

Oracle9i Application Server Oracle Reports Services

レポート Web 公開ガイド

リリース 1.0.2

2000 年 12 月

部品番号 : J02835-01

Oracle9i Application Server Oracle Reports Services レポート Web 公開ガイド, リリース 1.0.2

部品番号 : J02835-01

原本名 : Oracle Reports Services Publishing Reports to the Web with Oracle9i Application Server, Release 1.0.2

原本部品番号 : A86784-02

原著者 : Frank Rovitto

原協力者 : Pat Hinkley

Copyright © 1996, 2000, Oracle Corporation. All rights reserved.

Printed in Japan.

制限付権利の説明

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

当プログラムのリバース・エンジニアリング等は禁止されています。

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

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

危険な用途への使用について

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

Restricted Rights Notice

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

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

目次

はじめに	ix
Oracle Reports Services の新機能と変更点	ix
対象読者	ix
このマニュアルの構成	x
関連ドキュメント	x

第 I 部 レポートの公開

1 公開のアーキテクチャと概念

1.1	Oracle Reports Services	1-1
1.2	Oracle Reports Services のアーキテクチャ	1-2
1.2.1	Web アーキテクチャ：サーバーの構成	1-3
1.2.1.1	Web レポートの処理	1-5
1.2.2	非 Web アーキテクチャ：サーバーの構成	1-6
1.2.2.1	レポートの処理	1-6
1.3	Oracle Reports Services の構成方法	1-7
1.3.1	使用可能な Web リクエストおよび非 Web リクエスト	1-7
1.3.2	Oracle Reports Services サーバー CGI またはサープレットの選択	1-8
1.3.3	Oracle Reports Services のロケーションの選択	1-8

2 Oracle9i Application Server と Oracle Reports Services のインストール

2.1	Oracle Universal Installer について	2-1
2.2	Oracle HTTP Server powered by Apache について	2-2

3 Windows NT および UNIX 上での Oracle Reports Services サーバーの構成

3.1	Oracle Reports Services サーバーの起動および停止	3-1
3.1.1	Oracle Reports Services サーバーの起動	3-1
3.1.1.1	Windows NT 上での Oracle Reports Services サーバーの起動	3-2
3.1.1.2	UNIX 上での Oracle Reports Services サーバーの起動	3-3
3.1.2	Windows NT 上での非サービスとしての Oracle Reports Services サーバーの起動	3-3
3.2	Oracle Reports Services サーバーの停止	3-4
3.2.1	Windows NT 上での Oracle Reports Services サーバーの停止または削除	3-4
3.2.2	UNIX 上での Oracle Reports Services サーバーの停止	3-4
3.3	Oracle Reports Services サーバー・サブプレットの構成	3-5
3.3.1	JSDK を使用する Oracle Reports Services サーバー・サブプレットの構成	3-5
3.3.2	JServ を使用する Oracle Reports Services サーバー・サブプレットの構成	3-7
3.4	Oracle HTTP Server powered by Apache Listener の構成	3-8
3.5	Web サーバーの構成	3-9
3.5.1	Oracle Reports Services サーバー CGI の構成	3-10
3.5.1.1	Oracle Reports Services サーバー CGI の構成	3-10
3.5.1.2	Oracle Reports Services サーバーのサービス・エントリの作成	3-10
3.5.1.3	デフォルトで使用する Oracle Reports Services サーバーの設定 (オプション)	3-11
3.5.1.3.1	Windows NT	3-11
3.5.1.3.2	UNIX	3-11
3.6	環境変数による Oracle Reports Services サーバーの構成	3-12
3.6.1	Windows NT での環境変数による Oracle Reports Services サーバーの構成	3-12
3.6.1.1	環境変数の設定 (オプション)	3-12
3.6.1.2	Oracle Reports Services サーバーの起動	3-13
3.6.2	UNIX での環境変数による Oracle Reports Services サーバーの構成	3-13
3.6.2.1	環境変数の設定 (オプション)	3-13
3.6.2.2	UNIX 上での Oracle Reports Services サーバーの起動	3-14
3.7	環境変数	3-15
3.8	Web ブラウザからのレポート・リクエストの実行	3-16
3.8.1	他のステップ	3-16
3.9	Oracle Reports Services サーバーの構成変更 (オプション)	3-17
3.9.1	ジョブのキュー・アクティビティによるデータベースの更新	3-17
3.9.1.1	Oracle Reports Services サーバー・マシンでの作業	3-18

4 レポート・リクエストの実行

4.1	レポート・リクエスト・メソッド	4-1
4.2	重複ジョブの検出	4-2
4.2.1	使用上の注意	4-3
4.3	キー・マップ・ファイルの使用方法	4-4
4.3.1	キー・マッピングを使用可能にする方法	4-4
4.3.2	URL パラメータのマッピング	4-5
4.4	レポート・リクエストの指定	4-6
4.4.1	レポートの構築	4-6
4.4.2	Web ブラウザからのレポート・リクエストの指定	4-7
4.4.3	自動実行のためのレポートのスケジューリング	4-8

5 Oracle Portal を使用した Oracle Reports Services セキュリティ

5.1	概要	5-2
5.1.1	Windows 環境への Oracle Reports Services 6i 用のセキュリティ DLL の作成	5-3
5.1.2	UNIX 環境への Oracle Reports Services 6i 用のセキュリティ・ライブラリの作成	5-3
5.2	データベース・レベルのセキュリティ	5-4
5.3	アプリケーション・レベルのセキュリティ	5-6
5.4	Oracle Portal との統合	5-6
5.4.1	Oracle Portal と Oracle Reports Services サーバー間の認証情報の共有	5-8
5.5	Oracle Portal の統合アーキテクチャ	5-8
5.6	Oracle Portal への Oracle Reports Services セキュリティのインストール	5-8
5.6.1	ステップ 1: Oracle データベースへの Oracle Portal のインストール	5-9
5.6.2	ステップ 2: Oracle Reports Services のインストール	5-9
5.7	セキュリティ環境の構成	5-9
5.7.1	ステップ 1: Oracle Portal 内の Oracle Reports Services セキュリティを使用可能にする	5-9
5.7.1.1	RW_ADMINISTRATOR	5-10
5.7.1.2	RW_DEVELOPER	5-10
5.7.1.3	RW_POWER_USER	5-11
5.7.1.4	RW_BASIC_USER	5-11
5.7.2	ステップ 2: SECURITYTNSNAMES パラメータおよび PORTALUSERID パラメータを追加する	5-11
5.7.3	ステップ 3: Oracle Portal を起動する	5-12
5.8	プリンタ・アクセス	5-12
5.9	Oracle Portal コンテンツ領域の作成	5-13

5.10	レポートのセットアップおよび配置	5-14
5.11	セキュリティ管理のための Oracle Portal ユーザーの作成と有効化	5-14
5.11.1	セキュリティ管理のためのユーザー REPORTSDEV の作成と有効化	5-15
5.12	Oracle Portal のアクセス制御のセットアップ	5-20
5.13	レポートの登録	5-20
5.13.1	サーバーの登録	5-21
5.13.2	レポート定義ファイル・アクセスの作成	5-25
5.14	レポートの配置	5-36
5.14.1	Oracle Portal コンテンツ領域へのレポートの配置	5-37
5.15	レポートの実行	5-44
5.16	Oracle Portal 以外のレポートの公開	5-47

6 Oracle Reports Services サーバーのクラスタの構成

6.1	クラスタリングの概要	6-2
6.2	Oracle Reports Services サーバーをクラスタ構成にする例	6-3
6.2.1	マスターとスレーブ間の通信を可能にする方法	6-4
6.2.2	マスター・サーバーを構成する方法	6-4
6.2.3	クラスタ構成でレポートを実行する方法	6-7
6.2.4	エンジンがダウンしたときにジョブを再送信する方法	6-7
6.2.5	他のスレーブ・サーバーをマスターに付加する方法	6-8

7 実行時のレポートのカスタマイズ

7.1	概要	7-2
7.1.1	XML レポート定義の作成および使用方法	7-2
7.2	XML レポート定義の作成	7-3
7.2.1	必須タグ	7-4
7.2.2	部分レポート定義	7-4
7.2.2.1	書式変更の例	7-6
7.2.2.2	書式例外の例	7-8
7.2.2.3	プログラム単位とハイパーリンクの例	7-9
7.2.2.4	データ・モデルと書式の変更の例	7-10
7.2.3	完全なレポート定義	7-11
7.3	XML レポート定義の実行	7-17
7.3.1	実行時の XML レポート定義の適用	7-17
7.3.1.1	1 つの XML レポート定義の適用	7-18
7.3.1.2	複数の XML レポート定義の適用	7-18

7.3.1.3	PL/SQL での XML レポート定義の適用	7-18
7.3.1.3.1	ファイルに保存された XML 定義の適用	7-19
7.3.1.3.2	メモリーに保存された XML 定義の適用	7-19
7.3.2	XML レポート定義の単独での実行	7-22
7.3.3	バッチ変更の実行	7-22
7.4	XML レポート定義のデバッグ	7-23
7.4.1	XML パーサーのエラー・メッセージ	7-23
7.4.2	トレース・オプション	7-23
7.4.3	RWBLD60	7-25
7.4.4	TEXT_IO	7-26
7.5	XML タグ・リファレンス	7-27
7.5.1	<!-- コメント -->	7-27
7.5.2	<![CDATA[]]>	7-28
7.5.3	<condition>	7-29
7.5.4	<customize>	7-31
7.5.5	<data>	7-33
7.5.6	<dataSource>	7-34
7.5.7	<exception>	7-35
7.5.8	<field>	7-37
7.5.9	<formLike>	7-42
7.5.10	<formula>	7-43
7.5.11	<function>	7-45
7.5.12	<group>	7-46
7.5.13	<groupAbove>	7-48
7.5.14	<groupLeft>	7-49
7.5.15	<labelAttribute>	7-50
7.5.16	<layout>	7-52
7.5.17	<link>	7-55
7.5.18	<matrix>	7-57
7.5.19	<matrixCell>	7-60
7.5.20	<matrixCol>	7-61
7.5.21	<matrixRow>	7-62
7.5.22	<object>	7-63
7.5.23	<programUnits>	7-65
7.5.24	<properties>	7-66
7.5.25	<property>	7-67
7.5.26	<report>	7-70
7.5.27	<section>	7-72

7.5.28	<select>	7-74
7.5.29	<summary>	7-75
7.5.30	<tabular>	7-79

第 II 部 付録

A カレンダの定義によるレポートへのユーザー・アクセスの管理

A.1	使用可能日カレンダの作成	A-1
A.2	使用可能日カレンダの例	A-2
A.2.1	デイリー・カレンダの作成	A-2
A.2.2	メンテナンス・カレンダの作成	A-3
A.2.3	クリスマス・カレンダの作成	A-4
A.2.4	結合された使用可能日カレンダの作成	A-5

B RWCLI60 コマンドライン引数

B.1	構文	B-1
B.2	使用上の注意	B-1

C Oracle Reports Services 構成パラメータ

D 環境変数

E データベース接続文字列

F Web カートリッジから CGI への移行

F.1	CGI への移行の利点	F-1
F.2	CGI に移行するためのステップ	F-2
F.2.1	ステップ 1. ソフトウェアのインストール	F-2
F.2.2	ステップ 2. OAS の構成	F-3
F.2.3	ステップ 3. CGI の構成	F-3
F.2.4	ステップ 4. 環境変数の設定 (オプション).....	F-4
F.2.4.1	Windows NT	F-4
F.2.4.2	UNIX	F-4
F.2.5	ステップ 5. マップ・ファイルの名前の変更 (オプション).....	F-4

F.2.6	ステップ 6. CGI URL を使用したレポートの実行	F-5
F.2.7	Web ページ上のレポート・リンクの更新	F-5

G トラブルシューティング

用語集

索引

はじめに

このマニュアルではレポートを公開するための Oracle Reports Services ソフトウェアの構成方法について説明します。また Oracle 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触れます。

Oracle Reports Services の新機能と変更点

このリリースでは次の機能が新しくなりました。

新機能と変更点	下記参照
Oracle HTTP Server powered by Apache。 httpds.conf ファイルが httpd.conf に変更されました。	第 3 章「Windows NT および UNIX 上での Oracle Reports Services サーバーの構成」
Oracle Portal。Oracle Portal を使用した Oracle Reports Services 6i のセキュリティ。	第 5 章「Oracle Portal を使用した Oracle Reports Services セキュリティ」

対象読者

このマニュアルは Oracle Reports Services を使用して行うレポートの公開に興味をお持ちの方を対象に書かれています。たとえば、すでにレポートを作成し、自社内の大勢の人達にそれを公開するという方や、誰かがかわりに作成したレポートがあり、それを他のユーザがアクセスできるように運用するという方がいると思います。Oracle Reports Services ソフトウェアをレポートの公開用に設定するには、オペレーティング・システム（たとえば、Windows NT や Solaris）や Net8 について十分な知識を持つ必要があります。Web 上でレポートを動的に運用しようと計画している方は、使用している Web サーバーの構成についても十分理解しておく必要があります。

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 Oracle Reports Services のアーキテクチャおよびレポートを構成する前に必要な選択事項を紹介します。
- 第 2 章 インストールについての情報を提供します。
- 第 3 章 Oracle Reports Services サーバーの構成方法について説明します。
- 第 4 章 Oracle Reports Services サーバーにレポート・リクエストを出すためのいろいろな方法について説明します。
- 第 5 章 Oracle Reports Services と Oracle Portal を統合し、レポートへのユーザー・アクセスを管理できるようにする方法を説明します。
- 第 6 章 パフォーマンスと信頼性を高めるために、クラスタリング手法を使用して Oracle Reports Services を構成する方法について説明します。
- 第 7 章 XML を利用して実行時にレポートのカスタマイズを適用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、Oracle Portal または Oracle Reports Services の詳細は、次のマニュアルを参照してください。

- 『Oracle Reports Developer レポート作成ガイド リリース 6*i*』 J00918-01
- 『Oracle Reports Developer for Windows スタート・ガイド リリース 6*i*』 J00454-01
- 『Oracle Portal チュートリアル リリース 3.0』 J02977-01
- 『Oracle9*i* Application Server for Windows and UNIX Forms アプリケーション Web 利用ガイド リリース 1.0.2』 J02409-01

第I部

レポートの公開

第1章「公開のアーキテクチャと概念」

第2章「Oracle9i Application Server と Oracle Reports Services のインストール」

第3章「Windows NT および UNIX 上での Oracle Reports Services サーバーの構成」

第4章「レポート・リクエストの実行」

第5章「Oracle Portal を使用した Oracle Reports Services セキュリティ」

第6章「Oracle Reports Services サーバーの クラスタの構成」

第7章「実行時のレポートのカスタマイズ」

公開のアーキテクチャと概念

動きが速く競争の激しい今日のビジネス界では、しばしば地理的に分散された労働力の要件を正確、かつ適切に決定するために、明確で最新の情報が必要とされています。そのような情報のタイムリーな分配は、常に高信頼性で、コスト面でも優れ、かつその情報は求める人すべてにとってアクセス可能なものであることが必要です。Oracle Reports Services は、質の高いデータベース公開およびレポートのための、制限がなく、使いやすく、拡張性や操作性に優れた解決策を提供するものです。

Oracle Reports Services は、Web 上および企業全体における洗練された動的レポートを作成するために情報システム (IS) 開発者によって使用される非常に優れた企業レポート・ツールです。

Oracle Reports Services のサーバーベースのアーキテクチャでは、レポートの消費者が業界標準フォーマットのレポートを表示するのに必要なものは、Web ブラウザのみで十分です。Oracle Reports Services は、カスケード・スタイル・シートを使用した HTML および Adobe の Portable Document Format (PDF) の作成を通し、Web 上での質の高いレポートの需要に応じた配布をサポートしています。レポートは、集中的に維持・管理され、すべてのユーザーの PC に複雑なソフトウェアのインストールを必要としないため、メンテナンスの諸経費は軽減されています。

1.1 Oracle Reports Services

Oracle Reports Services を使用すれば、レポートを実行するために複数層アーキテクチャをインプリメントできます。また、Oracle Reports Services によって、リモート・アプリケーション・サーバー上でレポートを実行することもできます。

Oracle Reports Services サーバー CGI または Oracle Reports Services サーバー・サブレットと組み合わせて使用することで、Oracle Reports Services は標準的な URL 構文を使用して Web ブラウザからレポートを実行することも可能になります。Oracle Reports Services は、Windows NT、Windows 95 または Unix 上にインストールできます。クライアント要求を操作し、すべてのジョブ要求を 1 つのジョブ・キューに入れることによってレポートを実行できます。いずれかのサーバーのランタイム・エンジンが有効になると、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、サーバーはランタイム・エ

ンジンサーバーのプロセス開始時に指定した最大数まで増やすことができます。同様に、アイドル状態のエンジンは、指定した時間のアイドル状態を超えた場合、シャットダウンされます。

Oracle Reports Services は、定義済の過去の最大ジョブ数を追跡します。レポートの最終的なステータスとともに、ジョブがいつキューされたのか、開始されたのかおよび完了したのかについての情報が保持されます。これらの情報は、Windows の Oracle Reports Services Queue Manager (RWRQM60) から、もしくは API を使用して検索および再表示できます。Oracle Reports Services Queue Manager は、Oracle Reports Services と同一のマシン上に存在する場合も、クライアント・マシン上に存在する場合もあります。UNIX では、Oracle Reports Services Queue Viewer (RWRQV60) を使用して Oracle Reports Services のキューを表示できます。

1.2 Oracle Reports Services のアーキテクチャ

Oracle Reports Services には、ユーザーのリクエストに応じていくつかの構成方法が考えられます。Web 環境で使用する場合は、Oracle 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。¹

- シン・クライアント層
- Web サーバー層
- Oracle Reports Services 層
- データベース層

可能な構成方法の範囲は、すべての階層を同一マシン上に持つ場合から各階層をすべて別のマシン上に持つ場合まで、様々です。最も一般的な構成方法は、階層を 3 から 4 つのマシン上に分散させるものです。以下のグラフィックは、このような一般的な構成の概念を表しています。

注意： 非 Web の場合（後で説明しています）は Web サーバー層が不要なため、3 つの階層になります。

¹ 階層という言葉は、Oracle Reports Services のアーキテクチャに準ずる構成要素の論理的ロケーションを意味します。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。

1.2.1 Web アーキテクチャ：サーバーの構成

次に示す図は、Web 環境における Oracle Reports Services の、2 つの最も一般的な構成方法を示しています。2 つの構成の大きな相違点は、Oracle Reports Services と Web サーバーの階層が同じマシン上に存在するか、異なるマシン上に存在するかという点です。最初の構成の場合、Web サーバーと Oracle Reports Services は同一のマシン上に存在しています。2 つ目の構成の場合は、それらが異なるマシン上に存在しています。後者の場合、前者の場合とは若干異なるセットアップが必要となります。

図 1-1 Web アーキテクチャ、3 つのマシンを使用した構成

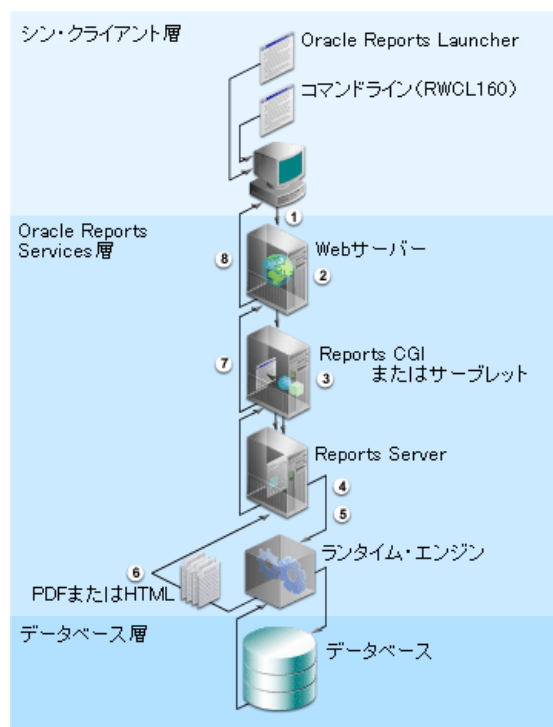
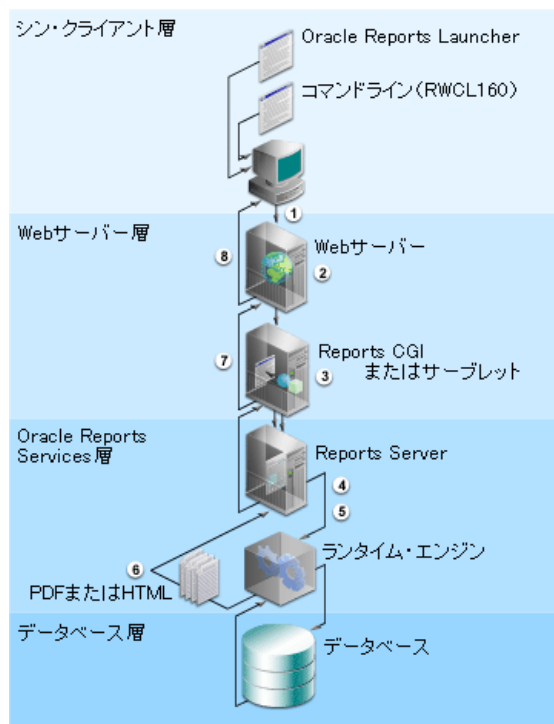


図 1-2 Web アーキテクチャ、4 つのマシンを使用した構成



1.2.1.1 Web レポートの処理

Web レポートは、次のように処理されます。

1. クライアントは、URL をタイプ入力するか、ハイパーリンクをクリックすることにより Web ブラウザにレポートを要求します。Web ブラウザが、URL を Web サーバーに受け渡します。
2. リクエストを処理するために、Web サーバーは構成されている Oracle Reports Services サーバー CGI または Oracle Reports Services サーバー・サブレットのどちらかをコールします。
3. Oracle Reports Services サーバー CGI またはサブレットは、そのリクエストを解釈します。ユーザーは、ログオンが必要な場合はログオンします。Oracle Reports Services サーバー CGI またはサブレットは、リクエストを Oracle Reports Services が実行可能なコマンドラインに変換し、指定された Oracle Reports Services サーバーに送信します。
4. リクエストに許容時間が含まれる場合²、Oracle Reports Services はその出力キャッシュをチェックして、そのリクエストを満たす出力がすでに存在しているかどうかを決定します。キャッシュ内にリクエストを満たす出力があった場合、サーバーはレポートの実行よりも、その出力を返すことを優先して行います。
5. Oracle Reports Services は、ジョブ・リクエストを受け取り、それをキューに入れます。ランタイム・エンジンのいずれかが使用可能になると、³そのランタイム・エンジンが実行するためのコマンドラインを送ります。
6. ランタイム・エンジンは、レポートを実行します。
7. Oracle Reports Services サーバー CGI またはサブレットは、レポート出力を Oracle Reports Services から受け取り、それを Web サーバーに送ります。
8. Web サーバーは、そのレポート出力をクライアントの Web ブラウザに送ります。

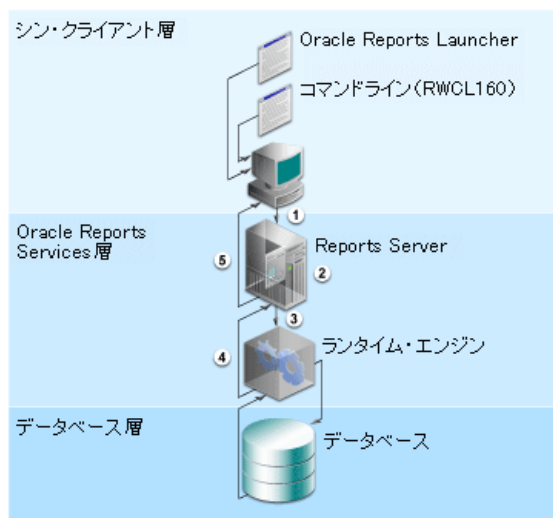
² Oracle Reports Services に送るどのようなジョブ・リクエストにも、TOLERANCE 引数を含めることができます。TOLERANCE は、要求者が許容できる最も古い出力を定義します。たとえば、要求者が TOLERANCE として 5 分を設定した場合、Oracle Reports Services はキャッシュをチェックして過去 5 分以内に作成された複製レポート出力を探します。

³ Oracle Reports Services を構成する際に、使用可能な最大ランタイム・エンジン数を指定できます。Oracle Reports Services においてこの最大数を下回る場合は、リクエストを扱うために新しいランタイム・エンジンを開始することもできます。そうでない場合は、現在のランタイム・エンジンが現在のジョブを完了するまで、リクエストの実行を待つ必要があります。

1.2.2 非 Web アーキテクチャ：サーバーの構成

非 Web アーキテクチャは、Web ブラウザや Web サーバーを持たないという点が Web アーキテクチャと異なります。レポート・リクエストは、Oracle Reports Services Launcher またはコマンドライン（RWCLI60）などのシン・クライアントから、Oracle Reports Services に送られます。非 Web アーキテクチャは、なんらかの理由によってレポートを実行するのに Web を使用できないユーザーにとって便利なものです。

図 1-3 非 Web アーキテクチャ



1.2.2.1 レポートの処理

非 Web 環境では、レポートは次のように処理されます。

1. クライアントは、コマンドライン（RWCLI60）、Oracle Reports Services Queue Manager または Oracle Reports Services Launcher（ActiveX コントロール）を使用してレポートを要求します。ユーザーは、ログオンが必要な場合はログオンします。
2. Oracle Reports Services は、ジョブ・リクエストを受け取り、それをキューに入れます。ランタイム・エンジンのいずれかが使用可能になると、そのランタイム・エンジンが実行するためのリクエストを送ります。
3. ランタイム・エンジンは、レポートを実行します。
4. Oracle Reports Services にジョブが完了したことが伝えられます。

5. Oracle Reports Services が同期的にコールされていた場合、クライアントにジョブが完了したことが伝えられます。また、ジョブ・リクエストにおいてコマンドライン・クライアントの宛先タイプ (DESTYPE) に localfile が設定されている場合、出力はクライアントに送られます。

1.3 Oracle Reports Services の構成方法

Oracle Reports Services の構成方法は、システムの要求事項によって様々です。Oracle Reports Services を構成する前に、自分の要求事項に基づいて重要事項を決定する必要があります。事前にそのような重要事項を決定しておくことにより、構成プロセスは非常に簡単になります。これらの決定事項は、次の各項で説明されています。

1.3.1 使用可能な Web リクエストおよび非 Web リクエスト

1.2 項「[Oracle Reports Services のアーキテクチャ](#)」にあったように、Oracle Reports Services は Web のシン・クライアントおよび非 Web のシン・クライアントの両方からのジョブ・リクエストを受けられます。Web の場合、ユーザーは Web ブラウザにおいて URL をクリックまたはタイプ入力することでレポートを実行でき、URL によって、レポート出力をブラウザ上に返したり、指定した宛先（たとえば、プリンタ）に送ることができます。非 Web の場合、ユーザーはマシンにインストールされているクライアントのソフトウェア (Net8 および、Oracle Reports Services Launcher、Oracle Reports Services Queue Manager、RWCLI60 から成る Oracle Reports Services Thin Client) を使用してジョブ・リクエストを出します。

ユーザーが Web クライアントからレポートを発信するためには、Oracle Reports Services サーバー CGI または サブレットを Web サーバーにインストールして、Web サーバーと Oracle Reports Services 間で通信が行われるようにする必要があります。CGI または サブレットは、Web サーバーが Web クライアントからのレポート・リクエストを処理するために必要です。詳細は、[1.3.2 項「Oracle Reports Services サーバー CGI またはサブレットの選択」](#)を参照してください。ユーザーが非 Web クライアントからレポートを発信するためには、レポート・リクエストを発信する予定のある各マシンに、必要なクライアント・ソフトウェア (Net8 や Oracle Reports Services Thin Client) をインストールする必要があります。

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

- Web リクエストを有効化する場合、Web サーバーにいくつかの追加ソフトウェア (Oracle Reports Services サーバー CGI またはサブレット) をインストールする必要がありますが、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 リクエストを有効化する場合、Oracle Reports Services にジョブ・リクエストを発信する予定のあるすべてのマシンにクライアント・ソフトウェアをインストールし、メンテナンスする必要があります。

Web の場合の方が、クライアントのメンテナンス・コストを軽減でき、明らかにコスト面で優れていることが分かります。しかし、他の理由で、非 Web リクエストを発信することが必要となる場合もあります。Oracle Reports Services は、Web および非 Web 両方のリクエストをサポートしており、相互排他的ではありません。

1.3.2 Oracle Reports Services サーバー CGI またはサープレットの選択

1.3.1 項「[使用可能な Web リクエストおよび非 Web リクエスト](#)」で説明したように、Web 環境で Oracle Reports Services を使用するには、Web サーバーと Oracle Reports Services 間におけるジョブ・リクエストの送信や出力を処理するために、Oracle Reports Services サーバー CGI またはサープレットをインストールして構成する必要があります。どちらをインストールするかを決定する場合の重要な考慮事項は、次のとおりです。

- CGI 関連の Web サーバー（たとえば、Oracle9i Application Server、Oracle Portal Listener）を使用している場合は、Oracle Reports Services サーバー CGI を選択します。
- Java ベースの Web サーバーを使用している場合は、Oracle Reports Services サーバー・サープレットを選択します。

1.3.3 Oracle Reports Services のロケーションの選択

1.2 項「[Oracle Reports Services のアーキテクチャ](#)」で説明しているように、Oracle Reports Services は Web サーバーと同じマシン上に設けることも、異なるマシン上に設けることもできます。どのマシン上に設けるかを決定する際、次の事柄を考慮してください。

- Oracle Reports Services と Web サーバーを同一マシン上に設ける場合、必然的により多くのマシンのリソースが必要となります。両方を同一マシンに設ける場合、マシンのリソース要件（メモリーおよびディスク容量）を決定する際にこのことを考慮する必要があります。
- Oracle Reports Services と Web サーバーを同一マシン上に設けることによって、ネットワークの通信量を軽減できます。Oracle Reports Services サーバー CGI またはサープレットは、Web サーバーと同一のマシン上に存在する必要があります。Oracle Reports Services を異なるマシン上に設ける場合、Oracle Reports Services サーバー CGI またはサープレットへ発信した情報は、ネットワーク全体を通ることになります。しかし、同一マシン上にあれば、情報がネットワーク上を通る必要はありません。

[第 3 章「Windows NT および UNIX 上での Oracle Reports Services サーバーの構成」](#)に、Oracle Reports Services サーバー・サープレットを使用した Oracle Reports Services の構成ガイドラインが示されています。

Oracle9i Application Server と Oracle Reports Services のインストール

Oracle Reports Services は、Oracle9i Application Server の Enterprise Edition の一部としてインストールされます。大量のトランザクションを処理する中規模から大規模の Web サイトでは、Enterprise Edition を使用することをお薦めします。

ユーザーの利便性を考慮して、Common Gateway Interface (CGI) をサポートする Web リスナーである Oracle HTTP Server powered by Apache が提供されています。Oracle HTTP Server powered by Apache は、Oracle Universal Installer によってインストールできます。Oracle Universal Installer は、Oracle9i Application Server に付属しています。

Oracle Reports Services のインストールの詳細は、『Oracle9i Application Server インストール・ガイド』を参照してください。必要な要件および作業はすべて、このマニュアルに記載されています。

2.1 Oracle Universal Installer について

Oracle9i Application Server では、Java ベースのツールである Oracle Universal Installer を使用して、環境変数の構成ならびにコンポーネントのインストールをします。インストーラでは、インストール・プロセスのステップごとに説明が表示されるので、様々な構成オプションを選択できます。

インストーラには、次の処理を実行する機能があります。

- 製品のインストール・オプションの参照および提供
- 事前設定された環境変数および構成設定の検出
- 環境変数および構成設定のインストール時の設定
- 製品の削除

2.2 Oracle HTTP Server powered by Apache について

Oracle9i Application Server では、Oracle HTTP Server powered by Apache Web サーバー・テクノロジーが使用されます。Apache Web サーバー・テクノロジーを使用すると、次のものが提供されます。

- 拡張性
- 安定性
- 速度
- 伸展性

Apache サーバーでは、HTTP リクエストの処理がそのモジュール (mods) に委譲されます。このモジュールによって、デフォルトではサーバーに備わっていない機能が追加されます。Apache API を使用すると、Apache 機能の拡張が容易になります。すでに多くの mods が作成されており、それらは CD-ROM に格納されています。デフォルトの Apache HTTP サーバーでは小規模なトランザクション¹のみがサポートされていますが、Apache JServ (mod_jserv) によって提供される機能を利用することによって、大規模なトランザクション²をサポートするように構成できます。Apache JServ (mod_jserv) は、『Oracle9i Application Server 概要』で説明されています。

¹ 小規模なトランザクションは、リクエストと応答から成ります。小規模なトランザクションでは、ユーザー (リクエスト) についての情報はシステムによって追跡されず、各トランザクションは、その前後のトランザクションと関連していません。

² 大規模なトランザクションは、データベース・セッションに似ています。これは、ユーザー (トランザクションの起動側) についての情報が、トランザクションの 1 つ以上のフェーズで、システムによって追跡されるためです。大規模なトランザクションでは、ユーザー情報の他に、トランザクション内の 1 つ以上の先行イベントの状態 (ある瞬間の状態セット) もシステムによって追跡されます。

Windows NT および UNIX 上での Oracle Reports Services サーバーの構成

Oracle HTTP Server powered by Apache とともに Oracle9i Application Server をインストールすると、Windows NT 環境および UNIX 環境において、Oracle Reports Services サーバー・サブレットと Oracle Reports Services サーバー CGI が自動的に構成されます。この章では、デフォルトで提供された構成を手動で変更する方法について説明します。

また、Oracle Reports Services サーバーの起動方法や停止方法、および構成環境変数についても説明します。

3.1 Oracle Reports Services サーバーの起動および停止

この章では、Oracle Reports Services サーバーを起動、停止および再起動するように求められます。これを行う方法を次に示します。

3.1.1 Oracle Reports Services サーバーの起動

次の項では、Windows NT または UNIX 上で Oracle Reports Services サーバーを起動する方法について説明します。

3.1.1.1 Windows NT 上での Oracle Reports Services サーバーの起動

Windows NT 上で Oracle Reports Services サーバーを起動するには、次のステップに従います。

1. Oracle Reports Services サーバー・マシンのデスクトップで、「**スタート**」「**設定**」「**コントロール パネル**」を選択して、 (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 Reports Server [repserver]**」 (repserver は Oracle Reports Services サーバー・インスタンス名) を選択して「**スタートアップ**」をクリックし、「サービス」ウィンドウを表示します。
3. スタートアップ・ダイアログ・ボックスで、「**ログオン**」セクションの「**アカウント**」をクリックし、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、そのサーバーがそのユーザーとして実行されるよう指定されます。

PostScript またはプリンタに出力するときは、Oracle Reports Services サーバーのサービスを実行しているユーザーがデフォルトのプリンタに対するアクセス権限を持っていることを確認してください。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。

4. 「**スタートアップの種類**」を「**自動**」に設定します。
5. 「**OK**」をクリックします。
6. 「**スタート**」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 Oracle Reports Services サーバーが起動した日時が表示されます。Oracle Reports Services サーバーが起動しない場合は、[付録 G「トラブルシューティング」](#)を参照してください。

Oracle Reports Services サーバーを初めて起動したとき、Oracle Reports Services サーバーの構成ファイル (たとえば repserver.ora) が ORACLE_HOME\REPORT60\SERVER ディレクトリに作成されます。Oracle Reports Services サーバーのキャッシュの設定はデフォルトで設定されます。構成ファイルを変更して、キャッシュ・ディレクトリを変更したり、レポートのソース・パスを設定できます。構成ファイルを変更したら、Oracle Reports Services サーバーを停止して再起動し、変更内容を有効にします。

3.1.1.2 UNIX 上での Oracle Reports Services サーバーの起動

UNIX 上で Oracle Reports Services サーバーを起動するには、次のステップに従います。

1. \$ORACLE_HOME/BIN ディレクトリから次のコマンドラインを実行して、Oracle Reports Services サーバーをフォアグラウンドで実行します。

```
rwmts60 name=repserver
```

あるいは、次のコマンドを実行して、Oracle Reports Services サーバーをバックグラウンドで実行します。

```
rwmts60 name=repserver &
```

2. \$ORACLE_HOME/BIN ディレクトリから次のコマンドラインを実行して、Oracle Reports Services サーバーが稼働していることを確認します。

```
rwrgv60 server=repserver
```

Oracle Reports Services サーバーのステータスの列 (NAME、OWNER、DEST など) が表示されます。ただし今はジョブが実行されていないのでステータス情報は表示されません。

Postscript あるいはプリンタに出力するときは、
(\$ORACLE_HOME/guicommon6/tk60/ADMIN ディレクトリにある) uiprint.txt ファイルでプリンタを構成する必要があります。

3.1.2 Windows NT 上での非サービスとしての Oracle Reports Services サーバーの起動

次のコマンドを実行します。

```
rwmts60 -listen repserver
```

あるいはバッチ・モードの場合は次のように入力します。

```
rwmts60 -listen repserver batch=yes
```

repserver にドメイン修飾子 (.world など) を付加する必要はありません。

3.2 Oracle Reports Services サーバーの停止

次の項では、Windows NT および UNIX 上で Oracle Reports Services サーバーを停止する方法について説明します。

3.2.1 Windows NT 上での Oracle Reports Services サーバーの停止または削除

Windows NT 上で Oracle Reports Services サーバーを停止するには、次のようにします。

1. Oracle Reports Services サーバー・マシンのデスクトップで、「**スタート**」 「**ファイル名を指定して実行**」を選択します。
2. 次のコマンドライン引数を入力します。

```
rwmts60 -uninstall repserver
```

あるいはバッチ・モードの場合は次のように入力します。

```
rwmts60 -uninstall repserver batch=yes
```

repserver にドメイン修飾子 (.world など) を付加する必要はありません。

3.2.2 UNIX 上での Oracle Reports Services サーバーの停止

Oracle Reports Services サーバーを停止するには、次のいずれかを行います。

- Oracle Report Services サーバーがフォアグラウンドで実行されている場合は、停止するウィンドウにフォーカスされていることを確認し、[Ctrl] キーを押しながら [C] を押します。
- Oracle Report Services サーバーがバックグラウンドで実行されている場合は、コマンドラインで次のように入力します。

```
ps -ef |grep 'rwmts60'
```

その後、次のように入力します。

```
kill -9 process_number
```

3.3 Oracle Reports Services サーバー・サブレットの構成

Oracle HTTP Server powered by Apache を使用する場合、次の 2 つの Oracle Reports Services サーバー・サブレット構成があります。いずれも手動で変更できます。

- JSDK を使用する Oracle Reports Services サーバー・サブレット
- JServ を使用する Oracle Reports Services サーバー・サブレット

3.3.1 JSDK を使用する Oracle Reports Services サーバー・サブレットの構成

次の構成では、Oracle HTTP Server powered by Apache が次のディレクトリにインストールされていることを前提としています。

```
/private1/ias
```

また、Oracle Reports Services サーバーが次のディレクトリにインストールされていることも前提としています。

```
/private1/ias/6iserver
```

JSDK を使用する Oracle Reports Services サーバー・サブレットを構成するには、次のステップに従います。

1. 次のエントリを Oracle Reports Services サーバー・サブレットのプロパティ・ファイル `servlet.properties` に追加します (Oracle Reports Services サーバー・サブレットのプロパティ・ファイルは、たとえば `/private1/ias/Apache/Jsdk/examples` にあります)。

```
servlet.RWServlet.code=oracle.reports.rwsgi.RWServlet
```

2. Web サーバーの Java クラス・ディレクトリに、ディレクトリ階層 `oracle/reports/rwsgi` を作成します。

```
/private1/ias/Apache/Jsdk/examples/oracle/reports/rwsgi
```

その後、この新しいディレクトリに、次の場所にある `RWServlet.class` ファイルをコピーします。

```
/private1/ias/6iserver/reports60/java
```

3. 前のステップのルート・ディレクトリを `CLASSPATH` 環境変数に追加します。また、`/private1/ias/Apache/Ojsp/lib/servlet.jar` を `CLASSPATH` 環境変数に追加します。次に例を示します。

```
setenv CLASSPATH /private1/ias/Apache/jdk/bin:
/private1/ias/Apache/jdk/lib/classes.zip:
/private1/ias/Apache/Jsdk/examples:/private1/ias/Apache/Ojsp/lib/servlet.jar:
$CLASSPATH
```

4. 次のように PATH 変数を設定します。

```
setenv PATH /private1/ias/6iserver/bin:/private1/ias/Apache/Apache/bin:
private1/ias/Apache/jdk/bin:
private1/ias/Apache/jsdk/bin:$PATH
```

5. Oracle Reports Services サーバーを起動します。
6. 次のコマンドを実行することによって、Oracle Reports Services Server Servlet Runner を起動します。

```
servletrunner &
```

7. 次のようにして、Oracle Reports Services サーバー・サープレットが実行されていることを確認します。
- a. ブラウザから次の行を実行し、インストールおよびセットアップが正常に行われていることを確認します。

```
http://hostname:portno/servlet/RWServlet/help?
```

各項目の内容は次のとおりです。

hostname	Apache Listener が実行されているマシン名
portno	Apache Listener が起動されているポート名

これにより、「ヘルプ」ページがアクティブであることが示されます。

- b. ブラウザから次の行を実行し、Oracle Reports Services サーバーが起動されていることを確認します。

```
http://hostname:portno/servlet/RWServlet/showjobs?
server=repserver
```

- c. ブラウザから次の行を入力し、レポートを実行します。

```
http://hostname:portno/servlet/RWServlet?server=repserver+
report=ReportName+destype=cache+userid=ConnectString+desformat=htmlcss
```

キー・マッピング用の cgicmd.dat ファイルを使用することもできます。

構成ファイルを変更した後は、変更内容を認識させるために Oracle Reports Services サーバーを停止して再起動する必要があります。

3.3.2 JServ を使用する Oracle Reports Services サーバー・サーブレットの構成

Jserv を使用する Oracle Reports Services サーバー・サーブレットを実行するように Oracle HTTP Server powered by Apache を構成するには、次のようにします。変更は `ias_home/Apache/Jserv/etc/jserv.properties` ファイルに対して行います。`ias_home` は、Oracle9i Application Server をインストールした場所です。

1. 次の行を追加します。

```
wrapper.classpath=ias_home/Apache/Jserv/servlets
```

2. `wrapper.env=ORACLE_HOME=ias_home` という行を次のように変更します。

```
wrapper.env=ORACLE_HOME=ias_home/6iserver
```

3. `wrapper.env=LD_LIBRARY_PATH=ias_home/lib` という行を次のように変更します。

```
wrapper.env=LD_LIBRARY_PATH=ias_home/lib:ias_home/6iserver/bin:ias_home/6iserver/lib
```

4. 次の行を `Apache/Jserv/etc/zone.properties` ファイルに追加します。

```
servlet..RWServlet.code=oracle.reports.rwcgi.RWServlet
```

5. `RWServlet.class` ファイルを次のディレクトリにコピーします（このディレクトリの作成が必要な場合もあります）。

```
ias_home/Apache/Jserv/servlets/oracle/reports/rwcgi
```

`http://host:port/servlet/RWServlet` という URL によってサーブレットが実行されます。

6. Oracle Reports Services サーバーを起動します。

7. 次のコマンドを使用して、Oracle HTTP Server powered by Apache Listener を起動します。

```
httpdctl start
```

8. 次のようにして、Oracle Reports Services サーバーが実行されていることを確認します。

- a. ブラウザから次の行を実行し、インストールおよびセットアップが正常に行われていることを確認します。

```
http://hostname:portno/servlets/RWServlet/help?
```

これにより、「ヘルプ」ページがアクティブであることが示されます。

- b. ブラウザから次の行を実行し、Oracle Reports Services サーバーが起動されていることを確認します。

```
http://hostname:portno/servlets/RWServlet/showjobs?  
server=repserver
```

- c. ブラウザから次の行を入力し、レポートを実行します。

```
http://hostname:portno/servlets/RWServlet?server=repserver+  
report=ReportName+destype=cache+userid=ConnectString+desformat=htmlcss
```

キー・マッピング用の `cgicmd.dat` ファイルを使用することもできます。

構成ファイルを変更した後は、変更内容を認識させるために Oracle Reports Services サーバーを停止して再起動する必要があります。

3.4 Oracle HTTP Server powered by Apache Listener の構成

Oracle Reports Services サーバー CGI を実行するために Oracle HTTP Server powered by Apache Listener のデフォルトの構成を変更するには、次のようにします。

1. (`/private1/ias/Apache/Apache/conf` にある) ファイル `httpd.conf` に次のエントリを追加します。

```
ScriptAlias /cgi-bin/ "/private1/ias/6iserver/bin"
```

2. Oracle Reports Services サーバーを起動します。
3. 次のコマンドを使用して、Oracle HTTP Server powered by Apache Listener を起動します。

```
httpdctl start
```

4. 次のようにして、Oracle Reports Services サーバー CGI が実行されていることを確認します。

- a. ブラウザから次の行を実行し、インストールおよびセットアップが正常に行われていることを確認します。

```
http://hostname:portno/cgi-bin/rwcgi60/help?
```

これにより、「ヘルプ」ページがアクティブであることが示されます。

- b. ブラウザから次の行を実行し、Oracle Reports Services サーバーが起動されていることを確認します。

```
http://hostname:portno/cgi-bin/rwcgi60/showjobs?  
server=repserver
```

- c. ブラウザから次の行を入力し、レポートを実行します。


```
http://hostname:portno/cgi-bin/rwcgi60?server=repserver+
report=ReportName+destype=cache+userid=ConnectString+desformat=htmlcss
```

キー・マッピング用の `cgicmd.dat` ファイルを使用することもできます。

構成ファイルを変更した後は、変更内容を認識させるために Oracle Reports Services サーバーを停止して再起動する必要があります。

3.5 Web サーバーの構成

この構成例は、次の内容を前提としています。

- Web リクエストを可能にするように Oracle Reports Services サーバーを構成します。
- CGI 対応の Web サーバーとして Oracle HTTP Server powered by Apache を用いて Oracle Reports Services サーバー CGI を使用します。
- Oracle Reports Services サーバーは Web サーバーとは異なるマシン上にインストールされています。

Web サーバー上の CGI-BIN ディレクトリには、CGI 実行ファイルがあります。次のステップは Web サーバー・マシン上で行います。

1. Web サーバーを起動します。
2. ブラウザを起動します。
3. リスナーを作成します。
4. Web サーバーのマッピングを構成して、物理ディレクトリおよび仮想ディレクトリを書き留めておきます。次に例を示します。

表 3-1 CGI-BIN 物理・仮想ディレクトリ

ディレクトリの説明	物理ディレクトリ例	仮想ディレクトリ例	必要な権限
CGI-BIN	c:\¥orant¥oas¥bin	/CGI-BIN	実行 (execute)
Apache Web サーバー CGI-BIN	c:\¥program files¥Apache Group¥Apache¥cgi-bin	/CGI-BIN	実行 (execute)

物理ディレクトリの位置は、Web サーバー・ソフトウェアのインストール時に指定したディレクトリ設定によって異なります。

Web サーバーを構成する方法の詳細は、ベンダーが提供している Web サーバーのドキュメントを参照してください。

3.5.1 Oracle Reports Services サーバー CGI の構成

次のステップは Web サーバー・マシン上で行います。

3.5.1.1 Oracle Reports Services サーバー CGI の構成

Oracle Reports Services サーバー CGI を構成するには、(ORACLE_HOME¥BIN ディレクトリにある) rwcgi60.exe を CGI-BIN ディレクトリにコピーします。

表 3-1 「CGI-BIN 物理・仮想ディレクトリ」では、CGI 物理ディレクトリは C:¥your_webserver¥bin で、Apache Web サーバーを使用する場合は C:¥Program Files¥Apache Group¥Apache¥cgi-bin です。

CGI-BIN ディレクトリは、Web サーバーの構成で定義されます。Oracle Reports Services サーバー CGI は、CGI ディレクトリとしてマップされているパスに存在する必要があります。また、Oracle Reports Services の RDF ファイルは、Oracle Reports Services サーバーのみがアクセス可能なパスに存在する必要があります。Oracle Reports Services 6i サーバーおよび Oracle Portal Listener のデフォルトのインストールを選択した場合は、次のパスに rwcgi60.exe があります。

D:¥orant¥bin¥rwcgi60.exe

3.5.1.2 Oracle Reports Services サーバーのサービス・エントリの作成

Web サーバーと Oracle Reports Services サーバーが別のマシン上にある場合は、Oracle Reports Services サーバーのサービス・エントリを Web サーバーに追加する必要があります。このサービス・エントリは Oracle Reports Services サーバー・マシンの tnsnames.ora ファイルに作成されています。tnsnames.ora ファイルは、Web サーバー・マシン上にもあります。エントリを追加すると、CGI 実行ファイルと Oracle Reports Services サーバーとの通信が可能になります。

Oracle Reports Services サーバーのサービス・エントリの設定を確認する場合は、Oracle Reports Services サーバー・マシンの ORACLE_HOME¥NET80¥ADMIN ディレクトリにある tnsnames.ora ファイルをオープンしてください。サービス・エントリをコピーするか書き留めておきます。

1. Web サーバー・マシン上で、テキスト・エディタを使用して (ORACLE_HOME¥NET80¥ADMIN ディレクトリにある) tnsnames.ora ファイルをオープンします。
2. 次の Oracle Reports Services サーバーのサービス・エントリを追加します。

```
repserver.world =(ADDRESS = (PROTOCOL = TCP) (Host =  
repserver_machine.mydomain) (Port = 1949))
```

各項目の内容は次のとおりです。

<code>repserver.world</code>	サーバー・インスタンスの名前。 <code>.world</code> は <code>sqlnet.ora</code> ファイルの <code>NAMES.DEFAULT_DOMAIN</code> の設定で指定されているドメインです。 <code>NAMES.DEFAULT_DOMAIN</code> 設定が <code>sqlnet.ora</code> ファイルで定義されていない場合は、サーバー・インスタンスの名前から <code>.world</code> を省略してください。
<code>repserver_machine.mydomain</code>	マシンのホスト名または IP アドレス。
<code>1949</code>	サーバーがリスニングするポート番号。

3.5.1.3 デフォルトで使用する Oracle Reports Services サーバーの設定 (オプション)

Windows NT プラットフォームおよび UNIX プラットフォームの両方において、オプションで、デフォルトで使用する Oracle Reports Services サーバーを設定できます。

3.5.1.3.1 Windows NT Windows NT での手順は、次のとおりです。

1. デスクトップで、「スタート」 「ファイル名を指定して実行」を選択します。
2. `regedit` と入力してレジストリ・エディタを表示します。
3. メニューで、`HKEY_LOCAL_MACHINE SOFTWARE ORACLE` を拡張します。
4. 「編集」 「新規作成」 「文字列」値を選択して次の環境変数を追加します。

`REPORTS60_REPORTS_SERVER`

`REPORTS60_REPORTS_SERVER` をダブル・クリックして、`repserver` の値を入力します。`repserver` は、構成中の Oracle Reports Services サーバーの名前 (Oracle Reports Services サーバーの TNS サービス・エントリ名) です。

3.5.1.3.2 UNIX UNIX の場合、`REPORTS60_REPORTS_SERVER` 環境変数に Oracle Reports Services サーバーの名前を設定します。

Web サーバー・マシン上の環境変数は、シェル・スクリプトを作成して設定することもできます。そうするには、次に説明するコマンドの入ったファイルを作成します。`repserver` は、構成中の Oracle Reports Services サーバーの名前 (Oracle Reports Services サーバーの TNS サービス・エントリ名) です。

```
setenv REPORTS60_REPORTS_SERVER repserver
```

3.6 環境変数による Oracle Reports Services サーバーの構成

この項では、環境変数によって Oracle Reports Services サーバーを構成し起動する方法について説明します。

3.6.1 Windows NT での環境変数による Oracle Reports Services サーバーの構成

Windows NT で環境変数によって Oracle Reports Services サーバーを構成する手順は、次の 2 つに分けられます。

1. 環境変数の設定（オプション）
2. Oracle Reports Services サーバーの起動

3.6.1.1 環境変数の設定（オプション）

オプションの環境変数を 2 つ設定できます。最初の環境変数により、要求されたレポートの場所を Oracle Reports Services サーバーに知らせます。REPORTS60_PATH 環境変数でレポートのソース・パスを設定します。2 番目の環境変数で tnsnames.ora ファイルの場所を設定します。

1. ソース・レポートのディレクトリ（たとえば /WEB_REPORTS）を作成します。
2. レポートを見つけるための次の REPORTS60_PATH 環境変数を設定します。

```
setenv REPORTS60_PATH /WEB_REPORTS
```

別の方法としては、Oracle Reports Services サーバーをインストールした後に、Oracle Reports Services サーバーの構成ファイルでソース・パスを設定できます。詳細は、[付録 C「Oracle Reports Services 構成パラメータ」](#)の SOURCEDIR パラメータを参照してください。

3. 次の TNS_ADMIN 環境変数を設定して、tnsnames.ora ファイルの場所をポイントします。

```
setenv TNS_ADMIN $ORACLE_HOME/NET80/ADMIN
```

3.6.1.2 Oracle Reports Services サーバーの起動

Oracle Reports Services サーバーを起動するには、次のようにします。

1. Oracle Reports Services サーバー・マシンのデスクトップで、「スタート」「設定」「コントロール パネル」を選択して、 (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 Reports Server [repserver]」(repserver は Oracle Reports Services サーバー・インスタンス名) を選択して「スタートアップ」をクリックし、「サービス」ウィンドウを表示します。
3. スタートアップ・ダイアログ・ボックスで、「ログオン」セクションの「アカウント」をクリックし、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、そのサーバーがそのユーザーとして実行されるよう指定されます。

PostScript またはプリンタに出力するときは、Oracle Reports Services サーバーのサービスを実行しているユーザーがデフォルトのプリンタに対するアクセス権を持っていることを確認してください。システム・アカウントは、通常、プリンタへのアクセス権を持っていません。

4. 「スタートアップの種類」を「自動」に設定します。
5. 「OK」をクリックします。
6. 「スタート」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 Oracle Reports Services サーバーが起動した日時が表示されます。Oracle Reports Services サーバーが起動しない場合は、[付録 G「トラブルシューティング」](#)を参照してください。

Oracle Reports Services サーバーを初めて起動したとき、Oracle Reports Services サーバーの構成ファイル (たとえば repserver.ora) が ORACLE_HOME\REPORT60\SERVER ディレクトリに作成されます。Oracle Reports Services サーバーのキャッシュの設定はデフォルトで設定されます。構成ファイルを変更して、キャッシュ・ディレクトリを変更したり、レポートのソース・パスを設定できます。構成ファイルを変更したら、Oracle Reports Services サーバーを停止して再起動し、変更内容を有効にします。

3.6.2 UNIX での環境変数による Oracle Reports Services サーバーの構成

UNIX で環境変数によって Oracle Reports Services サーバーを構成する手順は、次の 2 つに分けられます。

1. 環境変数の設定 (オプション)
2. Oracle Reports Services サーバーの起動

3.6.2.1 環境変数の設定 (オプション)

REPORTS60_PATH および TNS_ADMIN という 2 つの環境変数を設定できます。

REPORTS60_PATH は、Oracle Reports Services サーバーのソース・ファイル (RDF、TDF、

PLL など) の検索パスです。TNS_ADMIN では、tnsnames.ora および sqlnet.ora のデフォルトの場所が上書きされます。これらを設定するには、次のようにします。

1. ソース・レポートのディレクトリ (たとえば /WEB_REPORTS) を作成します。
2. レポートを見つけるための REPORTS60_PATH 環境変数を設定します。たとえば、C シェル構文を使用すると、次のようになります。

```
setenv REPORTS60_PATH /WEB_REPORTS
```

別の方法としては、Oracle Reports Services サーバーをインストールした後に、SOURCEDIR パラメータを使用してソース・パスを設定できます。詳細は、[付録 C 「Oracle Reports Services 構成パラメータ」](#)を参照してください。

3. TNS_ADMIN 環境変数を設定して、tnsnames.ora ファイルの場所をポイントします。たとえば、C シェル構文を使用すると、次のようになります。

```
setenv TNS_ADMIN $ORACLE_HOME/NET80/ADMIN
```

3.6.2.2 UNIX 上での Oracle Reports Services サーバーの起動

UNIX 上で Oracle Reports Services サーバーを起動するには、次のステップに従います。

1. \$ORACLE_HOME/BIN ディレクトリから次のコマンドラインを実行して、Oracle Reports Services サーバーをフォアグラウンドで実行します。

```
rwmts60 name=repserver
```

あるいは、次のコマンドを実行して、Oracle Reports Services サーバーをバックグラウンドで実行します。

```
rwmts60 name=repserver &
```

2. \$ORACLE_HOME/BIN ディレクトリから次のコマンドラインを実行して、Oracle Reports Services サーバーが稼働していることを確認します。

```
rwrvq60 server=repserver
```

Oracle Reports Services サーバーのステータスの列 (NAME、OWNER、DEST など) が表示されます。ただし今はジョブが実行されていないのでステータス情報は表示されません。

Postscript あるいはプリンタに出力するときは、(`$ORACLE_HOME/guicommon6/tk60/ADMIN` ディレクトリにある) `uiprint.txt` ファイルでプリンタを構成する必要があります。

3.7 環境変数

環境変数は、Oracle Reports Services サーバーの動作をコントロールまたはカスタマイズするために使用される構成パラメータです。環境変数は、Windows NT の場合はコマンドラインを使用して、UNIX の場合はシェル・スクリプトを使用して設定できます。

変数	説明
REPORTS60_COOKIE_EXPIRE	cookie の期限を分単位で決定します。デフォルト値は 30 です。 cookie は、ユーザーが保護された Oracle Reports Services サーバーにログインして、レポート・リクエストを実行する際に、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に保存します。ログインに成功すると、暗号化された cookie がブラウザに送られます。cookie が期限切れになると、それに続くリクエスト（保護された Oracle Reports Services サーバーに送られるリクエスト）のユーザーは、再度認証される必要があります。
REPORTS60_DB_AUTH	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。デフォルト値は dbauth.htm です。
REPORTS60_ENCRYPTION_KEY	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します。暗号化キーは、どの文字列でも設定できます。デフォルト値は reports6.0 です。
REPORTS60_REPORTS_SERVER	Web リクエスト用のデフォルトの Oracle Reports Services サーバーを指定します。このパラメータが設定されている場合、レポート・リクエストでデフォルトのサーバーを使用して処理を行う際に SERVER コマンドライン引数を使用して指定する操作を省略できます。また、SERVER コマンドライン引数を使用してデフォルトを書き換えることもできます。
REPORTS60_SSLPORT	SSL を使用していて 443 以外のポート番号を用いる場合、この変数を使用して別のポート番号を指定できます。デフォルト値は 443 です。
REPORTS60_SYS_AUTH	保護された Oracle Reports Services サーバーに対するレポート・リクエストをユーザーが実行する際に、ユーザー名とパスワードを認証するために使用する認証テンプレートを指定します。デフォルト値は sysauth.htm です。

3.8 Web ブラウザからのレポート・リクエストの実行

Web ブラウザからレポート・リクエストを実行するには、次のようにします。

1. Oracle Reports Services サーバーが正しく構成されていることを確認します。Web ブラウザで、次のリクエストを行います。

```
http://your_webserver/cgi-bin/rwcgi60.exe?report=your_report.rdf+
userid=username/passwordmy_db+desformat=html+destype=cache
```

各項目の内容は次のとおりです。

username/password	有効なデータベース・ログイン名に置き換えます。
my_db	事前に Oracle Reports Services サーバーのために作成した tnsnames.ora ファイルのエントリに置き換えます。 3.5.1.2 項「Oracle Reports Services サーバーのサービス・エントリの作成」 を参照してください。

SERVER コマンドライン引数がこのリクエストにない点に注意してください。Web サーバー・マシン上で REPORTS60_REPORTS_SERVER 環境変数を設定している場合は必要ありません。

レポートが実行されない場合、またはエラー・メッセージが表示される場合は、[付録 G「トラブルシューティング」](#)を参照してください。

2. リクエストのステータスを表示します（オプション）。
 - Windows NT の場合は、Oracle Reports Services Queue Manager を起動して、repserver のキューを表示します。詳細は Oracle Reports Services Queue Manager のオンライン・ヘルプを参照してください。
 - UNIX の場合は、次のコマンドを実行します。

```
rwrq60 server=repserver showjobs=current
```

3.8.1 他のステップ

さらに、次のステップを行うことができます。

1. （オプション）Oracle Reports Services サーバーを調整してパフォーマンスの最適化やアクセス制御などの追加機能の実現を図ります。この作業を行うと、[3.8 項「Web ブラウザからのレポート・リクエストの実行」](#)に示したパラメータをすべて表示する必要がなくなり、ユーザー名およびパスワードを保護できます。
2. ユーザーがレポートを利用できるようにします。実行リクエストを指定し、それらをユーザーが利用できるようにする方法の詳細は、[第 4 章「レポート・リクエストの実行」](#)を参照してください。

3.9 Oracle Reports Services サーバーの構成変更（オプション）

Oracle Reports Services サーバーを初めて起動したときは、Oracle Reports Services サーバーはデフォルトの構成設定になっています（最大、最小エンジンなど）。ある時点で、Oracle Reports Services サーバーの構成を変更し、パフォーマンスの調整、監視制御の設定、そして追加機能の実装などを行う必要が生じることがあります。

- データベースをジョブ・キュー情報に従って更新する場合は、[3.9.1 項「ジョブのキュー・アクティビティによるデータベースの更新」](#)を参照してください。
- レポートへのユーザー・アクセスを制御する方法の詳細は、[第 5 章「Oracle Portal を使用した Oracle Reports Services セキュリティ」](#)を参照してください。
- Oracle Reports Services サーバーの構成を変更して負荷を分散する方法の詳細は、[第 6 章「Oracle Reports Services サーバーのクラスタの構成」](#)を参照してください。
- ユーザーの要望に応じてレポートを実行時に修正する方法の詳細は、[第 7 章「実行時のレポートのカスタマイズ」](#)を参照してください。
- Oracle Reports Services サーバーの構成ファイルを修正する場合は、[付録 C「Oracle Reports Services 構成パラメータ」](#)の有効な構成パラメータに関する情報を参照してください。
- 環境変数を変更する方法の詳細は、[付録 D「環境変数」](#)を参照してください。

3.9.1 ジョブのキュー・アクティビティによるデータベースの更新

ジョブが実行されるたびに Oracle Reports Services サーバーのキュー・アクティビティのスナップショットを入手するように、データベースをセットアップできます。Oracle Reports Services サーバーを起動すると、データベースへの接続が確立されます。デフォルトでは、Oracle Reports Services サーバーは API をコールして、サーバーの再起動時にキュー情報を削除し、データベース・テーブルのキュー情報を更新します。

`rw_server.sql` スクリプトの API のソースを編集して、デフォルトを上書きすることでもできます（Oracle Reports Services サーバーの再起動時にキューが削除されないようにする、など）。プロシージャのプロトタイプ（プロシージャ名と、予想されるパラメータ）は、編集しないでください。

スクリプトの内容を変更したユーザーのアカウントでスクリプトを実行した後、Oracle Reports Services サーバーを停止して再起動し、変更内容を有効にします。

3.9.1.1 Oracle Reports Services サーバー・マシンでの作業

Oracle Reports Services サーバー・マシン上でジョブのキュー・アクティビティによってデータベースを更新するには、次のようにします。

1. テキスト・エディタで（ORACLE_HOME\REPORT60\SERVER ディレクトリにある）repserver.ora 構成ファイルをオープンします。
2. rw_server.sql スクリプトをコマンドラインから実行するためには、repserver_schema に少なくとも CREATE TABLE 権限および CREATE PACKAGE 権限を付与する必要があります。コマンドライン・プロンプトで次のように入力します。

```
cd C:\ORACLE_HOME\REPORT60\SQL <RETURN>
plus80 username/password@my_db <RETURN>
@rw_server.sql <RETURN>
quit <RETURN>
```

3. 次の構成パラメータを追加します。repserver_schema/password@my_db は、指定した Oracle Reports Services サーバーのキュー・アクティビティのスナップショットを入手するための、データベースのスキーマへの接続文字列です。この場合、repserver_schema が repserver キュー・アクティビティに対するスキーマです。

```
REPOSITORYCONN="repserver_schema/password@my_db"
```

いくつかの Oracle Reports Services サーバーからのスナップショットを入手するときは、データベース内に、スナップショットを入手する Oracle Reports Services サーバーごとに別個のスキーマを作成することをお勧めします。そうすれば、Oracle Reports Services サーバーの再起動時にキュー・アクティビティ・データを失うことがなくなります。

4. Oracle Reports Services サーバーを停止してから再起動して、構成ファイルに対して行った変更内容を受け入れます。Oracle Reports Services サーバーは起動時にデータベースに接続します。

注意： Oracle Reports Services サーバーを再起動すると、デフォルトでは、データベースのキュー・アクティビティは削除されます。デフォルトは、API を編集して上書きできます。

レポート・リクエストの実行

この章では、レポート・リクエストを指定する様々な方法について説明します。次の項目について説明します。

- レポート・リクエスト・メソッド
- 重複ジョブの検出
- マップ・ファイルを使用してリクエストの実行を簡単にする方法
- URL 実行リクエストの指定
- 自動実行のためのレポート・リクエストのスケジューリング

4.1 レポート・リクエスト・メソッド

レポート・リクエストは、次に示す様々なリクエスト・メソッドを使用して実行できます。

- RWCLI60 コマンドラインを使用すると、コマンドラインのプロンプトからレポート・リクエストを実行できます。RWCLI60 は、コマンドラインを解釈し、指定された Oracle Reports Services サーバーに送る実行形式のファイルです。このファイルでは、Oracle Reports Services Runtime の実行ファイル (RWRUN60) と類似のコマンドラインが使用されます。RWCLI60 コマンドライン・リクエストは、非 Web アーキテクチャを使用して行われます。典型的なコマンドライン・リクエストは次のようになります。

```
RWCLI60 REPORT=my_report.rdf USERID=username/password@my_db SERVER=repserver  
DESTYPE=HTML DESFORMAT=cache
```

有効な RWCLI60 コマンドライン引数のリストは、[付録 B「RWCLI60 コマンドライン引数」](#)を参照してください。

- URL 構文を使用すると、Web ブラウザからレポート・リクエストを実行できます。CGI およびサプレットは、URL 構文を Oracle Reports Services によって処理される RWCLI60 コマンドライン・リクエストに変換します。レポートの処理が終了すると、出力は Web サーバーが知っているロケーションにある HTML ファイルまたは PDF ファイルに送信され、そこから Web ブラウザに戻されます。ユーザーのブラウザから

レポート・リクエストを行うのに必要な URL 構文をユーザーに提供するか、または URL 構文をハイパーリンクとして Web サイトに追加することもできます。この章の次の項ではこのメソッドについて詳しく説明します。

- Oracle Portal コンポーネントを使用すると、リンクを Oracle Portal コンポーネントとして Oracle Portal サイトに追加できます。このリンクは、レポート・リクエストについての情報の入ったパッケージ・プロシージャをポイントします。Oracle Reports Services のシステム管理者は、Oracle Portal ウィザードを使用してパッケージ・プロシージャを作成し、Web を使用したレポートの公開をより便利で安全な方法で行えるようにします。Oracle Portal サイト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のみでレポートを実行できます。システム管理者は直接ウィザードからレポートを実行できます。詳細は、[第 5 章「Oracle Portal を使用した Oracle Reports Services セキュリティ」](#)を参照してください。
- ActiveX コントロールは、Oracle Reports Services を業界標準の ActiveX テクノロジーを通じて公開しているので、ActiveX コンテナからのレポートの実行が可能です。Oracle Reports Services Launcher は ActiveX コンテナの一例です。詳細は ActiveX および Oracle Reports Services Launcher のオンライン・ヘルプを参照してください。
- SRW.RUN_REPORT は、Oracle Reports Services Runtime のコマンドを実行する PL/SQL のパッケージ・プロシージャです。SRW.RUN_REPORT コマンドラインを指定するときは、SERVER 引数に Oracle Reports Services の TNS サービス・エントリ名を設定して、SRW.RUN_REPORT コマンドに RWCLI60 コマンドを実行したときと同様の動作をさせます。詳細は Oracle Reports Services Builder のオンライン・ヘルプを参照してください。

4.2 重複ジョブの検出

DESTYPE を cache に設定して、あるいは、TOLERANCE を任意の長さの分 (0 分以上) に設定してレポートを実行すると、レポート出力のコピーが Oracle Reports Services のキャッシュに保存されます。その後で、同じレポートが実行されると (つまり、完全に同じコマンドライン引数を使用して) 現行リクエストは重複ジョブとして認識されます。Oracle Reports Services は、指定された許容時間 (たとえば TOLERANCE=10) 内にリクエストが行われた場合は同じレポートを再度実行するのではなく、キャッシュ内の出力を再利用します。前回のジョブの終了時、あるいはそれがすでに終了している場合も、後続のレポートに対してキャッシュ内の出力が使用されます。ジョブの 1 つが取り消された場合 (たとえば Oracle Reports Services Queue Manager から取り消された場合) 他 のレポートはランタイム・エンジンによって通常どおりに実行されます。

DESTYPE と TOLERANCE のコマンドライン引数の詳細は、[付録 B「RWCLI60 コマンドライン引数」](#)を参照してください。

4.2.1 使用上の注意

次に示す使用上の注意をよく読んでください。

- 次のコマンドライン引数を比較して重複ジョブが検出されます。REPORT、USERID、DESFORMAT、PARAMFORM、CURRENCY、THOUSANDS、DECIMAL、PAGESIZE、ORIENTATION、MODE およびすべてのユーザー・パラメータ。
- レポートの出力を複数の宛先に配布するときは、サーバー上でレポートを一度実行してから、宛先と許容時間を変えて、同じコマンドを同じサーバーに対して発行します。Oracle Reports Services によって重複ジョブが検出されて、キャッシュ内のファイルが新しい宛先に再配布されます。
- 重複ジョブの検出は、繰り返されたジョブの各インスタンスに対して別々に行われます。
- Oracle Reports Services Queue Manager を使用して、あるいは手動で Oracle Reports Services の構成ファイルの CACHESIZE パラメータを設定することによってキャッシュ・サイズを設定できます。Oracle Reports Services は、前に使用されてから最も長い時間が経過しているキャッシュ・ファイルを削除して、キャッシュ・ファイルの合計サイズをこの制限値以下に抑えようとします。さらに、Oracle Reports Services 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、Oracle Reports Services Queue Manager のオンライン・ヘルプまたは[付録 C「Oracle Reports Services 構成パラメータ」](#)を参照してください。

- 同じジョブが送信されたときにレポートが処理中であった場合は、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、Oracle Reports Services は現在実行されているジョブの出力を再利用します。job_1 が Oracle Reports Services のエンジンのいずれかで現在実行中で、他のユーザーが job_1 と同一の job_2 を送信したとします。Oracle Reports Services は、job_1 からの出力を job_2 に対して使用します。この場合、job_2 はエンジンに送られて実行されるわけではないので、job_2 の処理速度はかなり速くなります。

4.3 キー・マップ・ファイルの使用方法

URL 構文をユーザーに提供するとき、あるいは URL 構文をハイパーリンクとして任意の Web サイトに追加するときは、キー・マップ・ファイルを使用して、URL リクエストにあるパラメータを単純化したり隠したりできます。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 を変更しないで、URL の実行構成のマッピングをし直す場合
- 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）を隠蔽する場合
- 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

レポートを Web サイトで公開するためのより便利で安全な方法として、Oracle Portal コンポーネントを作成する方法があります。詳細は、[第 5 章「Oracle Portal を使用した Oracle Reports Services セキュリティ」](#)を参照してください。

マップ・ファイルは URL パラメータを受け取って、レポート・リクエストを規定するコマンドライン引数にマップします。たとえば、URL リクエスト構文の中の 1 つの引数を、レポートの実行に必要なすべてのコマンドライン引数にマップできます。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引数をユーザーから隠蔽できます。

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。

次のコマンドラインを送信します。

```
http://your_webserver/cgi-bin/rwcgi60.exe?key+par1+par2+parN
```

ここで、キー・マップ・ファイルには次の行が含まれています。

```
KEY: module=myreport deptno=%1 myparam=%2 %*
```

すると、次のコマンドライン・リクエストに相当するリクエストが生成されます。

```
RWCL160 module=myreport deptno=par1 myparam=par2 parN
```

4.3.1 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 2 つの条件のいずれかが満足されたときに使用可能となります。

- Web サーバー上の REPORTS60_CGIMAP (CGI) 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場合。詳細は、[付録 D「環境変数」](#)を参照してください。
- 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・マシン上の ORACLE_HOME¥REPORT60 ディレクトリにある場合。

使用上の注意

キー・マッピングを行う場合は、次に示す使用上の注意をよく読んでください。

- キー・マッピングが使用可能になっていると、すべての RWCGI60 の URL は最初の引数がキーであるかのように処理されます。キー・マップ・ファイルはこのキーのみを検索します。キーが見つかったら、その定義済の値が Oracle Reports Services のコマンドラインに代入されます。見つからない場合は、エラーが生成されます。
- URL を HTML フォームを使用して提出すると、キーは隠蔽型の入力としてコード化されます。

4.3.2 URL パラメータのマッピング

この項では、キー・マッピング・エントリをキー・マップ・ファイルに追加する方法について説明します。

Web サーバー・マシン上：

1. テキスト・エディタで ORACLE_HOME\REPORT60 ディレクトリにある cgicmd.dat (CGI) ファイルをオープンします。

ヒント：使用中のマップ・ファイルの名前を確認するには、Web ブラウザで「http://your_webserver/cgi-bin/rwcgi60.exe/showmap?」と入力します。

2. キー・マッピング・エントリを追加します。基本的なキー・マッピング・エントリは次のようになります。key1 はキーの名前です。

```
key1: REPORT=your_report.rdf USERID=user_name/password@mydb DESFORMAT=html
SERVER=repserver DESTYPE=cache
```

ファイル自体で記述されている特殊なパラメータを除いて、コマンドライン引数は RWCLI60 の構文規則に従います。RWCLI60 のコマンドライン引数の詳細は、[付録 B 「RWCLI60 コマンドライン引数」](#)を参照してください。

REPORTS60_REPORTS_SERVER 環境変数を設定してリクエストをデフォルトのサーバーに送っている場合は、SERVER コマンドライン引数を省略できます。詳細は、[付録 D 「環境変数」](#)を参照してください。

3. Web ページでハイパーリンクを更新します。[4.4.2 項「Web ブラウザからのレポート・リクエストの指定」](#)を参照してください。

4.4 レポート・リクエストの指定

次のことを行うことによってレポートを指定できます。


- レポートの構築
- Web ブラウザからのレポート・リクエストの指定
- 自動実行のためのレポートのスケジューリング

4.4.1 レポートの構築

レポートを構築するには、次のようにします。

1. 使用中の Oracle Reports Services が搭載されているマシン上で、そのパスを使用してレポートを保存するためのレポートのソース・ディレクトリ（`C:\¥WEB_REPORTS` など）を作成します。このディレクトリは、必ず Oracle Reports Services の構成ファイルの `SOURCEDIR` パラメータで設定してください。付録 C「Oracle Reports Services 構成パラメータ」を参照してください。

レポートのソース・パスは `REPORTS60_PATH` 環境変数でも設定できます。詳細は、付録 D「環境変数」を参照してください。

Oracle Reports Services Builder を起動してレポートを作成します。このレポートは RDF ファイルまたは REP ファイルとして保存できます。このレポート定義ファイルは、必ず Oracle Reports Services のマシン上でレポート・ソース・ディレクトリ（`C:\¥WEB_REPORTS` など）にコピーしてください。レポートの作成の詳細は、『レポート作成ガイド』または Oracle Reports Services Builder のオンライン・ヘルプを参照してください。Oracle Reports Services Builder のオンライン・ヘルプにアクセスするには、 のアイコンをクリックして次のステップに従います。



1. このタスクについてのオンライン・ヘルプにアクセスするには、「ヘルプ」「Report Builder ヘルプ・トピック」を選択します。
2. 「キーワード」ページで次のように入力します。
レポート, 作成
3. 次に、「表示」をクリックして次のヘルプ・トピックを表示します。
標準レポートの作成

2. ユーザーがこのレポートを使用できるようにします。詳細は、4.4.2 項「Web ブラウザからのレポート・リクエストの指定」を参照してください。

4.4.2 Web ブラウザからのレポート・リクエストの指定

レポート・リクエストに必要な URL 構文をユーザーに提供できます。あるいは、URL 構文をハイパーリンクとして Web ページに追加することもできます。

レポートを Web サイトで公開するためのより便利で安全な方法として、Oracle Portal コンポーネントを作成する方法があります。詳細は、[第 5 章「Oracle Portal を使用した Oracle Reports Services セキュリティ」](#)を参照してください。

URL 構文は、次の形式で提示できます。

- 次のような完全な URL リクエスト

```
http://your_webserver/cgi-bin/rwcgi60.exe?report=your_report.rdf  
+userid=user_name/password@mydb+server=repserver+desformat=html  
+destype=cache
```

他のコマンドライン引数が必要な場合、有効な RWCLI60 のコマンドライン引数のリストが掲載されている[付録 B「RWCLI60 コマンドライン引数」](#)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次のような URL リクエスト

```
http://your_webserver/cgi-bin/rwcgi60.exe?report=key1
```

REPORTS60_REPORTS_SERVER 環境変数を設定してリクエストをデフォルトのサーバーに送っている場合は、SERVER コマンドライン引数を省略できます。詳細は、[付録 D「環境変数」](#)を参照してください。

URL 構文をハイパーリンクとして Web ページに追加する手順は次のとおりです。

1. リクエストをハイパーリンクとして Web ページに追加します。構文は次のようになります。

```
<A HREF="http://my_webserver/cgi-bin/rwcgi60.exe?key1">My report</A>
```

2. ユーザーにレポート・リクエストを公開する Web サイトの URL を提供します。ユーザーはこのリンクをクリックしてレポートを実行します。

レポートが実行されない場合、あるいは、Web ブラウザで期待どおりに表示されない場合、詳細は[付録 G「トラブルシューティング」](#)を参照してください。

4.4.3 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、Oracle Reports Services Queue Manager または Oracle Portal からレポートを自動的に実行することもできます。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、Oracle Reports Services 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 Oracle Portal サイトで Oracle Portal コンポーネントとして公開するときは、これらのレポート・リクエストは、自動的に実行されるようにスケジューリングして、出力されたレポートをサイト上の指定のフォルダに送信できます。詳細は、[第 5 章「Oracle Portal を使用した Oracle Reports Services セキュリティ」](#)を参照してください。

Oracle Portal を使用した Oracle Reports Services セキュリティ

Oracle Reports Services は、アクセス制限された Oracle Reports Services サーバーやプリンタ上で必要な権限（アクセス制御）を持つユーザーのみレポートを実行するようセキュリティ・チェックを行うため、Oracle Portal を使用します。アクセス制御によって特定される事項は、次のとおりです。

- レポート定義ファイル（RDF）、Oracle Reports Services サーバーおよびプリンタに対する制限の内容
- 要求されたレポートを、アクセス制限された Oracle Reports Services サーバー上で実行し、アクセス制限されたプリンタに出力する権限を持つユーザー
- RDF、Oracle Reports Services サーバーおよびプリンタを実行できる期間
- 実行時にユーザーが利用できるレポート・リクエスト・オプション（すなわち、必須および任意パラメータ）を制限して、レポート出力を配布する方法。ユーザーが利用できる Oracle Reports Services サーバーおよびプリンタの指定なども含まれます。

Oracle Portal には、パッケージ・プロシージャとしての RDF（レポートの実行方法）に関する情報が格納されます。レポートを実行するには、リクエスト受け用のアクセス制限された Oracle Reports Services サーバー、およびレポート出力の印刷に使用するすべてのプリンタに関するアクセス制御情報も、Oracle Portal に格納する必要があります。Oracle Portal にアクセス制御情報を追加するには、Oracle Reports Services セキュリティ・ウィザードを使用します。

Web 上でユーザーがレポート・リクエストを実行できるようにするには、次のことを行います。

- レポートのパッケージ・プロシージャを指す Oracle Portal コンテンツ領域に、リンクを Oracle Portal コンポーネントとして追加します。
- 自動的にリクエストが実行され、レポート出力が Oracle Portal コンテンツ領域に配置されてユーザーが参照できるように、スケジュールを設定します。
- 標準 URL 構文をハイパーリンクとして Web コンテンツ領域に追加します。

5.1 概要

管理が必要なセキュリティには、次の 2 つのレベルがあります。

- データベース・レベルのセキュリティ
- アプリケーション・レベルのセキュリティ

データベース・レベルのセキュリティは、データベース内のデータにアクセスできるユーザーおよびロールを定義します。このセキュリティについての権限付与は DBA が行います。このデータベース・レベルのセキュリティは、レポート環境の構成時に確定させて、適切に設定する必要があります。

さらに詳細なレポート環境を定義するには、アプリケーション・レベルのセキュリティを適切に設定して、ユーザーまたはグループがどのレポート・リクエストを生成できるかを指定します。そのような許可されたユーザーまたはグループのみが特定レポートを生成できるようにするには、このアプリケーション・レベルのセキュリティが非常に重要です。

Oracle Reports Services 6i では、Oracle Portal リリース 3.0（以前 Oracle WebDB と呼ばれていたもの）を使用したそのままのかたちの実装によるレポートの管理とセキュリティのために、オープン・インフラストラクチャが導入されています。Oracle Portal のすべてのコンテンツは Oracle データベースのリポジトリに格納されているため、アクセス制御データの入力、バックアップおよび検索が容易にできます。この章では、Oracle Portal リリース 3.0 を使用して Oracle Reports Services 6i を配置するための、セキュリティ・ポリシーの構成と設定について説明します。すべての Oracle Portal ユーザーは軽量ユーザーであり、アプリケーション・レベルのセキュリティ権限を持っています。Oracle Portal ユーザーには、データベース・レベルのセキュリティ権限はありません。

Oracle Reports Services をそのままのかたちで実装した場合、Oracle Portal には、既存の Oracle Reports Services サーバーで利用できるアプリケーション・レベルのアクセス制御データが格納されます。オープンな配置モデルにより、Oracle Portal コンテンツ領域またはカスタム・ポータルどちらを使用しても、レポート生成にアクセスできるようになっています。どちらの場合でも、Oracle Portal のリポジトリ内に格納されたセキュリティ・アクセス制御データは、エンド・ユーザーの認可のために使用されます。

Oracle Portal コンテンツ領域の外部に Oracle Reports Services を配置する場合、Oracle Reports Services サーバーを介してレポート・リクエストを渡すことによって、Oracle Portal に格納されたセキュリティ・アクセス制御データを使用することもできます。次のどちらかを使用して実行できます。

- RWCGI60（URL を介して渡された Oracle Reports Services CGI 実行可能ファイル）またはサブレット
- RWCLI60（Oracle Reports Services サーバーに対する Oracle Reports Services コマンドライン・インタフェース）

Oracle Portal とは別のセキュリティ・モデルを選択した場合は、提供されているオープン C API を使用して、ユーザー所有のアクセス制御サーバーへのカスタム・リンクを作成できます。API を記述することにより、他のセキュリティ・サーバーとの連絡ができます。（たとえば、Oracle データベースに記述されているカスタム・セキュリティ・スキーマを参照する

場合)。そのためには、次のステップで詳細に説明しているように、C インタフェースを記述する必要があります。説明は、セキュリティの実装がすでに作成されていることを前提としています。まだ作成していない場合は、まずそれを作成してください。Windows 環境と UNIX 環境で説明が分かれています。

5.1.1 Windows 環境への Oracle Reports Services 6i 用のセキュリティ DLL の作成

Windows 環境に Oracle Reports Services 6i 用のセキュリティ DLL を作成するには、次のステップが必要です。

1. Win32 C コンパイラ（たとえば、Visual C++）を使用して、新規プロジェクトを作成し、DLL を作成するように指定します。
2. RWKSS.C というファイルを作成し、RWKSS.H ヘッダー・ファイルをインクルードします。このヘッダー・ファイルは、%ORACLE_HOME%\report60\server\security ディレクトリにあります。
3. RWKSS.H ヘッダー・ファイルで指定された機能（Oracle Reports Services サーバーからコールされる）をインプリメントして、セキュリティ・リポジトリに対してセキュリティ・チェックを実行します。
4. RWK60.DEF ファイルに DLL とリンクを作成します。RWK60.DEF ファイルにはエクスポートされた機能のリストが含まれます。このファイルは、%ORACLE_HOME%\report60\server\security ディレクトリにあります。DLL には 'rwk60.dll' というファイル名を付けます。
5. %ORACLE_HOME%\bin ディレクトリにある既存の RWK60.DLL を改名し、そのディレクトリにユーザーの DLL をコピーします。
6. Oracle Reports Services サーバーを再起動します。

5.1.2 UNIX 環境への Oracle Reports Services 6i 用のセキュリティ・ライブラリの作成

UNIX 環境に Oracle Reports Services 6i 用のセキュリティ・ライブラリを作成するには、次のステップが必要です。

1. rwkss.c というファイルを作成し、rwkss.h ヘッダー・ファイルをインクルードします。このヘッダー・ファイルは、\$ORACLE_HOME/reports60/pub ディレクトリにあります。
2. RWKSS.H ヘッダー・ファイルで指定された機能（Oracle Reports Services サーバーからコールされる）をインプリメントして、セキュリティ・リポジトリに対してセキュリティ・チェックを実行します。

- 動的ライブラリをコンパイルして作成します。次に例を示します。

```
cc -c rwkss.c
ld -dy -G -o rwk60.so rwkss.o
```

- 次のように入力して、ライブラリが動的になっていることをチェックします。

```
file rwk60.so
```

ライブラリは動的であるという応答を確認してください。

- `$ORACLE_HOME/bin` ディレクトリにある既存の `rwk60.so` を改名した後、そのディレクトリにユーザーの新しい動的ライブラリを置きます。
- Oracle Reports Services サーバーを再起動します。

5.2 データベース・レベルのセキュリティ

データベース・レベルのセキュリティは、指定されたデータベース内のデータに対するアクセス権を持っているかどうかを判断するものです。ユーザー名とパスワードは、キー・マップ・ファイル (`cgicmd.dat`) に格納できます。または、特定のユーザー ID とパスワードを入力するようにプロンプトを表示することもできます。

ユーザー名とパスワードがキー・マップ・ファイルにハードコードされている場合（または URL の一部として入力されている場合）を除き、Oracle Reports Services にアクセスするユーザーはすべて、認証のために識別が要求されます。HTTP リリース 1.0 プロトコルではサーバーに対する各コールは事実上他のすべてのコールから独立しているため、レポート・リクエストごとにユーザー認証が必要になります。

この問題を解決して認証を 1 回にするために、レポートでは、現行セッションのブラウザ内で必要とされる認証情報を格納するクライアント側 cookie が使用されます。一度認証されると暗号化された cookie がブラウザに作成されるため、リクエストごとに再度認証されることなく複数のレポート・ジョブを送信できます。

注意： 所定のレポートの送信に対して再度認証する必要がある場合は、SHOWAUTH および AUTHTYPE コマンドライン引数を使用するか、またはキー・マップ・ファイルの各レポート・エントリに %D を含めます (%D を使用すると、レポートがコールされるたびにユーザー名とパスワードの再入力が必要になります)。

所定の Web アプリケーション内では、Oracle データベース（または、ODBC データ・ソース）の複数インスタンスに対して実行されるレポートが頻繁にアクセスされます。必要な認証回数を最小限（各サーバー当たり 1 回）にするために、暗号化された cookie が作成されます。cookie には多くのデータベース・インスタンスのデータベース認証情報が含まれているため、Oracle データベースの複数インスタンスへの接続が可能です。

データベース接続情報は USERID パラメータを指定できます。たとえば、
USERID=<\$username> を Net8 データベースの別名とともに使用してレポートを送信すると、接続情報は各参照データベースのインスタンスのブラウザに作成されます。たとえば、次のキー・マップ・ファイル・エントリの例では、ブラウザに作成された暗号化 cookie を介して、2 つの異なるデータベースに対して認証されることになります（この cookie は ORCL および PROD の両方で使用されます）。

```
Rep1: report=Rep1.rdf userid=$username@ORCL destype=CACHE desformat=HTML
```

```
Rep2: report=Rep1.ref userid=$username@PROD destype=CACHE desformat=HTML
```

その後のリクエストでは、ユーザー名とパスワードが適切な cookie から取り出され、該当するデータベースに対する認証に使用されます。接続文字列がコマンドラインに定義されていない（つまり、ユーザー ID がコマンドラインに指定されていない）場合、Oracle Reports Services CGI 実行可能ファイルでは、接続に成功した最後のデータベース接続文字列が使用されます。

cookie はブラウザ・セッションを閉じたときに削除されますが、所定のセッション内で cookie の存続期間を制限することが重要な場合もあります。たとえば、ログインした後に昼食に出かけて、長時間ブラウザ・セッションを開いたままにしたとします。このようなセキュリティ上の問題を制御するために、管理者は、CGI またはサプレットに REPORTS60_COOKIE_EXPIRE 環境変数を定義できます。Oracle Reports Services 実行可能ファイルは、クライアントからジョブ・リクエストを受け取ったときに、cookie に保存されている時間を現在のシステム時間と比較します。その時間が環境変数に定義した分単位の時間（たとえば、30 分）よりも長い場合、cookie が拒否され、認証のために再度ユーザーの識別が要求されます。次の表に、データベース・ユーザー認証に関する環境変数を示します。

表 5-1 ユーザー認証に関する環境変数

REPORTS60_COOKIE_EXPIRE	<p>Oracle Reports Services CGI またはサプレットに対して、cookie の有効期間を分単位で設定します。デフォルト値は 30 です。</p> <p>保護された Oracle Reports Services サーバーにログインしてレポート・リクエストを実行する際、クライアント側の暗号化されたユーザー名とパスワードが cookie に保存されます。正常にログインすると、暗号化された cookie がブラウザに送られます。cookie が期限切れになると、その後のレポート・リクエスト（保護されている Oracle Reports Services サーバーに送られるリクエスト）を実行するには、再度認証が必要になります。</p>
-------------------------	--

REPORTS60_DB_AUTH	<p>データベースにログインするためのデータベース認証ウィンドウ・テンプレート名を設定する HTML ファイルを指定します。ただし、Windows NT の場合、このファイルは次のディレクトリに格納されるため、完全なパスを指定する必要はありません。</p> <p>%ORACLE_HOME%\%REPORT60</p> <p>UNIX の場合は、次のディレクトリに格納されます。</p> <p>\$ORACLE_HOME/reports60</p> <p>デフォルト値は dbauth.htm です。</p> <p>REPORTS60_DB_AUTH 環境変数を使用すると、データベース認証の HTML フォームをカスタマイズできます。</p>
REPORTS60_ENCRYPTION_KEY	<p>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します。暗号化キーには任意の文字列を設定できます。デフォルト値は reports6.0 です。</p>

5.3 アプリケーション・レベルのセキュリティ

アプリケーション・レベルのセキュリティは、特定のレポートを実行する際に必要となる、リソースへの適切なアクセス権を確認するために必要です。これは、データベースのデータへのアクセス権（データベース・レベルのセキュリティ）のことではありません。アプリケーション・レベルのセキュリティの認証方式は、次の基準が必要です。

- 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
- レポートを実行または印刷するためにアクセスできるサーバーまたはプリンタ、およびレポートにアクセスできる期間
- 特定ユーザーが特定レポートで使用するパラメータ

5.4 Oracle Portal との統合

Oracle Portal は、ブラウザ・ベースの Web コンテンツの公開と開発のための解決策です。これにより、エンド・ユーザーおよび開発者は即座に情報を公開でき、データ駆動型の部門別のポータルを作成できます。

Oracle Portal リリース 3.0 は Oracle Reports Services と緊密に統合しているため、強力で安全なレポート環境を作成できます。Oracle Reports Services セキュリティのために、Oracle

Portal には新しいウィザードが追加されています。許可されたユーザーは、このウィザードにより、レポート、Oracle Reports Services サーバー、プリンタ、出力形式およびレポート・パラメータに対するアクセス制御を定義できます。

Oracle Portal のコンテンツ領域作成機能は、エンド・ユーザーが Web を使用してアクセスできるようにするために、レポートを公開する簡単なメカニズムを提供しています。ただし、このメカニズムは、Web を使用したレポートの公開に必須ではありません。

Oracle Portal 内で定義したアクセス制御情報は、Oracle Portal のリポジトリに格納されます。その後、Oracle Portal ユーザーとして、登録済の RDF をオプションで追加できます。これにより、登録済の RDF を Oracle Portal コンテンツ領域からアクセスできるようになります。また、Oracle Portal ユーザーとして、所定のレポート・リクエストを実行できます。そのとき、Oracle Portal リポジトリは、所定の Oracle Reports Services サーバーを使用して特定のレポートを実行する Oracle Portal のアクセス権限を検証するために使用されます。レポートを公開するために Oracle Portal を使用していない場合でも、セキュリティ・モデルを利用して、すべてのレポートを保護できます。これは、[5.16 項「Oracle Portal 以外のレポートの公開」](#)で説明するステップに従って、簡単に実行できます。

Oracle Reports Services では、Oracle Login Server シングル・サインオン機能、および軽量ユーザーの概念が利用されています。各 Oracle Portal ページには様々なポートレット・プロバイダからのデータが含まれています。各ポートレット・プロバイダは独自のログイン・プロシージャを持っています。ポートレット・プロバイダごとに常にユーザー ID を要求されないようにするために、Oracle Portal ではシングル・サインオン機能が提供されています。ユーザーがログインすると、Oracle Portal によって、登録されたポートレット・プロバイダとサブシステムのすべてに自動的にログインします。Login Server シングル・サインオンの詳細は、Oracle Portal のドキュメントを参照してください。

Oracle WebDB リリース 2.2 では、ユーザーとデータベース・ユーザー・アカウントは同義でした。Oracle WebDB リリース 2.2 の開発者は、デフォルトで、コンポーネントまたはオブジェクトを独自のデータベース・スキーマに作成できました。Oracle Portal リリース 3.0 では、ユーザーは通常、管理目的のみでデータベース・スキーマにマップされます。Oracle Portal リリース 3.0 では、コンポーネントを作成する機能は、開発者がコンポーネントをスキーマに作成する権限を持っているかどうかではなく、コンポーネントをアプリケーションに作成する権限を持っているかどうか依存します。

Oracle Portal リリース 3.0 では、グループがロールのかわりに使用されます。グループは、共通の利害や責任を共有するユーザーまたは他のグループの集合です。したがって、グループは共通の権限を持ちます。Oracle Portal 管理者だけではなく、Oracle Portal にログインしていれば誰でもグループを作成できます。

Oracle Portal ユーザーまたはグループを Oracle Portal 内で定義するには、Oracle Portal コンテンツ領域内または独自のカスタム・ポータル内から、特定の機能またはオブジェクトのどちらかへのアクセスを定義します。Oracle Reports Services セキュリティでは、Oracle Portal ユーザーおよびグループにより認証および認可がインプリメントされている（つまり、Oracle Reports Services セキュリティでは、誰が何にアクセスできるかが定義されている）ため、ユーザー独自のカスタム・ポータルを使用している場合でも、Oracle Reports Services セキュリティは Access Control List (ACL) のチェックの質問に回答できます。

ユーザーおよびグループの詳細は、Oracle Portal リリース 3.0 のドキュメントを参照してください。

5.4.1 Oracle Portal と Oracle Reports Services サーバー間の認証情報の共有

セキュリティ環境を構成する前に、AUTHID コマンドライン引数について知っておく必要があります。AUTHID コマンドライン引数は、アプリケーション・ユーザーを認証するために使用します。AUTHID コマンドライン引数は、Oracle Portal 固有のパラメータではありません。保護された Oracle Reports Services サーバーに対してレポートを実行するには、この認証情報が必要になります。

Oracle Portal の統合には、Oracle Portal へのログイン時に入力された情報が使用されます。Oracle Reports Services CGI では、Oracle Portal の内部または外部からレポートを実行するときに、AUTHID コマンドライン引数のかわりに、Oracle Portal のユーザー名とセッション ID が使用されます。この機能は Oracle Portal およびコンテンツ領域ビルダーの両方で動作します。

5.5 Oracle Portal の統合アーキテクチャ

Oracle Reports Services の Web の構成およびコンポーネントは旧リリースと同一ですが、CGI またはサブレット・インタフェースを使用してレポートを実行する機能が付加されています。Oracle Reports Services と Oracle Portal リポジトリ間の通信は、C API を使用して実現されます。C API は、デフォルトで、Oracle Portal リポジトリが存在する Oracle データベースと通信します。Oracle Portal リポジトリは、Oracle Portal ユーザーの妥当性チェック、およびレポート・リクエストのアクセスの確認のために検査されます。

このアーキテクチャはオープン API を利用しているため、独自のセキュリティ・システムに対するセキュリティ・チェックを再度インプリメントすることもできます。このオープン性により、ユーザーの Lightweight Directory Access Protocol (LDAP) サーバー、または適切に設定されているその他のカスタム・セキュリティ・サーバーに対して、ユーザーを認証できるようになります。詳細は、Oracle Portal のドキュメントを参照してください。

5.6 Oracle Portal への Oracle Reports Services セキュリティのインストール

この項では、Oracle Reports Services セキュリティおよび Oracle Portal を 1 つのマシンにインストールする方法について説明します。Oracle Reports Services セキュリティと Oracle Portal は、別々のマシンにインストールすることもできます。Oracle Portal を使用して適切な機能セキュリティ・モデルを利用する際に、Oracle Reports Services セキュリティと Oracle Portal は、同一のマシンに存在する必要はありません。複数サーバーでの Oracle Reports Services サーバーと Oracle Portal の構成の詳細は、[第 6 章「Oracle Reports Services サーバーのクラスタの構成」](#)を参照してください。

Oracle Reports Services 6i セキュリティと Oracle Portal をインストールするステップを次に示します。

- ステップ 1: Oracle Portal を Oracle データベースにインストールします。
- ステップ 2: Oracle Reports Services をインストールします。

インストールの詳細は、『Oracle9i Application Server インストレーション・ガイド』を参照してください。

5.6.1 ステップ 1: Oracle データベースへの Oracle Portal のインストール

Oracle データベース・リリース 8.1.6 以降とともに、Oracle Portal リリース 3.0 を別々の ORACLE_HOME にインストールします。Oracle Portal は、Web ブラウザを使用してのみアクセスできる Oracle データベースに対するオプションです。Oracle Portal リリース 3.0 は Oracle9i Application Server を介してインストールされます。詳細は、『Oracle9i Application Server インストレーション・ガイド』を参照してください。

5.6.2 ステップ 2: Oracle Reports Services のインストール

Oracle Reports Services コンポーネントをインストールします。インストーラによってマシンへの依存性が自動的に分析された後、選択したオプションに基づいて Oracle Reports Services が構成されます。Oracle Reports Services は、Oracle9i Application Server を介してインストールされます。

Oracle Reports Services Builder を使用する必要がある場合は、Oracle9i Developer Suite を使用してインストールします。

5.7 セキュリティ環境の構成

Oracle Reports Services 6i のセキュリティ環境を構成するには、次のステップを実行します。

- ステップ 1: Oracle Portal 内の Oracle Reports Services セキュリティを使用可能にします。
- ステップ 2: SECURITYTNSNAMES パラメータおよび PORTALUSERID パラメータを追加します。
- ステップ 3: Oracle Portal を起動します。

5.7.1 ステップ 1: Oracle Portal 内の Oracle Reports Services セキュリティを使用可能にする

Windows NT で Oracle Portal 内の Oracle Reports Services セキュリティを手動で使用可能にするには、次のディレクトリにある RWWWVINS.SQL スクリプトを実行します。

```
%ORACLE_HOME%\REPORT60\SERVER\SECURITY\3.0
```

UNIX で Oracle Portal 内の Oracle Reports Services セキュリティを手動で使用可能にするには、次のディレクトリにある `RWWWVINS.SQL` スクリプトを実行します。

```
$ORACLE_HOME/reports60/admin/server/security
```

Oracle Portal 管理者として、次のスクリプトを実行します。

```
sqlplus> @rwwwvins.sql <portal30 schema owner>
```

インストール中にプロンプトが表示されるので、Oracle Portal のスキーマ所有者を次のように入力します（たとえば、`portal30/portal30@orcl`）。

```
username/password@connectstring
```

このスクリプトにより、適切なオブジェクト定義、メニュー・エントリおよびグループが作成されます。次のグループが作成されます。

- `RW_ADMINISTRATOR`
- `RW_DEVELOPER`
- `RW_POWER_USER`
- `RW_BASIC_USER`

これらの 4 つのグループは、Oracle Portal と Oracle Reports Services セキュリティを使用可能にしたときに作成されます。セキュリティ認証のチェックのために、各 Oracle Portal ユーザーをこれらのグループのいずれかに割り当てする必要があります。

5.7.1.1 RW_ADMINISTRATOR

このグループに割り当てられた Oracle Portal ユーザー（たとえば、Oracle Reports Services 管理者、Oracle Portal 管理者または Login Server 管理者）は、Oracle Portal に登録済のレポート定義ファイル、サーバーおよびプリンタ・オブジェクトを、`CREATE`、`UPDATE` および `DELETE` できます。Oracle Reports Services 管理者は、他のユーザーに対してセキュリティ権限を割り当てたり、Oracle 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。ユーザーの作成および管理方法の詳細は、Oracle Portal のドキュメントを参照してください。

このユーザーは、Oracle Reports Services Queue Manager の管理者の機能に対するアクセス権を持っています。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、およびサーバーのシャットダウンなど、サーバーのキューを管理できます。

5.7.1.2 RW_DEVELOPER

`RW_DEVELOPER` は、`RW_POWER_USER` グループおよび `RW_BASIC_USER` グループの権限に加えて、システム環境を表示する `SHOWENV` や `SHOWMAP` など、CGI コマンドをすべて実行できます。このグループは、テストの実行や詳細なエラー・メッセージの取得を必要とする開発者にも割り当てることができます。

5.7.1.3 RW_POWER_USER

RW_POWER_USER グループの権限を持つ Oracle Portal ユーザーは、RW_BASIC_USER グループの権限に加えて、セキュリティ・チェックが失敗した場合に、より詳細なエラー・メッセージを見ることができます。たとえば、HTML に対してレポートを実行しようとして、これが許可されていない場合には、次のようなメッセージを受け取ります。

HTML レポートを実行できません。

5.7.1.4 RW_BASIC_USER

Oracle Portal が Oracle Portal ユーザーを作成すると、そのユーザーは自動的に RW_BASIC_USER グループに加わります。このグループの権限を持つ Oracle Portal ユーザーは、レポートの実行権限が付与されていなければレポートを実行できません。この Oracle Portal ユーザーは、セキュリティ・チェックが失敗した場合も、非常に単純なエラー・メッセージしか見ることができません。次のようなエラー・メッセージを受け取ります。

セキュリティ・チェック・エラー

注意： Oracle Reports Services 6i および Oracle Portal3.0 では、認証 cookie ドメインは使用されません。Oracle Reports Services では、PL/SQL コールを使用して、セッション ID とユーザー ID が CGI または サブレットに渡されます。

5.7.2 ステップ 2: SECURITYTNSNAMES パラメータおよび PORTALUSERID パラメータを追加する

このステップは、データベース・レベルのセキュリティ権限を持つユーザーによって実行されます。

Oracle Reports Services サーバーが実行中の場合は、まずシャットダウンする必要があります。次に、SECURITYTNSNAMES=<"tnsname"> パラメータおよび PORTALUSERID=<portal_username>/<portal_password> を、Oracle Reports Services の構成ファイル（たとえば、rep60_<machinename>.ora）に追加します。Windows NT の場合、この構成ファイルは次のディレクトリにあります。

```
%ORACLE_HOME%\REPORT60\SERVER
```

UNIX の場合、この構成ファイルは次のディレクトリにあります。

```
$ORACLE_HOME/reports60/server
```

各項目の内容は次のとおりです。

tnsname	Oracle Portal がインストールされているインスタンスの TNSname
portal_username	Oracle Portal がインストールされているデータベース・ユーザーの名前
portal_password	Oracle Portal がインストールされているデータベース・ユーザーのパスワード

Oracle Reports Services が存在するマシン上の tnsnames.ora ファイルに正しいエイリアスがあることを確認します。

Oracle Reports Services サーバーには、SECURITYTNSNAMES エントリおよび PORTALUSERID エントリが必要です。これらのエントリは、ユーザーがジョブ・リクエストを送信するときに、アクセス制御情報を探す場所を示します。Oracle Portal および Oracle Reports Services セキュリティのフレームワークがインストールされているデータベース・インスタンスの名前を、サーバーに指示する必要があります。SECURITYTNSNAMES エントリおよび PORTALUSERID エントリが Oracle Reports Services サーバーの構成ファイルに追加されると、Oracle Portal リポジトリのアクセス制御情報が実施されます。この Oracle Reports Services サーバーに対してレポートを実行するように指定した Oracle Portal ユーザーには、識別が要求されます。

レポートを実行するために Oracle Portal コンテンツ領域に正常にログインすると、Oracle Portal では、このログイン情報（ユーザー名とセッション ID）は AUTHID コマンドライン・パラメータのかわりに使用され、Oracle Portal を介して Oracle Reports Services サーバーにより検証されます。

5.7.3 ステップ 3: Oracle Portal を起動する

Web ブラウザから Oracle Portal を起動し、インストール中に識別したユーザーで Oracle Portal にログインします。このユーザーには、アプリケーション・レベルのセキュリティ権限があります。

5.8 プリンタ・アクセス

ユーザー環境において、Oracle Portal ユーザーがアクセスできるネットワーク・プリンタまたはローカル・プリンタを多数用意することができます。しかし、一定期間プリンタの使用を制限して Oracle 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、あるいは特定レポート出力の印刷に使用する特定プリンタを識別することが必要になる場合もあります。たとえば、非常に長い月次レポートであるため、情報技術（IT）部門の高速大量印刷用のプリンタに出力する必要があるような場合です。

プリンタ・アクセスには、次の情報が格納されます。

- Oracle Portal 内部からのレポート出力の印刷に使用できるプリンタ
- レポート出力のアクセス権限を持つユーザー
- 使用可能日カレンダーの割当てによるプリンタが処理可能な期間

使用可能日カレンダーを使用していても、Oracle Portal のセキュリティ・フレームワーク内へのプリンタの登録は必須ではありません。

Oracle Portal 内にプリンタを登録すると、それを Oracle Reports Services サーバー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。しかし、RDF ファイルを登録してプリンタに印刷する場合は、印刷に使用できるのは特定の Oracle Reports Services サーバーに関連付けられたプリンタのみです。

RDF に対するアクセス権を定義する際、レポート出力が送信されるプリンタの登録済サブセットをさらに制限するように選択できます。たとえば、Oracle Reports Services サーバー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ても、最高経営責任者以外の従業員が一般台帳レポートを実行する場合には、このプリンタを選択できないようにします。かわりに、レポート・リクエストを実行する Oracle Portal ユーザーにこのプリンタのサブセットをリストして、出力が送信される場所を選択できるようにします。

5.9 Oracle Portal コンテンツ領域の作成

注意： 次の情報はすべて、Oracle Portal ユーザー向けです。すべての Oracle Portal ユーザーはアプリケーション・レベルのセキュリティ権限を持っていますが、データベース・レベルのセキュリティ権限は持っていません。

Oracle Portal では Web コンテンツ領域の作成ウィザードが提供されていて、ステップに従って自動的に作成できるようになっています。このウィザードは Oracle Portal リポジトリにすべて含まれています。コンテンツ・ページを作成するには、Oracle Portal にログインして、Oracle Portal の「ナビゲータ」アイコン、「コンテンツ領域」タブ、「新規作成 ... 「コンテンツ領域」」の順にクリックします。コンテンツ領域を作成する際、コンテンツの所有者または DBA は、Oracle Portal コンテンツ領域に項目またはリンクを追加できます。項目として追加できるのは、URL、テキスト項目、イメージ・マップ、ファイル、PL/SQL コール、フォルダへのリンク、またはその他の Oracle Portal コンポーネントです。RDF は、Oracle Portal 内に登録すると他の Oracle Portal コンポーネントと同様に扱われ、Web コンテンツ領域に同様に追加できます。

このリンクによってレポートをただちに実行するように選択できます。このとき、ユーザーは Oracle Portal を介して認証され、出力は、許可または選択されたフォーマットで生成されます。あるいは、レポートをスケジュールし、出力を既存の Oracle Portal コンテンツ領域に配信するように選択することもできます。レポートのスケジュールの詳細は、[付録 A 「カレンダーの定義によるレポートへのユーザー・アクセスの管理」](#)を参照してください。

5.10 レポートのセットアップおよび配置

Oracle Reports Services セキュリティおよび Oracle Portal をインストールし、Oracle Portal コンテンツ領域をセットアップすると、ユーザーのセットアップおよび Oracle Portal によるレポートの配置を始めることができます。次の項では、Oracle Portal ユーザーのセットアップに必要なステップおよびレポートの配置方法について説明します。

- セキュリティ管理のための Oracle Portal ユーザーの作成と有効化
- Oracle Portal のアクセス制御のセットアップ
- レポートの登録
- レポートの配置
- レポートの実行

次の状況を前提とします。

- すでに Oracle Reports Services サーバーがインストールされ、Web レポート用に構成されている。また、URL を介して使用できるようになっている。次に例を示します。
`http://mycomputer.domain/cgi-bin/rwcgi60.exe`
- Oracle Portal コンテンツ領域がすでに存在している。このコンテンツ領域は Oracle Portal ナビゲータによりアクセスされる。
- `SECURITYTNSNAME=<"tnsname">` パラメータがすでに Oracle Reports Services サーバーの構成ファイルに追加されている。tnsname は Oracle Portal がインストールされているインスタンスを参照している。
- `PORTALUSERID=<portal_username>/<portal_password>` パラメータがすでに Oracle Reports Services サーバーの構成ファイルに追加されている。ここで、`<portal_username>/<portal_password>` は、Oracle Portal がインストールされているデータベースの有効なユーザー名とパスワードである。

5.11 セキュリティ管理のための Oracle Portal ユーザーの作成と有効化

Oracle Portal ユーザーが、レポート、サーバーまたはプリンタを登録し、さらに、これらのオブジェクトに対するアクセス権を他の Oracle Portal ユーザーに許可または付与できるようにするには、このステップを実行する必要があります。

概要

この例では、Oracle Portal ユーザーが、権限の付与、DBA グループの割当て、および RW_ADMINISTRATOR グループの割当てによって、Oracle Reports Services セキュリティを管理する方法について説明します。

前提条件

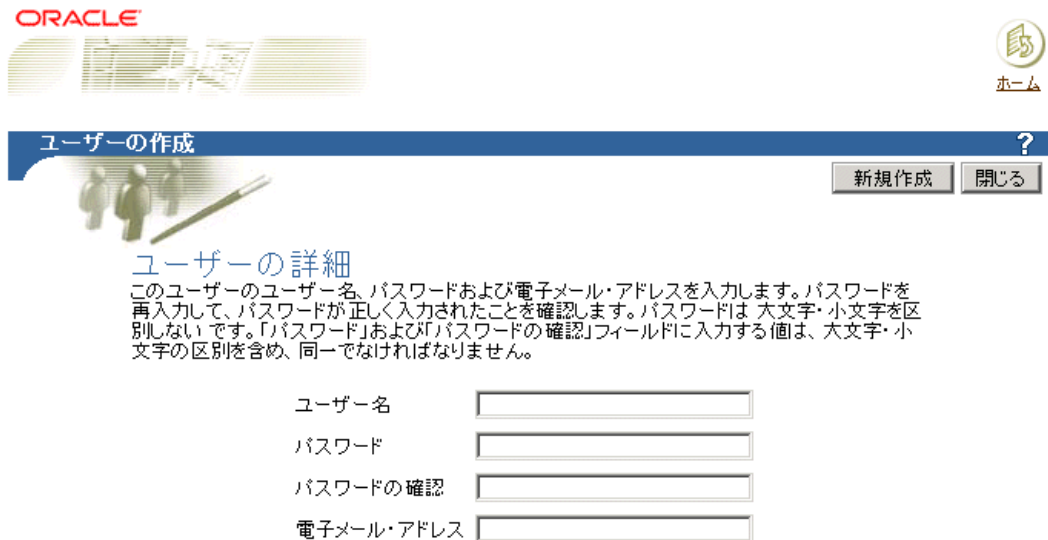
この例では、次の状況を前提とします。

- Oracle Portal 管理者が適切な URL を使用して Oracle Portal を開く。
- Oracle Portal 管理者がログインする。

5.11.1 セキュリティ管理のためのユーザー REPORTSDEV の作成と有効化

次のステップを実行する必要があります。

1. Oracle Portal ホーム・ページから「管理」タブをクリックします。
2. 「管理」ページから「ユーザー」ポートレットの「新規ユーザーの作成」をクリックします。次の画面が表示されます。



ORACLE

ホーム

ユーザーの作成 ?

新規作成 閉じる

ユーザーの詳細

このユーザーのユーザー名、パスワードおよび電子メール・アドレスを入力します。パスワードを再入力して、パスワードが正しく入力されたことを確認します。パスワードは 大文字・小文字を区別しないです。「パスワード」および「パスワードの確認」フィールドに入力する値は、大文字・小文字の区別を含め、同一でなければなりません。

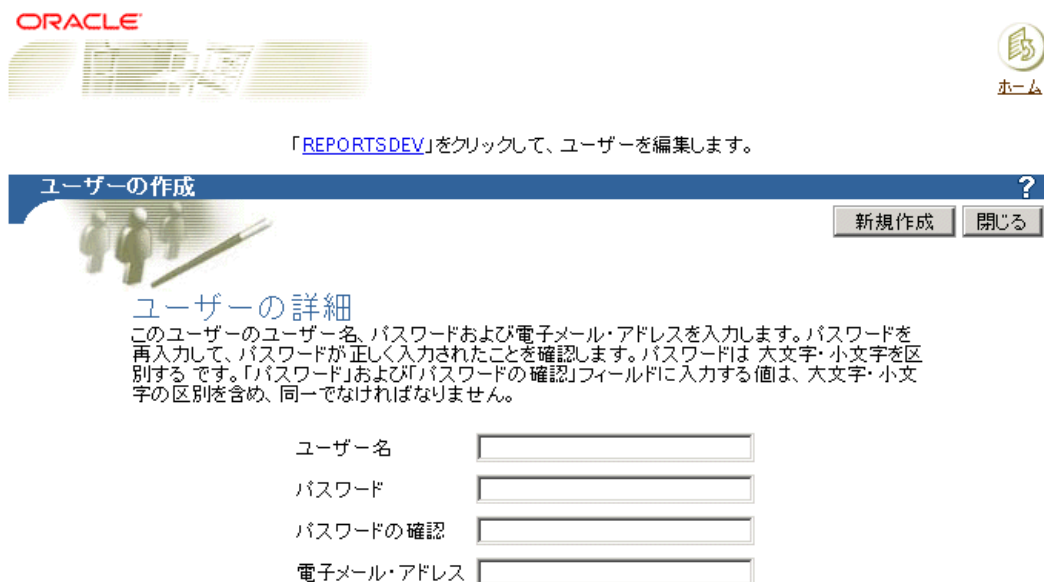
ユーザー名

パスワード

パスワードの確認

電子メール・アドレス

3. 「**ユーザーの詳細**」ヘッダーの下にある「**ユーザー名**」、「**パスワード**」および「**パスワードの確認**」フィールドに「**REPORTSDEV**」と入力します。次に、「**新規作成**」ボタンをクリックします。次の画面が表示されます。



ORACLE

ホーム

「[REPORTSDEV](#)」をクリックして、ユーザーを編集します。

ユーザーの作成 ?

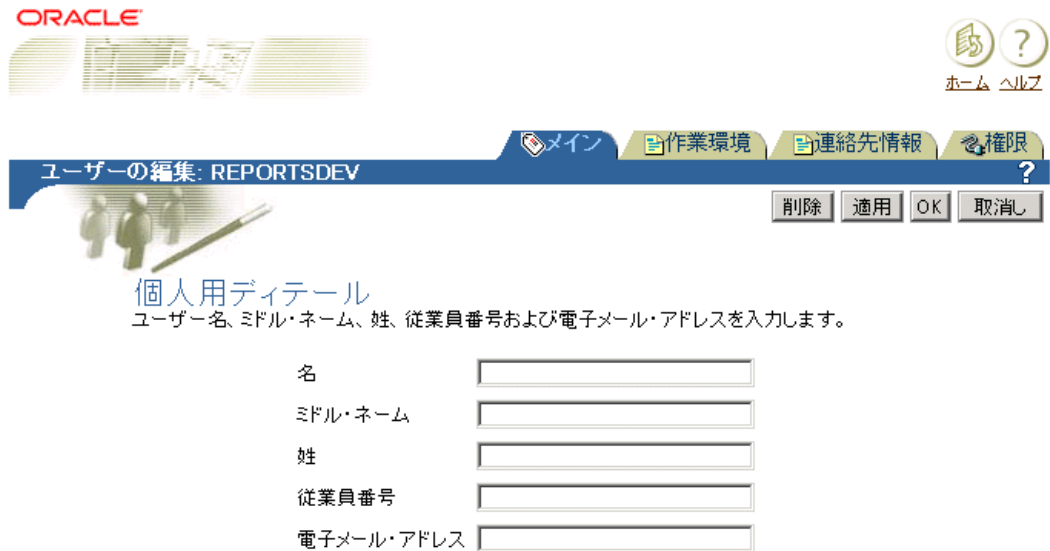
新規作成 閉じる

ユーザーの詳細

このユーザーのユーザー名、パスワードおよび電子メール・アドレスを入力します。パスワードを再入力して、パスワードが正しく入力されたことを確認します。パスワードは 大文字・小文字を区別する です。「パスワード」および「パスワードの確認」フィールドに入力する値は、大文字・小文字の区別を含め、同一でなければなりません。

ユーザー名	<input type="text" value="REPORTSDEV"/>
パスワード	<input type="password"/>
パスワードの確認	<input type="password"/>
電子メール・アドレス	<input type="text"/>

4. 「REPORTSDEV」をクリックして、ユーザーを編集します。次の画面が表示されます。



The screenshot shows the Oracle Portal user management interface. At the top, there is an Oracle logo and navigation links for 'ホーム' (Home) and 'ヘルプ' (Help). Below this is a blue header bar with the title 'ユーザーの編集: REPORTSDEV' and several tabs: 'メイン' (Main), '作業環境' (Work Environment), '連絡先情報' (Contact Information), and '権限' (Permissions). The 'メイン' tab is selected. Below the header, there are buttons for '削除' (Delete), '適用' (Apply), 'OK', and '取消し' (Cancel). The main content area is titled '個人用ディテール' (Personal Details) and contains a description: 'ユーザー名、ミドル・ネーム、姓、従業員番号および電子メール・アドレスを入力します。' (Enter user name, middle name, last name, employee number, and email address). Below this description are five input fields: '名' (First Name), 'ミドル・ネーム' (Middle Name), '姓' (Last Name), '従業員番号' (Employee Number), and '電子メール・アドレス' (Email Address).

5. 「グループ・メンバー」ヘッダーまでこの画面をスクロールして、次の画面に示すように、「DBA」および「PORTAL_ADMINISTRATORS」グループを選択します。

グループ・メンバー

このユーザーの追加先グループを選択します。所属するグループにより、ユーザーが実行可能なタスクが決定されます。

- ☒ DBA
- ☒ PORTAL_ADMINISTRATORS
- ☐ PORTAL_DEVELOPERS
- ☐ PORTLET_PUBLISHERS

6. 「適用」ボタンをクリックします。

7. 「権限」タブをクリックします。次の画面が表示されます。

The screenshot shows the '権限' (Permissions) tab for user 'REPORTSDEV'. The interface includes a header with navigation tabs: 'メイン' (Main), '作業環境' (Work Environment), '連絡先情報' (Contact Information), and '権限' (Permissions). Below the header, there are buttons for '削除' (Delete), '適用' (Apply), 'OK', and '取消し' (Cancel). The main content area is titled 'オブジェクト権限' (Object Permissions) and contains a table with columns 'オブジェクト・タイプ' (Object Type) and '権限' (Permissions). The table lists four object types: 'すべてのページ' (All Pages), 'すべてのスタイル' (All Styles), 'すべてのレイアウト' (All Layouts), and 'すべてのプロバイダ' (All Providers). Each row has a dropdown menu for selecting permissions, all currently set to 'なし' (None).

オブジェクト・タイプ	権限
すべてのページ	なし
すべてのスタイル	なし
すべてのレイアウト	なし
すべてのプロバイダ	なし

8. 「コンテンツ領域権限」ヘッダーまでこの画面をスクロールして、「権限」ドロップダウン・リストから「管理」を選択します。
9. 続けて、「アプリケーション権限」ヘッダーまでこの画面をスクロールして、「すべてのアプリケーション」および「すべての共有コンポーネント」用の「権限」ドロップダウン・リストから「管理」を選択します。
10. 「適用」ボタンをクリックします。
11. 「OK」ボタンをクリックします。「管理」ページが再び表示されます。
12. 「リスト」ボタンをクリックして、リストから「DBA」および「RW_ADMINISTRATOR」グループを選択します。

注意： [Ctrl] キーを押したまま選択項目をクリックすると、複数のオブジェクトを選択できます。

13. 「編集」ボタンをクリックします。次の画面が表示されます。



グループの編集: RW_ADMINISTRATOR

詳細 メンバー 権限 ?

削除 適用 OK 取消し

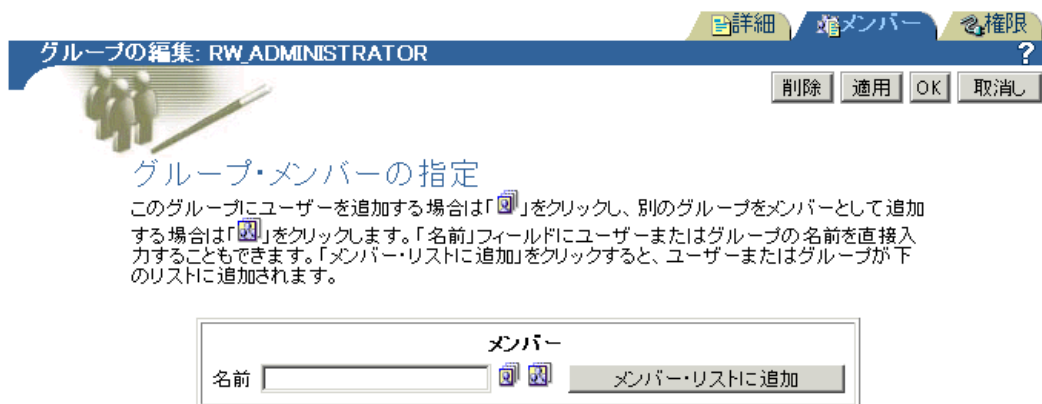
グループの詳細の編集
グループの名前を変更するか、グループの説明を記述します。

名前 RW_ADMINISTRATOR

適用 SHARED

説明

14. 「メンバー」タブをクリックします。次の画面が表示されます。



グループの編集: RW_ADMINISTRATOR

詳細 メンバー 権限 ?

削除 適用 OK 取消し

グループ・メンバーの指定
このグループにユーザーを追加する場合は「[ユーザーアイコン]」をクリックし、別のグループをメンバーとして追加する場合は「[グループアイコン]」をクリックします。「名前」フィールドにユーザーまたはグループの名前を直接入力することもできます。「メンバー・リストに追加」をクリックすると、ユーザーまたはグループが下のリストに追加されます。

メンバー

名前

メンバー・リストに追加

15. 「グループ・メンバーの指定」ヘッダーの下にある「名前」フィールドに「REPORTSDEV」と入力するか、または「ユーザーをブラウズ」ボタンをクリックして、リストから REPORTSDEV を選択します。

16. 「メンバー・リストに追加」ボタンをクリックします。その後、「グループ・メンバー・リスト」ヘッダーまでこの画面をスクロールすると、REPORTSDEV が追加されたことを確認できます。DBA および RW_ADMINISTRATOR グループの REPORTSDEV 権限が付与されています。
17. 「OK」ボタンをクリックします。Oracle Portal の「管理」ページが再び表示されます。
18. REPORTSDEV でログインしてセキュリティを管理できるようにするには、Oracle Portal 管理者のログアウトを行う必要があります。

REPORTSDEV が作成され、Oracle Reports のセキュリティを管理できます。

5.12 Oracle Portal のアクセス制御のセットアップ

Oracle Reports Services と Oracle Portal の統合では、Oracle Reports Services により実行されるレポートへのアクセスを制限するために、管理インタフェースがそのままのかたちで提供されています。セキュリティ・チェックは、Oracle Portal ユーザーが必要なアクセス制御を持っていることを確認するために実行されます。

Oracle Portal リポジトリに格納されているアクセス制御データとは、機能またはアプリケーション・レベルのセキュリティ、つまり、特定のレポートへの Oracle Portal ユーザーのアクセス権を意味することに留意してください。データ・セキュリティは、USERID パラメータにより処理され、実行時に渡すことができますが、Oracle Portal ユーザーにプロンプトを表示することもできます。

すべてのユーティリティでは、アクセス制御情報の作成、編集または削除のためにウィザードが使用されます。アクセス制御情報は、入力後、Oracle Portal リポジトリにメタデータとして格納されます。Oracle Portal のこのセキュリティ情報にアクセスできるのは、Oracle Reports Services システム管理者権限（DBA および RW_ADMINISTRATOR グループ）を持つ Oracle Portal ユーザーのみです。

5.13 レポートの登録

セキュリティを管理する機能により REPORTSDEV が作成されると、次のことを実行できます。

- レポートの保護
- サーバーの保護
- プリンタの保護
- 使用可能日カレンダーの定義
- レポート、サーバー、プリンタの実行およびアクセスをユーザーに許可

概要

この例では、次の処理について説明します。

- サーバーの登録
- RDF アクセスの作成

前提条件

この例では、次の状況を前提としています。

- Oracle Portal は、Oracle Portal インストール用の適切な URL を使用してオープンされている。
- REPORTSDEV は Oracle Portal にログインしている。
- REPORTSDEV には `accounting.rdf` というレポートが登録されている。
- REPORTSDEV は、SCOTT および BJ というユーザー（両方とも Oracle Portal ユーザー）にレポートの実行を許可している。
- PUBSVR という Oracle Reports Services サーバーがすでにセットアップされ、使用可能になっている。

5.13.1 サーバーの登録

Oracle Reports Services サーバー・アクセスには、次の情報が格納されます。

- ジョブ・リクエストの処理に使用できる Oracle Reports Services サーバー
- 登録済 Oracle Reports Services サーバーによる出力の印刷に使用できる登録済プリンタ
- 所定の Oracle Reports Services サーバーにレポート・リクエストを送信する権限を持つユーザー
- Oracle Reports Services サーバーがジョブ・リクエストの処理に使用できる期間
- 所定の Oracle Reports Services サーバーは、すべてのレポートを実行できるか、または保護アクセス用に登録されたレポートのみを実行できるか

Oracle Portal フレームワーク内の実際の Oracle Reports Services サーバーは、すでに存在していて、レポート・リクエストを実行するように構成する必要があります。

サーバー・アクセスを定義するには、次のことを実行します。

1. Oracle Portal ホーム・ページから「**管理**」タブをクリックします。
2. 「Oracle Reports Security」ポートレットの「Oracle Reports Security の**設定**」をクリックします。

3. 「Reports Server **アクセスを作成**」をクリックします。次の画面が表示されます。

ステップ 1 / 3

サーバー名およびプリンタ

サーバー名はOracle Portalでサーバーを識別するのに使用されます。

サーバー名

Reports ServerのTNS名

説明

4. 「サーバー名」フィールドに「NEWSERVER」と入力します。
5. 「Reports Server の TNS 名」フィールドに「PUBSVR」と入力します（この名前は、Oracle Reports Services サーバーの名前です）。
6. 「説明」フィールドに「Local reports server」と入力します。
7. 「Oracle Reports Web Gateway の URL」に Oracle Reports Services CGI およびサーバーレットの場所を入力します。次に例を示します。

```
http://mycompany.docmain/cgi-bin/rwcgi60.exe
```
8. 「登録されたレポート定義ファイルのみ実行する」および「プリンタ」を空白にしておきます。この例ではそれらは作成されません。また、この例ではプリンタは選択しません。ただし、プリンタを登録していれば、1 つ以上のプリンタを Oracle Reports Services サーバーに関連付けることができます。

注意：「登録されたレポート定義ファイルのみ実行する」にチェックマークを付けると、Oracle Portal 内で保護されているレポートを実行しないように、Oracle Reports Services サーバーに指示することになります。

9. Oracle Portal 内のプリンタを保護していた場合、この RDF ファイルに 1 つ以上のプリンタを関連付けることができます。しかし、この例ではプリンタは登録せず、この RDF はどのプリンタでも印刷できるものとします。「プリンタ」フィールドには以前に作成

したプリンタを含めることができますが、この例では必要ありません。プリンタの詳細は、5.8 項「[プリンタ・アクセス](#)」を参照してください。

10. すべてのフィールドを設定した後に、「次へ」ボタンをクリックします。次の画面が表示されます。

11. 「アクセス権限の付与」ヘッダーの下にある「ユーザー / ロール」フィールドに「SCOTT (ユーザー)」と入力するか、または「ユーザーをブラウズ」ボタンをクリックして、リストから SCOTT を選択します。
12. プルダウン・リストから「表示」を選択して、「追加」ボタンをクリックします。
13. 「アクセス権限の変更」ヘッダーまでスクロールして、サーバーに対するアクセス権がユーザーに付与されていることを確認します。
14. 次に、BJ にアクセス権を付与します。ステップ 11 ~ 13 を繰り返して、「ユーザー / ロール」フィールドに「BJ (ユーザー)」と入力するか、または「ユーザーをブラウズ」リストから BJ を選択します。

15. 「次へ」ボタンをクリックして続行します。次の画面が表示されます。

この例では、この画面は空白にしておきます。使用可能日カレンダーの詳細は、[付録 A 「カレンダーの定義によるレポートへのユーザー・アクセスの管理」](#)を参照してください。

注意： このサーバーを Oracle Portal のレポートに使用できる日や時間を制限する場合は、使用可能日カレンダーを作成した後、ここでその使用可能日カレンダーを選択します。

16. 「完了」ボタンをクリックします。次の画面が表示されます。



17. 「閉じる」ボタンをクリックします。「Oracle Reports Security の設定」ページが再び表示されます。

これで、Oracle Reports Services サーバーの登録が完了しました。次に、レポートを登録します。

5.13.2 レポート定義ファイル・アクセスの作成

レポート定義ファイル (RDF) アクセスには、次の情報が格納されます。

- 保護アクセス用に登録された Oracle Reports Services ファイル (RDF、REP、REX および XML)。レポートのフルパスを設定できます。しかし、オラクル社では、ファイル名にパスを設定せずに、RDF、REP、REX または XML のみを使用して、REPORTS60_PATH 環境変数または SOURCEDIR パスにファイルの場所を追加することをお勧めします。
- Oracle Portal 内でこのレポートのインスタンスを識別する名前 (Oracle Portal 内のみで適用可能)
- このレポートのインスタンスを実行するアクセス権限を持つユーザー
- このレポートのインスタンスを実行できる期間

- 特定のレポートに対するリクエストの処理に使用できる Oracle Reports Services サーバー
- パラメータ、指定した形式 (PDF、HTML または HTMLCSS) および有効化トリガーを使用してレポートを配布する方法
- このレポートが印刷されるプリンタ

RDF が Oracle Portal 内に登録されると、その RDF により、Oracle Portal コンポーネントが作成されます。オラクル社では、同一の RDF ファイルに複数のインスタンスを登録することはお薦めしません。

特定レポートにアクセス権を持つ Oracle Portal ユーザーの指定のほか、Oracle Portal ユーザーがレポートを対話的に操作する方法も指定できます。レポートに指定できるパラメータに加えて、Oracle Portal にユーザーのレポートを登録する際に使用するパラメータ・フォームも作成できます。

Oracle Portal パラメータ・フォームは、セキュリティの制限事項を設定するために使用されます。たとえば、所定のレポートを有効なレポート出力形式に限定したり、このような有効な形式に限定してパラメータ・フォームを表示したりします。セキュリティ情報は Oracle Portal リポジトリに格納されます。

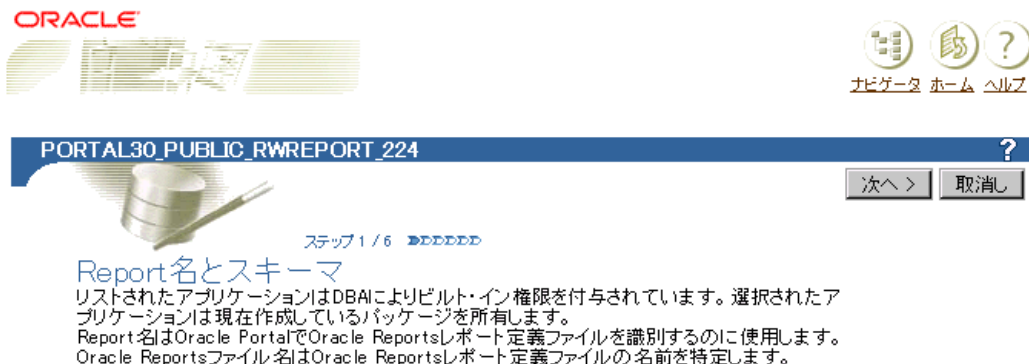
Oracle Reports Services のパラメータ・フォームにより、たとえば、PL/SQL トリガーを付加するなど、追加の制限事項を設定できます。オラクル社では、パラメータ・フォームの競合を回避するために、Oracle Portal パラメータ・フォームまたは Oracle Reports Services パラメータ・フォームのいずれか 1 つのみ使用することをお薦めします。

この任意のパラメータ・フォームは、ユーザーがアクセスできる値、またはこのレポートの出力に必要な追加パラメータの値を制限するために使用できます。たとえば、HTML レポートのページ・ストリームの強制実行、特定の値に基づくデータの表示、または定義済の LOV の表示などに使用できます。さらに、ジョブの送信中にどのパラメータをユーザーに公開するかを指定することもできるため、各ユーザーは同じレポートに対して異なるオプションを適用できます。たとえば、レポート出力を HTML、HTMLCSS または PDF のいずれにするかをユーザーが指定することもできます。

「**パラメータ入力フォーム**」では、選択する値とともに、パラメータも表示するかどうかを指定できます。これを行うには、「**コンポーネントの管理**」画面から「**カスタマイズ**」リンクを選択します。その後、エンド・ユーザーに表示するパラメータをチェックします。

RDF アクセスの作成およびレポートの登録を行うには、次のようにします。

1. 「Oracle Reports Security」画面が再び表示されます。
2. 「レポート定義ファイル・アクセス」ポートレットの「レポート定義ファイル・アクセスを作成」をクリックします。次の画面が表示されます。



「アプリケーション」、「Report 名」および「Reports Servers」フィールドは、すでに設定されています。

3. 「Report 名」フィールドを、たとえば、「Financial」に変更します。
4. 「Reports Servers」フィールドに NEWSERVER を選択します。Oracle Reports Services サーバーが 1 つしかない場合でも、それを選択して続行します。[Ctrl] キーを押したまま、使用する各サーバーをクリックすると、複数のサーバーを選択することもできます。
5. 「Oracle Reports ファイル名」フィールドに「accounting.rdf」と入力します。ここでは RDF が REPORTS60_PATH にあることを前提としていますが、フルパスをこのフィールドにハードコードすることもできます。
6. 「説明」フィールドは任意です。この例では「Financial statement」と入力します。

7. 「次へ」ボタンをクリックします。次の画面が表示されます。

8. 「アクセス権限の付与」ヘッダーの下にある「ユーザー / ロール」フィールドに「SCOTT (ユーザー)」と入力するか、または「ユーザーをブラウズ」ボタンをクリックして、リストから SCOTT を選択します。
9. ドロップダウン・リストから「管理」を選択します。
10. 「追加」ボタンをクリックします。
11. 「アクセス権限の変更」ヘッダーまでこの画面をスクロールして、RDF ファイルに対するアクセス権がユーザーに付与されていることを確認します。
12. 次に、BJ にアクセス権を付与します。ステップ 8 ~ 11 を繰り返して、ユーザーには「BJ (ユーザー)」を選択します。

13. 「次へ」ボタンをクリックします。次の画面が表示されます。

ORACLE

ナビゲータ ホーム ヘルプ

FINANCIAL ?

ステップ 3 / 6

使用可能日カレンダー

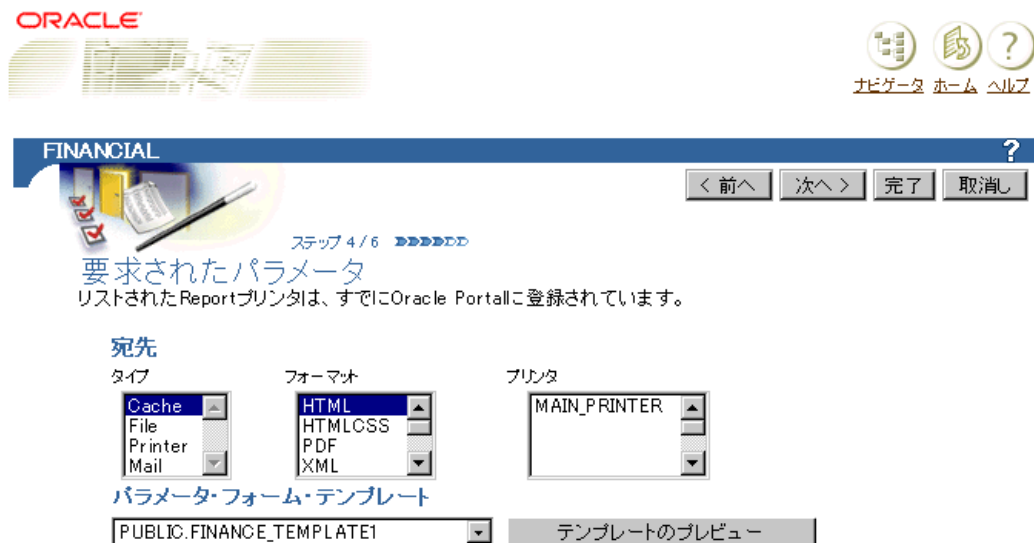
Oracle Reportsファイルで使用する使用可能日カレンダーを選択してください。

使用可能日カレンダー名

< 前へ 次へ > 完了 取消し

使用可能日はここでは指定しないため、この画面は空白のままにしておきます。

14. 「次へ」ボタンをクリックします。次の画面が表示されます。



ORACLE

ナビゲータ ホーム ヘルプ

FINANCIAL

ステップ 4 / 6

要求されたパラメータ

リストされた Report プリンタは、すでに Oracle Portal に登録されています。

宛先

タイプ フォーマット プリンタ

Cache
File
Printer
Mail

HTML
HTMLCSS
PDF
XML

MAIN_PRINTER

パラメータ・フォーム・テンプレート

PUBLIC.FINANCE_TEMPLATE1 テンプレートのプレビュー

15. この画面では「宛先」情報を選択します。「タイプ」、「フォーマット」、「プリンタ」および「パラメータ・フォーム・テンプレート」があります。たとえば、「タイプ」には「Cache」、「フォーマット」には「HTMLCSS」および「PDF」を選択します。

注意： [Ctrl] キーを押したまま選択項目をクリックすると、複数のオブジェクトを選択できます。

16. 「次へ」ボタンをクリックします。次の画面が表示されます。

ORACLE

ナビゲータ ホーム ヘルプ

FINANCIAL

ステップ 5 / 6

任意のパラメータ

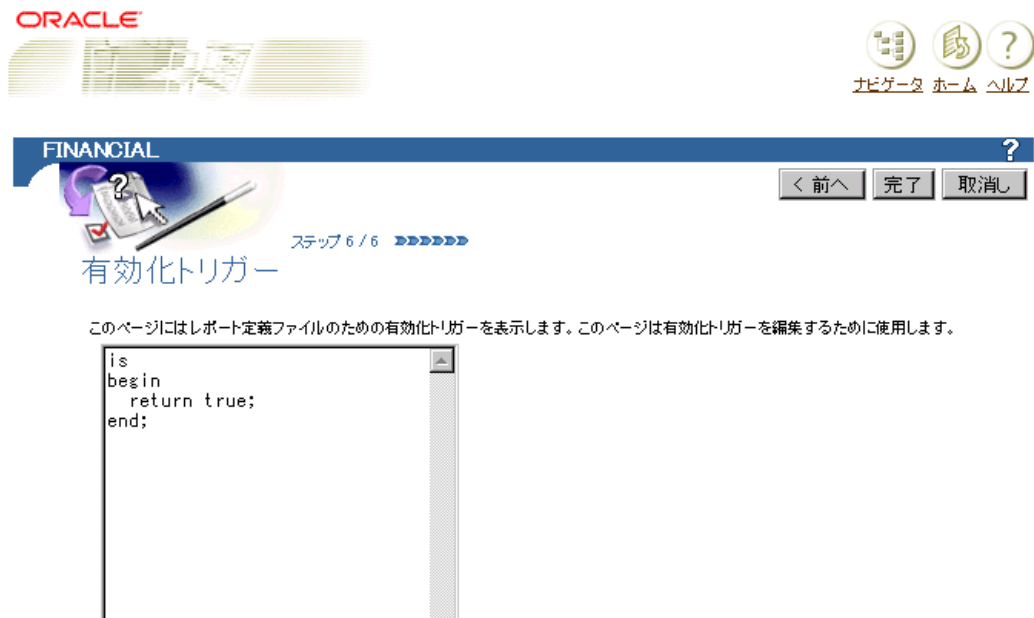
その他のパラメータを追加するときは、「パラメータの追加」ボタンをクリックして行を追加してください。

パラメータ名	LOV	下限	上限

パラメータ追加

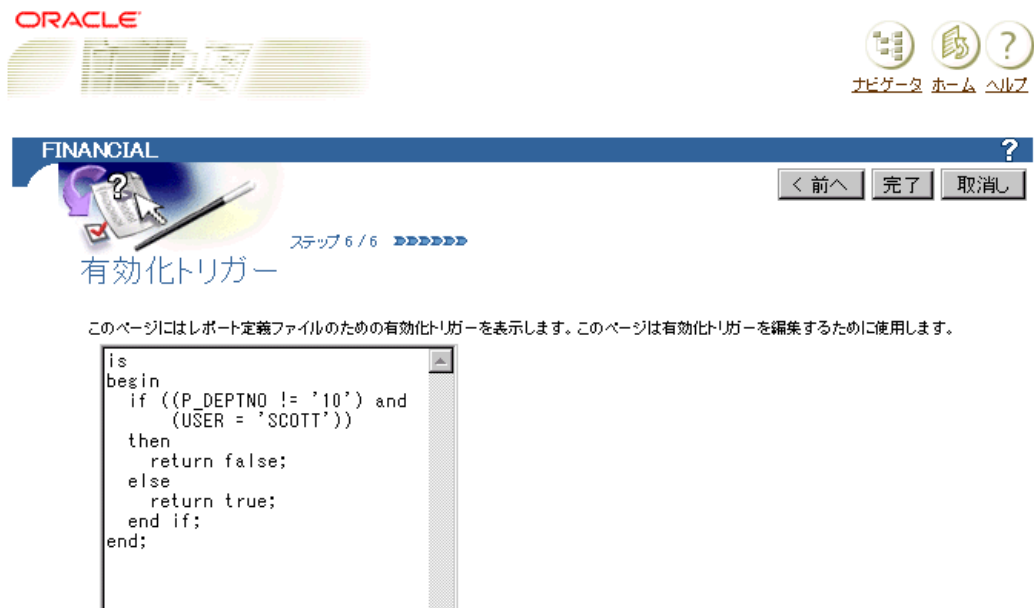
17. この例では、レポートに渡す部門番号を制限することにより、さらにアクセスを制限します。「パラメータ名」フィールドに「P_DEPTNO」と入力します。

18. 「次へ」ボタンをクリックします。次の画面が表示されます。



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「任意パラメータ」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。有効化トリガーは、ユーザーが「ランタイム・パラメータ・フォーム」を使用する場合に実行されます。

19. 次の画面のように、「有効化トリガー」画面の情報を変更します。SCOTT は、部門番号が 10 の場合のみ、このレポートを実行できます。その他のユーザーには、このような制限はありません。



20. 「完了」ボタンをクリックします。次の画面が表示されます。



The screenshot shows the Oracle Reports Services interface. At the top, a blue header bar contains the text '開発: EXAMPLE_APP.FINANCIAL' on the left and '開発' and '管理' tabs on the right. Below the header, there is a section titled 'レポート定義ファイル・アクセス' (Report Definition File Access) for 'FINANCIAL' and 'EXAMPLE_APP'. The 'バージョン状況' (Version Status) section shows '1 (PRODUCTION with VALID Package)'. The '最終変更' (Last Change) section shows '2000-11-28 火曜日 14:27 by REPORTSDEV'. The '実行リンク' (Execution Link) section shows 'PORTAL30_DEMO.FINANCIAL.show' and 'PORTAL30_DEMO.FINANCIAL.show_parms'. The 'PL/SQLソース' (PL/SQL Source) section shows 'パッケージ仕様' and 'パッケージ本体'. The 'コール・インタフェース' (Call Interface) section shows '表示'. At the bottom, there are icons for '編集' (Edit), '実行' (Execute), 'ポートレットとして実行' (Execute as Portlet), 'カスタマイズ' (Customize), 'Add to Favorites', '情報' (Info), and '削除' (Delete).

開発: EXAMPLE_APP.FINANCIAL

開発 管理 ?

開じる

レポート定義ファイル・アクセス FINANCIAL
アプリケーション EXAMPLE_APP

バージョン状況 [1 \(PRODUCTION with VALID Package\)](#)

最終変更 2000-11-28 火曜日 14:27 by REPORTSDEV

実行リンク PORTAL30_DEMO.FINANCIAL.show
PORTAL30_DEMO.FINANCIAL.show_parms

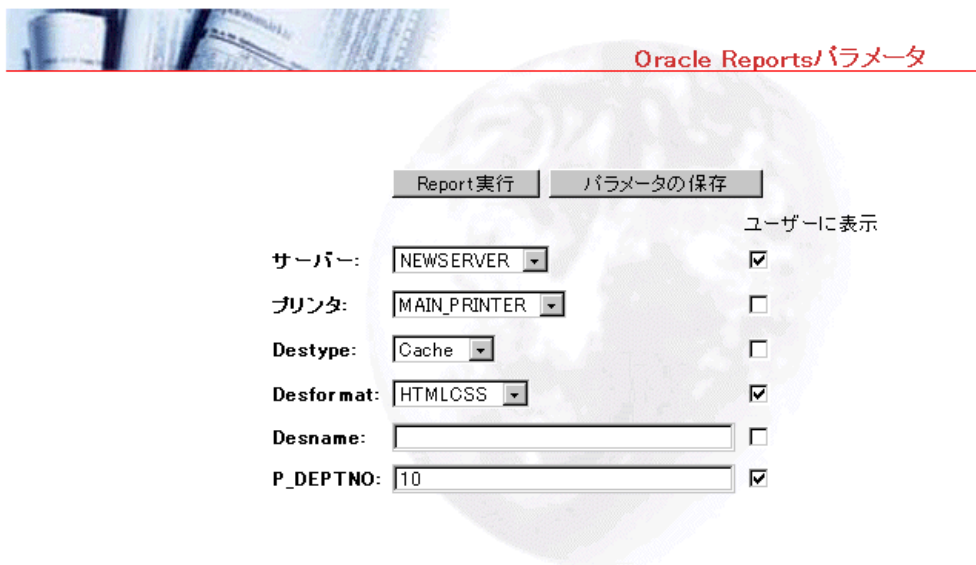
PL/SQLソース [パッケージ仕様](#) [パッケージ本体](#)

コール・インタフェース [表示](#)

[編集](#) [実行](#) [ポートレットとして実行](#) [カスタマイズ](#) [Add to Favorites](#) [情報](#) [削除](#)

21. 「カスタマイズ」リンクをクリックします。次の画面が表示されます。



Oracle Reports/パラメータ

Report実行 パラメータの保存

ユーザーに表示 ☒

サーバー: NEWSERVER ☐

プリンタ: MAIN_PRINTER ☐

Destype: Cache ☐

Desformat: HTMLCSS ☒

Desname: ☐

P_DEPTNO: 10 ☒

22. この画面では、「ユーザーに表示」チェックボックスで「Desformat」および「P_DEPTNO」を選択します。「P_DEPTNO」フィールドに「10」と入力します。
23. 「パラメータの保存」ボタンをクリックします。変更した内容を保存するには、必ずこのボタンをクリックしてください。変更を確認するための「パラメータを保存しました」という確認メッセージが表示されます。
24. 「Report 実行」ボタンをクリックします。次の画面が表示されます。

Database User Authentication

User Name:

Password:

Database:

Submit Cancel

25. 「User Name」フィールドに「SCOTT」、「Password」フィールドに「TIGER」、「Database」フィールドに「ORCL」と入力します。次に、「Submit」ボタンをクリックします。次の画面が表示され、レポートが正常に実行されたことを確認できます。



Report run on: August 2, 2000 11:19 AM

Department No	10	ACCOUNTING	Location	NEW YORK
Empno	Name	Job	Salary	Commission
7782	CLARK	MANAGER	\$2450.00	
7839	KING	PRESIDENT	\$5000.00	
7934	MILLER	CLERK	\$1300.00	

レポートの登録と REPORTSDEV、SCOTT および BJ への権限の付与が正常に終了しました (REPORTSDEV への権限の付与は自動的に行われます)。

5.14 レポートの配置

これでレポートを配置する準備ができました。

概要

この例では、Oracle Portal コンテンツ領域へのレポートの配置について説明します。

前提条件

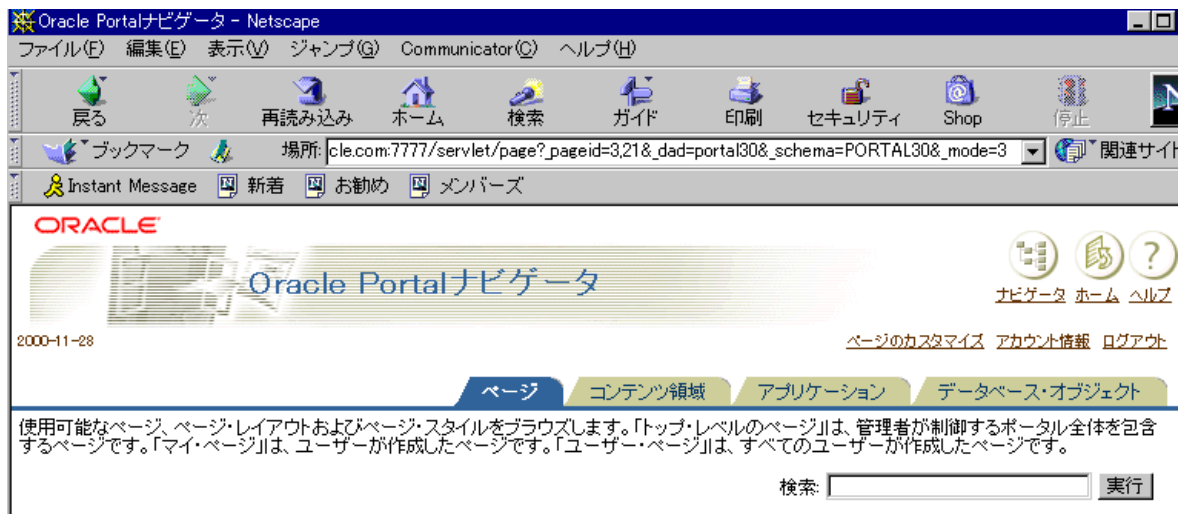
この例では、次の状況を前提としています。

- Oracle Portal は、Oracle Portal インストール用の適切な URL を使用してオープンされている。
- REPORTSDEV は Oracle Portal にログインされている。
- 「Reports Security Test」というコンテンツ領域がすでに存在している。

5.14.1 Oracle Portal コンテンツ領域へのレポートの配置

Oracle Portal コンテンツ領域にレポートを配置するには、次のステップを実行します。

1. 任意のページから「ナビゲータ」アイコンをクリックします。次の画面が表示されます。



2. 「コンテンツ領域」タブをクリックします。次の画面が表示されます。

2000-11-28

Oracle Portalナビゲータ

ナビゲータ ホーム ヘルプ

ページのカスタマイズ アカウント情報 ログアウト

ページ コンテンツ領域 アプリケーション データベース・オブジェクト

使用可能なコンテンツ領域とコンテンツ領域オブジェクトをブラウズしてください。共有オブジェクトは、個人用フォルダを含むすべてのコンテンツ領域間で共有されるオブジェクトです。

新規作成... [コンテンツ領域](#) 検索: 実行

パス: コンテンツ領域

タイプ ▲ ▼	名前 ▲ ▼	操作	所有者 ▲ ▼	更新 ▲ ▼	?
コンテンツ領域	Dev6i Publishing Reports	コンテンツ プロパティの編集 , 削除 , ルートフォルダの編集 , ルートフォルダのコピー	PORTAL30	22-Nov-2000	
コンテンツ領域	My Page	コンテンツ プロパティの編集 , 削除 , ルートフォルダの編集 , ルートフォルダのコピー	PORTAL30	16-Nov-2000	
コンテンツ領域	Oracle Portal Online Help	コンテンツ プロパティの編集 , 削除 , ルートフォルダの編集 , ルートフォルダのコピー	PORTAL30	15-Nov-2000	
コンテンツ領域	Reports Security Test	コンテンツ プロパティの編集 , 削除 , ルートフォルダの編集 , ルートフォルダのコピー	PORTAL30	28-Nov-2000	

3. 「Reports Security Test」をクリックします。次の画面が表示されます。

ようこそ REPORTSDEV

Reports Security Test (最終更新: 2000-11-29)

フォルダの編集 ?

サブフォルダ

[Oracle Reports Output](#) [Oracle Reports Status](#)

一般

4. 「一般」をクリックします。次の画面が表示されます。

ようこそ REPORTSDEV

Reports Security Test (最終更新: 2000-11-29)

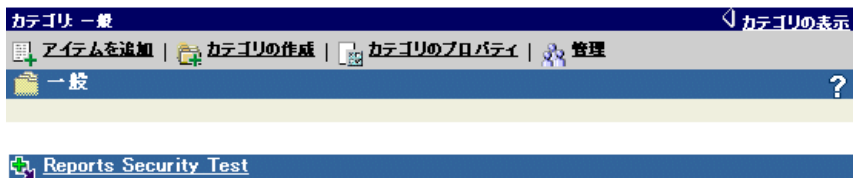
フォルダの編集 ?

サブフォルダ

[Oracle Reports Output](#) [Oracle Reports Status](#)

一般

5. 「カテゴリの編集」をクリックします。次の画面が表示されます。



6. 「アイテムを追加」をクリックします。次の画面が表示されます。



7. 「**アイテム・タイプ**」ドロップダウン・リストから「**アプリケーション・コンポーネント**」を選択します。その後、「**次へ**」ボタンをクリックします。次の画面が表示されます。



8. 「**フォルダ領域**」ヘッダーの下にある、適切なラジオ・ボタンをクリックします。この場合は、「**標準アイテム**」を選択します。
9. 「**プライマリ・アイテム属性**」ヘッダーまでスクロールして、「**アプリケーション・コンポーネント**」ドロップダウン・リストから「**EXAMPLE_APP: FINANCIAL**」を選択します。
10. 「**表示名**」フィールドに「**Financial report**」と入力します。
11. 「**フォルダ**」ドロップダウン・リストから「**Reports Security Test**」を選択します。
12. 「**カテゴリ**」ドロップダウン・リストから「**一般**」を選択します。

13. 「説明」フィールドに「Financial report for a department」と入力します。画面は次のようになります。

プライマリ アイテム 属性

アイテムがクリックされたときに実行されるコンポーネントを入力します。フォルダ領域に表示されるアイテムのリンク・テキストの表示名を入力します。アイテムの内容を最もよく表すカテゴリを選択し、説明を入力します。

アプリケーション・コンポーネント


表示名






フォルダ:

カテゴリ

説明

14. 「次へ」ボタンをクリックします。次の画面が表示されます。



[コンテンツ領域ホーム](#)
[管理](#)
[ナビゲータ](#)
[ホーム](#)
[ヘルプ](#)

アプリケーション・コンポーネント アイテムの追加
?

ステップ 3 / 3 >>>

パス FOLDER/REPORTSSECURITYTEST/

セカンダリ アイテム 属性

このアイテムのパスペクティブを1つまたは複数選択します。イメージ・ファイルの場所を入力し、イメージの配置を選択します。アイテムの検索に使用するキーワードを編集し、作成者名を編集します。アイテムに対して文書管理を設定し、チェックインおよびチェックアウトプロセスを通じてユーザー間でアイテムを共有できるようにします。また、表示モードでアイテムを非表示にすることも選択できます。

使用可能なパスペクティブ

パスペクティブ

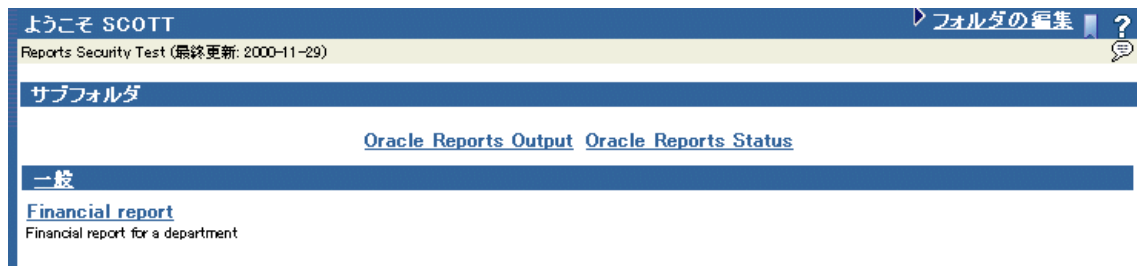
>>
>
<
<<

表示するパスペクティブ

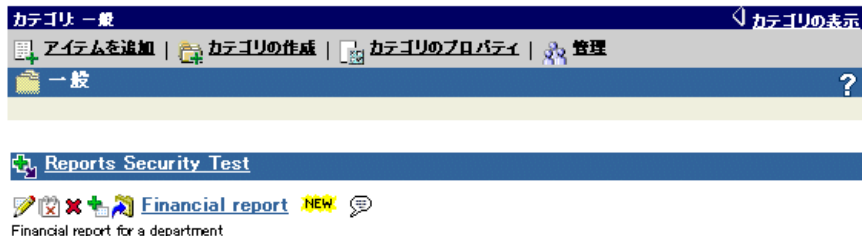
▲
▲
▼
▼

15. SCOTT がレポートを表示するには、次の 2 つの方法があります。

- 1 つ目の方法では、「Financial report」リンクをクリックします。このリンクは、「コンテンツ領域」にナビゲートした後、「Reports Security Test」にナビゲートすると見つかります。



- 2 つ目の方法では、SCOTT に追加オプションを与えます。ステップ 14 から続けて、「表示オプション」ヘッダーまでスクロールし、「パラメータ・フォームの表示」を選択します。その後、「完了」ボタンをクリックします。次の画面が表示され、レポートが追加されたことを確認できます。



16. 「Financial report」をクリックします。次の画面が表示されます。

Financial report.

Oracle Reports/パラメータ/スケジュール

パラメータ スケジュール

Report実行 パラメータの保存

Desformat: HTMLCSS

P_DEPTNO: 10

17. 「Report 実行」ボタンをクリックします。次の画面が表示され、レポートが正常に実行されたことを確認できます。



Report run on: August 2, 2000 11:19 AM

Department No	10	ACCOUNTING	Location	NEW YORK
Empno	Name	Job	Salary	Commission
7782	CLARK	MANAGER	\$2450.00	
7839	KING	PRESIDENT	\$5000.00	
7934	MILLER	CLERK	\$1300.00	

18. REPORTSDEV は、この時点でログアウトできます。

「FINANCIAL」というレポートが、Oracle Portal の「Reports Security Test」コンテンツ領域に正常に配置されました。

5.15 レポートの実行

REPORTSDEV によってレポートの登録、SCOTT と BJ へのレポート実行許可の付与、さらに「Reports Security Test」コンテンツ領域へのレポートの配置が完了したので、そのレポートは Oracle Portal ユーザーにより実行できる状態になっています。

概要

この例では、次の処理について説明します。

- レポートの検索
- レポートの実行

前提条件

この例では、次の状況を前提としています。

- BJ はログインされている。
- BJ は「Reports Security Test」コンテンツ領域にナビゲートしている。
- BJ は「Financial」というレポートを実行しようとしている。

レポートを実行するには、次のステップに従います。

1. 「**コンテンツ領域**」タブから「Reports Security Test」を選択します。次の画面が表示されます。



2. 「Financial report」をクリックします。次の画面が表示されます。

Financial report.

Oracle Reports/パラメータ/スケジュール

パラメータ スケジュール

Report実行 パラメータの保存

Desformat: HTMLCSS

P_DEPTNO: 10

3. 「スケジュール」タブをクリックします。次の画面が表示されます。

パラメータ スケジュール

Submit

開始

☒ 即時

☐ 時間指定 時 分 午前/午後 月 日 年(YYYY)

10 35 午前 10月 03 2000

繰返し

☒ 一度だけ

☐ 毎 時(間)

☐ 毎 第1 月曜日 (毎月)

☐ 毎月の最終 月曜日 (ただし 01 日以前)

☐ 再試行 回 時(間) 後(失敗してから)

宛先

コンテンツ領域 Reports Security Tes

ログ・ファイル・フォルダ

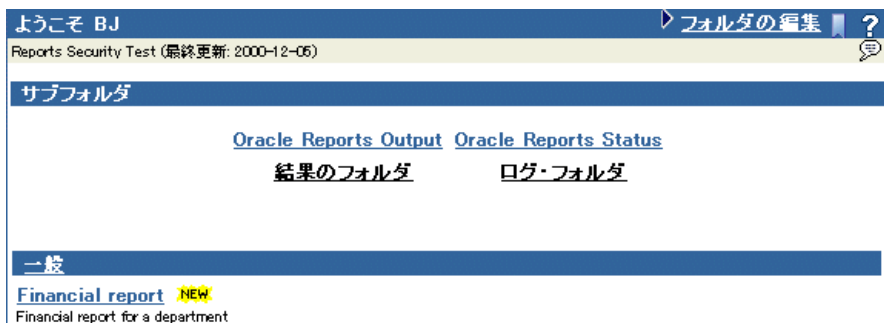
結果のタイトル Oracle Reportの結果

この画面をスクロールして、次の情報を設定します。

開始	「即時」を選択します。
繰返し	「毎」を選択して、空白のフィールドに「1」と入力します。これにより、1 時間ごとにレポートが実行されます。
宛先	次のように設定します。
コンテンツ領域	「Reports Security Test」がすでに入力されています。
ログ・ファイル・フォルダ	「ログ・ファイル」と入力します。このフォルダには、実際のレポートの実行に関する情報が含まれています。
結果のタイトル	「Financial information」と入力します。
結果のフォルダ	「結果のフォルダ」と入力します。レポート出力はこのフォルダに配信されます。
有効期間	ドロップダウン・リストから「永久」を選択します。
前回の結果を上書き	これをクリックすると、前回の結果は上書きされます。クリックしなければ、前回の結果が保存されます。

4. 「Submit」ボタンをクリックします。

「ようこそ BJ」フォルダ・ページに戻り、「**結果のフォルダ**」が作成されたことを確認します。レポートの結果は 1 時間ごとにこのフォルダに格納されるので、BJ はフォルダをクリックすればいつでも結果を確認できます。



5.16 Oracle Portal 以外のレポートの公開

Oracle Portal リポジトリ内に登録したレポートを公開する際、配置メカニズムとして Oracle Portal の使用が強制されているわけではありません。任意の Web コンテンツ領域または Oracle Portal コンテンツ領域に、レポート・リクエストの起動元となるリンクを作成できます。

次のステップでは、Oracle Portal コンテンツ領域の外部に、アプリケーション・レベルのセキュリティをインプリメントする方法について説明します。

1. アクセス制御情報が存在するインスタンスを指す `rep60_<machinename>.ora` ファイルに、`SECURITYTNSNAME` パラメータおよび `PORTALUSERID` パラメータを追加する必要があります (`PORTALUSERID=<portal_username>/<portal_password>` および `SECURITYTNSNAME=<tnsnames>`)。このエントリがない場合、Oracle Reports Services では、Oracle Portal リポジトリに存在するアクセス制御情報に対してユーザーが認証されません。
2. CGI またはサーブレットのキー・マップ・ファイル (`cgicmd.dat`) では、`AUTHID` コマンドライン・パラメータが使用されます。`AUTHID` パラメータはレポート・リクエストのコマンドラインに追加できます。`AUTHID` により、Oracle Portal で作成されたアプリケーション・ユーザーが識別されます。キー・マップ・ファイルの任意のレポート・エントリに `AUTHID=username/password` を追加して、ユーザー名とパスワードをハードコードできます。この場合は、ユーザーに対して、アプリケーション・レベルのセキュリティ用プロンプトが表示されます。

ユーザーがレポート・リクエストを実行するたびに認証されるようにプロンプトを表示する場合は、`SHOWAUTH` エントリおよび `AUTHTYPE=S` エントリを URL に追加するか、または、`%S` 引数をキー・マッピング・レポート・エントリに含めます。キー・マップ・ファイルのエントリの一例を次に示します。

```
emp: server=repserver report=emp.rdf userid=scott/tiger@orcl destype=CACHE
desformat=HTMLCSS %S
```

この例では、データベースのユーザー名、パスワードおよび接続文字列 (scott/tiger@orcl) がハードコードされているため、ユーザーに必要とされるのは、アプリケーション・レベルのセキュリティ用のユーザー名とパスワードを入力することだけです。つまり、Oracle Portal に格納されているユーザー名とパスワードは、選択したレポートを実行するアクセス権があるかどうかを識別するために使用されます。次に示すように、システム認証のダイアログ・ボックスには「User Name」と「Password」フィールドのみ表示されます。

SYSTEM USER AUTHENTICATION

User Name:

Password:

このように、システム認証のダイアログ・ボックスには「User Name」と「Password」フィールドのみ表示されます。キー・マッピング・エントリから USERID パラメータを省略した場合は、次の画面に示すように、データベース認証のためのプロンプトも表示されます。

Database User Authentication

User Name:

Password:

Database:

データベース認証を実施する別の方法としては、キー・マッピング・エントリに %D 引数を追加する方法もあります。次に例を示します。

```
emp: server=repserver report=emp.rdf destype=CACHE desformat=HTMLCSS %S %D
```

「Please do not edit this line」というコメント・タグが含まれていない部分については、認証テンプレート・ファイルを編集できます。

3. REPORTS60_COOKIE_EXPIRE の値を指定します。ユーザーがアプリケーションに正常にログインして、レポート・リクエストを実行するたびに、暗号化された cookie がブラウザに送られます。cookie が期限切れになると、その後のリクエストを実行するには、ユーザーは再度認証される必要があります。管理者は、サーバーに REPORTS60_COOKIE_EXPIRE 環境変数を定義できます。この変更内容を有効にするには、サーバーを停止して再起動する必要があることに注意してください。

クライアントのブラウザがクローズされると cookie は自動的に削除されます。しかし、ブラウザを長時間オープンしておく場合は、この環境変数によってセッションの長さを制御できます。Oracle Reports Services CGI 実行可能ファイルは、ジョブ・リクエストを受け取ったときに、cookie に保存されている時間を現在のシステム時間と比較します。その時間が環境変数に定義した分単位の時間（デフォルトは 30 分）よりも長い場合、cookie が拒否され、ユーザーは認証のために再度識別が要求されます。

ユーザー認証に適用する環境変数のリストは、[表 5-1 「ユーザー認証に関する環境変数」](#)を参照してください。

cookie の期限が切れるまでユーザーの認証を維持するには、キー・マップ・ファイルから AUTHID のパラメータまたは %S 引数を省略します。

Oracle Reports Services サーバーの クラスタの構成

この章では、Oracle Reports Services サーバーをクラスタ構成にして、パフォーマンスの向上と負荷の均衡化を図る方法について説明します。これは、急速に増加するユーザーへの情報の配布がますます要求されるにつれ、重要になります。

Oracle Reports Services サーバーのクラスタリングを使用すると、追加のアプリケーション・サーバーを必要に応じてプラグインできるため、所属企業の既存のハードウェア投資を活用してこのニーズに対応できます。これによって、Oracle Reports Services サーバーの処理能力を企業の成長にともなって成長させることができます。

クラスタリングに向けて Oracle Reports Services サーバーを構成する前に、Oracle Reports Services の基本的なアーキテクチャについて知っておく必要があります。詳細は、[第 1 章「公開のアーキテクチャと概念」](#)を参照してください。また、基本構成を使用して Oracle Reports Services のセットアップを完了しておくことも必要です。詳細は、[第 3 章「Windows NT および UNIX 上での Oracle Reports Services サーバーの構成」](#)を参照してください。

6.1 クラスタリングの概要

Oracle Reports Services サーバーとして構成された 3 つのマシンをクラスタ化すると仮定します。これらのマシンは次のとおりです。

表 6-1 サーバー・マシン例の説明

マシン / サーバーの TNS 名	説明	マスター / スレーブ
NT-1	4 CPU NT サーバー	マスター
NT-2	2 CPU NT サーバー	スレーブ
SUN-1	2 CPU Sun Solaris ワークステーション	スレーブ

注意： ここでは説明のために、NT-1 マシンをマスター・サーバーに決定しました。CPU の数は決定要因ではありません。

この概要で説明する、Oracle Reports Services サーバーをクラスタ構成にする方法についてのステップ・バイ・ステップの解説は、[6.2 項「Oracle Reports Services サーバーをクラスタ構成にする例」](#)を参照してください。

NT-1 をマスターに指定し、次に、CLUSTERCONFIG パラメータを設定して、このサーバーが NT-2 と SUN-1 をスレーブとして認識できるようにします。この例をわかりやすくするために、マスター・サーバーと各スレーブ・サーバーに対する MAXENGINE パラメータと MINENGINE パラメータには、各マシン上で使用できる CPU 数が設定されています。

マシンの構成が完了したら、レポート・リクエストを、レポートをスレーブに転送するマスター・サーバー (SERVER=NT-1) に送ります。マスター・サーバーは起動すると構成ファイルをチェックします。マスターはスレーブ・サーバーが構成ファイルにリストされている順序で各スレーブ・サーバーに接続し、定義された数のエンジン (たとえば、各 2 つずつ) を起動するよう通知します。スレーブのエンジンが起動すると、マスターのコントロール下に置かれ、マスターはラウンドロビン・アルゴリズムを使用してジョブを各エンジンに割り当てます。

マスター・サーバー (NT-1) は 7 つのレポート・リクエストを受け取るとします。マスターは自分の 4 つのエンジンを使用して最初の 4 つのレポートを実行します。第 5 と第 6 のレポートについては、マスターはリクエストを NT-2 の 2 つのエンジンに転送して実行させます。マスターは第 7 のレポートを受け取ると、そのリクエストを SUN-1 の第 1 のエンジンに転送して実行させます。すべての出力はセントラル・キャッシュ (すべてのサーバーが共有するキャッシュ) に書き込まれます。マスターは出力をリクエスト元 (Web ブラウザなど) に戻します。

スレーブ・サーバーは、マスター・サーバーから独立してエンジンを起動できるのであれば、当然、Oracle Reports Services サーバーとして完全に機能できる状態を維持できます。NT-2 の Oracle Reports Services 構成の MAXENGINE パラメータと MINIENGINE パラメータ

タが3に設定されているとします。これは、NT-2のOracle Reports Services サーバーに3つのエンジンが割り当てられていて、マスターの関知しないところでリクエストを受け取れることを意味しています。スレーブ・サーバーとして構成される場合は（すなわち、NT-2のマスター構成でMAXENGINEパラメータとMINENGINEパラメータが2に設定されている場合）、NT-2のOracle Reports Services では合計5つのエンジンが起動していることになります。NT-2サーバー専用のエンジンが3つとマスターへのスレーブ専用のエンジンが2つです。

6.2 Oracle Reports Services サーバーをクラスタ構成にする例

この項では、Oracle Reports Services サーバーをクラスタ構成にする方法をステップ・バイ・ステップで説明します。この例では、次の事項について説明します。

- マスターとスレーブ間の通信を可能にする方法
- マスター・サーバーを構成する方法
- クラスタ化されたサーバーへのレポート・リクエストを実行する方法
- エンジンがダウンしたときにジョブを再送信する方法
- サーバーに既存の構成を追加する方法

この例では、表 6-1「サーバー・マシン例の説明」で説明するとおり、サーバー・マシンをクラスタリングに向けて構成します。

各マシンに対しては次の事項が前提とされています。

- Oracle Reports Services のコンポーネントがインストールされている。
- Oracle Reports Services は、tnsnames.ora ファイルでマシン名を TNS サービス・エントリ名（NT-1 など）として使用して構成されている。
- セントラル・ファイル・サーバーが稼動していて、2つのディレクトリが設定されている。ソース・ディレクトリ（レポート定義ファイルが格納されている）とキャッシュ・ディレクトリ（キャッシュされたすべてのレポートの送信先）の2つです。

すべてのエンジンが出力をセントラル・キャッシュに書き込み、すべてのエンジンがセントラル・ソース・ディレクトリからレポート定義を読み込みます。セントラル・ソース・ディレクトリは、すべてのエンジンが同じレポートを実行していることを保証します。これによって、更新されたレポート定義ファイルを様々な場所にコピーする必要もありません。セントラル・キャッシュは、マスター・サーバーが重複したジョブをサブして、指定された許容時間内に、各スレーブ・サーバーのローカル・ディスクに送らずに実行できるようにします。

- すべてのエンジンは、プリンタに対する同じエイリアスを認識する（出力が常にデフォルトのプリンタに送られている場合ではない限り）。

6.2.1 マスターとスレーブ間の通信を可能にする方法

NT-1 マシン（マスター）上で、テキスト・エディタによって ORACLE_HOME¥NET80 ディレクトリにある tnsnames.ora をオープンして、次の行を追加します。nt-2.world と sun-1.world はサーバー・インスタンスの名前で、.world は sqlnet.ora ファイルの NAMES.DEFAULT_DOMAIN の設定で指定されているドメインです。NAMES.DEFAULT_DOMAIN 設定が sqlnet.ora ファイルで定義されていない場合は、サーバー・インスタンスの名前から .world を省略してください。

```
nt-2.world=(ADDRESS=(PROTOCOL=tcp) (HOST=nt-2) (PORT=1949))
sun-1.world=(ADDRESS=(PROTOCOL=tcp) (HOST=sun-1) (PORT=1949))
```

NT-2 マシン（スレーブ）上で、次のようにします。

1. テキスト・エディタで、ORACLE_HOME¥NET80 ディレクトリにある tnsnames.ora をオープンして、次の行を追加します。nt-1.world はサーバー・インスタンスの名前で、.world は sqlnet.ora ファイルの NAMES.DEFAULT_DOMAIN の設定で指定されているドメインです。NAMES.DEFAULT_DOMAIN 設定が sqlnet.ora ファイルで定義されていない場合は、サーバー・インスタンスの名前から .world を省略してください。

nt-1.world=(ADDRESS=(PROTOCOL=tcp) (HOST=nt-1) (PORT=1949))
2. ORACLE_HOME¥REPORT60¥SERVER ディレクトリの nt-2.ora (Oracle Reports Services の構成ファイル) をオープンして、INTEGINE パラメータを 0 に設定します。こうすることで、起動時に作成されたエンジンのみがマスターによって起動されることが保証されます。
3. SUN-1 サーバー・マシン上でステップ 1 とステップ 2 を繰り返します。ステップ 2 で、sun-1.ora 構成ファイルを編集します。

6.2.2 マスター・サーバーを構成する方法

この項では、次の設定を使用してマスターを構成します。

- マスター・サーバーの構成ファイルを編集して、マスターに対するスレーブ・サーバーを識別し、またマスター・サーバーと対応付けられたエンジンの数をコントロールします。
- 次の項目を定義するマスター・サーバーの構成ファイルのパラメータを設定します。
 - キャッシュとソース・ディレクトリを識別するエンジン設定が定義されます。
 - このマシン上には CPU が 4 つあるので、4 つのローカル・エンジンを使用して、サーバーとして同時に起動します。
 - この 4 つのエンジンは 60 分間アイドル状態が続くと停止され、50 個のジョブを実行してから再起動されます。
 - サーバーと一度に通信できるプロセスの数は、最大数の 40% に設定されます。

- CLUSTERCONFIG パラメータを設定してマスターに対するスレーブ・サーバーを識別します。この例では、マスターの起動時に各スレーブ・サーバー上の2つのエンジンを起動します。

マスター・サーバーのエンジンに対する ENGINE パラメータと MAXIDLE パラメータはすべてのスレーブ・エンジンに対して提示されます。MAXIDLE パラメータの単位は分で、ENGLIFE パラメータの単位はエンジン数です。

NT-1 サーバー・マシン (マスター) 上で、次のようにします。

1. ORACLE_HOME¥REPORT60¥SERVER ディレクトリの nt-1.ora (Oracle Reports Services の構成ファイル) をオープンします。
2. 構成ファイルを次の設定どおりに編集します。

```
maxconnect=4096
sourcedir="X:¥Source"
cachedir="X:¥Cache"
cachesize=50
minengine=0
maxengine=4
initengine=4
maxidle=60
englife=50
```

NT-1 マシンは「X: ドライブ」上のセントラル・サーバーにマップされます。

3. 構成ファイルを次の設定どおりに編集します。

```
clusterconfig="(server=nt-2
minengine=0
maxengine=2
initengine=2
cachedir="W:¥Cache")
(server=sun-1
minengine=0
maxengine=2
initengine=2
cachedir="/share/Cache") "
```

各項目の内容は次のとおりです。

<code>server</code>	スレーブ・サーバーの TNS サービス・エントリ名。
<code>minengine</code>	このマスター・サーバーがレポートの実行に使用するために持つべきランタイム・エンジンの最小数。
<code>maxengine</code>	このマスター・サーバーがレポートの実行に使用するために持つランタイム・エンジンの最大数。
<code>initengine</code>	このマスター・エンジンが起動したランタイム・エンジンの初期数。
<code>cachedir</code>	このマスター・サーバーに対するセントラル・キャッシュ・ディレクトリ。

使用上の注意

マスター・サーバーを構成する場合は、次の点を考慮する必要があります。

- 各スレーブの定義は括弧に入れる必要があります。
- NT マシンと UNIX マシンのキャッシュ・ディレクトリの設定は異なります。すべてのサーバーが共有ファイル・システムをこれと同じ定義（つまり、マスターは「X: ドライブ」に、スレーブは「W: ドライブ」にマップされるという）によって認識する必要はありません。
- スレーブ・サーバーの `REPORTS60_PATH` 環境変数には、`/share/Source`（SUN-1 サーバー・マシンの場合）と `W:¥Source`（NT-2 マシンの場合）が設定されている必要があります。
- マスター・サーバーをシャットダウンして再起動し、マスター・サーバーが新しい構成を認識できるようにします。

これで構成作業は完了です。8 つのエンジンがマスター・サーバーの起動時に起動します。

6.2.3 クラスタ構成でレポートを実行する方法

クラスタリングとして構成された Oracle Reports Services サーバーに対するレポート・リクエストを実行するには、SERVER コマンドライン引数 (SERVER=NT-1) で、シン・クライアントの実行ファイルに対する他の関連引数とともにマスター・サーバーを指定します。マスター・サーバーは送られてくるジョブをスレーブ・サーバー上のエンジンに割り当てます。

REPORTS60_REPORTS_SERVER 環境変数にマスター・サーバーを設定すると、SERVER コマンドライン引数を省略できます。詳細は、[付録 D「環境変数」](#)を参照してください。

使用できる様々なレポート・リクエスト・メソッドの詳細は、[第 4 章「レポート・リクエストの実行」](#)を参照してください。

レポート・リクエストをサーバー・クラスタに提出する上で問題がある場合は、[6.2.4 項「エンジンがダウンしたときにジョブを再送信する方法」](#)を参照してください。

マスター・サーバーのジョブは、Queue Manager の Oracle Reports Services Queue Viewer を使用して監視できます。詳細は、Oracle Reports Services Queue Manager のオンライン・ヘルプを参照してください。

6.2.4 エンジンがダウンしたときにジョブを再送信する方法

レポートの実行中にエンジンがダウンしたら、SCHEDULE コマンドライン引数で定義されている Retry の設定によって、そのジョブを再実行するかどうかが規定されます。Retry の設定が指定されていないときは、そのジョブは失われます。ただし、このジョブの失敗はサーバーのログ・ファイルに記録され、Queue Manager のジョブのリストに表示されます。コマンドラインに再試行の設定が含まれているときは、マスター・サーバーは次の使用可能なエンジンを使用してそのジョブを再実行します。

SCHEDULE のコマンドライン引数で「Retry」オプションを 2 に設定してジョブを送信したと仮定します。マスター・サーバーはレポート・リクエストを NT-2 サーバー上の第 2 のスレーブ・エンジン上で開始します。ただし、NT-2 の一時領域は不足するのでジョブは中止されます。するとマスター・サーバーはジョブを再送信します。他のジョブが割り当てられていないとすると、このジョブは SUN-1 サーバーの第 1 のエンジンに割り当てられます。

再試行オプションはフェイルオーバー・サポートを提供する点で便利ですが、慎重に使用してください。たとえば、再試行の回数を大きな値に設定しても問題が解決するとは限りません。再送信されたジョブは、根本的な問題がエンジンではなくレポートそのものにある場合は、必ず失敗します。

6.2.5 他のスレーブ・サーバーをマスターに付加する方法

既存のクラスタ構成に次の表で定義される別のスレーブ・サーバーを追加する場合があります。

表 6-2 追加のサーバーの説明

マシン / サーバーの TNS 名	説明	マスター / スレーブ
SUN-2	4 CPU Sun Solaris サーバー	スレーブ

この例では、このマシンはすでに Oracle Reports Services サーバーとして構成されているものとします。Oracle Reports Services サーバーに対する TNS サービス・エントリ名はマシン名です。

SUN-2 サーバー・マシン（スレーブ）上で、ORACLE_HOME¥REPORT60¥SERVER ディレクトリにある sun-2.ora（Oracle Reports Services の構成ファイル）をオープンして次の行を追加します。nt-1.world はサーバー・インスタンスの名前で、.world は sqlnet.ora ファイルの NAMES.DEFAULT_DOMAIN 設定で指定されているドメインです。NAMES.DEFAULT_DOMAIN 設定が sqlnet.ora ファイルで定義されていない場合は、サーバー・インスタンスの名前から .world を省略してください。

```
nt-1.world=(ADDRESS=(PROTOCOL=tcpx) (HOST=nt-1) (PORT=1949))
```

NT-1 サーバー・マシン（マスター）上で、次のようにします。

1. ORACLE_HOME¥NET80¥ADMIN ディレクトリにある tnsnames.ora をオープンして、次のエントリを追加します。sun-2.world はサーバー・インスタンスの名前で、.world は sqlnet.ora ファイルの NAMES.DEFAULT_DOMAIN の設定で指定されているドメインです。NAMES.DEFAULT_DOMAIN 設定が sqlnet.ora ファイルで定義されていない場合は、サーバー・インスタンスの名前から .world を省略してください。

```
sun-2.world=(ADDRESS=(PROTOCOL=tcpx) (HOST=sun-1) (PORT=1949))
```

2. nt-1.ora (Oracle Reports Services の構成ファイル) をオープンして、次の太字のテキストを既存の CLUSTERCONFIG パラメータに追加します。

```
clusterconfig="(server=nt-2
minengine=0
maxengine=2
initengine=2
cachedir="W:¥Cache")
(server=sun-1
minengine=0
maxengine=2
initengine=2
cachedir="/share/Cache")
(server=sun-2
minengine=0
maxengine=4
initengine=4
cachedir="/share/Cache")"
```

3. マスター・サーバーをシャットダウンして再起動し、マスター・サーバーが新しく構成されたスレーブ・サーバーを認識できるようにします。

SUN-2 マシンをスレーブ・サーバーとして構成している間に別の管理者が NT-2 マシンにアクセスしたとします (たとえばバックアップを行うため)。NT-2 マシンがまだダウン状態のときに、Oracle Reports Services サーバーを NT-1 マシン上で再起動します。NT-1 マシンは 2 つの Sun マシン上でスレーブ・エンジンを起動できましたが、NT-2 マシンはダウンしているため NT-2 マシン上のスレーブ・エンジンを起動できませんでした。

NT-1 サーバーはすべてのスレーブ・マシンのポーリングを行っているため、NT-2 マシンが回復して Oracle Reports Services が再起動すると NT-2 マシンは NT-1 サーバーによって自動的に検出されます。4 つのスレーブ・エンジンは、起動するとマスターからジョブを受け取れるようになります。

実行時のレポートのカスタマイズ

Oracle Reports Services では、XML タグで作成したレポート定義の実行および他のレポート定義とのマージができます。旧リリースでは、Oracle Reports Services で実行するレポートは、Oracle Reports Services Builder で作成し、保存することが必要でした。リリース 6i では、XML タグを使用してレポート定義を作成できます。XML レポート定義は、単独でも実行できますが、実行時に他のレポートに適用することで、各利用対象者にあわせて出力をカスタマイズできます。

XML レポート定義を使用して、次のことができます。

- オリジナルのレポートを変更せずに、実行時にレポートをカスタマイズできます。様々な XML レポート定義を作成して適用することで、各ユーザーまたはユーザー・グループ単位でレポートの出力を変更できます。この機能の利点は、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る点です。
- 既存レポートにバッチ更新を適用します。XML レポート定義を他のレポートに適用した場合は、結合した定義をファイルに保存できます。したがって、XML レポート定義を使用して、既存レポートのバッチ更新ができます。この利点は、ファイルを 1 つずつ Oracle Reports Services Builder でオープンして手動で変更せずに、多数のレポートを素早く更新できる点です。
- 完全なレポート定義を XML で作成します。この利点は、Oracle Reports Services Builder を使用せずに、レポートをすぐに作成できることです。XML タグを生成すると、レポート定義を作成し、Oracle Reports Services で実行することが可能になります。

7.1 概要

XML タグを使用して、完全または部分的なレポート定義を作成し、カスタマイズ・ファイルまたは完全な自己完結型レポートとして実行できます。完全なレポート定義は、すべてのデータ・モデルとレイアウトが XML で指定されており、単独で実行することも、カスタマイズのために他のレポートに適用することもできます。部分定義は、少量の情報で構成されており、他のレポートと結合した場合のみ使用できます（すなわち、単独では実行できません）。

カスタマイズ・ファイルは、既存レポート（RDF または XML）に適用されるレポート定義です。カスタマイズ・ファイルにより、フィールドの日付書式マスクまたはバックグラウンド・カラーなどの既存レポート・オブジェクトの一部の特性を変更できます。また、カスタマイズ・ファイルを使用して、まったく新しいオブジェクトを他のレポートに追加することもできます。カスタマイズ・ファイルは、完全レポート定義または部分レポート定義のどちらでも構いません。

XML レポートを単独で実行するには、完全なレポート定義が必要です。自己完結型 XML レポートとは、他のレポートに適用されずに単独で実行されるレポートです。

7.1.1 XML レポート定義の作成および使用方法

次の各ステップで、XML レポート定義の作成および使用方法を簡単に説明します。

1. [7.5 項「XML タグ・リファレンス」](#)で説明されている XML タグを使用して、完全または部分的なレポート定義を作成します。この定義は、エディタを使用して手動で作成するか、またはプログラムで作成できます。¹ 次に、部分レポート定義のサンプルを示します。

```
<report name="emp" DTDVersion="1.0">
  <layout>
    <section name="main">
      <field name="f_sal" source="sal" textColor="red"/>
      <field name="f_mgr" source="mgr" fontSize="18" font="Script"/>
      <field name="f_deptno" source="deptno" fontStyle="bold"
        fontEffect="underline"/>
    </section>
  </layout>
</report>
```

¹ プログラムで定義を作成する場合は、ユーザー入力に基づいて短時間でレポート定義を作成できます。

このサンプルを他のレポートに適用すると、一部のフィールドの書式設定が変更されます。この XML には完全なレポート定義が含まれていないので、単独では実行できません。データ・モデル定義がなく、部分的なレイアウト定義のみが含まれています。単独で実行するには、完全なデータ・モデルおよびレイアウト定義が必要です。

このステップの詳細は、[7.2 項「XML レポート定義の作成」](#)を参照してください。

2. XML レポート定義を Oracle Reports Services にアクセス可能な場所に格納します。²
3. XML レポート定義を他のレポートに適用します (CUSTOMIZE コマンドライン引数または PL/SQL のビルトイン SRW.APPLY_DEFINITION を使用)。または、XML レポート定義を単独で実行します (REPORT コマンドライン引数を使用)。

このステップの詳細は、[7.3 項「XML レポート定義の実行」](#)を参照してください。

この章の残りの部分では、XML レポート定義の作成および使用手順を詳しく説明し、また、定義を作成するときに使用する XML タグのリファレンスを掲載します。

7.2 XML レポート定義の作成

XML レポート定義の作成方法を理解する最も簡単な方法は、必要なタグのみを用意することから開始し、次に部分定義の作成、最後に完全な定義 (RDF ファイルなしに実行できるもの) へと作業を進めることです。この項では、次の XML 定義について説明します。

■ [7.2.1 項「必須タグ」](#)

XML タグには、部分的または完全にかかわらず、XML で作成するすべてのレポート定義に必要なタグがあります。この XML レポート定義では、レポート定義の正しい解析のために最小限必要な XML タグ・セットを示します。

■ [7.2.2 項「部分レポート定義」](#)

この種類の XML レポート定義は、完全なレポート定義より少ない情報で構成されます。したがって、カスタマイズ・ファイルとして他のレポートに適用することのみ可能です。単独では実行できません。

■ [7.2.3 項「完全なレポート定義」](#)

この種類の XML レポート定義は、完全なレポート定義を含みます。したがって、RDF ファイルに適用することも、単独で実行することもできます。

² XML レポート定義は、Oracle Reports Services Runtime および Oracle Reports Services Builder でも使用できます。

7.2.1 必須タグ

完全または部分的にかかわらず、すべての XML レポート定義には、次のタグ・ペアが必要です。

```
<report></report>
```

たとえば、最小限の XML レポート定義は次のとおりです。³

```
<report name="emp" DTDVersion="1.0">
</report>
```

`<report>` タグは、レポートの開始、レポート名およびこの XML レポート定義で使用する文書タイプ定義 (DTD) ファイルのバージョンを示します。⁴ `</report>` タグは、レポート定義の終わりを示します。

完全なレポート定義には、データ・モデルとレイアウトの両方が必要です。したがって、次のタグとその内容を定義する必要があります。

- `<data></data>`
- `<layout></layout>`

7.2.2 部分レポート定義

XML レポート定義の主な使用方法の 1 つは、実行時に他のレポートに変更を加えることです。XML レポート定義は、オリジナル・レポートに永続的な影響を与えずに、実行時に他のレポートのデータ・モデルまたは書式を簡単に変更できます。⁵ その利点は、1 つのレポートを複数の利用対象者に使用できることです。たとえば、RDF ファイルを 1 つ作成し、それに様々な部分 XML レポート定義を適用することで、各種利用対象者にあわせてファイルをカスタマイズできます。XML レポート定義には、いくつかのオブジェクトの外観を変更する少数のタグのみで構成された簡単なものから、レポートのすべてのオブジェクトに作用したり、新規オブジェクトを追加するような複雑なものまであります。

カスタマイズ・ファイルで実行可能な変更の種類を理解するには、いくつかの例を見ると役立ちます。『レポート作成ガイド』では、Oracle Reports Services Builder を使用していくつかのサンプル・レポートを作成する方法を説明しています。これらのレポート用の完成した RDF ファイルは、`ORACLE_HOME\TOOLS\DOC60\US\RBRR60` ディレクトリにあります。この章のために、これらのレポートのいくつかを変更する XML レポート定義が、RDF ファイ

³ この XML レポート定義には何も含まれていないので、他のレポートに適用しても効果はいいません。必要なタグがあるので解析はできますが、この定義は必須タグを理解するために使用しています。

⁴ DTD ファイルとは、XML タグに意味を持たせるファイルです。Oracle Reports Services には、レポート定義で使用する XML タグを定義した DTD ファイルが含まれています。サポートされる XML タグの詳細は、7.5 項「XML タグ・リファレンス」を参照してください。

⁵ RDF ファイルと XML レポート定義を結合して、新規 RDF ファイルとして保存できる点に注意してください。その方法は、この章で後述します。

ルと一緒にこのディレクトリに入っています。次の表で、これらの XML レポート定義を 1 つずつ詳しく説明します。

表 7-1 レポート作成用の XML レポート定義

XML ファイル	PDF ファイル	説明
cond.xml	cond.rdf	<p>cond.xml による変更:</p> <ul style="list-style-type: none"> ■ F_trade_date の書式マスクを MM/DD/RR に変更。 ■ F_Mincurrent_pricePersymbol および F_Maxcurrent_pricePersymbol の塗りカラーを変更。 <p>cond.xml による追加:</p> <ul style="list-style-type: none"> ■ HTML 出力生成時にレポート・エスケープの HTML を挿入。 <p>詳細は、7.2.2.1 項「書式変更の例」を参照してください。</p>
temp.xml	temp.rdf	<p>temp.xml による変更:</p> <ul style="list-style-type: none"> ■ F_high_365 および F_low_365 のフィールド・ラベル。 <p>temp.xml による追加:</p> <ul style="list-style-type: none"> ■ 10 より大きい値をハイライトするために、書式例外を F_p_e に追加。 ■ 10 より大きい値をハイライトするために、書式例外を F_p_e1 に追加。 <p>詳細は、7.2.2.2 項「書式例外の例」を参照してください。</p>
sect.xml	sect.rdf	<p>sect.xml による追加:</p> <ul style="list-style-type: none"> ■ レポートにプログラム単位を追加。 ■ レポートのメイン・セクションの詳細レコードにリンク先を追加。 ■ メイン・セクションの詳細レコードに、ヘッダー・セクションの従業員サマリーからのハイパーリンクを追加。 <p>詳細は、7.2.2.3 項「プログラム単位とハイパーリンクの例」を参照してください。</p>

表 7-1（続き）レポート作成用の XML レポート定義

XML ファイル	PDF ファイル	説明
ref.xml	ref.rdf	ref.xml による追加： <ul style="list-style-type: none">■ 新規問合せ Q_summary をデータ・モデルに追加。■ 新規問合せ Q_summary のデータを使用するレポートにヘッダー・セクションを追加。 詳細は、 7.2.2.4 項「データ・モデルと書式の変更の例」 を参照してください。

XML のカスタマイズを適用するには、引数を 1 つ追加して RDF ファイルを実行します。次に例を示します。

```
rwrun60 userid=scott/tiger report=cond.rdf
      customize=e:¥orant¥tools¥doc60¥us¥rbrbr60¥cond.xml
```

詳細は、[7.3 項「XML レポート定義の実行」](#)を参照してください。

これらの RDF ファイルを使用して、カスタマイズ・ファイルを指定した場合と、指定しなかった場合とで実行結果を比較します。次項では、これらの変更を行うために使用される XML を詳しく説明します。

7.2.2.1 書式変更の例

cond.xml ファイルの XML は、cond.rdf ファイルの書式の基本的な特性の一部を変更し、HTMLCSS 出力生成時にレポートの先頭および終わりに挿入される HTML コードを追加します。

この例のヒント

次のヒントは、この例を見る場合に役立ちます。

- この例では、<report> タグの名前属性が、RDF ファイルの名前と一致しています。異なる名前を使用しても構いません（たとえば、condnew など）。
- <field> および <section> タグの名前属性は、RDF ファイルに存在するフィールドおよびセクションの名前と一致しています。したがって、<field> タグのその他の属性は、RDF ファイルに定義されたレイアウトのメイン・セクション内の既存フィールドに適用されます。
- <customize> タグの内側のコードは、レポート・エスケープの前後を修飾します。beforeReportType プロパティは、レポート・エスケープ前の内容がファイルに入っていることを示します。beforeReportValue プロパティは、ファイル名 header_example.html とパスを示します（使用しているマシンで、ファイルが別の場所に存在する場合は、このパスを変更する必要があります）。

afterReportType プロパティは、2 番目のレポート・エスケープの内容が afterReportValue プロパティに存在することを示します。afterReportValue プロパティでは、HTML の前後に <![CDATA[]]> タグが使用されていることに注意してください。XML レポート定義で、XML タグと混同する恐れのある文字を使用する場合は、<![CDATA[]]> タグの内側にそれらのセグメントを入れてください。

- header_example.html ファイルには、グラフィック orep.gif への参照が入っています。このグラフィックは、レポートから生成される HTML と同じパスに存在する必要があります。
- <customize> タグ内のコードの効果を見るには、HTML 出力を生成する必要があります。このレポート出力は、HTMLCSS 出力 (DESFORMAT=HTMLCSS) およびページ・ストリーミング (PAGESTREAM=YES) を使用すると最適に表示されます。

```
<report name="cond" DTDVersion="1.0">
<layout>
  <section name="main">
    <field name="f_trade_date"
      source="trade_date"
      formatMask="MM/DD/RR"/>
    <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>
<customize>
  <object name="videosales" type="REP_REPORT">
    <properties>
      <property name="beforeReportType">File</property>
      <property name="beforeReportValue">
        d:\Yorant\tools\doc60\us\rbbr60\header_example.html
      </property>
      <property name="afterReportType">Text</property>
      <property name="afterReportValue">
        <![CDATA[
          <center>
            <font face="Arial,Helvetica"><font size=-1><font color="#000000">
              Send questions to <a
href="mailto:your_email_id">YourNameHere</a>.
            <br>&nbsp;
          </font>
        </center>
        </body>
```

```

        </html>
    ]>
</property>
</properties>
</object>
</customize>
</report>

```

7.2.2.2 書式例外の例

temp.xml の XML は、temp.rdf の 2 つのフィールドに書式例外を追加します。

この例のヒント

次のヒントは、この例を見る場合に役立ちます。

- 書式変更の定義に <exception> タグが使用されていることに注意してください。この書式例外は、<condition> タグに定義された条件を満たした場合のみ適用されます。
- <customize> セクションの中で <object> タグを使用すると、レイアウトの既存フィールドのラベルを変更できます。フィールドを新規作成する場合は、<field> タグのラベル属性を使用してラベルを指定できます。

```

<report name="temp" DTDVersion="1.0">
  <layout>
    <section name="main">
      <field name="f_p_e" source="p_e" alignment="right"
        formatMask="NNN0.00">
        <exception textColor="red">
          <condition source="p_e" operator="gt" operand1="10"/>
        </exception>
      </field>
      <field name="f_p_e1" source="p_e" alignment="right"
        formatMask="NNN0.00">
        <exception textColor="blue">
          <condition source="p_e" operator="gt" operand1="10"/>
        </exception>
      </field>
    </section>
  </layout>
  <customize>
    <object name="B_high_365" type="REP_GRAPHIC_TEXT">
      <properties>
        <property name="textSegment">High</property>
      </properties>
    </object>
    <object name="B_low_365" type="REP_GRAPHIC_TEXT">
      <properties>

```

```

    <property name="textSegment">Low</property>
  </properties>
</object>
</customize>
</report>

```

7.2.2.3 プログラム単位とハイパーリンクの例

sect.xml の XML は、sect.rdf に 2 つのプログラム単位を追加し、そのプログラム単位を使用してヘッダー・セクションを追加します。

この例のヒント

次のヒントは、この例を見る場合に役立ちます。

- パラメータ・フォームが表示されたとき、パラメータに 100 を入力します。
- プログラム単位は、データ・モデルおよびレイアウトの外側にある <programUnits> タグ内で作成されます。
- 関数は <field> タグの formatTrigger 属性から名前前で参照します。
- PL/SQL 関数の前後に <![CDATA[]]> タグが使用されていることに注意してください。PL/SQL コード中に特殊文字が使用されているので、このタグが必要になります。
- このレポートは、PDF で最適に表示されます。PDF 出力を生成するには、次のコマンドラインを使用します。

```

rwrn60 userid=scott/tiger@nt805 report=sect.rdf customize=sect.xml
destype=file desformat=htmlcss desname=d:¥sect.pdf

```

PDF ファイルをオープンし、マウスを使用して SSN 列の値に移動します。値をクリックすると、そのレコードの詳細が表示されます。

```

<report name="sect" DTDVersion="1.0">
  <layout>
    <section name="header">
      <field name="F_ssn1"
        source="ssn1"
        formatTrigger="F_ssn1FormatTrigger"/>
    </section>
    <section name="main">
      <field name="F_ssn"
        source="ssn"
        formatTrigger="F_ssnFormatTrigger"/>
    </section>
  </layout>
  <programUnits>
    <function name="F_ssn1FormatTrigger">
      <![CDATA[

```

```

function F_ssnFormatTrigger return boolean is
begin
    SRW.SET_HYPERLINK('#EMP_DETAILS_&<' || LTRIM(TO_CHAR(:SSN)) ||
'>');
    return (TRUE);
end;
]]>
</function>
<function name="F_ssnFormatTrigger">
<![CDATA[
function F_ssnFormatTrigger return boolean is
begin
    SRW.SET_LINKTAG('EMP_DETAILS_&<' || LTRIM(TO_CHAR(:SSN)) ||
'>');
    return (TRUE);
end;
]]>
</function>
</programUnits>
</report>

```

7.2.2.4 データ・モデルと書式の変更の例

ref.xml の XML は、ref.rdf のデータ・モデルに新規問合せを追加し、ヘッダー・セクションを追加します。

この例のヒント

次のヒントは、この例を見る場合に役立ちます。

- この XML レポート定義は、単独で実行することも、ref.rdf に適用することもできます。単独で実行できるのは、このレポート定義がデータ・モデルおよび完全なレイアウトの両方を持っているためです。
- SELECT 文の中で別名が使用されていることに注意してください。別名を使用すると、その名前がレポート列に確実に割り当てられるので、通常は SELECT リストで別名を使用すると便利です。別名を使用しなかった場合は、レポート列にデフォルト名が適用され、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> タグのソース属性を指定するときに、正しいソース列名を使用する必要がありますので、名前の付け方は大切なポイントになります。
- また、<labelAttribute> タグの使用方法にも注意してください。このタグは、レイアウトのフィールド・ラベルの書式を定義します。これは <field> タグの外側にあるので、表レイアウトのすべてのラベルに適用されます。フィールドの 1 つのみと関連付ける場合は、このタグを <field></field> タグ対の内側に置きます。ただし、グローバルおよびローカルの <labelAttribute> が両方とも存在する場合は、ローカルなラベル属性がグローバルよりも優先されることに注意してください。詳細は、7.5.8 項「<field>」を参照してください。


```

<report name="ref" DTDVersion="1.0">
  <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="corp2.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>

```

7.2.3 完全なレポート定義

XML レポート定義を使用して、他のレポートに依存せずに単独で実行できる、完全なレポート定義を XML で作成できます。その利点は、Oracle Reports Services Builder を使用せずにレポートを作成できることです。実際に、必要な XML を自分のフロントエンドで作成し、他のユーザーにそれぞれのレポートを動的に作成させることもできます。

次の例に、XML の完全なレポート定義を示します。この XML レポート定義は、videosales.xml という名前で ORACLE_HOME¥TOOLS¥DOC60¥US¥RBBR60 ディレクトリに存在します。

この例のヒント

次のヒントは、この例を見る場合に役立ちます。

- この XML レポート定義は完全であり、単独で実行できます。このレポート定義には完全なデータ・モデルとレイアウトが入っています。このレポートは、PDF で最適に表示されます。
- データ・モデル (Q_1) の最初の問合せを使用して、レポートのヘッダー・セクションにサマリー表レイアウトを配置します。2 番目の問合せ (Q_2) は、レポートのメイン・セクションのマトリックス・ブレイク・レイアウトのために使用します。<group>、<matrixRow>、<matrixCol> および <matrixCell> タグは、このレポート定義のサポートに必要なレイアウトとデータ・モデル構造の両方を定義しています。これらのタグの内側に置かれたフィールドに基づいて、グループおよび列がデータ・モデル内に配置されます。データ・モデルの理解を深めるために、Oracle Reports Services Builder でレポートを開き、データ・モデルを表示します。

```
rwblld60 userid=scott/tiger report=videosales.xml
```

- ヘッダー・セクションの地域値と都市値は、メイン・セクションの地域値と都市値へリンクされています。ハイパーリンクを設定するには、フォーマット・トリガーを地域値と都市が入っているそれぞれのフィールドに関連付けます。トリガー用の PL/SQL は、レポート定義の終わりにある <programUnits> タグの内側に存在します。このレポートを使用して PDF または HTMLCSS 出力を生成した場合、ヘッダー・セクションのサマリー値をクリックすると、レポートのメイン・セクションにある詳細にジャンプします。

```
<report name="videosales" author="Generated" DTDVersion="1.0">
  <data>
    <dataSource name="Q_1">
      <select>
        SELECT ALL VIDEO_CATEGORY_BY_QTR.QUARTER,
              VIDEO_CATEGORY_BY_QTR.SALES_REGION,
              VIDEO_CATEGORY_BY_QTR.STATE, VIDEO_CATEGORY_BY_QTR.CITY,
              VIDEO_CATEGORY_BY_QTR.PRODUCT_CATEGORY,
              VIDEO_CATEGORY_BY_QTR.TOTAL_SALES,
              VIDEO_CATEGORY_BY_QTR.TOTAL_COST,
              VIDEO_CATEGORY_BY_QTR.TOTAL_PROFIT
        FROM SCOTT.VIDEO_CATEGORY_BY_QTR
        WHERE VIDEO_CATEGORY_BY_QTR.SALES_REGION='West'
      </select>
    </dataSource>
    <dataSource name="Q_2">
      <select>
        SELECT ALL VIDEO_CATEGORY_BY_QTR.QUARTER,
              VIDEO_CATEGORY_BY_QTR.CITY,
              VIDEO_CATEGORY_BY_QTR.PRODUCT_CATEGORY,
              VIDEO_CATEGORY_BY_QTR.TOTAL_PROFIT,
              VIDEO_CATEGORY_BY_QTR.TOTAL_SALES,
              VIDEO_CATEGORY_BY_QTR.TOTAL_COST
        FROM SCOTT.VIDEO_CATEGORY_BY_QTR
        WHERE VIDEO_CATEGORY_BY_QTR.SALES_REGION='West'
```

```

</select>
</dataSource>
<summary name="SumTOTAL_SALESPerCITY1" source="total_sales1"/>
<summary name="SumTOTAL_COSTPerCITY1" source="total_cost1"/>
<summary name="SumTOTAL_PROFITPerCITY1" source="total_profit1"/>
<summary name="SumTOTAL_SALESPerQUARTER" source="total_sales"/>
<summary name="SumTOTAL_COSTPerQUARTER" source="total_cost"/>
<summary name="SumTOTAL_PROFITPerQUARTER" source="total_profit"/>
<summary name="SumTOTAL_SALESPerCITY" source="total_sales"/>
<summary name="SumTOTAL_COSTPerCITY" source="total_cost"/>
<summary name="SumTOTAL_PROFITPerCITY" source="total_profit"/>
<formula name="Profit_Margin" source="FormulaProfitMargin"
    datatype="number" width="9"/>
</data>
<layout>
<section name="header">
<groupLeft name="M_video_sales_summary" template="corpl.tdf">
    <group>
        <field name="f_quarter1" source="quarter1" label="Quarter"
            font="Arial" fontSize="8"
            formatTrigger="F_quarter1FormatTrigger">
            <labelAttribute font="Arial" fontSize="8"
                fontStyle="bold" textColor="yellow"/>
        </field>
    </group>
    <group>
        <field name="f_city1" source="city1" label="City"
            font="Arial" fontSize="8"
            formatTrigger="F_city1FormatTrigger">
            <labelAttribute font="Arial" fontSize="8"
                fontStyle="bold" textColor="yellow"/>
        </field>
        <field name="f_SumTOTAL_SALESPerCITY1"
            source="SumTOTAL_SALESPerCITY1"
            label="Sales" font="Arial" fontSize="8"
            formatMask="LNNNGNNNGNNNGNN0D00">
            <labelAttribute font="Arial" fontSize="8"
                fontStyle="bold" textColor="yellow"/>
        </field>
        <field name="f_SumTOTAL_COSTPerCITY1"
            source="SumTOTAL_COSTPerCITY1"
            label="Costs" font="Arial" fontSize="8"
            formatMask="LNNNGNNNGNNNGNN0D00">
            <labelAttribute font="Arial" fontSize="8"
                fontStyle="bold" textColor="yellow"/>
        </field>
        <field name="f_SumTOTAL_PROFITPerCITY1"

```

```
        source="SumTOTAL_PROFITPerCITY1"
        label="Profits" font="Arial" fontSize="8"
        formatMask="LNNNGNNNGNNNGNND00">
<labelAttribute font="Arial" fontSize="8"
        fontStyle="bold" textColor="yellow"/>
</field>
<field name="f_Profit_Margin" source="Profit_Margin"
        label="Margin%" font="Arial" fontSize="8"
        formatMask="N0%">
<labelAttribute font="Arial" fontSize="8"
        fontStyle="bold" textColor="yellow"/>
</field>
</group>
</groupLeft>
</section>
<section name="main">
<matrix name="M_video_sales" template="corp10.tdf">
<group>
<field name="f_quarter" source="quarter" label="Quarter:"
        font="Arial" fontSize="8"
        formatTrigger="F_quarterFormatTrigger">
<labelAttribute font="Arial" fontSize="8"
        fontStyle="bold" textColor="black"/>
</field>
<field name="f_SumTOTAL_SALESPerQUARTER"
        source="SumTOTAL_SALESPerQUARTER"
        label="Qtrly: Sales: " font="Arial" fontSize="8"
        fontStyle="bold"
        formatMask="LNNNGNNNGNNNGNND00">
<labelAttribute font="Arial" fontSize="8"
        fontStyle="bold" textColor="black"/>
</field>
<field name="f_SumTOTAL_COSTPerQUARTER"
        source="SumTOTAL_COSTPerQUARTER"
        label="Costs: " font="Arial" fontSize="8" fontStyle="bold"
        formatMask="LNNNGNNNGNNNGNND00">
<labelAttribute font="Arial" fontSize="8"
        fontStyle="bold" textColor="black"/>
</field>
<field name="f_SumTOTAL_PROFITPerQUARTER"
        source="SumTOTAL_PROFITPerQUARTER"
        label="Profits: " font="Arial" fontSize="8"
        fontStyle="bold"
        formatMask="LNNNGNNNGNNNGNND00">
<labelAttribute font="Arial" fontSize="8"
        fontStyle="bold" textColor="black"/>
</field>
</group>
```

```

</group>
<group>
  <field name="f_state" source="state" label="State:"
    font="Arial" fontSize="8">
    <labelAttribute font="Arial" fontSize="8"
      fontStyle="bold" textColor="black"/>
  </field>
</group>
<matrixCol name="g_city">
  <field name="f_city" source="city" label="City: "
    font="Arial" fontSize="8" textColor="yellow"
    formatTrigger="F_cityFormatTrigger"/>
  <field name="f_SumTOTAL_SALESPerCITY"
    source="SumTOTAL_SALESPerCITY"
    label="Sales: " font="Arial" fontSize="8" fontStyle="bold"
    textColor="yellow" formatMask="LNNNGNNGNNGNNGNOD00">
    <labelAttribute font="Arial" fontSize="8"
      fontStyle="bold" textColor="yellow"/>
  </field>
  <field name="f_SumTOTAL_COSTPerCITY"
    source="SumTOTAL_COSTPerCITY"
    label="Sales: " font="Arial" fontSize="8" fontStyle="bold"
    textColor="yellow" formatMask="LNNNGNNGNNGNNGNOD00">
    <labelAttribute font="Arial" fontSize="8"
      fontStyle="bold" textColor="yellow"/>
  </field>
  <field name="f_SumTOTAL_PROFITPerCITY"
    source="SumTOTAL_PROFITPerCITY"
    label="Sales: " font="Arial" fontSize="8" fontStyle="bold"
    textColor="yellow" formatMask="LNNNGNNGNNGNNGNOD00">
    <labelAttribute font="Arial" fontSize="8"
      fontStyle="bold" textColor="yellow"/>
  </field>
</matrixCol>
<matrixRow name="g_product_category">
  <field name="f_product_category" source="product_category"
    label="Product Category" font="Arial" fontSize="8"/>
</matrixRow>
<matrixCell name="g_total_sales">
  <field name="f_total_sales" source="total_sales" label="Total
Sales"
    font="Arial" fontSize="8" lineColor="noLine"
    formatMask="LNNNGNNGNNGNNGNOD00"/>
  <field name="f_total_cost" source="total_cost" label="Total Cost"
    font="Arial" fontSize="8" lineColor="noLine"
    formatMask="LNNNGNNGNNGNNGNOD00"/>
  <field name="f_total_profit" source="total_profit" label="Total

```

```

Profit"
    font="Arial" fontSize="8" lineColor="noLine"
    formatMask="LNNNGNNNGNNNGNND00"/>
</matrixCell>
</matrix>
</section>
</layout>
<programUnits>
  <function name="F_quarter1FormatTrigger">
    <![CDATA[
      function F_quarter1FormatTrigger return boolean is
      begin
        SRW.SET_HYPERLINK('#QUARTER_DETAILS_&' || LTRIM(:quarter1) ||
'>');
        return (TRUE);
      end;
    ]]>
  </function>
  <function name="F_quarterFormatTrigger">
    <![CDATA[
      function F_quarterFormatTrigger return boolean is
      begin
        SRW.SET_LINKTAG('QUARTER_DETAILS_&' || LTRIM(:quarter) ||
'>');
        return (TRUE);
      end;
    ]]>
  </function>
  <function name="F_city1FormatTrigger">
    <![CDATA[
      function F_city1FormatTrigger return boolean is
      begin
        SRW.SET_HYPERLINK('#QTR_CITY_DETAILS_&' || LTRIM(:quarter1)
||
                                LTRIM(:city1) || '>');
        return (TRUE);
      end;
    ]]>
  </function>
  <function name="F_cityFormatTrigger">
    <![CDATA[
      function F_cityFormatTrigger return boolean is
      begin
        SRW.SET_LINKTAG('QTR_CITY_DETAILS_&' || LTRIM(:quarter) ||
                                LTRIM(:city) || '>');
        return (TRUE);
      end;
    ]]>
  </function>

```

```

]]>
</function>
<function name="FormulaProfitMargin">
  <![CDATA[
    FUNCTION FormulaProfitMargin RETURN number IS
    BEGIN
      return ((:TOTAL_PROFIT1 / (:TOTAL_SALES1 - (0.07 * :TOTAL_SALES1)))
* 100);
    END;
  ]]>
</function>
</programUnits>
</report>

```

7.3 XML レポート定義の実行

作成した XML レポート定義は、次の方法で使用できます。

- [7.3.1 項「実行時の XML レポート定義の適用」](#)
CUSTOMIZE コマンドライン引数または SRW.APPLY_DEFINITION ビルトインを指定して、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。
- [7.3.2 項「XML レポート定義の単独での実行」](#)
REPORT コマンドライン引数を指定して、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。
- [7.3.3 項「バッチ変更の実行」](#)
CUSTOMIZE コマンドライン引数を使用して、RWCON60 によるバッチ変更を実行します。

次に続く項では、前述の例を 1 つずつ詳しく説明し、例を示します。

7.3.1 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 XML ファイルに適用するには、CUSTOMIZE コマンドライン引数または SRW.APPLY_DEFINITION ビルトインを使用します。CUSTOMIZE は、RWCLI60、RWRUN60、RWBLD60、RWCON60 および URL レポート・リクエストと一緒に使用できます。CUSTOMIZE を RWCON60 と使用方法の詳細は、[7.3.3 項「バッチ変更の実行」](#)を参照してください。

7.3.1.1 1 つの XML レポート定義の適用

XML レポート定義 emp.xml を RDF ファイル emp.rdf に適用するジョブ・リクエストを Oracle Reports Services に送信するためのコマンドラインは、次のとおりです。

```
rwcli60 report=emp.rdf customize=e:¥myreports¥emp.xml
      userid=username/password@mydb destype=file desname=emp.pdf desformat=PDF
      server=repserver
```

Oracle Reports Services Runtime を使用する場合は、次のとおりです。

```
rwrun60 userid=username/password@mydb report=emp.rdf
      customize=e:¥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引数を追加してレポート・リクエストを実行すると便利な場合があります。次に例を示します。

```
tracefile=emp.log tracemode=trace_replace traceopt=trace_app
```

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。

7.3.1.2 複数の XML レポート定義の適用

実行時に複数の XML レポート定義を適用するには、CUSTOMIZE コマンドライン引数にリストを指定します。2 つの XML レポート定義 emp0.xml および emp1.xml を RDF ファイル emp.rdf に適用するジョブ・リクエストを Oracle Reports Services に送信するためのコマンドラインは、次のとおりです。

```
rwcli60 report=emp.rdf
      customize="(e:¥corp¥myreports¥emp0.xml,
      e:¥corp¥myreports¥emp1.xml) "
      userid=username/password@mydb destype=file desname=emp.pdf desformat=PDF
      server=repserver
```

Oracle Reports Services Runtime を使用する場合は、次のとおりです。

```
rwrun60 report=emp.rdf
      customize="(e:¥corp¥myreports¥emp0.xml,
      e:¥corp¥myreports¥emp1.xml) "
      userid=username/password@mydb destype=file desname=emp.pdf desformat=PDF
```

7.3.1.3 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。

7.3.1.3.1 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 ビルトインを使用します。

```
SRW.APPLY_DEFINITION ('d:\orant\tools\doc60\us\rbbr60\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。

7.3.1.3.2 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 を使用して適用する必要があります。

次の例に、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、それを適用する方法を示します。この例の PL/SQL は、実際に videosales_custom.rdf というレポート・サンプルの AfterParameterForm トリガーで使用されます。このレポートは ORACLE_HOME\TOOLS\DOC60\US\RBBR60 ディレクトリにあります。

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。

この例のヒント

次のヒントは、この例を見る場合に役立ちます。

- SRW.APPLY_DEFINITION 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 列のサイズに上限があるので、たいへん大きい XML レポート定義の場合は、複数の列にまたがって指定することが必要な場合もあります。その場合は、1 つの大きな定義を作成して一度にまとめて適用するよりも、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="1.0">');
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="LNNNNNNNNNNNOD00"/>');
  SRW.ADD_DEFINITION(' <field name="F_TOTAL_SALES"
    source="TOTAL_SALES" formatMask="LNNNNNNNNNNNOD00"/>');
  SRW.ADD_DEFINITION(' <field name="F_TOTAL_COST"
    source="TOTAL_COST" formatMask="LNNNNNNNNNNNOD00"/>');
  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="1.0">');
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="1.0">');
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="1.0">');
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;
```

7.3.2 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 引数に XML ファイルを指定してリクエストを送信します。次のコマンドラインは、レポート emp.xml を単独で実行するジョブ・リクエストを Oracle Reports Services に送信します。

```
rwcli60 userid=username/password@mydb  
report=e:¥corp¥myreports¥emp.xml  
destype=file desname=emp.pdf desformat=PDF  
server=repserver
```

Oracle Reports Services Runtime を使用する場合は、次のとおりです。

```
rwrun60 userid=username/password@mydb  
report=e:¥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、ファイル拡張子は必ず XML です。また、CUSTOMIZE 引数を使用して、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。

7.3.3 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、RWCON60 で CUSTOMIZE コマンドライン引数を使用して、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ときに便利です（たとえば、フィールドの書式マスクの変更など）。Oracle Reports Services 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、RWCON60 を 1 回実行するのみで、素早く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 inven.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。

```
rwcon60 username/password@mydb  
stype=rdf  
source="(inven1.rdf, inven2.rdf, manu.rdf)"  
dtype=rdf  
dest="(inven1_new.rdf, inven2_new.rdf, manu_new.rdf)"  
customize="(e:¥apps¥trans¥translate.xml,  
e:¥apps¥custom¥customize.xml)" batch=yes
```

7.4 XML レポート定義のデバッグ

XML レポート定義のデバッグに役立つ機能は、次のとおりです。

- [7.4.1 項「XML パーサーのエラー・メッセージ」](#)
- [7.4.2 項「トレース・オプション」](#)
- [7.4.3 項「RWBLD60」](#)
- [7.4.4 項「TEXT_IO」](#)

7.4.1 XML パーサーのエラー・メッセージ

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なります。

7.4.2 トレース・オプション

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引数を追加してレポート・リクエストを実行すると便利な場合があります。次に例を示します。

```
rwrun60 username/password@mydb
report=e:¥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 つの引数は、レポートのフェッチおよび書式の詳細リストを備えたトレース・ファイルを生成します。次の例は、正常に実行されたレポートのトレース・ファイルの一部です。

```
LOG :
    Report: d:¥xml_reps¥test1.xml
    Logged onto server:
    Username:
LOG :
    Logged onto server: nt805
    Username: scott
+-----+
| Report customization/generation begins |
+-----+
Processing XML report definition 1 of 1.
*** Parsing the XML document ***
Creating XML parser object...
XML Parser Created!
Parsing report definition from:
```

```

d:\xml_reps\test1.xml
Report definition parsed successfully!
*** Setting Application Property ***
Setting module name to "test"...
Done with application level properties modification.
*** Creating PL/SQL Program Units ***
*** Defaulting the Data Model ***
Created query Q_depemp.
Applying SQL to query Q_depemp and creating columns...
Done with queries and columns creation/modification.
Done with groups creation/modification.
*** Defaulting the Layout ***
Start defaulting layout for main section...
Defaulting field f_deptno for column deptno...
Defaulting field f_mgr for column mgr...
Defaulting field f_job for column job...
Layout defaulted into new frame M_empform.
*** Modifying report objects' properties ***
+-----+
| Report customization/generation finished successfully |
+-----+
11:22:59 APP ( Frame
11:22:59 APP . ( Text Boilerplate B_DATE1_SEC2
11:22:59 APP . ) Text Boilerplate B_DATE1_SEC2
11:22:59 APP . ( Text Boilerplate B_PAGENUM1_SEC2
11:22:59 APP . ) Text Boilerplate B_PAGENUM1_SEC2
11:22:59 APP . ( Text Field F_DATE1_SEC2
11:22:59 APP .. ( Database Column Name unknown
11:22:59 APP .. ) Database Column Name unknown
11:22:59 APP . ) Text Field F_DATE1_SEC2
11:22:59 APP ) Frame
11:22:59 APP ( Frame
11:22:59 APP . ( Frame M_G_1_GRPFR
11:22:59 APP .. ( Frame M_G_1_HDR
11:22:59 APP ... ( Text Boilerplate B_DEPTNO
11:22:59 APP ... ) Text Boilerplate B_DEPTNO
11:22:59 APP ... ( Text Boilerplate B_MGR
11:22:59 APP ... ) Text Boilerplate B_MGR
11:22:59 APP ... ( Text Boilerplate B_JOB
11:22:59 APP ... ) Text Boilerplate B_JOB
11:22:59 APP .. ) Frame M_G_1_HDR
11:22:59 APP .. ( Repeating Frame R_G_1
11:22:59 APP ... ( Group G_1 Local Break: 0 Global
Break: 0
11:22:59 APP .... ( Query Q_depemp
11:22:59 SQL EXECUTE QUERY : select * from emp
11:22:59 APP .... ) Query Q_depemp

```

```

11:22:59 APP ... ) Group G_1
11:22:59 APP ... ( Text Field F_DEPTNO
11:22:59 APP .... ( Database Column DEPTNO
11:22:59 APP .... ) Database Column DEPTNO
.
.
.
+-----+
| Report Builder Profiler statistics |
+-----+
TOTAL ELAPSED Time: 11.00 seconds
Reports Time: 10.00 seconds (90.90% of TOTAL)
ORACLE Time: 1.00 seconds ( 9.09% of TOTAL)
UPI: 0.00 second
SQL: 1.00 seconds
TOTAL CPU Time used by process: N/A

```

7.4.3 RWBLD60

XML レポート定義の設計時には、Oracle Reports Services Builder でオープンした方が便利
な場合があります。Oracle Reports Services 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判断できます。たとえば、XML レポート定義にサマリー
を作成する場合、Oracle Reports Services 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。

Oracle Reports Services Builder で完全なレポート定義をオープンするには、REPORT キー
ワードを使用します。次に例を示します。

```

rwblld60 userid=username/password@mydb
report=e:¥corp¥myreports¥emp.xml

```

Oracle Reports Services 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。

```

rwblld60 userid=username/password@mydb report=emp.rdf
customize=e:¥myreports¥emp.xml

```

どちらの場合も、Oracle Reports Services Builder で有効な XML レポート定義がオープンし
ます。その後、Oracle Reports Services Editor の様々なビュー（ライブ・プレビューなど）
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すぐに判断でき
ます。

7.4.4 TEXT_IO

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。次のサンプル・プロシージャは、メモリーに転送するすべての行をドキュメント・バッファおよび（ファイルへの書き込みを選択した場合は）指定したファイルに書き込みます。

```
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、プロシージャを呼び出す PL/SQL で TEXT_IO.File_Type 型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 プロシージャを呼び出して、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" DIDVersion="1.0">',
        custom_summary);
```


7.5 XML タグ・リファレンス

Oracle Reports Services に組み込まれている文書タイプ定義 (DTD) ファイルには、XML レポート定義で使用可能なタグが定義されています。次に続く項では、それぞれのタグとその構文を説明し、使用例を示します。タグは階層順 (外側から内側の順) に記載されています。

警告： XML タグおよびその属性は大文字と小文字を区別するので、構文説明に示されたとおりに入力する必要があります。

7.5.1 <!-- コメント -->

説明

<!-- --> タグを使用すると、XML レポート定義の中にコメントを挿入できます。パーサーは、コメント・デリミタで囲まれた文字をすべて無視します。PL/SQL (SRW.ADD_DEFINITION) を使用して XML レポート定義を作成する場合は、PL/SQL のコメント・デリミタ (たとえば、-- または /* */) を使用してプログラム単位にコメントを組み込みます。

構文

このタグの構文は、次のとおりです。

```
<!--
  comment_content
-->
```

例

次の XML レポート定義セグメント例では、<!-- --> タグを使用してコメントを挿入しています。

```
<report name="cond" DTDVersion="1.0">
<!-- This report assumes that the file
      named header_example.html is located
      in d:\ORANT\TOOLS\DOC60\US\RBRR60.
      If it is not located there, the report
      will not run properly.
-->
```

7.5.2 <![CDATA[]]>

説明

<![CDATA[]]> タグを使用すると、XML レポート定義に特殊文字を挿入できます。パーサーは、<![CDATA[]]> タグの内側で検出された特殊文字をすべて無視します。これは、特殊文字を必要とする PL/SQL プログラム単位または SQL 問合せを組み込むときに、たいへん役立ちます。

構文

このタグの構文は、次のとおりです。

```
<![CDATA[
    content
]]>
```

例

次の XML レポート定義セグメント例では、<![CDATA[]]> タグを使用して、ハイパーリンクおよびハイパーリンクのリンク先をレポート内のオブジェクトに追加する PL/SQL 関数を保護しています。

```
<programUnits>
<function name="F_ssnFormatTrigger">
  <![CDATA[
    function F_ssnFormatTrigger return boolean is
    begin
      SRW.SET_HYPERlink('#EMP_DETAILS_&' || LTRIM(TO_CHAR(:SSN)) || '>');
      return (TRUE);
    end;
  ]]>
</function>
<function name="F_ssnFormatTrigger">
  <![CDATA[
    function F_ssnFormatTrigger return boolean is
    begin
      SRW.SET_linkTAG('EMP_DETAILS_&' || LTRIM(TO_CHAR(:SSN)) || '>');
      return (TRUE);
    end;
  ]]>
</function>
</programUnits>
```

次の XML レポート定義セグメント例では、<![CDATA[]]> タグを使用して、> 記号を含んだ SQL 文を保護しています。

```
<select>
  <![CDATA[
    SELECT ALL VIDEO_CATEGORY_BY_QTR.QUARTER,
           VIDEO_CATEGORY_BY_QTR.TOTAL_PROFIT
    FROM SCOTT.VIDEO_CATEGORY_BY_QTR
    WHERE (VIDEO_CATEGORY_BY_QTR.SALES_REGION='West'
           AND VIDEO_CATEGORY_BY_QTR.TOTAL_PROFIT>2000)
  ]]>
</select>
```

7.5.3 <condition>

説明

<condition> タグは、書式例外をフィールドに適用する条件を定義します。<condition> タグは、<exception> タグの内部でネストする必要があります。

詳細は、7.5.7 項「<exception>」を参照してください。

構文

このタグの構文は、次のとおりです。

```
<condition
  source="source_column_name"
  operator="eq | lt | lteq | neq | gt | gteq | btw | notBtw | like | notLike
          | null | notNull"
  [operand1="comparison_value"]
  [operand2="comparison_value"]
  [relation="and | or"]
/>
```

属性

次の表に、<condition> タグの属性を示します。

表 7-2 <condition> タグの属性

属性	必須 / 任意	説明
source	必須	条件で使用するソース列の名前。
operator	必須	他の値をソース列と比較するときに使用する演算子。有効な演算子は、次のとおりです。 <ul style="list-style-type: none">▪ eq (equal)▪ lt (less than)▪ lteq (less than or equal)▪ neq (not equal)▪ gt (greater than)▪ gteq (greater than or equal)▪ btw (between)▪ notBtw (not between)▪ like▪ notLike▪ null▪ notNull
operand1	任意	ソース列の比較対象となる値。演算子が null または notNull であれば、オペランドは不要です。演算子が btw または notBtw の場合は、operand2 も指定する必要があります。
operand2	任意	ソース列の比較対象となる 2 番目の値。演算子が比較用に 2 つの値を必要とする場合のみ (btw または notBtw の場合) operand2 を使用する必要があります。
relation	任意	複数の条件の有無を定義し、条件が複数ある場合、その関係を定義します。 <ul style="list-style-type: none">▪ and は、両方の条件を満たした場合のみ、書式例外を適用します。▪ or は、どちらか一方の条件を満たした場合に、書式例外を適用します。

使用上の注意

2 つの条件を結合するには、最初の condition タグに relation 属性を入力し、演算子 and または or のどちらかを指定する必要があります。

例

次の例では、フィールド f_ename に適用する 2 通りの書式例外を示しています。最初の exception は、条件を満たした場合にテキスト・カラーを赤に変更します。2 番目の exception は、条件を満たした場合にテキスト・カラーを青に変更します。

```
<field name="f_ename" source="ename" label="Employee" textColor="green">
  <exception textColor="red">
    <condition source="deptno" operator="btw" operand1="20"
      operand2="30" relation="and"/>
    <condition source="sal" operator="gt" operand1="1000"/>
  </exception>
  <exception textColor="blue">
    <condition source="deptno" operator="eq" operand1="30"/>
  </exception>
</field>
```

7.5.4 <customize>

説明

<customize> タグは、レポート定義の一部として指定する、あらゆるオブジェクト・プロパティの範囲を定めます。<customize> タグの内部でタグをネストすると (<object> <properties> および <property>)、レポートで指定するいくつかのオブジェクトのプロパティを設定できます。

構文

このタグの構文は、次のとおりです。

```
<customize>
  content_of_data_model
</customize>
```

例

次の例に、XML レポート定義のオブジェクト・プロパティのセグメントを示します。

```
<customize>
  <object name="videosales" type="REP_REPORT">
    <properties>
      <property name="beforeReportType">File</property>
      <property name="beforeReportValue">
        d:¥xml_reps¥header_example.html
      </property>
      <property name="afterReportType">Text</property>
      <property name="afterReportValue">
        <![CDATA[
          <center>
            <font face="Arial,Helvetica"><font size=-1><font color="#000000">
              Send questions to <a href="mailto:your_email_id">YourNameHere</a>.
            <br>&nbsp;
            </font>
          </center>
        </body>
        </html>
        ]]>
      </property>
    </properties>
  </object>
</customize>
```

次の例に示す XML レポート定義セグメントは、ボイラープレート・テキストの一部を変更します。これは、既存フィールドのラベルを変更する場合に役立ちます。

```
<customize>
  <object name="B_high_365" type="REP_GRAPHIC_TEXT">
    <properties>
      <property name="textSegment">High</property>
    </properties>
  </object>
  <object name="B_low_365" type="REP_GRAPHIC_TEXT">
    <properties>
      <property name="textSegment">Low</property>
    </properties>
  </object>
</customize>
```

7.5.5 <data>

説明

<data> タグは、レポート定義のデータ・モデルの始めと終わりを示します。

構文

このタグの構文は、次のとおりです。

```
<data>
  content_of_data_model
</data>
```

例

次の例に、XML レポート定義のデータ・モデルのセグメントを示します。

```
<data>
  <dataSource name="q_category">
    <select>
      SELECT          ic.category,
                      SUM (h.sales),
                      AVG (h.high_365),
                      AVG (h.low_365),
                      AVG (h.div),
                      AVG (h.p_e)
      FROM stock_history h, indcat ic
      WHERE h.symbol=ic.symbol
      GROUP BY ic.category
    </select>
  </dataSource>
</data>
```

次の XML レポート定義セグメント例では、<![CDATA[]]> タグを使用して、> 記号を含んだ SQL 文を保護しています。

```
<data>
  <dataSource name="Q_1">
    <select>
      <![CDATA[
        SELECT ALL VIDEO_CATEGORY_BY_QTR.QUARTER,
                VIDEO_CATEGORY_BY_QTR.TOTAL_PROFIT
        FROM SCOTT.VIDEO_CATEGORY_BY_QTR
        WHERE (VIDEO_CATEGORY_BY_QTR.SALES_REGION='West'
              AND VIDEO_CATEGORY_BY_QTR.TOTAL_PROFIT>2000)
      ]]>
    </select>
```

```
</dataSource>
</data>
```

7.5.6 <dataSource>

説明

<dataSource> タグは、データ・モデルの問合せの始めと終わりを示します。<dataSource> タグは、<data> タグの内部でネストする必要があります。このタグは、Oracle Reports Services (SQL および Express) でサポートされたすべてのデータ・ソースをサポートします。

構文

このタグの構文は、次のとおりです。

```
<dataSource>
  content_of_data_source
</dataSource>
```

例

次の例に、XML レポート定義のデータ・モデルのセグメントを示します。

```
<data>
  <dataSource name="q_category">
    <select>
      SELECT          ic.category,
                      SUM (h.sales),
                      AVG (h.high_365),
                      AVG (h.low_365),
                      AVG (h.div),
                      AVG (h.p_e)
      FROM stock_history h, indcat ic
      WHERE h.symbol=ic.symbol
      GROUP BY ic.category
    </select>
  </dataSource>
</data>
```

次の XML レポート定義セグメント例では、<![CDATA[]]> タグを使用して、> 記号を含んだ SQL 文を保護しています。

```
<data>
  <dataSource name="Q_1">
    <select>
      <![CDATA[
        SELECT ALL VIDEO_CATEGORY_BY_QTR.QUARTER,
```



```

        VIDEO_CATEGORY_BY_QTR.TOTAL_PROFIT
    FROM SCOTT.VIDEO_CATEGORY_BY_QTR
    WHERE (VIDEO_CATEGORY_BY_QTR.SALES_REGION='West'
        AND VIDEO_CATEGORY_BY_QTR.TOTAL_PROFIT>2000)
    ]]>
</select>
</dataSource>
</data>

```

7.5.7 <exception>

説明

<exception> タグは、フィールドに適用する書式例外の範囲を定めます（たとえば、フィールドの値が一定の範囲を超えたときに、赤に変更するなど）。<exception> タグは、<field> タグの内部でネストする必要があります。また、書式例外を適用する条件を定義する <condition> タグを、その内部でネストさせる必要があります。

詳細は、次の項を参照してください。

- [7.5.8 項「<field>」](#)
- [7.5.3 項「<condition>」](#)

構文

このタグの構文は、次のとおりです。

```

<exception
    [lineColor="color_name | noLine"]
    [fillColor="color_name | noFill"]
    [textColor="color_name"]
    [hide="yes | no"]
    [font="font_name"]
>
    condition_definition
</exception>

```

属性

次の表に、<exception> タグの属性を示します。

表 7-3 <exception> タグの属性

属性	必須 / 任意	説明
lineColor	任意	条件を満たした場合に適用する境界線カラーの名前。noLine を指定した場合、境界線は透明です（つまり、表示されません）。
fillColor	任意	条件を満たした場合に適用する塗りカラーの名前。noFill を指定した場合、背景は透明です。
textColor	任意	条件を満たした場合に適用するテキスト・カラーの名前。
hide	任意	条件を満たした場合に、フィールドを非表示にするかどうかを指定します。 <ul style="list-style-type: none">■ yes は、条件を満たした場合にフィールドを非表示にします。■ no は、条件を満たした場合にフィールドを非表示にしません。
font	任意	条件を満たした場合に適用するフォントの名前。
fontSize	任意	条件を満たした場合に適用するフォントのサイズ。
fontStyle	任意	条件を満たした場合に適用するフォントのスタイル。有効なスタイルは、次のとおりです。 <ul style="list-style-type: none">■ regular（標準）■ italic（斜体）■ bold（太字）■ boldItalic（太字斜体）
fontEffect	任意	条件を満たした場合に適用するフォントの効果。有効値は、次のとおりです。 <ul style="list-style-type: none">■ regular（標準）■ strikeout（取消し線）■ underline（下線）■ strikeoutUnderline（取消し線と下線）

使用上の注意

使用上の注意は、次のとおりです。

- 例外は、フィールド内で出現した順に処理されます。
- それぞれの例外に最大 3 つの条件を指定できます。
- フィールドに適用できる条件数に上限はありません。ただし、その結果発生するフォーマット・トリガーについての PL/SQL の最大長は制限されます。
- 複数の例外がある場合は、暗黙的 OR 関係によって制御されます。したがって、どれか 1 つの例外が適用されると（つまり、満たされた場合）他の例外はいっさい処理されません。

例

次の例では、フィールド `f_ename` に適用する 2 通りの書式例外を示しています。最初の exception は、条件を満たした場合にテキスト・カラーを赤に変更します。2 番目の exception は、条件を満たした場合にテキスト・カラーを青に変更します。

```
<field name="f_ename" source="ename" label="Employee" textColor="green">
  <exception textColor="red">
    <condition source="deptno" operator="btw" operand1="1"
      operand2="20" relation="and"/>
    <condition source="sal" operator="gt" operand1="1000"/>
  </exception>
  <exception textColor="blue">
    <condition source="deptno" operator="eq" operand1="30"/>
  </exception>
</field>
```

7.5.8 <field>

説明

<field> タグは、レポート定義のレイアウトのフィールドを定義し、属性を割り当てます。
 <field> タグは、<layout> タグの内部でネストする必要があります。他のほとんどの layout タグは、その内部で <field> をネストさせる必要があります（たとえば、<tabular>、<group>、<matrixCell> など）。RDF ファイル内のフィールドと同じフィールド名を使用すると、<field> タグによって RDF ファイルの既存フィールドが変更されます。それ以外の場合は、このタグを使用してレポートでまったく新しいフィールドを作成できます。

<field> タグ内に、<labelAttribute> および <exception> タグを入れることもできます。

<field> タグは、/> または </field> で終了できます。<field> タグの内側に <exception> または <labelAttribute> タグを入れる場合は、後者の方法（</field>）で終了する必要があります。次の例に、<field> タグの 2 通りの終了方法を示します。

```
<field name="f_deptno" label="Department" source="deptno"/>
<field name="f_mgr" label="Manager" source="mgr">
  <labelAttribute textColor="red" alignment="center"/>
</field>
```

詳細は、次の項を参照してください。

- [7.5.7 項「<exception>」](#)
- [7.5.15 項「<labelAttribute>」](#)

構文

```
<field
  name="field_name"
  source="source_column"
  [label="field_label"]
  [currency="currency_symbol"]
  [tsep="separator_character"]
  [formatTrigger="plsql_program_unit"]
  [font="font_name"]
  [fontSize="point_size"]
  [fontStyle="regular | italic | bold | boldItalic"]
  [fontEffect="regular | strikethrough | underline | strikethroughUnderline"]
  [lineColor="color_name | noLine"]
  [fillColor="color_name | noFill"]
  [textColor="color_name"]
  [alignment="start | left | center | right | end"]
  [hyperlink="URL"]
  [linkdest="hyperlink_target"]
  [formatMask="mask"]
/> | >[other_tags]</field>
```

属性

次の表に、<field> タグの属性を示します。

表 7-4 <field> タグの属性

属性	必須 / 任意	説明
name	必須	フィールドの識別子。XML の適用先 RDF ファイル内のフィールド名が name と一致した場合は、RDF ファイルの属性よりも指定した属性の方が優先します。

表 7-4 <field> タグの属性

属性	必須 / 任意	説明
source	フィールドを新規作成する場合は必須。 既存フィールドを変更する場合は任意。	フィールドのデータの取得元となるソース列。ソース列は、データ・モデルに必ず存在する必要があります。
label	任意	<p>フィールドに対応付けるボイラプレート・テキスト。ラベルの書式属性を制御するには、<labelAttribute> タグを使用する必要があります。詳細は、7.5.15 項「<labelAttribute>」を参照してください。</p> <p>label 属性は新規フィールドのみに作用し、RDF ファイルの既存フィールドのラベルは変更されません。既存フィールドのラベルを変更するには、<object> タグを使用します。詳細は、7.5.22 項「<object>」を参照してください。</p>
currency	任意	フィールドで使用する通貨記号(¥など)。この属性の他に、通貨記号の位置を示すために formatMask 属性も指定する必要があります。
tsep	任意	デリミタ付き出力を作成するときに使用するデリミタ。最も一般的なデリミタはタブであり、Microsoft Excel などの表計算プログラムで読み取れます。
formatTrigger	任意	フィールドのフォーマット・トリガーとして使用する PL/SQL プログラム単位の名前。フォーマット・トリガーは関数である必要があります。詳細は、Oracle Reports Services Builder オンライン・ヘルプ・システムを参照し、索引でフォーマット・トリガーを検索してください。
font	任意	フィールドの内容に使用するフォントの名前。
fontSize	任意	フィールドの内容に使用するフォントのサイズ。
fontStyle	任意	<p>フィールドの内容に使用するフォントのスタイル。有効値は、次のとおりです。</p> <ul style="list-style-type: none"> ■ regular (標準) ■ italic (斜体) ■ bold (太字) ■ boldItalic (太字斜体)

表 7-4 <field> タグの属性

属性	必須 / 任意	説明
fontEffect	任意	フィールドの内容に使用するフォントの効果。有効値は、次のとおりです。 <ul style="list-style-type: none"> ■ regular (標準) ■ strikeout (取消し線) ■ underline (下線) ■ strikeoutUnderline (取消し線と下線)
lineColor	任意	フィールドの境界線に使用するカラーの名前。noLine を指定した場合、フィールドの境界線は透明です (つまり、表示されません)。
fillColor	任意	フィールドの背景として使用するカラーの名前。noFill を指定した場合、背景は透明です。
textColor	任意	フィールドの内容に使用するカラーの名前。
alignment	任意	フィールド内の文字位置を揃える方法。有効値は、次のとおりです。 <ul style="list-style-type: none"> ■ start (始点詰め) ■ left (左揃え) ■ center (中央揃え) ■ right (右揃え) ■ end (終点詰め)
hyperlink	任意	HTML または PDF 出力生成時、フィールドの内容に対応付ける URL。この属性は、PostScript または ASCII などの他の出力タイプでは無視されます。
linkdest	任意	このフィールドの内容にハイパーリンクを設定するときに使用するターゲット。この属性は、HTML または PDF 出力の生成時にのみ使用されます。PostScript または ASCII などの他の出力タイプでは、この属性は無視されます。
formatMask	任意	フィールドの内容を表示するときに適用するマスク。書式マスクの構文の詳細は、Oracle Reports Services Builder オンライン・ヘルプ・システムを参照し、索引で書式マスクの項を検索してください。

例

次の例は、マトリックス・レポートの2つのブレイク・グループ内のフィールドを定義しているレポート定義のレイアウトの一部です。

```
<group>
  <field name="f_quarter" source="quarter" label="Quarter:"
    font="Arial" fontSize="8"
    formatTrigger="F_quarterFormatTrigger">
    <labelAttribute font="Arial" fontSize="8"
      fontStyle="bold" textColor="black"/>
  </field>
  <field name="f_SumTOTAL_SALESPerQUARTER"
    source="SumTOTAL_SALESPerQUARTER"
    label="Qtrly: Sales: " font="Arial" fontSize="8" fontStyle="bold"
    formatMask="LNNNGNNNGNNNGNND00">
    <labelAttribute font="Arial" fontSize="8"
      fontStyle="bold" textColor="black"/>
  </field>
  <field name="f_SumTOTAL_COSTPerQUARTER" source="SumTOTAL_COSTPerQUARTER"
    label="Costs: " font="Arial" fontSize="8" fontStyle="bold"
    formatMask="LNNNGNNNGNNNGNND00">
    <labelAttribute font="Arial" fontSize="8"
      fontStyle="bold" textColor="black"/>
  </field>
  <field name="f_SumTOTAL_PROFITPerQUARTER"
    source="SumTOTAL_PROFITPerQUARTER"
    label="Profits: " font="Arial" fontSize="8" fontStyle="bold"
    formatMask="LNNNGNNNGNNNGNND00">
    <labelAttribute font="Arial" fontSize="8"
      fontStyle="bold" textColor="black"/>
  </field>
</group>
<group>
  <field name="f_state" source="state" label="State:"
    font="Arial" fontSize="8">
    <labelAttribute font="Arial" fontSize="8"
      fontStyle="bold" textColor="black"/>
  </field>
</group>
```

次の例は、グループ左レポートのブレイク・グループ内のフィールドを定義したレポート定義のレイアウトの一部です。formatTrigger 属性は <programUnits> タグ内に定義される関数をポイントしています。

```
<group>
  <field name="f_quarter1" source="quarter1" label="Quarter"
    font="Arial" fontSize="8"
```

```
formatTrigger="F_quarter1FormatTrigger">
  <labelAttribute font="Arial" fontSize="8"
    fontStyle="bold" textColor="yellow"/>
</field>
</group>
```

7.5.9 <formLike>

説明

<formLike> タグは、レポートのレイアウトのセクション内でフォーム・スタイルの範囲を定めます。<formLike> タグを使用する場合は、<field> タグもネストさせてフォーム・レイアウトに入れるフィールドをリストする必要があります。

<field> タグの詳細は、[7.5.8 項「<field>」](#)を参照してください。

構文

このタグの構文は、次のとおりです。

```
<formLike>
  <field>
  </field>
  [...]
</formLike>
```

例

次の例では、フォーム・レイアウトを含んだセクションを定義した XML レポート定義セグメントを示します。

```
<section name="main">
  <formLike name="M_empform" template="corp2.tdf">
    <labelAttribute textColor="green" alignment="center"/>
    <field name="f_deptno" source="deptno" label="Department"/>
    <field name="f_mgr" source="mgr" label="Manager">
      <labelAttribute textColor="red" alignment="center"/>
    </field>
    <field name="f_job" label="Job" source="job"/>
  </formLike>
</section>
```


7.5.10 <formula>

説明

<formula> タグは、レポート定義のデータ・モデルの計算式列を定義します。計算式列では、PL/SQL 関数を使用して、演算（通常は複雑な計算）を実行します。一般的な計算の場合（たとえば、和、合計パーセントまたは標準偏差など）は、PL/SQL を必要としない <summary> タグを使用します。

詳細は、[7.5.29 項「<summary>」](#)を参照してください。

構文

このタグの構文は、次のとおりです。

```
<formula
  name="column_name"
  source="plsql_function_name"
  dataType="number | character | date"
  width="number"
/>
```

属性

次の表に、<formula> タグの属性を示します。

表 7-5 <formula> タグの属性

属性	必須 / 任意	説明
name	必須	計算式列の名前。
source	必須	<programUnits> タグの内部に定義された PL/SQL 関数の名前で、計算式の演算を実行します。
dataType	任意	式によって生成されるデータの型。たとえば、算術演算を実行した場合、演算結果は数値です。dataType 値として可能なものは、次のとおりです。 <ul style="list-style-type: none">■ number（数値）■ character（文字）■ date（日付）
width	任意	計算式の結果の幅に相当する文字数。

例

次の例では、計算式列が入っているデータ・モデルを定義した XML レポート定義のセグメントを示します。<layout> セクション内での関連フィールドの位置に基づいて、デフォルトのアルゴリズムが列を適切なグループに配置しています。

```
<data>
  <dataSource name="Q_1">
    <select>
      SELECT ALL VIDEO_CATEGORY_BY_QTR.QUARTER,
              VIDEO_CATEGORY_BY_QTR.SALES_REGION,
              VIDEO_CATEGORY_BY_QTR.STATE, VIDEO_CATEGORY_BY_QTR.CITY,
              VIDEO_CATEGORY_BY_QTR.PRODUCT_CATEGORY,
              VIDEO_CATEGORY_BY_QTR.TOTAL_SALES,
              VIDEO_CATEGORY_BY_QTR.TOTAL_COST,VIDEO_CATEGORY_BY_QTR.TOTAL_PROFIT
      FROM SCOTT.VIDEO_CATEGORY_BY_QTR
      WHERE VIDEO_CATEGORY_BY_QTR.SALES_REGION='West '
    </select>
  </dataSource>
  <dataSource name="Q_2">
    <select>
      SELECT ALL VIDEO_CATEGORY_BY_QTR.QUARTER, VIDEO_CATEGORY_BY_QTR.CITY,
              VIDEO_CATEGORY_BY_QTR.PRODUCT_CATEGORY,
              VIDEO_CATEGORY_BY_QTR.TOTAL_PROFIT,
              VIDEO_CATEGORY_BY_QTR.TOTAL_SALES,
              VIDEO_CATEGORY_BY_QTR.TOTAL_COST
      FROM SCOTT.VIDEO_CATEGORY_BY_QTR
      WHERE VIDEO_CATEGORY_BY_QTR.SALES_REGION='West '
    </select>
  </dataSource>
  <formula name="Profit_Margin" source="FormulaProfitMargin"
           datatype="number" width="9"/>
</data>
<programUnits>
  <function name="FormulaProfitMargin">
    <![CDATA[
      FUNCTION FormulaProfitMargin RETURN number IS
      BEGIN
        return ((:TOTAL_PROFIT1 / (:TOTAL_SALES1 - (0.07 * :TOTAL_SALES1))) *
100);
      END;
    ]]>
  </function>
</programUnits>
```

7.5.11 <function>

<function> タグは、レポート定義に追加する PL/SQL 関数を定義します。<function> タグは、<programUnits> タグの内部でネストする必要があります。関数を参照するには、<field> タグの formatTrigger 属性を使用します。

詳細は、次の項を参照してください。

- [7.5.23 項「<programUnits>」](#)
- [7.5.8 項「<field>」](#)

構文

このタグの構文は、次のとおりです。

```
<function
  name="function_name"
>
  PLSQL_function
</function>
```

属性

次の表に、<function> タグの属性を示します。

表 7-6 <function> タグの属性

属性	必須 / 任意	説明
name	必須	関数の識別子。関数を参照する（たとえば、<field> タグの formatTrigger 属性から参照する）ときは、この名前を使用する必要があります。

例

次に、PL/SQL 関数を定義する XML レポート定義セグメントを示します。関数は formatTrigger 属性を使用してレイアウトのフィールドから参照されます。

```
<layout>
  <section name="header">
    <field name="F_ssn1"
      source="ssn1"
      formatTrigger="F_ssn1FormatTrigger"/>
  </section>
  <section name="main">
    <field name="F_ssn"
      source="ssn"
      formatTrigger="F_ssnFormatTrigger"/>
  </section>
```

```
</layout>
<programUnits>
  <function name="F_ssnFormatTrigger">
    <![CDATA[
      function F_ssnFormatTrigger return boolean is
      begin
        SRW.SET_HYPERLINK('#EMP_DETAILS_&<' || LTRIM(TO_CHAR(:SSN)) ||
'>');
        return (TRUE);
      end;
    ]]>
  </function>
  <function name="F_ssnFormatTrigger">
    <![CDATA[
      function F_ssnFormatTrigger return boolean is
      begin
        SRW.SET_LINKTAG('EMP_DETAILS_&<' || LTRIM(TO_CHAR(:SSN)) || '>');
        return (TRUE);
      end;
    ]]>
  </function>
</programUnits>
```

7.5.12 <group>

説明

<group> タグは、マスター / デイテール・スタイル・レイアウトでマスター・グループの範囲を定めます。<group> タグは、<groupLeft>、<groupAbove> または <matrix> タグの内部でのみネストできます。<group> タグ内で <field> タグをネストして、マスター・グループに含めるフィールドをリストする必要があります。

詳細は、次の項を参照してください。

- [7.5.8 項「<field>」](#)
- [7.5.13 項「<groupAbove>」](#)
- [7.5.14 項「<groupLeft>」](#)
- [7.5.18 項「<matrix>」](#)

構文

このタグの構文は、次のとおりです。

```
<group>
  master_group_content
</group>
```

例

次の例は、マトリックス・レポートの2つのブレイク・グループ内のフィールドを定義しているレポート定義のレイアウトの一部です。

```
<group>
  <field name="f_quarter" source="quarter" label="Quarter:"
    font="Arial" fontSize="8"
    formatTrigger="F_quarterFormatTrigger">
    <labelAttribute font="Arial" fontSize="8"
      fontStyle="bold" textColor="black"/>
  </field>
  <field name="f_SumTOTAL_SALESPerQUARTER"
    source="SumTOTAL_SALESPerQUARTER"
    label="Qtrly: Sales: " font="Arial" fontSize="8" fontStyle="bold"
    formatMask="LNNNGNNNGNNNGNND00">
    <labelAttribute font="Arial" fontSize="8"
      fontStyle="bold" textColor="black"/>
  </field>
  <field name="f_SumTOTAL_COSTPerQUARTER" source="SumTOTAL_COSTPerQUARTER"
    label="Costs: " font="Arial" fontSize="8" fontStyle="bold"
    formatMask="LNNNGNNNGNNNGNND00">
    <labelAttribute font="Arial" fontSize="8"
      fontStyle="bold" textColor="black"/>
  </field>
  <field name="f_SumTOTAL_PROFITPerQUARTER"
    source="SumTOTAL_PROFITPerQUARTER"
    label="Profits: " font="Arial" fontSize="8" fontStyle="bold"
    formatMask="LNNNGNNNGNNNGNND00">
    <labelAttribute font="Arial" fontSize="8"
      fontStyle="bold" textColor="black"/>
  </field>
</group>
<group>
  <field name="f_state" source="state" label="State:"
    font="Arial" fontSize="8">
    <labelAttribute font="Arial" fontSize="8"
      fontStyle="bold" textColor="black"/>
  </field>
</group>
```

7.5.13 <groupAbove>

説明

<groupAbove> タグは、レポートのレイアウトのセクション内でマスター / ディテール・スタイルの範囲を定めます。マスター・レコードは、ディテール・レコードの上に配置されます。<groupAbove> タグを使用する場合は、グループ上レイアウトに含めるフィールドをリストするために <field> タグをネストさせ、さらにマスター・グループを識別するために <group> タグもネストさせる必要があります。

詳細は、次の項を参照してください。

- [7.5.8 項「<field>」](#)
- [7.5.12 項「<group>」](#)

構文

このタグの構文は、次のとおりです。

```
<groupAbove
  name="style_name"
>
  <group>
    master_group_content
  </group>
  detail_group_content
</groupAbove>
```

例

次の例では、<groupAbove> レイアウトを含んだセクションを定義した XML レポート定義セグメントを示します。

```
<section name="main">
  <groupAbove name="m_emp">
    <labelAttribute font="Arial" fontSize="10" fontStyle="bold"/>
    <group>
      <field name="f_deptno" source="deptno" label="Department "
        font="Arial" fontSize="10"/>
      <field name="f_sumsal" label="Total Salary" source="sumsal"
        textColor="red" font="Arial" fontSize="10"
        fontStyle="bold">
        <labelAttribute font="Arial" fontSize="10" fontStyle="bold"
          textColor="red"/>
      </field>
    </group>
    <field name="f_ename" source="ename" label="Name"
      font="Arial" fontSize="10"/>
  </groupAbove>
</section>
```

```

        <field name="f_sal" source="sal" label="Salary"
            font="Arial" fontSize="10"/>
    </groupAbove>
</section>

```

7.5.14 <groupLeft>

説明

<groupLeft> タグは、レポートのレイアウトのセクション内でマスター / ディテール・スタイルの範囲を定めます。マスター・レコードは、ディテール・レコードの左に配置されます。<groupLeft> タグを使用する場合は、<groupLeft> レイアウトに含めるフィールドをリストするために <field> タグをネストさせ、さらにマスター・グループを識別するために <group> タグもネストさせる必要があります。

詳細は、次の項を参照してください。

- [7.5.8 項「<field>」](#)
- [7.5.12 項「<group>」](#)

構文

このタグの構文は、次のとおりです。

```

<groupLeft
    name="style_name"
>
    <group>
        master_group_content
    </group>
    detail_group_content
</groupLeft>

```

例

次の例では、グループ左レイアウトを含んだセクションを定義した XML レポート定義セグメントを示します。

```

<section name="main">
    <groupLeft name="m_emp">
        <labelAttribute font="Arial" fontSize="10" fontStyle="bold"/>
        <group>
            <field name="f_deptno" source="deptno" label="Department "
                font="Arial" fontSize="10"/>
            <field name="f_sumsal" label="Total Salary" source="sumsal"
                textColor="red" font="Arial" fontSize="10"
                fontStyle="bold">

```

```
        <labelAttribute font="Arial" fontSize="10" fontStyle="bold"
                        textColor="red"/>
    </field>
</group>
<field name="f_ename" source="ename" label="Name"
        font="Arial" fontSize="10"/>
<field name="f_sal" source="sal" label="Salary"
        font="Arial" fontSize="10"/>
</groupLeft>
</section>
```

7.5.15 <labelAttribute>

説明

<labelAttribute> タグは、フィールド・ラベル用の書式の属性を定義します。

<labelAttribute> タグは、<field> タグまたはレイアウト・スタイル・タグ (<tabular> または <matrix> など) の内部でネストできます。<labelAttribute> を <field> タグの内部でネストした場合は、そのフィールドのラベルのみに適用します。

<labelAttribute> タグは新規フィールドのみに作用し、RDF ファイルの既存フィールドのラベルは変更されません。既存ラベルのテキストを変更するには、<property> タグの textSegment 属性を使用する必要があります。

詳細は、次の項を参照してください。

- [7.5.8 項「<field>」](#)
- [7.5.25 項「<property>」](#)

構文

このタグの構文は、次のとおりです。

```
<labelAttribute
  [font="font_name"]
  [fontSize="point_size"]
  [fontStyle="regular | italic | bold | boldItalic"]
  [fontEffect="regular | strikeout | underline | strikeoutUnderline"]
  [lineColor="color_name | noLine"]
  [fillColor="color_name | noFill"]
  [textColor="color_name"]
  [alignment="start | left | center | right | end"]
>
</labelAttribute>
```


属性

次の表に、<labelAttribute> タグの属性を示します。

表 7-7 <labelAttribute> タグの属性

属性	必須 / 任意	説明
font	任意	フィールド・ラベルに使用するフォントの名前。
fontSize	任意	フィールド・ラベルに使用するフォントのサイズ。
fontStyle	任意	フィールド・ラベルに使用するフォントのスタイル。有効値は、次のとおりです。 <ul style="list-style-type: none">■ regular (標準)■ italic (斜体)■ bold (太字)■ boldItalic (太字斜体)
fontEffect	任意	フィールドの内容に使用するフォントの効果。有効値は、次のとおりです。 <ul style="list-style-type: none">■ regular (標準)■ strikeout (取消し線)■ underline (下線)■ strikeoutUnderline (取消し線と下線)
lineColor	任意	フィールドの境界線に使用するカラーの名前。noLine を指定した場合、フィールドの境界線は透明です (つまり、表示されません)。
fillColor	任意	フィールドの背景として使用するカラーの名前。noFill を指定した場合、背景は透明です。
textColor	任意	フィールドの内容に使用するカラーの名前。
alignment	任意	フィールド内の文字位置を揃える方法。有効値は、次のとおりです。 <ul style="list-style-type: none">■ start (始点詰め)■ left (左揃え)■ center (中央揃え)■ right (右揃え)■ end (終点詰め)

例

次の例では、グループ左レイアウトを含んだセクションを定義した XML レポート定義セグメントを示します。最初の `<labelAttribute>` タグは、専用に埋め込まれた `<labelAttribute>` タグを持った `f_sumsal` を除く、レイアウト内のすべてのフィールドに適用されます。

```
<section name="main">
  <groupLeft name="m_emp">
    <labelAttribute font="Arial" fontSize="10" fontStyle="bold"/>
    <group>
      <field name="f_deptno" source="deptno" label="Department "
        font="Arial" fontSize="10"/>
      <field name="f_sumsal" label="Total Salary" source="sumsal"
        textColor="red" font="Arial" fontSize="10"
        fontStyle="bold">
        <labelAttribute font="Arial" fontSize="10" fontStyle="bold"
          textColor="red"/>
      </field>
    </group>
    <field name="f_ename" source="ename" label="Name"
      font="Arial" fontSize="10"/>
    <field name="f_sal" source="sal" label="Salary"
      font="Arial" fontSize="10"/>
  </groupLeft>
</section>
```

7.5.16 <layout>

説明

`<layout>` タグは、レポート定義のレイアウトの始めと終わりを示します。

構文

このタグの構文は、次のとおりです。

```
<layout>
  content_of_layout
</layout>
```

例

次の例では、XML レポート定義のレイアウト・セグメントを示します。これは完全なレイアウト・モデルではなく、カスタマイズとして RDF ファイルに適用する必要があります。

```
<layout>
  <section name="main">
    <field name="f_trade_date"
```

```

        source="trade_date"
        formatMask="MM/DD/RR"/>
<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>

```

次の例では、別の XML レポート定義のレイアウト・セグメントを示します。これは完全なレイアウトであり、適切なデータ・モデルが存在する場合は、RDF ファイルに適用することなく単独で使用できます。

```

<layout>
<section name="main">
<matrix name="M_video_sales" template="corp10.tdf">
<group>
<field name="f_quarter" source="quarter" label="Quarter:"
        font="Arial" fontSize="8"
        formatTrigger="F_quarterFormatTrigger">
<labelAttribute font="Arial" fontSize="8"
                fontStyle="bold" textColor="black"/>
</field>
<field name="f_SumTOTAL_SALESPerQUARTER"
        source="SumTOTAL_SALESPerQUARTER"
        label="Qtrly: Sales: " font="Arial" fontSize="8"
        fontStyle="bold"
        formatMask="LNNNGNNNGNNNGNN0D00">
<labelAttribute font="Arial" fontSize="8"
                fontStyle="bold" textColor="black"/>
</field>
<field name="f_SumTOTAL_COSTPerQUARTER"
        source="SumTOTAL_COSTPerQUARTER"
        label="Costs: " font="Arial" fontSize="8" fontStyle="bold"
        formatMask="LNNNGNNNGNNNGNN0D00">
<labelAttribute font="Arial" fontSize="8"
                fontStyle="bold" textColor="black"/>
</field>
<field name="f_SumTOTAL_PROFITPerQUARTER"
        source="SumTOTAL_PROFITPerQUARTER"
        label="Profits: " font="Arial" fontSize="8" fontStyle="bold"
        formatMask="LNNNGNNNGNNNGNN0D00">
<labelAttribute font="Arial" fontSize="8"

```

```
        fontStyle="bold" textColor="black"/>
    </field>
</group>
<group>
    <field name="f_state" source="state" label="State:"
        font="Arial" fontSize="8">
        <labelAttribute font="Arial" fontSize="8"
            fontStyle="bold" textColor="black"/>
    </field>
</group>
<matrixCol name="g_city">
    <field name="f_city" source="city" label="City: "
        font="Arial" fontSize="8" textColor="yellow"
        formatTrigger="F_cityFormatTrigger"/>
    <field name="f_SumTOTAL_SALESPerCITY" source="SumTOTAL_SALESPerCITY"
        label="Sales: " font="Arial" fontSize="8" fontStyle="bold"
        textColor="yellow" formatMask="LNNNGNNNGNNNGN0D00">
        <labelAttribute font="Arial" fontSize="8"
            fontStyle="bold" textColor="yellow"/>
    </field>
    <field name="f_SumTOTAL_COSTPerCITY" source="SumTOTAL_COSTPerCITY"
        label="Sales: " font="Arial" fontSize="8" fontStyle="bold"
        textColor="yellow" formatMask="LNNNGNNNGNNNGN0D00">
        <labelAttribute font="Arial" fontSize="8"
            fontStyle="bold" textColor="yellow"/>
    </field>
    <field name="f_SumTOTAL_PROFITPerCITY"
        source="SumTOTAL_PROFITPerCITY"
        label="Sales: " font="Arial" fontSize="8" fontStyle="bold"
        textColor="yellow" formatMask="LNNNGNNNGNNNGN0D00">
        <labelAttribute font="Arial" fontSize="8"
            fontStyle="bold" textColor="yellow"/>
    </field>
</matrixCol>
<matrixRow name="g_product_category">
    <field name="f_product_category" source="product_category"
        label="Product Category" font="Arial" fontSize="8"/>
</matrixRow>
<matrixCell name="g_total_sales">
    <field name="f_total_sales" source="total_sales" label="Total Sales"
        font="Arial" fontSize="8" lineColor="noLine"
        formatMask="LNNNGNNNGNNNGN0D00"/>
    <field name="f_total_cost" source="total_cost" label="Total Cost"
        font="Arial" fontSize="8" lineColor="noLine"
        formatMask="LNNNGNNNGNNNGN0D00"/>
    <field name="f_total_profit" source="total_profit" label="Total Profit"
        font="Arial" fontSize="8" lineColor="noLine">
```

```

        formatMask="LNNNGNNNGNNNGNNOD00"/>
    </matrixCell>
</matrix>
</section>
</layout>

```

7.5.17 <link>

説明

<link> タグは、データ・モデルのデータ・ソース間のリンクを定義します。<link> タグは、<data> タグの内部でネストする必要があります。データ・ソースは、列単位でリンクさせます。したがって、それぞれの列リンクでは、親列属性と子列属性、および列どうしを関連付ける条件属性が必要です。2 つの表またはビューを結合するには、SELECT 文の外部キー列に列別名が必要です。(これらの別名は、列リンク仕様の親列および子列の参照で使われます。)

構文

このタグの構文は、次のとおりです。

```

<link
    parentGroup="parent_group_name"
    parentColumn="parent_column_name"
    childQuery="child_query_name"
    childColumn="child_column_name"
    condition="eq | lt | lteq | neq | gt | gteq | like | notLike"
    sqlClause="startWith | having | where"
    name="link_name"
>
</link>

```

属性

次の表に、<link> タグの属性を示します。

表 7-8 <link> タグの属性

属性	必須 / 任意	説明
parentGroup	グループ・リンクの場合は必須。 列リンクの場合は任意。	子問合せと関連付ける親グループの名前。
parentColumn	列リンクの場合は必須。 グループ・リンクの場合は無視。	子問合せの列（子列）と関連した親問合せの列の名前。
childQuery	グループ・リンクの場合は必須。 列リンクの場合は任意。	親グループと関連した子問合せの名前。
childColumn	列リンクの場合は必須。 グループ・リンクの場合は無視。	親問合せの列（親列）と関連した子問合せの列の名前。
condition	必須	親列と子列の関係を定義する SQL 演算子。 condition の値として可能なものは、次のとおりです。 <ul style="list-style-type: none">■ eq (equal to)■ lt (less than)■ lteq (less than or equal to)■ neq (not equal to)■ gt (greater than)■ gteq (greater than or equal to)■ Like (1 つの列の値が、他の列のパターンと一致した場合に条件が満たされます。パターンではパーセント記号 (%) およびアンダースコア () をワイルドカード文字として使用できます。)■ notLike (1 つの列の値が、他の列のパターンと一致しない場合に条件が満たされます。パターンではパーセント記号 (%) およびアンダースコア () をワイルドカード文字として使用できます。)
sqlClause	必須	親グループを子問合せと関連付ける SQL 句のタイプ。デフォルトは WHERE 句です。

例

次の例では、2つの問合せがリンクで結合されたレポート定義のデータ・モデル・セグメントを示します。

```
<data>
  <dataSource name="Q_dept">
    <select>
      select deptno deptno_dept from dept
    </select>
  </dataSource>
  <dataSource name="Q_emp">
    <select>
      select deptno deptno_emp, ename, empno, sal from emp
    </select>
  </dataSource>
  <link
    parentColumn="deptno_dept"
    childColumn="deptno_emp"
    condition="eq"
    sqlClause="where"/>
</data>
```

7.5.18 <matrix>

説明

<matrix> タグは、レポートのレイアウトのセクション内でマトリックス・スタイルの範囲を定めます。<matrix> タグを使用する場合は、マトリックス・レイアウトに含めるフィールドをリストするために <field> タグをネストさせ、さらにマトリックスの各部分を識別するために <matrixRow>、<matrixCol> および <matrixCell> タグもネストさせる必要があります。

<group> タグを <matrix> タグと一緒に使用して、グループ・スタイルを持ったマトリックスを作成することもできます。

詳細は、次の項を参照してください。

- [7.5.8 項「<field>」](#)
- [7.5.12 項「<group>」](#)
- [7.5.20 項「<matrixCol>」](#)
- [7.5.21 項「<matrixRow>」](#)
- [7.5.19 項「<matrixCell>」](#)

構文

このタグの構文は、次のとおりです。

```
<matrix
  name="style_name"
>
  [<group>
    master_group_content
  </group>]
  <matrixCol>
    matrix_column content
  </matrixCol>
  <matrixRow>
    matrix_row_content
  </matrixRow>
  <matrixCell>
    matrix_cell_content
  </matrixCell>
</matrix>
```

例

次の例に、グループ・レイアウトを持ったマトリックスを定義している XML レポート定義セグメントを示します。

```
<matrix name="M_video_sales" template="corp10.tdf">
  <group>
    <field name="f_quarter" source="quarter" label="Quarter:"
      font="Arial" fontSize="8"
      formatTrigger="F_quarterFormatTrigger">
        <labelAttribute font="Arial" fontSize="8"
          fontStyle="bold" textColor="black"/>
      </field>
    <field name="f_SumTOTAL_SALESPerQUARTER"
      source="SumTOTAL_SALESPerQUARTER"
      label="Qtrly: Sales: " font="Arial" fontSize="8"
      fontStyle="bold"
      formatMask="LNNNGNNNGNNNGNND00">
        <labelAttribute font="Arial" fontSize="8"
          fontStyle="bold" textColor="black"/>
      </field>
    <field name="f_SumTOTAL_COSTPerQUARTER" source="SumTOTAL_COSTPerQUARTER"
      label="Costs: " font="Arial" fontSize="8" fontStyle="bold"
      formatMask="LNNNGNNNGNNNGNND00">
        <labelAttribute font="Arial" fontSize="8"
          fontStyle="bold" textColor="black"/>
      </field>
    <field name="f_SumTOTAL_PROFITPerQUARTER"
```



```

        source="SumTOTAL_PROFITPerQUARTER"
        label="Profits: " font="Arial" fontSize="8" fontStyle="bold"
        formatMask="LNNNGNNNGNNNGNN0D00">
        <labelAttribute font="Arial" fontSize="8"
            fontStyle="bold" textColor="black"/>
    </field>
</group>
<group>
    <field name="f_state" source="state" label="State:"
        font="Arial" fontSize="8">
        <labelAttribute font="Arial" fontSize="8"
            fontStyle="bold" textColor="black"/>
    </field>
</group>
<matrixCol name="g_city">
    <field name="f_city" source="city" label="City: "
        font="Arial" fontSize="8" textColor="yellow"
        formatTrigger="F_cityFormatTrigger"/>
    <field name="f_SumTOTAL_SALESPerCITY" source="SumTOTAL_SALESPerCITY"
        label="Sales: " font="Arial" fontSize="8" fontStyle="bold"
        textColor="yellow" formatMask="LNNNGNNNGNNNGNN0D00">
        <labelAttribute font="Arial" fontSize="8"
            fontStyle="bold" textColor="yellow"/>
    </field>
    <field name="f_SumTOTAL_COSTPerCITY" source="SumTOTAL_COSTPerCITY"
        label="Sales: " font="Arial" fontSize="8" fontStyle="bold"
        textColor="yellow" formatMask="LNNNGNNNGNNNGNN0D00">
        <labelAttribute font="Arial" fontSize="8"
            fontStyle="bold" textColor="yellow"/>
    </field>
    <field name="f_SumTOTAL_PROFITPerCITY" source="SumTOTAL_PROFITPerCITY"
        label="Sales: " font="Arial" fontSize="8" fontStyle="bold"
        textColor="yellow" formatMask="LNNNGNNNGNNNGNN0D00">
        <labelAttribute font="Arial" fontSize="8"
            fontStyle="bold" textColor="yellow"/>
    </field>
</matrixCol>
<matrixRow name="g_product_category">
    <field name="f_product_category" source="product_category"
        label="Product Category" font="Arial" fontSize="8"/>
</matrixRow>
<matrixCell name="g_total_sales">
    <field name="f_total_sales" source="total_sales" label="Total Sales"
        font="Arial" fontSize="8" lineColor="noLine"
        formatMask="LNNNGNNNGNNNGNN0D00"/>
    <field name="f_total_cost" source="total_cost" label="Total Cost"
        font="Arial" fontSize="8" lineColor="noLine"

```

```
        formatMask="LNNNGNNNGNNNGNN0D00"/>
    <field name="f_total_profit" source="total_profit" label="Total Profit"
        font="Arial" fontSize="8" lineColor="noLine"
        formatMask="LNNNGNNNGNNNGNN0D00"/>
</matrixCell>
</matrix>
```

7.5.19 <matrixCell>

説明

<matrixCell> タグは、マトリックス・スタイル・レイアウトのセルの範囲を定めます。
<matrixCell> タグは、<matrix> タグの内部のみでネストできます。<matrixCell> タグ内で
<field> タグをネストして、マトリックス・セルとして含めるフィールドをリストする必要があります。

詳細は、次の項を参照してください。

- [7.5.8 項「<field>」](#)
- [7.5.18 項「<matrix>」](#)

構文

このタグの構文は、次のとおりです。

```
<matrixCell>
    master_group_content
</matrixCell>
```

例

次の例に、マトリックス・セルを定義している XML レポート定義セグメントを示します。

```
<matrixCell name="g_total_sales">
    <field name="f_total_sales" source="total_sales" label="Total Sales"
        font="Arial" fontSize="8" lineColor="noLine"
        formatMask="LNNNGNNNGNNNGNN0D00"/>
    <field name="f_total_cost" source="total_cost" label="Total Cost"
        font="Arial" fontSize="8" lineColor="noLine"
        formatMask="LNNNGNNNGNNNGNN0D00"/>
    <field name="f_total_profit" source="total_profit" label="Total Profit"
        font="Arial" fontSize="8" lineColor="noLine"
        formatMask="LNNNGNNNGNNNGNN0D00"/>
</matrixCell>
```

7.5.20 <matrixCol>

説明

<matrixCol> タグは、マトリックス・スタイル・レイアウトの列フィールドの範囲を定めます。<matrixCol> タグは、<matrix> タグの内部のみ でネストできます。<matrixCol> タグ内で <field> タグをネストして、マトリックス列として含めるフィールドをリストする必要があります。

詳細は、次の項を参照してください。

- [7.5.8 項「<field>」](#)
- [7.5.18 項「<matrix>」](#)

構文

このタグの構文は、次のとおりです。

```
<matrixCol>
  master_group_content
</matrixCol>
```

例

次の例に、マトリックス・レイアウトの列寸法を定義している XML レポート定義セグメントを示します。

```
<matrixCol name="g_city">
  <field name="f_city" source="city" label="City: "
    font="Arial" fontSize="8" textColor="yellow"
    formatTrigger="F_cityFormatTrigger"/>
  <field name="f_SumTOTAL_SALESPerCITY" source="SumTOTAL_SALESPerCITY"
    label="Sales: " font="Arial" fontSize="8" fontStyle="bold"
    textColor="yellow" formatMask="LNNNGNNGNNGNNGN0D00">
    <labelAttribute font="Arial" fontSize="8"
      fontStyle="bold" textColor="yellow"/>
  </field>
  <field name="f_SumTOTAL_COSTPerCITY" source="SumTOTAL_COSTPerCITY"
    label="Sales: " font="Arial" fontSize="8" fontStyle="bold"
    textColor="yellow" formatMask="LNNNGNNGNNGNNGN0D00">
    <labelAttribute font="Arial" fontSize="8"
      fontStyle="bold" textColor="yellow"/>
  </field>
  <field name="f_SumTOTAL_PROFITPerCITY" source="SumTOTAL_PROFITPerCITY"
    label="Sales: " font="Arial" fontSize="8" fontStyle="bold"
    textColor="yellow" formatMask="LNNNGNNGNNGNNGN0D00">
    <labelAttribute font="Arial" fontSize="8"
      fontStyle="bold" textColor="yellow"/>
  </field>
```

```
    </field>
  </matrixCol>
```

7.5.21 <matrixRow>

説明

<matrixRow> タグは、マトリックス・スタイル・レイアウトの行フィールドの範囲を定めます。<matrixRow> タグは、<matrix> タグの内部のみでネストできます。<matrixRow> タグ内で <field> タグをネストして、マトリックス行として含めるフィールドをリストする必要があります。

詳細は、次の項を参照してください。

- [7.5.8 項「<field>」](#)
- [7.5.18 項「<matrix>」](#)

構文

このタグの構文は、次のとおりです。

```
<matrixRow>
  master_group_content
</matrixRow>
```

例

次の例に、マトリックス・レイアウトの行寸法を定義している XML レポート定義セグメントを示します。

```
<matrixRow name="g_product_category">
  <field name="f_product_category" source="product_category"
    label="Product Category" font="Arial" fontSize="8"/>
</matrixRow>
```

7.5.22 <object>

説明

<object> タグは、プロパティを変更するレポート内のオブジェクトを識別します。<object> タグの内部に <properties> および <property> タグをネストさせるのが一般的です。

構文

このタグの構文は、次のとおりです。

```
<object
  name="object_name"
  type="REP_REPORT | REP_GROUP | REP_COL_MAP | REP_GRAPHIC_TEXT"
>
  property_definitions
</object>
```

属性

次の表に、<object> タグの属性を示します。

表 7-9 <object> タグのプロパティ

属性	必須 / 任意	説明
name	必須	プロパティを適用するオブジェクトの識別子。
type	必須	プロパティを適用するオブジェクトの種類。 <ul style="list-style-type: none">■ REP_REPORT はレポート自体です。■ REP_GROUP はレポートのデータ・モデル内のグループです。■ REP_COL_MAP はレポートのデータ・モデル内の列です。■ REP_GRAPHIC_TEXT はレポートのレイアウト内のボイラプレート・オブジェクトです。

例

次の例に、オブジェクト・プロパティを定義している XML レポート定義セグメントを示します。

```
<customize>
  <object name="videosales" type="REP_REPORT">
    <properties>
      <property name="beforeReportType">File</property>
      <property name="beforeReportValue">
        d:¥xml_reps¥header_example.html
      </property>
      <property name="afterReportType">Text</property>
      <property name="afterReportValue">
        <![CDATA[
          <center>
            <font face="Arial,Helvetica"><font size=-1><font color="#000000">
              Send questions to <a href="mailto:your_email_id">YourNameHere</a>.
              <br>&nbsp;
            </font>
          </center>
        </body>
        </html>
        ]]>
      </property>
    </properties>
  </object>
</customize>
```

次の例に示す XML レポート定義セグメントは、ボイラープレート・テキストの一部を変更します。これは、既存フィールドのラベルを変更する場合に役立ちます。

```
<customize>
  <object name="B_high_365" type="REP_GRAPHIC_TEXT">
    <properties>
      <property name="textSegment">High</property>
    </properties>
  </object>
  <object name="B_low_365" type="REP_GRAPHIC_TEXT">
    <properties>
      <property name="textSegment">Low</property>
    </properties>
  </object>
</customize>
```

7.5.23 <programUnits>

説明

<programUnits> タグは、レポート定義に追加するあらゆる PL/SQL の範囲を定めます。

<programUnits> タグの内部に <function> タグをネストさせるのが一般的です。

詳細は、[7.5.11 項「<function>」](#)を参照してください。

構文

このタグの構文は、次のとおりです。

```
<programUnits>
  program_unit_definitions
</programUnits>
```

例

次に、ある PL/SQL を定義した XML レポート定義セグメントを示します。

<programUnits> タグは <layout> タグの外側にあり、formatTrigger 属性を使用してレイアウトのフィールドから関数が参照されます。

```
<layout>
  <section name="header">
    <field name="F_ssn1"
      source="ssn1"
      formatTrigger="F_ssn1FormatTrigger"/>
  </section>
  <section name="main">
    <field name="F_ssn"
      source="ssn"
      formatTrigger="F_ssnFormatTrigger"/>
  </section>
</layout>
<programUnits>
  <function name="F_ssn1FormatTrigger">
    <![CDATA[
      function F_ssn1FormatTrigger return boolean is
      begin
        SRW.SET_HYPERLINK('#EMP_DETAILS_&' || LTRIM(TO_CHAR(:SSN)) ||
'>');
        return (TRUE);
      end;
    ]]>
  </function>
  <function name="F_ssnFormatTrigger">
    <![CDATA[
```

```
function F_ssnFormatTrigger return boolean is
begin
    SRW.SET_LINKTAG('EMP_DETAILS_&' || LTRIM(TO_CHAR(:SSN)) || '>');
    return (TRUE);
end;
]]>
</function>
</programUnits>
```

7.5.24 <properties>

説明

<properties> タグは、オブジェクトのプロパティの範囲を定めます。<properties> タグは、<object> タグの内部にネストさせる必要があり、通常はそこに <property> タグをネストさせます。

構文

このタグの構文は、次のとおりです。

```
<properties>
  property_definitions
</properties>
```

例

次の例に、オブジェクトのプロパティを定義している XML レポート定義のセグメントを示します。

```
<customize>
  <object name="videosales" type="REP_REPORT">
    <properties>
      <property name="beforeReportType">File</property>
      <property name="beforeReportValue">
        d:¥xml_reps¥header_example.html
      </property>
      <property name="afterReportType">Text</property>
      <property name="afterReportValue">
        <![CDATA[
          <center>
            <font face="Arial,Helvetica"><font size=-1><font color="#000000">
              Send questions to <a href="mailto:your_email_id">YourNameHere</a>.
            <br>&nbsp;
            </font>
          </center>
        </body>
```



```

    </html>
  ]>
</property>
</properties>
</object>
</customize>

```

次の例に示す XML レポート定義セグメントは、ボイラプレート・テキストの一部を変更します。これは、既存フィールドのラベルを変更する場合に役立ちます。

```

<customize>
  <object name="B_high_365" type="REP_GRAPHIC_TEXT">
    <properties>
      <property name="textSegment">High</property>
    </properties>
  </object>
  <object name="B_low_365" type="REP_GRAPHIC_TEXT">
    <properties>
      <property name="textSegment">Low</property>
    </properties>
  </object>
</customize>

```

7.5.25 <property>

説明

<property> タグは、オブジェクトの 1 つのプロパティの範囲を定めます。<property> タグは、<properties> タグの内部にネストさせる必要があり、通常はプロパティの値を定義するために、その内部にテキストをネストさせます。

構文

このタグの構文は、次のとおりです。

```

<property
  name="xmlTag | xmlAttribute | xmlSuppress | prologType | prolog |
    beforeReportValue | beforeReportType | afterReportValue | afterReportType |
    beforePageValue | beforePageType | afterPageValue | afterPageType |
    beforeFormValue | beforeFormType | afterFormValue | afterFormType |
    pageNavigationControlValue | pageNavigationControlType | textSegment
>
  property_value
</property>

```

属性

次の表に、<property> タグの属性を示します。

表 7-10 <property> タグの属性

属性	必須 / 任意	説明
name	必須	指定するプロパティの名前。使用可能なプロパティは、オブジェクト・タイプごとに異なります。詳細は、「 使用上の注意 」を参照してください。

使用上の注意

次の表に、それぞれのオブジェクト・タイプで使用可能なプロパティをリストします。

表 7-11 オブジェクト・タイプに有効なプロパティ

オブジェクト	有効なプロパティ
レポート・オブジェクト (REP_REPORT)	<ul style="list-style-type: none">xmlTagxmlAttributexmlSuppressprologTypeprologbeforeReportValuebeforeReportTypeafterReportValueafterReportTypebeforePageValuebeforePageTypeafterPageValueafterPageTypebeforeFormValuebeforeFormTypeafterFormValueafterFormTypepageNavigationControlValuepageNavigationControlType

表 7-11 オブジェクト・タイプに有効なプロパティ

オブジェクト	有効なプロパティ
グループ・オブジェクト (REP_GROUP)	<ul style="list-style-type: none">■ xmlTag■ xmlAttribute■ outerXMLTag■ outerXMLAttribute■ xmlSuppress
列オブジェクト (REP_COL_MAP)	<ul style="list-style-type: none">■ xmlTag■ xmlAttribute■ XMLSuppress■ containXML
バイラプレート・オブジェクト (REP_GRAPHIC_TEXT)	<ul style="list-style-type: none">■ textSegment

例

次の例に、オブジェクトのプロパティを定義している XML レポート定義のセグメントを示します。

```
<customize>
  <object name="videosales" type="REP_REPORT">
    <properties>
      <property name="beforeReportType">File</property>
      <property name="beforeReportValue">
        d:¥xml_reps¥header_example.html
      </property>
      <property name="afterReportType">Text</property>
      <property name="afterReportValue">
        <![CDATA[
          <center>
            <font face="Arial,Helvetica"><font size=-1><font color="#000000">
              Send questions to <a href="mailto:your_email_id">YourNameHere</a>.
            <br>&nbsp;
            </font>
          </center>
        </body>
        </html>
      ]]>
    </property>
  </properties>
</object>
</customize>
```

次の例に、ボイラプレート・オブジェクトのテキストを変更するカスタマイズ・セクションを示します。これは、既存フィールドのラベルを変更する場合に役立ちます。

```
<customize>
  <object name="B_high_365" type="REP_GRAPHIC_TEXT">
    <properties>
      <property name="textSegment">High</property>
    </properties>
  </object>
  <object name="B_low_365" type="REP_GRAPHIC_TEXT">
    <properties>
      <property name="textSegment">Low</property>
    </properties>
  </object>
</customize>
```

7.5.26 <report>

説明

<report> タグは、レポート定義の始めと終わりを示します。レポート全体に適用する属性を <report> タグに付加できます。

構文

このタグの構文は、次のとおりです。

```
<report DTDVersion=1.0"
  [name="report_name"]
  [title="report_title"]
  [author="author_name"]
>
  content_of_report
</report>
```

例

この例では、cond.rdf という名前の RDF ファイルに適用する XML カスタマイズ・ドキュメントを示しています。この例は、データ・モデルにいったい触れていません。レイアウト内の一部のフィールドの書式を変更するのみです。

```

<report name="cond" DTDVersion="1.0">
<!-- This report assumes that the file
      named header_example.html is located
      in d:\ORANT\TOOLS\DOC60\US\RB60.
      If it is not located there, the report
      will not run properly.
-->
<layout>
  <section name="main">
    <field name="f_trade_date"
          source="trade_date"
          formatMask="MM/DD/RR"/>
    <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>
<customize>
  <object name="videosales" type="REP_REPORT">
    <properties>
      <property name="beforeReportType">File</property>
      <property name="beforeReportValue">
        d:\xml_reps\header_example.html
      </property>
      <property name="afterReportType">Text</property>
      <property name="afterReportValue">
        <![CDATA[
          <center>
            <font face="Arial,Helvetica"><font size=-1><font color="#000000">
              Send questions to <a href="mailto:your_email_id">YourNameHere</a>.
            <br>&nbsp;
          </font>
        </center>
        </body>
        </html>
        ]]>
      </property>
    </properties>
  </object>
</customize>
</report>

```

属性

次の表に、<report> タグの属性を示します。

表 7-12 <report> タグの属性

属性	必須 / 任意	説明
name	任意	レポート名を記録します。名前を指定しなかった場合は、デフォルトにより UNTITLED に設定されます。レポート定義を RDF ファイルに適用する予定がある場合は、ファイル名から RDF 拡張子を除いたものと同じ名前にする必要があります。
dtdVer	必須	この XML レポート定義の生成に使用した Oracle Reports Services DTD のバージョンを記録します。DTD はバージョンごとに変更される可能性があるため、新規定義するすべてのレポートに使用バージョンの情報を入れる必要があります。これにより、将来のリリースで下位互換性を確保できます。
title	任意	指定したタイトルがレポートの先頭に配置されます。RDF ファイルで定義タイトルを適用すると、既存のレポート・タイトルより優先されます。
author	任意	作成者名を記録します。

7.5.27 <section>

説明

<section> タグは、レポート定義のレイアウト内の特定セクションの始めと終わりを示します。<section> タグは、<layout> タグの内部でネストする必要があります。レポートでは、レイアウト内に最大 3 セクションまで許可されます。

次のタグを使用して、各セクションにレイアウト・スタイルを定義することもできます。

- [7.5.30 項「<tabular>」](#)
- [7.5.18 項「<matrix>」](#)
- [7.5.9 項「<formLike>」](#)
- [7.5.13 項「<groupAbove>」](#)
- [7.5.14 項「<groupLeft>」](#)

構文

このタグの構文は、次のとおりです。

```
<section
  name= "header | main | trailer"
  width="section_width"
  height="section_height"
>
  section_contents
</section>
```

属性

次の表に、<section> タグの属性を示します。

表 7-13 <section> タグの属性

属性	必須 / 任意	説明
name	必須	セクションの名前。header、main または trailer です。
width	任意	レポートの長さの単位で示された 1 つの物理 ページ（余白を含む）の幅（たとえば、8.5 インチ）。
height	任意	レポートの長さの単位で示された 1 つの物理 ページ（余白を含む）の高さ（たとえば、11 インチ）。

例

次に、<section> の定義例を示します。

```
<layout>
  <section name="header">
    <field name="F_ssn1"
      source="ssn"
      formatTrigger="F_ssn1FormatTrigger"/>
  </section>
  <section name="main">
    <field name="F_ssn"
      source="ssn"
      formatTrigger="F_ssnFormatTrigger"/>
  </section>
</layout>
```

7.5.28 <select>

説明

<select> タグは、データ・モデル内の SELECT 文の始めと終わりを示します。<select> は、<dataSource> タグの内部でネストする必要があります。

構文

このタグの構文は、次のとおりです。

```
<select>
  content_of_SELECT
</select>
```

例

次の例では、XML レポート定義のデータ・ソースのセグメントを示します。

```
<data>
  <dataSource name="q_category">
    <select>
      SELECT          ic.category,
                      SUM (h.sales),
                      AVG (h.high_365),
                      AVG (h.low_365),
                      AVG (h.div),
                      AVG (h.p_e)
      FROM stock_history h, indcat ic
      WHERE h.symbol=ic.symbol
      GROUP BY ic.category
    </select>
  </dataSource>
</data>
```

このタグをバインド参照として SELECT 文に入れておくと、ユーザー・パラメータが自動的に生成されます。次に例を示します。

```
<select>
  select * from dept where deptno > :p_dept;
</select>
```

この SELECT 文により、p_dept という名前のユーザー・パラメータが自動的に生成されます。したがって、このパラメータをレポート定義の中で手動で作成する必要はありません。

次の XML レポート定義セグメント例では、<![CDATA[]]> タグを使用して、> 記号を含んだ SQL 文を保護しています。

```
<select>
  <![CDATA[
    SELECT ALL VIDEO_CATEGORY_BY_QTR.QUARTER,
           VIDEO_CATEGORY_BY_QTR.TOTAL_PROFIT
    FROM SCOTT.VIDEO_CATEGORY_BY_QTR
    WHERE (VIDEO_CATEGORY_BY_QTR.SALES_REGION='West'
           AND VIDEO_CATEGORY_BY_QTR.TOTAL_PROFIT>2000)
  ]]>
</select>
```

7.5.29 <summary>

説明

<summary> タグは、レポート定義のデータ・モデルのサマリー列を定義します。サマリー列は、別の列のデータ値に基づいて算術関数を実行する場合に使用します。標準サマリー関数の 1 つではない関数を実行する場合は、<formula> タグを使用して、PL/SQL を使用した複雑な計算を実行する式列を作成できます。

詳細は、[7.5.10 項「<formula>」](#)を参照してください。

構文

このタグの構文は、次のとおりです。

```
<summary
  source="src_col_name"
  function="sum|average|minimum|maximum|count|first|last|pctTotal|stddeviation
           |variance"
  compute="group+names"
  reset="group_name"
  productOrder="group_name"
  nullval="value_if_null"
/>
```

属性

次の表に、<summary> タグの属性を示します。

表 7-14 <summary> タグの属性

属性	必須 / 任意	説明
source	必須	値が集計対象になる列の名前。
function	任意	サマリー値を出力するために適用する算術演算。 <ul style="list-style-type: none">■ average は、リセット・グループ内の列値の平均を計算します。■ count は、リセット・グループ内のレコードの件数をカウントします。■ first は、リセット・グループについてフェッチされた列の最初の値を印刷します。■ last は、リセット・グループについてフェッチされた列の最後の値を印刷します。■ maximum は、リセット・グループ内の列の最大値を計算します。■ minimum は、リセット・グループ内の列の最小値を計算します。■ pctTotal は、リセット・グループ内の列の合計パーセントを計算します。■ stddeviation は、リセット・グループについて列の正の平方偏差の根を計算します。■ sum は、リセット・グループ内の列値の合計を計算します。■ variance は、列の各値とリセット・グループの平均値との差を 2 乗して合計したものを、値の件数から 1 を引いた数値で割ります。

表 7-14 <summary> タグの属性

属性	必須 / 任意	説明
compute	任意	<p>合計パーセント・サマリー列の計算対象となるグループ。compute は、合計パーセントの関数が入っている列のみで使います。この値により、ソース列の値がパーセントになっているものの合計を判断できます。パーセントを計算するときは、値を合計で割ります（たとえば、SMITH 氏の給与 ÷ 部署の総支給額）。compute は、パーセントの計算について合計を定義します。マトリックス・レポートの場合は、Compute At の計算対象に複数のグループを指定できます。</p> <p>各ページの合計値またはレポート全体のパーセントを計算する場合は、この属性をページまたはレポートにも設定できます。</p>
reset	任意	<p>サマリー列値がゼロ（function が Count の場合）null（function が Count 以外の場合）または nullval（サマリーが null 値を 1 つ持っている場合）にリセットされるグループ。reset は、サマリーが継続的サマリーなのか、あるいは定期的（たとえば、グループ・レベル）サマリーなのか判断します。</p> <p>各ページの合計値またはレポート全体のパーセントを計算する場合は、この属性をページまたはレポートにも設定できます。</p>
productOrder	任意	<p>サマリーのクロス積で各グループを評価する順序。productOrder は、クロス積グループのサマリー、式またはブレースホルダの頻度も定義します。つまり、サマリー、式またはブレースホルダは、productOrder に関与するグループの値のそれぞれの組合せについて 1 つの値を持っています。</p> <p>productOrder 属性は、クロス積グループが所有する列に対してのみ使います。クロス積には複数のグループが関連するので、いくつもの異なる順序によってクロス積のグループを評価できます。したがって、クロス積のサマリーを作成するときは、productOrder を使用してグループが評価される順序を指定する必要があります。また、productOrder を使用して、クロス積内部のサマリー、式またはブレースホルダの頻度を指定する必要があります。</p>
nullval	任意	<p>列のあらゆる null 値と置き換える値。たとえば、このフィールドに X を入力すると、列でフェッチされた null 値が X に置き換えられて表示されます。空白のままにすると、null 値は置換されません。</p>

デフォルト値

通常、<summary> タグの任意の属性は、実行時にデフォルト値が有効になるので、何も指定する必要はありません。任意の値を指定する必要があるのは、デフォルトを無効にする場合のみです。次の表に、それぞれのレイアウト・スタイルについて任意の属性のデフォルト値を示します。

表 7-15 ブレーク・グループのサマリーのデフォルト値

任意の属性	デフォルト値
function	合計
compute	サマリー列のグループの親グループ
reset	サマリー列のグループの親グループ

表 7-16 マトリックス・レポートのサマリーのデフォルト値

任意の属性	デフォルト値
function	合計
compute	クロス積グループ
productOrder	<ul style="list-style-type: none">サマリーが入っているグループ（ディメンション・サマリーの場合）マトリックス行を定義するグループのリスト（セル・サマリーの場合）
reset	最も頻度の高い productOrder のグループ

例

次に、2 つの問合せからなるデータ・モデルについて、いくつかのサマリーの例を示します。最初の 3 つのサマリーは表レイアウトのもので、残りの 6 つはマトリックス・ブレーク・レポートのものです。名前、ソース列および関数のみが指定されているので、レイアウトでの関連フィールドの位置に基づいて、デフォルトのアルゴリズムが列をそれぞれ適切なグループに配置しています。

```
<data>
  <dataSource name="Q_1">
    <select>
      SELECT ALL VIDEO_CATEGORY_BY_QTR.QUARTER,
              VIDEO_CATEGORY_BY_QTR.SALES_REGION,
              VIDEO_CATEGORY_BY_QTR.STATE, VIDEO_CATEGORY_BY_QTR.CITY,
              VIDEO_CATEGORY_BY_QTR.PRODUCT_CATEGORY,
              VIDEO_CATEGORY_BY_QTR.TOTAL_SALES,
              VIDEO_CATEGORY_BY_QTR.TOTAL_COST,
              VIDEO_CATEGORY_BY_QTR.TOTAL_PROFIT
```

```

FROM SCOTT.VIDEO_CATEGORY_BY_QTR
WHERE VIDEO_CATEGORY_BY_QTR.SALES_REGION='West '
</select>
</dataSource>
<dataSource name="Q_2">
<select>
SELECT ALL VIDEO_CATEGORY_BY_QTR.QUARTER, VIDEO_CATEGORY_BY_QTR.CITY,
        VIDEO_CATEGORY_BY_QTR.PRODUCT_CATEGORY,
        VIDEO_CATEGORY_BY_QTR.TOTAL_PROFIT,
        VIDEO_CATEGORY_BY_QTR.TOTAL_SALES,
        VIDEO_CATEGORY_BY_QTR.TOTAL_COST
FROM SCOTT.VIDEO_CATEGORY_BY_QTR
WHERE VIDEO_CATEGORY_BY_QTR.SALES_REGION='West '
</select>
</dataSource>
<summary name="SumTOTAL_SALESPerCITY1" source="total_sales1"/>
<summary name="SumTOTAL_COSTPerCITY1" source="total_cost1"/>
<summary name="SumTOTAL_PROFITPerCITY1" source="total_profit1"/>
<summary name="SumTOTAL_SALESPerQUARTER" source="total_sales"/>
<summary name="SumTOTAL_COSTPerQUARTER" source="total_cost"/>
<summary name="SumTOTAL_PROFITPerQUARTER" source="total_profit"/>
<summary name="SumTOTAL_SALESPerCITY" source="total_sales"/>
<summary name="SumTOTAL_COSTPerCITY" source="total_cost"/>
<summary name="SumTOTAL_PROFITPerCITY" source="total_profit"/>
<formula name="Profit_Margin" source="FormulaProfitMargin"
datatype="number"
        width="9"/>
</data>

```

7.5.30 <tabular>

説明

<tabular> タグは、レポートのレイアウトのセクション内で表スタイルの範囲を定めます。
 <tabular> タグを使用する場合は、<field> タグもネストさせて表レイアウトに入れるフィールドをリストする必要があります。

詳細は、[7.5.8 項「<field>」](#)を参照してください。

構文

このタグの構文は、次のとおりです。

```
<tabular>
  <field>
  </field>
  [...]
</tabular>
```

例

次の例では、表レイアウトを持ったセクションを定義している XML レポート定義セグメントを示します。

```
<section name="header"> "  
<tabular name="M_summary" template="corp2.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>
```

第 II 部

付録

付録 A 「カレンダーの定義によるレポートへのユーザー・アクセスの管理」

付録 B 「RWCLI60 コマンドライン引数」

付録 C 「Oracle Reports Services 構成パラメータ」

付録 D 「環境変数」

付録 E 「データベース接続文字列」

付録 F 「Web カートリッジから CGI への移行」

付録 G 「トラブルシューティング」

カレンダーの定義によるレポートへの ユーザー・アクセスの管理

第 5 章「Oracle Portal を使用した Oracle Reports Services セキュリティ」で説明されているように、アクセス制御では、Oracle Reports Services で実行するレポートへのユーザー・アクセスを制限できます。Oracle Reports Services では、Oracle Portal によりセキュリティ・チェックを実行して、アクセスの制限された Oracle Reports Services サーバーやプリンタ上でレポートを実行するときに、必要な権限を持つユーザーのみにアクセスを許可します。カレンダーの定義はオプションのステップで、レポート定義ファイル (RDF)、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決めることにより、これらへのアクセスをさらに制限できます。

A.1 使用可能日カレンダーの作成

使用可能日カレンダーは、RDF、Oracle Reports Services サーバーおよびプリンタを使用して処理できる時間を決定します。RDF、Oracle Reports Services サーバーおよびプリンタが常に使用可能であり、いつでも処理できる場合、使用可能日カレンダーは不要です。

次の 2 種類の使用可能日カレンダーを作成できます。

- シンプル

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、毎日、日曜日～土曜日の午前 12:00 ～午後 10:00 など）。

- 結合

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（たとえば、デイリー・カレンダーとメンテナンス・カレンダーの結合）。

RDF、Oracle Reports Services サーバーまたはプリンタに関連付けられ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する必要があります。

A.2 使用可能日カレンダーの例

この例では、すべての曜日、定期メンテナンス日および休日の使用可能日を決定する運用カレンダーを作成します。運用カレンダーを作成するには、次の使用可能日カレンダーを作成する必要があります。

- 日曜日～土曜日の毎日午前 12:00 ～ 午後 10:00 を使用可能期間として指定したシンプルなデイリー・カレンダー。
- 毎週土曜日の午後 3:00 ～ 午後 10:00 を使用可能期間として指定したシンプルなメンテナンス・カレンダー。
- 12 月 25 日午前 12:00 ～ 12 月 26 日午前 12:00 を使用可能期間として指定したシンプルなクリスマス・カレンダー。
- 前述のカレンダーをすべて結合した後に、メンテナンスおよびクリスマス・カレンダーを除外した運用カレンダーを作成します。カレンダーを除外すると、それらのカレンダーの使用可能日ルールに基づいた処理が禁止されます。

A.2.1 デイリー・カレンダーの作成

次のことを行い、日曜日～土曜日の午前 12:00 ～ 午後 10:00 を使用可能期間として指定したデイリー・カレンダーを作成します。

1. Oracle Portal にアクセスしてログインします。Oracle Reports Services セキュリティ・ウィザードにアクセスするには、RW_ADMINISTRATOR 権限および DBA 権限が必要です。
2. Oracle Portal ホーム・ページで、「**管理**」タブをクリックします。
3. 「**管理**」ページの「Oracle Reports Security」ポートレットで、「Oracle Reports Security の設定」をクリックします。
4. 「Oracle Reports Security の設定」ページの「Reports カレンダー・アクセス」ポートレットで、「**シンプルな Reports カレンダー・アクセスを作成**」をクリックしてシンプルなカレンダーを作成します。
5. 「**シンプルな使用可能日カレンダー**」ページで、「**カレンダー名**」フィールドに「Daily」と入力します。デイリー・カレンダーがすでに存在する場合は、末尾に自分のイニシャルを付け足してください（たとえば、DailyAA）。
6. 「**次へ**」ボタンをクリックして続行します。
7. この画面では、**使用可能な日付 / 時間**を設定します。「**期間**」に、今日の日付を開始年月日として指定し、午前 12:00 を開始時刻として指定します。今日の日付を終了年月日として指定し、午後 10:00 を終了時刻として指定します。
8. 「**繰り返し**」オプションとして「**毎日**」を選択します。これにより、**期間パターン**が毎日繰り返されます。たとえば、開始日付を 2000 年 1 月 4 日、月曜日とすると、この日付からパターンが完了するまで、このパターンが毎日繰り返されます。

- 9. 「次へ」ボタンをクリックして続行します。
- 10. 「シンプルな使用可能日カレンダーのサマリー」ページで、「カレンダーを表示」をクリックしてデイリー・カレンダーを表示することもできます。緑色の部分は使用可能時間を示します。見終わったら、カレンダーをクローズします。
- 11. 「完了」ボタンをクリックします。
- 12. 「コンポーネントの管理」画面が表示されます。「クローズ」ボタンをクリックします。

A.2.2 メンテナンス・カレンダーの作成

毎週土曜日の午後 3:00 ~ 午後 10:00 を使用可能期間として指定したメンテナンス・カレンダーを作成します。後述のステップで、このカレンダーを運用カレンダーに一度追加した後、除外します。これにより、このカレンダーに指定された日付と時刻に基づいた処理が禁止されます。

- 1. 「Oracle Reports Security の設定」ページの「Reports カレンダー・アクセス」ポートレットから、「シンプルな Reports カレンダー・アクセスを作成」オプションをクリックしてシンプルなカレンダーを作成します。
- 2. 「シンプルな使用可能日カレンダー」ページで、「カレンダー名」フィールドに「Maintenance」と入力します。メンテナンス・カレンダーがすでに存在する場合は、末尾に自分のイニシャルを付け足してください（たとえば、MaintenanceAA ）。
- 3. 「次へ」ボタンをクリックして続行します。
- 4. 使用可能な日付 / 時間に次のものを定義します。

表 A-1 メンテナンス・カレンダー・ルール

フィールド	値
期間	
開始	開始日付は土曜日の日付（たとえば、2000 年 1 月 8 日）とし、時刻は午後 3:00 を指定します。
終了	開始日付と同じ日付を指定し、終了時刻は午後 10:00 にします。
繰返し	「毎週」を選択します。

- 5. 「次へ」ボタンをクリックして続行します。
- 6. 「シンプルな使用可能日カレンダーのサマリー」ページで、「カレンダーを表示」をクリックしてメンテナンス・カレンダーを表示することもできます。緑色の部分は使用可能時間を示します。見終わったら、カレンダーをクローズします。
- 7. 「完了」ボタンをクリックします。
- 8. 「コンポーネントの管理」画面が表示されます。「クローズ」ボタンをクリックします。

A.2.3 クリスマス・カレンダーの作成

毎年 12 月 25 日午前 12:00 ~ 12 月 26 日午前 12:00 を使用可能期間として指定したクリスマス・カレンダーを作成します。後述のステップで、このカレンダーを運用カレンダーに一度追加した後、除外します。これにより、このカレンダーに指定された日付と時刻に基づいた処理が禁止されます。

- 1. 「Oracle Reports Security の設定」ページの「Reports カレンダー・アクセス」ポートレットから、「シンプルな Reports カレンダー・アクセスを作成」オプションをクリックしてカレンダーを作成します。
- 2. 「シンプルな使用可能日カレンダー」ページで、「カレンダー名」フィールドに「Christmas」と入力します。クリスマス・カレンダーがすでに存在する場合は、末尾に自分のイニシャルを付け足してください（たとえば、ChristmasAA）。
- 3. 「次へ」ボタンをクリックして続行します。
- 4. 使用可能な日付 / 時間に次のものを定義します。

フィールド	値
期間	
開始	12 月 25 日、午前 12:00 を指定します。
終了	12 月 26 日、午前 12:00 を指定します。
繰り返し	「毎年」を選択します。

- 5. 「次へ」ボタンをクリックして続行します。
- 6. 「シンプルな使用可能日カレンダーのサマリー」ページで、「カレンダーを表示」をクリックしてクリスマス・カレンダーを表示することもできます。緑色の部分は使用可能時間を示します。見終わったら、カレンダーをクローズします。
- 7. 「完了」ボタンをクリックします。
- 8. 「コンポーネントの管理」画面が表示されます。「クローズ」ボタンをクリックします。

A.2.4 結合された使用可能日カレンダーの作成

この例では、デイリー、メンテナンスおよびクリスマス・カレンダーを結合して運用カレンダーを作成した後に、メンテナンスおよびクリスマス・カレンダーを除外して、それらの使用可能日ルールに基づいた処理を禁止します。

1. 「Oracle Reports Security の設定」ページから、「**結合された Reports カレンダー・アクセスを作成**」をクリックして、作成した 3 つのカレンダーを 1 つに結合するカレンダーを作成します。
2. 「**結合された使用可能日カレンダー**」ページで、「**カレンダー名**」フィールドに「Production」と入力します。運用カレンダーがすでに存在する場合は、末尾に自分のイニシャルを付け足してください（たとえば、ProductionAA）。
3. 「次へ」ボタンをクリックして続行します。
4. 「**使用可能日カレンダーを選択**」ページで、[Ctrl] キーを押しながら「**使用可能日カレンダー**」リスト・ボックスの Daily、Maintenance および Christmas の各カレンダーをクリックします。
5. 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・ボックスに移動するか、二重右矢印をクリックしてすべての使用可能日カレンダーを選択します。
6. 「次へ」ボタンをクリックして続行します。
7. 「**使用可能日カレンダーを除外**」ページで、[Ctrl] キーを押しながら「**使用可能日カレンダー**」リスト・ボックスの Maintenance および Christmas の各カレンダーをクリックします。
8. 右矢印をクリックして Maintenance および Christmas の各カレンダーを「**除外された使用可能日カレンダー**」リスト・ボックスに移動します。これにより、それぞれのカレンダーに指定された日付と時刻に処理を実行できなくなります。
9. 「次へ」ボタンをクリックして続行します。
10. 「**結合された使用可能日カレンダーのサマリー**」ページで、「**カレンダーを表示**」をクリックして使用可能日カレンダーを表示します。緑色の部分は使用可能時間を示します。見終わったら、カレンダーをクローズします。

この時点で結合カレンダーをチェックすることをお勧めします。処理を禁止したカレンダーの指定期間が除外されていることを確認してください。12 月までスクロールして、12 月 25 日が処理可能日から除外されていることを確認します。「日」オプションを選択し、土曜日までスクロールして、午後 3 時から処理を実行できなくなっていることを確認します。

11. 「完了」ボタンをクリックします。
12. 「コンポーネントの管理」ページで、「クローズ」ボタンをクリックします。

シンプルなカレンダーおよび結合カレンダーの両方が正常に作成されました。このカレンダーは、RDF、Oracle Reports Services サーバーおよびプリンタへのアクセスをさらに制限するために使用できます。RDF、Oracle Reports Services サーバーおよびプリンタの制限の詳細は、[第 5 章「Oracle Portal を使用した Oracle Reports Services セキュリティ」](#)を参照してください。

RWCLI60 コマンドライン引数

この付録には RWCLI60 コマンドライン引数の説明が含まれています。RWCLI60 はコマンドラインを解析して、指定された Oracle Reports Services (RWMTS60) に転送します。コマンドラインの使用方法は RWRUN60 と非常に似ています。

B.1 構文

RWCLI60 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引数です。

```
RWCLI60 MODULE|REPORT=runfile USERID=userid  
[ [keyword=]value| (value1, value2, ...) ] SERVER=tnsname
```

B.2 使用上の注意

RWCLI60 コマンドラインの使用上の注意は、次のとおりです。

- クライアントのコマンドラインで指定されたすべてのファイル名とパスは、サーバー・マシン上のファイルとディレクトリを参照します。ただし、コマンド・ファイルは参照しません。
- コマンドラインが CMDFILE= を含んでいる場合、コマンド・ファイルは、Oracle Reports Services へ送出される前に元のコマンドラインに読み込まれて追加されます。コマンド・ファイルは、ランタイム・エンジンによって再度読み込まれることはありません。

MODULE|REPORT

説明 MODULE|REPORT は実行するレポートの名前です。(REPORT は旧バージョンと互換性があります。)

構文 [MODULE|REPORT=] runfile

値 任意の有効な実行ファイル（つまり、拡張子が RDF、REP または XML のファイル）。ファイル拡張子を入力しないと、Oracle Reports Services Runtime は REP 拡張子、RDF 拡張子、XML 拡張子、拡張子なしの順にファイルを検索します。Oracle Reports Services Runtime はファイル・パス検索順でファイルを検索します。

USERID

説明 USERID は ORACLE ユーザー名またはブレース・ホルダのユーザー名（つまり、\$username）とパスワードです。任意のデータベース名、リモート・データベースヘアクセスするための Net8 通信プロトコルまたは ODBC データソース名（Oracle 以外のデータソースヘアクセスする場合）を伴います。パスワードを省略すると、データベース・ログオン・フォームが表示されます。

ユーザーがデータベースにログオンできるようにするには、レポート・リクエストから USERID コマンドライン引数を省いてください。ユーザーがレポート・リクエストを実行するたびにログオンできるようにするには、CGI コマンド SHOWAUTH と AUTHTYPE=S をレポートの URL で使用するか、%D 引数を cgicmd.dat（CGI）ファイルのキー・マッピング・エントリに含めます。

値 ログオン定義は次のいずれかのフォームにする必要があり、長さは 512 バイトを超えないようにしてください。

```
username [/password]
username [/password] [@database]
[user [/password]]@ODBC:datasource[:database] or [user [/password]]@ODBC:*

<$username> [/password]
<$username> [/password] [@database]
```

有効な接続文字列のリストは、[付録 E「データベース接続文字列」](#)を参照してください。

PARAMFORM

説明 PARAMFORM を指定する場合は NO にする必要があります。

構文 [PARAMFORM=] NO

CMDFILE

説明 CMDFILE は、RWRUN60 コマンドの引数を含むファイルです。このオプションにより、RWRUN60 を起動するたびにいくつもの引数を指定しなくてもレポートを実行できます。

構文 [CMDFILE=] cmdfile

値 任意の有効なコマンド・ファイル。

制限事項 次の制限事項が適用されます。

- コマンド・ファイルは別のコマンド・ファイルを参照することがあります。
- RWRUN60 引数のコマンド・ファイル構文は、コマンドラインで使用される構文と同じものです。
- コマンドラインに入力する値は、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 RWRUN60 を指定するのに、コマンドラインで COPIES の値に 1、CMDFILE の値に RUNONE (コマンド・ファイル) を入力するとします。RUNONE では COPIES は 2 に設定されています。この場合、レポートのコピーは 1 つのみ生成されます。
- このキーワードの引数は、オペレーティング・システムに依存する可能性があります。

TERM

説明 TERM は、RWRUN60 を使用する端末の種類です。TERM はランタイム・パラメーター・フォームおよび Runtime プレビューアのみで有効です。このキーワードはキャラクタ・モードでのみ使用できます。

構文 [TERM=] termtype

値 任意の有効な端末の種類。

デフォルト インストレーションに依存。(互換性の定義については、Oracle Reports Services のシステム管理者に相談してください。)

使用上の注意 このキーワードの引数は、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。

ARRAYSIZE

説明 ARRAYSIZE は、ORACLE 配列処理で使用するサイズ (単位は KB) です。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 1 ~ 9,999 の数値。Oracle Reports Services Runtime ではレポートで問合せを行う際に、ここで指定した KB 数のメモリーを使用できることを意味します。

デフォルト デフォルトの配列サイズは 10K です。ORACLE 配列処理の詳細は、『Oracle8i Server 管理者ガイド』を参照してください。

DESTYPE

説明 DESTYPE は、レポート出力を受け取るデバイス・タイプです。

構文 [DESTYPE=] { CACHE | LOCALFILE | FILE | PRINTER | SYSOUT | MAIL }

値

CACHE	Oracle Reports Services のキャッシュへ出力を直接送出します。 DESTYPE=CACHE は、DISTRIBUTE キーワードと一緒にには使用できません。サーバーはコマンドラインに DISTRIBUTE を見つけると、DESTYPE=CACHE コマンドライン引数を無視します。
LOCALFILE	出力をクライアント・マシン上のファイルに送って、BACKGROUND 値にかかわらず同期コールを強制的に行います。
FILE	出力を DESNAME で指定されたサーバー・マシン上のファイルに送ります。
PRINTER	出力を DESNAME で指定されたサーバー・マシン上のプリンタに送ります。Oracle Reports Services サーバー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。
MAIL	出力を DESNAME で指定したメール・ユーザーに送ります。任意の MAPI 互換のメール・システムまたはこのサービス・プロバイダのドライバがインストールされているメール・システムにメールを送信できます。レポートは添付ファイルとして送信されます。
SYSOUT	出力をクライアント・マシンのデフォルトの出力デバイスに送って、同期コールを強制的に行います。

デフォルト DESTYPE パラメータの初期値プロパティから受け取ります。

使用上の注意 RWCLI60 では、DESTYPE に対して Screen と Preview を使用することはできません。

DESNAME

説明 DESNAME は、レポート出力の送信先のファイルあるいはプリンタ名、または電子メール ID（または配布リスト）です。レポート出力を電子メールで送信するためには、電子メール ID を通常の電子メール・アプリケーション（Windows 上の MAPI 互換アプリケーションまたは UNIX 上のネイティブのメール・アプリケーション）の場合と同様にして指定します。複数のユーザー名を指定するときは、ユーザー名を括弧に入れ、それぞれをカンマで区切ります（例：（name, name, ...name））。

構文 [DESNAME=] desname

値 1KB を超えない長さの任意の有効なファイル名、プリンタ名または電子メール ID。プリンタ名の場合は、任意でポートを指定することもできます。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受け取ります。
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 使用上の注意は、次のとおりです。

- DESTYPE が SCREEN である場合はこのキーワードは無視されます。
- DESTYPE が PREVIEW の場合、Oracle Reports Services Builder は DESNAME を使用して、どのプリンタのフォントで出力を表示するか決定します。
- このキーワードの引数は、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。

場合によっては、このパラメータは使用中のオペレーティング・システムによって上書きされることがあります。

DESFORMAT

説明 ビットマップ環境では、DESFORMAT は DESTYPE が FILE であるときに使用されるプリンタ・ドライバを指定します。キャラクタ・モード環境では、DESNAME で指定されたプリンタの特徴を指定します。

構文 [DESFORMAT=] desformat

値 1KB を超えない長さの任意の有効な宛先フォーマットを指定します。このキーワードの有効な値の例としては、hpl、hplwide、dec、decwide、decland、dec180、dflt、wide などがあります。有効な宛先フォーマットのリストについては、システム管理者に問い合せてください。

PDF	レポート出力が PDF ビューアで読み込むことのできるファイルに送信されることを意味します。PDF 出力は、システム上で現在設定されているプリンタをベースにしています。現在選択されているプリンタを使用して出力が生成されるので、レポートを実行しているマシンに対してプリンタを設定する必要があります。
HTML	レポート出力が HTML 3.0 互換ブラウザ（たとえば Netscape 2.2）で読み込むことのできるファイルに送信されることを意味します。
HTMLCSS	ファイルに送信されたレポート出力には、HTML 3.0 互換ブラウザで読み込むことのできる、カスケード式のスタイル・シートをサポートするスタイル・シートの拡張機能が含まれることを意味します。

HTMLCSSIE	ファイルに送信されるレポート出力には、Microsoft Internet Explorer 3.x で読み込むことのできるスタイル・シートの拡張機能が含まれていることを意味します。
RTF	レポート出力が標準のワード・プロセッサ（Microsoft Word など）で読み込むことのできるファイルに送信されることを意味します。Microsoft Word でファイルをオープンした場合、レポートにあるすべてのグラフィックスおよびオブジェクトを表示させるには、「表示」「ページレイアウト」を選択する必要があります。
DELIMITED	レポート出力が標準のスプレッドシート・ユーティリティ（Microsoft Excel など）で読み込むことのできるファイルに送信されることを意味します。デリミタを選んでいない場合、デフォルトのデリミタとして TAB が使用されます。
XML	レポートが XML 文書として出力され、拡張子 XML の付いた別ファイルとして保存されることを意味します。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込んだりできます。

デフォルト DESFORMAT パラメータの初期値プロパティから取り込みます。ビットマップ環境の Oracle Reports Services Builder では、DESFORMAT が空白または `df1t` の場合、現行のプリンタ・ドライバ（「**ファイル**」「**ページ設定（プリンタ選択）**」）が使用されます。「**ページ設定（プリンタ選択）**」で何も選択されていないときは、デフォルトの PostScript が使用されます。

使用上の注意 使用上の注意は、次のとおりです。

- DESTYPE が SCREEN である場合はこのキーワードは無視されます。
- このキーワードの値は、オペレーティング・システムによっては、大文字と小文字の区別が必要な場合があります。

CACHELOB

説明 CACHELOB は、検索された Oracle8 のラージ・オブジェクトを（REPORTS60_TMP で指定される）テンポラリ・ファイルのディレクトリにキャッシュするかどうかを指定します。

値 YES は、LOB をテンポラリ・ファイル・ディレクトリにキャッシュします。NO は、LOB をテンポラリ・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 使用上の注意は、次のとおりです。

- このオプションはコマンドラインでのみ設定できます。
- テンポラリ・ファイル・ディレクトリのロケーション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薦めします。ただし、値を NO に設定すると、LOB をサーバーから複数回フェッチする必要があるため、パフォーマンスが低下する可能性があります。

COPIES

説明 COPIES はプリンタに出力されるレポートの部数です。

構文 [COPIES=]n

値 1 から 9,999 までの任意の有効な整数。

デフォルト COPIES パラメータの初期値プロパティから取り込みます。

使用上の注意 使用上の注意は、次のとおりです。

- DESTYPE が Printer 以外の場合は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。

CURRENCY

説明 CURRENCY は数字フォーマットで使用される通貨記号です。

構文 [CURRENCY=]currency_symbol

値 1KB を超えない長さの任意の有効な英数字文字列です。

デフォルト ORACLE の場合のデフォルトは ORACLE の各国語サポート機能で決定されます。CURRENCY パラメータの初期値プロパティで最高 4 文字までのデフォルトを設定できます。

使用上の注意 コマンドラインで入力された CURRENCY 値は、Property Palette に入力された CURRENCY 値によって上書きされます。

THOUSANDS

説明 THOUSANDS は数字フォーマットで使用される 3 桁区切り文字です。

構文 [THOUSANDS=]thousands_symbol

値 任意の有効な英数字

デフォルト ORACLE の場合のデフォルトは ORACLE の各国語サポート機能で決定されます。THOUSANDS パラメータの初期値プロパティで最高 4 文字までのデフォルトを設定できます。

使用上の注意 使用上の注意は、次のとおりです。

- コマンドラインに入力された THOUSANDS 値は、Parameter プロパティ・シートに入力された THOUSANDS 値によって上書きされます。
- THOUSANDS 値として定義された英数字が実際に返される値です。たとえば、THOUSANDS 値として "," を定義すると、"," が返されます。

DECIMAL

説明 DECIMAL は数字フォーマットで使用される小数点記号です。

構文 [DECIMAL=]decimal_symbol

値 任意の有効な英数字

デフォルト ORACLE の場合のデフォルトは ORACLE の各国語サポート機能で決定されます。また、DECIMAL パラメータの初期値プロパティでデフォルトを設定することもできます。

使用上の注意 使用上の注意は、次のとおりです。

- コマンドラインに入力された DECIMAL 値は、Parameter プロパティ・シートに入力された DECIMAL 値によって上書きされます。
- DECIMAL 値として定義された英数字が実際に返される値です。たとえば、DECIMAL 値として "." を定義すると、"." が返されます。

READONLY

説明 READONLY は、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します。ORACLE からデータにアクセスすると、読取り一貫性は SET TRANSACTION READ ONLY 文によって遂行されます (SET TRANSACTION READ ONLY 文の詳細は『Oracle8i Server SQL リファレンス』を参照してください)。

構文 [READONLY=] {YES|NO}

値 YES は読取り一貫性を要求します。NO は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 1 回のみ問合せレポートに対して、ORACLE によってロックを行わずに自動的に読取り一貫性機能が提供されるため、このキーワードが有効なのは複数回の問合せが行われているレポートに対してのみです。

制限事項 Report トリガーの実行順において、SET TRANSACTION READ ONLY がどこで発生するか注意してください。

LOGFILE

説明 LOGFILE は、「ファイル」「ページ設定 (プリンタ選択)」からの出力の送信先のファイルの名前です。指定されたファイルがすでに存在する場合は、出力はそのファイルに追加されます。このキーワードはキャラクタ・モードでのみ使用できます。

構文 [LOGFILE=] logfile

値 任意の有効なファイル名。

デフォルト 現行ディレクトリの dflltrep.log

BUFFERS

説明 BUFFERS は KB 数で表された仮想メモリー・キャッシュのサイズです。この設定値を調整して、メモリー領域がレポートを実行するためには十分で、しかも、システム・リソースを使用しすぎない程度であることを確実にしてください。

構文 [BUFFERS=] n

値 1 から 9,999 までの数。オペレーティング・システムによっては、上限がより低い場合があります。

デフォルト 640KB

使用上の注意 この設定値をセッションの途中で変更した場合は、変更した設定値は次のレポート実行時まで有効ではありません。

BATCH

説明 BATCH を指定する場合は YES にする必要があります。

構文 [BATCH=] YES

PAGESIZE

説明 PAGESIZE は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）です。ページはレポートが入る大きさである必要があるので注意してください。たとえば、レポートのフレームがページ・ディメンションを超えるサイズになっている場合、レポートは実行されません。

構文 [PAGESIZE=]width x height

値 フォームに対する任意の有効なページ・ディメンション。ページ幅×ページ高さ（それぞれ 0 以上の数）。最大幅・高さは測定単位によって異なります。インチでは、最大幅・高さは 512 インチ。cm では 1312cm。パイカでは 36,864 パイカ。

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。 $(8.5 * 80) / 80 \times (11 * 20) / 66 = 8.5 \times 3.33$

使用上の注意 使用上の注意は、次のとおりです。

- 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。Oracle Reports Services Builder で、使用するプリンタが許容するプリント可能面積を超えるページ幅とページ高さを定義すると、レポートを出力した際にクリッピングが生じる可能性があります。クリッピングを避けるには、プリンタのプリント可能面積を大きくするか（オペレーティング・システムが対応できる場合）、ページ幅×ページ高さをそのページのプリント可能面積に設定するかします。
- この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。

PROFILE

説明 PROFILE はレポートの実行に関するパフォーマンス統計を格納するファイルの名前です。ファイル名を指定すると、レポートの実行に消費された経過時間と CPU 時間に関する統計が Oracle Reports Services Builder によって計算されます。PROFILE は次の統計を計算します。

- TOTAL ELAPSED TIME は RWBLD60 を発行してからデザイナを終了するまでに経過した時間です。TOTAL ELAPSED TIME は Oracle Reports Services Builder Time と ORACLE Time の合計です。
- Oracle Reports Time は Oracle Reports Services Builder で消費された時間です。
- ORACLE Time はデータベースで消費された時間で、次の要素から構成されています。
 - UPI はデータベースへの接続や SQL の解釈、データのフェッチなどの処理に消費された時間です。
 - SQL は SRW.DO_SQL の実行に消費された時間です。
- プロセスが使用した TOTAL CPU Time は、Builder で消費された CPU 時間です。

注意： データベースは Oracle Reports Services Builder のプロセスに含まれているので、オペレーティング・システムによっては、Oracle Reports Services Builder 時間にデータベース時間が含まれる場合もあります。

構文 [PROFILE=] profiler_file

値 現行ディレクトリにある任意の有効なファイル名。

RUNDEBUG

説明 RUNDEBUG は、レポート中の論理エラーに対して追加のランタイム・チェックを実行させるかどうかを指定します。RUNDEBUG はエラーではないが望ましくない出力となる可能性のある事項をチェックします。RUNDEBUG がチェックするのは次の事項です。

- 重なっているが他のオブジェクトを包括はしない枠および繰り返し現れる枠。これは、出力の際にあるオブジェクトによって他のオブジェクトが上書きされる原因となる可能性があります。
- サイズが固定されていないページ依存リファレンスを持つレイアウト・オブジェクト。Oracle Reports Services Builder はこのようなオブジェクトのサイズを、「垂直拡張度」プロパティあるいは「水平拡張度」プロパティにかかわらず固定します。
- PL/SQL で誤った頻度で参照されているバインド変数。

構文 [RUNDEBUG=] {YES|NO}

値 YES は追加のランタイム・エラー・チェックを実行します。NO は追加のランタイム・エラー・チェックを実行しません。

デフォルト YES

ONSUCCESS

説明 ONSUCCESS は、レポートの実行が終了した時点で COMMIT または ROLLBACK を実行させるかどうかを指定します。

構文 [ONSUCCESS=] { COMMIT | ROLLBACK | NOACTION }

値 COMMIT は、レポート完了時に COMMIT を実行します。ROLLBACK は、レポート完了時に ROLLBACK を実行します。NOACTION は、レポート完了時にはどちらも行いません。

デフォルト USERID が入力されているときは COMMIT。外部ソース（たとえば Oracle Forms Services）から USERID の入力なしにコールされたときは NOACTION。

使用上の注意 ONSUCCESS に対する COMMIT または ROLLBACK は、レポートの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 コマンドの項を参照してください。

ONFAILURE

説明 ONFAILURE は、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します。

構文 [ONFAILURE=] { COMMIT | ROLLBACK | NOACTION }

値 COMMIT は、レポートが完了しなかった場合に COMMIT を実行します。ROLLBACK は、レポートが完了しなかった場合に ROLLBACK を実行します。NOACTION は、レポートが完了しなかった場合、動作を行いません。

デフォルト USERID が入力されているときは ROLLBACK。外部ソース（たとえば Oracle Forms Services）から USERID の入力なしにコールされたときは NOACTION。

使用上の注意 ONFAILURE に対する COMMIT または ROLLBACK は、レポートの完了に失敗し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 コマンドの項を参照してください。

KEYIN

説明 KEYIN は実行時に実行するキーストローク・ファイルの名前です。KEYIN は、KEYOUT で生成されたキーストローク・ファイルを実行するのに使用されます。KEYIN はキーストローク・ファイルを実行するので、キャラクタ・モード環境でレポートを実行しているときのみ使用されます。

構文 [KEYIN=]keyin_file

値 現行ディレクトリにある任意の有効なキー・ファイル名。

KEYOUT

説明 KEYOUT は、使用したすべてのキーストロークを Oracle Reports Services Runtime で記録するときに使用するキーストローク・ファイルの名前です。この後、KEYIN を使用してキーストローク・ファイルを実行できます。KEYOUT と KEYIN は、レポート実行時に必ず使用する一定のキーストロークがある場合に便利です。また、デバッグにも役立ちます。KEYOUT はキーストローク・ファイルの作成に使用されるため、キャラクタ・モード環境でレポートを実行しているときのみ使用されます。

構文 [KEYOUT=]keyout_file

値 任意の有効なファイル名。

ERRFILE

説明 ERRFILE は、Oracle Reports Services Builder でエラー・メッセージを格納するために使用するファイルの名前です。

構文 [ERRFILE=]error_file

値 任意の有効なファイル名。

LONGCHUNK

説明 LONGCHUNK は、Oracle Reports Services Builder で LONG の列値を検索する際の増分の大きさ（単位は 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 Oracle7 および Oracle8 のみに適用されます。

構文 [LONGCHUNK=]n

値 1 から 9,999 までの数。オペレーティング・システムによっては、上限がより低い場合があります。

デフォルト 10KB

ORIENTATION

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 は、プリント方向は現行のプリンタ設定どおりとします。LANDSCAPE は横方向です。PORTRAIT は縦方向です。

デフォルト DEFAULT

使用上の注意 使用上の注意は、次のとおりです。

- 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必要があります。
- Motif 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。

BACKGROUND

説明 BACKGROUND は、コールが同期的 (BACKGROUND=NO) か非同期的 (BACKGROUND=YES) かを指定します。同期的コールは、レポートの待機、ランタイム・エンジンへの割当て、実行、終了までクライアントが待機することを意味します。非同期的コールは、クライアントは単にコールを送るのみで、その完了まで待機しないことを意味します。クライアント・プロセスが同期的コールの最中に中止された場合、そのジョブは取り消されます。

構文 [BACKGROUND=] {YES|NO}

値 YES または NO

デフォルト NO

MODE

説明 MODE は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します。この指定によって、ビットマップ環境の Oracle Reports Services Builder からキャラクタ・モードのレポートを実行したり、逆の操作を行うことができます。たとえば、レポートを端末 (たとえば vt220 など) から PostScript プリンタに送信するときは、キャラクタ・モードの RWRUN60 を起動して、レポートを MODE=BITMAP で実行できます。Windows では、MODE=CHARACTER と指定すると、Oracle Reports Services Builder の ASCII ドライバを使用して編集可能な ASCII 出力が生成されます。

構文 [MODE=] {BITMAP|CHARACTER|DEFAULT}

値 次の値が適用されます。

- BITMAP
- DEFAULT は、使用中の現行実行ファイルのモードでレポートを実行することを意味します。
- CHARACTER

デフォルト DEFAULT

PRINTJOB

説明 PRINTJOB は、レポートを実行する前に「印刷ジョブ」ダイアログ・ボックスを表示させるかどうかを指定します。

構文 [PRINTJOB=] {YES|NO}

値 YES または NO

デフォルト NO

使用上の注意 使用上の注意は、次のとおりです。

- レポートが派生プロセスで実行されるときは (RWRUN60 などの実行ファイルが、RWBLD60 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、「印刷ジョブ」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、「印刷ジョブ」ダイアログ・ボックスは表示されません。

TRACEFILE

説明 TRACEFILE は、Oracle Reports Services Builder でトレース情報が記録されるファイルの名前です。

構文 [TRACEFILE=] tracefile

値 任意の有効なファイル名。

使用上の注意 使用上の注意は、次のとおりです。

- トレース情報は RDF ファイルを実行しているときしか生成できません。REP ファイルの実行中はログを指定できません。
- TRACEFILE の他に LOGFILE あるいは ERRFILE も指定すると、トレース情報はすべて一番後で指定されたファイルに格納されます。たとえば、次の場合は、指定されたトレース情報はすべて err.log に格納されます。それが RWRUN60 コマンドで最後に指定されたファイルであるためです。

```
RWRUN60 MODULE=order_entry
USERID=scott/tiger
TRACEFILE=trace.log LOGFILE=mylog.log
ERRFILE=err.log
```

TRACEMODE

説明 TRACEMODE は、Oracle Reports Services Builder でトレース情報をファイルに追加するか、または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 トレース情報は RDF ファイルを実行しているときしか生成できません。REP ファイルの実行中はログを指定できません。

TRACEOPTS

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_ERR|TRACE_PRF|TRACE_APP|TRACE_PLS|TRACE_SQL|TRACE_TMS|TRACE_DST|TRACE_ALL|(opt1, opt2, ...)}

値 次の値が適用されます。

- 括弧に入ったオプションのリストは、括弧内のすべてのオプションを使用することを指定します。たとえば、TRACE_OPTS=(TRACE_APP,TRACE_PRF) は、TRACE_APP と TRACE_PRF を適用することを意味します。
- TRACE_ALL は、トレース情報であると思われるものをすべてトレース・ファイルに記録することを意味します。
- TRACE_APP は、すべてのレポート・オブジェクトにあるトレース情報をトレース・ファイルに記録することを意味します。
- TRACE_BRK は、トレース・ファイル内のブレークポイントをリストすることを意味します。
- TRACE_DST は、トレース・ファイル内の配布リストをリストすることを意味します。この情報を使用して、どのセクションがどの宛先に送信されたのか、判断できます。トレース・ファイルのフォーマットは DST ファイルのフォーマットとよく似ているので、カット・アンド・ペーストによってトレース・ファイルから DST ファイルを生成できます。
- TRACE_ERR は、トレース・ファイルにあるエラー・メッセージと警告をリストすることを意味します。

- TRACE_PLS は、すべての PL/SQL オブジェクトにあるトレース情報をトレース・ファイルに記録することを意味します。
- TRACE_PRF は、トレース・ファイルにあるパフォーマンス統計を記録することを意味します。
- TRACE_SQL は、すべての SQL にあるトレース情報をトレース・ファイルに記録することを意味します。
- TRACE_TMS は、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することを意味します。

デフォルト TRACE_ALL

使用上の注意 トレース情報は RDF ファイルを実行しているときしか生成できません。REP ファイルの実行中はログを指定できません。

AUTOCOMMIT

説明 データベースの変更（CREATE など）をデータベースに自動的にコミットするかどうかを指定します。ある種の非 ORACLE データベース（SQL Server など）では、AUTOCOMMIT=YES とする必要があります。

構文 [AUTOCOMMIT=] {YES|NO}

値 YES または NO

デフォルト NO

NONBLOCKSQL

説明 NONBLOCKSQL は、Oracle Reports Services Runtime がデータベースからデータをフェッチしているときに他のプログラムの実行を認めるかどうか指定します。

構文 [NONBLOCKSQL=] {YES|NO}

値 YES は、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 は、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

ROLE

説明 ROLE は、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します。RWBLD60 については ROLE は無視されます。

構文 [ROLE=] {rolename/[rolepassword]}

値 有効なロールと（任意で）ロール・パスワード

DISABLEPRINT

説明 DISABLEPRINT は、Runtime プレビューアの「ファイル」「印刷」、または「ファイル」「ページ設定（プリンタ選択）」（Motif 上の）およびそれに対応するツールバー・ボタンを使用不可能にするかどうか指定します。

構文 [DISABLEPRINT=] {YES|NO}

値 YES または NO

デフォルト NO（レポート出力に印刷したくない空白ページがある場合）

DISABLEMAIL

説明 DISABLEMAIL は、Runtime プレビューアの「ファイル」「メール」メニューとそれに対応するツールバー・ボタンを使用不可能にするかどうか指定します。

構文 [DISABLEMAIL=] {YES|NO}

値 YES または NO

デフォルト NO

DISABLEFILE

説明 DISABLEFILE は、Runtime プレビューアの「ファイル」「ファイル出力」メニューを使用不可能にするかどうか指定します。

構文

[DISABLEFILE=] {YES|NO}

値 YES または NO

デフォルト NO

DISABLENEW

説明 DISABLENEW は、「表示」「新規プレビュー」メニューを使用不可能にして、Runtime プレビューの新しいインスタンスを表示する機能を使用できないようにするかどうかを指定します。

構文 [DISABLENEW=] {YES|NO}

値 YES または NO

デフォルト NO

DESTINATION

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 DST ファイルの名前を指定できます。

構文 [DESTINATION=] filename.DST

値 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 DST ファイルの名前

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。

DISTRIBUTE

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あるいは DST 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可能または不可能にできます。

構文 [DISTRIBUTE=] {YES|NO}

値 YES は、レポートを配布リストにある宛先に配布します。

NO は、配布リストを無視して、DESNAME および DESFORMAT の指定どおりにレポートを出力します。これは基本的には、配布目的としてセットアップされているレポートを実際には配布することなく実行できるようにするデバッグ・モードです。

デフォルト NO

使用上の注意 DESTINATION キーワードを使用可能にするためには、DISTRIBUTE=YES と指定する必要があります。

PAGESTREAM

説明 PAGESTREAM は、レポートが HTML あるいは HTMLCSS 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・ストリームを可能または不可能にします。

- Report 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 [PAGESTREAM=] {YES|NO}

値 YES はページ・ストリームを行います。NO はページ・ストリームを行わずにレポートを出力します。

デフォルト NO

BLANKPAGES

説明 BLANKPAGES は、レポートの印刷時に空白ページを抑止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 はすべての空白ページを印刷します。NO は空白ページは印刷しません。

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、空白の物理ページの印刷はすべて抑止するときに特に便利です。

SERVER

説明 SERVER は Oracle Reports Services の TNS サービス・エントリ名です。

構文 [SERVER=] tnsname

値 任意の TNS サービス・エントリ名

使用上の注意 Web サーバー・マシン上で REPORTS60_REPORTS_SERVER 環境変数を設定すると、デフォルト・サーバーを使用してリクエストを処理するための SERVER コマンドライン引数を省略できます。あるいは、SERVER コマンドライン引数を指定してデフォルト値を上書きすることもできます。

JOBNAME

説明 JOBNAME は、Oracle Reports Services Queue Manager に表示されるジョブ名です。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。このキーワードが指定されていないときは、キュー・マネージャ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

SCHEDULE

説明 SCHEDULE はスケジューリング・コマンドです。デフォルトは「ただちに開始」です。スケジューリング・コマンドの引用を不要にするためには、スペースではなくアンダースコア (_) を使用します。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
schedule=last_weekday_before_15_from 15:53_Oct_23,_1999_retry_after_1_hour

注意： 従来のフォームの SCHEDULE 構文はサポートされていますが、ここでは現行の SCHEDULE 構文についてのみ説明します。

構文 現行の構文は、次のとおりです。

[SCHEDULE=] string

ここで、string は次のとおりです。

[FREQ from] TIME [retry {n} + after LEN]

FREQ	hourly daily weekly monthly {every LEN DAYREPEAT}} {last {WEEKDAYS weekday weekend} before {n}+}
LEN	{n}+ {minute[s] hour[s] day[s] week[s] month[s]}
DAYREPEAT	{first second third fourth fifth} WEEKDAYS of month
WEEKDAYS	mon tue wed thu fri sat sun
TIME	now CLOCK [DATE]
CLOCK	h:m h:mm hh:m hh:mm
DATE	today tomorrow {MONTHS {d dd} [,year]}
MONTHS	jan feb mar apr may jun jul aug sep oct nov dec

TOLERANCE

説明 TOLERANCE は、重複ジョブの検出に対する許容時間を分で表したものです。TOLERANCE は、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を決定します。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 [4.2 項「重複ジョブの検出」](#) を参照してください。

構文 [TOLERANCE=] number

値 0 から始まる任意の時間（単位は分）

使用上の注意 使用上の注意は、次のとおりです。

- 許容時間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、Oracle Reports Services はそのレポートを再実行します。
- 同一ジョブが提出されたときにレポートが処理中（つまり、現行のジョブ・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、Oracle Reports Services は現在実行中のジョブの出力を再利用します。

DELIMITER

説明 DELIMITER は、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 次のような、任意の英数字または英数字の文字列です。

,	各セルがカンマで区切られることを意味します。
.	各セルがピリオドで区切られることを意味します。

また、次の 4 つの予約値のいずれも使用できます。

tab	各セルがタブで区切られることを意味します。
space	各セルがスペースで区切られることを意味します。
return	各セルが改行で区切られることを示します。
none	デリミタを使用しないことを意味します。

また、次のような、ASCII のキャラクタ・セットに基づいたエスケープ・シーケンスも使用できます。

¥t 各セルがタブで区切られることを意味します。

¥n 各セルが改行で区切られることを示します。

デフォルト Tab

使用上の注意 この引数を使用できるのは、DESFORMAT=DELIMITED と指定しているときのみです。

CELLWRAPPER

説明 CELLWRAPPER は、レポート出力において区切られたセルの周囲に表示される文字を指定します。

構文 [CELLWRAPPER=]value

値 次のような、任意の英数字または英数字の文字列です。

" 二重引用符がセルの両側に表示されることを意味します。

' 一重引用符がセルの両側に表示されることを意味します。

また、次の 4 つの予約値のいずれも使用できます。

tab セルの両側にタブが表示されることを意味します。

space セルの両側にスペースが 1 個表示されることを意味します。

return セルの両側に改行が表示されることを意味します。

none セル・ラッパーが使用されないことを意味します。

また、次のような、ASCII のキャラクタ・セットに基づいたエスケープ・シーケンスも使用できます。

¥t セルの両側にタブが表示されることを意味します。

¥n セルの両側に改行が表示されることを意味します。

デフォルト none

使用上の注意 使用上の注意は、次のとおりです。

- この引数を使用できるのは、DESFORMAT=DELIMITED と指定しているときのみです。
- セル・ラッパーは実際のデリミタとは異なります。

DATEFORMATMASK

説明 DATEFORMATMASK は、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 任意の日付書式マスク

使用上の注意 この引数を使用できるのは、DESFORMAT=DELIMITED と指定しているときのみです。

NUMBERFORMATMASK

説明 NUMBERFORMATMASK は、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

使用上の注意 この引数を使用できるのは、DESFORMAT=DELIMITED と指定しているときのみです。

EXPRESS_SERVER

説明 EXPRESS_SERVER は接続先の OLAP Server を指定します。

構文 EXPRESS_SERVER="server=[server]/domain=[domain]/user=[userid]/password=[passwd]"

RAM 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 (") に囲まれた有効な接続文字列。ここで、

server	OLAP Server 文字列 (たとえば <code>ncacn_ip_tcp:olap2-pc/s1=x/st=x/ct=x/sv=x/</code>)。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 にログオンするためのユーザー ID。
password	そのユーザー 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、 <code>¥¥ServerName¥ShareName¥</code> の後に任意の下位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「設定の編集」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。
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。4 つのパラメータがすべて必要ですが、指定する順序は任意です。次の表は、各パラメータとそれぞれの設定値をまとめたものです。

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 Connect セキュリティ 2: ホスト（ドメイン・ログイン）と Call セキュリティ 3: ホスト（ドメイン・ログイン）と Packet セキュリティ 4: ホスト（ドメイン・ログイン）と Integrity セキュリティ 5: ホスト（ドメイン・ログイン）と Privacy セキュリティ 注意: 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

使用上の注意 使用上の注意は、次のとおりです。

- 文字列の全体が引用の内部に含まれるので、必要に応じて（たとえば、ユーザー ID が John Smith だった場合など）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。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。

AUTHID

説明 AUTHID は、制限付きの Oracle Reports Services サーバーのユーザーを認証するのに使用されるユーザー名とパスワードです。ユーザー認証によって、レポート・リクエストを行っているユーザーが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。ユーザーがログオンに成功すると、Oracle Portal に登録されている、保護されている Oracle Reports Services サーバーに対してユーザーを認証する暗号化された cookie が送信されます。デフォルトでは、cookie は 30 秒で無効になります。cookie が期限切れになると、その後のリクエスト（保護されている Oracle Reports Services サーバーに送られるリクエスト）は、再度認証される必要があります。

REPORTS60_COOKIE_EXPIRE 環境変数を使用して、認証 cookie の有効期限を変更できます。詳細は、[付録 D「環境変数」](#)を参照してください。

ユーザーが cookie の期限が切れるまでは認証され続けるようにする場合は、レポート・リクエストから AUTHID のコマンドライン引数を省略します。ユーザーが、レポート・リクエストを実行するたびに認証されるようにする場合は、CGI コマンド SHOWAUTH と AUTHTYPE=S をレポートの URL で使用します。あるいは、%S 引数を cgicmd.dat（CGI）ファイルのキー・マッピング・エントリに含めます。

構文 [AUTHID=]username/password

値 Oracle Portal で生成した任意の有効なユーザー名とパスワード。Oracle 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相談してください。

CUSTOMIZE

説明 CUSTOMIZE は、レポート実行時に適用させる XML ファイルを指定します。XML ファイルには、レポート定義をなんらかの形で変更する、カスタマイズされた設定（フォントや色の変更など）が入っています。

構文 [CUSTOMIZE=]filename.xml | (filename1.xml, filename2.xml, ...)

値 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。必要に応じて、ファイル名の前にパス情報を付加します。

SAVE_RDF

説明 SAVE_RDF は、RDF ファイルと XML カスタマイズ・ファイルの複合ファイルの保存先のファイルを指定します。この引数が真価を発揮するのは、XML ファイルを CUSTOMIZE キーワードを使用して適用させたい RDF ファイルがあって、その 2 つのファイルを複合したものを新しい RDF ファイルに保存する場合です。

構文 [SAVE_RDF=] filename.rdf

値 任意の有効なファイル名。

Oracle Reports Services 構成パラメータ

この付録には、Oracle Reports Services の構成パラメータの包括的なリストが収録されています。

パラメータ	説明
CACHEDIR	<p>CACHEDIR は Oracle Reports Services のキャッシュです。CACHEDIR はマシン上の任意のディレクトリあるいは論理ドライブに設定できます。指定されていないときのデフォルトは ORACLE_HOME¥REPORT60¥SERVER¥CACHE です。次に例を示します。</p> <p>CACHEDIR="C:¥ORACLE_HOME¥Report60¥cache"</p>
CACHESIZE	<p>CACHESIZE は MB で表したキャッシュ・サイズです。数多くのレポートの出力を Oracle Reports Services のキャッシュに格納するときは、この設定値を大きくする必要がある場合があります。キャッシュにはそれほど多くの出力を格納する予定がなく、システム・リソースに余裕がないときは、この設定値を小さくする必要がある場合もあります。キャッシュがこの設定サイズを超えると、Oracle Reports Services はキャッシュされたファイルを先入れ先出し方式でクリーン・アップします。デフォルト値は 50 です。</p> <p>注意: このパラメータは Queue Manager から設定できます。Queue Manager を開いて管理者としてログインします。「キュー」「プロパティ」を選択してから、「CACHESIZE (MB)」の設定値を変更します。</p>

パラメータ	説明
CLUSTERCONFIG	<p>CLUSTERCONFIG はマスター・サーバーに対するスレーブ・サーバーの構成です。クラスタ化によって、レポートを複数の Oracle Reports Services で実行することが可能になります。マスター・サーバーは、使用可能なスレーブ・サーバーを識別して、それらのエンジンを必要に応じて起動できます。マスター・サーバーに対して、多くのサーバーをスレーブとしてセットアップできます。マスター・サーバーの構成ファイルで次の構文を使用します。</p> <pre>Clusterconfig="(server=<servername> minengine=<minimum number of master engines> maxengine=<maximum number of master engines> initengine=<initial number of master engines> cachedir=<directory of central cache>)"</pre> <p>注意：各スレーブの定義は括弧で囲む必要があります。</p> <p>手順の詳細は、第 6 章「Oracle Reports Services サーバーのクラスタの構成」を参照してください。</p>
ENGLIFE	<p>ENGLIFE は、1 つのエンジンが停止するまでに実行するレポートの最大数です。その後 Oracle Reports Services によって、新しいリクエストに対して新たなエンジンが起動されます。デフォルト値は 50 です。</p>
FAILNOTEFILE	<p>FAILNOTEFILE は、実行に失敗したジョブに対して指定された電子メール・アドレスに送信された通知メッセージのテンプレートのパスとファイル名です。次に例を示します。</p> <pre>FAILNOTEFILE="C:\ORACLE_HOME\Report60\failnote.dat"</pre>
IDENTIFIER	<p>IDENTIFIER は暗号化されたキューの管理者 ID とパスワードの入った内部設定値ですこれは変更しないでください。IDENTIFIER が指定されていない場合、削除されている場合、あるいは構成ファイルが存在しない場合は、誰でも Oracle Reports Services Queue Manager からユーザー ID とパスワードを入力して、キュー管理者としてログインできます。このような方法でログインした者がいると、そのとき指定されたユーザー ID とパスワードが、Oracle Reports Services Queue Manager で変更されない限り、キュー管理者のユーザー ID とパスワードとなります。</p>
INITENGINE	<p>INITENGINE は、Oracle Reports Services によって起動されるランタイム・エンジンの初期数です。サーバー・プロセスは起動時に、この値に表されている数のエンジンを派生します。サーバ・プロセスはこれらのエンジンが接続するまで 2 分間待機し、接続されない場合は停止します。エンジンがこの時間内に接続できない場合は、通常、セットアップに問題があります。デフォルト値は 1 です。</p>

パラメータ	説明
LOGOPTION	<p>LOGOPTION はログ・ファイルに挿入させたいログ情報の種類です。オプションは alljob、failedjob、succeededjob です。次に例を示します。</p> <p>LOGOPTION="alljob"</p>
MAILPROFILE	<p>DESTYPE=MAIL の場合、Oracle Report Services によってメールが特定の宛先に送信されます。MAILPROFILE は、Oracle Report Services からレポートを送信するときに使用されるメール・プロファイルとパスワードです。次に例を示します。</p> <p>MAILPROFILE="mailprofileid/password"</p> <p>このパラメータは、Windows NT にのみ適用できます。Windows NT には独自の Windows メッセージ・システムが備わっており、MS Exchange はそのシステム（特に MAPI）を使用します。MAPI を機能させるには、MAPI が送信者の情報を認識できるように、MS Exchange 内に作成されたエントリに対応するプロファイル・エントリを提供する必要があります。</p> <p>Netscape 4.7 以降を使用している場合、MAILPROFILE パラメータを設定する必要はありません。そのかわり、Netscape の電話帳にすべての受信者のエントリを作成する必要があります。</p>
MAXCONNECT	<p>MAXCONNECT は一度にサーバー・プロセスに接続できるプロセスの最大数です。この設定値はエンジンとクライアントの合計で、2（エンジンとクライアントが最低 1 つずつ）を超える数とする必要があります。デフォルト値は 20 です。</p>
MAXENGINE	<p>MAXENGINE は Oracle Reports Services でレポートを実行するのに使用できるランタイム・エンジンの最大数です。サーバー・プロセスはアクティブなエンジンの数をこの値で示される数以下に抑えようとします。この数のエンジンに対応できるだけのメモリーとリソースを使用できるか確認してください。デフォルト値は 1 です。</p> <p>注意: このパラメータは Oracle Reports Services Queue Manager から設定できます。Oracle Reports Services Queue Manager を開いて管理者としてログインします。「キュー」「プロパティ」を選択してから、「同時実行エンジン: 最大」の設定を変更します。</p>
MAXIDLE	<p>MAXIDLE は、エンジンが停止する前にアイドル状態でいられる最長時間です。Oracle Reports Services は、エンジンを停止すると使用できるエンジンの数が MINENGINE で定義された数を下回ることになる場合は、停止を行いません。デフォルト値は 30 分です。</p> <p>注意: このパラメータは Oracle Reports Services Queue Manager から設定できます。Oracle Reports Services Queue Manager を開いて管理者としてログインします。「キュー」「プロパティ」を選択してから、「エンジン停止前の最大アイドル時間（分）」の設定値を変更します。</p>

パラメータ	説明
MINENGINE	<p>MINENGINE は、Oracle Reports Services がレポートの実行に使用するために持つべきランタイム・エンジンの最小数です。サーバー・プロセスはアクティブなエンジンの数をこの値で示される数以上に維持しようとします。この数のエンジンに対応できるだけのメモリーとリソースを使用できるか確認してください。デフォルト値は 0 です。</p> <p>注意： このパラメータは Oracle Reports Services Queue Manager から設定できます。Oracle Reports Services Queue Manager を開いて管理者としてログインします。「キュー」「プロパティ」を選択してから、「同時実行エンジン：最小」の設定を変更します。</p>
PERSISTFILE	<p>PERSISTFILE は Oracle Reports Services の DAT ファイルの保存場所を示します。このファイルにはスケジュール済のジョブの詳細が入っています。PERSISTFILE が指定されていないときのデフォルトは ORACLE_HOME¥REPORT60¥SERVER です。次に例を示します。</p> <p>PERSISTFILE="C:¥ORACLE_HOME¥Report60¥repserver.dat"</p>
REPOSITORYCONN	<p>REPOSITORYCONN はサーバーの起動時に Oracle Reports Services をデータベースに接続するデータベース接続文字列です。データベースは、ジョブが実行されているときは必ず Oracle Reports Services のキュー・アクティビティ（スケジュール済のジョブ）のスナップショットを入手します。</p> <p>データベースにキュー・アクティビティ表を作成するには、rw_server.sql スクリプトを実行する必要があります。次に例を示します。</p> <p>REPOSITORYCONN="repserver_schema/password@mydb"</p>
SECURITY	<p>SECURITY は、Oracle Reports Services Queue Manager を通じてキャッシュされた出力ファイルにアクセスするためのセキュリティ・レベル（0、1、2 または 3）です。0 はジョブのキャッシュにある出力に誰でもアクセスできることを意味します。1 は、そのジョブを実行したユーザーと同じユーザー ID を持つユーザーのみにキャッシュ内の出力へのアクセスが認められることを意味します。2 は、そのジョブを送信したのと同じプロセスのみにキャッシュ内の出力へのアクセスが認められることを意味します。3 は、キャッシュ内の出力へのアクセスが許可されていないことを意味します。</p> <p>デフォルト値は 1 です。</p>

パラメータ	説明
SECURITYTNSNAME	<p>SECURITYTNSNAME は、Oracle Reports Services に対するユーザーの認証に使用される Oracle Portal データベースの TNS 名です。</p> <p>Oracle Reports Services では、Oracle Portal を使用してセキュリティ・チェックを行い、また、アクセスの制限された Oracle Reports Services サーバーに対するレポートの実行、ならびに、リクエストがある場合は、アクセスの制限されたプリンタへの出力に対するアクセス特権をユーザーが持っているかを確認します。</p> <p>SECURITYTNSNAME パラメータが設定されている場合、Oracle Reports Services を使用してレポート・リクエストを処理するには、Oracle Reports Services サーバー、プリンタ、Oracle Portal にあるレポートについての情報を追加する必要があります。次に例を示します。</p> <p>SECURITYTNSNAME="sec_db"</p> <p>詳細は、第 5 章「Oracle Portal を使用した Oracle Reports Services セキュリティ」を参照してください。</p>
SOURCEDIR	<p>SOURCEDIR は、レポートやその他のランタイム・ファイルを検索する際に、REPORTS60_PATH の前に検索されるパスです。各 Oracle Reports Services は異なるディレクトリを検索できるため、この設定値は、同じ ORACLE_HOME を共有する複数の Oracle Reports Services がある場合に便利です。次に例を示します。</p> <p>SOURCEDIR="C:\my_reports"</p>
SUCCNOTEFILE	<p>SUCCNOTEFILE は、実行に成功したジョブに対して指定された電子メール・アドレスに送信された通知メッセージのテンプレートのパスとファイル名です。次に例を示します。</p> <p>SUCCNOTEFILE="C:\ORACLE_HOME\Report60\succnote.dat"</p>
TEMPDIR	<p>TEMPDIR は、テンポラリ・ファイルを作成するときに REPORTS60_TMP のかわりに使用されるディレクトリです。</p> <p>TEMPDIR はマシン上の任意のディレクトリあるいは論理ドライブに設定できます。次に例を示します。</p> <p>TEMPDIR="C:\ORACLE_HOME\Report60\temp"</p>

環境変数

この付録では、Oracle Reports Services に属する環境変数と構成パラメータに関する詳しい情報を提供します。CGI およびサープレットの環境変数の一覧については、次の表を参照してください。

環境変数は、Oracle Reports Services の動作をコントロールまたはカスタマイズするために使用される構成パラメータです。Windows NT の場合、環境変数はレジストリ・エディタを使用して設定します。UNIX の場合は、シェル・スクリプトを使用して設定します。

変数	説明
REPORTS60_COOKIE_EXPIRE	cookie の期限を分単位で決定します。デフォルト値は 30 です。 cookie は、ユーザーが保護された Oracle Reports Services サーバーにログインして、レポート・リクエストを実行する際に、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に保存します。ログインに成功すると、暗号化された cookie がブラウザに送られます。cookie が期限切れになると、それに続くリクエスト（保護された Oracle Reports Services サーバーに送られるリクエスト）のユーザーは、再度認証される必要があります。
REPORTS60_DB_AUTH	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。デフォルト値は dbauth.htm です。
REPORTS60_ENCRYPTION_KEY	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します。暗号化キーは、どの文字列でも設定できます。デフォルト値は reports6.0 です。
REPORTS60_CGIDIAGBODYTAGS	Oracle Reports Services サーバー CGI で、RWCGI60 の診断およびデバッグ出力に <BODY...> タグとして挿入される HTML タグを指定します。たとえば、テキストや背景の色および画像の設定にこの環境を使用できます。

変数	説明
REPORTS60_CGIDIAGHEADTAGS	Oracle Reports Services サーバー CGI で、RWCGI60 の診断およびデバッグ出力の <HEAD> タグと </HEAD> タグの間に挿入される HTML タグを指定します。たとえば、<TITLE> または <META...> タグの設定にこの環境を使用できます。
REPORTS60_CGIHELP	<p>Oracle Reports Services サーバー CGI で、次のように空のリクエストとともに RWCGI60 が呼び出された場合に呼び出される RWCGI60 ヘルプ・ファイルの URL または URI を定義します。</p> <p><code>http://your_webserver/rwcgi60?</code></p> <p>たとえば、これを <code>http://www.yahoo.com</code> に設定するとその URL に移動し、次のように <code>myhelpfile.htm</code> に設定するとファイルを表示します。</p> <p><code>http://your_webserver/myhelpfile.htm</code></p> <p>このパラメータが未定義の場合、デフォルトのヘルプ画面が表示されます。</p>
REPORTS60_CGIMAP	<p>Oracle Reports Services サーバー CGI で、マップ・ファイル構成を使用している場合、RWCGI60 マップ・ファイルの完全修飾ファイル名および位置を定義します。次に例を示します。</p> <p><code>C:\¥ORANT¥REPORT60¥cgicmd.dat</code></p>
REPORTS60_CGINODIAG	<p>Oracle Reports Services サーバー CGI で、ヘルプやショーマップなどのデバッグおよび診断出力が定義されている場合、それらすべての RWCGI60 からの出力を無効にします。たとえば、REPORTS60_CGINODIA が定義されていると、次の行は機能しません。</p> <p><code>http://your_webserver/rwcgi60/help?</code></p>
REPORTS60_REPORTS_SERVER	CGI リクエスト用のデフォルトの Oracle Reports Services サーバーを指定します。この環境変数が設定されている場合、レポート・リクエストでデフォルトのサーバーを使用して処理を行う際に SERVER コマンドライン引数を使用して指定する操作を省略できます。また、SERVER コマンドライン引数を使用してデフォルトを書き換えることもできます。
REPORTS60_SSLPORT	SSL を使用していて 443 以外のポート番号を用いる場合、この変数を使用して別のポート番号を指定できます。デフォルト値は 443 です。

変数	説明
REPORTS60_SYS_AUTH	アクセスの制限された Oracle Reports Services サーバーに対するレポート・リクエストをユーザーが実行する際に、ユーザー名とパスワードを認証するために使用する認証テンプレートを指定します。

データベース接続文字列

ここでは、CGI または サブレット を使用して レポート・リクエスト を指定するときに使用できる、一般的な データベース接続文字列 をリストしています。データベース接続文字列とは、データベースに接続するために USERID コマンドライン引数の中で使用する値を指します。

USERID コマンドライン引数の詳細は、[付録 B「RWCLI60 コマンドライン引数」](#) を参照してください。

データベース接続文字列	Oracle Reports Services の応答	ユーザー・アクション
USERID 未設定	データベース認証フォームを返します。	Oracle または プレースホルダ・ユーザー名およびパスワードを入力します。
Oracle username@database	Oracle の接続文字列表からユーザー名とデータベースの組を探して、パスワードを取得します。Oracle Reports Services がパスワードを見つけると、レポートが実行されます。 パスワードが見つからなかった場合、Oracle Reports Services はデータベース認証フォームを返します。	なし。 データベース・パスワードを入力します。
Oracle username/password@database	接続文字列を受け、レポートを実行します。	なし。

データベース接続文字列	Oracle Reports Services の応答	ユーザー・アクション
Oracle username/password	ローカルのデータベースを使用し、レポートを実行します。	なし。
	ローカルのデータベースがない場合、Oracle Reports Services は、データベース認証フォームを返します。	Oracle データベースを入力します。
<\$username>@database	接続文字列表のブレースホルダ・ユーザー名を探します。ユーザー名が見つからなかった場合、Oracle Reports Services はデータベース認証フォームを返します。	Oracle ユーザー名およびパスワードを入力します。
	Oracle Reports Services が表中のブレースホルダ・ユーザー名を見つけられなかった場合は、Oracle ユーザー名およびデータベース名を探して、パスワードを取得します。Oracle Reports Services がパスワードを見つけると、レポートが実行されます。	なし。
	パスワードが見つからなかった場合、Oracle Reports Services はデータベース認証フォームを返します。	データベース・パスワードを入力します。

データベース接続文字列	Oracle Reports Services の応答	ユーザー・アクション
<\$username>/password@database	<p>接続文字列表のプレースホルダ・ユーザー名を探します。ユーザー名が見つかると、Oracle Reports Services はレポートを実行します。</p> <p>プレースホルダ・ユーザー名が見つからなかった場合、Oracle Reports Services はデータベース認証フォームを返します。ユーザーがレポートを実行するためには、認証が必要となります。</p>	<p>なし。</p> <p>Oracle ユーザー名およびパスワードを入力します。</p>

Web カートリッジから CGI への移行

この付録では、Web カートリッジから CGI に移行する方法を、順を追って説明します。ここでは、Web カートリッジを使用して Oracle Application Server (OAS) 上に構成されている Oracle Reports Services サーバーがすでに存在することを前提としています。

注意： Oracle Portal Listener、CGI 対応の任意の Web サーバーまたは Java 関連の任意のサーブレットを使用して Oracle Reports Services サーバーを構成した場合は、すでに Oracle Reports Services サーバー CGI の構成が終了しています。移行は必要ありません。

F.1 CGI への移行の利点

CGI は Web の HTTP プロトコルのコンポーネントです。これは、Oracle Reports Services サーバーと動的に通信するための、プラットフォームに依存しない標準の方法です。利点は、次のとおりです。

- オープン性

ほとんどの Web サーバーが CGI をサポートしています。これが最も一般的な実装です。

- 容易な実装

CGI の実装は、Web カートリッジよりも短時間で済み、かつ容易です。

F.2 CGI に移行するためのステップ

CGI に移行するには、次のステップが必要です。

1. ソフトウェアのインストール
2. OAS の構成
3. CGI の構成
4. 環境変数の設定（オプション）
5. マップ・ファイルの名前の変更（オプション）
6. CGI URL を使用したレポートの実行
7. Web ページ上のレポート・リンクの更新

これらのステップは、OAS マシン上で実行します。また、これらのステップは、Web カートリッジを使用する Oracle Reports Services サーバーがすでに存在することを前提としています。

F.2.1 ステップ 1. ソフトウェアのインストール

ソフトウェアをインストールするには、次のようにします。

1. Oracle Reports Services Developer Thin Client をインストールします（まだインストールしていない場合）。
2. Oracle Reports Services 用の TNSnames サービス・エントリが、Oracle Reports Services サーバーと通信するために CGI によって使用される `tnsnames.ora` ファイルの中にあることを確認します。
 - OAS が Oracle Reports Services とは異なるマシン上にインストールされている場合は、OAS の `ORACLE_HOME` にある `tnsnames.ora` ファイルを調べて、Oracle Reports Services サーバーの TNSnames サービス・エントリが存在することを確認します。
 - OAS リリース 4.0.8 が Oracle Reports Services と同じマシン上の異なる Oracle ホームにインストールされている場合は、Oracle Reports Services サーバーの TNSnames サービス・エントリを OAS の `ORACLE_HOME` にある `tnsnames.ora` ファイルに追加する必要があります。
 - OAS リリース 4.0（OAS 4.0.8 より前のリリース）が Oracle Reports Services と同じマシンにインストールされていて、同じ `ORACLE_HOME` 内にある場合は、追加の TNSnames エントリは不要です。

F.2.2 ステップ 2. OAS の構成

OAS を構成するには、次のようにします。

1. ブラウザを起動します。
2. Oracle Application Server の Welcome ページで、「OAS Manager」をクリックします。
3. OAS マネージャのナビゲーション・ツリーで、Web ホーム・サイト・アイコンの横の「+」アイコンをクリックします。
4. HTTP リスナー・ノードを拡張します。
5. 必要に応じて、リスナーを作成するか、使用するリスナーを拡張します。
6. 「ディレクトリ」をクリックし、次の表の情報を使用して OAS ディレクトリ・マッピングを構成します。

ディレクトリの説明	物理ディレクトリ例	仮想ディレクトリ例	必要な権限
BIN	C:\OAS\BIN	/CGI-BIN	読み込みおよび実行

F.2.3 ステップ 3. CGI の構成

CGI を構成するには、次のようにします。

1. Windows では、(ORACLE_HOME\BIN ディレクトリにある) rwcgi60.exe ファイルを CGI-BIN ディレクトリにコピーします。UNIX では、(ORACLE_HOME\BIN ディレクトリにある) rwcgi60 ファイルを CGI-BIN ディレクトリにコピーします。CGI-BIN ディレクトリは、OAS ディレクトリ・マッピングで定義されます。この例の CGI-BIN ディレクトリは、C:\OAS\BIN です。
2. OAS と Oracle Reports Services が異なるホーム・ディレクトリにある場合は、Oracle Reports Services ホームを OAS ホームにコピーすることも必要です。

Windows では、Oracle Reports Services ホームが D:\ORANT\REPORTS60 で OAS ホームが E:\ORANT\OAS である場合、E:\ORANT\OAS 内に REPORTS60 サブディレクトリを作成し、テンプレート・ファイルを D:\ORANT\REPORTS60 から E:\ORANT\OAS\REPORTS60 に移動する必要があります。

UNIX では、Oracle Reports Services ホームが /private1/oracle6i/reports60 で OAS が /private1/oas である場合、次のコマンドを実行します。

```
cd /private1/oas
mkdir reports60
cd reports60
cp /private1/oracle6i/reports60/dbauth.htm
cp /private1/oracle6i/reports60/sysauth.htm
cp /private1/oracle6i/reports60/dbsysdif.htm
cp /private1/oracle6i/reports60/dvsyssam.htm
```

F.2.4 ステップ 4. 環境変数の設定（オプション）

Web カートリッジ環境変数（たとえば、マップ・ファイルまたはヘルプ・ファイルの場所を指定するための `REPORTS60_OWSHELP`）を設定する場合は、CGI 用の環境変数（たとえば `REPORTS60_CGIHELP`）を設定する必要があります。

現在使用されている Web カートリッジ環境変数を表示するには、ブラウザを起動して、環境変数表示コマンドとともに OAS カートリッジの URL を入力します。たとえば、次のように入力します。

```
http://my_webserver/rwows?showenv
```

F.2.4.1 Windows NT

オラクル社では、変更を行う前に、まずレジストリのバックアップを取ることをお勧めしています。次のようにします。

1. デスクトップで、「スタート」 「ファイル名を指定して実行」を選択します。
2. `regedit` と入力してレジストリ・エディタを表示します。
3. `Hkey_Local_machine Software Oracle` を拡張します。
4. 「編集」 「新規作成」 「文字列」値を選択して、CGI 環境変数を追加します。

詳細はオペレーティング・システムのドキュメントを参照してください。

F.2.4.2 UNIX

OAS マシン上の環境変数は、シェル・スクリプトを作成して設定することもできます。そうするには、設定する各環境変数について、次のようなコマンドの入ったファイルを作成します。次に例を示します。

```
setenv REPORTS60_CGIHELP myhelp.html
```

F.2.5 ステップ 5. マップ・ファイルの名前の変更（オプション）

キー・マップ・ファイルを使用してパラメータを単純化したり隠したりする場合は、Web カートリッジ用のキー・マップ・ファイルの名前（たとえば `owscmd.dat`）を、CGI が認識できるファイル名（たとえば `cgicmd.dat`）に変更する必要があります。
(`ORACLE_HOME¥REPORTS60` にある) `owscmd.dat` ファイルをコピーして、`cgicmd.dat` に改名できます。

これで、Web カートリッジから CGI への移行が完了します。

F.2.6 ステップ 6. CGI URL を使用したレポートの実行

正常に CGI に移行できたかどうかをテストする必要があります。構成をテストして、その構成で Oracle Reports Services サーバーと通信できることを確認します。CGI の URL は、Web カートリッジの URL とは異なります。次の CGI URL を使用してテストを実行します。この例では、REPORTS60_REPORT_SERVER 環境変数がデフォルトの Oracle Reports Services サーバーをポイントするように設定されていることを前提としています。この場合、SERVER コマンドライン引数は不要です。

```
http://your_webserver/CGI-BIN/RWCGI60.EXE?REPORT=your_report.RDF+userid=username/  
password@mydb+DESFORMAT=HTML+DESTYPE=CACHE
```

(Web カートリッジ用の) RWOWS60 実行ファイルを使用せずに、(ステップ 2 で定義した) CGI-BIN パスから (CGI 用の) RWCGI60 実行ファイルをコールして、URL をコールすることに注意してください。Oracle Reports Services サーバーとの通信に Web カートリッジまたは CGI のどちらを使用している場合でも、? (疑問符) の後の引数は同じです。

F.2.7 Web ページ上のレポート・リンクの更新

レポート・リクエストを実行するリンクを持つ Web ページを保持する場合は、CGI-BIN パスから RWCGI60 実行ファイルをコールするように URL 参照を変更する必要があります。

Oracle Portal を使用したアクセス制御に対応するよう Oracle Reports Services サーバーを構成した場合は、Oracle Reports Services サーバー・アクセス制御にある、Oracle Reports Services サーバー用に作成された Web ゲートウェイ値を変更する必要があります。

トラブルシューティング

付録 F には、Oracle Reports Services 構成のトラブルシューティングに関する情報が記載されています。

トラブル	考えられる原因と対処方法
Oracle Reports Services を起動してもハング・アップしているように見える	<code>tnsnames.ora</code> ファイルに構文エラーがあり、Oracle Reports Services が TNS 名を解決できない可能性があります。 メモリーに原因がある場合には、再起動してください。
「デーモンはポートのリスニングに失敗しました」というエラー・メッセージが表示される	ある Oracle Reports Services が動作しているポートと同じポートをリスニングしている Oracle Reports Services を起動すると、このエラーが発生します。Net8 か TCP/IP セットアップの問題の可能性があります。
プリンタ (REP 3002) が初期化できない	Oracle Reports Services がプリンタへのアクセス権限を持っているかどうかを確認してください。Windows NT の場合、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。Oracle Reports Services を NT サービスとしてインストールし、「 ログオン 」フィールドにプリンタ・アクセス権限を設定せずに システム・アカウント または別のアカウントを使用している状況が考えられます。「 ログオン 」フィールドで、デフォルトのプリンタ・アクセス権限を持つアカウントを指定する必要があります。このプリンタは必ず存在する必要はありませんが、ドライバは必ずインストールする必要があります。 UNIX の場合は、 <code>uiprint.txt</code> ファイル内にプリンタを構成してください。

トラブル	考えられる原因と対処方法
Oracle Reports Services を起動すると、サーバー・エラー 186 が表示される	<p>通常、これは <code>tnsnames.ora</code> または <code>sqlnet.ora</code> に問題があることを示しています。<code>tnsnames.ora</code> 内の Oracle Reports Services のエントリを調べてください。通常のエントリは次のような記述になっています。</p> <pre>repserver.world = (ADDRESS=(PROTOCOL=tcp) (HOST=144.25.87.182)(PORT=1951))</pre> <p>この例では、<code>.world</code> が名前に付加されていますが、これは <code>world</code> が <code>sqlnet.ora</code> ファイルに指定されているドメインであるためです。<code>sqlnet.ora</code> に <code>NAMES.DEFAULT_DOMAIN</code> 設定値が定義されていない場合は、サーバー・インスタンスの名前の <code>.world</code> は省略してください。</p> <p><code>tnsnames.ora</code> ファイルに問題がないようであれば、<code>sqlnet.ora</code> ファイルを調べてください。このファイルで使用する適正なデフォルト値は、次のとおりです。</p> <pre>TRACE_LEVEL_CLIENT=OFF names.directory_path = (TNSNAMES) names.default_domain = world name.default_zone = world</pre> <p>プロトコルが TCP の場合は、Net8 TCP/IP アダプタと Net8 がインストールされていることを確認してください。最後に、インストールされている Net8 のバージョンが、Oracle Reports Services に付属しているバージョンより前のものでないことを確認してください。</p>
レポートを開くときにエラーとなる	<p>レポートの名前と拡張子を調べてください。UNIX マシンの場合、実際のレポート名の太文字と小文字は URL で指定されているとありである必要があります。Windows でエクスプローラを使用している場合は、コピーしているファイルおよび名前を変更しているファイルの拡張子を非表示にしないでください。(エクスプローラで「表示」「オプション」を調べてください。) この状態になっていると、<code>your_report.rdf.txt</code> のような名前でファイルが作成できません。かわりに DOS ウィンドウでファイルを操作してください。</p> <p>または、<code>REPORTS60_PATH</code> 環境変数で定義されたパスにレポートが存在することを確認してください。</p>

トラブル	考えられる原因と対処方法
Windows NT サービスとして Oracle Reports Services を実行中のトラブル	<p>SYSTEM 以外のユーザーの元で運用するように Oracle Reports Services サービスをインストールしている場合は、次の事項を確認してください。</p> <ul style="list-style-type: none"> ■ ユーザー・アカウントに、ユーザー マネージャで選択されている「パスワードを無期限にする」オプションが付いている。 ■ ユーザー・アカウントが、Oracle Reports Services を実行し、レポート・ファイルにアクセスするための適切なグループのメンバーになっている。 ■ ユーザー・アカウントが少なくともデフォルト・プリンタの印刷権限を持っている。 ■ ユーザー・アカウントがサービスにログインできる。「スタート」「プログラム」「管理ツール」「ユーザー マネージャ」「原則」「ユーザーの権利」の順で選択してください。「高度なユーザー権利の表示」をオンにします。「権利」リストから「サービスとしてログオン」を選択します。「この権利を与えられたアカウント」リストにユーザーがまだ登録されていない場合は、「追加」ボタンをクリックします。 <p>サービスを起動するときは、ユーザー名の他にドメインも明示的に指定する必要があります（ユーザー名およびドメイン）。Windows NT から、このサービスが正常に処理できなかったというメッセージが送られ、エラー・メッセージ番号が返された場合は、Oracle Reports Services Builder オンライン・ヘルプでメッセージ番号を探してください。</p>
ops\$ アカウントが機能しない	<p>セキュリティ上の理由により、Oracle Reports Services では ops\$ アカウントはサポートされません。USERID=/ を付けてコマンドラインを Oracle Reports Services に渡すと、エラーとなります。Oracle Reports Services はクライアントのユーザー名ではなく Oracle Reports Services が処理するユーザー名の使用を試みるためです。</p>
データベース・ロールが予想どおり動作しない	<p>データベース・ロールを利用している場合、Oracle Reports Services はジョブ・リクエストのデータベース接続に適したデフォルトのロールを取り込み、それを設定します。デフォルトのロールがパスワードを必要とする場合、Oracle Reports Services はいったんログアウトしてから、再びデータベースにログインします。したがって、ロール名レポート・プロパティを使用して、パスワードを必要とするロールをレポートそのものに入れておくのが賢明な方法です。Oracle Reports Services はジョブ単位にデフォルトのロールを取り込んで設定するので、ジョブ間でロールを共有させることはできません。これは、セキュリティを確保するためのものです。</p>

トラブル	考えられる原因と対処方法
URL マッピングが機能しない	<p>キー・マップ・ファイルが有効か確認してください。このファイルは、REPORT60 ディレクトリ内にある cgicmd.dat という名前のファイル（Oracle Reports Services サーバー CGI またはサブレットの場合）か、REPORTS60_CGIMAP 環境変数に設定されている値に従った名前のファイルである必要があります。</p> <p>キー・マップ・ファイルを見つけ出すために、まず次のように入力し（CGI の例）、キー・エントリが生成されたページ上で正しく構文解析されていることを確認してください。</p> <p><code>http://your_webserver/your_virtual_cgi_dir/rwcgi60.exe/showmap?</code></p> <p>その後、次のようにキー・マップ・エントリを使用してレポートを実行してください。your_key はキー・マップ・ファイルの有効なキー・エントリです。</p> <p><code>http://your_webserver/your_virtual_cgi_dir/rwcgi60.exe?your_key</code></p>
Oracle Reports Services Queue Manager からキューを停止できない	<p>管理者として最初にログインするときは、ユーザー名とパスワードを必ず入力する必要があります。Oracle Reports Services Queue Manager からキュー管理者として最初にログインするときは（「オプション」「管理者ログオン」）、任意のユーザー名とパスワードを指定できます。最初に指定するユーザー名とパスワードは、変更されるまで管理者のユーザー名とパスワードになります。</p>
LocalSystem の下で Oracle Reports Services を NT サービスとして利用できない	<p>LocalSystem のユーザー ID の下で Oracle Reports Services を NT サービスとして実行できない場合は、sqlnet.ora ファイル内に次の行が記述されているかをシステム管理者が確認する必要があります。この行が記述されていない場合は、サーバーにアクセスできません。</p> <p><code>sqlnet.authentication_services=(NONE)</code></p>
ファイルが見つかりません	<p>ネットワーク・ドライブはユーザーごとにドライブ文字にマップされるので、Windows NT ユーザーがログアウトすると、これらのマッピングは効力を失います。ドライブ文字を使用して Oracle Reports Services にこれらのドライブを参照させることはできません。かわりに、UNC パス名を使用する必要があります。次に例を示します。</p> <p>¥¥SALES¥DOCUMENTS¥REPORTS</p> <p>これは Oracle Reports Services のパラメータ、CGI およびサブレットのコマンドのマッピング、ならびに実行中のそれぞれのレポート内のハードコード化されたパス名にも該当します。</p>

トラブル	考えられる原因と対処方法
レポート出力が正常に開けないと、Web サーバーから報告があった	Web サーバーからレポート出力が正常に開けないと報告があった場合、レポート名と拡張子を調べてください。UNIX マシンの場合、実際のレポート名の太文字・小文字は URL で指定されているとおりになっている必要があります。エクスプローラを使用して Windows 上で操作している場合は、コピーしているファイルおよび名前を変更しているファイルの拡張子がエクスプローラ上で非表示になっていないか確認してください(「表示」「オプション」)。この状態になっていると、your_report.rdf.txt のような名前でファイルが作成できません。かわりに DOS ウィンドウでファイルを操作してください。
レポートは設計プラットフォーム(例: Windows)上では正常に実行されるが、サーバー・プラットフォーム(例: UNIX)上では正常に実行されない	設計プラットフォームで使用しているバージョンがサーバー上で使用しているものと一致しているか確認してください。一致していない場合は、バージョンの相違が問題の原因になっている可能性があります。
Oracle Portal で Oracle Reports Services レポート定義ファイルへのアクセス権限を作成しようとしたとき、無効なパッケージが作成された	<p>Oracle Portal で、プリンタ、Oracle Reports Services サーバーおよびレポート定義ファイルに対して定義したアクセス制御を確認してください。</p> <p>次の点を確認してください。</p> <ul style="list-style-type: none"> ■ プリンタ・アクセス・ウィザードで定義した OS プリンタ名が適正か。Reports 定義ファイル・アクセス・ウィザードの「必須パラメータ」ページにプリンタが表示されていない場合、OS プリンタ名が正しく入力されていない可能性があります。 ■ Oracle Reports Services サーバーへのアクセス権限とプリンタ(オプション)が生成されているか。 ■ レポート定義ファイル、サーバーおよびプリンタへのアクセス権限を必要とするユーザーに、それらのアクセス権限が与えられているか。 <p>必要であれば修正し、このレポート定義ファイルに有効な製品パッケージを作成してください。このレポートを特定の Oracle Reports Services サーバーに対して発行するには、有効な製品パッケージを作成する必要があります。</p>

トラブル	考えられる原因と対処方法
URL が要求されているとき、レポートが実行されない	<p data-bbox="601 262 879 288">次の点を確認してください。</p> <ul style="list-style-type: none"> <li data-bbox="601 305 1276 409">■ Web サーバーが応答していることを確認してください（たとえば、Web サーバー管理ページを呼び出してみてください）。Web サーバーのインストール・ドキュメントを参照してください。 <li data-bbox="601 427 1276 531">■ CGI またはサブルーットの実行可能ファイルが検索でき、応答しているか確認してください。Windows 95 および Windows NT の場合は、ブラウザの URL フィールドに次のいずれかの行を入力してください。 http://your_webserver/your_virtual_cgi_dir/rwcgim60.exe または http://your_webserver/rwows UNIX の場合は、次のいずれかの行を入力してください。 http://your_webserver/your_virtual_cgi_dir/rwcgi60 または http://your_webserver/rwows ヘルプ・ページが表示されます。表示されない場合は、Web サーバー構成ファイル内の your_virtual_cgi_dir（通常は cgi-bin）のマッピングを調べてください。Web サーバー上の既存の物理ディレクトリにマップされている必要があります。この物理ディレクトリに RWCGI60 実行形式のファイルのコピーも存在している必要があります。 <li data-bbox="601 904 1276 1138">■ REPORTS60_CGINODIAG（CGI またはサブルーットの場合）環境変数が定義されていないことを確認してください。定義されている場合は、すべての診断出力が無効になります。次のいずれかの行を入力して、これを確認してください。 http://your_webserver/your_virtual_cgi_dir/rwcgi60.exe/ showenv? http://your_webserver/rwows/showenv? 上の行を入力すると、他のパラメータまたは環境変数を確認できます。

トラブル	考えられる原因と対処方法
	<ul style="list-style-type: none"> <div data-bbox="646 269 1330 413"> <p>REPORTS60_PATH 環境変数が定義されていることを確認してください。次のいずれかの行を入力して、環境変数を調べてください。</p> <pre>http://your_webserver/you_virtual_cgi_dir/rwcgi60.exe/showenv?http://your_webserver/rwows/showenv?</pre> </div> <div data-bbox="646 421 1330 644"> <p>次のいずれかの行を入力して、単純なレポートをブラウザに表示してください。</p> <pre>http://your_webserver/your_virtual_cgi_dir/rwcgi60.exe?server=your_repserver+report=your_report.rdf+userid=scott/tiger@mydb+desformat=html http://your_webserver/rwows?server=your_repserver+report=your_report.rdf+userid=scott/tiger@my_db+desformat=html</pre> </div> <p>レポートが表示されない場合は、</p> <ul style="list-style-type: none"> <div data-bbox="646 696 1330 829"> <p>Oracle Reports Services Builder より <code>Your_report.rdf</code> が正常に実行されているか、もしくは指定されたディレクトリに、Oracle Reports Services Runtime の <code>Your_report.rdf</code> が <code>REPORTS60_PATH</code> に基づいて配置されているか確認してください。</p> </div> <div data-bbox="646 843 1330 892"> <p>データベース接続文字列が適正であることを確認してください。</p> </div> <div data-bbox="646 906 1330 1041"> <p>レポートを発行しようとしている Oracle Reports Services サーバーが、制限されていないかを確認してください。制限を受けている場合には、サーバーへのアクセス特権を得る必要があります。Oracle Reports Services のシステム管理者に相談してください。</p> </div> <div data-bbox="646 1055 1330 1189"> <p>発行しようとするレポートが制限されていないかを確認してください。その場合には、制限されている Oracle Reports Services サーバーを実行するためのアクセス権限を得る必要があります。Oracle Reports Services のシステム管理者に相談してください。</p> </div> <p>Oracle Reports Services サーバーは、レポートとレポートが使用するすべての外部ファイルへアクセスする権限を持っている必要があることに注意してください。</p> <p>レポートを Oracle Reports Services サーバーに送るとき、レポート定義にパラメータ値が明示的に設定されている場合は、パラメータ用の In Report 値を使用する必要があります。たとえば、Oracle Reports Services Queue Manager からレポートを起動していると想定します(「ジョブ」「新規作成」)。「レポート・モード」および「オリエンテーション」パラメータに「レポート」を指定し、そのどちらもレポート定義に指定されている値を持っていない場合は、このジョブは正常に実行されません。</p>

トラブル	考えられる原因と対処方法
レポートがプリンタに出力されない	<p>制限された Oracle Reports Services サーバーに対してレポートを発行するアクセス権限を持っていますが、レポートを出力するプリンタに対するアクセス権限を持っていない可能性があります。Oracle Reports Services のシステム管理者に相談してください。</p>
ホスト名参照エラー	<p>レポート・リクエスト実行時に、誤った URL が入力されました。正しい URL を使用してレポート・リクエストをもう一度出力してください。URL についての十分な知識がない場合は、システム管理者に相談してください。</p> <p>制限された Oracle Reports Services サーバーに対してレポートを発行している場合、Oracle Portal のサーバー・アクセスに定義されている Web Gateway URL が不適正な可能性があります。</p> <p>Oracle Portal で、Oracle Portal メニューから「管理」をクリックします。次に「Oracle Reports Developer セキュリティ」と「サーバー・アクセス」をクリックします。編集する Oracle Reports Server アクセスを検索してください。サーバー・アクセス・ウィザードの「サーバー名とプリンタ」ページ上の Web Gateway URL を確認してください。</p> <p>注意: Oracle Portal の Oracle Reports Services セキュリティにアクセスできるのは、Oracle Reports Services システム管理者としての権限を持つユーザーのみです。</p>

用語集

CGI (Common Gateway Interface: コモン・ゲートウェイ・インタフェース)

Web サーバー上でアプリケーションを実行するための標準技術。CGI は、Web サーバー上で実行されているプログラムが他のコンピュータと通信を行い、ユーザーから入力された情報に応じて動的な HTML 文書の生成を可能にする。

Cookie

ユーザーに関する情報を記憶するために Web サーバーがユーザーのハードディスクに設ける特殊なテキスト・ファイル。ユーザーがレポート・リクエストを保護された Oracle Reports Services サーバーに出すと、認証が行われる。ログオンに成功すると、暗号化された cookie がブラウザに送られる。cookie が期限切れになると、それに続くリクエスト（保護された Oracle Reports Services サーバーに送られるリクエスト）は、再度認証される必要がある。

CSS (Cascading Style Sheets: カスケード・スタイル・シート)

CSS 付き HTML によって、複数の Web ページのスタイルやレイアウトを一度にコントロールすることが可能になる。スタイル・シートは、テンプレートのような役割を果たすものであり、フォント属性や色などのスタイル情報の集まりである。カスケードは、スタイル情報をどのように使用するかを決めるために Web ブラウザが使用する一連のルールのことを言う。Netscape Navigator 4.0 以降および Internet Explorer 4.0 以降では、カスケード・スタイル・シートがサポートされている。

HTML (Hypertext Markup Language: ハイパーテキスト・マークアップ言語)

インターネットの Web サーバー上のドキュメントのコンテンツや他のドキュメントへのリンクを指定するために使用される、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: ハイパーテキスト・トランスファ・プロトコル)

Web ブラウザを持つコンピュータとアクセス中の Web サーバーの間で Web トラフィックの転送に使用されるプロトコル。

IP (Internet Protocol: インターネット・プロトコル)

インターネットにおける基本的なプロトコル。これにより、あるホストから別のホストへ個別のパケットを配布できる。通常、パケットが配布されるか、配布にどれぐらいの時間がかかるのか、複数のパケットを配布する場合、送られたとおりの順番で届くのかなどについては、保証されない。しかし、プロトコルが設けられることにより、信頼性や接続の概念が加えられる。

Net8

すべてのネットワークにおいて、クライアントとサーバー間、およびサーバーとサーバー間の通信を可能にする、オラクル社のリモート・データ・アクセス・ソフトウェア。送られた処理や送られたデータベースの能力をサポートし、多くの通信プロトコルとかがわりを持ち、それらのプロトコル全体をカバーしている。

Oracle Portal

Oracle Portal は、拡張可能で確実な HTML アプリケーションや Web サイトを構築するための HTML ベースの開発ツールである。Oracle Reports Services では、レポート・リクエスト、保護されたサーバーおよびレポート出力のプリントに使用される Oracle Reports Services プリンタに関する情報を保持することによって、Web 上で公開されているレポートへのエンド・ユーザーのアクセスをコントロールするために、Oracle Portal が使用されている。

Oracle Portal コンポーネント

Oracle Portal コンポーネント・ウィザードによって作成された PL/SQL のストアド・プロシージャ（たとえば、チャート、フォームまたは Oracle Reports Services のレポート定義ファイル・パッケージ）。ストアド・プロシージャを実行することで、コンポーネントを表示するために使用される HTML コードが作成される。

Oracle Reports Services

複数層アーキテクチャを持つリモート・サーバー上でレポートを実行できる。Windows NT、Windows 95 または UNIX 上にインストールできる。Oracle Reports Services は、クライアント・リクエストを操作して、すべてのジョブ・リクエストを 1 つのジョブ・キューに入れてレポートを実行できる。

Oracle Reports Services Launcher

指定したレポートの実行リクエストを Oracle Reports Services に送信するなど、Oracle Reports Services の Active X コントロールによって提供される機能を使用するアプリケーション。

Oracle Reports Services Queue Manager

これにより、Oracle Reports Services に送られたジョブ・リクエストを監視、操作できる。

Oracle Reports Services サーバー CGI

CGI 関連の Web サーバーと Oracle Reports Services Runtime の間のインタフェース。これにより、Web ブラウザからレポートを動的に実行することが可能になる。

Oracle Reports Services サーバー・サーブレット (Oracle Reports Services Server Servlet)

Java ベースの Web サーバーと Oracle Reports Services Runtime 間のインタフェース。これにより、Web ブラウザからレポートを動的に実行することが可能となる。

Oracle9i Application Server

Oracle9i Application Server は、ネットワーク・アプリケーションの実行において非常に重要なプラットフォームである。アプリケーション論理をアプリケーション・サーバーに移動させ、ネットワーク・クライアントを実行することにより、組織は複雑さの改善、操作性の向上および開発と実行の簡便性の向上によって、実質的な利益を得られる。Oracle9i Application Server は、データベースの Web 配布の簡便化、および従来のクライアント・サーバーから、ネットワーク・アプリケーション・アーキテクチャへの移行を通じての財産の完全統合を実現する、ビジネスにおいては非常に重要なプラットフォームである。

ORACLE_HOME

いくつかのディレクトリ・ベースのオペレーティング・システムにおける、Oracle のディレクトリ階層の最上層の別名。Oracle 製品のルート・ディレクトリを示す環境変数。

PDF (Portable Document Format: ポータブル・ドキュメント形式)

そのドキュメントを作成するために使用されたオリジナルのアプリケーション・ソフトウェア、ハードウェアおよびオペレーティング・システムからの影響を受けないドキュメントを示す (Adobe Acrobat 用の) ファイル形式。PDF ファイルは、装置独自の、または解像度特有の形式を持つテキスト、グラフィックおよび画像のいかなる組み合わせも含むことができる。

RWCLI60

コマンドラインを解釈し、指定した Oracle Reports Services (RWMTS60) に送る実行ファイル。

TCP/IP (Transmission Control Protocol based on Internet Protocol)

あるホストから別のホストへの信頼性の高いデータ配信のためのインターネット・プロトコル。

tnsnames.ora

サービス名に割り当てられた接続記述を含む Net8 ファイル。このファイルは、すべてのユーザーが使用するのか、個別のユーザーが使用するのかによって、総合的またはローカルに保持される。

URI (Uniform Resource Identifier: 同一リソース識別子)

抽象的または物理的リソースを識別するために使用されるロケーション (URL) の簡潔な文字列表記。URI は、エンコードされたアドレス文字列を使用してオブジェクトにアクセスする目的でインターネットのために開発されたアドレッシング・スキームまたはプロトコルの 1 つである。

URL (Uniform Resource Locator)

URI の形式である URL は、インターネット上で有効なリソースのロケーションを表す簡潔な文字列表記である。また、これは Oracle9i Application Server へのリクエストをエンコードするためにクライアントが使用するテキスト文字列の形式でもある。

Web サーバー (Web Server)

リモート Web ブラウザからの HTTP リクエストに応じて Web ページを表示した Web サイト上で実行されるサーバー・プロセス (HTTP デモン)。

Web ブラウザ (Web browser)

エンド・ユーザーが、コンピュータに保存されている HTML ドキュメントやプログラムを読むために使用するプログラム (Web サーバーによって動作する)。

キャッシュ (cache)

現在アクセスしている、またはユーザーによって変更されたデータベースのデータ、またはユーザーをサポートするために Oracle Reports Services サーバーが要求するデータのための一時記憶領域。多くは互換的に使用される。

ドメイン (Domain)

データベースのようなネットワーク・オブジェクトを、ネットワーク・サービスのネーミングを明確にしつつグループ化したもの。

認証 (Authentication)

ユーザー、装置またはコンピュータ・システムにおけるその他のエンティティの識別を確認するプロセス。多くはシステム内のリソースへのアクセスを許可するための必要条件として行われる。

フェイルオーバー (Fail-over)

あるコンポーネントが失敗した場合に、類似した別のアクティブ・コンポーネントを使用して計算システムを再構成する機能。

プッシュ配布 (Push delivery)

クライアントからのリクエストではなく、サーバーによって行われる情報の配布。Oracle Reports Services は、レポート・リクエストが保護された Oracle Reports Services サーバー上で自動的に実行されるようにスケジュールすることにより、レポートを Oracle Portal サイトにプッシュできる。エンド・ユーザーは Oracle Portal サイト上のリンクをクリックすることでレポートを表示できる。

プレースホルダ・ユーザー名 (Placeholder user name)

ユーザーは、プレースホルダ・ユーザー名によって、Oracle のデータベース・ユーザー名ではなくそのユーザー独自のユーザー名を使用してデータベースにログオンできる（たとえば、\$user_name@database）。プレースホルダ・ユーザー名によって、次のことが可能になる。

- ユーザーが、複数のレポートを 1 つのデータベースから一度に実行できる。
- 複数のエンド・ユーザーが、同じレポートを実行し、それぞれの結果を取得できる（たとえば、あるユーザーは東日本の販売結果を取得し、別のユーザーは西日本の販売結果を取得する、など）。

ただし、ユーザーが最初にデータベースにログオンするときは、Oracle ユーザー名およびパスワードを使用してログオンする必要がある。それに続くリクエストに対しては、Oracle Reports Services がユーザー独自のユーザー名をデータベース接続表から探し出す。独自のユーザー名が見つかり、Oracle Reports Services サーバーは対応するパスワードを cookie から取得し、レポートを実行する。

ポート (Port)

特定のプログラムに送られた、または特定のプログラムから送られてきたデータの経路を決めるために TCP が使用する番号。

索引

記号

<!-- --> XML タグ・リファレンス, 7-27
<![CDATA[]]> XML タグ・リファレンス, 7-28
<condition> XML タグ・リファレンス, 7-29
<customize> XML タグ・リファレンス, 7-31
<data> XML タグ・リファレンス, 7-33
<dataSource> XML タグ・リファレンス, 7-34
<exception> XML タグ・リファレンス, 7-35
<field> XML タグ・リファレンス, 7-37
<formLike> XML タグ・リファレンス, 7-42
<formula> XML タグ・リファレンス, 7-43
<function> XML タグ・リファレンス, 7-45
<group> XML タグ・リファレンス, 7-46
<groupAbove> XML タグ・リファレンス, 7-48
<groupLeft> XML タグ・リファレンス, 7-49
<labelAttribute> XML タグ・リファレンス, 7-50
<layout> XML タグ・リファレンス, 7-52
<link> XML タグ・リファレンス, 7-55
<matrix> XML タグ・リファレンス, 7-57
<matrixCell> XML タグ・リファレンス, 7-60
<matrixCol> XML タグ・リファレンス, 7-61
<matrixRow> XML タグ・リファレンス, 7-62
<object> XML タグ・リファレンス, 7-63
<programUnits> XML タグ・リファレンス, 7-65
<properties> XML タグ・リファレンス, 7-66
<property> XML タグ・リファレンス, 7-67
<report> XML タグ・リファレンス, 7-70
<section> XML タグ・リファレンス, 7-72
<select> XML タグ・リファレンス, 7-74
<summary> XML タグ・リファレンス, 7-75
<tabular> XML タグ・リファレンス, 7-79

A

ActiveX リクエスト・メソッド, 4-2
Apache, 3-9, 3-10
Apache サーバー、「Oracle HTTP Server powered by Apache」を参照

C

CACHESIZE パラメータ, 4-3
CLUSTERCONFIG パラメータ, 6-2, 6-5
CURRENCY コマンドライン引数, 4-3
CUSTOMIZE
 キーワード, 7-25
 コマンドライン引数, 7-3

D

DECIMAL コマンドライン引数, 4-3
DESFORMAT
 コマンドライン引数, 4-3
DESTYPE
 コマンドライン引数, 4-2

E

ENGLIFE パラメータ, 6-5

M

MAXENGINE パラメータ, 6-2, 6-3
MAXIDLE パラメータ, 6-5
MINENGINE パラメータ, 6-2, 6-3
MODE コマンドライン引数, 4-3

O

- OAS (Oracle Application Server), F-1
- Oracle Application Server (OAS), F-1
- Oracle HTTP Server powered by Apache, 2-2
- Oracle Portal
 - コンポーネント・リクエスト・メソッド, 4-2
- Oracle Reports Services
 - UNIX でジョブのステータスを表示する, 3-3, 3-14
 - アーキテクチャ, 1-2
 - 構成
 - クラスタリングに向けた, 6-1
 - パラメータ, C-1
 - 層, 1-2
 - 重複ジョブの検出, 4-2
- Oracle Reports Services Queue Manager
 - 実行するジョブのスケジューリング, 4-8
 - ジョブのステータスの監視, 3-16
- Oracle Reports Services サーバー
 - 起動, 3-1
 - UNIX, 3-3, 3-12, 3-13
 - Windows NT, 3-2
 - Windows NT 上で非サービスとして, 3-3
 - 構成
 - UNIX での環境変数による, 3-13
 - Windows NT での環境変数による, 3-12
 - 環境変数による, 3-12
 - サービス・エントリの作成, 3-10
 - 停止, 3-1, 3-4
 - UNIX, 3-4
 - Windows NT, 3-4
 - データベース・キュー, 3-17
 - デフォルトを設定する, 3-11
 - UNIX, 3-11
 - Windows NT, 3-11
- Oracle Reports Services サーバーの起動, 3-1
 - UNIX, 3-3
 - Windows NT, 3-2
 - Windows NT (非サービス), 3-3
- Oracle Reports Services サーバーの停止, 3-1
 - UNIX, 3-4
 - Windows NT, 3-4
- Oracle Universal Installer, 2-1
- ORACLE_HOME/guicommon6/tk60/ADMIN ディレクトリ, 3-14
- ORACLE_HOME¥REPORT60¥SERVER ディレクトリ, 3-13

- ORACLE_HOME¥REPORT60 ディレクトリ, 4-4
- ORIENTATION コマンドライン引数, 4-3

P

- PAGESIZE コマンドライン引数, 4-3
- PARAMFORM コマンドライン引数, 4-3

R

- RDF
 - ファイル, 4-6
- REPORT
 - キーワード, 7-25
 - コマンドライン引数, 4-3, 7-3
- REPORTS_PATH 構成環境変数, 3-14
- REPORTS60_CGIMAP 環境変数, 4-4
- REPORTS60_COOKIE_EXPIRE 環境変数, 3-15
- REPORTS60_DB_AUTH 環境変数, 3-15
- REPORTS60_ENCRYPTION_KEY 環境変数, 3-15
- REPORTS60_PATH 環境変数, 3-12, 3-14, 4-6
- REPORTS60_REPORT_SERVER 環境変数, F-5
- REPORTS60_REPORTS_SERVER 環境変数, 3-11, 3-15, 3-16, 4-5, 4-7, 6-7
- REPORTS60_SSLPORT 環境変数, 3-15
- REPORTS60_SYS_AUTH 環境変数, 3-15
- REP ファイル, 4-6
- rw_server.sql スクリプト, 3-18
- RWCLI60
 - コマンド, 4-1, 4-2
 - コマンドライン引数, 4-5, 4-7, B-1
 - コマンドライン・リクエスト, 4-1
- RWRUN60 実行ファイル, 4-1

S

- SCHEDULE コマンドライン引数, 6-7
- SERVER
 - コマンドライン引数, 3-15, 3-16, 4-2, 4-5, 4-7, 6-7, F-5
- SOURCEDIR パラメータ, 3-12, 3-14, 4-6
- SRW.RUN_REPORT リクエスト・メソッド, 4-2
- SRW.RUN.REPORT コマンド, 4-2

T

THOUSANDS コマンドライン引数, 4-3
TNS_ADMIN
 環境変数, 3-12
 構成環境変数, 3-14
TOLERANCE コマンドライン引数, 4-2

U

uiprint.txt ファイル, 3-14
UNIX、Oracle Reports Services サーバー
 環境変数による構成, 3-13
 起動, 3-3
 停止, 3-4
 デフォルトを設定する, 3-11
URL 構文
 完全な構文の例, 4-7
 コマンドライン引数の隠蔽, 4-4
 単純化した構文の例, 4-7
 ハイパーリンクの追加, 4-7
 ブラウザから実行する方法, 4-7
 リクエストの単純化, 4-4
 レポート・リクエスト・メソッド, 4-2
USERID
 コマンドライン引数, 4-3

W

Web
 CGI, 1-5
 キー・マップ・ファイル, 4-4
 サーバー層、Oracle Reports Services, 1-2
 サーバー、構成, 3-9
Web カートリッジから CGI への移行, F-1
Windows NT、Oracle Reports Services サーバー
 環境変数による構成, 3-12
 起動, 3-2
 停止, 3-4
 デフォルトを設定する, 3-11
 非サービスとしての起動, 3-3

X

XML
 RWBLD60, 7-25
 TEXT_IO, 7-26

作成
 レポート定義, 7-2, 7-3
 レポート定義の必須タグ, 7-4
実行
 単独でのレポート定義, 7-22
 レポート定義, 7-17
 タグ・リファレンス, 7-27
適用
 PL/SQL でのレポート定義, 7-18
 実行時のレポート定義, 7-17
 ファイルに保存されたレポート定義, 7-19
 複数のレポート定義, 7-18
 メモリーに保存されたレポート定義, 7-19
トレース・オプション, 7-23
パーサーのエラー・メッセージ, 7-23
レポート定義, 7-1
 Oracle Reports Services Builder で表示する,
 7-25
 PL/SQL による適用, 7-18
 PL/SQL 例, 7-9
 概要, 7-2
 完全, 7-11
 実行, 7-22
 書式の例, 7-6
 書式例外の例, 7-8
 タグ, 7-27
 追加オブジェクト, 7-10
 適用, 7-17
 デバッグ, 7-23
 パーサー, 7-23
 ハイパーリンク例, 7-9
 バッチ変更, 7-22
 必須タグ, 7-4
 ファイルへの書込み, 7-26
 部分, 7-4
 レポート作成用, 7-5
 レポート定義の使用方法, 7-2
 レポート定義のデバッグ, 7-23
XML での適用
 PL/SQL でのレポート定義, 7-18
 実行時のレポート定義, 7-17
 ファイルに保存されたレポート定義, 7-19
 複数のレポート定義, 7-18
 メモリーに保存されたレポート定義, 7-19
XML でのバッチ変更の実行, 7-22
XML のトレース・オプション, 7-23
XML のパーサーのエラー・メッセージ, 7-23

あ

アーキテクチャ

- Oracle Reports Services , 1-2
- Oracle Reports Services 層 , 1-2
 - Web サーバー , 1-2
 - シン・クライアント , 1-2
 - データベース , 1-2
- Web サーバーの構成 , 1-3
- アクセス制御
 - 使用可能日カレンダー , A-1

い

インストール

- Oracle9i Application Server , 2-1
- UNIX 上での Oracle Reports Services サーバーの起動 , 3-3 , 3-12 , 3-13 , 3-14
- Windows NT で非サービスとして起動 , 3-3

か

概念 , 1-1

概要

- XML , 7-2
- クラスタリング , 6-2

カスタマイズ

- XML レポート定義 , 7-3
- 概要 , 7-2
- 実行時のレポート , 7-1

環境変数

- REPORTS60_CGIMAP , 4-4
- REPORTS60_COOKIE_EXPIRE , 3-15
- REPORTS60_DB_AUTH , 3-15
- REPORTS60_ENCRYPTION_KEY , 3-15
- REPORTS60_PATH , 3-12 , 3-14 , 4-6
- REPORTS60_REPORT_SERVER , F-5
- REPORTS60_REPORTS_SERVER , 3-11 , 3-15 , 3-16 , 4-5 , 4-7 , 6-7
- REPORTS60_SSLPORT , 3-15
- REPORTS60_SYS_AUTH , 3-15
- TNS_ADMIN , 3-12 , 3-14
- 構成 , 3-15
 - REPORTS_PATH , 3-14
 - TNS_ADMIN , 3-14

き

キー・マップ・ファイル

- cgicmd.dat CGI , 4-4
- URL パラメータのマッピング , 4-5
- エントリのマッピング , 4-5
- 使用 , 4-4
- 使用可能にする方法 , 4-4
- どのようなときに使用するか , 4-4
- 例 , 4-4 , 4-5

キーワード

- CUSTOMIZE , 7-25
- REPORT , 7-25

キャッシュ

- サイズ , C-1
- キュー・アクティビティ、データベース , 3-17
- 許容値 , B-22

く

クラスタリング

- 概要 , 6-2
- 構成
 - Oracle Reports Services , 6-3
 - Oracle Reports Services サーバー , 6-3
 - マスター・サーバー , 6-4
- 再送信 , 6-7
- マスターとスレーブ間の通信を可能にする方法 , 6-4
- レポートの実行 , 6-7

け

結合された使用可能日カレンダー , A-5

こ

構成

- Oracle Reports Services , 1-7
- Oracle Reports Services サーバー
 - UNIX での環境変数による , 3-13
 - Windows NT での環境変数による , 3-12
 - 環境変数による , 3-12
 - 変更 , 3-17
- Oracle Reports Services サーバー CGI , 3-8 , 3-10
- Oracle Reports Services サーバー・サブレット , 3-5

- JSDK を使用する, 3-5
- JServ を使用する, 3-7
- Oracle Reports Services サーバーのクラスタリング, 6-1, 6-3
- Web サーバー, 3-9
- マスター・サーバーのクラスタリング, 6-4

コマンド

- Oracle Reports Services Runtime, 4-2
- READONLY, B-8, B-12
- RWCLI60, 4-1, 4-2, 7-18, 7-22
- RWRUN60, 7-9, 7-18, 7-22, B-2, B-15
- SCHEDULE, B-21
- setenv, 3-11
- SHOWAUTH, B-2, B-27
- SRW.RUN.REPORT, 4-2
- ライン引数, 1-5

コマンドライン引数, B-1

- AUTHID, B-27
- CURRENCY, 4-3
- CUSTOMIZE, 7-3, 7-17, 7-18, 7-22
- DECIMAL, 4-3
- DESFORMAT, 4-3
- DESTYPE, 4-2, B-4
- MODE, 4-3
- ORIENTATION, 4-3
- PAGESIZE, 4-3
- PARAMFORM, 4-3
- REPORT, 4-3, 7-3, 7-17
- RWCLI60, 4-5, 4-7, B-1
- SCHEDULE, 6-7
- SERVER, 3-15, 3-16, 4-2, 4-5, 4-7, 6-7, B-20, D-2, F-5
- THOUSANDS, 4-3
- TOLERANCE, 4-2
- URL パラメータのマッピング, 4-4
- USERID, 4-3, B-2, E-1

さ

- サブレット, 1-5

作成

- Oracle Reports Services サーバーのサービス・エン
トリ, 3-10
- XML
 - レポート定義, 7-2, 7-3
 - レポート定義の必須タグ, 7-4
- 結合された使用可能日カレンダー, A-5

- 使用可能日カレンダー, A-1
- デイリー・カレンダー, A-2
- メンテナンス・カレンダー, A-3

し

実行

- Web ブラウザからのレポート・リクエスト, 3-16
- XML
 - 単独でのレポート定義, 7-22
 - レポート定義, 7-17
 - クラスタ構成でのレポート, 6-7

実行時

- カスタマイズ
 - XML レポート定義, 7-3
 - 概要, 7-2

- 使用、XML レポート定義, 7-2

- 使用可能日カレンダー, A-1

- ジョブの再送信, 6-7

処理

- Web レポート, 1-5
- レポート, 1-6

- シン・クライアント層、Oracle Reports Services, 1-2

す

- スクリプト、rw_server.sql, 3-18

せ

- セキュリティ, 5-1

設定

- デフォルトの Oracle Reports Services サーバー, 3-11
- UNIX, 3-11
- Windows NT, 3-11

た

タグ

- リファレンス、XML, 7-27
- レポート定義用 XML, 7-27

ち

重複ジョブの検出

- Oracle Reports Services の扱い, 4-2, B-22
- 複数の出力先, 4-3

て

デイリー・カレンダー, A-2

ディレクトリ

- ORACLE_HOME/guicommon6/tk60/ADMIN, 3-14
- ORACLE_HOME¥REPORT60, 4-4
- ORACLE_HOME¥REPORT60¥SERVER, 3-13

データベース層、Oracle Reports Services, 1-2

デバッグ

- XML レポート定義, 7-23
- トレース・オプション, 7-23

デフォルトのプリンタ、アクセス設定, 3-3, 3-14

に

認証 cookie

- 期限切れ, B-27

は

バッチ

レポート

- Oracle Portal サイトからの, 4-8
- レポートの変更, 7-22

パラメータ

- CACHESIZE, 4-3
- CLUSTERCONFIG, 6-2, 6-5
- ENGLIFE, 6-5
- MAXENGINE, 6-2, 6-3
- MAXIDLE, 6-5
- MINENGINE, 6-2, 6-3
- Oracle Reports Services の構成, C-1
- RWCLI60 コマンドライン引数, B-1
- SOURCEDIR, 3-12, 3-14, 4-6

ふ

ファイル

- RDF, 4-6
- REP, 4-6

RWRUN60 実行ファイル, 4-1

- uiprint.txt, 3-14

付加

- 他のスレーブ・サーバーをマスターに, 6-8

め

メンテナンス・カレンダー, A-3

れ

例

```
<!-- --> XML タグ・リファレンス, 7-27
<![CDATA[]]> XML タグ・リファレンス, 7-28
<condition> XML タグ・リファレンス, 7-31
<customize> XML タグ・リファレンス, 7-32
<data> XML タグ・リファレンス, 7-33
<dataSource> XML タグ・リファレンス, 7-34
<exception> XML タグ・リファレンス, 7-37
<field> XML タグ・リファレンス, 7-41
<formLike> XML タグ・リファレンス, 7-44
<formula> XML タグ・リファレンス, 7-42
<function> XML タグ・リファレンス, 7-45
<group> XML タグ・リファレンス, 7-47
<groupAbove> XML タグ・リファレンス, 7-48
<groupLeft> XML タグ・リファレンス, 7-49
<labelAttribute> XML タグ・リファレンス, 7-52
<layout> XML タグ・リファレンス, 7-52
<link> XML タグ・リファレンス, 7-57
<matrix> XML タグ・リファレンス, 7-58
<matrixCell> XML タグ・リファレンス, 7-60
<matrixCol> XML タグ・リファレンス, 7-61
<matrixRow> XML タグ・リファレンス, 7-62
<object> XML タグ・リファレンス, 7-64
<programUnits> XML タグ・リファレンス, 7-65
<properties> XML タグ・リファレンス, 7-66
<property> XML タグ・リファレンス, 7-69
<report> XML タグ・リファレンス, 7-70
<section> XML タグ・リファレンス, 7-73
<select> XML タグ・リファレンス, 7-74
<summary> XML タグ・リファレンス, 7-78
<tabular> XML タグ・リファレンス, 7-80
RWCLI60 コマンドライン・リクエスト, 4-1
XML レポート定義
  PL/SQL, 7-9
  完全, 7-11
  書式, 7-6
```

- 書式例外, 7-8
- 追加オブジェクト, 7-10
- ハイパーリンク, 7-9
- 完全な URL 構文, 4-7
- キー・マッピング, 4-4, 4-5
- 単純化した URL 構文, 4-7
- レジストリのエントリ, D-1
- レポート定義、XML, 7-1
- レポートのソース・パスの設定, 3-12
- レポート・リクエスト
 - サーバーがクラスタ化されている場合の, 6-7
 - 重複ジョブの検出, 4-2
 - ブラウザから実行する方法, 4-7
- メソッド
 - ActiveX, 4-2
 - Oracle Portal コンポーネント, 4-2
 - RWCLI60 コマンドライン, 4-1
 - SRW.RUN_REPORT, 4-2
 - URL 構文, 4-2
- リクエストの指定, 4-6
- レポートの構築, 4-6
- レポート・リクエストの指定, 4-6
 - Web ブラウザから, 4-7
 - 自動実行のためのスケジューリングによる, 4-8
 - レポートの構築による, 4-6

ろ

- ロード・バランス
 - ジョブの再送信, 6-7
 - レポートの実行, 6-7

