

# Oracle Reports Developer

パブリッシング・レポート

リリース 6i

2000 年 4 月

部品番号 : J00618-01

ORACLE®

---

Oracle Reports Developer パブリッシング・レポート , リリース 6i

部品番号 : J00618-01

原本名 : Oracle Reports Developer Release 6i: Publishing Reports

原本部品番号 : A73173-01

原本著者 : Frank Rovitto

原本協力者 : Chan Fonseka, Shaun Lin, Paul Narth, Ashok Natesan, Danny Richardson

Copyright © 1996, 1999, 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 .....	xi
Reports Server の新機能 .....	xi
対象読者 .....	xii
このマニュアルの構成 .....	xii
関連ドキュメント .....	xii
表記規則 .....	xiii

## 第 I 部   パブリッシング・レポート

### 1   概要

1.1   Oracle Reports Server .....	1
1.1.1   Oracle Reports Server .....	2

### 2   公開のアーキテクチャと概念

2.1   Oracle Reports Server .....	3
2.2   Reports Server のアーキテクチャ .....	4
2.2.1   Web アーキテクチャ：サーバーの構成 .....	5
2.2.1.1   Web レポートの処理 .....	6
2.2.2   非 Web アーキテクチャ：サーバーの構成 .....	7
2.2.2.1   レポートの処理 .....	8
2.3   Reports Server の構成方法 .....	9
2.3.1   使用可能な Web リクエストおよび非 Web リクエスト .....	9
2.3.2   Reports Web CGI またはサープレットの選択 .....	9
2.3.3   Reports Server のロケーションの選択 .....	10

### 3 Reports Server の構成

3.1	インストールレーションと構成の概要 .....	11
3.1.1	Reports Server の自動構成 .....	11
3.1.1.1	ステップ 1: Oracle Installer の起動 .....	11
3.1.1.2	ステップ 2: Oracle インストール設定の選択 .....	12
3.1.1.3	ステップ 3: Oracle Tools インストール・オプションの選択 .....	12
3.1.1.4	ステップ 4: インストール・オプションの選択 .....	12
3.1.1.4.1	標準インストール .....	12
3.1.1.4.2	カスタム・インストール .....	14
3.1.2	Reports Server の手動での構成 .....	14
3.2	Reports Server の構成 : Reports Web CGI の例 .....	14
3.2.1	ステップ 1. ソフトウェアのインストール .....	15
3.2.1.1	Web サーバーでの作業 .....	15
3.2.1.2	Reports Server マシンでの作業 .....	15
3.2.2	ステップ 2. Web サーバーの構成 .....	16
3.2.3	ステップ 3. Reports Web CGI の構成 .....	16
3.2.3.1	Reports Web CGI を構成する手順 .....	16
3.2.3.2	Reports Server のサービス・エントリを作成する手順 .....	17
3.2.3.3	(オプション) デフォルトの Reports Server を設定する手順 .....	17
3.2.3.3.1	Windows NT プラットフォーム .....	17
3.2.3.3.2	UNIX プラットフォーム .....	18
3.2.4	ステップ 4. Reports Server の構成 .....	18
3.2.4.1	Windows NT で Reports Server を構成する手順 .....	18
3.2.4.1.1	(オプション) ソース・パスの設定 .....	18
3.2.4.1.2	Reports Server の起動 .....	19
3.2.4.2	UNIX で Reports Server を構成する手順 .....	19
3.2.4.2.1	(オプション) 環境変数の設定 .....	19
3.2.4.2.2	Reports Server の起動 .....	20
3.2.5	ステップ 5. Web ブラウザからのレポートの実行 .....	21
3.2.5.1	次のステップ .....	21
3.2.6	ステップ 6. (オプション) Reports Server の構成の変更 .....	22
3.2.6.1	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 .....	22
3.2.6.1.1	Reports Server マシン上 : .....	22
3.3	コマンド・ラインからの Reports Server の起動と停止 .....	23
3.3.1	別の Reports Server の起動 .....	24
3.3.1.1	Reports Server のサービス・エントリを作成する手順 .....	24
3.3.1.2	Windows NT で Reports Server を構成する手順 .....	25

3.3.1.2.1	(オプション) ソース・パスの設定 .....	25
3.3.1.2.2	Reports Server の起動 .....	25
3.3.1.3	UNIX で Reports Server を構成する手順 .....	26
3.3.1.3.1	(オプション) 環境変数の設定 .....	26
3.3.1.3.2	Reports Server の起動 .....	27
3.3.2	Windows NT における Reports Server の非サービスとしての起動 .....	28
3.3.3	Reports Server の停止 .....	28
3.3.3.1	Windows NT の場合 .....	28
3.3.3.2	UNIX の場合 .....	28
3.4	Reports サブレットの構成ガイドライン .....	28
3.4.1	Web サーバーの構成 .....	29

## 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 .....	31
4.2	重複ジョブの検出 .....	32
4.3	キー・マップ・ファイルの使用方法 .....	33
4.3.1	キー・マッピングを使用可能にする方法 .....	34
4.3.2	URL パラメータのマッピング .....	34
4.4	レポート・リクエストの指定 .....	35
4.4.1	レポートの構築 .....	35
4.4.2	Web ブラウザからのレポート・リクエストの指定 .....	36
4.4.3	自動実行のためのレポートのスケジューリング .....	37

## 5 レポートへのユーザー・アクセスの管理

5.1	アクセス・コントロールの構成およびセットアップの概要 .....	40
5.2	WebDB の Reports Security のインストールと構成 .....	41
5.2.1	ステップ 1. WebDB の Reports Security を構成する .....	41
5.2.1.1	WebDB をインストールする手順 .....	41
5.2.1.2	Reports Server セキュリティ・パッケージを WebDB にインストールする手順 .....	41
5.2.1.3	認証 cookie ドメインを設定する手順 .....	42
5.3	WebDB のアクセス・コントロールのセットアップ .....	43
5.3.1	ステップ 1. アクセス・コントロールに対応した Reports Server を構成する .....	44
5.3.1.1	WebDB に接続する TNS 名の別名を作成する手順 .....	44
5.3.1.2	Reports Server へのアクセスの制限 .....	45
5.3.2	ステップ 2. ユーザー・アカウントを作成する .....	46
5.3.2.1	Reports Developer システム管理者のユーザー・アカウントを作成する手順 .....	46

5.3.2.2	レポートの実行のためのユーザー・アカウントを作成する手順 .....	47
5.3.3	ステップ 3. 使用可能日カレンダーを作成する .....	48
5.3.3.1	デイリー・カレンダーを作成する手順 .....	48
5.3.3.2	メンテナンス・カレンダーを作成する手順 .....	50
5.3.3.3	クリスマス・カレンダーを作成する手順 .....	50
5.3.3.4	複合型の使用可能日カレンダーを作成する手順 .....	51
5.3.4	ステップ 4. WebDB に Reports Developer プリンタへのアクセスを追加する .....	52
5.3.5	ステップ 5. WebDB に Reports Server へのアクセスを追加する .....	53
5.3.6	ステップ 6. WebDB に Reports Server へのアクセスを追加する .....	54
5.3.6.1	Lastname ユーザー・パラメータの値リストを作成する手順 .....	55
5.3.6.2	レポート定義ファイルへのアクセスを追加する手順 .....	56
5.3.7	ステップ 7. 「Reports Developer パラメータ・フォーム」でパラメータ値を設定する ....	59
5.3.7.1	レポート出力をキャッシュに保存する手順 .....	59
5.3.7.2	(オプション) レポート出力をアクセス制限されたプリンタに送る手順 .....	60
5.3.7.3	実行時にユーザーが利用するデフォルト・パラメータを設定する手順 .....	61
5.3.8	ステップ 8. ユーザーがレポートを利用できるようにする .....	62
5.3.8.1	WebDB サイトを作成する手順 .....	62
5.3.8.2	WebDB サイトにフォルダを作成する手順 .....	62
5.3.8.3	レポート・リクエストをフォルダに追加する手順 .....	64
5.3.8.4	ユーザーの立場からこのレポートを実行する手順 .....	64
5.3.9	ステップ 9. レポートを実行して出力を WebDB サイトに 配信するスケジュールを設定する .....	65
5.3.9.1	個人フォルダを作成する手順 .....	66
5.3.9.2	レポートをスケジュールする手順 .....	66
5.3.9.3	配信されたレポート出力を表示する手順 .....	68
5.3.10	まとめ .....	68

## 6 Reports Server のクラスタの構成

6.1	クラスタリングの概要 .....	71
6.2	Reports Server をクラスタに構成する例 .....	72
6.2.1	マスターとスレーブ間の通信を可能にする方法 .....	73
6.2.2	マスター・サーバーを構成する方法 .....	74
6.2.3	クラスタ構成でレポートを実行する方法 .....	76
6.2.4	エンジンがダウンしたときにジョブの再送信を行う方法 .....	76
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 .....	77

## 7 実行時のレポートのカスタマイズ

7.1	概要 .....	79
7.1.1	XML レポート定義の作成および使用方法 .....	81
7.2	XML レポート定義の作成 .....	82
7.2.1	必須タグ .....	82
7.2.2	部分レポート定義 .....	83
7.2.2.1	書式変更の例 .....	85
7.2.2.2	書式例外の例 .....	87
7.2.2.3	プログラム単位とハイパーリンクの例 .....	88
7.2.2.4	データ・モデルと書式の変更の例 .....	89
7.2.3	完全なレポート定義 .....	90
7.3	XML レポート定義の実行 .....	95
7.3.1	実行時の XML レポート定義の適用 .....	96
7.3.1.1	1 つの XML レポート定義の適用 .....	96
7.3.1.2	複数の XML レポート定義の適用 .....	96
7.3.1.3	PL/SQL での XML レポート定義の適用 .....	97
7.3.1.3.1	ファイルに保存された XML 定義の適用 .....	97
7.3.1.3.2	メモリーに保存された XML 定義の適用 .....	97
7.3.2	XML レポート定義を単独で実行する .....	100
7.3.3	バッチ変更の実行 .....	100
7.4	XML レポート定義のデバッグ .....	101
7.4.1	XML パーサーのエラー・メッセージ .....	102
7.4.2	トレース・オプション .....	102
7.4.3	RWBLD60 .....	104
7.4.4	TEXT_IO .....	104
7.5	XML タグ・リファレンス .....	105
7.5.1	<!-- コメント --> .....	105
7.5.2	<![CDATA[ ]]> .....	106
7.5.3	<condition> .....	107
7.5.4	<customize> .....	108
7.5.5	<data> .....	109
7.5.6	<dataSource> .....	110
7.5.7	<exception> .....	111
7.5.8	<field> .....	113
7.5.9	<formLike> .....	118
7.5.10	<formula> .....	118
7.5.11	<function> .....	120

7.5.12	<group> .....	121
7.5.13	<groupAbove> .....	122
7.5.14	<groupLeft> .....	123
7.5.15	<labelAttribute> .....	124
7.5.16	<layout> .....	126
7.5.17	<link> .....	129
7.5.18	<matrix> .....	131
7.5.19	<matrixCell> .....	133
7.5.20	<matrixCol> .....	134
7.5.21	<matrixRow> .....	135
7.5.22	<object> .....	135
7.5.23	<programUnits> .....	137
7.5.24	<properties> .....	138
7.5.25	<property> .....	139
7.5.26	<report> .....	141
7.5.27	<section> .....	143
7.5.28	<select> .....	145
7.5.29	<summary> .....	146
7.5.30	<tabular> .....	150

## 第 II 部 付録

### A RWCLI60 コマンド・ライン引数

### B Reports Server 構成パラメータ

### C 環境変数

### D データベース接続文字列

### E Reports Web カートリッジ

E.1	Reports Web カートリッジの例 .....	197
E.1.1	ステップ 1. ソフトウェアをインストールする .....	197
E.1.2	ステップ 2. Web サーバーを構成する .....	198
E.1.3	ステップ 3. Reports Web カートリッジを構成する .....	198
E.1.3.1	Reports Server のサービス・エントリを作成する手順 .....	199



E.1.3.2	デフォルトの Reports Server を構成する .....	199
E.1.3.3	Reports Web カートリッジを構成する ( Windows NT および UNIX ) .....	200
E.1.4	ステップ 4. Reports Server を構成する .....	201
E.1.4.1	Windows NT 上で Reports Server を構成する手順 .....	202
E.1.4.1.1	( オプション ) ソース・パスを設定する .....	202
E.1.4.2	Windows NT 上で Reports Server を構成する手順 .....	202
E.1.4.2.1	( オプション ) ソース・パスを設定する .....	202
E.1.4.2.2	Reports Server をサービスとして Windows NT にインストールする .....	202
E.1.4.2.3	Reports Server を非サービスとして Windows NT に構成する .....	203
E.1.4.2.4	Windows NT 上の Reports Server をアンインストールする .....	204
E.1.4.3	Reports Server を UNIX 上で構成する手順 .....	204
E.1.4.3.1	( オプション ) 環境変数を設定する .....	204
E.1.4.3.2	Reports Server を起動する .....	205
E.1.5	ステップ 5. レポート・リクエストを実行する .....	205
E.1.6	ステップ 6. ( オプション ) 構成を修正する .....	206
E.1.6.1	データベースのジョブのキュー・アクティビティを更新する .....	206
E.2	環境変数 .....	207

## F   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、Oracle Reports Developer を使用したレポートの公開用に用意されている各種のオプションについて説明します。また、レポートを公開するための Reports Server ソフトウェアの構成方法についても触れます。

## Reports Server の新機能

新機能	下記参照
<b>データベース・キュー</b> : Reports Server キュー・アクティビティを使用してデータベースを更新します。	3.2.6.1 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」
<b>Reports サブレット</b> : Java が使用可能な Web サーバーを使用して Reports Server を構成します。	3.4 項「Reports サブレットの構成ガイドライン」
<b>レポートへのアクセス権限を管理</b> : Reports Server 上で実行するレポートへのアクセス権限を限定。Reports Developer では、ユーザーが、制限された Reports Server 上でレポートを実行するアクセス権限を持っているかどうかを Oracle WebDB を使用してチェックします。	第 5 章「レポートへのユーザー・アクセスの管理」
<b>クラスタリング</b> : Reports Server をクラスタ化すると、パフォーマンスが向上するとともに、負荷を分散させることができます。	第 6 章「Reports Server のクラスタの構成」
<b>レポートのカスタマイズ</b> : オリジナルのレポートを変更せずに実行時にレポートのカスタマイズを適用し、利用対象者によって異なる各種の出力を生成します。	第 7 章「実行時のレポートのカスタマイズ」

## 対象読者

このマニュアルは Oracle Reports Developer 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 Developer 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（例：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

このマニュアルは次の章から構成されています。

- |       |   |
|-------|---|
| 第 1 章 | Reports Server を紹介します。  |
| 第 2 章 | Reports Server 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 3 章 | CGI 関連の Web サーバーでの構成例を使用して Reports Server を構成する方法を説明します。                |
| 第 4 章 | Reports Server 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 | Reports Server を Oracle WebDB に組み込み、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 Reports Server 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 WebDB または Oracle Reports Server の詳細について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』 J00918-01
- 『Oracle Reports Developer for Windows スタート・ガイド』 J00454-01
- 『Oracle WebDB スタート・ガイド』 J00111-01
- 『Forms アプリケーション Web 利用ガイド』 J00496-01

# 表記規則

このマニュアルでは、次のような表記上の規則を使用しています。

規則	意味
太字	強調を目的として使用。メニュー項目、ボタン名、ラベルなどユーザー・インタフェース要素にも使用。
イタリック体	初めて現われる用語の紹介に使用。
クーリエ・フォント	パスおよびファイル名、入力するコートおよびテキストに使用。
大文字のクーリエ	次の場合に使用。 <ul style="list-style-type: none"><li>■ ファイル拡張子（.PLL または .FMX）</li><li>■ SQL コマンド</li></ul>
CAPS	次の場合に使用。 <ul style="list-style-type: none"><li>■ 環境変数</li><li>■ ビルトイン / パッケージ名</li><li>■ 実行可能ファイル名</li></ul>



# 第Ⅰ部

---

## パブリッシング・レポート

第1章「概要」

第2章「公開のアーキテクチャと概念」

第3章「Reports Server の構成」

第4章「レポート・リクエストの実行」

第5章「レポートへのユーザー・アクセスの管理」

第6章「Reports Server 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」





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。Reports Developer 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。

Reports Developer 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の作成のために 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。

Reports Developer のサーバーベースのアーキテクチャでは、レポートのコンシューマ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 Web ブラウザは 1 つのみで十分です。Oracle Reports Server は、カスケード・スタイル・シートおよび Adobe の Portable Document Format (PDF) を持つ HTML 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 Server

Reports Server 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することができます。また、このサーバーによって、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。

Reports Web CGI または Reports サブレットと組み合わせて使用することで、Reports Server 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。Reports Server は、Windows NT、Windows 95 または Unix 上にインストールすることができます。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・エンジンを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。

Reports Server は、定義済み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共に、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 Reports 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。Reports Queue Manager は、Reports Server と同一のマシン上に存在する場合も、クライアントのマシン上に存在する場合もあります。UNIX では、Reports Queue Viewer (RWRQV60) を使用すると Reports Server のキューを参照することができます。

### 1.1.1 Oracle Reports Server

Reports Server は、社内のイントラネット、社外のエクストラネットまたはインターネットでの新規または既存の Oracle Reports Developer レポートの配置を可能にする新しいタイプの開発ツールである、Oracle Reports Server のコンポーネントの 1 つです。

Oracle Reports Server には、Reports Server だけでなく Graphics Server も含まれます。Graphics Server は Oracle グラフィックの配置を支援します。グラフィックは、HTML に埋め込んだり、Reports Developer レポートに取り込んだりできる Graphic Interchange Format (.GIF) ファイルとして配置されます。

Oracle Reports Server は、複数の階層化された環境に Oracle Reports Developer アプリケーション (Reports および Graphics) を配置するために最適化したアプリケーション・サーバーです。これによって Web の使用やアクセスが簡単になり、Web に単なる静的な情報公開メカニズムを超えた、複雑で動的なアプリケーションをサポートできる環境としての機能を与えます。

Forms Server および Graphics Server の詳細は、『Forms Server for Windows and UNIX Forms アプリケーション Web 利用ガイド リリース 6i』マニュアルを参照してください。

---

## 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 Developer 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 Developer 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の作成のために 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 Developer のサーバーベースのアーキテクチャでは、レポートのコンシューマ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 Web ブラウザは 1 つのみで十分です。Oracle Reports Server は、カスケード・スタイル・シートおよび Adobe の Portable Document Format (PDF) を持つ HTML 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。

### 2.1 Oracle Reports Server

Reports Server 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することができます。また、このサーバーによって、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ができます。

Reports Web CGI または Reports サブレットと組み合わせて使用することで、Reports Server 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。Reports Server は、Windows NT、Windows 95 または Unix 上にインストールすることができます。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・エンジンを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。

Reports Server は、定義済み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共に、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 Reports 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。Reports Queue Manager は、Reports Server と同一のマシン上に存在する場合も、クライアントのマシン上に存在する場合もあります。UNIX では、Reports Queue Viewer (RWRQV60) を使用すると Reports Server のキューを参照することができます。

## 2.2 Reports Server のアーキテクチャ

Reports Server 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、Reports Server のアーキテクチャは次の 4 つの階層で構成されます。<sup>1</sup>

- シン・クライアント層
- Web サーバー層
- Reports Server 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、さまざまです。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）、Web サーバー層が不要なため、3 つの階層しか存在しません。

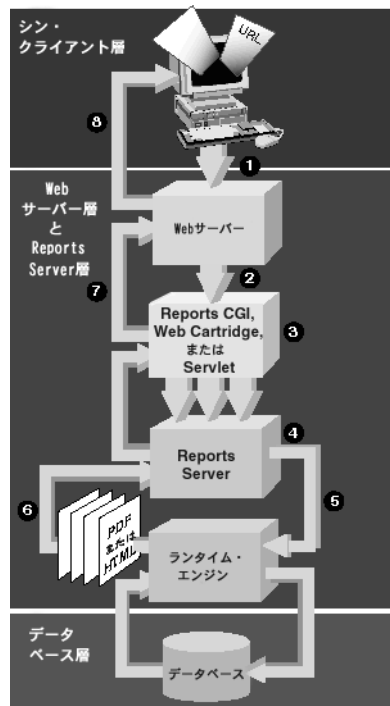
---

<sup>1</sup> 階層という言葉は、Reports Server 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。

## 2.2.1 Web アーキテクチャ：サーバーの構成

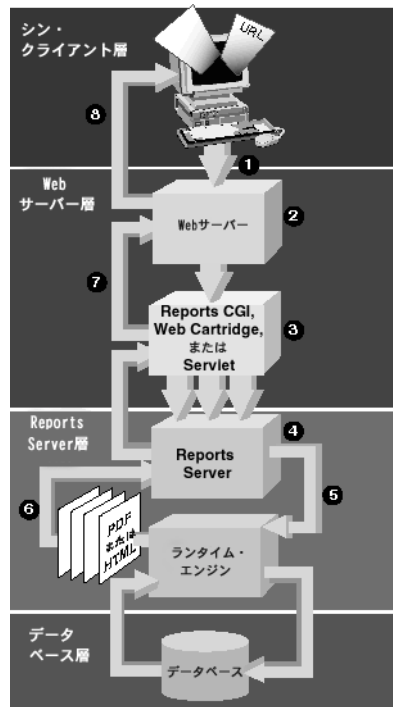
次に示す図は、Web 環境における Reports Server の、2 つの最も一般的な構成方法を示しています。2 つの構成の大きな相違点は、Reports Server と Web サーバーの階層が同じマシン上に存在するか、異なるマシン上に存在するかという点です。最初の構成の場合、Web サーバーと Reports Server は同一のマシン上に存在しています。2 つ目の構成の場合は、それらが異なるマシン上に存在しています。後者の場合、前者の場合と若干異なるセットアップが必要となります。

図 2-1 Web アーキテクチャ、3 つのマシンを使用した構成<sup>1</sup>



<sup>1</sup> Reports Web カートリッジは、Oracle Application Server を Web サーバーとして使用している場合に Reports Server と通信する別の手段です。

図 2-2 Web アーキテクチャ、4 つのマシンを使った構成<sup>1</sup>



### 2.2.1.1 Web レポートの処理

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 Reports Web CGI または Reports サープレット のどちらか、使用している方をコールします。
3. Reports Web CGI またはサープレットは、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。Reports CGI またはサープレットは、リクエストを Reports Server が実行可能なコマンド・ラインに変換し、指定された Reports Server に送信します。

<sup>1</sup> Reports Web カートリッジは、Oracle Application Server を Web サーバーとして使用している場合に Reports Server と通信する別の手段です。

4. リクエストに許容時間が含まれる場合<sup>1</sup>、Reports Server 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. Reports Server 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、<sup>2</sup> 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. Reports Web CGI またはサーブレットは、レポート出力を Reports Server 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。

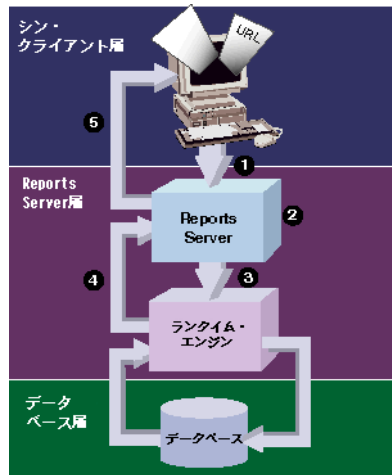
## 2.2.2 非 Web アーキテクチャ：サーバーの構成

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、Reports 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、Reports Server に送られます。非 Web アーキテクチャは、何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。

<sup>1</sup> Reports Server 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。例えば、要求者が TOLERANCE として 5 分を設定した場合、Reports Server はキャッシュをチェックして過去 5 分以内に作成された複製レポート出力を探します。

<sup>2</sup> Reports Server を構成する際に、使用可能な最大ランタイム・エンジン数を指定することができます。Reports Server 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。

図 2-3 非 Web アーキテクチャ



### 2.2.2.1 レポートの処理

1. クライアントは、コマンド・ライン (RWCLI60)、Reports Queue Manager または Reports 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. Reports Server 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. Reports Server にジョブが完了したことが伝えられます。
5. Reports Server 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・ラインの宛先タイプ (DESTYPE) にローカルファイル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。



## 2.3 Reports Server の構成方法

Reports Server の構成方法は、システムの要求事項によってさまざまです。Reports Server 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。

### 2.3.1 使用可能な Web リクエストおよび非 Web リクエスト

2.2 項「Reports Server のアーキテクチャ」にあったように、Reports Server 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・リクエストを受け取ることができます。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戻したり、指定した宛先（例：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、Reports Launcher、Reports Queue Manager、RWCLI60 から成る Reports 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、Reports Web CGI またはサーブレットを Web サーバーにインストールして、Web サーバーと Reports Server 間で通信が行われるようにする必要があります。Web CGI またはサーブレットは、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、2.3.2 項「Reports Web CGI またはサーブレットの選択」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 Reports 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（Reports Web CGI またはサーブレット）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、Reports Server 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。Reports Server 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。

### 2.3.2 Reports Web CGI またはサーブレットの選択

2.3.1 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 Reports Server を使用するには、Web サーバーと Reports Server 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、Reports Web CGI またはサーブレットをインストール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 Oracle Application Server、WebDB Listener）を使用している場合は、Reports Web CGI を選ぶ。
- Java ベースの Web サーバーを使用している場合は、Reports サーブレットを選ぶ。

### 2.3.3 Reports Server のロケーションの選択

2.2 項「Reports Server のアーキテクチャ」で説明しているように、Reports Server 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。

- Reports Server 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・リクエスト（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。
- Reports Server 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することができます。Reports CGI またはサーブレットは、Web サーバーとして機能する同一マシン上に存在する必要があります。Reports Server を異なるマシン上に設ける場合、Reports CGI およびサーブ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。

構成方法を決定したら、Reports Server の構成を行うことができます。3.2 項「Reports Server の構成 : Reports Web CGI の例」に、Web CGI を使用した構成例が手順を追って説明されています。

また、3.4 項「Reports サーブレットの構成ガイドライン」に、Reports サーブレットを使用した Reports Server の構成ガイドラインが示されています。

---

## Reports Server の構成

この章では、Oracle Reports Server を構成する方法について説明します。まず一般的な説明から始め、次に、Web CGI を使用した Reports Server の構成例を提示します。この例では、ユーザーはレポートを動的に実行することができ、レポート出力を Web ブラウザで受け取ることができます。レポートの実行にクライアント・ソフトウェアは不要です。

### 3.1 インストールと構成の概要

Reports Server のインストールと構成は、この章で説明するステップ・バイ・ステップの指示に従って自動または手動で行えます。そのためには、Oracle Reports Server ソフトウェアが必要です。

Reports Server のインストールと構成を行う方法は次のとおりです。

- Reports Server の自動インストールおよび自動構成。自動インストールでは、構成作業のほとんどが自動実行され、可能な限り速やかに設定および実行が行われます。
- Reports Server の手動インストールおよび手動構成。Reports Server の構成に習熟していて、構成プロセスをより詳細に制御したい場合、Reports Server を手動で構成できます。詳細は、3.1.2 項「Reports Server の手動での構成」を参照してください。

#### 3.1.1 Reports Server の自動構成

自動インストールでは、マシンへの依存度が分析された後、選択したオプションに基づいて Reports Server が構成されます。

##### 3.1.1.1 ステップ 1: Oracle Installer の起動

自動インストールを開始するには、次のようにします。

自動インストールの最後に、設定された構成設定値をまとめた構成の説明ファイルが作成されます。この構成の説明ファイルでは、手動で定義する際に必要となるその他の構成設定値についても説明しています。

### 3.1.1.2 ステップ 2: Oracle インストール設定の選択

最初に表示される画面は、「Oracle インストール設定」画面です。この画面では、Oracle のホーム・ディレクトリについて次のように設定します。

- 「名前」フィールドは DEFAULT\_HOME に設定されています。このフィールドを変更する必要はありません。
- 「場所」フィールドは、ユーザーが設定します。例えば、次のように設定します。

D:\orant

以前に Reports Server をインストールした場合、このフィールドはすでに設定されています。

- 「言語」フィールドは「Japanese」に設定されています。このフィールドは、必要に応じて変更できます。

### 3.1.1.3 ステップ 3: Oracle Tools インストール・オプションの選択

この画面では、次の製品を選択できます。ここでは、Oracle Reports Server を選択してください。

- Oracle Forms Developer
- Oracle Forms Server
- Oracle Reports Developer
- Oracle Reports Server

### 3.1.1.4 ステップ 4: インストール・オプションの選択

この画面では、標準インストールまたはカスタム・インストールのいずれかを選択できます。

**注意:** 自動インストールでは、インストール作業時に「概要」画面が表示されることがあります。「OK」を選択し、インストールを続行してください。また、例としての変数 username-pc.jp.oracle.com は、インストール環境の適切な値と置き換えてください。

**3.1.1.4.1 標準インストール** 標準インストールでは、次の画面が表示されます。

**注意:** 「コントロール パネル」を開いている場合は、インストールを続行する前に閉じてください。閉じていない場合、システムに wdbctrl.cpl をコピーできません。

#### Server:Web リスナー

この画面では、「Oracle WebDB Listener の使用」および「その他の Web リスナーの使用」という 2 つのオプションが提供されます。デフォルトは Oracle WebDB Listener です。

#### プロセスの起動

この画面では、WebDB Listener と Oracle Reports Server を自動的に設定および作成できます。「はい」を選択すると、これらのサービスをユーザーに代わって自動的に設定します。「いいえ」を選択すると、関連する `tnsnames.ora` エントリの作成のみが行われます。以降の画面では、このエントリを作成するための情報を入力します。そのため、「はい」、「いいえ」のいずれを選択しても表示されます。

### WebDB Listener

この画面では、WebDB Listener 名およびホスト名を入力します。たとえば、WebDB Listener の名前として、インストール先 PC の名前である `username-pc.jp.oracle.com` を入力します。PC 名は、「コントロール パネル」の「ネットワーク」に表示されています。次に、WebDB Listener のポート番号として 80 を入力します。ポート番号は、データベースで使用されているものと同一である必要があります。デフォルトは 80 です。

「キャンセル」をクリックすると、この設定作業はスキップされ、依存性の分析が開始されます。Oracle Installer により「Reports Server パラメータ」画面が表示されます。

### Reports Server パラメータ

この画面では、Reports Server の TNS 名（たとえば `Rep60_username-pc`）およびポート番号（たとえば 1949）を指定します。このポート番号は、WebDB Listener のポート番号とは別のものにする必要があります。

「ディスク領域が不足しています」というメッセージが表示されたら、ディスクの空き領域を増やして、必要な容量を確保する必要があります。必要領域を確保した後にのみ、先の作業に進むことができます。

### Reports Server の構成

この画面には、構成ファイルが表示されます。

### Oracle WebDB Listener

この画面には、追加したリスナーに変更を加えるための 2 ファイルの格納場所が表示されます。例：

```
http://username-pc.jp.oracle.com:80/WebDB/admin_/gateway.htm
```

```
http://username-pc.jp.oracle.com:80/WebDB/admin_/listener.htm
```

### 概要

この「概要」画面は最後に表示され、インストールが正常に終了したことを通知します。

WebDB に正しくアクセスできることを確認するには、ブラウザを開いて次のように入力します。

```
http://username-pc.jp.oracle.com:80/dev60cgi/Rwgci60.exe
```

ここで、`username-pc.jp.oracle.com` は、インストール時に入力した Web リスナーのホスト名です。

**3.1.1.4.2 カスタム・インストール**「インストール・オプション」画面で「カスタム」を選択すると、「Software Asset Manager」画面が表示されます。この画面では、製品の手動アンインストールと、別の製品のインストールが行えます。この画面での作業を正常に終了した後は、標準インストールと同様の手順で「Server:Web リスナー」画面以降のインストール作業を続行できます。

## 3.1.2 Reports Server の手動での構成

Reports Server を手動でインストールし構成するには、次のステップに従ってください。

ステップ 1. ソフトウェアをインストールする。

ステップ 2. Web サーバーを構成する。詳細は、使用している Web サーバーのマニュアルを参照してください。

ステップ 3. Reports Web CGI または Reports サブレットを構成する。

ステップ 4. Reports Server を構成する。

ステップ 5. レポート・リクエストを実行する。

ステップ 6. (オプション) 構成を変更する。

新規または追加の Reports Server のインストールと構成、および既存の Reports Server のアンインストールの詳細は、3.3 項「コマンド・ラインからの Reports Server の起動と停止」を参照してください。

Reports Server を構成する方法の詳細は次の各項を参照してください。

- Web CGI を使用した Reports Server の構成の詳細は、3.2 項「Reports Server の構成 : Reports Web CGI の例」を参照してください。
- Reports サブレットを使用して Reports Server を構成するためのガイドラインは、3.4 項「Reports サブレットの構成ガイドライン」で提供しています。

## 3.2 Reports Server の構成 : Reports Web CGI の例

この項では、Web CGI と任意の CGI 関連の Web サーバーを使用して Reports Server を手動で構成する方法についてステップ・バイ・ステップの説明を提供します。Java をベースとした Web サーバーを使用している場合の詳細は、3.4 項「Reports サブレットの構成ガイドライン」を参照してください。

**ヒント:**ここでは、使用しているマシンの C ドライブにソフトウェアをインストールしてサーバー設定を構成し、.RDF ファイルを保存することを前提としています。

この構成作業を意味のあるものとするためには、いくつかの前提が必要です。この項の趣旨に沿って、次の状況を前提とします。

- Web リクエストを可能にするように Reports Server を構成しています。

- 任意の Web CGI 関連の Web サーバーと Reports Web CGI を併用しています。
- Reports Server 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。

### 3.2.1 ステップ 1. ソフトウェアのインストール

Reports Server をインストールして構成するには、Oracle Reports Server ソフトウェアをインストールしておく必要があります。Oracle Reports Server コンポーネントおよび Report Builder をインストールする方法の詳細は、『Oracle Reports Developer for Windows スタート・ガイド リリース 6i』( 部品番号 :J00454-01 ) を参照してください。

**ヒント:** Reports Thin Client、Reports Server または Report Builder のみをインストールするには、Oracle Installer を起動して、「カスタム」オプションを選択します。「使用可能な製品」リストボックスから、「Oracle Developer - Reports」ノードを拡張して、Reports Thin Client、Oracle Reports Server または Report Builderの中から希望するものを選択します。『Oracle Reports Developer for Windows スタート・ガイド リリース 6i』( 部品番号 :J00454-01 ) の説明に従って、インストール作業を進めてください。

#### 3.2.1.1 Web サーバーでの作業

Web サーバーをインストールしてから Oracle Reports Thin Client コンポーネントをインストールします。Web サーバーをインストールする方法の詳細は、ベンダーが提供する Web サーバーのドキュメントを参照してください。

#### 3.2.1.2 Reports Server マシンでの作業

1. Oracle Reports Server コンポーネントをインストールします。このマシンでレポートを作成する場合は、次に Report Builder もインストールしてください。Report Builder は、レポートを作成する場合のみ必要になります。Reports Server マシンを配布マシンとしてのみ使用する場合は、Report Builder コンポーネントはインストールしないでください。
2. Oracle Reports Server のインストール時には、Reports Server の TNS 名とポート番号を指定します。この情報は、ORACLE\_HOME\NET80\ADMIN ディレクトリにある tnsnames.ora ファイルに追加されます。

**ヒント:** Reports Server のエントリは、テキスト・エディタで tnsnames.ora ファイルをオープンして編集できます。

- Reports Server TNS 名。repserver.world と入力するか、デフォルト値を使用してください。

ここで、repserver.world はサーバー・インスタンス名、.world は sqlnet.ora ファイルの NAMES.DEFAULT\_DOMAIN に設定したドメインです。sqlnet.ora ファイルに NAMES.DEFAULT\_DOMAIN 設定を定義していない場合は、サーバー・インスタンス名の .world を省略してください。

- ポート。サーバーが聞きとるポート番号を入力します。または、デフォルト値を使用します。デフォルトは 1949 です。
3. Reports Server の TNS 名とポート番号を控えておいてください。この情報は、この Reports Server に対するサービス・エントリを Web サーバー・マシン上の `tnsnames.ora` ファイルに追加するときに必要になります。詳細は、3.2.3.2 項「Reports Server のサービス・エントリを作成する手順」を参照してください。

3.2.2 ステップ 2. Web サーバーの構成

このステップは Web サーバー・マシン上で行います。

**注意:** Web サーバー上の Web cgi-bin ディレクトリには CGI 実行ファイルがあります。

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、次の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いてください。

表 3-1 Web 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
Web cgi-bin	c:\¥orant¥oas¥bin	/CGI-BIN	実行

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。

3.2.3 ステップ 3. Reports Web CGI の構成

**注意:** Reports サブレットを使用して Web サーバーを構成する方法の詳細は、3.4 項「Reports サブレットの構成ガイドライン」を参照してください。

このステップは Web サーバー・マシン上で行います。

3.2.3.1 Reports Web CGI を構成する手順

1. `rwcgi60.exe` ( `ORACLE_HOME¥BIN` ディレクトリにある ) を cgi-bin ディレクトリにコピーします。



**ヒント:** cgi-bin ディレクトリは Web サーバーの構成で定義されます。詳細は Web サーバーのドキュメントを参照してください。表 3-1 「Web cgi-bin 物理・仮想ディレクトリ」では、Web CGI の物理ディレクトリは C:\¥orant¥oas¥bin です。

このステップは Web サーバー・マシン上で行います。

### 3.2.3.2 Reports Server のサービス・エントリを作成する手順

Web サーバーと Reports Server が別のマシン上にある場合は、Reports Server 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 Reports Server 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、Web CGI 実行ファイルと Reports Server との通信が可能になります。

**ヒント:** Reports Server のサービス・エントリの設定を確認する場合は、Reports Server 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 tnsnames.ora ファイル (ORACLE\_HOME¥NET80¥ADMIN ディレクトリにある) をオープンします。
2. 次の Reports Server 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。

repserver.world はサーバー・インスタンス名、.world は sqlnet.ora ファイルの NAMES.DEFAULT\_DOMAIN に設定したドメインです。sqlnet.ora ファイルに NAMES.DEFAULT\_DOMAIN 設定を定義していない場合は、サーバー・インスタンス名の .world を省略してください。

repserver\_machine.mydomain は、ホスト名かマシンの IP アドレスです。

1949 は、サーバーが聞きとり対象とするポートの番号です。

### 3.2.3.3 (オプション) デフォルトの Reports Server を設定する手順

**3.2.3.3.1 Windows NT プラットフォーム** 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。3.2.4.1.1 項「(オプション) ソース・パスの設定」を参照してください。

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 Reports Server の名前 (Reports Server の TNS サービス・エントリ名) です。

詳細は、3.2.1 項「ステップ 1. ソフトウェアのインストール」を参照してください。

**注意:** 設定値の大文字と小文字が区別されるどうかを判断するには、使用しているオペレーティング・システムのドキュメントを参照してください。

**3.2.3.3.2 UNIX プラットフォーム** UNIX での手順は、次のとおりです。

**ヒント:** 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。

REPORTS60\_REPORTS\_SERVER 環境変数に Reports Server の名前を設定します。

```
setenv REPORTS60_REPORTS_SERVER repserver
```

ここで、repserver は、構成中の Reports Server の名前、つまり Reports Server の TNS サービス・エントリ名です。詳細は、3.2.1 項「ステップ 1. ソフトウェアのインストール」を参照してください。

**ヒント:** 設定値の大文字と小文字が区別されるどうかを判断するには、使用しているオペレーティング・システムのドキュメントを参照してください。

## 3.2.4 ステップ 4. Reports Server の構成

この項では、環境変数とデフォルト・プリンタを設定し、サーバーを起動して Reports Server を構成する方法を説明します。

### 3.2.4.1 Windows NT で Reports Server を構成する手順

Windows NT で Reports Server を構成する手順は、2 つに分けられます。

- (オプション) ソース・パスの設定
- Reports Server の起動


**3.2.4.1.1 (オプション) ソース・パスの設定** 別の方法としては、Reports Server をインストールした後に、Reports Server の構成ファイルでソース・パスを設定することができます。詳細は、付録 B「Reports Server 構成パラメータ」の SOURCEDIR パラメータを参照してください。

1. ソース・ディレクトリをまだ作成していない場合は、作成します。
2. 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。
3. regedit と入力してレジストリ・エディタを表示します。

4. HKEY\_LOCAL\_MACHINE SOFTWARE ORACLE を拡張します。
5. REPORTS60\_PATH をダブルクリックします。
6. 新しいレポートのソース・パス（たとえば C:\¥WEB\_REPORTS など）を追加します。

**注意:** 設定値の大文字と小文字が区別されるどうかを判断するには、使用しているオペレーティング・システムのドキュメントを参照してください。

#### 3.2.4.1.2 Reports Server の起動 次の手順に従ってください。

1. Reports Server 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、 (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] 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. サービス・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。

**ヒント:** Postscript またはプリンタに出力するときは、Reports Server 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 Reports Server がいつ起動したか表示されます。Reports Server が起動しない場合は、付録 F「トラブルシューティング」を参照してください。

**ヒント:** Reports Server を初めて起動したとき、ORACLE\_HOME¥REPORT60¥SERVER ディレクトリに Reports Server の構成ファイル（たとえば repserver.ora）が作成されます。Reports Server 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・パスを設定することができます。構成ファイルを変更したら、Reports Server を停止して再起動し、変更内容を有効にします。

### 3.2.4.2 UNIX で Reports Server を構成する手順

UNIX で Reports Server を構成する手順は、2 つに分けられます。

- (オプション) 環境変数の設定
- Reports Server の起動

#### 3.2.4.2.1 (オプション) 環境変数の設定

**ヒント:** 環境変数の設定と RWMTS60 実行ファイルの起動は、シェル・スクリプトを作成して行うことができます。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。

**注意:** UNIX では、実行ファイルの名前には小文字が使われています。UNIX では大文字と小文字が区別されます。

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
```

REPORTS60\_PATH には、Reports のソース・ファイルやライブラリなど、Oracle Reports Server で検索対象とするディレクトリがすべて含まれます。

**ヒント:** 別の方法としては、Reports Server のインストール後に、Reports Server の構成ファイルでソース・パスを設定することもできます。詳細は、付録 B「Reports Server 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
```

**ヒント:** 設定値の大文字と小文字が区別されるどうかを判断するには、使用しているオペレーティング・システムのドキュメントを参照してください。

**3.2.4.2.2 Reports Server の起動** Oracle Reports Server のインストールの実行中に、Reports Server を起動する RWMTS60 コマンドがシェル・スクリプトに追加されました。

1. シェル・スクリプトを実行して Reports Server を起動します。

**ヒント:** Reports Server を初めて起動すると、\$ORACLE\_HOME/REPORT60/SERVER ディレクトリに、Reports Server の構成ファイル（repserver.ora など）が作成されます。Reports Server のキャッシュの設定はデフォルトで設定されます。キャッシュ・ディレクトリを変更するとき、あるいはレポートのソース・パスを設定するときは、構成ファイルを変更します。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 Reports Server を停止して再起動する必要があります。

2. (オプション) \$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 Reports Server が稼働していることを確認します。

```
rwrtqv60 server=repserver
```

この Reports Server 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。

**ヒント:** Reports Server を終了するには、[Ctrl-C] を押します。

**ヒント:** Postscript ファイルあるいはプリンタに出力するときは、  
(`$ORACLE_HOME/GUICOM6/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。

### 3.2.5 ステップ 5. Web ブラウザからのレポートの実行

1. Reports Server 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/password@my_db+desformat=html+destype=cache
```

ここで、`username/password@my_db` は有効なデータベース・ログイン名に置き換えてください。`my_db` は、事前に Reports Server 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。3.2.3.2 項「Reports Server のサービス・エントリを作成する手順」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。

**ヒント:** 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 F「トラブルシューティング」を参照してください。

2. (オプション) リクエストのステータスを表示します。
  - Windows NT の場合は、Reports Queue Manager を起動して、`repserver` のキューを表示します。詳細は Reports 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。

```
rwrv60 server=repserver showjobs=current
```

これで、CGI 対応 Web サーバーで WebCGI を使用して Reports Server を構成する例は終わりです。動的レポートに対応した Reports Server の構成が正常に終了しました。

#### 3.2.5.1 次のステップ

1. (オプション) Reports Server を調整してパフォーマンスの最適化やアクセス・コントロールなどの追加機能の実現を図ります。詳細は、3.2.6 項「ステップ 6. (オプション) Reports Server の構成の変更」を参照してください。この作業を行うと、3.2.5 項「ステップ 5. Web ブラウザからのレポートの実行」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。

## 3.2.6 ステップ 6. ( オプション ) Reports Server の構成の変更

Reports Server を初めて起動したときは、Reports Server 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、Reports Server 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、3.2.6.1 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「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。
- Reports Server の構成を変更して負荷を分散する方法の詳細は、第 6 章「Reports Server 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。
- Reports Server の構成ファイルを修正する場合は、付録 B「Reports Server 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 C「環境変数」を参照してください。

### 3.2.6.1 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

データベースの設定を変更すると、ジョブが実行されるたびに Reports Server のキュー・アクティビティのスナップショットを取得できます。Reports Server 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、Reports Server 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もできます ( Reports Server 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。

データベースに接続するには、構成ファイルで `REPOSITORYCONN` パラメータを設定します。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` スクリプトを実行します。

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、Reports Server を停止して再起動し、変更内容を有効にします。

#### 3.2.6.1.1 Reports Server マシン上 :

1. `ORACLE_HOME\REPORT60\SERVER` ディレクトリにある `repserver.ora` 構成ファイルを、テキスト・エディタでオープンします。
2. 次の構成パラメータを追加します。

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

repserver\_schema/password@database は、指定した Reports Server 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。

**ヒント:** いくつかの Reports Server 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 Reports Server 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、Reports Server 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。

3. ORACLE\_HOME¥REPORT60¥SQL にある rw\_server.sql script 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>
```

4. Reports Server 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。Reports Server は起動時にデータベースに接続します。

**注意:** Reports Server を再起動すると、デフォルトではデータベースのキュー・アクティビティが削除されます。デフォルトは API を編集して上書きすることができます。

### 3.3 コマンド・ラインからの Reports Server の起動と停止

コマンド・ラインから追加の Reports Server を起動したり、既存の Reports Server を停止したりできます。それには、Reports Server コンポーネントがインストール済みであることが必要です。詳細は、3.1 項「インストレーションと構成の概要」を参照してください。

RWMTS60 は、Reports Server の実行形式のファイルです。Windows NT の場合、このファイルには次の 3 つのコマンドが対応付けられています。

- **Install:** Reports Server を、Windows NT 上でサービスとしてインストールします。
- **Listen:** Reports Server を、Windows NT 上で非サービスとしてインストールします。
- **Uninstall:** Reports Server を、Windows NT 上でサービスとしてアンインストールします。

Reports Server は、手動あるいはバッチ・モードで Install、Listen および Uninstall できます。インストレーション中に発生したエラー・メッセージを無視してオペレーションが中断されることなく完了するようにするには、BATCH 引数に Yes を設定します。インストレーション中に発生したエラーやステータス情報は、Reports Server のログ・ファイル

(`ORACLE_HOME\REPORT60\SERVER\repserver.log`にある)で調べることができます。  
BATCH のデフォルトは No です。

次のことができます。

- 他の Reports Server を起動する。詳細は、3.3.1 項「別の Reports Server の起動」を参照してください。
- Reports Server を非サービスとして Windows NT で起動する。詳細は、3.3.2 項「Windows NT における Reports Server の非サービスとしての起動」を参照してください。
- Reports Server を停止する。詳細は、3.3.3 項「Reports Server の停止」を参照してください。

### 3.3.1 別の Reports Server の起動

他の Reports Server を起動するには、次のことが必要です。

- Reports Server のサービス・エントリを作成する。
- (オプション) 環境変数を設定する(ソース・パスの設定など)。
- Reports Server を起動する。

#### 3.3.1.1 Reports Server のサービス・エントリを作成する手順

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

```
repserver.world = (ADDRESS = (PROTOCOL = TCP) (Host =  
repserver2_machine.mydomain) (Port = 1950))
```

各項目の内容は次のとおりです。

`repserver2.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` の設定値で指定されたドメインです。  
`NAMES.DEFAULT_DOMAIN` の設定値が `sqlnet.ora` ファイルで指定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。

`repserver2_machine.mydomain` は、ホスト名かマシンの IP アドレスです。

1950 は、追加サーバーが聞きとり対象とするポートの番号です。デフォルトでは、第一サーバーは 1949 を聞きとります。

3. Reports Server 以外のマシンにインストールした場合は、ステップ 1 とステップ 2 を Web サーバー・マシン上で繰り返します。



### 3.3.1.2 Windows NT で Reports Server を構成する手順

Windows NT で Reports Server を構成するために必要な手順は、次のとおりです。

- (オプション) ソース・パスの設定
- Reports Server の起動

#### 3.3.1.2.1 (オプション) ソース・パスの設定

要求したレポートを Reports Server で実行するには、レポートの場所を認識させる必要があります。REPORTS60\_PATH 環境変数でレポートのソース・パスを設定します。

別の方法としては、Reports Server をインストールした後に、Reports Server の構成ファイルでソース・パスを設定することができます。詳細は、付録 B「Reports Server 構成パラメータ」の SOURCEDIR パラメータを参照してください。

1. ソース・ディレクトリをまだ作成していない場合は、作成します。
2. 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。
3. regedit と入力してレジストリ・エディタを表示します。
4. HKEY\_LOCAL\_MACHINE SOFTWARE ORACLE を拡張します。
5. REPORTS60\_PATH をダブルクリックします。
6. 新しいレポートのソース・パス (たとえば C:\WEB\_REPORTS など) を追加します。

**注意:** 設定値に大文字と小文字の区別が必要かどうかを判断するには、使用しているオペレーティング・システムのドキュメントを参照してください。

#### 3.3.1.2.2 Reports Server の起動


1. Reports Server 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 -install repserver tcpip
```

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 -install repserver batch=yes
```

**ヒント:** 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. 「はい」をクリックして Reports Server を NT サービスとしてインストールし、「OK」をクリックします。
4. 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。

5. 「サービス」ダイアログ・ボックスで、Oracle Reports Server [repserver] を選択して「スタートアップ」をクリックします。
6. スタートアップ・ダイアログ・ボックスから「ログオン」セクションの「アカウント」を選択して、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入力あるいは選択します。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。

**ヒント:** Postscript またはプリンタに出力するときは、Reports Server 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。

7. サービスを、システム起動時に自動的に起動する設定にします。
8. 「OK」をクリックします。
9. 「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 Reports Server がいつ起動したか表示されます。Reports Server が起動しない場合は、付録 F「トラブルシューティング」を参照してください。

**ヒント:** Reports Server を初めて起動したとき、`ORACLE_HOME\REPORT60\SERVER` ディレクトリに Reports Server の構成ファイル（たとえば `repserver.ora`）が作成されます。Reports Server 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・パスを設定することができます。構成ファイルを変更したら、Reports Server を停止して再起動し、変更内容を有効にします。

### 3.3.1.3 UNIX で Reports Server を構成する手順

UNIX で Reports Server を構成するために必要な手順は、次のとおりです。

- （オプション）環境変数の設定
- Reports Server の起動

#### 3.3.1.3.1 （オプション）環境変数の設定

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 Reports Server 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。

**ヒント:** シェル・スクリプトを作成して、環境変数の設定と RWMTS60 実行ファイルの起動を行うことができます。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。

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
```

**ヒント:** 別の方法としては、Reports Server のインストール後に、Reports Server の構成ファイルでソース・パスを設定することもできます。詳細は、付録 B「Reports Server 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
```

**ヒント:** 設定値に大文字と小文字の区別が必要かどうかを判断するには、使用しているオペレーティング・システムのドキュメントを参照してください。

### 3.3.1.3.2 Reports Server の起動

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 Reports Server をフォアグラウンドで実行します。

```
rwmts60 name=repserver
```

あるいは、Reports Server をバックグラウンドで実行するときは、次のコマンド・ラインを実行します。

```
rwmts60 name=repserver &
```

2. サーバーが稼働しているかどうかを確認するには、ステップ 3 に進んでください。Reports Server が起動しない場合や、エラー・メッセージが表示される場合は、付録 F「トラブルシューティング」を参照してください。

**ヒント:** Reports Server を初めて起動すると、\$ORACLE\_HOME/REPORT60/SERVER ディレクトリに、Reports Server の構成ファイル (repserver.ora など) が作成されます。Reports Server のキャッシュの設定はデフォルトで設定されます。キャッシュ・ディレクトリを変更するとき、あるいはレポートのソース・パスを設定するときは、構成ファイルを変更します。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 Reports Server を停止して再起動する必要があります。

3. (オプション) \$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 Reports Server が稼働していることを確認します。

```
rwrvq60 server=repserver
```

この Reports Server 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ため、ステータス情報は表示されません。

**ヒント:** Postscript あるいはプリンタに出力するときは、\$ORACLE\_HOME/GUICOM6/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。

### 3.3.2 Windows NT における Reports Server の非サービスとしての起動

次のコマンド・ラインを実行します。

```
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
```

**ヒント:** 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。

### 3.3.3 Reports Server の停止

#### 3.3.3.1 Windows NT の場合

1. Reports Server 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
```

**ヒント:** 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。

#### 3.3.3.2 UNIX の場合

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。

## 3.4 Reports サブプレットの構成ガイドライン

Java ベースの Web サーバー (Oracle Lite、Oracle Application Server など) を使用しているときは、Reports サブプレットを使用して Reports Server を構成できます。Reports サブプレットを使用した Reports Server の構成作業は Web CGI の構成作業と似ています。ただし、サブプレットが使用可能な Web サーバー・ソフトウェアもインストールする必要があります。

次のガイドラインに従って、Reports サブプレットを使用して Java ベースの Web サーバーを構成してください。

1. Oracle Reports Thin Client ソフトウェアと、サブプレットが使用可能な Web サーバー・ソフトウェアをインストールします。Oracle Reports Thin Client をインストールする方法の詳細は、『Oracle Reports Developer for Windows スタート・ガイド リリース 6i』

( 部品番号 :J00454-01 ) を参照してください。Web サーバーのインストールの詳細は、Web サーバーのドキュメントを参照してください。

2. Web サーバーを構成します。詳細は Web サーバーのドキュメントを参照してください。
3. Reports サブレットを構成します。詳細は、3.4.1 項「Web サーバーの構成」を参照してください。
4. Reports Server を構成します。Reports Server を構成する方法の詳細は、3.3 項「コマンド・ラインからの Reports Server の起動と停止」を参照してください。

### 3.4.1 Web サーバーの構成

この作業は、Web サーバー・マシンにおいて次の手順で行います。

1. サブレットのプロパティ・ファイル ( `servlet.properties` など ) に、次のエントリを追加します。

```
# Oracle Report servlet
servlet.RWServlet.code=oracle.reports.rwcgi.RWServlet
```

2. CLASSPATH 環境変数に Oracle Reports の `RWServlet.class` ファイルの入っているディレクトリを設定します。
3. Reports Server の構成を完了したら、Reports Server 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/RWServlet?report=your_report.rdf+
userid=username/password@my_db+desformat=html+destype=cache
```

ここで、`username/password@my_db` は有効なデータベース・ログイン名に置き換えてください。`my_db` は、事前に Reports Server 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。3.2.3.2 項「Reports Server のサービス・エントリを作成する手順」を参照してください。

このリクエストには、SERVER コマンド・ライン引数が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` 環境変数に `repserver` を設定している場合は必要ありません。



---

## 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
- マッピング・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

レポートはさまざまなリクエスト・メソッドを使用して実行することができます。それらは次にリストするとおりです。

- **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を実行することができます。RWCLI60 は、コマンド・ラインを解釈し、指定された Reports Server に送る実行形式のファイルです。このファイルでは、Reports 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 A「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・リクエストを実行することができます。Web CGI およびサプレットは、URL 構文を Reports Server 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。

- **WebDB コンポーネント**を使用すると、リンクを Oracle WebDB コンポーネントとして WebDB サイトに追加することができます。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。Reports Developer のシステム管理者は、Oracle WebDB 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。WebDB 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することができます。システム管理者は直接ウィザードからレポートを実行することができます。詳細は、第 5 章「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。
- **ActiveX コントロール**は、Reports Server 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。Reports Launcher は ActiveX コンテナの一例です。詳細は ActiveX および Reports Launcher のオンライン・ヘルプを参照してください。
- **SRW.RUN\_REPORT** は、Reports Runtime 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 Reports Server 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 Report 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 Reports Server 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。Reports Server は、指定された許容時間（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 つがキャンセルされた場合（たとえば Reports Queue Manager からキャンセルされた場合）、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 to 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 A「RWCLI60 コマンド・ライン引数」を参照してください。

### 使用上の注意

- 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。Reports Server によって重複ジョブが検出され、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。
  - Reports Queue Manager を使用して、あるいは手動で Reports Server 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。Reports Server 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、Reports Queue Manager を使用してキャッシュを空にすることもできます。
- キャッシュの設定の詳細は、Reports Queue Manager のオンライン・ヘルプまたは付録 B「Reports Server 構成パラメータ」を参照してください。
- 同一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、Reports Server 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 Reports Server 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。Reports Server 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することができます。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

**注意：** 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、WebDB 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「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。

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することができます。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することができます。

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・マッピング・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。

```
RWCLI60 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 (Web 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 C「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。キーが見つかると、その定義済みの値が Reports Server 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 (Web CGI) ファイルをオープンします。

**ヒント：**使用中のマッピング・ファイルを確認するために 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・ファイルは次のようになります。

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

ここで、key1 はキーの名前です。

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 A 「RWCLI60 コマンド・ライン引数」を参照してください。

**ヒント:** 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 C 「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」参照。

## 4.4 レポート・リクエストの指定

### 4.4.1 レポートの構築

1. 使用中の Reports Server 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) を作成します。このディレクトリは、必ず Reports Server 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 B 「Reports Server 構成パラメータ」参照。

**ヒント:** 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 C 「環境変数」を参照してください。

Report 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することができます。このレポート定義ファイルは、必ず Reports Server 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。レポートの作成についての詳細は、『レポート作成ガイド』マニュアルまたは Report Builder のオンライン・ヘルプを参照してください。



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. 「索引」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。  
標準的なレポートの構築

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

## 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することができます。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。

**注意:** 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、WebDB 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「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。

URL 構文は、次の形式で提示することができます。

- 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
```

**注意:** 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 A「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
```

**ヒント:** 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 C「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. URL 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 F「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、Queue Manager または Oracle WebDB 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 Reports Queue Manager のオンライン・ヘルプを参照してください

レポートを WebDB で WebDB 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することができます。詳細は、第 5 章「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。



---

## レポートへのユーザー・アクセスの管理

アクセス・コントロールにより、Oracle Reports Server 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。Reports Developer では、アクセスの制限された Reports Server やプリンタ上でレポートを実行するときに、WebDB を使用してセキュリティ・チェックを実行し、必要な権限を持ったユーザーのみにアクセスを許可します。アクセス・コントロール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、Reports Server 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の制限された Reports Server 上で実行し、プリンタに出力する権限を持ったユーザー。
- レポート定義ファイル、Reports Server 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 Reports Server およびプリンタの指定なども含みます。

WebDB には、レポート定義ファイル（すなわち、レポートの実行方法）に関する情報がパッケージ・プロシージャとして格納されます。レポートを実行するには、リクエストを受け付ける Reports Server のアクセス制限、レポート出力の印刷に使用するあらゆるプリンタに関するアクセス・コントロール情報も WebDB に格納する必要があります。WebDB にアクセス・コントロール情報を追加するには、Oracle Reports セキュリティ・ウィザードを使用します。WebDB にアクセス・コントロール情報を追加できるのは、Reports Developer システム管理者の権限を持ったユーザーのみです。

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。

- リンクを WebDB コンポーネントとして、レポートのパッケージ・プロシージャをポイントする WebDB サイトに追加する。手順の詳細は、5.3.8 項「ステップ 8. ユーザーがレポートを利用できるようにする」を参照してください。
- 自動的にリクエストが実行され、レポート出力が WebDB サイトに配信されてユーザーに表示されるように、スケジュールを設定する。詳細は、5.3.9 項「ステップ 9. レポー

トを実行して出力を WebDB サイトに配信するスケジュールを設定する」を参照してください。

- 標準 URL 構文をハイパーリンクとして Web サイトに追加する。詳細は、4.4 項「レポート・リクエストの指定」を参照してください。

**注意:** システム管理者は、WebDB の Reports Developer Security からレポート・リクエストを実行できます。詳細は、5.3.7 項「ステップ 7. 「Reports Developer パラメータ・フォーム」でパラメータ値を設定する」を参照してください。

## 5.1 アクセス・コントロールの構成およびセットアップの概要

この項では、アクセス・コントロールのための Reports Server の構成方法、アクセスの制限された Reports Server でレポート・リクエストを実行するときに使用するアクセス情報を WebDB に追加する方法を説明します。

次のステップでは、Web CGI またはサーブレットを使用した Reports Server の構成が完了していることを前提にしています。詳細は、第 3 章「Reports Server の構成」を参照してください。Reports Developer でアクセス・コントロールを実現するための詳細な例は、5.3 項「WebDB のアクセス・コントロールのセットアップ」を参照してください。

### Reports Developer Security のインストールと構成

ステップ 1. Reports Developer Security に対応した WebDB を構成します。

#### アクセス・コントロールのセットアップ

ステップ 1. アクセス・コントロールに対応した Reports Server を構成します。

ステップ 2. ユーザー・アカウントを作成します。

ステップ 3. (オプション) WebDB に使用時間カレンダーを作成します。

ステップ 4. WebDB にプリンタへのアクセスを追加します。

ステップ 5. WebDB に Reports Server へのアクセスを追加します。

ステップ 6. WebDB にレポート定義ファイルへのアクセスを追加し、パッケージ・プロシージャを作成します。

**注意:** 複数のレポートを WebDB にバッチで一括登録するには、Reports Developer Batch Registering コーティリティを使用します。詳細は、OTN (<http://technet.oracle.com>) に掲示された Reports Developer Batch Registering Reports テクニカル・ホワイト・ペーパーを参照してください。

ステップ 7. 「パラメータ・フォーム」でパラメータ値を設定します。

ステップ 8. WebDB サイト上でレポート・リクエストを公開します。

ステップ 9. (オプション) レポートを実行して出力を WebDB サイトに配信するためのスケジュールを設定します。



## 5.2 WebDB の Reports Security のインストールと構成

WebDB の Reports Developer Security のインストールと構成には、WebDB および Reports Server Security 機能のインストール、認証 cookie ドメインの設定が含まれます。

WebDB の Reports Server Security のインストールと構成が完了した後に、アクセス・コントロールに対応した Reports Server の構成方法、Reports Server、レポートおよびプリンタ・アクセスの WebDB への追加方法について、5.3 項「WebDB のアクセス・コントロールのセットアップ」を参照してください。

### 5.2.1 ステップ 1. WebDB の Reports Security を構成する

Reports Developer Security 機能を WebDB にインストールして構成するには、次の作業を実行する必要があります。

- WebDB リリース 2.2 以降をインストールする。
- WebDB に Reports Security 機能をインストールする。
- 認証 cookie ドメインを設定する。

#### 5.2.1.1 WebDB をインストールする手順

WebDB をまだインストールしていない場合は、WebDB を Oracle 7.3.4、Oracle 8.0.5 または Oracle 8i データベースにインストールします。詳細は、『Oracle WebDB スタート・ガイド』を参照してください。

#### 5.2.1.2 Reports Server セキュリティ・パッケージを WebDB にインストールする手順

Reports Server セキュリティ・パッケージは、あらゆるマシン（WebDB または Reports Server をインストールしたマシンなど）からインストールできます。

1. Reports Server セキュリティ・パッケージのみをインストールする場合は、Oracle Installer を起動して「**カスタム・インストール**」を選択します。「使用可能な製品」リスト・ボックスで、Oracle Reports Developer というノードを拡張し、「**Reports Server Security Packages**」を選択します。インストール・プロセスの詳細は、『スタート・ガイド』を参照してください。
2. インストール処理が完了したら、SQL スクリプトを実行してセキュリティ・パッケージを WebDB にインストールします。
  - Windows NT の場合、「Oracle Reports 6i Admin」メニューから「**Install Oracle Reports Security**」を選択します。
  - UNIX の場合、ORACLE\_HOME/REPORT60/SERVER/SECURITY ディレクトリに移動し、コマンド・ラインで次のように入力します。

```
sqlplus /nolog @rwwwvins.sql
```

- 3. Enter Connection String プロンプトで次の事項を入力して WebDB スキーマ ( username/password@database ) にログインします。


表 5-1 WebDB スキーマにログインするための接続


フィールド	説明
Username	WebDB スキーマにログインできるデータベース管理者 ( DBA ) 権限を持ったユーザー名。WebDB スキーマにログインできない場合は、DBA に相談してください。
Password	WebDB スキーマにログインするためのパスワード。
Database	WebDB スキーマとの接続に使用するデータベース名。

- 4. SQL スクリプトを完成したら、WebDB を起動し、WebDB スキーマにログインします。
- 5. メイン・メニューで「管理」をクリックします。「Oracle Reports セキュリティ」メニュー項目が表示されます。

5.2.1.3 認証 cookie ドメインを設定する手順

認証 cookie ドメインを設定することで、レポート送信先の Reports Server への認証情報の送信が可能になります。



- 1. このタスクに関する WebDB オンライン・ヘルプを表示するには、タイトル・バーでヘルプ・ボタンをクリックします。
- 2.  をクリックし、次のように入力します。  
認証 cookie
- 3. 「検索」をクリックし、次のトピックをクリックします。  
「認証 cookie ドメインの設定」  
トピックが表示されます。

WebDB をインストールしたマシンで、次の作業を実行します。

- 1. テキスト・エディタで wdbsvr.cfg ファイル ( ORACLE\_HOME¥LISTENER¥CFG ディレクトリにある ) をオープンします。[SERVER] セクション下で、次の構文を使用して構成パラメータを設定します。

```
ORCookieDomain=my_company.com
```

ここで、my\_company.com は Reports Server のドメイン名です。

- 2. 変更内容を保存し、構成ファイルをクローズします。

## 5.3 WebDB のアクセス・コントロールのセットアップ

この例では、アクセス・コントロールに対応した Reports Server を構成するための手順をステップ・バイ・ステップで説明します。レポート定義ファイル、Reports Server およびプリンタへのアクセスを WebDB に追加します。最後に、レポート・リクエストを WebDB サイトで公開し、許可されたユーザーがアクセスの制限されたレポートを実行できるようにします。

この例では、次の事項が前提になっています。

- Web CGI 構成例を使用して、Reports Server の構成が完了している。詳細は、3.2 項「Reports Server の構成 : Reports Web CGI の例」を参照してください。
- Reports Developer が認識できるプリンタがインストールされ、稼働中である。
- Reports Developer システム管理者は、WebDB サイトの管理者権限を持っている。これにより、Reports Developer システム管理者は、WebDB サイトへのアイテムの追加、他のユーザーにアイテムの管理権限を付与することが可能になります。
- `security.rdf` にアクセスできる。このレポートは、従業員の 401K レポートを生成します。このレポートの情報は、WebDB に追加されます。このファイルは、`ORACLE_HOME¥TOOLS¥DOC60¥US¥RBBR60` ディレクトリに提供されています。
- Reports Server で `security.rdf` ファイルを実行するには、Oracle Reports Developer のデモ表にアクセスする必要があります。デモ用の SQL スクリプトを使用してデモ表をデータベースにインストールします。

WebDB からアクセスできるレポートとして追加する 401K レポートには、4 名の架空従業員に関する 401K 投資情報が入っています。この機密レポートへのアクセスを、これら 4 人の従業員に限定します。さらに、従業員がレポートを要求した場合は、本人の個人情報のみにアクセスでき、他の従業員の情報にはアクセスできないことを保証します。これを実現するための作業は、次のとおりです。

- ユーザー・アクセスをレポートそのものに制限します。つまり、401K 加入者の 1 人である Jeff Abers 氏について、この例で使用するために特定のユーザー・アカウントを作成し、このユーザーにレポートへのアクセス権を与えます。
- ユーザーのアクセスを本人の個人情報のみに限定します。レポートの実行を許可されたユーザーは、自分の 401K レポートを検索するときに、姓と社会保障番号の組合せを正確に入力する必要があります。 `security.rdf` レポートには、姓と社会保障番号 (SSN) という 2 つのユーザー・パラメータが定義されています。WebDB で作成するランタイム・パラメータ・フォームには、401K 加入者の姓の値リストと、任意の社会保障番号を入力するためのパラメータがあります。

Jeff Abers 氏が自分の 401K 投資を再検討する場合を想定します。ランタイム・パラメータ・フォームで、Abers 氏は宛先パラメータの他に、値リストから姓を選択し、社会保障番号を入力する必要があります。レポートを実行するときは、ログインする必要があります。ログイン時に WebDB でチェックが行われ、レポートを実行するのに必要なアクセス権を持ったユーザーであるか確認されます。ログインに成功すると、Reports Server によってリクエス

トが処理されます。姓と社会保障番号の組合せが正確に入力された場合、リクエストに従って個人の 401K レポートが配信されます。

### 5.3.1 ステップ 1. アクセス・コントロールに対応した Reports Server を構成する

このステップを実行する前に、Reports Server をインストールし、構成しておく必要があります。詳細は第 3 章「Reports Server の構成」を参照してください。

アクセス・コントロールに対応した Reports Server を構成するための作業は、次のとおりです。

- WebDB に接続する TNS 名の別名を作成する。
- Reports Server へのアクセスを制限する。

#### 5.3.1.1 WebDB に接続する TNS 名の別名を作成する手順

WebDB で使用する TNS 名の別名は、Reports Server がインストールされたマシン上の `tnsnames.ora` ファイルで作成します。これにより、Reports Server と WebDB の通信が可能になります。

TNS 名の別名を作成するには、Net8 Easy Config ツールを使用するか、またはテキスト・エディタで `tnsnames.ora` ファイルを編集します。

TNS 名の別名を作成するときに必要な情報は、次のとおりです。

- WebDB インスタンス用の TNS 名の別名。
- WebDB がインストールされたデータベースのホスト名。
- WebDB がインストールされたデータベースのポート番号。
- WebDB がインストールされたデータベースのシステム識別子 (SID)。

**ヒント:** ホスト名、ポート番号および SID は、データベースがインストールされたマシン上の `ORACLE_HOME\NETWORK\ADMIN` ディレクトリの `tnsnames.ora` ファイルに入っています。

Reports Server マシン上:

1. 次のいずれかを実行します。
  - Net 8 Easy Config を起動し (マシンにインストールされている場合) ウィザードの指示に従って TNS 名の別名を作成します。
  - `ORACLE_HOME\NET80\ADMIN` ディレクトリの `tnsnames.ora` ファイルをオープンします。ステップ 2 へ進みます。

**ヒント:** Reports Server Security 機能を Reports Server マシンからインストールした場合は、WebDB 用の TNS 名の別名は自動的に作成されています。このステップを省略して 5.3.1.2 項「Reports Server へのアクセスの制限」に進んでください。

2. WebDB との接続に使用する TNS 名の別名を次のように入力します。

```
sec_rep.world =
  (DESCRIPTION =
    (ADDRESS =
      (PROTOCOL = TCP)
      (HOST = my_pc.my_domain)
      (PORT = 1521)
    )
    (CONNECT_DATA = (SID = ORCL)
  )
)
```

各項目の内容は次のとおりです。

repserver2.world は WebDB 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 の設定値で指定されたドメインです。NAMES.DEFAULT\_DOMAIN の設定値が sqlnet.ora ファイルで指定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。

TCP は、プロトコル・アドレス情報です。

my\_pc.my\_domain は、WebDB がインストールされたマシンのホスト名または IP アドレスです。

1521 は WebDB がインストールされたデータベースへのポート番号です。

ORCL は、WebDB がインストールされたデータベースの Oracle システム識別子です。

3. tnsnames.ora ファイルを保存し、クローズします。

### 5.3.1.2 Reports Server へのアクセスの制限

Reports Server へのアクセスを制限するには、Reports Server の構成ファイルに SECURITYTNSNAME パラメータを設定します。このパラメータを設定すると、アクセス・コントロールが実施されます。アクセスの制限された Reports Server へレポートを要求するには、許可されたユーザーであることを証明する必要があります。

レポート・リクエストを実行するには、レポート定義ファイルへのアクセスを WebDB に追加する必要があります。制限なしであらゆるレポート・リクエストを実行する場合は、その Reports Server を使用する WebDB で「サーバー・アクセス」ウィザードの「登録済みのレポート定義ファイル (RDF) のみ実行」オプションをオフに設定してください。この場合も、ユーザーがレポートを実行するには Reports Server に対する認証が必要です。

Reports Server マシン上：

1. テキスト・エディタで repserver.ora 構成ファイル  
(ORACLE\_HOME¥REPORT60¥SERVER ディレクトリにある) をオープンします。次の構文  
を使用して、SECURITYTNSNAME パラメータを設定します。

```
SECURITYTNSNAME="sec_rep"
```

ここで、sec\_rep は tnsnames.ora ファイルで定義した WebDB サーバー・インスタ  
ンスの TNS 名の別名です。

2. Reports Server の構成ファイルを保存し、クローズします。
3. Reports Server を一度停止してから再起動し、Reports Server 構成ファイルの変更を反  
映させます。

## 5.3.2 ステップ2. ユーザー・アカウントを作成する


次のユーザー・アカウントを作成する必要があります。

- Reports Developer システム管理者。これにより、アクセス制限された Reports Server、  
レポート定義ファイルおよびプリンタ用のアクセス・コントロールを作成し、管理しま  
す。5.3.2.1 項「Reports Developer システム管理者のユーザー・アカウントを作成する  
手順」参照。Reports Developer システム管理者を作成した後に、アクセスを制限する  
Reports Server、レポート定義ファイルおよびプリンタ用のアクセス・コントロールを  
作成します。
- アクセスの制限された Reports Server およびプリンタに対して制限されたレポートを実  
行できるようにアクセス権を与えるすべてのユーザー。5.3.2.2 項「レポートの実行のた  
めのユーザー・アカウントを作成する手順」参照。ユーザー・アカウントは、いつでも  
作成できます。しかし、Reports Server、レポート定義ファイルまたはプリンタへのア  
クセスを制限する場合、アクセス権の必要なユーザーがあらかじめわかっているとき  
は、最初にユーザー・アカウントを作成するのが一番よい方法です。

### 5.3.2.1 Reports Developer システム管理者のユーザー・アカウントを作成 する手順

WebDB でセキュリティ管理を実行するには、RW\_ADMINISTRATOR ロールを持ったユー  
ザー・アカウントが必要です。RW\_ADMINISTRATOR ロールを持ったユーザーのみが、  
WebDB の Oracle Reports セキュリティ・ウィザードにアクセスできます。加えて、レポ  
ートのパッケージ・プロシージャや作成する値リスト (LOV) を所有するスキーマに対する  
BUILD IN 権限が必要です。ユーザー・アカウントに DBA 権限が与えられている場合は、  
ユーザー・アカウントを作成できます。そうでない場合は、DBA に連絡してユーザー・ア  
カウントを作成してもらいます。



1. このタスクに関する WebDB オンライン・ヘルプを表示するには、タイトル・バーでヘルプ・ボタンをクリックします。
  2.  をクリックし、次のように入力します。  
新規ユーザーの作成
  3. 「検索」をクリックし、次のトピックをクリックします。  
「新規ユーザーの作成」
- トピックが表示されます。

**注意:** パッケージ・プロシージャや作成するパラメータ値リストを、異なるスキーマで所有することができます。その場合は、複数のスキーマに対する BUILD IN 権限が必要です。

レポート・アイテムを WebDB サイトに追加するには、WebDB サイトを作成しておく必要があります。WebDB サイトを作成する場合は、「Grant」オプションによって SYS.DBMS\_SQL パッケージ・プロシージャに対する Execute 権限を付与された DBA である必要があります。この権限により、サイトを作成し、アイテムの管理権限を他のユーザーに付与できます。


**ヒント:** サイト管理者でない場合は、アイテムの追加先フォルダの Own、Manage Item または Create With Approval 権限をサイト管理者から付与してもらう必要があります。詳細は、DBA またはサイト管理者に相談してください。

### 5.3.2.2 レポートの実行のためのユーザー・アカウントを作成する手順

レポート・リクエストを実行するアクセス権が与えられたすべてのユーザーには、WebDB の認識できるユーザー・アカウントが必要です。Reports Developer には、ユーザーに割当て可能なロールが 4 つ定義されています。各ロールは、ジョブの監視またはエラー・メッセージの表示など、特定のシステム管理機能へのアクセス権をユーザーに与えます。ユーザーに特定の Reports Developer ロールを割り当てなかった場合は、デフォルトにより、Reports Developer 基本ユーザー機能（つまり、RW\_BASIC\_USER ロール）が暗黙的に割り当てられます。

ユーザー・アカウントに DBA 権限が与えられている場合は、ユーザー・アカウントを作成できます。そうでない場合は、DBA に連絡してユーザー・アカウントを作成してもらいます。



1. このタスクに関する WebDB オンライン・ヘルプを表示するには、タイトル・バーでヘルプ・ボタンをクリックします。
  2.  をクリックし、次のように入力します。  
新規ユーザーの作成
  3. 「検索」をクリックし、次のトピックをクリックします。  
「新規ユーザーの作成」
- トピックが表示されます。

この例では、401K プランに加入している従業員の 1 人である Jeff Abers 氏のユーザー・アカウントを作成します（または作成を依頼します）。彼のユーザー・アカウントを JABERS とします。彼には基本ユーザー・ロールが割り当てられています。レポート・リクエストを実行するアクセス権が必要なユーザーのアカウントを作成する場合は、DBA に相談してください。Reports Developer ロールは、必要に応じてユーザーに割り当てます。

**ヒント:** JABERS ユーザー・アカウントがすでに存在する場合は、末尾にあなたのイニシャルを付け足してください（たとえば、JABERSAA）。

### 5.3.3 ステップ 3. 使用可能日カレンダーを作成する

使用可能日カレンダーは、レポート定義ファイル、Reports Server およびプリンタを使用して処理できる時間を決定したものです。

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）。
- 複合型の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）。

レポート定義ファイル、Reports Server またはプリンタに関連付けることができる使用可能日カレンダーは 1 つのみです。実働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。

**注意:** レポート定義ファイル、Reports Server 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用です。



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。作成する使用可能日カレンダーは、次のとおりです。

- 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。

#### 5.3.3.1 デイリー・カレンダーを作成する手順

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。




**ヒント:** 現在のウィザード・ページに関する状況判断ヘルプを参照するには  をクリックします。WebDB のヘルプ・システムにアクセスするには、タイトル・バーの  をクリックします。

1. WebDB にアクセスし、Reports Developer システム管理者としてログインします。




**ヒント:** 「Oracle Reports セキュリティ」ウィザードにアクセスするには、Reports Developer システム管理者としてログインする必要があります。

2. Oracle WebDB ホーム・ページで、「**管理**」をクリックします。

**ヒント:** WebDB のどのページでもナビゲーション・ツールバーの  をクリックして「管理」ページにアクセスできます。

3. 「管理」ページで、「**Reports Developer セキュリティ**」をクリックします。
4. 「Oracle Reports セキュリティ」ページで、「**使用可能日カレンダー**」をクリックします。
5. 「使用可能日カレンダー」ページで、「**シンプル**」オプションをクリックしてカレンダーを新規作成します。
6. 「**新規作成**」をクリックしてシンプルな使用可能日カレンダーを作成します。
7. 「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」に Daily と入力します。

**ヒント:** Daily カレンダーがすでに存在する場合は、末尾にあなたのイニシャルを付け足してください（たとえば、DailyAA ）。

8.  をクリックします。
9. 「使用可能日時」ページで、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。
10. 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。
11. Daily を「繰り返し」オプションとして選択します。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終了するまで、このパターンが毎日繰り返されます。
12.  をクリックします。
13. (オプション)「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。
14.  をクリックします。
15. 「シンプルな使用可能日カレンダーを作成」カレンダー・ページで、「OK」をクリックしてカレンダーを作成します。

5.3.3.2 メンテナンス・カレンダーを作成する手順

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. 「使用可能日カレンダー」ページで、「**シンプルな使用可能日カレンダー**」オプションをクリックしてカレンダーを作成します。
- 2. 「**新規作成**」をクリックします。
- 3. 「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」に Maintenance と入力します。  
**ヒント:** メンテナンス・カレンダーがすでに存在する場合は、末尾にあなたのイニシャルを付け足してください (たとえば、MaintenanceAA )。
- 4.  をクリックします。
- 5. 5.3.3.1 項「デイリー・カレンダーを作成する手順」のステップ 9 ~ ステップ 15 に従って、次の使用可能日ルールを定義します。

表 5-2 メンテナンス・カレンダー・ルール

フィールド	値
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。
繰り返し	「Weekly (毎週)」を選択します。

5.3.3.3 クリスマス・カレンダーを作成する手順

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. 「使用可能日カレンダー」ページで、「**シンプルな使用可能日カレンダー**」オプションをクリックして 3 番目のカレンダーを作成します。
- 2. 「**新規作成**」をクリックします。
- 3. 「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」に Christmas と入力します。  
**ヒント:** クリスマス・カレンダーがすでに存在する場合は、末尾にあなたのイニシャルを付け足してください (たとえば、ChristmasAA )。


4.  をクリックします。
5. 5.3.3.1 項「デイリー・カレンダーを作成する手順」のステップ 9 ～ ステップ 15 に従って、次のルールを定義します。

表 5-3 クリスマス・カレンダー・ルール







フィールド	値
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	「Yearly (毎年)」を選択します。

#### 5.3.3.4 複合型の使用可能日カレンダーを作成する手順

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. 「使用可能日カレンダー」ページで、「**結合された使用可能日カレンダー**」オプションをクリックして、作成した 3 つのカレンダーから 1 つの複合カレンダーを作成します。
2. 「**新規作成**」をクリックします。
3. 「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」に `Production` と入力します。

**ヒント:** 運用カレンダーがすでに存在する場合は、末尾にあなたのイニシャルを付け足してください (たとえば、`ProductionAA`)。

4.  をクリックします。
5. 「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・ボックスのデイリー、メンテナンスおよびクリスマス・カレンダーをクリックします。
6.  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、 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。
7.  をクリックします。
8. 「使用可能日カレンダーを除外」で、[Ctrl] キーを押しながら「使用可能日カレンダー」リスト・ボックスのメンテナンスおよびクリスマス・カレンダーをクリックします。
9.  をクリックしてメンテナンスおよびクリスマス・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。
10.  をクリックします。

11. 「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。

**ヒント:** 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かどうかを確認します。



12.  をクリックします。
13. 「結合された使用可能日カレンダーを作成」ページで、「OK」をクリックして運用カレンダーを作成します。

### 5.3.4 ステップ 4. WebDB に Reports Developer プリンタへのアクセスを追加する

プリンタ・アクセスで定義する内容は、次のとおりです。

- WebDB でレポート出力の印刷に使用できるプリンタ。
- レポート出力を印刷するプリンタのアクセス権を持ったユーザー。
- レポート・リクエストの印刷のためにこのプリンタを使用できる期間。






**前提条件:** Reports Developer 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。詳細は、付録 A「RWCLI60 コマンド・ライン引数」で説明している DESNAME および DESFORMAT コマンド・ライン引数を参照してください。

**ヒント:** 現在のウィザード・ページに関する状況判断ヘルプを参照するには  をクリックします。WebDB のヘルプ・システムにアクセスするには、タイトル・バーの  をクリックします。

1. ナビゲーション・ツールバーのすぐ上にあるリンク履歴から「Oracle Reports **セキュリティ**」をクリックします。
2. 「Oracle Reports セキュリティ」ページで、「**プリンタ・アクセス**」をクリックします。
3. 「**新規作成**」をクリックしてプリンタ・アクセスを WebDB に追加します。
4. 「プリンタ名」ページで、「プリンタ名」フィールドに Reports\_Printer と入力します。

**ヒント:** このプリンタ名がすでに存在する場合は、末尾にあなたのイニシャルを付け足してください（たとえば、Reports\_PrinterAA）。



5. プリンタのオペレーティング・システム名を「OS プリンタ名」フィールドに入力します（たとえば、Windows NT の OS プリンタ名は ¥¥net\_machine¥my\_printer などです）。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。

6.  をクリックします。
7. 「ユーザーとロール」ページで、JABERS と Reports Developer システム管理者のユーザー・アカウントを「All Users and Roles」リスト・ボックスから選択して、レポートをプリンタに出力できるユーザーを指定します。
8.  をクリックして、このユーザーを「Selected Users and Roles」リスト・ボックスに移動します。
9.  をクリックします。
10. 「使用可能日カレンダー」ページで、使用可能日カレンダーとして `Production` と入力するか、または  をクリックして使用可能日カレンダーを検索します。  
**ヒント:** このプリンタをいつでも使用できるように設定する場合は、カレンダーを指定しないでください。
11.  をクリックします。
12. 「プリンタ・アクセスを追加」ページで、「OK」をクリックしてプリンタへのアクセスを WebDB に追加します。

### 5.3.5 ステップ 5. WebDB に Reports Server へのアクセスを追加する

Reports Server アクセスでは、次の内容を WebDB で定義します。

- WebDB でリクエスト処理に使用できる Reports Server。
- Reports Server が使用できるプリンタ。
- この Reports Server にレポート・リクエストを送信するアクセス権を持ったユーザー。
- この Reports Server がレポート・リクエストを受け付ける期間。

**ヒント:** 現在のウィザード・ページに関する状況判断ヘルプを参照するには  をクリックします。WebDB のヘルプ・システムにアクセスするには、タイトル・バーの  をクリックします。


1. リンク履歴から「Reports Developer セキュリティ」をクリックします。
2. 「Reports Developer セキュリティ」ページで、「サーバー・アクセス」をクリックします。
3. 「新規作成」をクリックします。
4. 「サーバー名とプリンタ」ページで、「サーバー名」フィールドに `Repserver` と入力します。  
**ヒント:** このサーバー名がすでに存在する場合は、末尾にあなたのイニシャルを付け足してください (たとえば、`RepserverAA`) 。
5. 「Reports Server の TNS 名」フィールドに `repserver` と入力します。

**ヒント:** Reports Developer TNS 名は、Reports Server のインストールおよび構成時に `tnsname.ora` ファイルに追加した Reports Server のサービス・エントリ名です。詳細は、第 3 章「Reports Server の構成」を参照してください。


6. Reports Developer Web Gateway URL を次のように小文字で入力します。


`http://my_webserver/cgi-bin/rwcgi60.exe`


**ヒント:** Reports Developer Web Gateway URL は、Web CGI の仮想ディレクトリによって決まります。3.2.2 項「ステップ 2. Web サーバーの構成」参照。

7. 「プリンタ」リスト・ボックスから「Reports\_Printer」を選択します。
8.  をクリックします。
9. 「Users and Roles」ページで、JABERS と Reports Developer システム管理者のユーザー・アカウントを「All Users and Roles」リスト・ボックスから選択して、このサーバーにアクセスできるユーザーを指定します。


**ヒント:** プリンタへのアクセス権を与えたときと同じユーザーを選択する必要があります。

10.  をクリックして、このユーザーを「Selected Users and Roles」リスト・ボックスに移動します。

11.  をクリックします。

12. 「使用可能日カレンダー」ページで、使用可能日カレンダーとして `Production` と入力するか、または  をクリックして使用可能日カレンダーを検索します。

**ヒント:** このサーバーをいつでも使用できるように設定する場合は、カレンダーを指定しないでください。

13.  をクリックします。
14. 「サーバーアクセスを追加」ページで、「OK」をクリックしてサーバーへのアクセスを WebDB に追加します。



### 5.3.6 ステップ 6. WebDB に Reports Server へのアクセスを追加する

レポート定義ファイル・アクセスでは、次の内容を WebDB で定義します。

- WebDB からアクセス可能な Reports Developer.RDF、.REP または .XML ファイル。
- このレポート定義ファイルを実行するアクセス権を持ったユーザー。
- このレポート定義ファイルを実行できる期間。
- 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 Reports Server およびプリンタの指定なども含みます。

この例では、次の情報に基づいて、WebDB の security.rdf ファイル  
(ORACLE\_HOME¥TOOLS¥DOC60¥US¥RBBR60 ディレクトリにある) へのアクセスを制限します。

- 宛先タイプをキャッシュおよびプリンタに制限する。
- 宛先フォーマットを HTMLCSS および PDF に制限する。
- P\_LASTNAME ユーザー・パラメータを WebDB に定義された値リストに制限する。
- P\_SSN ユーザー・パラメータを使用して社会保障番号と姓の組合せを検証する。
- COPIES システム・パラメータを使用して 2 コピーまでに制限し、ユーザーが印刷できるレポートの部数を 2 以下に制限する。
- USERID システム・パラメータを制限して、ユーザーがデータベース・ログイン情報を「ランタイム・パラメータ・フォーム」に保存できるようにする。ユーザーがレポートを自動的に実行するスケジュールを設定する場合は、USERID を制限付きパラメータとして指定する必要があります。

**ヒント:** 現在のウィザード・ページに関する状況判断ヘルプを参照するには  をクリックします。WebDB のヘルプ・システムにアクセスするには、タイトル・バーの  をクリックします。

### 5.3.6.1 Lastname ユーザー・パラメータの値リストを作成する手順

あらゆるシステム・パラメータまたは「オプション・パラメータ」ページで定義したユーザー・パラメータについて、ユーザーがそれらの値を値リストから選択できるようにする場合は、WebDB にリストを作成する必要があります。

security.rdf レポートは、企業の 401K プランに加入している従業員の投資に関する情報を収集することを思い出してください。このレポートへのアクセスを、プランに加入している従業員のみに限定します。この例では、従業員の姓をリストする P\_Lastname ユーザー・パラメータの値リストを作成します。

**ヒント:** レポート・リクエストを WebDB サイト上で公開しない場合は、WebDB に値リストを作成する必要はありません。値リストは、パラメータ・フォーム・エディタを使用して Report Builder で作成できます：



1. この作業に関する Report Builder 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. 「索引」ページで次のように入力します。  
パラメータ、値リスト
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。  
「パラメータの値リスト (LOV) の作成」

1. ナビゲーション・ツールバーで  をクリックします。
2. 「共有コンポーネント」メニューで、「値リスト (LOV)」をクリックします。

- 3. 「静的 - 静的な値」オプションを選択し、「LOV 作成」をクリックします。
- 4. 「静的値リスト作成」ページで、この値リストの所有スキーマで特定のスキーマを選択します。
- 5. 「権限」リスト・ボックスで「PUBLIC」を選択してすべてのユーザーがこの LOV にアクセスできるようにします。
- 6. LOV の名前に `LASTNAME_LOV` と入力します。  
**ヒント:** この LOV がすでに存在する場合は、末尾にあなたのイニシャルを付け足してください(たとえば、`LASTNAME_LOVAA`)。
- 7. 「コンボ・ボックス」をデフォルト・フォーマットとして選択します。
- 8. 表に次の値を入力します。

表 5-4 静的値リスト


表示値	戻り値	表示順序
Abers	Abers	1
Costner	Costner	2
Matsumoko	Matsumoko	3
Williams	Williams	4

- 9. 「LOV 追加」をクリックします。
- 10. 「値リスト管理」ページで、新規作成した LOV が「最近編集された値リストを選択」セクションに表示されます。LOV をテストする場合は、ここで実行できます。






5.3.6.2 レポート定義ファイルへのアクセスを追加する手順

**ヒント:** ウィザード・ページを表示しているときに前画面に戻った場合（つまり、Web ブラウザで「戻る」をクリックした場合）そのページで定義した設定値は消失します。変更する必要がある場合は、最初にウィザードを完了してレポートのパッケージ・プロシージャを作成します。次に、「コンポーネントの管理」ページで「編集」をクリックしてパッケージを編集します。




- 1. ナビゲーション・ツールバーで  をクリックし、「Oracle Reports セキュリティ」をクリックします。
- 2. 「Oracle Reports セキュリティ」ページで、「レポート定義ファイル」をクリックします。
- 3. 「新規作成」をクリックします。



4. 「レポート名およびスキーマ」ページで、「所有者」リスト・ボックスから特定のスキーマを選択します。選択したスキーマは、このレポートのパッケージ・プロシージャを所有することになります。
5. 「レポート名」フィールドに `Investment_Report` と入力します。  
 ヒント: レポート名の先頭に数字を使用することはできません (たとえば、`401K_report` は無効なファイル名であり、`my_401K_report` は有効です)。  
 ヒント: このレポート名がすでに存在する場合は、末尾にあなたのイニシャルを付け足してください (たとえば、`Investment_ReportAA`)。
6. 「Reports Server」リスト・ボックスで「`repserver`」を選択します。
7. Reports Developer ファイル名として `security.rdf` と入力します。  
 ヒント: Reports Server がこのレポート定義ファイルを見つけ出せるかどうかを確認してください。レポートのソース・パスは、Reports Server 構成ファイルの `SOURCEDIR` パラメータに設定するか、`REPORTS60_PATH` 環境変数に設定する必要があります。
8.  をクリックします。
9. 「ユーザーとロール」ページで、JABERS と Reports Developer システム管理者のユーザー・アカウントを「All Users and Roles」リスト・ボックスから選択して、このレポートを実行できるユーザーを指定します。  
 ヒント: プリンタと Reports Server へのアクセス権を与えたときと同じユーザーを選択する必要があります。
10.  をクリックして、このユーザーを「Selected Users and Roles」リスト・ボックスに移動します。
11.  をクリックします。
12. 「使用可能日カレンダー」ページで、使用可能日カレンダーとして `Production` と入力するか、または  をクリックして使用可能日カレンダーを検索します。  
 ヒント: このレポート定義ファイルをいつでも使用できるように設定する場合は、カレンダーを指定しないでください。
13.  をクリックします。
14. 「必須パラメータ」ページで、[Ctrl] キーを押しながら「タイプ」リスト・ボックスの「キャッシュ」と「プリンタ」をクリックします。
15. [Shift] キーを押しながら「書式」リスト・ボックスの「HTMLCSS」と「PDF」をクリックします。
16. 「プリンタ」リスト・ボックスから「`Reports_Printer`」を選択します。  
 ヒント: 定義したプリンタが表示されない場合は、プリンタ・アクセスを作成したときに、正しくない OS プリンタ名を入力した可能性があります。現在のレポート定義ファイル・パッケージの作成を終了してください。無効なパッケージが作成される可能性があります。


あります。「プリンタ・アクセス」ウィザードに戻り、Reports\_Printer のアクセスを編集します。プリンタ・アクセスを編集した後、「レポート定義ファイル」ウィザードに戻り、このレポートのレポート定義ファイル・アクセスを編集し、パッケージを新規作成します。

17. (オプション) 別のパラメータ・フォーム・テンプレートを選択します。選択したテンプレートによって、「ランタイム・パラメータ・フォーム」を表示するページ・スタイルが決定します。

18.  をクリックします。

19. 「オプション・パラメータ」ページで、「パラメータ名」列に P\_LASTNAME と入力します。実行時に、ユーザーがこのレポートを実行すると、姓を選択してレポートを実行するように要求されます。

**ヒント:** P\_LASTNAME は、レポートに定義されたパラメータ名です。Report Builder で security.rdf をオープンし、「パラメータ・フォーム」エディタにパラメータを表示してパラメータ名を特定します。

20. LOV 列に LASTNAME\_LOV と入力して、ユーザーが値リストから 401K 加入者の姓を選択できるようにします。または、 をクリックして LOV を検索します。


21. 「パラメータ名」列の 2 行目に P\_SSN と入力して、「ランタイム・パラメータ・フォーム」の社会保障番号の入力をユーザーに要求します。

22. 「パラメータ名」列の 3 行目に COPIES と入力して、レポートをプリンタに出力するときの印刷部数を制限します。

23. 「Low 値」列に 1 を入力します。

24. 「High 値」列に 2 を入力します。

25. 「パラメータ名」列の 4 行目に USERID と入力します。これにより、レポートを自動的に実行するスケジュールを設定するときに、接続先データベースの指定が可能になります。

26.  を 2 度クリックして「有効化トリガー」ページを省略して次へ進みます。

27. 「レポート定義ファイル・アクセスを追加」ページで、「OK」をクリックしてこのレポートのパッケージ・プロシージャを作成します。パッケージが作成されると、「コンポーネントの管理」ページが表示されます。このページでは、レポート・アクセスの編集、レポートの実行または「パラメータ・フォーム」のセットアップができます。次の例では、レポートの実行に使用する「パラメータ・フォーム」にデフォルト・パラメータ値を設定する方法を説明します。

**ヒント:** 無効なパッケージを作成した場合は、次のステップに進めません。プリンタ、Reports Server およびレポートに対して定義したアクセス・コントロールを検証してください。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生産パッケージを作成してください。

レポート定義ファイルへのアクセスを編集するには、ナビゲーション・ツールバーの



をクリックします。「Reports Developer セキュリティ」メニューで、「**レポート定義ファイル**」を選択します。次に、特定レポートの「コンポーネントの管理」ページにアクセスするために、そのレポートを検索するか、または「最近編集されたレポート定義ファイル」セクションでレポートを選択します。「コンポーネントの管理」ページで、「**編集**」をクリックします。

## 5.3.7 ステップ 7. 「Reports Developer パラメータ・フォーム」でパラメータ値を設定する


Reports Developer システム管理者としてログインし、先ほど作成したアクセス制限されたレポート・リクエストを実行して、期待どおりに実行されるかどうかを確認します。また、実行時にユーザーが利用できるデフォルト・パラメータを設定します。レポートの実行およびデフォルト・パラメータ値の設定は、「コンポーネントの管理」ページで行います。

### 5.3.7.1 レポート出力をキャッシュに保存する手順

次の 2 例では、パラメータ値を選択して、実行時にユーザーが利用できるデフォルト値を設定するのではなく、デバッグのためにレポートをキャッシュに保存します。デフォルト値の設定は、5.3.7.3 項「実行時にユーザーが利用するデフォルト・パラメータを設定する手順」で行います。

1. 「コンポーネントの管理」ページで、「**パラメータ**」をクリックしてデフォルト・パラメータを設定し、「ランタイム・パラメータ・フォーム」に表示するパラメータを選択します。



**ヒント:** 「コンポーネントの管理」ページにアクセスするには、 をクリックします。「Oracle Reports セキュリティ」メニューで、「**レポート定義ファイル**」を選択します。次に、レポートを検索するか、「最近編集されたレポート定義ファイル」セクションからレポートを選択します。

2. 「Reports Developer パラメータ・フォーム」で、次のパラメータを設定します。

表 5-5 キャッシュ出力をデバッグするためのパラメータ・フォームの設定

パラメータ	値
Server	repserver
Printer	Reports_Printer
Destype	Cache
Desformat	HTMLCSS

表 5-5 キャッシュ出力をデバッグするためのパラメータ・フォームの設定

パラメータ	値
Desname	空白
COPIES	1
P_LASTNAME	Abers
P_SSN	559014203
USERID	username/password@my_db ここで、username/password@my_db は接続先データベース のユーザー名とパスワードです。

注意: デバッグのためのパラメータ値を設定するとき、社会保障番号などの機密扱いのパラメータ値は、このパラメータ・フォームから必ず削除してください（または、保存しないでください）。そうしないと、このレポート・リクエストを WebDB サイトに追加したときに、この機密情報が公開されます。

3. 「レポートの実行」をクリックして要求したレポートを実行します。

5.3.7.2 （オプション）レポート出力をアクセス制限されたプリンタに送  
る手順

1. 出力をプリンタへ送信する場合は、「コンポーネントの管理」ページに戻り、「パラメータ」をクリックします。
2. 「Reports Developer パラメータ・フォーム」で、次のパラメータ値を選択します。

表 5-6 プリンタ出力をデバッグするためのパラメータ・フォームの設定

パラメータ	値
Server	repserver
Printer	Reports_Printer
Destype	Printer
Desformat	PDF
Desname	デフォルトでプリンタ名が設定される
COPIES	1
P_LASTNAME	Abers
P_SSN	559014203

表 5-6 プリンタ出力をデバッグするためのパラメータ・フォームの設定

パラメータ	値
USERID	username/password@my_db  ここで、username/password@my_db は接続先データベースのユーザー名とパスワードです。

3. 「レポートの実行」をクリックします。
4. レポートが正常に印刷されたことを示すメッセージが表示されたら、「OK」をクリックします。

### 5.3.7.3 実行時にユーザーが利用するデフォルト・パラメータを設定する手順

設定した制約条件に基づいてレポートが実行されることを確認できたら、デフォルト・パラメータ値を設定し、ユーザーが「ランタイム・パラメータ・フォーム」で利用できるパラメータを選択します。

1. 「Reports Developer パラメータ・フォーム」で、次のパラメータを設定します。

表 5-7 ユーザーのためのデフォルト・パラメータ設定

パラメータ	デフォルト値	ユーザーに表示する
Server	repserver	いいえ
Printer	Reports_Printer	いいえ
Destype	Cache	はい
Desformat	HTMLCSS	はい
Desname	空白	いいえ
COPIES	1	はい
P_LASTNAME	空白	はい
P_SSN	あなたの SSN を入力	はい
USERID	データベースのログイン ID を入力	はい

**ヒント:** 「ランタイム・パラメータ・フォーム」上でユーザーに表示されるパラメータは、リクエストを実行するときにユーザーが特定の処置をとる（つまり、値を選択または入力する）必要のあるもののみにします。この例では、Server および Printer パラメータは 1 つのサーバーおよびプリンタに限定されています。宛先タイプとしてプリン

タを選択すると、Desname パラメータにプリンタ名が自動的に設定されます。レポートを実行するときに、ユーザーはこれらのパラメータを入力する必要はありません。

WebDB サイトからレポートを実行するときに、ユーザーは「ランタイム・パラメータ・フォーム」に用意されたデフォルト・パラメータ値を各個人の作業環境に合わせて設定できます。詳細は、5.3.8 項「ステップ 8. ユーザーがレポートを利用できるようにする」を参照してください。

2. 「**パラメータを保存**」をクリックして「パラメータ・フォーム」に加えた変更を保存します。


## 5.3.8 ステップ 8. ユーザーがレポートを利用できるようにする

WebDB サイトでユーザーがレポートを利用できるようにするには、リンクを INVESTMENT\_REPORT パッケージ・プロシージャをポイントする WebDB コンポーネントとして追加します。

### 5.3.8.1 WebDB サイトを作成する手順

1. WebDB サイトをまだ作成していない場合は、このサイトを作成します。



1. このタスクに関する WebDB オンライン・ヘルプを表示するには、タイトル・バーでヘルプ・ボタンをクリックします。
2.  をクリックし、次のように入力します。  
web サイト
3. 「**検索**」をクリックし、次のトピックをクリックします。  
「Web サイトの作成」

**トピックが表示されます。**



---

**ヒント:** WebDB サイトを作成するには、Reports Developer システム管理者がサイト管理者の権限を持っている必要があります（つまり、「Grant」オプションによって SYS.DBMS\_SQL パッケージ・プロシージャの Execute 権限を持った DBA）。サイト管理者でない場合は、WebDB サイトの作成をサイト管理者に依頼してください。

### 5.3.8.2 WebDB サイトにフォルダを作成する手順

レポートのパッケージ・プロシージャを入れるフォルダを作成します。デフォルトでは、このフォルダおよびこれに追加されたすべてのアイテムは、フォルダの所有者（つまり、Reports Developer システム管理者）のみが利用できます。このフォルダをすべてのユーザー（つまり、PUBLIC ユーザー）に対して利用可能にすることも、またはアクセス権を与えられたユーザーのみに利用可能にすることもできます。この例では、このレポートを実行するアクセス権を持ったユーザー（つまり、JABERS）のみがフォルダにアクセスできるように設定します。


**ヒント:** フォルダをパブリックにすると、PUBLIC ユーザー（つまり、WebDB サイトにログインしていないユーザー）がレポートの「パラメータ / スケジュール」フォームにアクセスし、偶然に個人情報を保存することがあります。この機密情報は、後続の PUBLIC ユーザーに表示されます。これを防止するために、フォルダへのアクセスは、レポートを実行するアクセス権を持ったユーザーのみに制限するのが最適です。アクセスの制限されたフォルダへは、ユーザーがログインしなければアクセスできません。ログインが完了すると、「パラメータ / スケジュール」フォームに保存した情報は保護されます（つまり、そのユーザーのみに表示されます）。

**ヒント:** 現在のウィザード・ページに関する状況判断ヘルプを参照するには  をクリックします。WebDB のヘルプ・システムにアクセスするには、タイトル・バーの  をクリックします。

1. Web ブラウザから、WebDB サイトの URL を入力します。たとえば、次のように URL を入力します。



`http://my_webdb_server.com:1111/my_webdb_site`


**ヒント:** 管理者権限を持ち、サイトを作成できる場合は、ナビゲーション・ツールバー

から  をクリックして「サイト」ページにアクセスします。「サイトのホーム・ページ」をクリックして WebDB サイトにアクセスします。

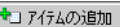
2. Reports Developer システム管理者としてログインします。

**ヒント:** WebDB コンポーネントを WebDB サイトに追加するには、サイト管理者の権限を持った Reports Developer システム管理者ユーザー・アカウントが必要です（つまり、「Grant」オプションによって SYS.DBMS\_SQL パッケージ・プロシージャの Execute 権限を持った DBA）。サイト管理者としての権限がない場合は、コンポーネントを追加するフォルダについて、所有、アイテムの管理または承認付きで作成権限が必要です。詳細は、DBA またはサイト管理者に相談してください。

3. WebDB サイト・ホームページで、 をクリックします。
4.  フォルダの追加 をクリックして新しいフォルダをサイトに追加します。
5. 「フォルダ・マネージャ」ページで、新規フォルダの内部フォルダ名として Benefits と入力します。
6. WebDB サイトに表示されるフォルダのタイトルとして Benefits と入力します。  
**ヒント:** Benefits フォルダがすでに別のユーザーによって作成されている場合は、フォルダ名の末尾にあなたのイニシャルを付け足します（たとえば、BenefitsAA）。
7. 「作成」をクリックしてフォルダを作成します。
8. リスト・ボックスから「Benefits」を選択します。

9. 「編集」をクリックします。
10. 「ユーザー」タブをクリックします。
11. 「Benefits」ページで、このフォルダへのアクセス権を持つユーザー名として JABERS と入力します。
12. 「アクセス・リストに追加」をクリックします。JABER が表示権限を持ったユーザーとしてユーザー・アクセス・リストに掲載されたことに注意してください。このデフォルトを保存します。
13.  をクリックして「Benefits」フォルダに戻ります。



#### 5.3.8.3 レポート・リクエストをフォルダに追加する手順

1. 「Benefits」で、 をクリックしてアイテム・ウィザードにアクセスし、レポート・リクエストをこのフォルダに追加します。
2. 「アイテムを追加」ページで、「WebDB コンポーネント」をアイテム・タイプとして選択します。
3. 「標準アイテム」を表示オプションとして選択します。
4. 「次へ」をクリックします。
5. 「WebDB コンポーネント」ページで、リスト・ボックスから WEBDB.INVESTMENT\_REPORT を選択します。ここで、WEBDB は、401K レポートについて、このレポートのパッケージ・プロシージャを所有するスキーマ名です。
6. タイトルとして Investment Summary Report と入力します。
7. カテゴリとして「General」を選択します。
8. 「説明」テキスト・ボックスに Restricted 401K Report と入力します。
9. 「次へ」をクリックします。
10. 2 番目の「WebDB コンポーネント」ページで、「パラメータ・フォームの表示」オプションを選択します。
11. 「完了」をクリックします。レポート・リクエストの入ったパッケージ・プロシージャへのリンクが Benefits フォルダに表示されます。

#### 5.3.8.4 ユーザーの立場からこのレポートを実行する手順

このレポートを、Reports Developer システム管理者としてではなく、JABERS として実行します。この例では、Jeff Abers 用のデフォルト・パラメータ値を設定し、レポートを実行します。



**ヒント:** 現在のウィザード・ページに関する状況判断ヘルプを参照するには  をクリックします。WebDB のヘルプ・システムにアクセスするには、タイトル・バーの  をクリックします。

1. ナビゲーション・バーの「**ログオフ**」をクリックして Reports Developer システム管理者としてログオフします。
2. 「**ログイン**」をクリックして JABERS としてログインします。
3. 「**サイトマップ**」をクリックして「Benefits」フォルダにアクセスします。
4. 「Benefits」フォルダをクリックします。
5. 「Benefits」フォルダの「Investment Summary Report」をクリックします。
6. 「パラメータ / スケジューリング」ページで、次のパラメータを選択します。

**表 5-8 ユーザーのデフォルト・パラメータ設定**

パラメータ	デフォルト値
Destype	Cache
Desformat	HTMLCSS
COPIES	1
P_LASTNAME	Abers
P_SSN	559014203
USERID	username/password@my_db ここで、username/password@my_db は接続先データベースのユーザー名とパスワードです。

7. 「**パラメータを保存**」をクリックしてあなた専用の設定値を保存します。

ここで保存したデフォルト値は、このユーザーのみがアクセスできる設定値であることに注意してください。あなた（または別の人）が別のユーザーとしてログインした場合は、Reports Developer システム管理者が定義したデフォルト設定が表示されます。この場合、ユーザーは自分の設定値を保存して自分専用のものにできます。

8. 「**レポートの実行**」をクリックします。

### 5.3.9 ステップ 9. レポートを実行して出力を WebDB サイトに配信するスケジュールを設定する



Jeff Abers 氏が自分の 401K 投資を毎月 1 度ずつ再検討する場合を想定します。また、このレポートを自動的に実行し、毎月最後の金曜日の午前 9 時までに Jeff 氏専用のフォルダに配信


するとします。最初に、Jeff 氏の個人フォルダ（彼のユーザー・アカウントのみアクセスが許可されるフォルダ）を作成します。次に、レポートを自動的に実行するスケジュールを設定します。

**前提条件:** 5.3.8 項「ステップ 8. ユーザーがレポートを利用できるようにする」の例がすでに完了している必要があります。

### 5.3.9.1 個人フォルダを作成する手順

個人的なレポートが指定したユーザーのみにアクセスされることを確保するには、自分専用の個人フォルダを作成します（この例では、JABERS としてログインします）。



**ヒント:** 現在のウィザード・ページに関する状況判断ヘルプを参照するには  をクリックします。WebDB のヘルプ・システムにアクセスするには、タイトル・バーの  をクリックします。

1. Web ブラウザから WebDB サイトにアクセスし、まだログインしていない場合は JABERS としてログインします。
2. ナビゲーション・バーから「**管理**」をクリックします。
3. 「アクセス・マネージャ」セクション下で、「**個人情報**」をクリックします。
4. 「個人用フォルダの作成」ボックスをオンにします。「個人用フォルダの作成」ボックスが表示されない場合は、作成済みの個人フォルダがすでに存在します。
5. 必要に応じて個人情報を入力します。
6.  をクリックします。

これで、あなたは個人フォルダの所有者になりました。許可を与えないかぎり、他のユーザーはこのフォルダにいったいアクセスできません。これで、401K レポートを自動的に実行し、JABERS 個人フォルダに配信するためのスケジュールを設定できます。

### 5.3.9.2 レポートをスケジュールする手順

この例では、毎月最後の金曜日午前 9 時にレポートを実行するようにスケジュールします。また、2 か月分の 401K の結果を履歴に保存します。

**ヒント:** 現在のウィザード・ページに関する状況判断ヘルプを参照するには  をクリックします。WebDB のヘルプ・システムにアクセスするには、タイトル・バーの  をクリックします。

1. ナビゲーション・バーから「**サイトマップ**」をクリックします。

**ヒント:** 必ず JABERS としてログインしてください。

2. 「Benefits」フォルダをクリックします。

3. 「Investment Summary Report」をクリックします。デフォルト・パラメータをまだ設定していない場合は、5.3.8.4 項「ユーザーの立場からこのレポートを実行する手順」に戻り、デフォルト・パラメータを設定します。
4. 「パラメータ / スケジューリング」ページで、「スケジュール」をクリックします。
5. 今日の日付の午前 9 時にジョブを開始するように選択します。  
**ヒント:** 今すぐレポートを実行する場合は、「即時」オプションを選択します。
6. 次の「繰り返し」オプションを選択します。  
 毎月、30 日またはそれ以前の最後の**金曜日**。  
**ヒント:** 月末にレポートが実行されるまで待つかわりに、 $n$  時間ごとに繰り返すように、繰り返しオプションを設定します。レポート出力が結果フォルダに正常に配信されることが確認できたら、繰り返しパターンを再設定します。
7. 次の出力先オプションを設定します。

表 5-9 出力先の設定

フィールド	値
サイト	結果フォルダが置かれた WebDB サイトの名前
ログ・ファイル・フォルダ	JABERS
結果タイトル	My 401K Report
結果フォルダ	JABERS
前回の結果を上書きする	オフ
有効期間	60 日

注意: ログ・ファイル・フォルダと結果フォルダの名前は、大文字と小文字が区別されます。レポート出力とステータス情報を既存フォルダに配信する場合は、正確なフォルダ名を入力する必要があります。入力したフォルダ名に誤りがあると、WebDB はそれを検索できず、デフォルトにより、指定したフォルダが Reports Developer 出力フォルダおよび Reports Developer ステータス・フォルダに追加されます。デフォルトで、これらのフォルダにはパブリック・アクセス権が与えられます（つまり、すべてのユーザーがあなたの個人レポートを表示できます）。フォルダの定義は慎重に行ってください。

8. 「**送る**」をクリックします。レポートが正常にスケジュールされたことを示すメッセージが表示されます。

**ヒント:** レポートをすぐ実行するようにスケジュールした場合、レポート出力がブラウザに表示されます。

9. 「OK」をクリックします。ジョブはスケジュールされた日付、時刻および繰り返しパターンに基づいて実行されます。

### 5.3.9.3 配信されたレポート出力を表示する手順

1. ナビゲーション・バーから「**サイトマップ**」をクリックします。

**ヒント:** JABERS としてログインしたまま続けます。

2. 「JABERS」をクリックしてフォルダをオープンします。
3. 「My 401K Report」をクリックしてレポートを表示します。

レポート自体のリンクの他に、レポートについてのステータス情報のリンクも利用できる点に注意してください。このステータス・リンクは、スケジュールしたレポートの実行で発生したあらゆる問題の解決に役立ちます。表示されるステータスの詳細は、ユーザーに割り当てられた Reports Developer ロール（たとえば、RW\_BASIC\_USER など）によって異なります。レポートのスケジュールおよび実行がうまくいかない場合は、Reports Developer システム管理者に相談してください。

## オプション問題

人事部の責任者があなた（Reports Developer システム管理者）に、従業員全員が利用できる株式レポートの作成を依頼した場合を想定します。従業員が特定の株式の状況を監視できるように、この株式レポートを毎朝自動的に実行します。このレポートをパブリック・フォルダに配信して、すべての従業員が表示できるようにします。

この章の例を参照して、template.rdf レポート

(ORACLE\_HOME¥TOOLS¥DOC60¥US¥RBBR60 ディレクトリ) へのアクセスを WebDB に追加します。

**ヒント:** このレポートを実行し、パブリック・フォルダに配信するようにスケジュールするのはあなた（Reports Developer システム管理者）ですから、Reports Developer システム管理者のみがこのレポートにアクセスできれば十分です。

このレポートのパッケージ・プロシージャを WebDB コンポーネントとして「Benefits」フォルダに追加します。次に、このレポートを毎朝 10 時に実行して出力を「Stocks」という新規フォルダに配信するようにスケジュールします。

**ヒント:** 「Stock」フォルダは PUBLIC ユーザーに表示されるように設定する必要があります。

## 5.3.10 まとめ

アクセス・コントロールに対応した Reports Server の構成が正常に終了しました。この章で習得した内容は、次のとおりです。

- 使用可能日カレンダーを作成して、レポート定義ファイル、Reports Server およびプリンタを使用して処理できる時間を決定する。

- アクセスできるユーザーおよび処理できる時間を限定して、プリンタ、Reports Server およびレポート定義ファイルへのアクセスを WebDB に追加する。
- パッケージ・プロシージャとして格納されたレポート・リクエストを WebDB コンポーネントとして WebDB サイトに追加する。
- ユーザーが自分専用のデフォルト・パラメータを設定し、WebDB サイトからレポート・リクエストを実行できるようにする。
- レポートを自動的に実行し、結果を個人フォルダに配信するためのスケジュールを設定する。



## Reports Server のクラスタの構成

この章では、Reports Server をクラスタに構成して、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。

Reports Server のクラスタリングを用いると、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することができます。これによって、Reports Server 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 Reports Server を構成する前に、Reports Server の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 2 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 Reports Server 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「Reports Server の構成」を参照してください。

### 6.1 クラスタリングの概要

3 つのマシンがクラスタ化する Reports Server として構成されていると仮定します。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、Reports Server をクラスタに構成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「Reports Server をクラスタに構成する例」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 と MINENGINE には、各マシン上で使用できる CPU 数が設定されています。

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（例えば、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。

**注意：**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なら、当然、Reports Server として完全に機能できる状態を維持することができます。NT-2 の Reports Server 構成の MAXENGINE パラメータと MINENGINE パラメータが 3 に設定されているとします。これは、NT-2 の Reports Server 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ることができ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 Reports Server 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 つとマスター専用のスレーブ・エンジンが 2 つです。

## 6.2 Reports Server をクラスタに構成する例

この項では、Reports Server をクラスタに構成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 Server コンポーネントがインストールされている。
- Reports Server 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。Reports Server の構成の詳細は 3.2 項「Reports Server の構成 : Reports Web CGI の例」参照。
- セントラル・ファイル・サーバーが稼働していて、2つのディレクトリが設定されている。Source ディレクトリ (レポート定義が格納されている) と cache 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。

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別名を認識する (出力が常にデフォルトのプリンタに送られているのでない限り)。

## 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:

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

ここで、`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・インスタンス名の `.world` を省略してください。

NT-2 マシン (スレーブ) 上:

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=(ADDRESS=(PROTOCOL=tcp) (HOST=nt-1) (PORT=1949))
```

ここで、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。sqlnet.ora に NAMES.DEFAULT\_DOMAIN 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 .world は省略してください。

2. ORACLE\_HOME¥REPORT60¥SERVER ディレクトリの nt-2.ora (Reports Server の構成ファイル) をオープンして、INITEGINE 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。
  - エンジンの設定およびキャッシュとソース・ディレクトリの識別子。
  - 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 4096 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ている点に注意してください。

NT-1 サーバー・マシン (マスター) 上:

1. ORACLE\_HOME¥REPORT60¥SERVER ディレクトリの nt-1.ora (Reports Server 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
```

**注意:** NBT-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。

**server** はスレーブ・サーバーの TNS サービス・エントリ名です。

**minengine** は、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・エンジンの最小数です。

**maxengine** は、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・エンジンの最大数です。

**initengine** は、このマスター・エンジンが起動したランタイム・エンジンの初期数です。

**cachedir** は、このマスター・サーバーに対するセントラル・キャッシュ・ディレクトリです。

**注意:**

- 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・システムを同じ定義によって認識する必要はありません（マスターは「X: ドライブ」にマップされ、スレーブは「W: ドライブ」にマップされます）。
- 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。

4. 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 Reports Server 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、他のクライアントの実行ファイルに対する適切な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。

**ヒント:** 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 C「環境変数」を参照してください。

使用できるさまざまなレポート・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の再送信を行う方法」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 Queue Viewer を使用して監視することができます。詳細は 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の再送信を行う方法

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

**注意:** この演習では、このマシンはすでに Reports Server として構成されているものとし  
ます。Reports Server に対する TNS サーバー・エントリ名はマシン名です。

NT-2 サーバー・マシン (スレーブ) 上:

ORACLE\_HOME¥REPORT60¥SERVER ディレクトリにある sun-2.ora (Reports Server 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=(ADDRESS=(PROTOCOL=tcp) (HOST=nt-1) (PORT=1949))
```

ここで、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 の設定で指定されたドメインです。  
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・インスタンス名の .world を省略してください。

NT-1 サーバー・マシン (マスター) 上:

1. ORACLE\_HOME¥NET80¥ADMIN ディレクトリにある tnsnames.ora ファイルをオープン  
して次のエントリを追加します。

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

ここで、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 の設定で指定されたドメインで  
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・インスタンス名の .world を省略してください。

2. nt-1.ora (Reports Server 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、Reports Server を NT-1 マシン上で再起動しました。NT-1 マシンは 2 つの Sun マシン上でスレーブ・エンジンを起動することができましたが、NT-2 マシンはダウンしているので NT-2 マシン上のスレーブ・マシンを起動することはできませんでした。

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ので、NT-2 マシンが回復して Reports Server 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ることができるようになります。

## 実行時のレポートのカスタマイズ

Oracle Reports Developer 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。以前は、Oracle Reports Developer で実行するレポートは Report Builder で作成し、保存することが必要でした。これからは、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。

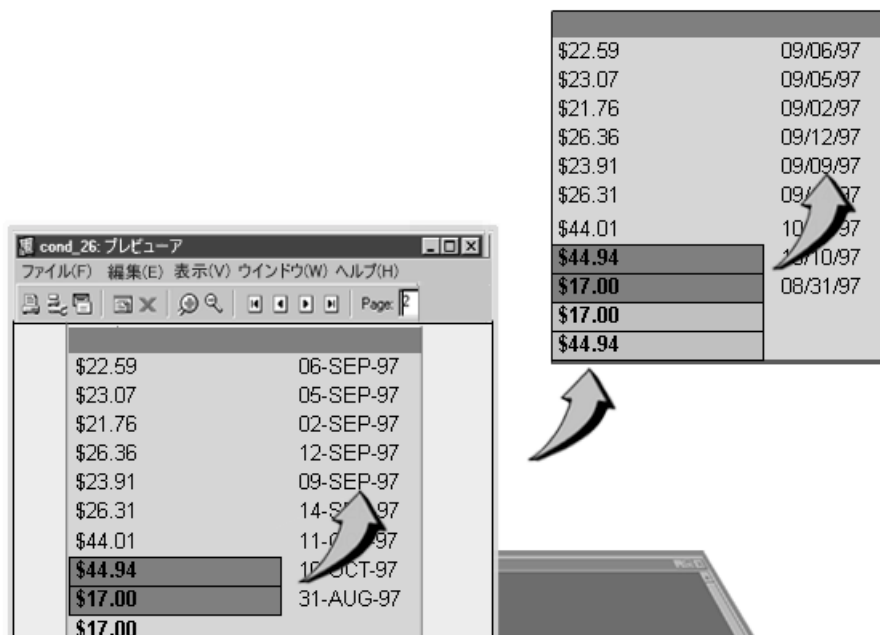
- オリジナルのレポートを変更せずに、実行時にレポートをカスタマイズすることができます。さまざまな XML レポート定義を作成して適用することで、各ユーザーまたはユーザー・グループ単位でレポートの出力を変更できます。この機能の利点は、同じレポートを使用して利用対象者ごとに異なる出力を生成できる点です。
- 既存レポートにバッチ更新を適用します。XML レポート定義を他のレポートに適用した場合は、結合した定義をファイルに保存できます。したがって、XML レポート定義を使用して、既存レポートのバッチ更新ができます。この利点は、ファイルを 1 つずつ Report Builder でオープンして手動で変更しなくても、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、Report 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 Developer 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。次の図で示すとおり、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。

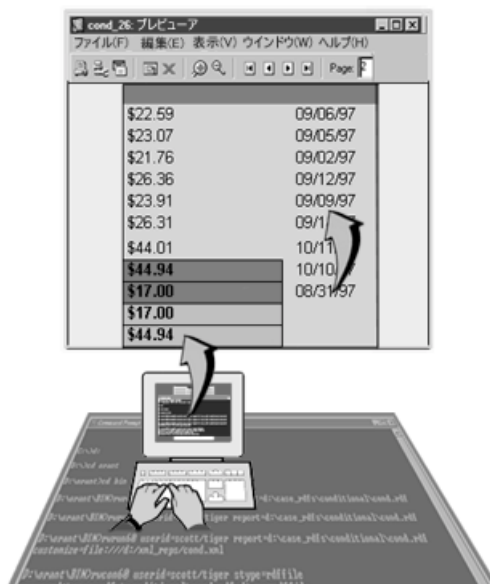


The diagram illustrates the application of a customization file to a report. It shows a preview window titled 'cond\_26: プレビュー' with a menu bar (File, Edit, View, Window, Help) and a toolbar. The main area displays a table with two columns: monetary values and dates. A magnifying glass highlights a specific row, which is then shown in a larger, detailed view to the right.

\$22.59	06-SEP-97
\$23.07	05-SEP-97
\$21.76	02-SEP-97
\$26.36	12-SEP-97
\$23.91	09-SEP-97
\$26.31	14-SEP-97
\$44.01	11-OCT-97
<b>\$44.94</b>	15-OCT-97
<b>\$17.00</b>	31-AUG-97
<b>\$17.00</b>	
<b>\$44.94</b>	

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。次の図に示すとおり、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。<sup>1</sup> 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>
```

<sup>1</sup> 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。

**ヒント:** 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、単独では実行できないことに注意してください。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 Reports Server にアクセス可能な場所に格納します。<sup>1</sup>
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 (すなわち .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。

### 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>`

---

<sup>1</sup> XML レポート定義は、Reports Runtime および Report Builder でも使用できることに注意してください。

たとえば、最小限の XML レポート定義は次のとおりです。<sup>1</sup>

```
<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。<sup>2</sup> </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。<sup>3</sup> 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、それにさまざまな部分 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』マニュアルでは、Report 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¥RBBR60 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。

<sup>1</sup> 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つさいないことに注意してください。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のに役立つのみです。

<sup>2</sup> 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 Developer には、レポート定義で利用できる 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。

<sup>3</sup> .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。

表 7-1 レポート作成用の XML レポート定義

XML ファイル	.RDF 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"><li>■ F_trade_date の書式マスクを MM/DD/RR に変更。</li><li>■ F_Mincurrent_pricePersymbol および F_Maxcurrent_pricePersymbol の塗りカラーを変更。</li></ul> <p>cond.xml による追加：</p> <ul style="list-style-type: none"><li>■ HTML 出力生成時にレポート・エスケープの HTML を挿入。</li></ul> <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"><li>■ F_high_365 および F_low_365 のフィールド・ラベル。</li></ul> <p>temp.xml による追加：</p> <ul style="list-style-type: none"><li>■ 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。</li><li>■ 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。</li></ul> <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"><li>■ レポートにプログラム単位を追加。</li><li>■ 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。</li><li>■ 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。</li></ul> <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 レポート作成用の XML レポート定義

XML ファイル	.RDF ファイル	説明
ref.xml	ref.rdf	ref.xml による追加： <ul style="list-style-type: none"><li>■ 新規問合せ Q_summary をデータ・モデルに追加。</li><li>■ 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。</li></ul> 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。たとえば、次のように URL を入力します。

```
rwrun60 userid=scott/tiger report=cond.rdf
customize=e:¥orant¥tools¥doc60¥us¥rbr60¥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のに使用される XML を詳しく説明します。

7.2.2.1 書式変更の例

cond.xml の XML は、cond.rdf 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

- 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ることに注意してください。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ない

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 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:¥orant¥tools¥doc60¥us¥rbbrr60¥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

- 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ることに注意してください。

```

<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

- 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> タグ内で作成されることに注意してください。
- 関数は <field> タグの formatTrigger 属性から名前で参照します。
- 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。

```
rwrun60 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_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>
</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

- この XML レポート定義は、単独で実行することも、ref.rdf に適用することも可能であることに注意してください。単独で実行できるのは、このレポート定義がデータ・モデルと完全なレイアウトの両方を持っているからです。
- もう 1 つの重要な点は、SELECT 文の中で別名が使用されている点です。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、Report 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

- この XML レポート定義は完全であり、単独で実行できることに注意してください。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ることに注意してください。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、Report Builder にレポートを発行し、Report Editor にデータ・モデル・ビューを表示します。

```
rwbl60 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のハイパーリンクが設定されています。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 1 つ 1 つ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="corp1.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="LNNNGNNNGNNNGNN0D00">
    <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="LNNNGNNNGNNNGN0D00">
        <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="LNNNGNNNGNNNGN0D00">
        <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="LNNNGNNNGNNNGN0D00">
        <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="LNNNGNNNGNNNGNND00">
            <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="LNNNGNNNGNNNGNND00"/>
        <field name="f_total_cost" source="total_cost" label="Total Cost"
            font="Arial" fontSize="8" lineColor="noLine"
            formatMask="LNNNGNNNGNNNGNND00"/>
        <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 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 ファイルに適用する。
- 7.3.2 項「XML レポート定義を単独で実行する」: 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）実行する。
- 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する。

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、RW60CLI、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 Reports Server 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
```

Reports Runtime を使用する場合は、次のとおりです。

```
rwrn60 userid=username/password@mydb report=emp.rdf
customize=e:¥myreports¥emp.xml destype=file desname=emp.pdf
desformat=PDF
```

**ヒント:** 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。たとえば、次のように URL を入力します。

```
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。次のコマンド・ラインにより、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・リクエストが Reports Server に送信されます。

```
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
```

Reports Runtime を使用する場合は、次のとおりです。

```
rwrn60 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、Before Form または After Form 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 Before Form または After Form 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¥rbb60¥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 After Parameter Form 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 の After Parameter Form トリガーに入っている PL/SQL が実行する内容は、次のとおりです。

- 実行時にユーザーによって入力されたパラメータ値に基づいて、条件と合致したフィールドをハイライトする。
- 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する。

### この例のヒント

- 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ることに注意してください。
- 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 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1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="LNNNNNNNNNNN0D00"/>');
SRW.ADD_DEFINITION(' <field name="F_TOTAL_SALES"
source="TOTAL_SALES" formatMask="LNNNNNNNNNNN0D00"/>');
SRW.ADD_DEFINITION(' <field name="F_TOTAL_COST"
source="TOTAL_COST" formatMask="LNNNNNNNNNNN0D00"/>');
SRW.ADD_DEFINITION(' <field name="F_SumTOTAL_PROFITPerCITY"
source="SumTOTAL_PROFITPerCITY" formatMask="LNNNNNNNNNNN0D00"/>');
SRW.ADD_DEFINITION(' <field name="F_SumTOTAL_SALESPerCITY"
source="SumTOTAL_SALESPerCITY" formatMask="LNNNNNNNNNNN0D00"/>');
SRW.ADD_DEFINITION(' <field name="F_SumTOTAL_COSTPerCITY"
source="SumTOTAL_COSTPerCITY" formatMask="LNNNNNNNNNNN0D00"/>');
SRW.ADD_DEFINITION(' </section>');
SRW.ADD_DEFINITION(' </layout>');
ELSIF :MONEY_FORMAT=' $NNNN' THEN
SRW.ADD_DEFINITION(' <layout>');
SRW.ADD_DEFINITION(' <section name="main">');
SRW.ADD_DEFINITION(' <field name="F_TOTAL_PROFIT"
source="TOTAL_PROFIT" formatMask="LNNNNNNNNNNN0"/>');
SRW.ADD_DEFINITION(' <field name="F_TOTAL_SALES"
source="TOTAL_SALES" formatMask="LNNNNNNNNNNN0"/>');
SRW.ADD_DEFINITION(' <field name="F_TOTAL_COST"
source="TOTAL_COST" formatMask="LNNNNNNNNNNN0"/>');
SRW.ADD_DEFINITION(' <field name="F_SumTOTAL_PROFITPerCITY"
source="SumTOTAL_PROFITPerCITY" formatMask="LNNNNNNNNNNN0"/>');
SRW.ADD_DEFINITION(' <field name="F_SumTOTAL_SALESPerCITY"
source="SumTOTAL_SALESPerCITY" formatMask="LNNNNNNNNNNN0"/>');
SRW.ADD_DEFINITION(' <field name="F_SumTOTAL_COSTPerCITY"
source="SumTOTAL_COSTPerCITY" formatMask="LNNNNNNNNNNN0"/>');
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 レポート定義を単独で実行する

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 Reports Server 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
```

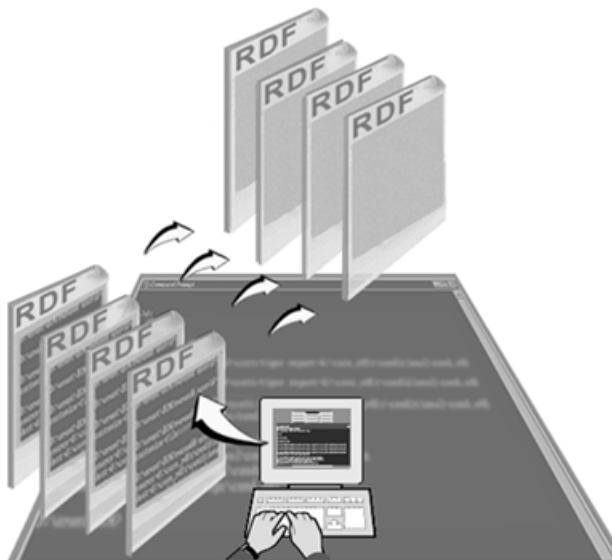
Reports Runtime を使用する場合は、次のとおりです。

```
rwrun60 userid=username/password@mydb  
report=e:¥corp¥myreports¥emp.xml  
destype=file desname=emp.pdf desformat=PDF
```

**ヒント:** 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できることに注意してください。

### 7.3.3 バッチ変更の実行

次図に示すとおり、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）。Report 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 inven1.rdf、inven2.rdf および manu.rdf に適用し、変更後の定義を新しいファイル inven1\_new.rdf、inven2\_new.rdf および manu\_new.rdf に保存します。

```
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。たとえば、次のように URL を入力します。

```
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 seconds
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、Report Builder でオープンした方が便利な場合があります。Report 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、Report 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。

Report 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。たとえば、次のように URL を入力します。

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

Report 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。たとえば、次のように URL を入力します。

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

どちらの場合も、Report Builder で有効な XML レポート定義がオープンします。その後、Report Editor のいろいろなビュー（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ることに注意してください。たとえば、次のように URL を入力します。

```
custom_summary Text_IO.File_Type;
```

また、書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込み先ファイルを渡す必要があります。たとえば、次のように URL を入力します。

```
custom_summary := Text_IO.Fopen(:file_directory || 'vid_summ_per.xml', 'w');
addaline('<report name="video_custom" author="Generated" DTDVersion="1.0">',
        custom_summary);
```

## 7.5 XML タグ・リファレンス

Oracle Reports Developer 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。

**ヒント：** 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めます。

**構文：**

```
<!--
    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\RB60.
      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ことがある 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:**

```
<![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、次の項を参照してください。

7.5.7 項「<exception>」

**構文:**

```
<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。 <ul style="list-style-type: none"><li>■ eq(equal)</li><li>■ lt ( less than )</li><li>■ lteq(less than or equal)</li><li>■ neq(not equal)</li><li>■ gt ( greater than )</li><li>■ gteq(greater than or equal)</li><li>■ btw ( between )</li><li>■ notBtw ( not between )</li><li>■ like</li><li>■ notLike</li><li>■ null</li><li>■ notNull</li></ul>

表 7-2 <condition> タグの属性

属性	必須 / 任意	説明
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、オペランドは不用です。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"><li>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。</li><li>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。</li></ul>

**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：** <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:**

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

**例:** 次の例では、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> 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 Developer (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:**

```
<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：

```
<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"><li>■ 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。</li><li>■ 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。</li></ul>
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。 <ul style="list-style-type: none"><li>■ regular（標準）</li><li>■ italic（斜体）</li><li>■ bold（太字）</li><li>■ boldItalic（太字斜体）</li></ul>



表 7-3 &lt;exception&gt; タグの属性

属性	必須 / 任意	説明
fontEffect	任意	条件を満たした場合に適用するフォントの効果。 <ul style="list-style-type: none"> <li>regular (標準)</li> <li>strikeout (取消し線)</li> <li>underline (下線)</li> <li>strikeoutUnderline (取消し線と下線)</li> </ul>

- 使用上の注意：**
- 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。

**ヒント:** <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 | strikeout | underline | strikeoutUnderline"]
  [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 &lt;field&gt; 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、&lt;labelAttribute&gt; タグを使用する必要があります。詳細は、7.5.15 項「&lt;labelAttribute&gt;」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、&lt;object&gt; タグを使用します。詳細は、7.5.22 項「&lt;object&gt;」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るので注意してください。
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ることができます。
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・トリガーは関数でなければならないことに注意してください。詳細は、Report 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。</p> <ul style="list-style-type: none"> <li>■ regular (標準)</li> <li>■ italic (斜体)</li> <li>■ bold (太字)</li> <li>■ boldItalic (太字斜体)</li> </ul>

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。 <ul style="list-style-type: none"><li>■ regular (標準)</li><li>■ strikeout (取消し線)</li><li>■ underline (下線)</li><li>■ strikeoutUnderline (取消し線と下線)</li></ul>
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、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。 <ul style="list-style-type: none"><li>■ start (始点詰め)</li><li>■ left (左揃え)</li><li>■ center (中央揃え)</li><li>■ right (右揃え)</li><li>■ end (終点詰め)</li></ul>
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、Report 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="LNNNGNNNGNNNGN0D00">
  <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="LNNNGNNNGNNNGN0D00">
  <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="LNNNGNNNGNNNGN0D00">
  <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ーク・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ることに注意してください。

```

<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:**

```
<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、次の項を参照してください。

7.5.29 項「<summary>」

**構文:**

```
<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 &lt;formula&gt; 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 関数の名前です。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"> <li>■ number (数値)</li> <li>■ character (文字)</li> <li>■ date (日付)</li> </ul>
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る点に注意してください。

```

<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：

```
<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ることに注意してください。

```
<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>

```

## 7.5.12 <group>

**説明:** <group> タグは、マスター / ディテール・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:**

```

<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="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>
```

## 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:**

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

**例:**

次の例では、グループ上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"/>
    <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、グループ左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:**

```
<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ることに注意してください。

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:**

```
<labelAttribute
  [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"]
>
</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。 <ul style="list-style-type: none"> <li>■ regular (標準)</li> <li>■ italic (斜体)</li> <li>■ bold (太字)</li> <li>■ boldItalic (太字斜体)</li> </ul>
fontEffect	任意	フィールドの内容に使用するフォントの効果。 <ul style="list-style-type: none"> <li>■ regular (標準)</li> <li>■ strikethrough (取消し線)</li> <li>■ underline (下線)</li> <li>■ strikethroughUnderline (取消し線と下線)</li> </ul>
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、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。

表 7-7 <labelAttribute> タグの属性

属性	必須 / 任意	説明
alignment	任意	フィールド内の文字位置を揃える方法。 <ul style="list-style-type: none"><li>■ start ( 始点詰め )</li><li>■ left ( 左揃え )</li><li>■ center ( 中央揃え )</li><li>■ right ( 右揃え )</li><li>■ end ( 終点詰め )</li></ul>

**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ることに注意してください。

```
<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：** <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ることに注意してください。

```
<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えることに注意してください。

```
<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="LNNNGNNNGNNNGN0D00">
<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="LNNNGNNNGNNNGNNOD00"/>
<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> 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用します。）

**構文：**

```
<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。

表 7-8 <link> タグの属性

属性	必須 / 任意	説明
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。  eq（等しい）  lt（より小さい）  lteq（より小さいか、または等しい）  neq（等しくない）  gt（より大きい）  gteq（より大きいか、または等しい）  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。

**ヒント:** <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:**

```

<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="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>
  <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="LNNNGNNNGNNNGNND00">
      <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="LNNNGNNNGNNNGN0D00"/>
</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:**

```

<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="LNNNGNNNGNNNGNNOD00"/>
  <field name="f_total_cost" source="total_cost" label="Total Cost"
    font="Arial" fontSize="8" lineColor="noLine"
    formatMask="LNNNGNNNGNNNGNNOD00"/>
  <field name="f_total_profit" source="total_profit" label="Total Profit"
    font="Arial" fontSize="8" lineColor="noLine"
    formatMask="LNNNGNNNGNNNGNNOD00"/>
</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:**

```
<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="LNNNGNNNGNNNGNNOD00">
    <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="LNNNGNNNGNNNGNNOD00">
    <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>

```

## 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:**

```

<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:**

```

<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"><li>■ REP_REPORT はレポート自体です。</li><li>■ REP_GROUP はレポートのデータ・モデル内のグループです。</li><li>■ REP_COL_MAP はレポートのデータ・モデル内の列です。</li><li>■ 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。</li></ul>

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、次の項を参照してください。

7.5.11 項「<function>」

**構文:**

```

<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ることに注意してください。

```

<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>
</programUnits>

```

```
        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、通常はその中に <property> タグをネストさせます。

**構文:**

```
<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>

```

## 7.5.25 <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：**

```

<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・タイプごとに異なることに注意してください。詳細は <b>使用上の注意</b> 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)	<div><div>■ xmlTag</div><div>■ xmlAttribute</div><div>■ xmlSuppress</div><div>■ prologType</div><div>■ prolog</div><div>■ beforeReportValue</div><div>■ beforeReportType</div><div>■ afterReportValue</div><div>■ afterReportType</div><div>■ beforePageValue</div><div>■ beforePageType</div><div>■ afterPageValue</div><div>■ afterPageType</div><div>■ beforeFormValue</div><div>■ beforeFormType</div><div>■ afterFormValue</div><div>■ afterFormType</div><div>■ pageNavigationControlValue</div><div>■ pageNavigationControlType</div></div>
グループ・オブジェクト (REP_GROUP)	<div><div>■ xmlTag</div><div>■ xmlAttribute</div><div>■ outerXMLTag</div><div>■ outerXMLAttribute</div><div>■ xmlSuppress</div></div>
列オブジェクト (REP_COL_MAP)	<div><div>■ xmlTag</div><div>■ xmlAttribute</div><div>■ XMLSuppress</div><div>■ containXML</div></div>
バイラプレート・オブジェクト (REP_GRAPHIC_TEXT)	<div><div>■ textSegment</div></div>

**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、ボイラースキートン・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:**

```
<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ないことに注意してください。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¥RBBR60.
      If it it 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>.
            </font>
          </center>
        ]]>
      </property>
    </properties>
  </object>
</customize>
```

```
        <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レポート 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ので、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・タイトルより優先します。
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：**

```
<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:**

```
<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>
```

**ヒント:** 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、次の項を参照してください。

7.5.10 項「<formula>」

**構文:**

```
<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。

表 7-14 &lt;summary&gt; タグの属性

属性	必須 / 任意	説明
function	任意	<p>サマリー値を出力するために適用する算術演算。</p> <p><i>average</i> は、リセット・グループ内の列値の平均を計算します。</p> <p><i>count</i> 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。</p> <p><i>first</i> 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。</p> <p><i>last</i> 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。</p> <p><i>maximum</i> は、リセット・グループ内の列の最大値を計算します。</p> <p><i>minimum</i> は、リセット・グループ内の列の最小値を計算します。</p> <p><i>pctTotal</i> 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。</p> <p><i>stddeviation</i> 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。</p> <p><i>sum</i> は、リセット・グループ内の列値の合計を計算します。</p> <p><i>variance</i> 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。</p>
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>

表 7-14 <summary> タグの属性

属性	必須 / 任意	説明
reset	任意	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。  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。
productOrder	任意	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。productOrder は、クロス積グループが所有する列に対してのみ使用します。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。
nullval	任意	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。

**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

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
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"><li>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)</li><li>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)</li></ul>
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、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る点に注意してください。

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

```
</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、次の項を参照してください。

7.5.8 項「<field>」

**構文:**

```
<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 「RWCLI60 コマンド・ライン引数」

付録 B 「Reports Server 構成パラメータ」

付録 C 「環境変数」

付録 D 「データベース接続文字列」

付録 E 「Reports Web カートリッジ」

付録 F 「トラブルシューティング」



---

# 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 Reports Server（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。

## 構文

```
RWCLI60 MODULE|REPORT=runfile USERID=userid  
[ [keyword=]value| (value1, value2, ...) ] SERVER=tnsname
```

ここで、keyword=value は有効なコマンド・ライン引数です。

## 使用上の注意

- クライアントのコマンド・ラインで指定されたすべてのファイル名とパスは、サーバー・マシン上のファイルとディレクトリを参照します。ただし、コマンド・ファイルは参照しません。
- 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、Reports Server 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、Reports 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。Reports 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 ユーザー名またはプレース・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・データベースへ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へアクセスする場合）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、Web 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` (Web 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、長さが 512 バイトを超えることはできません。

```
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 D「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

- 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・ラインで使用す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 つだけ生成されます。
- 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 プレビューのみで有効です。このキーワードはキャラクタ・モードでしか使用しません。

### 構文

[TERM=] termttype

### 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、Reports Developer 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。Reports 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、『Oracle8 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	Reports Server 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使用できないことに注意してください。コマンド・ラインに DISTRIBUTE が見つかると、サーバーによって DESTYPE=CACHE コマンド・ライン引数が無視されます。
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。Reports Developer 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することができます。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・アプリケーションの場合と同様にして指定します（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・アプリケーションで)。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。たとえば、次のように URL を入力します。

```
DESNAME=printer,LPT1:  
DESNAME=printer,FILE:
```

### 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。DESTYPE=FILE と 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

- 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、Report 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わせてください。

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ています。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ています。
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ています。
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ています。
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ています。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ることに注意してください。
DELIMITED	これは、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ています。デリミタを選んでいないときは、デフォルトのデリミタとして TAB が使用されることに注意してください。
XML	レポートが XML 文書として出力され、拡張子 .XLM の付いた別ファイルとして保存されることを意味しています。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 Report Builder では、DESFORMAT が空白または `df1t` なら、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。Choose Printer 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

- 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。



---

## CACHELOB

**説明** CACHELOB は、検索された Oracle8 のラージ・オブジェクトをテンポラリ・ファイルのディレクトリ (REPORTS60\_TMP で指定される) にキャッシュするかどうか指定します。

### 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

- このオプションはコマンド・ラインでしか設定できません。
- 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勧めします。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生じるため、パフォーマンスが低下する可能性があります。

## 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

- 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することができます。

## 使用上の注意

- 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わ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

- 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『Oracle8 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 回のみのお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 READONLY 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・モードでしか使用しません。

#### 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 dfltrep.log

## BUFFERS

**説明** BUFFERS はキロバイト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 が指定されているときは YES と指定されている必要があります。

## 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りますので注意してください。たとえば、レポートのフレームがページ・ディメンションを超えるサイズになっている場合、レポートは実行されません。

## 構文

[PAGESIZE=] width x height

## 値

---

フォームに対する任意の有効なページ・ディメンション。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。センチメートルでは 1312 センチメートル。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って決定されます。（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 なら、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

- 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。Report 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。
- 「ランタイム・パラメータ・フォーム」で入力された PAGESIZE 値は、Parameter プロパティ・シートに入力された DECIMAL の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、レポートの実行に費やされた経過時間と CPU 時間に関する統計が Report 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 Report Builder 時間と ORACLE 時間の合計です。
- **時間** は Report Builder で費やされた時間です。
- **ORACLE 時間** はデータベースで費やされた時間で、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費やされた時間です。
- **SQL** は SRW.DO\_SQL の実行に費やされた時間です。

- **プロセスが使用した TOTAL CPU 時間**はデザイナーで費やされた CPU 時間です。

**注意:** データベースは Report 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、Report 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。

### RUNDEBUB

**説明** RUNDEBUB 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。RUNDEBUB 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。RUNDEBUB 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われる枠。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。Report 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

```
[RUNDEBUB=] {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、レポートが作成された時点で 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、レポート作成完了時に COMMIT を実行します。

ROLLBACK は、レポート作成完了時に ROLLBACK を実行します。

NOACTION は、レポート作成時には動作を行いません。

## デフォルト

USERID が入力されているときは COMMIT

外部ソース（たとえば Oracle Forms Developer）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** エラーが発生してレポートが完成されなかった場合に ONFAILURE は 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、レポートが完成しなかった場合に COMMIT を実行します。

ROLLBACK は、レポートが完成しなかった場合に ROLLBACK を実行します。

NOACTION は、レポートが完成しなかった場合、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 Developer）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、レポートの完成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 は実行時に実行したいキーストローク・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しか使用されません。

### 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 Reports 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・ファイルを実行することができます。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しか使用されません。

### 構文

```
[KEYOUT=]keyout_file
```

### 値

任意の有効なファイル名

## ERRFILE

**説明** ERRFILE は、Report Builder でエラー・メッセージを格納するの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、Report 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 にしか適用されません。

### 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

10K

### 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

- キャラクタ・モードのレポートに対して ORIENTATION=LANDSCAPE なら、プリンタ定義ファイルに横方向を指定する句が含まれていることが必要です。
- 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だけで、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、そのジョブはキャンセルされます。

#### 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 Report 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 で実行することができます。Windows では、MODE=CHARACTER と指定すると、Report Builder の ASCII ドライバを使って編集可能な ASCII 出力が生成されます。

## 構文

```
[MODE=] {BITMAP|CHARACTER|DEFAULT}
```

## 値

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

- 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 Report 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

- 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することはできません。
- 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だからです。

```
RWRUN60 MODULE=order_entry  
USERID=scott/tiger  
TRACEFILE=trace.log LOGFILE=mylog.log  
ERRFILE=err.log
```

## TRACEMODE

**説明** TRACEMODE は、Report Builder でトレース情報をファイルに追加するのか、ファイル全体を上書きするのかを指定します。

### 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することはできません。

---

## 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

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、判断することができます。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 ファイルを生成することができます。

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することはできません。

### **AUTOCOMMIT**

**説明** データベースの変更（例えば 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指定します。非 ORACLE データベースによっては（例えば SQL Server など）、AUTOCOMMIT=YES とする必要があるので注意してください。

#### **構文**

[AUTOCOMMIT=] {YES|NO}

#### **値**

YES または NO

#### **デフォルト**

NO

### **NONBLOCKSQL**

**説明** NONBLOCKSQL は、Reports 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

(レポート出力にプリント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えなく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することができます。

### 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** DELIMITER は、レポート出力のセルを区切るのに使用する文字を指定します。

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することができます。

### 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したいときに特に便利です。

## SERVER

**説明** SERVER は Reports Server 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 Server 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・ライン引数を省略することができます。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、Reports 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。たとえば、次のように URL を入力します。

```
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

```
[SCHEDULE=] string
```

ここで文字列は次のとおりです。

[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、ジョブ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済み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

- 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、Reports Server 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、Reports Server 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の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意味します。

### 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** CELEWRAPPER 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。

tab      セルの両側にタブが表示されることを意味します。

tab      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

なし

### 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。
- 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 は、OLAP Server 文字列 (例えば ncacn\_ip\_  
tcp:olap2-pc/sl=x/st=x/ct=x/sv=x/) です。サーバー文字列の詳細は次を参照してください。

domain は、OLAP Server のドメインです。

user は、OLAP Server にログオンするためのユーザー ID です。

password は、そのユーザー ID に対するパスワードです。

ramuser は、RDBMS にログインするためのユーザー ID です。

rampassword は、RDBMS に対するパスワードです。

ramexpressid は、Oracle Sales Analyzer データベースのユーザー ID です。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 は、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) です。このファイルは、コードやデータのデータベースの場所などの情報を指定します。UNC ( Universal Naming Convention ) 構文を使用すると、同じドライブ文字をその場所にマップすることなく、複数のユーザーが同じ接続を使用してデータにアクセスすることを許可できます。UNC 構文は、  
¥¥ServerName¥ShareName¥ の後に任意の下位フォルダまたはファイル、あるいはその両方が続きます。

rammasterdb は、最初に付加される Relational Access Manager データベースの名前です。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」タブで確認することができます。

ramconnecttype は、Express 接続の種類です。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 <b>注意:</b> Windows NT ではすべての設定値を使用します。 UNIX システムでは、設定値、0、-1、-2 だけしか使用しません。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

- 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。

- 文字列に順スラッシュ (/) が必要なときは、エスケープ文字として前方スラッシュを使う必要があります。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することができます。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 Reports Server 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、WebDB に登録されている、保護されている Reports Server 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 Reports Server 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 の有効期限を変更することができます。詳細は、付録 C「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したいときは、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したいときは、Web 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 （Web 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 WebDB で生成した任意の有効なユーザー名とパスワード。WebDB 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。

## CUSTOM

**説明** CUSTOMIZE は、レポート実行時に適用させたい XML ファイルを指定します。XML ファイルには、レポート定義を何らかの形で変更する、カスタマイズされた設定（例えば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 ファイルに保存したいときです。

### **構文**

```
[SAVE_RDF=] filename.rdf
```

### **値**

任意の有効なファイル名



# Reports Server 構成パラメータ

この付録には、Reports Server 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。

**表 B-1 Reports Server 構成パラメータ**

パラメータ	説明
CACHEDIR	<p>CACHEDIR は Reports Server 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することができます。指定されていないときのデフォルトは <code>ORACLE_HOME¥REPORT60¥SERVER¥CACHE</code> です。</p> <p><b>例</b></p> <p><b>CACHEDIR="c:¥ORACLE_HOME¥Report60¥cache"</b></p>
CACHESIZE	<p>CACHESIZE はメガバイト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 Reports Server 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ります。キャッシュがこの設定サイズを超えると、Reports Server 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。</p> <p>デフォルト値 =50</p> <p><b>注意:</b> このパラメータは Queue Manager から設定することができます。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、「キャッシュ・サイズ (MB)」の設定値を変更します。</p>

**表 B-1 Reports Server 構成パラメータ**

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 Reports Server 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することができますマスター・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することができます。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=&lt;servername&gt; minengine=&lt;minimum number of master engines&gt; maxengine=&lt;maximum number of master engines&gt; initengine=&lt;initial number of master engines&gt; cachedir=&lt;directory of central cache&gt;)"</pre> <p><b>注意:</b> 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「Reports Server 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 Reports Server により、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。</p> <p>デフォルト値 =50</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。</p> <p><b>例</b></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。</p> <p>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 Reports Queue Manager からユーザー ID パスワードを入力して、キュー管理者としてログインすることができます。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、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>

表 B-1 Reports Server 構成パラメータ

パラメータ	説明
INITENGINE	<p>INITENGINE は、Reports Server により起動されたランタイム・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。</p> <p>デフォルト値 =1</p>
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、faildjob、succeededjob です。</p> <p><b>例</b></p> <p>LOGOPTION="alljob"</p>
MAILPROFILE	<p>( Windows NT のみ ) MAILPROFILE はレポートの送信時に使用するメール・プロファイルとパスワードです。</p> <p><b>例</b></p> <p>MAILPROFILE="mailprofileid/password"</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。</p> <p>デフォルト値 =20</p>
MAXENGINE	<p>MAXENGINE は Reports Server 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。</p> <p>デフォルト値 =1</p> <p><b>注意:</b> このパラメータは Queue Manager から設定することができません。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>

**表 B-1 Reports Server 構成パラメータ**

パラメータ	説明
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。Reports Server 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。</p> <p>デフォルト値 =30</p> <p><b>注意:</b> このパラメータは Queue Manager から設定することができます。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、「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>
MINENGINE	<p>MINENGINE は、Reports Server 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とリソースを使用できるか確認してください。</p> <p>デフォルト値 =0</p> <p><b>注意:</b> このパラメータは Queue Manager から設定することができます。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、「同時実行エンジン: 最小」の設定を変更します。</p>
PERSISTFILE	<p>PERSISTFILE は Reports Server 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み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¥SERVER です。</p> <p><b>例</b></p> <p>PERSISTFILE="C:¥ORACLE_HOME¥Report60¥repserver.dat"</p>
REPOSITORYCONN	<p>REPOSITORYCONN は Reports Server の起動時にサーバー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 Reports Server のキュー・アクティビティ（スケジュール済みのジョブ）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。</p> <p>Web CGI の構成は、3.2.6 項「ステップ 6.（オプション）Reports Server の構成の変更」を参照してください。</p> <p><b>例</b></p> <p>REPOSITORYCONN="repserver_schema/password@mydb"</p>

**表 B-1 Reports Server 構成パラメータ**

パラメータ	説明
SECURITY	<p>SECURITY は、Reports 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 =1</p>
SECURITYTNSNAME	<p>SECURITYTNSNAME は、Reports Server に対するユーザーの認証に使用される Oracle WebDB データベースの TNS 名です。Reports Server では、Oracle WebDB を使用してセキュリティ・チェックを行い、また、アクセスの限定された Reports Server 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、アクセスの限定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確認します。</p> <p>SECURITYTNSNAME パラメータが設定されている場合、Reports Server を使用してレポート・リクエストを処理するには、Reports Server、プリンタ、WebDB 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。</p> <p>詳細は、第 5 章「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。</p> <p><b>例</b></p> <p>SECURITYTNSNAME="sec_db"</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 Reports Server は異なるディレクトリを検索することができるため、この設定値は、同じ ORACLE_HOME を共有する複数の Reports Server がある場合に便利です。</p> <p><b>例</b></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。</p> <p><b>例</b></p> <p>SUCCNOTEFILE="c:\ORACLE_HOME\REPORT60\succnote.dat"</p>

---

**表 B-1 Reports Server 構成パラメータ**

パラメータ	説明
TEMPDIR	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。 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 ブに設定することができます。  <b>例</b> TEMPDIR="c:¥ORACLE_HOME¥Report60¥temp"

## 環境変数

この付録では、Reports Server 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。Web CGI およびサーブレットの環境変数の一覧については、表 C-1「Web CGI およびサーブレットの環境変数」を参照してください。

環境変数は、Reports Server 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・エディタを使って設定します。UNIX の場合は、シェル・スクリプトを使って設定します。

**表 C-1 Web CGI およびサーブレットの環境変数**

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 =30  cookie は、ユーザーが保護された Reports Server 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 Reports Server 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 =dbauth.htm
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 =reports6.0

**表 C-1 Web CGI およびサプレットの環境変数**

変数	説明
REPORTS60_CGIDIAGBODYTAGS	Reports Web CGI の場合に、RWCGI60 の診断 / 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。例えば、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。
REPORTS60_CGIDIAGHEADTAGS	Reports Web CGI の場合に、RWCGI60 の診断 / 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。例えば、<TITLE> または <META...> タグの設定にこの環境を使用できます。
REPORTS60_CGIHELP	Reports Web CGI の場合に、空のリクエスト <code>http://your_webserver/rwcgi60?</code> 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。例えば、これを <code>http://www.yahoo.com</code> に設定するとその URL に移動し、 <code>myhelpfile.htm</code> に設定すると、ファイル <code>http://your_webserver/myhelpfile.htm</code> を表示します。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。
REPORTS60_CGIMAP	Reports Web CGI の場合、マップ・ファイル構成を使用している場合に、完全に限定された RWCGI60 マップ・ファイルのファイル名およびロケーションを定義します。(例: <code>C:\¥ORANT¥REPORT60¥cgicmd.dat</code> )
REPORTS60_CGINODIAG	Reports Web 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。例えば、REPORTS60_CGINODIAG が定義されていると、 <code>http://your_webserver/rwcgi60/help?</code> は機能しません。
REPORTS60_REPORTS_SERVER	Web カートリッジ・リクエスト用のデフォルトの Reports Server を指定します。この環境変数が設定されている場合、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することができます。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たい場合、この変数を使用して別のポート番号を指定することができます。 デフォルト値 =443



表 C-1 Web CGI およびサープレットの環境変数

変数	説明
REPORTS60_SYS_AUTH	ユーザーが制限された Reports Server に対するレポート・リクエストを実行する際に、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、Web CGI または サブレット を用いて 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 A 「RWCLI60 コマンド・ライン引数」を参  
照してください。

表 D-1 データベース接続文字列

データベース接続文字列	Reports Server 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。Reports Server 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、Reports Server からデータベース認証フォームが返されます。	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
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。  ローカルのデータベースがない場合、Reports Server から、データベース認証フォームが返されます。	なし  Oracle データベースを入力します。

**表 D-1 データベース接続文字列**

データベース接続文字列	Reports Server の応答	ユーザー・アクション
<\$username>@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らなかった場合、Reports Server からデータベース認証フォームが返されます。</p> <p>Reports Server が表中のプレースホルダ・ユーザー名を見つけることが出来なかった場合は、Oracle ユーザー名及びデータベース名を探して、パスワードを取得します。Reports Server がパスワードを見つけると、レポートが実行されます。</p> <p>パスワードが見つからなかった場合、Reports Server からデータベース認証フォームが返されます。</p>	<p>Oracle ユーザー名およびパスワードを入力します。</p> <p>なし</p> <p>データベース・パスワードを入力します。</p>
<\$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ったら、Reports Server によりレポートが実行されます。</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、Reports Server 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>

---

## Reports Web カートリッジ

### E.1 Reports Web カートリッジの例

ここでは、Oracle Applications Server (OAS) の Web カートリッジを使用して Reports Server を手動で構成する方法を、順を追って説明します。構成手順に意味を持たせるためには、いくつかの事項を仮定する必要があります。この項の趣旨に沿って、次の状況を前提とします。

- Web リクエストを可能にするように Reports Server を構成していること。
- Oracle Application Server のリリース 4.0 またはそれ以降において、Reports Web カートリッジを使用していること。
- Reports Server は、Oracle Application Server と異なるマシン上にインストールされていること。

#### E.1.1 ステップ 1. ソフトウェアをインストールする

Reports Server を構成するには、Oracle Reports Server ソフトウェアをインストールしておく必要があります。Oracle Reports Thin Client および Oracle Reports Server のコンポーネントと Report Builder のインストールの詳細は、『スタート・ガイド』を参照してください。

**ヒント:** Reports Thin Client、Reports Server または Report Builder のみをインストールする場合は、Oracle Installer を起動し、「カスタム・インストール」オプションを選択してください。「使用可能な製品」リストボックスから「Oracle Reports Developer」ノードを拡張して、Reports Thin Client、Oracle Reports Server または Report Builder の中からインストールするものを選択します。『スタート・ガイド』マニュアルの説明に従ってインストールを進めます。

Web サーバー・マシン上:

1. Web サーバーをインストールしてから Oracle Reports Thin Client コンポーネントをインストールします。Web サーバーのインストールの詳細は、Web サーバーのドキュメントを参照してください。

Reports Server マシン上:

1. Oracle Reports Server のコンポーネントと Report Builder をインストールします。Report Builder が必要なのは、レポートのビルドが必要な場合のみですので注意してください。Reports Server マシンを開発マシンとしてのみ使用する場合は、Report Builder コンポーネントはインストールしないでください。
2. Oracle Reports Server をインストールする際に、Reports Server の TNS 名とポート番号の識別を求められます。この情報は `tnsnames.ora` ファイル ( `ORACLE_HOME\NET80\ADMIN` ディレクトリにある ) に追加されます。
  - Reports Server の TNS 名。 `repserver.world` と入力するか、デフォルト値を使用してください。  
 ここで、 `repserver.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` の設定値で指定されたドメインです。 `NAMES.DEFAULT_DOMAIN` の設定が `sqlnet.ora` ファイルで指定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。
  - ポート。サーバーの受信の対象となるポート番号を入力するか、デフォルト値を使用します。

**ヒント :** Reports Server のサービス・エントリは、テキスト・エディタの `tnsnames.ora` ファイルを開くことでいつでも編集することができます。
3. Reports Server の TNS 名とポート番号を控えておいてください。この情報は、この Reports Server に対するサービス・エントリを Web サーバー・マシン上の `tnsnames.ora` ファイルに追加するときに必要になります。詳細は、E.1.3.1 項「Reports Server のサービス・エントリを作成する手順」を参照してください。

## E.1.2 ステップ 2. Web サーバーを構成する

このステップは Web サーバー・マシン上で行います。

**Oracle Application Server を構成するには、次のようにします。**

これらの手順の詳細は、Oracle Application Server のドキュメントを参照してください。

1. OAS ノード・マネージャを起動します。
2. ブラウザを起動します。
3. ブラウザで、OAS ノード・マネージャのポートを指します。サイトおよび HTTP リスナー・ノードを拡張します。
4. 必要に応じて、リスナーを作成するか、使用したいリスナーを拡張します。

## E.1.3 ステップ 3. Reports Web カートリッジを構成する

このステップは Web サーバー・マシン上で行います。

### E.1.3.1 Reports Server のサービス・エントリを作成する手順

Web サーバーが Reports Server とは異なるマシン上にある場合は、Reports Server マシンで生成された Reports Server のサービス・エントリを Web サーバー・マシン上の `tnsnames.ora` ファイルに追加する必要があります。これにより、Web CGI 実行ファイルと Reports Server の通信が可能になります。

**ヒント:** Reports Server のサービス・エントリの設定を覚えていない場合、Reports Server マシン上の `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ます。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 `tnsnames.ora` ファイル (`ORACLE_HOME\NET80\ADMIN` ディレクトリにある) をオープンします。
2. 次の Reports Server 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。

`repserver.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` の設定値で指定されたドメインです。  
`NAMES.DEFAULT_DOMAIN` 設定値が `sqlnet.ora` ファイルで指定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。

### E.1.3.2 デフォルトの Reports Server を構成する

#### Windows NT の場合:

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」値を選択して次の環境変数を追加します。

```
REPORTS60_REPORTS_SERVER=repserver
```

ここで、`repserver` は構成中の Reports Server の名前 (Reports Server の TNS サービス・エントリ名) です。E.1.1 項「ステップ 1. ソフトウェアをインストールする」参照。

**注意:** 設定値に大文字と小文字の区別が必要かどうかを判断するには、使用しているオペレーティング・システムのドキュメントを参照してください。

## UNIX の場合：

**ヒント：**Web サーバー・マシン上で環境変数を設定するシェル・スクリプトの作成を必要とする場合があります。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。

1. REPORTS60\_REPORTS\_SERVER 環境変数に Reports Server の名前を設定します。

```
setenv REPORTS60_REPORTS_SERVER repserver
```

ここで、repserver は構成中の Reports Server の名前（Reports Server の TNS サービス・エントリ名）です。E.1.1 項「ステップ 1. ソフトウェアをインストールする」参照。

**ヒント：**設定値に大文字と小文字の区別が必要かどうかを判断するには、使用しているオペレーティング・システムのドキュメントを参照してください。

### E.1.3.3 Reports Web カートリッジを構成する（Windows NT および UNIX）

次の手順は、Oracle Application Server、リリース 4.0 またはそれ以降のものをすでに構成済みであることを仮定したものです。詳しい情報は、Oracle Application Server のドキュメントを参照してください。

1. Oracle Application Server の Welcome ページで、「OAS Manager」をクリックします。
2. OAS マネージャのナビゲーション・ツリーで、Web サイトのアイコンの横の「+」アイコンをクリックします。
3. アプリケーション  アイコンをクリックします。
4. 追加  アイコンをクリックして、アプリケーションの追加フォームを表示させます。
5. 「アプリケーションの追加」フォームの「アプリケーション・タイプ」フィールドで、「C Web」を指定します。
6. 「適用」をクリックします。
7. 「アプリケーションの追加」フォームの「アプリケーション名」フィールドで、アプリケーションの名前を指定します（例：Reports）。
8. 「表示名」フィールドで、表示名を指定します（例：Oracle Reports）。
9. 「アプリケーションのバージョン」フィールドで、バージョン番号を指定します（例：6.0）。
10. 「適用」をクリックします。
11. 「成功」ダイアログで、「このアプリケーションにカートリッジを追加」をクリックします。
12. 「CWeb カートリッジの追加」フォームの「カートリッジ名」フィールドで、Reports60Cartridge と入力します。
13. 「表示名」フィールドで、Reports60Cartridge と入力します。



14. 「カートリッジ共有オブジェクト」フィールドで、共有されたオブジェクト・ファイル (rwws60.dll) のロケーションを指定します。例えば、D:\ORACLE\_HOME\BIN\RWWWS60.DLL
15. 「カートリッジのエントリ・ポイント」フィールドで、次のように入力します。rwws\_start
16. 「仮想パス」フィールドで、/rwws と入力します。
17. 「適用」をクリックします。
18. 「成功」ダイアログで、「OK」をクリックします。
19. カートリッジ構成フォルダで、「チューニング」をクリックします。
20. 「最大インスタンス数」フィールドで、100 と入力します。
21. 「適用」をクリックします。
22. 「成功」ダイアログで、「OK」をクリックします。

## E.1.4 ステップ 4. Reports Server を構成する

Reports Server を、サービスまたは非サービスとして Windows NT 上で構成することも、Reports Server を UNIX 上で構成することもできます。Reports Server を構成するには、tnsname.ora ファイルに TNS サービス名を作成し、環境変数を設定してから、Reports Server をインストールする必要があります。

RWMTS60 が、Reports Server の実行形式です。RWMTS60 には、3 つの関連コマンドがあります。

- **Install:** これは、Reports Server を Windows NT 上でサービスとしてインストールできるようにします。Windows NT に関しては、E.1.4.2.2 項「Reports Server をサービスとして Windows NT にインストールする」を参照してください。UNIX に関しては、E.1.4.3 項「Reports Server を UNIX 上で構成する手順」を参照してください。
- **Listen:** これは、Reports Server を Windows NT 上で非サービスとしてインストールできるようにします。E.1.4.2.3 項「Reports Server を非サービスとして Windows NT に構成する」参照。
- **Uninstall:** これは、Reports Server を Windows NT 上でサービスとしてアンインストールできるようにします。E.1.4.2.4 項「Windows NT 上の Reports Server をアンインストールする」参照。

Windows NT の場合、Reports Server は手動またはバッチ・モードで Install、Listen および Uninstall することができます。インストール中に発生したエラー・メッセージを無視してオペレーションが中断されことなく完了するようにするには、BATCH 引数に Yes を設定します。インストール中に発生したエラーやステータス情報は、Reports Server のログ・ファイル (ORACLE\_HOME\REPORT60\SERVER\repserver.log にある) で調べることができます。BATCH のデフォルトは No です。

この手順は、Reports Server マシン上で行います。

### E.1.4.1 Windows NT 上で Reports Server を構成する手順

#### E.1.4.1.1 (オプション) ソース・パスを設定する

別の方法としては、Reports Server をインストールした後に、Reports Server の構成ファイルでソース・パスを設定することができます。詳細は、付録 B「Reports Server 構成パラメータ」の SOURCEDIR パラメータを参照してください。

1. ソース・ディレクトリがまだ作成されていないときはそれを作成します。
2. 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。
3. regedit と入力してレジストリ・エディタを表示します。
4. HKEY\_LOCAL\_MACHINE SOFTWARE ORACLE を拡張します。
5. REPORTS60\_PATH をダブルクリックします。
6. 新しいレポートのソース・パス (C:\¥WEB\_REPORTS など) を追加します。

**注意:** 設定値に大文字と小文字の区別が必要かどうかを判断するには、使用しているオペレーティング・システムのドキュメントを参照してください。

### E.1.4.2 Windows NT 上で Reports Server を構成する手順

#### E.1.4.2.1 (オプション) ソース・パスを設定する

別の方法としては、Reports Server をインストールした後に、Reports Server の構成ファイルでソース・パスを設定することができます。詳細は、付録 B「Reports Server 構成パラメータ」の SOURCEDIR パラメータを参照してください。

1. ソース・ディレクトリがまだ作成されていないときはそれを作成します。
2. 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。
3. regedit と入力してレジストリ・エディタを表示します。
4. HKEY\_LOCAL\_MACHINE SOFTWARE ORACLE を拡張します。
5. REPORTS60\_PATH をダブルクリックします。
6. 新しいレポートのソース・パス (C:\¥WEB\_REPORTS など) を追加します。

**注意:** 設定値に大文字と小文字の区別が必要かどうかを判断するには、使用しているオペレーティング・システムのドキュメントを参照してください。

#### E.1.4.2.2 Reports Server をサービスとして Windows NT にインストールする


1. Reports Server 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 -install repserver tcpip
```

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 -install repserver batch=yes
```

**ヒント:** Repserver は、付属のドメイン修飾子（例：.world）を付ける必要はありません。

3. 「はい」をクリックして Reports Server を NT サービスとしてインストールし、「OK」をクリックします。
4. 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。
5. 「サービス」ダイアログ・ボックスで「Oracle Reports Server [repserver]」を選択し、「スタートアップ」をクリックします。
6. スタートアップ・ダイアログ・ボックスから「ログオン」セクションの「アカウント」を選択して、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入力あるいは選択します。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。

**ヒント:** Postscript ファイルまたはプリンタに出力するときは、Reports Server 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かどうかを確認してください。通常、システム・アカウントはプリンタへのアクセスを持っていません。

7. 自動的に（システム起動時に）起動するようにサービスを設定します。
8. 「OK」をクリックします。
9. 「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、Reports Server が起動した日時が表示されます。Reports Server が起動しない場合は、付録 F「トラブルシューティング」を参照してください。

**ヒント:** Reports Server を初めて起動したときは、Reports Server の構成ファイル（repserver.ora など）が ORACLE\_HOME¥REPORT60¥SERVER ディレクトリに作成されます。Reports Server 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・パスを設定することができます。構成ファイルを変更したら Reports Server を停止して再起動し、変更内容を有効にします。

#### E.1.4.2.3 Reports Server を非サービスとして Windows NT に構成する

1. 次のコマンド・ラインを実行します。

```
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
```

**ヒント:** Repserver にはドメイン修飾名 ( .world など ) を付加する必要はありません。

#### E.1.4.2.4 Windows NT 上の Reports Server をアンインストールする

1. Reports Server 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
```

**ヒント:** Repserver は、付属のドメイン修飾子 ( 例 : .world ) を付ける必要はありません。

### E.1.4.3 Reports Server を UNIX 上で構成する手順

#### E.1.4.3.1 ( オプション ) 環境変数を設定する

**ヒント:** 環境変数を設定するシェル・スクリプトを作成して RWMTS60 実行ファイルを起動する必要がある場合があります。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。

1. ソース・レポート ( /WEB\_REPORTS など ) のディレクトリを作成します。
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
```

**ヒント:** 別の方法としては、Reports Server をインストールした後に、Reports Server の構成ファイルでソース・パスを設定することもできます。詳細は、付録 B 「Reports Server 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
```

**ヒント:** 設定値に大文字と小文字の区別が必要かどうかを判断するには、使用しているオペレーティング・システムのドキュメントを参照してください。

### E.1.4.3.2 Reports Server を起動する

1. \$ORACLE\_HOME/BIN ディレクトリに移動し、コマンド・ライン上で次のコマンドを実行して Reports Server をフォアグラウンドで実行します。

```
rwmts60 name=repserver
```

Reports Server をバックグラウンドで実行するときは、次のコマンドを実行します。

```
rwmts60 name=repserver &
```

サーバーが実行中かどうかを確認するには、ステップ 6 に進んでください。Reports Server を起動できない場合や、エラー・メッセージが返された場合には付録 F「トラブルシューティング」を参照してください。

**ヒント:** Reports Server を初めて起動したときは、Reports Server の構成ファイル (repserver.ora など) が \$ORACLE\_HOME/REPORT60/SERVER ディレクトリに作成されます。Reports Server のキャッシュはデフォルト状態に設定されます。キャッシュ・ディレクトリを変更するとき、あるいはレポートのソース・パスを設定するときは、構成ファイルを変更します。構成ファイルを変更したら Reports Server を停止して再起動し、変更内容を有効にします。

2. (オプション) \$ORACLE\_HOME/BIN ディレクトリに移動して次のコマンドを実行し、Reports Server が稼動していることを確認します。

```
rwrvq60 server=repserver
```

この Reports Server 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。

**ヒント:** Reports Server を無効にするには、[Ctrl] キーを押しながら [C] を押します。

**ヒント:** Postscript あるいはプリンタに出力するときは、プリンタをファイル (\$ORACLE\_HOME/GUICOM6/TK60/ADMIN ディレクトリにある) uiprint.txt で構成する必要があります。

## E.1.5 ステップ 5. レポート・リクエストを実行する

1. Reports Server 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、次のリクエストを作成します。

```
http://your_webserver/rwows60?report=your_report.rdf+
userid=username/password@mydb+desformat=html+destype=cache
```

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 環境変数に repserver を設定している場合は必要ありません。

**ヒント:** レポートが実行されない場合、またはエラー・メッセージが発生する場合は、付録 F「トラブルシューティング」を参照してください。

これで、Web カートリッジを構成する場合の Reports Server の構成は完了です。Reports Web カートリッジを使用した動的なレポートに対応する Reports Server の構成が正常に終了しました。

次のステップ：

- （オプション）Reports Server を調整して、パフォーマンスの最適化およびアクセス・コントロールなどの機能の追加を行います。E.1.6 項「ステップ 6.（オプション）構成を修正する」参照。
- ユーザーがレポートを利用できるようにします。実行要求を指定し、それらをユーザーに有効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。

## E.1.6 ステップ 6.（オプション）構成を修正する

Reports Server はデフォルトの構成を使用して動作しますが、いくつかの点において、パフォーマンスを調整、モニターのコントロールを設定、または追加の機能を実行するために構成に修正を加えることが可能です。

- 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、E.1.6.1 項「データベースのジョブのキュー・アクティビティを更新する」を参照してください。
- レポートへのユーザー・アクセスをコントロールする場合は、第 5 章「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。
- ロードの均衡化のために Reports Server を構成する場合は、第 6 章「Reports Server のクラスターの構成」を参照してください。
- 対象者に基づいてランタイムにレポートを修正する場合は、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。
- Reports Server の構成ファイルを修正する場合は、付録 B「Reports Server 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。
- 環境変数を修正する場合は、E.2 項「環境変数」を参照してください。

### E.1.6.1 データベースのジョブのキュー・アクティビティを更新する

データベースの設定を変更して、ジョブが実行される度に Reports Server 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こともできます。Reports Server 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、Reports Server はサーバーの再起動時に API をコールして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、デフォルトを上書きすることもできます（例えば Reports Server 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）を絶対に編集しないように注意してください。

データベースに接続するには、構成ファイルで REPOSITORYCONN パラメータを設定します。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 スクリプトを実行します。

Reports Server マシン上：

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. 次の構成パラメータを追加します。

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

ここで repserver\_schema/password@database は、指定した Reports Server 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。

**ヒント：**複数の Reports Server からのスナップショットを入手するときは、スナップショットを入手する各 Reports Server に対してデータベース内に異なるスキーマを作成することをお勧めします。これにより、Reports Server の再起動時のキュー・アクティビティ・データの損失を防ぐことができます。

3. コマンド・ラインから (ORACLE\_HOME¥REPORT60¥SQL ディレクトリにある) rw\_server.sql スクリプトを実行します。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>
```

4. Reports Server を停止してから再起動し、構成ファイルの変更内容を有効にします。Reports Server は起動時にデータベースに接続します。

**注意：**Reports Server を再起動すると、デフォルトではデータベースのキュー・アクティビティが削除されます。デフォルトは API を編集して上書きすることができます。

## E.2 環境変数

環境変数は、Reports Server 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・エディタを使って設定します。UNIX の場合は、シェル・スクリプトを使って設定します。

表 E-1 Web カートリッジ環境変数

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 =30 cookie は、ユーザーが保護された Reports Server 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 Reports Server 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 =dbauth.htm
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 =reports6.0
REPORTS60_OWSDIAGBODYTAGS	Reports Web カートリッジの場合に、RWOWS の不良診断 / 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。例えば、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。
REPORTS60_OWSDIAGHEADTAGS	Reports Web カートリッジの場合に、RWOWS の不良診断 / 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。例えば、<TITLE> または <META...> タグの設定にこの環境を使用できます。
REPORTS60_OWSHELP	Reports Web カートリッジの場合に、空のリクエスト <code>http://your_webserver/rwows?</code> とともに RWOWS 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/URI を定義します。例えば、これを <code>http://www.yahoo.com</code> に設定するとその URL に移動し、 <code>myhelpfile.htm</code> に設定すると、ファイル <code>http://your_webserver/myhelpfile.htm</code> を表示します。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。



表 E-1 Web カートリッジ環境変数

変数	説明
REPORTS60_OWSMAP	Reports Web カートリッジの場合、マップ・ファイル構成を使用している場合に、完全に限定された RWOWS マップ・ファイルのファイル名 / ロケーションを定義します。(例: C:\¥ORANT¥REPORT60¥OWSCMD.DAT)
REPORTS60_OWSNODIAG	Reports Web カートリッジの場合に、ヘルプやショーマップなどのデバッグ / 不良診断出力が定義されていたらそれらすべての RWOWS からの出力を無効にします。例えば、 <code>http://your_webserver/rwows/help?</code> は、REPORTS60_OWSNODIAG が定義されていないと機能しません。
REPORTS60_REPORTS_SERVER	Web カートリッジ・リクエスト用のデフォルトの Reports Server を指定します。このパラメータが設定されている場合、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することができます。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たい場合、この変数を使用して別のポート番号を指定することができます。 デフォルト値 =443
REPORTS60_SYS_AUTH	ユーザーが、保護された Reports Server に対するレポート・リクエストを実行するときに、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 =sysauth.htm



## トラブルシューティング

付録 F には、Reports Server 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。

表 F-1 トラブルシューティング情報

トラブル	考えられる原因と対処方法
Reports Server を起動してもハング・アップしているように見える	tnsnames.ora ファイルに構文エラーがあるか、または Reports Server が TNS 名を解決できてい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。
「デーモンがポートの聞き取りに失敗しました」というエラー・メッセージが現われる	ある Reports Server が動作しているポートと同じポートを聞き取っている Reports Server 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	Reports Server 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。Reports Server 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、uiprint.txt ファイル内にプリンタを構成してください。

表 F-1 トラブルシューティング情報

トラブル	考えられる原因と対処方法
Reports Server 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 tnsnames.ora か sqlnet.ora に問題があることを示しています。tnsnames.ora 内の Reports Server のエントリを調べてください。典型的な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、.world が名前に付加されていますが、これは world が sqlnet.ora ファイルに指定されているドメインであるためです。sqlnet.ora に NAMES.DEFAULT_DOMAIN 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 .world は省略してください。</p> <p>tnsnames.ora ファイルに問題がないようであれば、sqlnet.ora 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、Reports Server に付属しているバージョン以前のものと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大文字と小文字は URL で指定されているとお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が非表示になっていないか確認してください。(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、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。</p> <p>または、REPORTS60_PATH 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>

表 F-1 トラブルシューティング情報

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 Reports Server 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 Reports Server 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"> <li>■ 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。</li> <li>■ ユーザー・アカウントが、Reports Server 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。</li> <li>■ 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。</li> <li>■ 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。</li> </ul> <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（ドメイン名¥ユーザー名）。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、Report 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、Reports Server 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 Reports Server に渡すと、エラーとなります。Reports Server はクライアントのユーザー名ではなく Reports Server 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、Reports Server 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、Reports Server 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。Reports Server 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>

表 F-1 トラブルシューティング情報

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・マッピング・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 (Reports Web CGI またはサープレットの場合) か owscmd.dat (Reports Web カートリッジの場合) という名前のファイル、もしくは REPORTS60_CGIMAP または REPORTS60_OWSMAP 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・マッピング・ファイルを見つけ出すために、まず次のように入力し (Web CGI の例)</p> <pre>http://your_webserver/your_virtual_cgi_dir/rwcgi60.exe/showmap?</pre> <p>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た上で、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。</p> <pre>http://your_webserver/your_virtual_cgi_dir/rwcgi60.exe?your_key</pre> <p>your_key はキー・マッピング・ファイルの有効なキー・エントリです。</p>
Reports 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。Reports 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 (「<b>オプション</b>」 「<b>管理者ログオン</b>」) 任意のユーザー名とパスワードを指定できません。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 Reports Server を NT サービスとして利用できない	<p>LocalSystem userid の下で Reports Server 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> <pre>sqlnet.authentication_services=(NONE)</pre>
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、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 Reports Server 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 (例: ¥¥SALES¥DOCUMENTS¥REPORTS) を使用してください。これは Reports Server のパラメータ、Web CGI、サープレットおよび Web カートリッジ・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>

表 F-1 トラブルシューティング情報

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。
WebDB で Reports Developer 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>WebDB で、プリンタ、Reports Server 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・コントロール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"> <li>■ 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「必修パラメータ」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。</li> <li>■ Reports Server 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。</li> <li>■ 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。</li> </ul> <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 Reports Server 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>

表 F-1 トラブルシューティング情報

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	<div><ul style="list-style-type: none"><li>■ Web サーバーが応答していることを確認してください（例：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><li>■ Web CGI、サブリットまたは Web カートリッジ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/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。  <code>http://your_webserver/your_virtual_cgi_dir/rwcgim60.exe</code> or <code>http://your_webserver/rwows</code>  UNIX の場合は、次のいずれかの行を入力してください。  <code>http://your_webserver/your_virtual_cgi_dir/rwcgi60</code> or <code>http://your_webserver/rwows</code>  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 <code>your_virtual_cgi_dir</code>（通常は <code>cgi-bin</code>）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 <code>RWCGI60</code> 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><li>■ <code>REPORTS60_CGINODIAG</code>（Web CGI またはサブリットの場合）または <code>REPORTS60_OWSNODIAG</code>（Web カートリッジの場合）環境変数は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には、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。  <code>http://your_webserver/your_virtual_cgi_dir/rwcgi60.exe/ showenv?</code> <code>http://your_webserver/rwows/showenv?</code>  上の行を入力すると、他のカートリッジ・パラメータまたは環境変数を確認することができます。</li></ul></div>



表 F-1 トラブルシューティング情報

トラブル	考えられる原因と対処方法
	<ul style="list-style-type: none"> <li>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。  <pre>http://your_webserver/you_virtual_cgi_dir/rwcgi60.exe/ showenv?http://your_webserver/rwows/showenv?</pre> </li> <li>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。  <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@mydb+desformat=html</pre> </li> </ul> <p>レポートが表示されない場合は、</p> <ul style="list-style-type: none"> <li>Report 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、Reports Runtime <code>Your_report.rdf</code> が REPORTS60_PATH に基づいて配置されているか確認してください。</li> <li>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。</li> <li>レポートを発行しようとしている Reports Server が、制限されていないかどう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付与してもらう必要があります。Reports Developer システム管理者にご相談ください。</li> <li>発行しようとするレポートが制限されていないか確認してください。その場合には、制限されている Reports Server を実行するためのアクセス権限を付与してもらう必要があります。Reports Developer システム管理者にご相談ください。</li> </ul> <p>Reports Server 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 Reports Server 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、Reports Queue Manager からレポートを起動していると想定します（「<b>ジョブ</b>」「<b>新規作成</b>」）。「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>

**表 F-1 トラブルシューティング情報**

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 Reports Server にレポートを発行する場合はアクセス権限を持っている必要がありますが、レポートを出力するプリンタに対してアクセス権限を持つ必要はありません。Reports Developer システム管理者にご相談ください。</p>
ホスト名ルックアップ・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、システム管理者にご相談ください。</p> <p>レポートを制限された Reports Server に対して発行している場合、WebDB の Server Access に定義されている Web Gateway URL が不適切である可能性があります。</p> <p>WebDB で、WebDB メニューから「<b>管理</b>」をクリックします。次に「<b>Oracle Reports Developer セキュリティ</b>」と「<b>サーバー・アクセス</b>」をクリックします。編集する 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 ゲートウェイの URL を確認してください。</p> <p><b>注意:</b> WebDB の Oracle Reports Developer セキュリティにアクセスできるのは、Reports Developer システム管理者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 Reports Server 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 Reports Server 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定めるために 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 および Internet Explorer 4.0 は、カスケード・スタイル・シートをサポートしている。

## **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われるプロトコル。

## **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することができる。通常、パケットが配布されるか、配布にどれだけ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**

いかなるネットワークにおいても、クライアントからサーバー、サーバーからクライアントの双方向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わりを持ち、それらのプロトコル全体をカバーしている。

## **OAS (Oracle Application Server)**

Oracle 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、実質的な利益を得ることができる。Oracle 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。

## **Reports Launcher**

指定したレポートの実行リクエストを Reports Server に送信するなど、Active X がコントロールするレポートによる機能を使用するアプリケーション。

## **Reports Queue Manager**

Reports Server に送られたジョブ・リクエストを監視、操作することができる。

## **Reports Server**

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することができる。Windows NT、Windows 95 または Unix 上にインストールすることができる。Reports

Server 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することができる。

### **Reports Web CGI**

CGI 関連の Web サーバーと Reports Runtime の間のインターフェース。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。

### **Reports サブレット**

Java ベースの Web サーバーと Reports Runtime の間のインターフェース。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。

### **RWCLI60**

コマンド・ラインを解釈し、指定した Reports Server に送る実行ファイル (RWMTS60)。

### **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 Oracle 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。

### **WebDB**

Oracle WebDB 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 Developer 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 Reports Developer 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、WebDB が使用されている。

### **WebDB コンポーネント**

WebDB 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（例：チャート、フォームまたは Reports Developer 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。

### **Web サーバー**

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 daemon）。

### **Web ブラウザ**

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）。

### **カートリッジ（Cartridge）**

オープン API を通してクライアント、アプリケーション・サーバーまたはデータベースにプラグインするための操作可能なオブジェクト。

### **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 Server 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。Reports Server は、レポート・リクエストが保護された Reports Server 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 WebDB サイトにプッシュすることができる。エンド・ユーザーは WebDB 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することができる。

### **プレースホルダ・ユーザー名 ( Placeholder username )**

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することができる。プレースホルダ・ユーザー名によって、以下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することができる。
- 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することができる ( 例 : あるユーザーは、東日本の販売結果を取得し、また別のユーザーは、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、Reports Server 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、Reports Server 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 が用いる番号。





## 記号

---

<!-- --> , 105  
<![CDATA[]]> , 106  
<condition> , 107  
<customize> , 108  
<data> , 109  
<dataSource> , 110  
<exception> , 111  
<field> , 113  
<formLike> , 118  
<formula> , 118  
<function> , 120  
<group> , 121  
<groupAbove> , 122  
<groupLeft> , 123  
<labelAttribute> , 124  
<layout> , 126  
<link> , 129  
<matrix> , 131  
<matrixCell> , 133  
<matrixCol> , 134  
<matrixRow> , 135  
<object> , 135  
<programUnits> , 137  
<properties> , 138  
<property> , 139  
<report> , 141  
<section> , 143  
<select> , 145  
<summary> , 146  
<tabular> , 150

## A

---

ActiveX リクエスト・メソッド , 32

## C

---

CGI , 11

## D

---

Data Access Descriptor,WebDB

Listener を参照

## R

---

Reports Queue Manager

実行するジョブのスケジューリング , 37

ジョブのステータスの監視 , 21

Reports Server

UNIX で起動する , 26

UNIX でジョブのステータスを表示する , 27

Web CGI の構成例 , 14

WebDB のセキュリティのインストール , 41

WebDB のセキュリティ・パッケージのインストール , 41

Windows NT での非サービスの起動 , 28

アーキテクチャ , 4

概要 , 1

クラスタリングに向けた構成 , 72

構成の概要 , 14

構成パラメータ , 190

コマンド・ラインでの起動 , 24

新機能 , xi

重複ジョブの検出 , 32

停止 , 28

- データベース・キュー, 22
- デフォルトを設定する, 17, 199
- Reports Server 層、Reports Server, 4
- REPORTS60\_PATH 環境変数、UNIX, 20
- RWCLI60
  - コマンド・ライン引数, 155
  - コマンド・ライン・リクエスト, 31
- RWMTS60 コマンド, 20
- RWMTS60 実行ファイル, 23

## S

---

- SOURCEDIR パラメータ, 18
- SRW.RUN\_REPORT リクエスト・メソッド, 32

## U

---

- UNIX 環境変数, 「環境変数」を参照
- URL 構文
  - 完全な構文の例, 36
  - コマンド・ライン引数の隠蔽, 33
  - 単純化した構文の例, 36
  - ハイパーリンクの追加, 36
  - ブラウザから実行する方法, 36
  - リクエストの単純化, 33
  - レポートリクエストメソッド, 32

## W

---

- Web
  - CGI, 6
- Web CGI
  - キー・マップ・ファイル, 34
  - 構成例, 14
- WebDB
  - Listener, 12
  - コンポーネント・リクエスト・メソッド, 32
  - レポートのスケジュール, 66
- Web サーバー層、Reports Server, 4

## X

---

- XML レポート定義
  - PL/SQL による適用, 97
  - PL/SQL 例, 88
  - Report Builder で表示する, 104
  - 概要, 79

- 完全, 90
- 実行, 100
- 書式の例, 85
- 書式例外の例, 87
- 追加オブジェクト, 89
- 適用, 96
- デバッグ, 101
- パーサー, 102
- ハイパーリンク例, 88
- バッチ変更, 100
- 必須タグ, 82
- ファイルへの書込み, 104
- 部分, 83
- XML レポート定義タグ, 105

## あ

---

- アーキテクチャ
  - Reports Server, 4
  - Reports Server 層
    - Reports Server, 4
    - Web サーバー, 4
    - シン・クライアント, 4
    - データベース, 4
  - Web
    - サーバーの構成, 5
- アクセス
  - データベース, 43
  - デモ表, 43
- アクセス・コントロール
  - Oracle Reports システム管理者, 47
  - WebDB のセキュリティ・パッケージのインストール, 41
  - 概要, 40
  - 使用可能日カレンダー, 48
  - デバッグ, 59
  - デフォルト・パラメータの設定, 61
  - プリンタ, 52
  - 例, 43
  - レポート定義ファイル, 54
  - レポートのバッチ登録, 40

## い

---

- インストール
  - UNIX で Reports Server を起動する, 26, 27
  - Windows NT サービスとして起動する, 25

Windows NT での非サービスの起動, 28  
自動, 11  
自動インストールを使用した, 11  
手動, 14, 15  
追加の Reports Server, 23  
Reports Server Security, 41

## か

---

概念, 3  
カスタマイズ  
XML レポート定義, 82  
概要, 79  
環境変数  
UNIX  
REPORTS60\_PATH, 20

## き

---

キー・マッピング  
cgicmd.dat (Web CGI), 34  
エントリのマッピング, 34  
使用可能にする方法, 34  
どのようなときに使用するか, 33  
例, 34, 35  
キャッシュ・サイズ, 185  
キュー・アクティビティ  
データベース, 22  
許容値, 178

## く

---

クラスタリング  
構成, 72  
再送信, 76  
レポートの実行, 76

## こ

---

構成  
Reports Server  
ファイル, 20  
Reports Server の概要, 14  
Web CGI の例, 14  
コマンド  
RWMTS60, 20  
コマンド・ライン引数, 155

## さ

---

サブレット, 6, 14

## し

---

シェル・スクリプト, 20  
実行時のカスタマイズ  
XML レポート定義, 82  
概要, 79  
自動インストール, 11  
Reports Server, 11  
手動での構成、Reports Server, 14  
使用可能日カレンダー, 48  
処理  
Web レポート, 6  
新機能, xi  
シン・クライアント層、Reports Server, 4

## す

---

スクリプト  
シェル, 20

## せ

---

セキュリティ  
Oracle Reports システム管理者, 47  
実現, 40

## た

---

タグ、レポート定義用 XML, 105

## ち

---

重複ジョブの検出  
Reports Server の扱い, 32, 178  
複数の出力先, 33

## て

---

データベース層  
Reports Server, 4  
テキストの表記規則, xiii  
デバッグ  
XML レポート定義, 101

アクセス制限されたレポート, 59  
トレース・オプション, 102  
デフォルト・プリンタ  
アクセス設定, 26, 27

## に

---

認証 cookie  
期限切れ, 183  
ドメインの設定, 42

## は

---

バッチ  
登録, 40  
レポート  
Reports Queue Manager, 37  
WebDB サイトからの, 37, 66  
パラメータ  
Reports Server の構成, 190  
RWCLI60 command line arguments, 155  
SOURCEDIR, 18

## ふ

---

負荷均衡化  
構成, 72  
ジョブの再送信, 76  
レポートの実行, 76

## む

---

無効なパッケージ・プロシージャ, 58

## れ

---

例  
<!-- -->, 105  
<![CDATA[]]>, 106  
<condition>, 108  
<customize>, 109  
<data>, 110  
<dataSource>, 111  
<exception>, 113  
<field>, 116  
<formLike>, 119  
<formula>, 118

<function>, 120  
<group>, 122  
<groupAbove>, 123  
<groupLeft>, 124  
<labelAttribute>, 126  
<layout>, 127  
<link>, 130  
<matrix>, 131  
<matrixCell>, 133  
<matrixCol>, 134  
<matrixRow>, 135  
<object>, 136  
<programUnits>, 137  
<properties>, 138  
<property>, 141  
<report>, 142  
<section>, 144  
<select>, 145  
<summary>, 149  
<tabular>, 150  
RWCLI60 コマンド・ライン・リクエスト, 31  
Web CGI の構成, 14  
XML レポート定義  
PL/SQL, 88  
完全, 90  
書式, 85  
書式例外, 87  
追加オブジェクト, 89  
hyperlink, 88  
アクセス・コントロール, 43  
完全な URL 構文, 36  
キー・マッピング, 34, 35  
クラスタ構成, 72  
単純化した URL 構文, 36  
レジストリのエントリ, 191  
レポートの自動実行, 37  
レポートのスケジュール, 37, 66  
レポートのソース・パスの設定, 26  
レポートのバッチ変更, 100  
レポート・リクエスト  
Manage Component ページから, 59  
WebDB サイト, 62  
WebDB でのスケジュール, 66  
サーバーがクラスタ化されている場合の, 76  
スケジューリング, 37  
重複ジョブの検出, 32  
ブラウザから実行する方法, 36

- リクエストの指定 , 35
- レポートのビルド , 35
- レポート・リクエスト・メソッド
  - ActiveX , 32
  - RWCLI60 コマンド・ライン , 31
  - SRW.RUN\_REPORT , 32
  - URL 構文 , 32
  - WebDB コンポーネント , 32

