  instance 属性, 15-11  
  srcType 属性, 15-11  
buffers コマンド・キーワード, A-25

## C

CA\_GPREFS, B-2  
CA\_UPREFS, B-2  
cache, 3-11  
cacheDir, 3-11  
cachelob コマンド・キーワード, A-25  
cacheSize プロパティ, 3-11  
cache 宛先タイプ, 7-5  
cache 構成要素, 3-10, 13-24  
  class 属性, 3-11  
callBackTimeOut 属性, 3-13  
cauprefs.ora, 8-27  
cc コマンド・キーワード, A-26  
cc 属性, mail, 15-8  
cellwrapper コマンド・キーワード, A-27  
CGI, 1-2, 1-5  
  URL 構文, 13-5  
  下位互換性, 3-9  
cgicmd.dat, 3-40, 13-24 ~ 13-27  
  エントリの追加, 13-25  
  使用方法, 13-26  
CID フォント, 4-34  
class, JDBC 構成ファイル, 8-5  
classPath 属性, engine, 3-12  
class 属性  
  cache, 3-11  
  destination, 3-18  
  engine, 3-12  
  jobStatusRepository, 3-23  
  notification, 3-21

- security, 3-16
- cluster 構成要素, 3-28
- cmdfile コマンド・キーワード, A-28
- cmdkey コマンド・キーワード, A-29
- cmdkey パラメータ, イベント, 17-8
- compatible 構成要素, 3-3, 3-9
  - version 属性, 3-9
- confidential 属性, 7-6
- connection, JDBC 構成ファイル, 8-6
- connection 構成要素, 3-26
  - idleTimeOut 属性, 3-27
  - maxConnect 属性, 3-27
- connectString, 8-5
- contentarea コマンド・キーワード, A-30
- Cookie, 10-2
- cookieexpire, rwservlet.properties, 3-47
- copies コマンド・キーワード, A-31
- copies 属性, 20-22
  - printer, 15-19
- Custom Tag Handler, 1-5
- customize キーワード, 16-17
- customize コマンド・キーワード, 16-1, 16-3, 16-21, 16-23, A-32

## D

---

- DAS, 「Delegated Administration Service」を参照
- dateformatmask コマンド・キーワード, A-33
- DB2 ドライバ, 8-9
- dbauth, rwservlet.properties, 3-43
- dbms\_AQadm パッケージ, 17-9
- dbms\_aq.dequeue, 17-12
- delauth コマンド・キーワード, A-33
- Delegated Administration Service, 9-11, 10-2, 11-4
- delimited\_hdr コマンド・キーワード, A-34
- DELIMITED\_LINE\_END, B-3
- delimiter コマンド・キーワード, A-35
- DEQUEUE, 17-9
- desformat コマンド・キーワード, A-36
- desname コマンド・キーワード, A-38
- destinations 配布要素, 15-4
- destination 構成要素, 3-17, 7-5
  - class 属性, 3-18
  - destype 属性, 3-18
- destination コマンド・キーワード, A-40
- destype コマンド・キーワード, A-41
- destype 属性, 3-18, 7-5

- cache, 7-5
- file, 7-5
- mail, 7-5
- oraclePortal, 7-5
- printer, 7-5
- destype 配布要素, 15-19
  - id 属性, 15-21
  - instance 属性, 15-21
  - name 属性, 15-21
- dest コマンド・キーワード, A-39
- DEVELOPER\_NLS\_LANG, 18-7, 18-11, B-3
- diagbodytags, rwservlet.properties, 3-48
- diagheadtags, rwservlet.properties, 3-48
- diagnostic, rwservlet.properties, 3-41
- distribute コマンド・キーワード, A-46
- distribution.dtd, 15-2, 15-34
- distribution.xsl, 15-34
- dist パラメータ, イベント, 17-8
- DOC, B-3
- DTD, 内部, 8-2
- 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-46
- dunit コマンド・キーワード, A-48

## E

---

- encrypted 属性, 3-17, 7-6
- encryptionkey, rwservlet.properties, 3-48
- engineId 属性, 3-20
- enginesresponsetimeout コマンド・キーワード, A-48
- engine 構成要素, 3-11
  - callBackTimeOut 属性, 3-13
  - classPath 属性, 3-12
  - class 属性, 3-12
  - engLife 属性, 3-13
  - id 属性, 3-12
  - initEngine 属性, 3-13
  - maxEngine 属性, 3-13
  - maxIdle 属性, 3-13
  - minEngine 属性, 3-13
- engLife 属性, 3-13
- ENQUEUE, 17-9
- environment 構成要素, 3-32
  - id 属性, 3-33
- envVariable 構成要素
  - envVariable 属性, 3-33
  - name 属性, 3-33

errortemplate, 3-46  
expiration コマンド・キーワード, 1-7, A-49  
expiredays コマンド・キーワード, A-50  
EXPLAIN PLAN  
TKPROF, SQL トレース, 20-7  
express\_server コマンド・キーワード, A-50

## F

---

fileName 属性, 3-30  
file 宛先タイプ, 7-5  
file 配布要素, 15-15  
format 属性, 15-17  
id 属性, 15-17  
instance 属性, 15-17  
name 属性, 15-17  
foreach 配布要素, 15-5  
format 属性  
attach, 15-12  
body, 15-11  
file, 15-17  
formsize コマンド・キーワード, A-53  
from コマンド・キーワード, A-53  
from 属性, mail, 15-9

## G

---

gateway パラメータ, イベント, 17-4  
getjobid コマンド・キーワード, A-54  
getserverinfo コマンド・キーワード, A-54

## H

---

help, rwservlet.properties, 3-48  
help コマンド・キーワード, A-55  
HTML  
診断の出力, 3-48  
デバッグの出力, 3-48  
HTTP Secure Sockets Layer, 1-5  
HTTP Server, 1-5, 9-10  
HTTPS, 1-5

## I

---

identifier 構成要素, 3-30  
idleTimeOut 属性, 3-27  
id 属性, 3-33

destype, 15-21  
engine, 3-12  
file, 15-17  
mail, 15-8  
notification, 3-21  
orbClient, 3-27  
printer, 15-18  
security, 3-16  
image\_url, rwservlet.properties, 3-47  
include 配布要素, 15-13  
src 属性, 15-15  
Informix ドライバ, 8-12  
initEngine 属性, 3-13  
install コマンド・キーワード, A-56, A-108  
instance 属性  
attach, 15-12  
body, 15-11  
destype, 15-21  
file, 15-17  
printer, 15-19  
itemtitle コマンド・キーワード, A-56

## J

---

Java 2 Enterprise Edition, 1-5  
Java Importer, 20-11  
Java サブレット, 1-2  
Java スタアド・プロシージャ, 20-10  
JDBC-ODBC ドライバ, 8-7  
JDBC 構成ファイル, 8-2  
JDBC 問合せ, 8-14  
JDBC プラuggブル・データソース (PDS), 8-1  
JDBC レポート, 8-20  
jobname コマンド・キーワード, A-57  
jobStatusRepository 構成要素, 3-22  
class 属性, 3-23  
jobtype コマンド・キーワード, A-57  
jobType 属性, 3-20  
job 構成要素, 3-19  
engineId 属性, 3-20  
jobType 属性, 3-20  
securityId 属性, 3-20  
JSP, 1-2, 1-5  
NLS, 18-8 ~ 18-11  
URL 構文, 13-3  
VALIDATTAG, 20-21  
イメージ, 3-47

キー・マッピングの使用法, 13-26  
キャラクタ・セットの指定, 18-8 ~ 18-11  
JSP レポート, Web またはペーパーへの配布, 13-9

## K

---

KeyMapFile, 3-41  
killjobid コマンド・キーワード, A-59

## L

---

LDAP, 11-2  
loginTimeout, 8-6  
log 構成要素, 3-22  
    option 属性, 3-22  
longchunk コマンド・キーワード, 20-22, A-59

## M

---

mail 宛先タイプ, 7-5  
mail 配布要素, 15-7  
    bcc 属性, 15-8  
    cc 属性, 15-8  
    from 属性, 15-9  
    id 属性, 15-8  
    organization 属性, 15-9  
    priority 属性, 15-9  
    replyTo 属性, 15-9  
    returnReceipt 属性, 15-9  
    subject 属性, 15-9  
    to 属性, 15-8  
maxConnect 属性, 3-27  
maxEngine 属性, 3-13  
maxIdle 属性, 3-13  
maxQueueSize 属性, 3-29  
Merant ドライバ, 8-7  
MESSAGES, 17-9  
mimetype コマンド・キーワード, A-60  
minEngine 属性, 3-13  
mod\_oc4j, 1-5  
mode コマンド・キーワード, A-61  
module キーワード, 16-15  
module コマンド・キーワード, 16-23, A-61  
MyIdent.AuthID, 17-5  
MyIdent.GatewayURL, 17-5  
MyIdent.JobID, 17-5  
MyIdent.ServerName, 17-5

## N

---

name 属性, 3-32  
    attach, 15-12  
    destype, 15-21  
    envVariable, 3-33  
    file, 15-17  
    printer, 15-19  
NLS, 18-1 ~ 18-15  
    ALTER\_SESSION, 18-14  
    DEVELOPER\_NLS\_LANG, 18-7  
    JSP, 18-8 ~ 18-11  
    NLS\_LANG, 18-3, 18-12  
    Unicode, 18-12 ~ 18-14  
    USER\_NLS\_LANG, 18-7  
    アーキテクチャ, 18-2  
    アプリケーションの翻訳, 18-14  
    環境変数, 18-2 ~ 18-8  
        NLS\_LANG, 18-12  
        キャラクタ・セット, 18-5  
        言語依存データ, 18-2  
        言語に依存しない機能, 18-2  
        フォントのサポート, 18-13  
NLS\_CALENDAR, B-4  
NLS\_CREDIT, B-4  
NLS\_CURRENCY, B-4  
NLS\_DATE\_FORMAT, B-4  
NLS\_DATE\_LANGUAGE, B-4  
NLS\_DEBIT, B-4  
NLS\_ISO\_CURRENCY, B-4  
NLS\_LANG, 18-3, 18-12, 18-13, B-4  
    charset, 18-4  
    language, 18-3  
    territory, 18-3  
    言語の表記規則, 18-6  
    構文, 18-3  
    地域の表記規則, 18-6  
    定義, 18-5  
NLS\_LANG 環境変数, 18-12, 18-13  
NLS\_LIST\_SEPARATOR, B-5  
NLS\_MONETARY\_CHARACTERS, B-5  
NLS\_NUMERIC\_CHARACTERS, B-6  
NLS\_SORT, B-6  
nonblocksq コマンド・キーワード, A-62  
notification 構成要素, 3-20  
    class 属性, 3-21  
    id 属性, 3-21

notifyfailure コマンド・キーワード, A-62  
notifysuccess コマンド・キーワード, A-63  
numberformatmask コマンド・キーワード, A-64

## O

---

OC4J, 9-10  
OEM, 19-1 ~ 19-26

- Reports Server の起動, 2-2, 19-7
- Reports Server の再起動, 2-2, 19-7
- Reports Server の再構成, 19-23
- Reports Server の停止, 2-2, 19-7
- Reports Services へのナビゲート, 19-2
- 起動, 19-2
  - ジョブ・キューの管理, 19-8
  - ジョブ・キューの表示, 19-8
  - パフォーマンスの監視, 19-16

OID, 「Oracle Internet Directory」を参照

olap\_con コマンド・キーワード, A-65

onfailure コマンド・キーワード, A-65

onsuccess コマンド・キーワード, A-66

OPMN (Oracle Process Manager and Notification Server), 2-2

option 属性, 3-22

ORA\_PROF

- PL/SQL, 20-9

Oracle Advanced Queuing, 17-2, 17-9, 17-11

- dbms\_AQadm パッケージ, 17-9
- dbms\_aq.dequeue, 17-12
- DEQUEUE, 17-9
- ENQUEUE, 17-9
- MESSAGES, 17-9

Oracle Containers for J2EE, 1-5

Oracle Containers for Java 2 Enterprise Edition, 9-10

Oracle Enterprise Manager, 「OEM」を参照

Oracle HTTP Server, 1-5, 2-1, 9-10

Oracle Internet Directory, 9-11, 11-2

Oracle JDBC OCI (Thick) ドライバ, 8-7

Oracle JDBC Thin ドライバ, 8-7

Oracle Login Server, 9-11

Oracle Single Sign-On, 9-13

Oracle Technology Network Japan, 11-1

ORACLE\_AFM, B-6

ORACLE\_HOME, 3-1, 20-4, 20-5, B-6

ORACLE\_HPDP, B-7

ORACLE\_PATH, B-7

ORACLE\_PPD, B-7

ORACLE\_TFM, B-8

OracleAS Portal, 11-1 ~ 11-23

- destination 要素, 例, 7-7
- RW\_ADMINISTRATOR, 11-3
- RW\_BASIC\_USER, 11-3
- RW\_DEVELOPER, 11-3
- RW\_POWER\_USER, 11-3
- 概要, 9-12
  - サーバー・アクセス, 11-13
  - 使用可能日カレンダー結合, 11-8
  - シンプル, 11-6
- プリンタ・アクセス, 11-10
- ユーザーとグループ, 11-2
- ランタイム・パラメータ・フォーム, 11-21
- レポート・アクセス, 11-16
- レポート・ポートレットの公開, 13-16
- レポート要求, 13-6

OracleAS Portal へのレポートのバッチ登録, C-1

OracleAS Reports Services

- Single Sign-On, 9-13

oraclePortal 宛先タイプ, 7-5

ORAINFONAV\_DOCPATH, B-8

orbClient 構成要素, 3-27

- id 属性, 3-27
- publicKeyFile 属性, 3-27

ORDER BY

- ブレイク・グループ, 20-14

organization 属性, mail, 15-9

orientation コマンド・キーワード, A-67

OTN-J, 「Oracle Technology Network Japan」を参照

outputfolder コマンド・キーワード, A-68

outputimageformat コマンド・キーワード, A-69

outputpage コマンド・キーワード, A-70

overwrite コマンド・キーワード, A-71

## P

---

p\_availability コマンド・キーワード, A-71

p\_description コマンド・キーワード, A-72

p\_formats コマンド・キーワード, A-73

p\_name コマンド・キーワード, A-73

p\_owner コマンド・キーワード, A-74

p\_pformtemplate コマンド・キーワード, A-74

p\_printers コマンド・キーワード, A-75

p\_privilege コマンド・キーワード, A-75

p\_servers コマンド・キーワード, A-76

p\_trigger コマンド・キーワード, A-76  
 p\_types コマンド・キーワード, A-77  
 pagegroup コマンド・キーワード, A-77  
 pagesize コマンド・キーワード, A-78  
 pagestream コマンド・キーワード, A-79  
 paramform コマンド・キーワード, 20-22, A-80  
 parsequery コマンド・キーワード, A-81  
 PATH, 2-5  
 pdfcomp コマンド・キーワード, A-81  
 pdfembed コマンド・キーワード, A-82  
 PDF 機能  
   アクセシビリティ, 6-14  
   圧縮, 6-2  
   フォント・エイリアシング, 6-3  
   フォントの埋込み, 6-9  
   分類, 6-15  
 persistFile 構成要素, 3-3, 3-29  
   fileName 属性, 3-30  
 PL/SQL, 20-21  
   ALTER\_SESSION, 18-14  
   ORA\_PROF, 20-9  
     例, 20-9  
   SRW\_PARAMETER, 17-2  
   SRW\_PARAMLIST, 17-2, 17-4, 17-9  
   srw\_test.sql, 17-7  
   SRW.ADD\_DEFINITION, 16-17, 16-23  
   SRW.ADD\_PARAMETER, 17-3  
   srwAPIdrop.sql, 17-2  
   srwAPIgrant.sql, 17-2  
   srwAPIins.sql, 17-2  
   SRW.APPLY\_DEFINITION, 16-3, 16-15, 16-17  
   SRW.CANCEL\_REPORT, 17-6  
   SRW.CLEAR\_PARAMETER\_LIST, 17-4  
   SRW.DO\_SQL, 20-10  
     例, 20-10  
   SRW.JOB\_IDENT, 17-4  
   SRW.MAXROW\_UNSET, 20-14  
   SRW.REMOVE\_PARAMETER, 17-3  
   SRW.REPORT\_STATUS, 17-5  
   SRW.SET\_MAXROW(), 20-13  
     例, 20-14  
   SRW.SET\_TEXT\_COLOR, 20-18  
   SRW.START\_DEBUGGING, 17-7  
   SRW.STATUS\_RECORD, 17-5  
   SRW.STOP\_DEBUGGING, 17-7  
   SRW パッケージ, 17-2  
   ストアド・プロシージャ, 20-9  
     ブロックの翻訳, 18-14  
 PL/SQL, 拡張配布, 15-3  
 Pluggable  
   destinations, 7-2  
   cache, 3-10  
   destinations, 3-18  
   engine, 3-12  
   notification, 3-20  
   repository, 3-23  
   security, 3-16  
   イベント, 3-19  
 pluginParam  
   notification とともに使用, 3-21  
 pluginParam 構成要素, 3-31, 15-9  
   destination とともに使用, 3-19  
   name 属性, 3-32  
   type 属性, 3-32  
 Portal, 11-1 ~ 11-23  
   RW\_ADMINISTRATOR, 11-3  
   RW\_BASIC\_USER, 11-3  
   RW\_DEVELOPER, 11-3  
   RW\_POWER\_USER, 11-3  
   概要, 9-12  
   サーバー・アクセス, 11-13  
   使用可能日カレンダー, 結合, 11-8  
   使用可能日カレンダー, シンプル, 11-6  
   プリンタ・アクセス, 11-10  
   ユーザーとグループ, 11-2  
   ランタイム・パラメータ・フォーム, 11-21  
   レポート・アクセス, 11-16  
   レポート要求, 13-6  
 PostScript, 出力先, 2-4  
 prefs.ora, 8-27  
 PRINTER, B-9  
 printer 宛先タイプ, 7-5  
 printer 配布要素, 15-17  
   copies 属性, 15-19  
   id 属性, 15-18  
   instance 属性, 15-19  
   name 属性, 15-19  
 printjob コマンド・キーワード, A-82  
 priority 属性, mail, 15-9  
 property, JDBC 構成ファイル, 8-6  
 property 配布要素, 15-22  
 publicKeyFile 属性, 3-27

## Q

Queue Viewer, 1-2  
Queue Manager, 1-2, 13-22  
queue 構成要素  
    maxQueueSize 属性, 3-29  
queue 要素, 3-28

## R

readonly コマンド・キーワード, A-83, A-84  
RECURSIVE\_LOAD コマンド・キーワード, 20-21  
reload\_keymap, rwservlet.properties, 3-41  
replaceitem コマンド・キーワード, A-85  
replyto コマンド・キーワード, A-86  
replyTo 属性, mail, 15-9  
report\_server\_name.conf, 3-2, 20-4, 20-5  
Reports  
    URL 構文, 13-2  
    現行ジョブ・キュー, 19-8  
    失敗したジョブ・キュー, 19-14  
    終了したジョブ・キュー, 19-11  
    処理, 1-6  
    スケジュールされたジョブ・キュー, 19-10  
    パッケージ・プロシージャを使用した要求, 13-7  
Reports CGI, 1-5  
Reports Engine, 1-5, 1-7  
Reports Engine, クラスタ化のアーキテクチャ, ロード・バランシング, 12-2  
Reports JSP, 1-5  
Reports Queue Manager, 1-2, 13-22  
Reports Queue Viewer, 1-2  
Reports Server  
    OEM による起動, 2-2, 19-7  
    OEM による再起動, 2-2, 19-7  
    OEM による再構成, 19-23  
    OEM による停止, 2-2, 19-7  
    アクセス制御, 11-13  
    宛先処理, 7-3  
    宛先タイプの登録, 7-4  
    永続性, 3-29, 13-24  
    クラスタ, 12-1 ~ 12-6  
    構成ファイル, 1-7, 3-2  
    コマンドラインからの起動, 2-5  
    サービスとしての起動, 2-3  
    サブレットとしての起動, 2-5  
    再起動, 12-6

ステータス・レコード, 17-6  
名前の変更, 12-3  
パフォーマンスの監視, 19-16

### Reports Services

cache API, 3-10  
Destinations API, 3-18, 7-2, 15-1  
engine API, 3-12  
events API, 3-19  
notification API, 3-20  
repository API, 3-23  
security API, 3-16  
Single Sign-On, 9-13  
アーキテクチャ, 1-1  
永続性, 1-2  
概要, 1-2  
監視, 19-1 ~ 19-26  
管理, 19-1 ~ 19-26  
起動と停止, 2-1  
コンポーネント, 1-4

### Reports Servlet, 1-5, 3-3, 9-10

URL 構文, 13-2

REPORTS\_ADD\_HWMARGIN, B-9  
REPORTS\_ARABIC\_NUMERAL, 6-22, B-10  
REPORTS\_BIDI\_ALGORITHM, B-11  
REPORTS\_CGIDIAGBODYTAGS, B-11  
REPORTS\_CGIDIAGHEADTAGS, B-12  
REPORTS\_CGIHELP, B-12  
REPORTS\_CGIMAP, B-13  
REPORTS\_CGINODIAG, B-13  
REPORTS\_CLASSPATH, B-14  
REPORTS\_COOKIE\_EXPIRE, B-15  
REPORTS\_DB\_AUTH, B-16  
REPORTS\_DEFAULT\_DISPLAY, B-16, B-22  
REPORTS\_DEFAULT\_PIXEL\_SIZE, B-17  
REPORTS\_ENCRYPTION\_KEY, B-18  
REPORTS\_GRAPH\_IMAGE\_DPI, B-18  
REPORTS\_JVM\_OPTIONS, B-19  
REPORTS-NLS\_XML\_CHARSETS, B-20  
REPORTS\_NO\_DUMMY\_PRINTER, B-20  
REPORTS\_NO\_HTML\_SPACE\_REPLACE, B-21  
REPORTS\_PATH, B-23  
REPORTS\_RESOURCE, B-23  
REPORTS\_SERVER, B-24  
REPORTS\_SPACE\_BREAK, B-24  
REPORTS\_SSLPORT, B-25  
REPORTS\_SYS\_AUTH, B-25  
REPORTS\_TAGLIB\_URI, B-26

- REPORTS\_TMP, B-26
- REPORTS\_USEREXIT, B-27
- report キーワード, 16-15
- report コマンド・キーワード, 16-23, A-61
- report パラメータ, イベント, 17-4, 17-8
- returnReceipt 属性, mail, 15-9
- role コマンド・キーワード, A-86
- rundebug コマンド・キーワード, 20-21, A-87
- RW\_ADMINISTRATOR, 11-3
- RW\_BASIC\_USER, 11-3
- RW\_DEVELOPER, 11-3
- RW\_POWER\_USER, 11-3
- rwbuilder コマンド, 16-23, A-10
- rwcgi コマンド, A-15
- rwclient, 13-6
- rwclient コマンド, 16-16, 16-17, 16-20, A-6
- rwconverter
  - バッチ登録用 SQL スクリプトの生成, C-2
- rwconverter コマンド, 16-16, 16-21, A-10
- rwproxy, 3-9
- RWRUN, 20-5
  - RECURSIVE\_LOAD, 20-21
- rwruntime.jar, 12-5
- rwruntime コマンド, 16-16, 16-17, 16-20, 16-22, A-8
- rwserver, 2-6
  - install, 2-3
  - server, 2-5
- rwserverconf.dtd, 3-2, 3-4, 7-4, 7-6
  - cache 要素, 3-10
  - cluster 要素, 3-28
  - compatible 要素, 3-9
  - connection 要素, 3-26
  - destination 要素, 3-17
  - engine 要素, 3-11
  - environment 要素, 3-32
  - identifier 要素, 3-30
  - jobStatusRepository 要素, 3-22
  - job 要素, 3-19
  - log 要素, 3-22
  - notification 要素, 3-20
  - orbClient 要素, 3-27
  - persistFile 要素, 3-29
  - pluginParam 要素, 3-31
  - queue 要素, 3-28
  - security 要素, 3-15
  - server 要素, 3-8
  - trace 要素, 3-24

- rwserver.template, 3-3
- rwserver コマンド, A-18
- RWSERVLET, 20-21
- rwervlet, 1-5, 2-5
  - Web サービスとして公開, 13-22
- rwervlet.properties, 3-3
  - cookieexpire, 3-47
  - dbauth, 3-43
  - diagbodytags, 3-48
  - diagheadtags, 3-48
  - diagnostic, 3-41
  - encryptionkey, 3-48
  - errortemplate, 3-46
  - help, 3-48
  - image\_url, 3-47
  - KeyMapFile, 3-41
  - reload\_keymap, 3-41
  - server, 3-46
  - server\_in\_process, 3-46
  - Single Sign-On, 3-49
  - sysauth, 3-43
  - tracefile, 3-44
  - tracemode, 3-44
  - traceopts, 3-43
- rwervlet コマンド, A-12
- RW 環境変数, B-27

## S

---

- save\_rdf コマンド・キーワード, A-88
- schedule コマンド・キーワード, 13-22, A-88
- Secure Sockets Layer, 1-5
- securityId 属性, 3-20
- security 構成要素, 3-15
  - class 属性, 3-16
  - id 属性, 3-16
- server, rwervlet.properties, 3-46
- server\_in\_process, 3-46
- server\_name.conf, 3-2
- SERVEROUT, 17-7
- server 構成要素, 3-8
- server コマンド・キーワード, 2-5, 2-6, A-89
- server パラメータ, イベント, 17-4
- showenv コマンド・キーワード, A-91
- showjobs コマンド・キーワード, 1-2, 17-5, A-92
- showmap コマンド・キーワード, A-93
- showmyjobs コマンド・キーワード, A-94



shutdown コマンド・キーワード, 2-6, A-94  
Single Sign-On, 9-13  
    rwservlet.properties, 3-49  
    機能, 9-13  
sitename コマンド・キーワード, A-95  
SMTP, 3-21, 15-8, 15-9  
sourceDatabase, 8-5  
source コマンド・キーワード, A-96  
SQL Server ドライバ, 8-10  
SQL\*Plus, 17-7  
SQL トレース  
    TKPROF, EXPLAIN PLAN, 20-7  
srcType 属性  
    attach, 15-12  
    body, 15-11  
src 属性, include, 15-15  
SRW\_PARAMETER, 17-2  
SRW\_PARAMLIST, 17-2, 17-4, 17-9  
SRW\_ParamList, 17-2  
SRW\_ParamList\_Object, 17-2  
srw\_test.sql, 17-7  
SRW.ADD\_DEFINITION, 16-17, 16-23  
SRW.ADD\_PARAMETER, 17-3  
srwAPIdrop.sql, 17-2  
srwAPIgrant.sql, 17-2  
srwAPIins.sql, 17-2  
SRW.APPLY\_DEFINITION, 16-3, 16-15, 16-17  
SRW.CANCEL\_REPORT, 17-6  
SRW.CLEAR\_PARAMETER\_LIST, 17-4  
SRW.DO\_SQL, 20-10  
SRW.JOB\_IDENT, 17-4  
SRW.REMOVE\_PARAMETER, 17-3  
SRW.REPORT\_STATUS, 17-5  
SRW.RUN\_REPORT, 13-7  
SRW.START\_DEBUGGING, 17-7  
SRW.STATUS\_RECORD, 17-5  
SRW.STOP\_DEBUGGING, 17-7  
SRW パッケージ, 17-2  
SSL, 1-5  
SSO, 3-49  
ssoconn コマンド・キーワード, A-97  
statusfolder コマンド・キーワード, A-99  
statusformat コマンド・キーワード, A-100  
statuspage コマンド・キーワード, A-100  
stype コマンド・キーワード, A-101  
subject コマンド・キーワード, A-102  
subject 属性, mail, 15-9

subProtocol, 8-5  
Sybase ドライバ, 8-8  
sysauth, rwservlet.properties, 3-43

## T

---

TK\_PRINT\_STATUS, B-28  
TK\_PRINTER, B-29  
TK90\_AFM, B-30  
TK90\_HPD, B-30  
TK90\_PPD, B-31  
TK90\_TFM, B-31  
TKPROF  
    EXPLAIN PLAN, SQL トレース, 20-7  
tnsnames.ora, 3-9  
tolerance コマンド・キーワード, 1-7, 13-23, A-103  
to 属性, mail, 15-8  
trace\_all, 3-26  
trace\_app, 3-25  
trace\_brk, 3-25  
trace\_dbg, 3-26  
trace\_dst, 3-26  
trace\_err, 3-26  
trace\_exc, 3-26  
trace\_inf, 3-26  
trace\_log, 3-26  
trace\_pls, 3-25  
trace\_prf, 3-25  
trace\_sql, 3-25  
trace\_sta, 3-26  
trace\_tms, 3-25  
trace\_wrn, 3-26  
tracefile コマンド・キーワード, A-104  
tracefile サブレット・パラメータ, 3-44  
traceFile 属性, 3-24  
tracemode コマンド・キーワード, A-105  
tracemode サブレット・パラメータ, 3-44  
traceMode 属性, 3-25  
traceopts コマンド・キーワード, A-106  
traceopts サブレット・パラメータ, 3-43  
traceOpts 属性, 3-25  
trace 構成要素, 3-24  
    traceFile, 20-4, 20-5  
    traceFile 属性, 3-24  
    traceMode, 20-4  
    traceMode 属性, 3-25  
    traceOpts, 20-4

traceOpts 属性, 3-25  
TrueType Big Font, 18-13  
TrueType フォント, 4-32  
Type1 フォント, 4-32  
type 属性, 3-32

## U

---

uifont.ali, 18-6  
Unicode, 4-30, 18-12 ~ 18-14  
urlparameter コマンド・キーワード, A-109  
URL エンジン  
  構成, 3-49  
  要求の送信, 13-21  
  要素, 3-49  
URL ジョブ要求, 13-1 ~ 13-27  
URL, ランタイム構文, 13-2  
USER-NLS-LANG, 18-7, 18-11, B-32  
userid コマンド・キーワード, A-109  
userid パラメータ, イベント, 17-4, 17-8  
USERNAME, B-31  
UTF8, 18-12

## V

---

value 属性, 3-33  
version 属性, 3-9

## W

---

webserver\_debug コマンド・キーワード, A-110  
webserver\_docroot コマンド・キーワード, A-111  
webserver\_port コマンド・キーワード, A-111  
Web サービス, 14-1  
  rwservlet の公開, 13-22  
  WSDL の表示, 14-3  
  操作, 14-6  
Web リスナー, 1-5, 9-10  
Windows のレジストリ, 18-5

## X

---

XML  
  1 つの定義の適用, 16-16  
  distribution.dtd, 15-2, 15-34  
  PL/SQL での定義の適用, 16-17  
  PL/SQL による適用, 16-17

Reports Builder でのオープン, 16-23  
解析, 16-2  
拡張配布, 15-1 ~ 15-32  
カスタマイズの作成, 16-3, 16-3 ~ 16-9  
カスタマイズの適用, 16-3  
カスタマイズのデバッグ, 16-21 ~ 16-24  
カスタマイズのトレース・オプション, 16-22  
グループ階層の作成, 16-11  
クロス積グループの作成, 16-12  
サマリーの作成, 16-13  
式の作成, 16-13  
実行時の適用, 16-16  
書式マスクの変更, 16-5  
書式例外の追加, 16-5  
新規問合せの追加, 16-8  
スタイルの変更, 16-4  
単独での実行, 16-20  
データソースのリンク, 16-10  
データの検証, 20-11  
データ・モデルの作成, 16-9 ~ 16-15  
適用, 16-15 ~ 16-21  
デバッグ, 16-23  
パーサーのエラー・メッセージ, 16-22  
ハイパーリンクの追加, 16-6  
配布 XML ファイルの使用, 15-32  
バッチ変更, 16-21  
パラメータの作成, 16-14  
必須カスタマイズ・タグ, 16-4  
複数の定義の適用, 16-17  
複数のデータソースの作成, 16-9  
プレースホルダの作成, 16-13  
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, 16-12  
レポートのカスタマイズ, 16-1 ~ 16-24  
XML, JDBC 構成ファイル, 8-2  
XML 属性, 変数の使用, 15-3  
XSL, distribution.xsl, 15-34

## あ

---

アーキテクチャ  
  NLS, 18-2  
  Reports Services, 1-1  
  宛先, 7-3  
アクセス制御, 11-5 ~ 11-23  
  サーバー, 11-13  
  使用可能日カレンダー, 結合, 11-8

- 使用可能日カレンダー, シンプル, 11-6
- プリンタ, 11-10
- レポート, 11-16
- 宛先
  - 宛先タイプ, 7-5
  - クラス, 7-5
  - 有効な値, 7-5
- 宛先アーキテクチャ, 7-3
- 宛先タイプ, 3-18, 7-4
- アドバンスト・キューイング, 17-2, 17-9, 17-11
  - dbms\_AQadm パッケージ, 17-9
  - dbms\_aq.dequeue, 17-12
  - DEQUEUE, 17-9
  - ENQUEUE, 17-9
  - MESSAGES, 17-9
- アプリケーションの翻訳, 18-14
- 暗号化, 7-6
- 一括処理, 15-5
  - 配布, 20-20
- イベント
  - authid パラメータ, 17-4
  - cmdkey パラメータ, 17-8
  - dist パラメータ, 17-8
  - gateway パラメータ, 17-4
  - MyIdent.AuthID, 17-5
  - MyIdent.GatewayURL, 17-5
  - MyIdent.JobID, 17-5
  - MyIdent.ServerName, 17-5
  - report パラメータ, 17-4, 17-8
  - server パラメータ, 17-4
  - SRW\_PARAMETER, 17-2
  - SRW\_PARAMLIST, 17-2, 17-4, 17-9
  - SRW\_ParamList, 17-2
  - SRW\_ParamList\_Object, 17-2
  - srw\_test.sql, 17-7
  - SRW.ADD\_PARAMETER, 17-3
  - srwAPIdrop.sql, 17-2
  - srwAPIgrant.sql, 17-2
  - srwAPIins.sql, 17-2
  - SRW.CANCEL\_REPORT, 17-6
  - SRW.CLEAR\_PARAMETER\_LIST, 17-4
  - SRW.JOB\_IDENT, 17-4
  - SRW.REMOVE\_PARAMETER, 17-3
  - SRW.REPORT\_STATUS, 17-5
  - SRW.START\_DEBUGGING, 17-7
  - SRW.STATUS\_RECORD, 17-5
  - SRW.STOP\_DEBUGGING, 17-7

- SRW パッケージ, 17-2
- userid パラメータ, 17-4, 17-8
- エンキュー・プロシージャの作成, 17-10
- ジョブの取消し, 17-6
- デキュー・プロシージャの作成, 17-11
- デバッグ, 17-7
- メッセージ・キューの作成, 17-9
- レポートの起動, 17-7
- イベント・ドリブンの公開, 1-8, 17-1 ~ 17-12
- インプロセス・サーバー, 1-2, 1-5, 3-46
- 永続性, 1-2, 3-29, 13-24
- エラー・メッセージ, 3-41, 3-46
- エラー・メッセージ, XML, 16-22
- エンキュー, プロシージャの作成, 17-10
- 大文字と小文字の区別, 2-8

## か

---

- カスタム・ドライバ, 8-13
- 環境切替え, 3-34
- 環境変数
  - CA\_GPREFS, B-2
  - CA\_UPREFS, B-2
  - DELIMITED\_LINE\_END, B-3
  - DEVELOPER-NLS\_LANG, 18-7, B-3
  - DOC, B-3
  - NLS, 18-13
  - NLS\_CALENDAR, B-4
  - NLS\_CREDIT, B-4
  - NLS\_CURRENCY, B-4
  - NLS\_DATE\_FORMAT, B-4
  - NLS\_DATE\_LANGUAGE, B-4
  - NLS\_DEBIT, B-4
  - NLS\_ISO\_CURRENCY, B-4
  - NLS\_LANG, 18-3, B-4
  - NLS\_LIST\_SEPARATOR, B-5
  - NLS\_MONETARY\_CHARACTERS, B-5
  - NLS\_NUMERIC\_CHARACTERS, B-6
  - NLS\_SORT, B-6
  - ORACLE\_AFM, B-6
  - ORACLE\_HOME, B-6
  - ORACLE\_HPDP, B-7
  - ORACLE\_PATH, B-7
  - ORACLE\_PPD, B-7
  - ORACLE\_TFM, B-8
  - ORAINFONAV\_DOCPPATH, B-8
  - PATH, 2-5

PRINTER, B-9  
 REPORTS\_ADD\_HWMARGIN, B-9  
 REPORTS\_ARABIC\_NUMERAL, 6-22, B-10  
 REPORTS\_BIDI\_ALGORITHM, B-11  
 REPORTS\_CGIDIAGBODYTAGS, B-11  
 REPORTS\_CGIDIAGHEADTAGS, B-12  
 REPORTS\_CGIHELP, B-12  
 REPORTS\_CGIMAP, B-13  
 REPORTS\_CGINODIAG, B-13  
 REPORTS\_CLASSPATH, B-14  
 REPORTS\_COOKIE\_EXPIRE, B-15  
 REPORTS\_DB\_AUTH, B-16  
 REPORTS\_DEFAULT\_DISPLAY, B-16, B-22  
 REPORTS\_DEFAULT\_PIXEL\_SIZE, B-17  
 REPORTS\_ENCRYPTION\_KEY, B-18  
 REPORTS\_GRAPH\_IMAGE\_DPI, B-18  
 REPORTS\_JVM\_OPTIONS, B-19  
 REPORTS-NLS\_XML\_CHARSETS, B-20  
 REPORTS\_NO\_DUMMY\_PRINTER, B-20  
 REPORTS\_NO\_HTML\_SPACE\_REPLACE, B-21  
 REPORTS\_PATH, B-23  
 REPORTS\_RESOURCE, B-23  
 REPORTS\_SERVER, B-24  
 REPORTS\_SPACE\_BREAK, B-24  
 REPORTS\_SSLPORT, B-25  
 REPORTS\_SYS\_AUTH, B-25  
 REPORTS\_TAGLIB\_URI, B-26  
 REPORTS\_TMP, B-26  
 REPORTS\_USEREXIT, B-27  
 RW, B-27  
 TK\_PRINT\_STATUS, B-28  
 TK\_PRINTER, B-29  
 TK90\_AFM, B-30  
 TK90\_HPDP, B-30  
 TK90\_PPD, B-31  
 TK90\_TFM, B-31  
 USER-NLS\_LANG, 18-7, B-32  
 USERNAME, B-31  
 環境変数, NLS, 18-2 ~ 18-8  
 環境変数, 編集, B-1  
 監視, Reports Services, 19-1 ~ 19-26  
 管理, Reports Services, 19-1 ~ 19-26  
 キー・マップ・ファイル, 13-24 ~ 13-27  
   URL パラメータのマッピング, 13-25  
   位置の指定, 3-40  
   エントリの追加, 13-25  
   再ロード, 3-41  
   使用可能にする方法, 13-25  
   使用方法, 13-26  
   パラメータ・フォームを使用した制限付きの実行,  
     13-26  
   利点, 13-24  
 キーワード  
   accessible, A-18  
   arraysize, A-19  
   authid, A-20  
   autocommit, A-20  
   autostart, A-21  
   background, A-22  
   batch, A-22  
   bcc, A-23  
   blankpages, A-24  
   buffers, A-25  
   cachelob, A-25  
   cc, A-26  
   cellwrapper, A-27  
   cmdfile, A-28  
   cmdkey, A-29  
   contentarea, A-30  
   copies, A-31  
   customize, 16-1, 16-3, 16-15, 16-17, 16-21,  
     16-23, A-32  
   dateformatmask, A-33  
   delauth, A-33  
   delimited\_hdr, A-34  
   delimiter, A-35  
   desformat, A-36  
   desname, A-38  
   dest, A-39  
   destination, A-40  
   destype, A-41  
   distribute, A-46  
   dtype, A-46  
   dunit, A-48  
   enginesresponsetimeout, A-48  
   expiration, A-49  
   expiredays, A-50  
   express\_server, A-50  
   formsize, A-53  
   from, A-53  
   getjobid, A-54  
   getserverinfo, A-54  
   help, A-55  
   install, A-56, A-108

itemtitle, A-56  
 jobname, A-57  
 jobtype, A-57  
 killjobid, A-59  
 longchunk, A-59  
 mimetype, A-60  
 mode, A-61  
 module, 16-15, 16-23, A-61  
 nonblocksql, A-62  
 notifyfailure, A-62  
 notifiysuccess, A-63  
 numberformatmask, A-64  
 olap\_con, A-65  
 onfailure, A-65  
 onsuccess, A-66  
 orientation, A-67  
 outputfolder, A-68  
 outputimageformat, A-69  
 outputpage, A-70  
 overwrite, A-71  
 p\_availability, A-71  
 p\_description, A-72  
 p\_formats, A-73  
 p\_name, A-73  
 p\_owner, A-74  
 p\_pformtemplate, A-74  
 p\_printers, A-75  
 p\_privilege, A-75  
 p\_servers, A-76  
 p\_trigger, A-76  
 p\_types, A-77  
 pagegroup, A-77  
 pagesize, A-78  
 pagestream, A-79  
 paramform, A-80  
 parsequery, A-81  
 pdfcomp, A-81  
 pdfembed, A-82  
 printjob, A-82  
 readonly, A-83, A-84  
 replaceitem, A-85  
 replyto, A-86  
 report, 16-15, 16-23, A-61  
 role, A-86  
 rundebug, A-87  
 save\_rdf, A-88  
 schedule, 13-22, A-88  
 server, A-89  
 showenv, A-91  
 showjobs, 17-5, A-92  
 showmap, A-93  
 showmyjobs, A-94  
 shutdown, A-94  
 sitename, A-95  
 source, A-96  
 ssoconn, A-97  
 statusfolder, A-99  
 statusformat, A-100  
 statuspage, A-100  
 stype, A-101  
 subject, A-102  
 tolerance, 13-23, A-103  
 tracefile, A-104  
 tracemode, A-105  
 traceopts, A-106  
 urlparameter, A-109  
 userid, A-109  
 webserver\_debug, A-110  
 webserver\_docroot, A-111  
 webserver\_port, A-111  
 北アフリカ圏の言語, 18-11  
 キャッシュ, 13-23  
 キャッシュ・キー, 13-23  
 キャラクタ・セット, 4-30, 18-1, 18-5, 18-8 ~ 18-11  
   Unicode, 18-12 ~ 18-14  
   UTF8, 18-12  
   設計上の考慮事項, 18-5  
   フォントのエイリアシング, 18-6  
 キュー  
   OEM による管理, 19-8  
   OEM による表示, 19-8  
   現行ジョブ, 19-8  
   失敗したジョブ, 19-14  
   終了したジョブ, 19-11  
   スケジュールされたジョブ, 19-10  
 クラスタ, 3-47, 12-1 ~ 12-6  
   OEM による表示, 19-26  
   OEM によるリンク, 19-26  
   概要, 12-2  
   セットアップ, 12-3  
   重複ジョブの検出, 13-24  
   要求送信, 12-6  
 グループ・フィルタ, 20-15  
   WHERE 句, 20-15

- 計算, 20-12
  - 例, 20-13
- 言語
  - 北アフリカ圏, 18-11
  - 中東圏, 18-11
- 現行ジョブ・キュー, 19-8
- 公開鍵, 12-5
- 構成, 3-1 ~ 3-58
  - OEM を使用, 19-23
  - URL エンジン, 3-49
  - 考慮事項, 1-8
  - プロキシ情報, 3-50
- 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
- 構成要素
  - cache, 3-10, 13-24
  - cluster, 3-28
  - compatible, 3-3, 3-9
  - connection, 3-26
  - destination, 3-17, 7-5
  - engine, 3-11
  - environment, 3-32
  - identifier, 3-30
  - job, 3-19
  - jobStatusRepository, 3-22
  - log, 3-22
  - notification, 3-20
  - orbClient, 3-27
  - persistFile, 3-3, 3-29
  - pluginParam, 3-31, 15-9
  - queue, 3-28
  - security, 3-15
  - server, 3-8
  - trace, 3-24
- 構文, Reports の URL, 13-2
- 構文, コマンド, A-18
- コマンド
  - rwbuilder, 16-23, A-10
  - rwcgi, A-15
  - rwclient, 16-16, 16-17, 16-20, A-6
  - rwconverter, 16-16, 16-21, A-10
  - rwrund, 16-16, 16-17, 16-20, 16-22, A-8
  - rwserver, A-18
  - rwervlet, A-12
    - 概要, A-2
    - 構文, A-18
- コマンド・キーワード
  - accessible, A-18
  - arraysize, 20-21, A-19
  - authid, 2-6, A-20
  - autocommit, A-20
  - autostart, 2-3, A-21
  - background, A-22
  - batch, 2-3, 2-5, 2-6, A-22
  - bcc, A-23
  - blankpages, A-24
  - buffers, A-25
  - cachelob, A-25
  - cc, A-26
  - cellwrapper, A-27
  - cmdfile, A-28
  - cmdkey, A-29
  - contentarea, A-30
  - copies, A-31
  - customize, 16-1, 16-3, 16-15, 16-17, 16-21, 16-23, A-32
  - dateformatmask, A-33
  - delauth, A-33
  - delimited\_hdr, A-34
  - delimiter, A-35
  - desformat, A-36
  - desname, A-38
  - dest, A-39
  - destination, A-40
  - destype, A-41
  - distribute, A-46
  - dtype, A-46
  - dunit, A-48
  - engineerresponsetimeout, A-48
  - expiration, 1-7, A-49
  - expirerdays, A-50
  - express\_server, A-50
  - formsize, A-53
  - from, A-53
  - getjobid, A-54
  - getserverinfo, A-54
  - help, A-55
  - install, A-56, A-108
  - itemtitle, A-56
  - jobname, A-57

jobtype, A-57  
killjobid, A-59  
LONGCHUNK, 20-22, A-59  
mimetype, A-60  
mode, A-61  
module, 16-15, 16-23, A-61  
nonblocksql, A-62  
notifyfailure, A-62  
notifysuccess, A-63  
numberformatmask, A-64  
olap\_con, A-65  
onfailure, A-65  
onsuccess, A-66  
orientation, A-67  
outputfolder, A-68  
outputimageformat, A-69  
outputpage, A-70  
overwrite, A-71  
p\_availability, A-71  
p\_description, A-72  
p\_formats, A-73  
p\_name, A-73  
p\_owner, A-74  
p\_pformtemplate, A-74  
p\_printers, A-75  
p\_privilege, A-75  
p\_servers, A-76  
p\_trigger, A-76  
p\_types, A-77  
pagegroup, A-77  
pagesize, A-78  
pagestream, A-79  
paramform, A-80  
parsequery, A-81  
pdfcomp, A-81  
pdfembed, A-82  
printjob, A-82  
readonly, A-83, A-84  
RECURSIVE\_LOAD, 20-21  
replaceitem, A-85  
replyto, A-86  
report, 16-15, 16-23, A-61  
role, A-86  
rundebg, 20-21, A-87  
save\_rdf, A-88  
schedule, 13-22, A-88  
server, 2-5, 2-6, A-89

showenv, A-91  
showjobs, 1-2, 17-5, A-92  
showmap, A-93  
showmyjobs, A-94  
shutdown, 2-6, A-94  
sitename, A-95  
source, A-96  
ssoconn, A-97  
statusfolder, A-99  
statusformat, A-100  
statuspage, A-100  
stype, A-101  
subject, A-102  
tolerance, 1-7, 13-23, A-103  
tracefile, A-104  
tracemode, A-105  
traceopts, A-106  
urlparameter, A-109  
userid, A-109  
webserver\_debug, A-110  
webserver\_docroot, A-111  
webserver\_port, A-111  
セルの囲み文字, 20-12  
コマンドライン, 指定, 13-6

## オ

---

サーバー, 9-10  
    rwsrv コマンド, A-18  
    インプロセス, 1-2, 1-5, 3-46  
サーバー・アクセス制御, 11-13  
サーバー・クラスタ, 3-28, 3-47  
サーバーのクラスタ化, 3-28  
サブレット, 1-2, 1-5, 2-5, 3-3, 9-10  
    rwservlet, A-12  
    URL 構文, 13-2  
    カスタム・ヘルプの追加, 3-48  
事前フェッチ, 20-19  
実行, レポート, 13-1  
失敗したジョブ・キュー, 19-14  
自動実行, レポート, 13-22  
    OracleAS Portal の使用, 13-18  
終了したジョブ・キュー, 19-11  
出力  
    PostScript, 2-4  
    プリンタ, 2-4  
出力処理, 7-1 ~ 7-4

- 出力タイプ, 7-2
- 使用可能日カレンダー, 11-5 ~ 11-10
  - 結合, 11-8
  - シンプル, 11-6
- ジョブ・キュー
  - OEMによる管理, 19-8
  - OEMによる表示, 19-8
  - 現行, 19-8
  - 失敗, 19-14
  - 終了済, 19-11
  - スケジュール済, 19-10
- ジョブ, 実行, 13-1 ~ 13-27
- ジョブの取消し, 17-6
- シングルバイト, 18-12
- スクリプト
  - srw\_test.sql, 17-7
  - srwAPIdrop.sql, 17-2
  - srwAPIgrant.sql, 17-2
  - srwAPIlins.sql, 17-2
- スケジュールされたジョブ・キュー, 19-10
- ステータス・レコード, 17-6
- セッションCookie, 10-2
- 設定ファイル, 8-27
- セルの囲み文字
  - テキスト・データソース, 20-12
- 双方向サポート, 18-1 ~ 18-15
- ソート基準, 言語, 18-6
- 属性, 変数の使用, 15-3

## た

---

- 多言語テキストの表示, 18-13
- 中東圏の言語, 18-11
- 重複ジョブの検出, 13-24
- 月名, 言語, 18-6
- データ型
  - BLOB, 20-22
  - CLOB, 20-22
  - DATE, 18-3
  - LONG, 20-22
  - LONG RAW, 20-22
  - NUMBER, 18-3
- データ型ディクショナリ
  - distribution.dtd, 15-2, 15-34
  - DTD, 20-12
  - rwserverconf.dtd, 3-2, 3-4
- データソース

- XMLによるグループ階層, 16-11
- XMLによる作成, 16-9
- XMLによるリンク, 16-10
- データ定義
  - DTDスキーマ, 20-12
  - XMLスキーマ, 20-12
- データベース・トリガー, 17-7
- データベース認証Cookie, 3-47, 3-48
- データベースの索引, 20-12
  - SQL WHERE 句, 20-12
- データ・モデル, 作成, 16-9 ~ 16-15
- テキストの表示, 多言語, 18-13
- テキストの読取り順序, 18-1
- デキュー, プロシージャの作成, 17-11
- デバッグ・イベント, 17-7
- テンプレート, rwserver.template, 3-3
- 動的な環境切替え, 3-34
- 同等の記号, 18-6
- ドライバ, 8-7
- ドライバ, カスタム, 8-13
- トリガー, データベース, 17-7
- トレース, 16-16
- トレース・ファイル出力
  - Oracle 時間, 20-7
  - SQL, 20-7
  - UPI, 20-7
  - 合計経過時間, 20-6
- トレーラ, 20-20

## な

---

- 名前 / 値のペア, 宛先, 7-6
- 認証Cookie, 3-47, 3-48

## は

---

- バーコード・フォント, 4-33
- ハイパーリンク, XMLによる追加, 16-6
- 配布, 15-1 ~ 15-32
  - 一括処理, 15-5, 20-20
    - 「繰返し」プロパティ, 20-20
    - グループ・レコード・レベル, 20-20
    - 「配布」プロパティ, 20-20
    - ヘッダー, 20-20
- 配布, XMLファイルの使用, 15-32
- 配布の概要, 15-2
- 配布の例, 15-22 ~ 15-32



- file, 15-28
- foreach, 15-22
- mail, 15-25
- printer, 15-29
- 配布要素
  - attach, 15-11
  - body, 15-10
  - destinations, 15-4
  - destype, 15-19
  - file, 15-15
  - foreach, 15-5
  - include, 15-13
  - mail, 15-7
  - printer, 15-17
  - property, 15-22
- バッチ変更, XML, 16-21
- パフォーマンスの監視, 19-16
- パラメータ・フォーム, 11-21
  - キー・マップ・ファイル, 13-26
- パラメータ・リスト (イベント)
  - 作成, 17-2 ~ 17-4
  - 操作, 17-2 ~ 17-4
- 秘密鍵, 12-5
- 表記の方向, 言語, 18-7
- ファイアウォール
  - プロキシ情報, 3-50
- フォーマット
  - グラフィカル・レイアウト・オブジェクト, 20-17
  - 垂直拡張度, 20-16
  - 水平拡張度, 20-16
  - テキスト・フィールドの行数, 20-17
  - 非グラフィカル・レイアウト・オブジェクト, 20-16
- フォーマット・トリガー, 20-17
- フォント, TrueType Big, 18-13
- フォントのエイリアシング, 18-6
- フォントのサポート, NLS, 18-13
- フォント・マッピング, 18-6
- プラグガブル
  - デスティネーション, 15-1, 15-34
- プリンタ・アクセス制御, 11-10
- プリンタ, 出力先, 2-4
- ブレイク・グループ
  - ORDER BY, 20-14
- プロキシ情報
  - 構成, 3-50
- 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, 13-17
- 変数, NLS の環境, 18-2 ~ 18-8
- 変数, XML 属性での使用, 15-3
- 方向, 言語, 18-7
- ポートレット
  - ページへの追加, 13-17, 13-20
  - レポート定義ファイル・アクセスの作成, 13-17
  - レポートに対するプロバイダの作成, 13-17
- ボディ, 20-20
- 翻訳
  - PL/SQL ブロック, 18-14
  - TranslationHub ツール, 18-14

## ま

---

- マップ
  - URL パラメータ, キー・マップ・ファイル, 13-25
- マルチバイト, 18-1, 18-5, 18-12
- メッセージ・キュー, 作成, 17-9
- メッセージ, 言語, 18-6

## や

---

- 要素, 「配布」, 「カスタマイズ」または「構成」を参照
- 曜日名, 言語, 18-6
- 読取り順序, 18-1, 18-11

## ら

---

- ランタイム URL, 13-1 ~ 13-27
- ランタイム URL 構文, 13-2
- ランタイム・パラメータ・フォーム, 11-21
- リスナー, 1-5
- 例, 配布, 15-22 ~ 15-32
  - file, 15-28
  - foreach, 15-22
  - mail, 15-25
  - printer, 15-29
- レジストリ, Windows, 18-5
- レジストリ, 編集, B-1
- レポート
  - OracleAS Portal からのバッチ削除, C-5
  - Portal 経由の要求, 13-6
  - URL 要求, 13-6
  - XML カスタマイズ, 16-3 ~ 16-9
  - XML データ・モデル, 16-9 ~ 16-15

- 一括処理, 15-5
- イベントによる起動, 17-7
- カスタム XML の適用, 16-15 ~ 16-21
- カスタム XML のデバッグ, 16-21 ~ 16-24
- キャッシュ, 13-23
- コマンドライン要求, 13-6
- 自動実行, 13-22
- スケジューリング, 13-22
- 要求の方法, 13-6
- レポート・アクセス制御, 11-16
- レポート, 実行, 13-1 ~ 13-27
- レポート定義, XML, 16-1 ~ 16-24
- レポート, トレース
  - トレース・ファイルの生成
    - Reports Builder, 20-4
- レポート・トレース, 20-3
  - トレース・ファイルの生成, 20-4
    - RWBUILDER, 20-5
    - RWRUN, 例, 20-6
    - RWSERVER, 20-5
    - RWSERVLET, 20-5
- レポート要求の指定
  - URL エンジン, 13-21
  - Web ブラウザから, 13-21
  - 自動実行のスケジューリング, 13-22
  - レポート・ポートレットの公開, 13-16