

# Oracle9i Reports

レポート作成ガイド

リリース 9.0.2

2002 年 8 月

部品番号 : J06221-01

ORACLE®

---

Oracle9i Reports レポート作成ガイド, リリース 9.0.2

部品番号 : J06221-01

原本名 : Oracle9i Reports Building Reports, Release 9.0

原本部品番号 : A92101-01

原著者 : Vanessa Wang

原協力者 : Christian Bauwens, Darren McBurney, Jim Safcik, Philipp Weckerle

Copyright © 2002, 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おります。

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 .....	v
------------	---

## 1 バーコードを使用したレポートの作成

1.1	この例の前提条件 .....	1-2
1.1.1	サンプル・ファイル .....	1-2
1.1.2	サンプル・スキーマへのアクセス .....	1-3
1.1.3	REPORTS_CLASSPATH 環境変数の更新 .....	1-4
1.2	紙出力用バーコード・レポートの作成 .....	1-4
1.2.1	Reports Builder への Java クラスのインポート .....	1-5
1.2.2	情報を格納するパッケージの作成 .....	1-5
1.2.3	Before Report トリガーの作成 .....	1-6
1.2.4	式列を含むデータ・モデルの作成 .....	1-7
1.2.5	レポートのレイアウトの作成 .....	1-12
1.3	Web 用バーコード・レポートの作成 .....	1-13
1.3.1	既存の HTML ファイルにおける問合せの作成 .....	1-14
1.3.2	データ・モデルにおける 3 つの式列の作成 .....	1-16
1.3.3	バーコード JavaBean の初期化とそのプロパティの設定 .....	1-19
1.3.4	Web に対するレポートの実行 .....	1-22
1.4	まとめ .....	1-24

## 2 レポートの一括処理と配布

2.1	この例の前提条件 .....	2-2
2.1.1	サンプル・ファイル .....	2-2
2.1.2	サンプル・スキーマへのアクセス .....	2-3

2.2	既存のレポートを一括処理するための設定 .....	2-3
2.3	配布 XML 定義の編集 .....	2-5
2.4	レポートの実行 .....	2-6
2.5	まとめ .....	2-7

### 3 REF カーソルを使用したペーパー・レポートの作成

3.1	この例の前提条件 .....	3-3
3.1.1	サンプルの Human Resources スキーマへのアクセス .....	3-4
3.2	REF カーソル型の定義 .....	3-4
3.3	REF カーソル問合せの作成 .....	3-6
3.4	データ・モデルの調整 .....	3-9
3.5	REF カーソル問合せ間のリンクの作成 .....	3-11
3.6	サマリー列の追加 .....	3-13
3.7	レイアウトの作成 .....	3-15
3.8	パッケージへの SELECT 文の移動 .....	3-17
3.9	ライブラリへのパッケージの移動 .....	3-20
3.10	まとめ .....	3-21

### 4 XML トランスポートابل・データ・ソースを使用したレポートの作成

4.1	この例の前提条件 .....	4-2
4.1.1	サンプル・ファイル .....	4-2
4.1.2	サンプル・スキーマへのアクセス .....	4-3
4.2	SQL 問合せと XML 問合せを使用したレポートの手動作成 .....	4-3
4.2.1	新規レポートの SQL 問合せの作成 .....	4-3
4.2.2	XML データ・ソースへのアクセスに使用する XML 問合せの作成 .....	4-5
4.2.3	2 つの問合せ間のデータ・リンクの作成 .....	4-7
4.2.4	レポート・ウィザードを使用したレポートのレイアウトの作成 .....	4-9
4.2.5	レポートの行カラーの交互適用 .....	4-10
4.2.6	グループを使用した XML データのフィルタリング .....	4-13
4.3	ペーパーを対象としたレポートの実行 .....	4-16
4.4	まとめ .....	4-17

### 5 テキスト・トランスポートابل・データ・ソースを使用するレポートの作成

5.1	この例の前提条件 .....	5-2
5.2	textpds.conf ファイルの設定 .....	5-2

5.3	レポート・ウィザードを使用したレポートの作成 .....	5-4
5.4	まとめ .....	5-6

## 6 Express データを使用したレポート作成

6.1	この例の前提条件 .....	6-2
6.1.1	サンプル・ファイル .....	6-2
6.1.2	OLAP Server へのアクセス .....	6-3
6.2	レポート・ウィザードによる Express レポートの作成 .....	6-3
6.3	Express クエリーの修正 .....	6-7
6.4	データ・モデルへのサマリー列およびカスタム・メジャーの追加 .....	6-9
6.4.1	データ・オブジェクト名の変更 .....	6-10
6.4.2	サマリー列の作成 .....	6-11
6.4.3	カスタム・メジャーの作成 .....	6-12
6.5	レポート・レイアウトの拡張 .....	6-14
6.5.1	レポートへのサマリー・フィールド挿入 .....	6-14
6.5.2	レポートへのカスタム・メジャー・フィールド挿入 .....	6-16
6.5.3	ディメンション値のソート .....	6-18
6.5.4	ペーパー・デザイン・ビューでの書式変更 .....	6-19
6.6	まとめ .....	6-20

## 用語集

## 索引



---

---

# はじめに

このマニュアルで示されている例は、Oracle9i Reports とその機能の習得に役立ちます。

## 対象読者

このマニュアルは、Oracle9i Reports を使用してレポートを作成するすべてのユーザーを対象としています。初級者から上級者までの幅広い読者のニーズに対応しています。各章では、レポートを作成するための手順を初めから終わりまで順を追って説明します。各レポートを作成することにより、Reports Builder の多くの優れた機能の使用方法がわかるようになっています。

## 構成

このマニュアルは、次の章で構成されています。

**表 0-1 マニュアルの内容の概要**

章	内容
第 1 章「バーコードを使用したレポートの作成」	バーコード Java Bean を使用するレポートの作成方法について説明します。
第 2 章「レポートの一括処理と配布」	既存のペーパー・レポートをセクションに一括処理するように設定し、それを複数の電子メール・アドレスに配布する方法について説明します。
第 3 章「REF カーソルを使用したペーパー・レポートの作成」	PL/SQL を使用してデータをフェッチする REF カーソル問合せの定義方法について説明します。

**表 0-1 マニュアルの内容の概要（続き）**

章	内容
第 4 章「XML トランスポート・データ・ソースを使用したレポートの作成」	XML データ・ソースと SQL ベースのデータ・ソースを 1 つの問合せに結合してレポートを作成する方法について説明します。
第 5 章「テキスト・トランスポート・データ・ソースを使用するレポートの作成」	テキスト・プラグ可能データ・ソースを使用してペーパー・レポートを作成する方法について説明します。
第 6 章「Express データを使用したレポート作成」	Oracle Express をデータ・ソースとしてペーパー・レポートを作成する方法について説明します。

Oracle9i Reports の使用例は『Oracle9i Reports Developer スタート・ガイド』から参照してください。

## その他のリソース

Oracle9i Reports の使用に関する詳細は、Oracle Technology Network、Reports Builder オンライン・ヘルプおよび『Oracle9iAS Reports Services レポート Web 公開ガイド』で参照できます。

### Oracle Technology Network

Oracle Technology Network (<http://otn.oracle.com/> 英語サイト) および Oracle Technology Network Japan (<http://otn.oracle.co.jp/> 日本語サイト) では、製品の一般情報にアクセスできるほか、さらに詳しい情報を入手できます。

### Oracle9i Reports チュートリアル

このチュートリアルでは、グラフを含む簡単な JSP ベースの Web レポートを作成する方法、および同じデータ・モデルからペーパー・レポートを生成する方法について説明します。

### Reports Builder オンライン・ヘルプ

Reports Builder オンライン・ヘルプにアクセスするには、任意のダイアログ・ボックスで「ヘルプ」ボタンをクリックするか、「ヘルプ」⇒「ヘルプの目次」を選択します。

### Oracle9iAS Reports Services レポート Web 公開ガイド

このマニュアルは Oracle9iAS および Oracle9iDS のドキュメント CD に収録されています。このマニュアルでは、必要なタスクを実行できるように Oracle9iAS Reports Services を設定する方法について説明しています。たとえば、配布を目的として Reports サーバーを設定する場合は、このマニュアルで詳細情報を参照してください。



# バーコードを使用したレポートの作成

Reports Builder では、バーコードを表示する、あらゆるタイプのレポートを作成できます。Oracle9i Reports のバーコード JavaBean を使用することにより、発注や従業員 ID の追跡などを容易にするバーコードを表示した Web 用のレポートまたは紙出力用のレポートを作成できます。Reports 6i では、バーコード・フォントを使用してバーコードを生成する必要がありました。Oracle9i Reports Builder では、JavaBean によって自動的にバーコードが生成されます。

バーコード JavaBean の詳細は、『Oracle9i Reports Developer スタート・ガイド』を参照してください。

## 使用例

この項では、紙出力用と Web 用の 2 つのレポートを作成します。ペーパー・レポートには、ある会社の複数の品目を注文したある顧客に対する請求書が表示されます。バーコードは、注文の追跡情報を示します。

どちらのレポートを作成する場合も、事前に [1.1 項「この例の前提条件」](#) を参照する必要があります。

**表 1-1 バーコード Bean の例で説明する機能**

機能	項
ペーパー・レポートに対し、Java Importer を使用してバーコード JavaBean を追加する。	<a href="#">1.2.1 項「Reports Builder への Java クラスのインポート」</a>
ペーパー・レポートに対し、プログラム・ユニット・エディタを使用して PL/SQL パッケージを作成する。	<a href="#">1.2.2 項「情報を格納するパッケージの作成」</a>

表 1-1 バーコード Bean の例で説明する機能（続き）

機能	項
ペーパー・レポートに対し、バーコード JavaBean を設定するための Before Report トリガーを作成する。	1.2.3 項「Before Report トリガーの作成」
ペーパー・レポートに対し、データ・モデル・ビューとツールバーを使用して式列を含むデータ・モデルを作成する。	1.2.4 項「式列を含むデータ・モデルの作成」
単純な JSP ベースの Web レポートを作成する。	1.3.1 項「既存の HTML ファイルにおける問合せの作成」
Web レポートに対し、バーコード・データと呼び出すための式列を作成する。	1.3.2 項「データ・モデルにおける 3 つの式列の作成」
Web ソース・ビューで JSP コードを編集する。	1.3.3 項「バーコード JavaBean の初期化とそのプロパティの設定」
JSP ベースの Web レポートをブラウザに表示する。	1.3.4 項「Web に対するレポートの実行」

1.1 この例の前提条件

このマニュアルの例を作成するには、サンプル・ファイルが必要です。また、Oracle9i データベース付属のサンプル・スキーマにアクセスする必要があります。

1.1.1 サンプル・ファイル

サンプル・ファイルが手元にない場合は、この例を完了するために必要なファイルを『Oracle9i Reports Developer スタート・ガイド』から入手し、使用するマシンにインストールできます。

サンプル・ファイルを手元でインストールするには：

- 『Oracle9i Reports Developer スタート・ガイド』を起動します。
- 「インデックス」をクリックし、"Building a Paper Report with a Barcode JavaBean" および "Building a Web Report with a Barcode JavaBean" という例を検索します。この章を完了するには、両方のファイル・セットが必要です。これは、バーコード Bean を使用したペーパー・レポートと Web レポート両方の作成方法をこの章で説明するためです。
- BarcodePaper.zip と BarCodeWeb.zip ファイルを、使用するマシンの一時ディレクトリにダウンロードします ("d:\temp" など)。

4. このファイルを、ディレクトリ構造を維持したまま、使用するマシンのサンプル・ファイル用のディレクトリに展開します (d:\¥orawin90¥examples など)。

この zip ファイルには、次のファイルが含まれています。

表 1-2 バーコード JavaBean サンプル・レポートの作成に必要なファイル

ファイル	説明
<i>Examples¥BarcodeBeanPaper¥result¥ShippingManifest.pdf</i>	バーコードを含んだペーパー・レポートの最終 PDF バージョン。
<i>Examples¥BarcodeBeanPaper¥scripts¥oraclebarcode.jar</i>	バーコード JavaBean。
<i>Examples¥BarcodeBeanPaper¥source¥ShippingManifest.rdf</i>	サンプル・ペーパー・レポートのソース・ファイル。Reports Builder でこの RDF を実行すると、ペーパー・レポートの最終結果がペーパー・デザイン・ビューに表示される。
<i>Examples¥BarcodeBeanWeb¥result¥ShippingManifestWeb.jsp</i>	バーコードを含んだ Web レポートの最終 JSP バージョン
<i>Examples¥BarcodeBeanWeb¥result¥ShippingManifestWeb.rdf</i>	バーコードを含んだ Web レポートの最終 RDF バージョン
<i>Examples¥BarcodeBeanWeb¥result¥assets</i>	JSP が実行されたときに Oracle9i Reports が生成したイメージ。
<i>Examples¥BarcodeBeanWeb¥scripts¥SQL.txt</i>	入力する必要がある問合せの SQL。
<i>Examples¥BarcodeBeanWeb¥source¥shippingLabel.html</i>	Web レポートのベースとして使用する HTML ページ。
<i>Examples¥BarcodeBeanWeb¥source¥assets¥*</i>	JSP ベースの Web レポートを Web 上で正しく表示するために必要なイメージとその他のファイル。
<i>Examples¥BarcodeBeanWeb¥source¥assets¥BLAFbeige_logo.gif</i>	JSP ベースの Web レポートで使用するイメージ。

1.1.2 サンプル・スキーマへのアクセス

Oracle9i データベース付属のサンプル・スキーマに対するアクセス権があるか不明な場合は、データベース管理者に問い合せてください。この例を完了させるには、スキーマの "Order Entry" 部分に対するアクセス権が必要です。

### 1.1.3 REPORTS\_CLASSPATH 環境変数の更新

Reports JavaBean（紙出力用または Web 用）を使用する前に、いくつかのステップを実行する必要があります。まず、Bean の正しいクラス・パスを使用するように環境を設定する必要があります。ペーパー・レポートの場合は、その次に Java Importer を使用して JavaBean を Reports Builder にインポートします。Web レポートの場合は、JSP ベース（JavaServer Page）レポートから JavaBean を呼び出します。

この項では、Reports のクラス・パスを JavaBean の場所を使用して更新します。Reports Builder を起動すると、この新しいクラス・パスからバーコード Bean の場所が認識されます。

---

**注意：** この項の手順を実行しておかないと、提供されている完成済みレポート（ShippingManifestPaper.rdf および ShippingManifestWeb.rdf）も実行できません。

---

1. クラス・パスを検索します。
  - Windows の場合は、regedit を使ってレジストリを開き（レジストリを変更する前に、レジストリのバックアップ取得をお勧めします）、REPORTS\_CLASSPATH 環境変数を更新します。
  - UNIX の場合は、REPORTS\_CLASSPATH 環境変数を更新します。
2. 次の場所が反映されるようにクラス・パスを変更します。

ORACLE\_HOME/Examples/BarcodeBeanPaper/Scripts/oraclebarcode.jar

---

**注意：** ORACLE\_HOME とは Reports Builder がインストールされている場所（d:\¥orawin90¥ など）で、Examples とは例がインストールされている場所です。

---

3. 新しい環境変数を保存します。

これでレポートを作成する準備が完了しました。

## 1.2 紙出力用バーコード・レポートの作成

この項では、特定の顧客に対する請求書を表示するペーパーベースのレポートを作成します。この請求書には、顧客の住所、注文内容および注文の追跡番号を表すバーコードが表示されます。このバーコードをスキャンすることで、注文の状態を確認できます。

次に、JavaBean をインポートし、紙出力用バーコード・レポート（Acrobat PDF）を作成します。バーコードを使用する Web 用 JSP ベース・レポートの作成方法を確認する場合は、[1.3 項](#)に進んでください。

## 1.2.1 Reports Builder への Java クラスのインポート

バーコード JavaBean を使用してペーパー・レポートを作成するには、まず 2 つの Java クラスを Reports Builder にインポートする必要があります。これらの Java クラスをインポートすると、Reports Builder によってレポートの作成に必要なパッケージが自動的に作成されます。

---

---

**注意：** Web レポートを作成する場合は、JavaBean を呼び出す JSP を作成するため、この作業が不要となります。

---

---

### Java クラスをインポートするには：

1. Report Builder を起動します。

---

---

**注意：** 新しいクラス・パスを使用するには、ここで Reports Builder を起動する必要があります。

---

---

2. 「取消」をクリックして「ようこそ」ダイアログ・ボックスを閉じます。
3. 「プログラム」⇒「Java クラスをインポート ...」を選択します。
4. 「Import Java Classes」ダイアログ・ボックスの「Import Classes」フィールドに「oracle.apps.barcode.util.BarCodeConstants」と入力し、「Import」をクリックします。
5. パッケージが作成されたら、2 つ目の JavaBean である oracle.apps.barcode.BarCodeMaker をインポートします。
6. 「Close」をクリックします。
7. オブジェクト・ナビゲータで、「MODULE 1」というレポートの下を参照します。「プログラム・ユニット」ノードをクリックします。Reports Builder によって、パッケージ仕様部とパッケージ本体が 2 つずつ（BARCODECONSTANTS および BARCODEMAKER）作成されたことがわかります。

## 1.2.2 情報を格納するパッケージの作成

このレポートでは、情報を格納するパッケージを作成する必要があります。

### 情報を格納するパッケージを作成するには：

1. オブジェクト・ナビゲータで、新規レポートの下の「プログラム・ユニット」ノードをクリックします。
2. 「作成」アイコンをクリックしてプログラム・ユニットを追加します。

3. 「New Program Unit」ダイアログ・ボックスで、「globals」と入力します。
4. 「Package」ラジオ・ボタンを選択し、「OK」をクリックします。
5. 次のコードをエディタに入力します。

```
PACKAGE globals IS
    bcobj ora_java.jobject;
    barcode_to_use varchar2(256);
    tempdir varchar2(100);
    directory_sep varchar2(2);
END
```

完成済みの例 (Examples¥BarCodeBeanPaper¥source¥ShippingManifest.rdf) からコードをコピーして貼り付けることもできます。オブジェクト・ナビゲータで RDF を開きます。例の名前の下の「プログラム・ユニット」ノードを開きます。"Globals" パッケージ本体がリストされます。このパッケージを開き、その中のコードをコピーして新しいプログラム・ユニットに貼り付けます。

6. 「Compile」をクリックして、コードにエラーがないことを確認します。

---

---

**注意：** コードがコンパイルされない場合は、前述のコードを正確に入力したかどうかを確認してください。

---

---

7. コードがコンパイルされた後、「閉じる」をクリックします。
8. オブジェクト・ナビゲータで、レポート ("MODULE 1" など) を選択します。
9. 「ファイル」⇒「保存」を選択します。
10. ファイルに shippingmanifest\_< 自分のイニシャル >.rdf という名前 (shippingmanifest\_vw など) を付けて、新規ファイルを格納する新しいディレクトリ ("My Examples" など) に保存します。
11. 「保存」をクリックします。

これで、レポートのグローバル情報を格納するパッケージの作成が完了しました。

### 1.2.3 Before Report トリガーの作成

Before Report トリガーを使用して、レポートの実行前に実行される特定のタスクを初期化できます。ここでは、レポートで使用するバーコードのタイプ、およびバーコード・イメージを保存する一時ディレクトリを定義します。

1. オブジェクト・ナビゲータで、SHIPPINGMANIFEST\_< 自分のイニシャル > の下の「レポート・トリガー」ノードをクリックします。

2. 「BEFORE REPORT」をダブルクリックします。
3. エディタで、既存のテキスト（太字で示されている箇所）の間に次のコードを入力します。

```
function BeforeReport return boolean is
begin
    globals.barcode_to_use := BarCodeConstants.BAR_CODE_128;
    globals.bcobj := barcodemaker.new();
    return (TRUE);
end;
```

BarCodeConstants.BAR\_CODE\_128 の値は、有効であれば任意の値に変更できます。どの値が有効か知るには、「プログラム・ユニット」ノードの下で **Object Navigator** を使用して BarCodeConstants パッケージ仕様部を開き、パッケージの内容を確認します。

---

---

**注意：** 完成済みの例

(*Examples¥BarCodeBeanPaper¥source¥ShippingManifest.rdf*) からコードをコピーして貼り付けることもできます。オブジェクト・ナビゲータで RDF を開きます。例の名前の下の「レポート・トリガー」ノードを開きます。「BEFORE REPORT TRIGGER」がリストされているのがわかります。このトリガーを開き、エディタからコードをコピーして新しいファンクションに貼り付けます。

---

---

4. 「Compile」をクリックしてエラーがないことを確認します。

---

---

**注意：** エラーがある場合は、必要な Java クラスをインポートしたか、および入力したコードが前述のコードと一致しているか確認してください。コードを変更した場合は、再度そのコードをコンパイルします。

---

---

5. コードがコンパイルされたら、「閉じる」をクリックします。
6. レポートを保存します。

これで、レポートを実行したときにバーコード・タイプを設定するトリガーの作成が完了しました。

## 1.2.4 式列を含むデータ・モデルの作成

この項では、レポートがサンプル・スキーマからデータを取得するために使用する問合せを手動で作成します。また、JavaBean と通信してバーコードを作成する式列を作成し、その後生成されたイメージのファイル名を返します。

### 1.2.4.1 問合せの作成

1. オブジェクト・ナビゲータで、SHIPPINGMANIFEST < 自分のイニシャル > の下の「データ・モデル」をダブルクリックしてレポートのデータ・モデル・ビューを表示します。
2. ツール・パレットで「SQL 問合せ」アイコンをクリックし、データ・モデル・ビューのメイン領域（キャンバス領域）をクリックします。
3. 「SQL 問合せ文」フィールドに次のコードを入力（または貼り付け）します。

```
SELECT ALL CUSTOMERS_A1.CUST_FIRST_NAME,
       CUSTOMERS_A1.CUSTOMER_ID, CUSTOMERS_A1.CUST_LAST_NAME,
       CUSTOMERS_A1.CUST_ADDRESS.STREET_ADDRESS,
       CUSTOMERS_A1.CUST_ADDRESS.POSTAL_CODE,
       CUSTOMERS_A1.CUST_ADDRESS.CITY,
       CUSTOMERS_A1.CUST_ADDRESS.STATE_PROVINCE,
       CUSTOMERS_A1.CUST_ADDRESS.COUNTRY_ID, ORDERS.ORDER_ID,
       ORDERS.ORDER_DATE,
       ORDERS.ORDER_TOTAL, ORDER_ITEMS.LINE_ITEM_ID,
       PRODUCTS.PRODUCT_NAME,
       ORDER_ITEMS.UNIT_PRICE, ORDER_ITEMS.QUANTITY,
       COUNTRIES.COUNTRY_NAME
FROM CUSTOMERS CUSTOMERS_A1, ORDER_ITEMS, ORDERS,
     PRODUCTS, HR.COUNTRIES
WHERE ((ORDER_ITEMS.ORDER_ID = ORDERS.ORDER_ID)
      AND (ORDERS.CUSTOMER_ID = CUSTOMERS_A1.CUSTOMER_ID)
      AND (ORDER_ITEMS.PRODUCT_ID = PRODUCTS.PRODUCT_ID)
      AND (CUSTOMERS_A1.CUST_ADDRESS.COUNTRY_ID =
          HR.COUNTRIES.COUNTRY_ID))
      AND ORDERS.ORDER_ID = :P_ORDER_ID
ORDER BY order_ID, line_item_ID
```

完成済みの例（Examples¥BarCodeBeanPaper¥source¥ShippingManifest.rdf）からコードをコピーして貼り付けることもできます。オブジェクト・ナビゲータで RDF を開きます。例の名前の下で「データ・モデル」をダブルクリックして、データ・モデル・ビューを表示します。問合せ（entitled Q\_1）をダブルクリックして、「SQL 問合せ文」フィールドを表示します。

4. 「OK」をクリックします。

サンプル・スキーマが格納されているデータベースに接続されていない場合は、ここでログインする必要があります。接続文字列がわからない場合は、データベース管理者に問い合せてください。

5. バインド・パラメータ p\_order\_id が作成されたことを示すメッセージが表示された後、「OK」をクリックします。

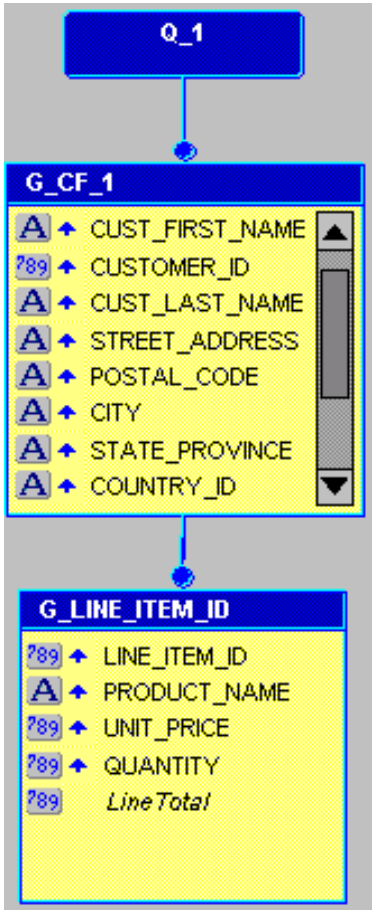


6. 作成したデータ・モデルで、次の列すべてをマウスと [Shift] キーで一度に選択し、現行の問合せの下の詳細グループまでドラッグします。

- LINE\_ITEM\_ID
- PRODUCT\_NAME
- UNIT\_PRICE
- QUANTITY

結果のデータ・モデルは次のようになります。

図 1-1 問合せのデータ・モデル



### 1.2.4.2 式列の作成

#### 式列を作成するには：

1. レポートのデータ・モデルで、「式列」アイコンをクリックして式列を作成します。
2. マスター・グループ（列名のほとんどを含んでいるメイン・グループ）内をクリックします。  
式列（CF\_1）のプロパティ・インスペクタが表示されます。
3. 列の下「データ型」プロパティの横にあるドロップダウン・リストから「Character」を選択します。
4. 「PL/SQL 式」プロパティをクリックして、式列のエディタを表示します。
5. エディタで次のコードを入力します（既存のテキストは太字で表記されています）。

```
function CF_1Formula return Char is
  myFilename varchar2(20);
  result varchar2(20);
  barcodeData VarChar2(50) := :customer_ID ||
:order_ID;
begin
  myFileName := srw.create_temporary_filename;
  barcodemaker.setBarWidthInch(globals.bcobj, 0.005);
  barcodemaker.setBaseCodeData(globals.bcobj,barcodeData);
  barcodemaker.setBarCodeType(globals.bcobj,globals.barcode_to_use);
  myfilename := barcodeData;
  barcodemaker.setFullPath(globals.bcobj, myFileName);
  barcodemaker.renderBarCode(globals.bcobj);
  return(myfilename);
end;
```

完成済みの例（*Examples¥BarCodeBeanPaper¥source¥ShippingManifest.rdf*）からコードをコピーして貼り付けることもできます。オブジェクト・ナビゲータで RDF を開きます。データ・モデル・ビューで、「CF\_1」という式列のプロパティ・インスペクタを開きます。「PL/SQL 式」プロパティをクリックして、式列のコードを表示します。

6. 「Compile」をクリックしてエラーがないことを確認します。

---

---

**注意：** エラーがある場合は、必要な Java クラスをインポートしたか、および入力したコードが前述のコードと一致しているか確認してください。コードを変更した場合は、再度そのコードをコンパイルします。

---

---

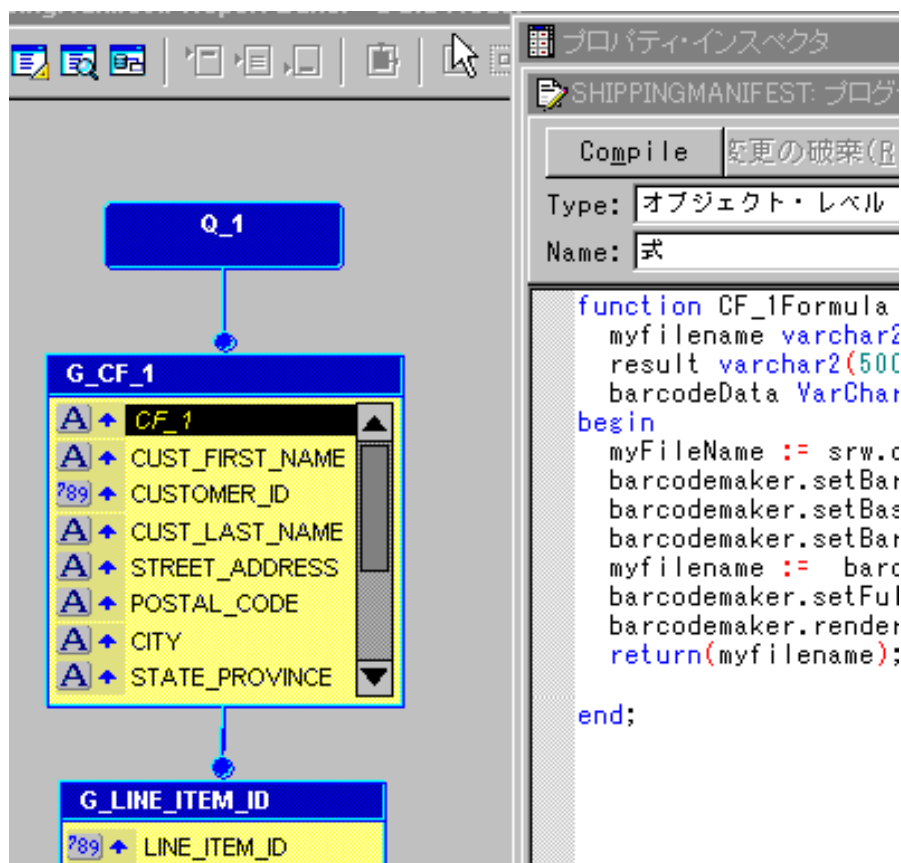
7. コードがコンパイルされたら、「閉じる」をクリックします。

8. プロパティ・インスペクタで、「ファイルからの読み込み」プロパティを検索し、これを「はい」に設定します。
9. 「ファイル形式」プロパティを「イメージ」に設定します。
10. レポートを保存します。

これでバーコード・レポートのデータ・モデルの作成が完了しました。このモデルには、バーコード情報を取得してレポートにバーコード・イメージを表示する式列が含まれています。

作成したデータ・モデルと式列の PL/SQL は次のようになります。

図 1-2 データ・モデルと式列 CF\_1 のプログラム・ユニット・エディタ



## 1.2.5 レポートのレイアウトの作成

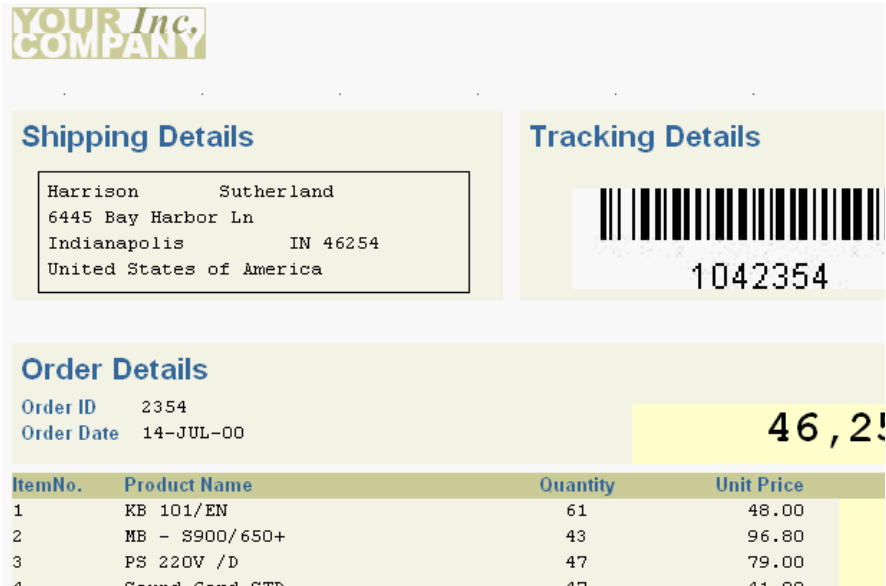
レポートを実行するには、事前にレイアウトを作成する必要があります。

### ペーパー・レイアウトを作成するには：

1. オブジェクト・ナビゲータで、レポートのノードの下にある「ペーパー・レイアウト」を右クリックし、ポップアップ・メニューから「レポート・ウィザード」を選択します。
2. 「レポート・タイプ」ページで、「次へ」をクリックして Web レイアウトとペーパー・レイアウトの両方を作成します。
3. 「スタイル」ページで、「グループ上」ラジオ・ボタンを選択し、「次へ」をクリックします。
4. 「データ・ソース」ページで「SQL Query」を選択し、「次へ」をクリックします。
5. 「データ」ページで「次へ」をクリックします。
6. 「グループ」ページで、「グループ・フィールド」列に次のフィールドが表示されていることを確認します。
  - ORDER\_ID
  - ORDER\_DATE
  - CUSTOMER\_ID
  - CUST\_FIRST\_NAME
  - CUST\_LAST\_NAME
  - STREET\_ADDRESS
  - POSTAL\_CODE
  - COUNTRY\_NAME
  - CITY
  - STATE\_PROVINCE
  - COUNTRY\_ID
  - ORDER\_TOTAL
  - CF\_1
7. 「フィールド」ページで、「>>」ボタンをクリックして、すべてのフィールドを表示します。
8. 「合計」ページで、合計が一切表示されていないことを確認してから「完了」をクリックします。

これでペーパー・レポートのレイアウトが完成しました。レポートを実行すると、ランタイム・パラメータ・フォームが表示されます。P\_ORDER\_ID の横に、「2354」と入力します。作成したレポートが、次のようにペーパー・デザイン・ビューに表示されます。

図 1-3 バーコード・ペーパー・レポートのペーパー・デザイン・ビュー



**注意：** 意図した結果が得られたかどうか分からない場合は、提供されているファイル ShippingManifest.pdf を Acrobat Reader で開きます。あるいは、ペーパーに対して ShippingManifest.rdf を実行すると、レポートがペーパー・デザイン・ビューに表示されます。

### 1.3 Web 用バーコード・レポートの作成

この項のステップは、1.2.1 項「Reports Builder への Java クラスのインポート」でインポートしたバーコード JavaBean を使用する JavaServer Pages (JSP) を使用して Web レポートを作成する方法を示しています。バーコードを含んだペーパー・レポートを作成する場合は、1.2 項「紙出力用バーコード・レポートの作成」を参照してください。

JSP ベースの Web レポートの作成に関する概要は、『Oracle9i Reports チュートリアル』を参照してください。

この項で作成するレポートは、紙出力用に作成したレポートと同じです。つまり、特定の顧客に対する請求書を表示するレポートを作成します。この請求書には、顧客の住所、注文内容および注文の追跡番号を表すバーコードが表示されます。このバーコードをスキャンすることで、注文の状態を確認できます。

提供されている JSP レポートの最終バージョンを実行すると、これらのステップでの作成結果を確認できます。ただし、Web に対してレポートを実行する前に、ソース・コード内でイメージの場所を更新する必要があるので注意してください（[1.3.3.1 項「参照先へのパスの更新」](#)を参照）。

JSP ベースの Web レポートの作成方法がわからない場合は、『Oracle9i Reports チュートリアル』で、単純な JSP ベース Web レポートの作成方法を確認してください。

---

**注意：** この項での作業を開始する前に、必要なファイルをすべてインポートしたか、および Java クラスをインポートしてクラス・パスを設定したか確認する必要があります。[1.1 項「この例の前提条件」](#) および [1.2.1 項「Reports Builder への Java クラスのインポート」](#) を参照してください。

---

### 1.3.1 既存の HTML ファイルにおける問合せの作成

JSP ベースの Web レポートを作成する際に、既存の HTML ファイルをテンプレートとして使用できます。

#### 既存の HTML ファイルに問合せを作成するには：

1. Reports Builder で、ファイル `Examples¥BarcodeBeanWeb¥source¥ShippingLabel.html` を開きます。  
  
ファイルの開き方がわからない場合は、次のステップに従ってください。
  - a. Reports Builder で「ファイル」⇒「開く」を選択します。
  - b. サンプル・ファイル用のディレクトリまでナビゲートします。これは、サンプル・ファイルを入手したディレクトリです。
  - c. `ShippingLabel.html` というファイルを検索して開きます。
2. オブジェクト・ナビゲータで、SHIPPINGLABEL の下の「データ・モデル」ノードをクリックして、レポートのデータ・モデル・ビューを表示します。
3. ツール・パレットで「SQL 問合せ」アイコンをクリックし、データ・モデル・ビューのメイン領域（キャンバス領域）をクリックします。
4. 「SQL 問合せ文」フィールドに次のコードを入力（または貼り付け）します。

```
SELECT ALL CUSTOMERS_A1.CUST_FIRST_NAME,  
       CUSTOMERS_A1.CUSTOMER_ID, CUSTOMERS_A1.CUST_LAST_NAME,  
       CUSTOMERS_A1.CUST_ADDRESS.STREET_ADDRESS,
```

```
CUSTOMERS_A1.CUST_ADDRESS.POSTAL_CODE,  
CUSTOMERS_A1.CUST_ADDRESS.CITY,  
CUSTOMERS_A1.CUST_ADDRESS.STATE_PROVINCE,  
CUSTOMERS_A1.CUST_ADDRESS.COUNTRY_ID,  
ORDERS.ORDER_ID, ORDERS.ORDER_DATE, ORDERS.ORDER_TOTAL,  
COUNTRIES.COUNTRY_NAME FROM CUSTOMERS CUSTOMERS_A1, ORDERS,  
HR.COUNTRIES  
WHERE ((ORDERS.CUSTOMER_ID = CUSTOMERS_A1.CUSTOMER_ID)  
AND (CUSTOMERS_A1.CUST_ADDRESS.COUNTRY_ID = HR.COUNTRIES.COUNTRY_ID))  
AND ORDERS.ORDER_ID = :P_ORDER_ID ORDER BY order_ID
```

---

---

**注意：** 提供されているファイル

*Examples¥BarCodeBeanWeb¥scripts¥SQL.txt* からコードをコピーして貼り付けることもできます。

---

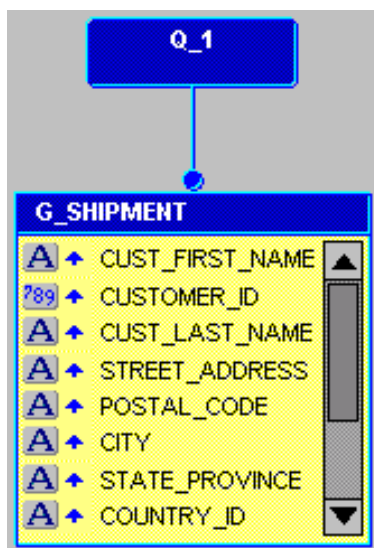
---

5. 「OK」をクリックします。

サンプル・スキーマが格納されているデータベースに接続されていない場合は、ここでログインする必要があります。接続文字列がわからない場合は、データベース管理者に問い合せてください。

6. バインド・パラメータが作成されたことを示すメッセージが表示された後、「OK」をクリックします。
7. データ・モデル・ビュー内でグループを右クリックし、「プロパティ・インスペクタ」を選択してグループのプロパティを表示します。
8. 「名前」フィールドで、既存のテキストを削除し、「G\_SHIPMENT」と入力して [Enter] キーを押します。
- これによって、グループの名前が "G\_SHIPMENT" に変更されます。
9. プロパティ・インスペクタを閉じます。
10. 作成したデータ・モデルは次のようになります。

図 1-4 JSP ベース Web レポートの問合せのデータ・モデル



11. レポートを ShippingLabel\_<自分のイニシャル>.jsp という名前で保存します。

これによって、このレポートに JSP ベースの Web ソースが作成されます。

これで、レポート用のデータを取り込む問合せの作成が完了しました。

## 1.3.2 データ・モデルにおける 3 つの式列の作成

注文の追跡番号、注文元および注文先を取得するために、レポートに 3 つの式列を作成する必要があります。

### 1.3.2.1 TrackingNumber 式列の作成

**TrackingNumber 式列を作成するには：**

1. データ・モデル・ビューのツール・パレットで、「式列」アイコンをクリックします。
2. グループ内をクリックし、CF\_1 という一時的な名前で最初の式列を作成します。
3. CF\_1 のプロパティ・インスペクタで、列の名前を TrackingNumber に変更します。
4. 「データ型」プロパティの横にあるリストから「Character」を選択します。
5. 「PL/SQL 式」プロパティの横にあるフィールドをクリックしてコード・エディタを表示します。



6. エディタで次のコードを入力します（既存のテキストは太字で表記されています）。

```
function TrackingNumberFormula return char is
begin
    return(:Customer_id||:Order_ID||:country_ID);
end;
```

完成済みの例（*Examples¥BarcodeBeanWeb¥result¥ShippingManifestWeb.rdf*）からコードをコピーして貼り付けることもできます。オブジェクト・ナビゲータで RDF を開きます。「データ・モデル」ノードをダブルクリックしてデータ・モデルを表示します。グループの中で下へスクロールして「TrackingNumber」式列を見つけ、その列のプロパティ・インスペクタを開きます。「PL/SQL 式」の横のフィールドをクリックしてコードを表示します。

7. 「Compile」をクリックします。

---

---

**注意：** コードがコンパイルされない場合は、提供されているコードを正確に入力したかどうかを確認してください。

---

---

8. コードがコンパイルされた後、「閉じる」をクリックします。

新しい式列 *TrackingNumber* が、データ・モデルに表示されます。

### 1.3.2.2 OriginScan 式列の作成

**OriginScan 式列を作成するには：**

1. G\_SHIPMENT グループに、CF\_1 という一時的な名前でもう 1 つの式列を作成します。
2. CF\_1 のプロパティ・インスペクタで、列の名前を OriginScan に変更します。
3. 「Character」データ型を選択します。
4. 「PL/SQL 式」プロパティの横にあるフィールドをクリックしてコード・エディタを表示します。
5. エディタで次のコードを入力します（既存のテキストは太字で表記されています）。

```
function OriginScanFormula return char is
begin
    return('34324-OH-US');
end;
```

完成済みの例（*Examples¥BarcodeBeanWeb¥result¥ShippingManifestWeb.rdf*）からコードをコピーして貼り付けることもできます。オブジェクト・ナビゲータで RDF を開き

ます。「データ・モデル」ノードをダブルクリックしてデータ・モデルを表示します。グループの中で下へスクロールして「OriginScan」式列を見つけ、その列のプロパティ・インスペクタを開きます。「PL/SQL 式」の横のフィールドをクリックしてコードを表示します。

6. 「Compile」をクリックします。

---

---

**注意：** コードがコンパイルされない場合は、提供されているコードを正確に入力したかどうかを確認してください。

---

---

7. コードがコンパイルされた後、「閉じる」をクリックします。  
新しい式列 *OriginScan* が、データ・モデルに表示されます。

### 1.3.2.3 DestinationScan 式列の作成

**DestinationScan 式列を作成するには：**

1. G\_SHIPMENT グループに、CF\_1 という一時的な名前でもう 1 つの式列を作成します。
2. CF\_1 のプロパティ・インスペクタで、列の名前を DestinationScan に変更します。
3. 「Character」データ型を選択します。
4. 「PL/SQL 式」プロパティの横にあるフィールドをクリックしてコード・エディタを表示します。
5. エディタで次のコードを入力します（既存のテキストは太字で表記されています）。

```
function DestinationScanFormula return char is
begin
    return(:postal_code||'-'||:state_province||'-'||:country_ID);

end;
```

完成済みの例（*Examples¥BarcodeBeanWeb¥result¥ShippingManifestWeb.rdf*）からコードをコピーして貼り付けることもできます。オブジェクト・ナビゲータで RDF を開きます。「データ・モデル」ノードをダブルクリックしてデータ・モデルを表示します。グループの中で下へスクロールして「DestinationScan」式列を見つけ、その列のプロパティ・インスペクタを開きます。「PL/SQL 式」の横のフィールドをクリックしてコードを表示します。

6. 「Compile」をクリックします。

---

---

**注意：** コードがコンパイルされない場合は、提供されているコードを正確に入力したかどうかを確認してください。

---

---

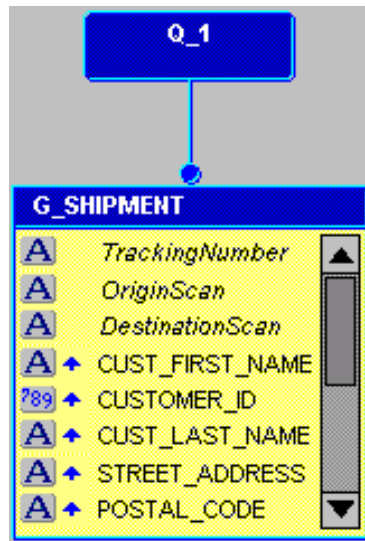
7. コードがコンパイルされた後、「閉じる」をクリックします。

新しい式列 *DestinationScan* が、データ・モデルに表示されます。

8. レポートを JSP として保存します。

これで、注文の追跡番号、注文元および注文先の値を取り込む 3 つの式列の作成が完了しました。作成したデータ・モデルは次のようになります。

図 1-5 3 つの式列を持つデータ・モデル



### 1.3.3 バーコード JavaBean の初期化とそのプロパティの設定

JSP ベースの Web レポートで JavaBean と通信するには、JSP でその JavaBean を初期化する必要があります。ペーパー・レポートで JavaBean を使用する場合は異なり、Java Importer を使用して Java クラスをインポートする必要はありません。

この項では、正しいデータを使用してバーコードが作成されるように、Bean のプロパティを設定する方法についても説明します。

ソース・ファイル (*Examples/BarcodeBeanWeb/result/ShippingManifestWeb.rdf*) を開いて Web ソースの該当箇所をレポートにコピーすると、コードを入力する手間が省けます。

### 1.3.3.1 参照先へのパスの更新

JavaBean が正しいバーコード・イメージを参照できるようにするには、正しいパスを使用してレポートを更新する必要があります。

**パスを更新するには：**

1. JSP ベースの Web レポートの Web ソース (ShippingLabel\_< 自分のイニシャル >)を開きます。  
データ・モデル・ビューで、「Web ソース」アイコンをクリックします。
2. Web ソースで、Define Path information for your barcode images というテキストを検索します。
3. このテキストの下で、イメージが生成されるディレクトリをポイントするようにパスを更新します (d:\temp\docroot\images\ and images など)。提供されているパスとの整合性を維持するようにしてください。

---

**注意：** これらのファイルが配置されるディレクトリは、Web サーバーからアクセス可能であることが必要です。通常、このディレクトリは、JSP を保持するディレクトリの下にあります。

---

### 1.3.3.2 JavaBean の初期化

**JavaBean を初期化するには：**

1. Web ソースで、Initialize the JavaBeans というテキストを検索します。
2. このテキストの下に次のコードを入力します。

```
<jsp:useBean id="BC" scope="page"
class="oracle.apps.barcode.util.BarCodeConstants" />
<jsp:useBean id="BM" scope="page" class="oracle.apps.barcode.BarCodeMaker" />
```

### 1.3.3.3 バーコード JavaBean のプロパティの設定

**JavaBean のプロパティを設定するには：**

1. Web ソースで、Setting the barcodes properties というテキストを検索します。
2. このテキストの下に次のコードを入力します。

```
<jsp:setProperty name="BM" property="BarCodeType" value="<%= BC.BAR_CODE_128
%>" />
<jsp:setProperty name="BM" property="BarWidthInch" value="0.01"/>
<jsp:setProperty name="BM" property="Directory" value="<%= BarcodePhysicalPath
%>"/>
```

### 1.3.3.4 バーコード変数の定義

バーコード変数を定義するには：

1. Web ソースで、Define variables to hold the data for the three barcodes というテキストを検索します。
2. このテキストの下に次のコードを入力します。

```
<%! private String BarCodeData1 = "12345-XX-XX"; %>
<%! private String BarCodeData2 = "12345-XX-XX"; %>
<%! private String BarCodeData3 = "12345-XX-XX"; %>
```

### 1.3.3.5 For Each ループの作成

For Each ループを作成するには：

1. Web ソースで、Replace this with your RW:FOREACH open tag というテキストを検索します。
2. この行を次のコードで置き換えます。

```
<rw:foreach id="R_G_SHIPMENT" src="G_SHIPMENT">
```

3. Replace this with your RW:FOREACH close tag というテキストを検索します。
4. この行を次のコードで置き換えます。

```
</rw:foreach>
```

### 1.3.3.6 バーコードをレンダリングする式列のコーディング

1. Web ソースで、\*\*BARCODEShippingTrackingNumber\*\* というテキストを検索します。
2. このテキストの下に次のコードを入力します。

```
<!-- Get the value of the TrackingNumber and assign it to the variable -->
<rw:getValue id="BarCodeData1" src="TrackingNumber"/>
<!-- Set the data for the barcode and the filename -->
<jsp:setProperty name="BM" property="BaseCodeData" value="<%= BarCodeData1 %>"/>
<jsp:setProperty name="BM" property="FileName" value="<%= BarCodeData1 %>"/>
<!-- Render the barcode -->
<% BM.renderBarCode(); %>
<!-- View the image in the page -->

```

"<!--" で始まる行はコメントです。コードにコメントを挿入しない場合は、これらの行を追加する必要はありません。

3. Web ソースで、**\*\*BARCODEOriginScan\*\*** というテキストを検索します。
4. このテキストの下に次のコードを入力します。

```
<!-- Get the value of the OriginScan and assign it to the variable -->
<rw:getValue id="BarCodeData2" src="OriginScan"/>
<!-- Set the data for the barcode and the filename -->
<jsp:setProperty name="BM" property="BaseCodeData" value="<%= BarCodeData2 %>"/>
<jsp:setProperty name="BM" property="FileName" value="<%= BarCodeData2 %>"/>
<!-- Render the barcode -->
<% BM.renderBarCode(); %>
<!-- View the image in the page -->

```

"<!--" で始まる行はコメントです。コードにコメントを挿入しない場合は、これらの行を追加する必要はありません。

5. Web ソースで、**\*\*BARCODEDestinationScan\*\*** というテキストを検索します。
6. このテキストの下に次のコードを入力します。

```
<!-- Get the value of the DestinationScan and assign it to the variable -->
<rw:getValue id="BarCodeData3" src="DestinationScan"/>
<!-- Set the data for the barcode and the filename -->
<jsp:setProperty name="BM" property="BaseCodeData" value="<%= BarCodeData3 %>"/>
<jsp:setProperty name="BM" property="FileName" value="<%= BarCodeData3 %>"/>
<!-- Render the barcode -->
<% BM.renderBarCode(); %>
<!-- View the image in the page -->

```

---

---

**注意：** "<!--" で始まる行はコメントです。コードにコメントを挿入しない場合は、これらの行を追加する必要はありません。

---

---

7. レポートを JSP として保存します。コードを正しくコピーしたかどうか分からない場合は、*Examples/BarcodeBeanWeb/result/ShippingManifestWeb.rdf* を確認してください。

### 1.3.4 Web に対するレポートの実行

Web レポートの作成が完了した後、このレポートを Web に対して実行し、結果を確認する必要があります。

1. 作成したレポート *ShippingManifestWeb\_<自分のイニシャル>* が、オブジェクト・ナビゲータで選択されていることを確認します。

2. ツールバーで「Web レイアウトを実行」アイコンをクリックして、レポートをブラウザに対して実行します。

作成したレポートが、次のように Web ブラウザに表示されます。

図 1-6 バーコードを含んだ最終的な JSP ベース Web レポートのスナップショット



**注意：** 意図したい結果が得られたかどうか分からない場合は、*results* ディレクトリにある *ShippingManifestWeb.html* を Web ブラウザでいつでも開くことができます。あるいは、Web に対して *ShippingManifestWeb.jsp* を実行すると、レポートがブラウザに表示されます。

## 1.4 まとめ

以上ですべての作業が完了しました。バーコード JavaBean を使用してバーコード・イメージを生成するペーパー・レポートおよび JSP ベースの Web レポートが完成しました。

この章で学習した内容は次のとおりです。

- Java Importer を使ったペーパー・レポートへの Java クラスの追加
- JSP を使用したレポートからの JavaBean の呼び出し
- PL/SQL パッケージの作成
- Before Report トリガーによる、使用するバーコード・イメージのタイプの Reports Builder に対する指定
- SQL 問合せと式列を含んだデータ・モデルの手動作成
- JSP ベースの Web レポートで使用する Web ソースの編集
- JSP での JavaBean の設定

JSP (JavaServer Pages) の詳細は、『Oracle9i Reports チュートリアル』または Reports Builder オンライン・ヘルプを参照してください。



---

## レポートの一括処理と配布

Oracle9i Reports では、1 つのレポートを同時に複数の宛先に配布できます。この機能を利用して、レポートを 1 つだけ作成し、それを任意のフォーマット（PDF や HTML など）で複数の宛先に配布できます。

この例では、提供されている単純なレポートを変更して、各セクションを個別のレポートに一括処理します。次に、サンプルの配布 XML ファイルを変更して、個々のレポートを添付ファイルとする電子メールを各宛先に送信します。また、複数の電子メールに 1 つのファイル（レポート全体）を添付して同じ電子メール・アドレスに送信します。

### 一括処理と配布について

Oracle9i Reports では、1 つのレポートを同時に複数の宛先に配布できます。この新しく拡張された配布機能を使用すると、レポートが実行されたときにそのレポートが電子メールの宛先、ポータル、プリンタまたはその他の場所に配布されるように設定できます。また、様々な形式や宛先についてデータを取り出すのは一度だけで済むため、パフォーマンスを向上させることもできます。さらに、1 回のジョブ要求で複数の宛先にレポートを公開することができるため、メンテナンスの手間も軽減されます。これをさらに調整して、ヘッダー・セクションを一部の受信者に、メイン・セクションを別の受信者に、そしてレポート全体を全く異なる受信者に送信できます。

一括処理と配布の詳細は、『Oracle9iAS Reports Services レポート Web 公開ガイド』を参照してください。

### 使用例

ある製造会社のレポート開発者として、月次情報を会社の各倉庫に配布する必要があるとします。この例では、既存のレポートを変更して、各倉庫 ID に基づいてセクションに一括処理し、倉庫ごとに個別の PDF レポートが作成されるようにします。その後、提供されている配布 XML ファイルを編集して、各セクションを添付した電子メールを個々の倉庫に送信します。

表 2-1 一括処理と配布の例で説明する機能

機能	項
セクションに一括処理するように既存のレポート（またはレポート・ウィザードで作成したレポート）を変更する。	2.2 項「既存のレポートを一括処理するための設定」
配布に使用する XML 定義ファイルを変更する。	2.3 項「配布 XML 定義の編集」
PDF フォーマットに対してレポートを実行し、そのレポートが distribution.xml ファイルに定義された宛先に電子メールで送信されるようにする。	2.4 項「レポートの実行」

2.1 この例の前提条件

このマニュアルの例を作成するには、サンプル・ファイルが必要です。また、Oracle9i データベース付属のサンプル・スキーマにアクセスする必要があります。

2.1.1 サンプル・ファイル

サンプル・ファイルが手元にない場合は、この例を完了するために必要なファイルを『Oracle9i Reports Developer スタート・ガイド』から入手し、使用するマシンにインストールできます。

サンプル・ファイルを入手してインストールするには：

- 1. 『Oracle9i Reports Developer スタート・ガイド』を起動します。
- 2. 「インデックス」をクリックし、"Bursting and Distributing a Report" という例を検索します。
- 3. ファイル Distribution.zip を、使用するマシンの一時ディレクトリにダウンロードします ("d:¥temp" など)。
- 4. このファイルを、ディレクトリ構造を維持したまま、使用するマシンのサンプル・ファイル用のディレクトリに展開します (d:¥orawin90¥examples など)。

この zip ファイルには、次のファイルが含まれています。

**表 2-2 一括処理および配布するサンプル・レポートの作成に必要なファイル**

ファイル	説明
<i>Examples¥Distribution¥source¥distribution.xml</i>	レポートの配布プロパティを制御する XML ファイル。
<i>Examples¥Distribution¥source¥Inventory_report_dist.rdf</i>	配布するレポート。
<i>Examples¥Distribution¥result¥Inventory_report_dist.rdf</i>	一括処理するレポート。
<i>Examples¥Distribution¥result¥Rep_*.pdf</i>	レポートを配布および一括処理する際に生成される PDF。

---

**注意：** 提供されている *distribution.xml* ファイルを保存しておいて再利用すれば、配布 XML ファイルを最初から作る必要がなくなります。

---

## 2.1.2 サンプル・スキーマへのアクセス

Oracle9i データベース付属のサンプル・スキーマに対するアクセス権があるか不明な場合は、データベース管理者に問い合わせてください。この例を完了させるには、スキーマの "Order Entry" 部分に対するアクセス権が必要です。

## 2.2 既存のレポートを一括処理するための設定

この章のために、一括処理に使用できる RDF ファイルがあらかじめ用意されています。このレポートを開くときは、「ファイル」⇒「接続」を選択してデータベース内のサンプル・スキーマに接続する必要があります。スキーマの "Order Entry" 部分に対する接続情報が不明である場合は、データベース管理者に問い合わせてください。

この項では、データが各倉庫 ID で一括処理されるように繰返し枠を設定します。こうして、各セクションをレポートとして出力したデータを、後から個々の倉庫に送信できます。

### 一括処理のための繰返し枠を設定するには：

1. Reports Builder で「ファイル」⇒「開く」を選択します。
2. サンプル・ファイル用のディレクトリまでナビゲートし、ファイル *inventory\_report\_dist.rdf* を開きます。

オブジェクト・ナビゲータにレポートが表示されます。

- 3. オブジェクト・ナビゲータで、レポート名の下の「ペーパー・レイアウト」ノードをダブルクリックしてペーパー・レイアウト・ビューを表示します。
- 4. ペーパー・レイアウト・ビューで、最も外側にある繰返し枠（プロパティ・インスペクタでの名前は R\_G\_WAREHOUSE\_ID）を選択し、これを削除します。

図 2-1 繰返し枠の削除



**注意：** 既存の繰返し枠を削除する必要があるのは、レポートを各セクションに一括処理するように変更するためです。後から、各セクションを各倉庫の電子メール ID に配布します。

**繰返し枠を削除するには：**

- オブジェクト・ナビゲータで枠を選択します。
- レポート名の下の「ボディ」ノードをダブルクリックして、個々のレイアウト要素のリストを開きます。
- 「ボディ」ノードの下「R\_G\_WAREHOUSE\_ID」ノードをダブルクリックします。
- ペーパー・レイアウト・ビューで繰返し枠が選択されます。
- ペーパー・レイアウト・ビューで繰返し枠を削除します。

**ヒント：** 繰返し枠内のオブジェクトもすべて削除するため、オブジェクト・ナビゲータでは繰返し枠を削除しないでください。

枠と繰返し枠の詳細は、Reports Builder オンライン・ヘルプを参照してください。

- 5. オブジェクト・ナビゲータで、「メイン・セクション」ノードを右クリックし、「プロパティ・インスペクタ」を選択します。  
メイン・セクションのプロパティ・インスペクタが表示されます。

6. セクションの下で「繰返し」プロパティを検索し、これを「G\_WAREHOUSE\_ID」に設定します。  
「繰返し」プロパティを設定するには、ドロップダウン・リストをクリックして項目を選択します。
7. レポートを `inventoryreport_dist_<自分のイニシャル>.rdf` という名前で保存します。

これで、レポートを倉庫 ID に基づいて一括処理するよう設定する処理が完了しました。

## 2.3 配布 XML 定義の編集

Oracle9i Reports の配布 XML ファイルを使用して、配布の詳細を指定できます。たとえば、電子メールを通じて配布する場合は、受信者、返信先、件名などの詳細を指定できます。

この項では、配布 XML ファイルの変更方法を説明します。配布の実施に必要な情報を入力する場所はすでに示しました。この `distribution.xml` ファイルを保存しておいて、後で使用すれば、配布を使用するたびに手動で別のファイルを作成する必要がなくなります。サンプル・ファイルには、後で役立つようにコメントも記述されています。

レポートを配布する場合は、次のいずれかの作業が必要です。

- ソース・レポート (`inventoryreport_dist.rdf`) と配布 XML ファイル (`distribution.xml`) が同じディレクトリにあることを確認する。  
または
- Reports Server からレポートを実行する場合は、宛先を XML ファイルのパスに設定する。(これについては [2.4 項「レポートの実行」](#) で説明します。)

これらのファイルは、どちらも `Examples/Distribution/source/` というディレクトリに用意されています。

配布の詳細は、『Oracle9iAS Reports Services レポート Web 公開ガイド』マニュアルを参照してください。

### 配布 XML ファイルを編集するには：

1. メモ帳などのテキスト・エディタで、Oracle 提供のファイル `distribution.xml` を開きます。
2. プレースホルダ・テキスト `<YourFilePath>` を検索し、このテキストを生成される PDF の格納先となる場所の名前で置き換えます。

例：プレースホルダ・テキストを次の場所の名前で置き換えます。

```
d:\temp\
```

これにより、パスが次のように変更されます。

```
d:¥temp¥Rep_%amp;&lt;city&gt;.pdf
```

このフル・パスを使用することで、生成された PDF ファイルが自分の d:¥temp ディレクトリに配置されます。

3. プレースホルダ・テキスト <OriginEmailAddress> を検索し、このテキストを送信者の電子メール・アドレスで置き換えます。
4. プレースホルダ・テキスト <OriginEmailAddress> のすべてのインスタンスに対してステップ 3 を実行します。
5. プレースホルダ・テキスト <DestinationEmailAddress> を検索し、最初の受信者の電子メール・アドレスで置き換えます。
6. プレースホルダ・テキスト <DestinationEmailAddress> のすべてのインスタンスに対してステップ 5 を実行します。

---

---

**注意：** この例では、複数の電子メールを一度に送信する方法は説明しません。これは、Reports Builder には組込み電子メール・アドレスが供給されていないためです。ただし、レポートを様々な電子メール・アドレスに送信する場合は、データ・モデルに受信者フィールドを作成する必要があります。さらに、distribution.xml の "ex2" セクションで、プレースホルダ・テキスト <DestinationEmailAddress> を "&amp;&lt;recipient&gt;" で置き換えます。これで、distribution.xml ファイルの最初のセクション ("ex1") を削除できます。

---

---

7. XML ファイルを、inventoryreport\_dist\_<自分のイニシャル>.rdf を保存したディレクトリに保存します。

---

---

**注意：** XML ファイルの宛先は実行時に指定できるため、XML ファイルを必ずしも RDF と同じディレクトリに保存する必要はありません。

---

---

これで配布 XML ファイルのカスタマイズが完了しました。このカスタマイズによって、個々の倉庫のレポートがすべて 1 つの電子メールに添付されて本社に送信され、各レポートが 1 つずつ添付された複数の電子メールが個々の倉庫に送信されるようになります。

## 2.4 レポートの実行

この例は Web レポートであるため、コマンドラインまたは Web ブラウザから配布できます。どちらの方法でも、次のパラメータが適用されます。

<YourPath> は、RDF ファイルおよび distribution.xml ファイルが置かれている場所です。ログイン ID と Reports サーバー名は、サンプルの RDF で使用したサンプル・スキーマに対するログイン情報です。

コマンドラインを使用する場合

次のテキストをコマンドラインに入力します。

```
RWRUN REPORT=<YourPath>/inventoryreport_dist_<YourInitials> USERID=<Your Login ID>
SERVER=<Your Server Name> DISTRIBUTE=YES DESTINATION=<YourPath>/distribution.xml
```

URL を使用する場合

次のテキストをブラウザの「場所」フィールドに入力します。

```
../RWSERVLET?REPORT=<YourPath>/inventoryreport_dist_<YourInitials>&USERID=<Your
Login ID>&SERVER=<Your Server Name>&DISTRIBUTE=YES&DESTINATION=<YourPath>
/distribution.xml
```

レポートを実行すると、倉庫 ID に基づいて倉庫ごとにファイルが作成され、指定のディレクトリに配置されます。レポートが配布されるときは、これらのファイルすべてが添付された 1 通の電子メールが 1 つのアドレスに送信されます。

distribution.xml ファイルに複数の電子メール・アドレスを設定した場合は、各倉庫（つまり各電子メール・アドレス）にその倉庫のレポートを含むファイルが添付された電子メールが送信されます。

## 2.5 まとめ

以上ですべての作業が完了しました。レポートは無事に配布されています。この章で学習した内容は次のとおりです。

- レイアウト内のセクションに一括処理されるように既存のレポートのレイアウトを変更
- distribution.xml ファイルを変更して、電子メールでレポートを配布





---

## REF カーソルを使用したペーパー・レポートの作成

Reports Builder では、REF カーソルを使用して問合せを容易に管理できます。REF カーソルは PL/SQL カーソル・データ型であるため、これを使用することで PL/SQL 問合せからカーソルを参照できます。たとえば、作成した多数の問合せをレポートで再利用する場合は、レポート・データ・モデル内で REF カーソルを使用するだけで、それらの問合せにアクセスできます。

この章では、REF カーソルの使用に関する Report Builder の機能の使用方法を学習します。このペーパー・レポートを作成するには、データ・モデル・ビューを使用して複数の問合せを含むデータ・モデルを作成した後に、レポート・ウィザードを使用してレポート・レイアウトを作成します。データ・モデル・ビューにおいて詳細な調整を手動で行います。

### REF カーソル問合せについて

REF カーソルは、問合せの中でデータをフェッチするために使用する PL/SQL データ型です。各 REF カーソル問合せは、強く型付けされた REF カーソルを戻す PL/SQL ファンクションに関連付けられています。PL/SQL ファンクションでは、確実に、REF カーソルがオープンし、REF カーソル型と一致する SELECT リストが存在する SELECT 文と関連付けられている必要があります。次のことを行うときに問合せの基準として REF カーソルを使用します。

- より簡単な SQL の管理
- レポート内での文字パラメータの使用の回避
- 他のアプリケーションとのデータ・ソースの共有
- 制御とセキュリティの向上
- サブプログラム内のロジックのカプセル化

また、REF カーソルのインプリメントにストアド・プログラム単位を使用する場合は、Oracle データベースにそのプログラム単位を同時に格納できるという利点があります。

次の図には、レポートの REF カーソル問合せに SELECT 文を使用するレポートを作成する様子が示されています。また、データベース内のパッケージに SELECT 文を格納できること



表 3-1 この REF カーソルの例で説明する機能

機能	項
REF カーソルを定義するパッケージ仕様部を作成する。	3.2 項「REF カーソル型の定義」
REF カーソルを使用する REF カーソル問合せを作成する。	3.3 項「REF カーソル問合せの作成」
データ・モデル内のオブジェクトの名前を変更して、意味のある名前を指定する。	3.4 項「データ・モデルの調整」
REF カーソル問合せ間にグループ間データ・リンクを作成して、これらの問合せ間に関係を構築する。	3.5 項「REF カーソル問合せ間のリンクの作成」
データを詳細に説明するサマリーを作成する。	3.6 項「サマリー列の追加」
レポート・ウィザードを使用して、レポート・レイアウトを作成する。	3.7 項「レイアウトの作成」
REF カーソル問合せで使用する SELECT 文を、レポートから REF カーソル型を定義するパッケージに移動する。	3.8 項「パッケージへの SELECT 文の移動」
パッケージを PL/SQL ライブラリに移動して、他のレポートとコードを共有する。	3.9 項「ライブラリへのパッケージの移動」

### 3.1 この例の前提条件

このマニュアルの例を作成するには、サンプル・ファイルが必要です。サンプル・ファイルが手元にない場合は、この例を完了するために必要なファイルを『Oracle9i Reports Developer スタート・ガイド』から入手し、使用するマシンにインストールできます。

#### サンプル・ファイルを手入手してインストールするには：

- 『Oracle9i Reports Developer スタート・ガイド』を起動します。
- 「インデックス」をクリックし、"Building a Paper Report with Ref Cursors" という例を検索します。
- ファイル refcursor.zip を、使用するマシンの一時ディレクトリにダウンロードします ("d:\temp" など)。
- このファイルを、ディレクトリ構造を維持したまま、使用するマシンのサンプル・ファイル用のディレクトリに展開します (d:\orawin90\examples など)。

この zip ファイルには、次のファイルが含まれています。

表 3-2 REF カーソルを使用したサンプル・レポートの作成に必要なファイル

ファイル	説明
<i>Examples¥RefCursor¥result¥ref_emp*.rdf</i>	RDF の様々な段階。この章の各項を読み終わるたびにこのファイルを参照すると参考になります。ファイル <i>ref_emp68.rdf</i> が最終的なレポートです。
<i>Examples¥RefCursor¥scripts¥refcursor_code.txt</i>	この章で作成する PL/SQL コード。このファイルを基準として、作成したコードが正しいことを確認するか、または単にこのファイルからコードをコピーして Reports Builder に貼り付けます。

3.1.1 サンプルの Human Resources スキーマへのアクセス

Oracle9i データベースに付属するサンプルの Human Resources スキーマに対するアクセス権があるか不明な場合は、データベース管理者に問い合せてください。

3.2 REF カーソル型の定義

REF カーソル問合せを作成するには、まず、REF カーソルを定義するパッケージ仕様部を作成します。その後で、REF カーソルを使用する問合せを作成します。この項のステップでは、REF カーソルを定義するパッケージ仕様部を作成します。

REF カーソル型を定義するには：

- 1. Report Builder を開きます。「ようこそ」ダイアログ・ボックスが表示された場合は、「新規レポートを手動で作成」をクリックしてから「OK」をクリックしてください。このダイアログ・ボックスが現れないときは、「ファイル」⇒「新規」⇒「レポート」を選択します。「新規レポートを手動で作成」をクリックし、「OK」をクリックします。
- 2. オブジェクト・ナビゲータで、UNTITLED レポート・ノードの下の「プログラム・ユニット」ノードをクリックします。
- 3. オブジェクト・ナビゲータのツールバーで「作成」アイコンをクリックしてプログラム単位を追加します。
- 4. 「New Program Unit」ダイアログ・ボックスで、「concl\_cv」をプログラム単位の名前として入力します。
- 5. 「Package Spec」をクリックし、「OK」をクリックします。
- 6. エディタで次のパッケージ仕様部定義を入力します（既存のテキストは太字で表記されています）。

```
PACKAGE concl_cv IS
  type conclclass_rec is RECORD
    (EMPLOYEE_ID NUMBER(6),
     FIRST_NAME VARCHAR2(20),
     LAST_NAME VARCHAR2(25),
```

```

EMAIL VARCHAR2(25),
PHONE_NUMBER VARCHAR2(20),
HIRE_DATE DATE,
JOB_ID VARCHAR2(10),
SALARY NUMBER(8,2),
DEPARTMENT_ID NUMBER(4);
type conclass_refcur is REF CURSOR return conclass_rec;
END;
```

このパッケージ仕様部では次の2つのことを行います。

- データベースから選択するデータを記述するレコード (conclass\_rec) を定義します。
- レコードによって記述されたフォーマットでデータを戻す REF カーソルを定義します。

---

**注意：** ファイル *Examples/RefCursor/scripts/refcursor\_code.txt* を開き、そのコードをコピーして **Reports Builder** に貼り付けることができます。

---

7. 「Compile」をクリックします。
8. コンパイル・エラーが発生した場合は、構文エラーのコードをチェックして必要に応じてコンパイルを再実行します。
9. 「閉じる」をクリックします。
10. ステップ2～8を繰り返して、次の特性を持つパッケージ仕様部をあと2つ作成します (既存のテキストは太字で表記されています)。

■ **パッケージ仕様部名: cont\_cv**

```

PACKAGE cont_cv IS
    type container_rec is RECORD
        (EMPLOYEE_ID NUMBER(6),
        START_DATE DATE,
        END_DATE DATE,
        JOB_ID VARCHAR2(10),
        DEPARTMENT_ID NUMBER(4));
    type container_refcur is REF CURSOR return container_rec;
END;
```

■ パッケージ仕様部名 :port\_cv

```
PACKAGE port_cv IS
  type portdesc_rec is RECORD
    (DEPARTMENT_ID NUMBER(4),
     DEPARTMENT_NAME VARCHAR2(30));
  type portdesc_refcur is REF CURSOR return portdesc_rec;
END;
```

---

---

**注意：** ファイル *Examples/RefCursor/scripts/refcursor\_code.txt* を開き、そのコードをコピーして Reports Builder に貼り付けることができます。

---

---

11. 「ファイル」⇒「別名保存」を選択します。このレポートに ref61\_< 自分のイニシャル>.rdf という名前を付けて、任意のディレクトリに保存します。

---

---

**注意：** レポートの設計段階では、レポートを異なるファイル名で頻繁に保存することをお勧めします。そうすれば、エラーが生じたり、変更した内容に問題がある場合でも、前に保存したファイルに簡単に戻ることができるので、その時点からの修正を加えるだけで済みます。

---

---

## 3.3 REF カーソル問合せの作成

この項で説明したように、REF カーソルを定義するパッケージ仕様部を作成すると、問合せを定義する準備が完了します。

### 参照カーソル問合せを作成するには：

1. オブジェクト・ナビゲータで、「データ・モデル」ノードをダブルクリックしてデータ・モデル・ビューを表示します。
2. 「REF カーソル問合せ」アイコンをクリックします。
3. データ・モデル・ビューのメイン領域（キャンバス領域）をクリックします。
4. キャンバス領域内でマウスをドラッグして、プログラム単位エディタを表示します。

5. プログラム単位エディタで、次の太字のコードを入力してファンクションを定義します。新規コードは太字表記になっています。

```
function q_portdescRefCurDS return port_cv.portdesc_refcur is
temp_portdesc port_cv.portdesc_refcur;
begin
    open temp_portdesc for select department_id, department_name from departments;
    return temp_portdesc;
end;
```

---

**注意：** ファイル *Examples/RefCursor/scripts/refcursor\_code.txt* を開き、そのコードをコピーして Reports Builder に貼り付けることができます。

---

6. 「Compile」をクリックします。
7. コンパイル・エラーが発生した場合は、構文エラーのコードをチェックして必要に応じてコンパイルを再実行します。
8. 「閉じる」をクリックします。データ・モデル・ビュー内にデータ・オブジェクトが表示されます。
9. データ・モデル・ビュー内で、REF カーソル問合せオブジェクト (QR\_1) をクリックし、次に「ツール」⇒「プロパティ・インスペクタ」を選択します。
10. 「一般情報」ノードの下で、「名前」プロパティを **q\_portdesc** に変更します。

**ヒント：** オブジェクトには常に意味のある名前を付けることをお勧めします。特に、多数のオブジェクトのあるレポートを作成するときには、意味のある名前を付けておくとう便利です。後にレイアウトを作成するとき、意味のある名前の問合せおよびグループが存在すると便利です。

11. [Enter] キーまたは [Return] キーを押すか、プロパティ・インスペクタ内のその他のフィールドをクリックして、変更を受け入れます。
12. プロパティ・インスペクタを閉じます。
13. ステップ 2～11 を繰り返して、次の特性を持つ問合せをあと 2 つ作成します。問合せを作成した後、必ずプロパティ・インスペクタを使用して問合せの名前を変更してください。新規コードは太字表記になっています。

■ 問合せ名: **q\_container**

```
function q_containerRefCurDS return cont_cv.container_refcur is
temp_container cont_cv.container_refcur;
begin
    open temp_container for
```

```
        select employee_id,  
               start_date  
        from job_history;  
    return temp_container  
end;
```

■ 問合せ名 : q\_conclass

```
function q_conclassRefCurDS return concl_cv.conclass_refcur is  
temp_concl concl_cv.conclass_refcur;  
begin  
    open temp_concl for  
        select employee_id,  
               first_name,  
               last_name,  
               email,  
               phone_number,  
               hire_date,  
               job_id,  
               salary,  
               department_id  
        from employees;  
    return temp_concl;  
end;
```

---

---

**注意：** ファイル *Examples/RefCursor/scripts/refcursor\_code.txt* を開き、そのコードをコピーして Reports Builder に貼り付けることができます。

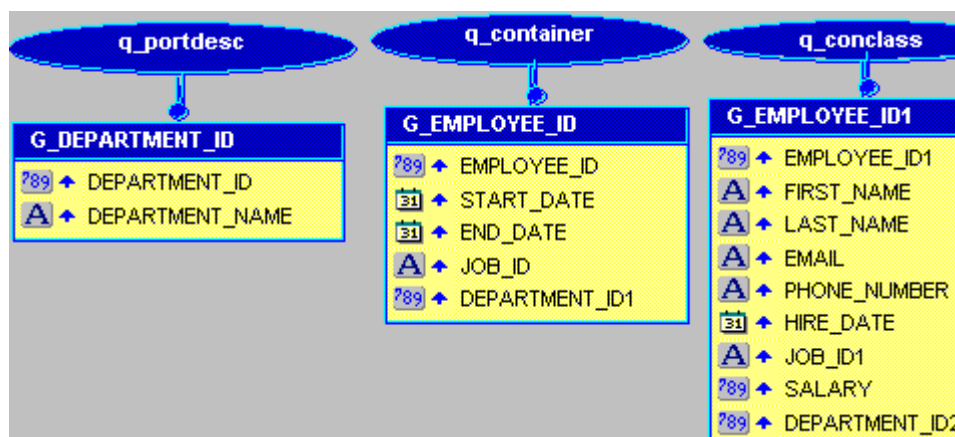
---

---



データ・モデルが、次の図と同じようになります。

図 3-2 3つの問合せを持つデータ・モデル



14. レポートを ref\_62\_< 自分のイニシャル >.rdf という名前で保存します。

---

**注意：** ファイル *Examples/RefCursor/result/ref\_emp62.rdf* を開き、データ・モデルを表示して作成結果と比較できます。

---

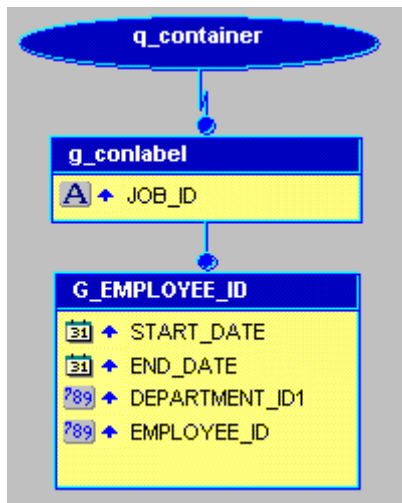
## 3.4 データ・モデルの調整

この項では、データ・モデル内の一部のオブジェクトの名前を変更して、オブジェクトに意味のある名前を指定します。また、ブレーク・グループも作成します。

**データ・モデルを調整するには：**

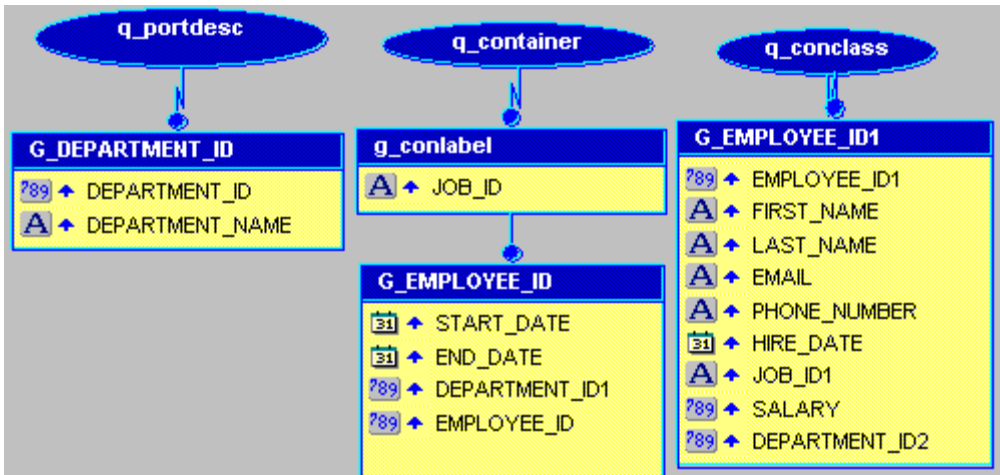
1. データ・モデル・ビューで、グループ G\_EMPLOYEE\_ID のタイトル・バーを数インチ下にドラッグしてグループ全体を移動します。
2. JOB\_ID という名前の列をクリックし、そのまま G\_EMPLOYEE\_ID の外側上部までドラッグして新規のブレーク・グループを作成します。次の図を参照してください。

図 3-3 グループを持つ問合せ



3. JOB\_ID が含まれている新規グループのタイトル・バーをクリックし、「ツール」=>「プロパティ・インスペクタ」を選択します。
4. 「一般情報」ノードで、「名前」プロパティを G\_conlabel に変更します。
5. [Enter] キーまたは [Return] キーを押すか、プロパティ・インスペクタ内のその他のフィールドをクリックして、変更を受け入れます。
6. プロパティ・インスペクタを閉じます。
7. データ・モデル・ビューにおいてデータ・モデルが次の図と同じようになります。

図 3-4 グループを持つデータ・モデル



**注意：** ファイル *Examples/RefCursor/result/ref\_emp63.rdf* を開き、データ・モデルを表示して作成結果と比較できます。

8. レポートを ref\_63\_< 自分のイニシャル >.rdf という名前で保存します。

### 3.5 REF カーソル問合せ間のリンクの作成

現時点では、作成した問合せの間に関連性はありません。問合せ間に関係を構築するには、グループ間データ・リンクを作成する必要があります。この項のステップでは、リンクを作成します。

#### REF カーソル問合せ間にリンクを作成するには：

1. データ・モデル・ビュー内で、「データ・リンク」アイコンをクリックします。
2. G\_DEPARTMENT\_ID のタイトル・バーをクリックし、そのまま G\_EMPLOYEE\_ID のタイトル・バーまでドラッグします。
3. 「q\_container」をダブルクリックします。プログラム単位エディタが表示されます。
4. ここで、SELECT 文の WHERE 句にコードを追加して、主キーおよび外部キーとして使用する列を指定します。

from job\_history の後に次のコードを追加します。

```
where :department_id = department_id;
```

セミコロン (;) の前に WHERE 句があることを確認します。

`::department_id` は、`G_DEPARTMENT_ID` 内の `DEPARTMENT_ID` を参照するバインド変数であることに注意してください。

5. 「Compile」をクリックします。
6. コンパイル・エラーが発生した場合は、構文エラーのコードをチェックして必要に応じてコンパイルを再実行します。
7. 「閉じる」をクリックします。
8. 「データ・リンク」アイコンをクリックします。
9. `G_EMPLOYEE_ID` のタイトル・バーをクリックし、そのまま `G_EMPLOYEE_ID1` のタイトル・バーまでドラッグします。
10. 「q\_conclass」をダブルクリックします。
11. ここで、SELECT 文に WHERE 句を追加します。カーソルを `FROM EMPLOYEES` とセミコロン (;) の間に挿入し、[Enter] キーまたは [Return] キーを押して新しい行を作成します。
12. 次のコードを追加します。

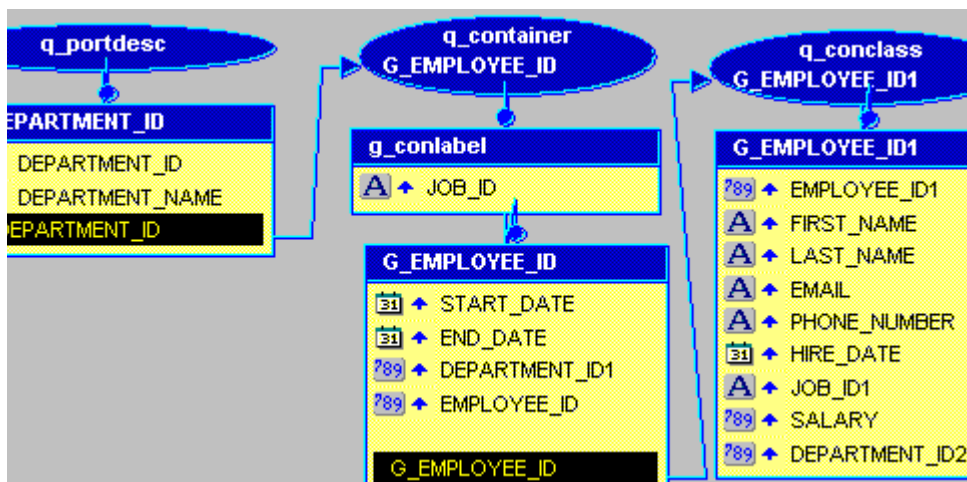
```
where :employee_id = employee_id;
```

セミコロン (;) の前に WHERE 句があることを確認します。

`:employee_id` は、`G_employee_id` 内の `EMPLOYEE_ID` 列を参照するバインド変数であることに注意してください。

13. 「Compile」をクリックします。
14. コンパイル・エラーが発生した場合は、構文エラーのコードをチェックして必要に応じてコンパイルを再実行します。
15. 「閉じる」をクリックします。
16. データ・モデルが、次の図と同じようになります。

図 3-5 リンクを持つデータ・モデル



**注意：** ファイル *Examples/RefCursor/result/ref\_emp64.rdf* を開き、データ・モデルを表示して作成結果と比較できます。

17. レポートを `ref_64_<自分のイニシャル>.rdf` という名前で保存します。

## 3.6 サマリー列の追加

現在、問合せが完成しリンクが作成されています。この項のステップでは、データを集計する列を作成します。

**サマリー列を追加するには：**

1. データ・モデル・ビュー内で、「サマリー列」アイコンをクリックします。
2. G\_EMPLOYEE\_ID グループの内部をクリックします。これにより新しい列、CS\_1 が作成されます。
3. 新しく作成した列をダブルクリックしてプロパティ・インスペクタを開きます。
4. 「一般情報」ノードで、「名前」プロパティを CS\_classcount に変更します。
5. 「サマリー」ノードで、次の設定を変更します。

表 3-3 サマリー列の設定

プロパティ	設定
ファンクション	カウント
ソース	employee_id
リセット位置	G_department_id

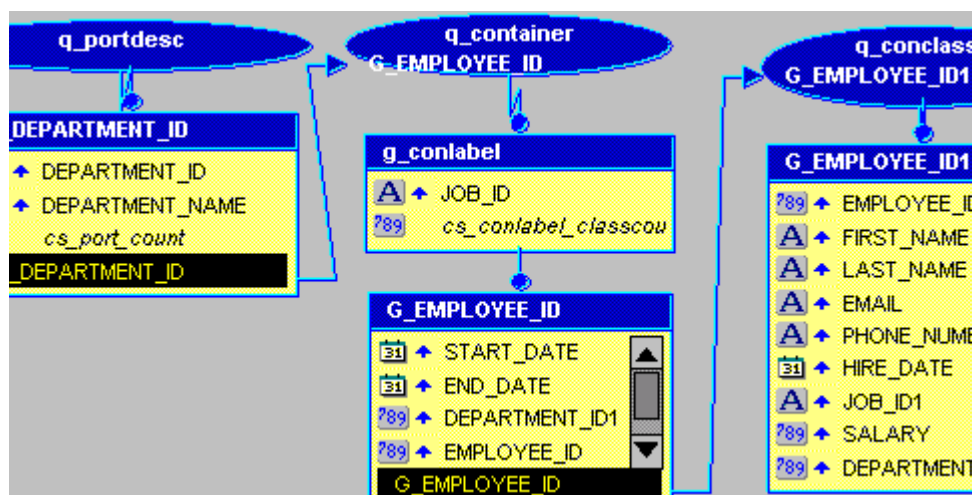
6. プロパティ・インスペクタのその他のフィールドをクリックして、変更を受け入れます。
7. プロパティ・インスペクタを閉じます。

これで、従業員数をカウントするサマリーが作成されました。レポートのレイアウト内ではサマリーを使用しませんが、後で他に対するソースとしてサマリーを使用します。
8. ステップ 1～5 を繰り返して、次の特性を持つサマリーを作成します。

作成グループ	名前	ファンクション	ソース	リセット位置
G_conlabel	CS_conlabel_classcount	合計	CS_classcount	G_conlabel
G_department_id	CS_port_count	合計	CS_conlabel_classcount	G_DEPARTMENT_ID

これらのサマリーについてはまだ学習していません。これらのサマリーの目的は、レポート・レイアウトを作成してライブ・データをプレビューするときに説明します。  
データ・モデルが、次の図と同じようになります。

図 3-6 サマリー列を持つデータ・モデル



**注意：** 作成結果を提供されているファイル（ref\_emp65.rdf）と比較することもできます。

9. レポートを ref\_65\_< 自分のイニシャル >.rdf という名前で保存します。

## 3.7 レイアウトの作成

作業データ・モデルはすでにあるので、この項のステップではレイアウトを作成します。

### ペーパー・レイアウトを作成するには：

1. オブジェクト・ナビゲータでレポート名を右クリックし、「レポート・ウィザード」を選択します。
2. 「レポート・タイプ」ページで、「ペーパー・レイアウトのみ作成」ラジオ・ボタンを選択し、「次へ」をクリックします。
3. 「スタイル」ページで、「My Employees」をタイトルとして入力します。
4. レポート・スタイルとして「グループ上」をクリックします。
5. 「次へ」をクリックします。
6. 「グループ」ページで G\_conlabel をクリックし、「下」をクリックします。

7. 次に対してステップ 5 を繰り返します。
  - G\_DEPARTMENT\_ID
  - G\_EMPLOYEE\_ID
8. 「次へ」をクリックします。
9. 「フィールド」 ページで、DEPARTMENT\_NAME をクリックし、右矢印をクリックします。
10. 次に対してステップ 8 を繰り返します。
  - EMPLOYEE\_ID
  - START\_DATE
  - END\_DATE
  - JOB\_ID
  - CS\_port\_count
11. 「次へ」をクリックします。
12. 「ラベル」 ページで、次の表に示すラベルおよび幅を入力します。

列	ラベル	幅
DEPARTMENT_NAME	Department	10
EMPLOYEE_ID	Employee ID	8
START_DATE	Start Date	9
END_DATE	End Date	9
JOB_ID	Job ID	10
CS_port_count	Total:	12

13. 「次へ」をクリックします。
14. 「テンプレート」 ページで、「事前定義テンプレート」 をクリックし、「Beige」 を選択します。
15. 「完了」をクリックします。レポートが自動的にペーパー・デザイン・ビューに表示されます。



図 3-7 REF カーソル・レポートのペーパー・デザイン・ビュー



Employment History

Department Administration		
Total:		
Department Marketing		
Total:		
Department Purchasing		
Job # PU_CLERK	Total: 7	
Start Date 08-MAR-97	End Date 01-JUL-99	
Employee #	First Name	Last Name
14	Den	Raphaely
Start Date 01-OCT-96	End Date 11-AUG-98	
Employee #	First Name	Last Name
45	John	Russell
Start Date 19-AUG-93	End Date 17-MAY-96	
Employee #	First Name	Last Name
03	Alexander	Hunold

**注意：** ファイル *Examples/RefCursor/result/ref\_emp66.rdf* を開き、ペーパー・デザイン・ビューを表示して、作成結果と比較できます。

16. レポートを ref\_66\_<自分のイニシャル>.rdf という名前で保存します。

### 3.8 パッケージへの SELECT 文の移動

現行のレポート構成では、REF カーソル問合せによって使用される SELECT 文はレポート内に置かれています。多くの場合、REF カーソル型を定義するパッケージ内に SELECT 文を配置しておくとう利です。これにより、同じ SELECT 文を使用するレポートそれぞれに SELECT 文を直接入力しなくても、パッケージを簡単に参照できるようになります。SELECT 文を変更（たとえば、句を修正または追加）する必要がある場合は、パッケージ内で SELECT 文を単純に一度更新するのみです。各レポートで更新する必要はありません。

この項のステップでは、REF カーソル型を定義するパッケージに SELECT 文を移動します。

### SELECT 文をパッケージに移動するには：

1. オブジェクト・ナビゲータで、レポートの「プログラム・ユニット」ノードをクリックします。
2. 「作成」アイコンをクリックして、プログラム単位を追加します。
3. 「New Program Unit」ダイアログ・ボックスで、プログラム単位の名前として「cont\_cv」を入力します。
4. 「Package Body」をクリックし、「OK」をクリックします。
5. 次のコードをエディタに入力します。

```
PACKAGE BODY cont_cv IS
    function query_container (p_department_id number) return container_
        refcur is tempcv_container cont_cv.container_refcur;
begin
    open tempcv_container for
        select employee_id,
               start_date,
               end_date,
               job_id,
               department_id
        from   job_history
        where  :department_id=department_id;
    return tempcv_container;
end;
END;
```

---

**注意：** ファイル *Examples/RefCursor/scripts/refcursor\_code.txt* を開き、そのコードをコピーして Reports Builder に貼り付けることができます。

---

6. 「Compile」をクリックします。
7. コンパイル・エラーが発生した場合は、構文エラーのコードをチェックして必要に応じてコンパイルを再実行します。
8. 「閉じる」をクリックします。
9. これでファンクションが定義されました。このファンクションをパッケージ仕様部に追加して参照できるようにします。他のプログラム単位がパッケージ本体内のファンクションを認識するのは、パッケージ仕様部にファンクションが記述されている場合のみです。

オブジェクト・ナビゲータで「CONT\_CV(Package Spec)」オブジェクトをダブルクリックします。

10. プログラム単位エディタで、END; 文の上に次の行を入力します。

```
function query_container (p_department_id number) return container_refcur;
```

11. 「閉じる」をクリックします。
12. 「プログラム」⇒「コンパイル」⇒「すべて」を選択します。
13. 終了したら、「OK」をクリックします。
14. オブジェクト・ナビゲータで、「プログラム・ユニット」オブジェクトの下下の「Q\_CONTAINERREFCURDS」オブジェクトをダブルクリックします。
15. コードを次のように編集します。

```
function Q_containerRefCurDS return cont_cv.container_refcur is
    temp_container cont_cv.container_refcur;
begin
    temp_container:=cont_cv.query_container (:department_id);
    return temp_container;
end;
```

終了すると、問合せのロジックはすべて、query\_container という名前のファンクション内に置かれます。これ以降 query\_container を変更するときは、この問合せおよびこの問合せを参照するその他の問合せを変更します。

---

**注意：** ファイル *Examples/RefCursor/scripts/refcursor\_code.txt* を開き、そのコードをコピーして Reports Builder に貼り付けることができます。

---

16. 「Compile」をクリックします。
17. コンパイル・エラーが発生した場合は、構文エラーのコードをチェックして必要に応じてコンパイルを再実行します。
18. 「閉じる」をクリックします。
19. 「ペーパー・デザイン」アイコンをクリックしてペーパー・デザイン・ビューにレポートを表示します。
20. レポートを ref\_67\_ < 自分のイニシャル > .rdf という名前で保存します。

#### 演習（オプション）：

レポート内の他の 2 つの問合せについてステップ 1 ～ 19 を繰り返します。

## 3.9 ライブラリへのパッケージの移動

同一の REF カーソル型および SELECT 文を使用する多数のレポートが存在する場合は、ファイルまたはデータベースに格納されている PL/SQL ライブラリ内に作成したプログラム単位を移動することにより、別のレポートでコードを容易に共有できます。この項のステップでは、プログラム単位を PL/SQL ライブラリに移動します。

### パッケージをライブラリに移動するには：

1. オブジェクト・ナビゲータで、「PL/SQL ライブラリ」オブジェクトをクリックします。
2. 「作成」アイコンをクリックして新しいライブラリを追加します。
3. 「ファイル」⇒「別名保存」を選択します。
4. 「DEPT\_CONTAINER」をライブラリとして入力します。
5. 「File System」をクリックします。
6. 「OK」をクリックします。
7. 次を示すレポートのプログラム単位を、新規に作成した DEPT\_CONTAINER ライブラリの「プログラム・ユニット」ノードにドラッグ・アンド・ドロップします。
  - CONCL\_CV(Package Spec)
  - CONT\_CV(Package Spec)
  - CONT\_CV(Package Body)
  - PORT\_CV(Package Spec)
8. DEPT\_CONTAINER を保存します。
9. ペーパー・デザイン・ビューが開いている場合は、閉じてください。
10. オブジェクト・ナビゲータで、自分のレポートの「プログラム・ユニット」ノードの下にある「CONCL\_CV(Package Spec)」、「CONT\_CV(Package Spec)」、「CONT\_CV(Package Body)」および「PORT\_CV(Package Spec)」を削除します。

---

**注意：** レポートからパッケージを削除するときにペーパー・デザイン・ビューが開いていると、エラーが発生する場合があります。

---

11. レポートの「連結ライブラリ」ノードをクリックします。
12. 「作成」アイコンをクリックして新しい連結ライブラリを追加します。
13. 「連結ライブラリ」ダイアログ・ボックスで「File System」をクリックします。

14. 「Browse」をクリックして DEPT\_CONTAINER ライブラリを検索します。このライブラリのファイル拡張子は、.PLL です。DEPT\_CONTAINER を見つけ、選択して「開く」をクリックします。
15. 「Attach」をクリックします。
16. 「プログラム」⇒「コンパイル」⇒「すべて」を選択します。
17. 「OK」をクリックして「コンパイル」ウィンドウを閉じます。
18. ツールバーで、「ペーパー・デザイン」アイコンをクリックしてレポートを表示します。

---

**注意：** レポートを表示するときにエラーが発生する場合は、ステップ 16 ～ 18 を繰り返してください。

---

19. レポートを ref\_68\_<自分のイニシャル>.rdf という名前で保存します。

**演習（オプション）：**

PL/SQL ライブラリをファイルではなくデータベースに格納してください。このオプション演習を完了するには、データベースでの " 作成 " 権限が必要です。

## 3.10 まとめ

以上ですべての作業が完了しました。REF カーソル問合せのサンプル・レポートが完成しました。この章で学習した内容は次のとおりです。

- REF カーソルを定義するパッケージ仕様部の作成
- REF カーソル問合せの作成
- REF カーソル問合せ間のデータ・リンクの作成
- データを説明するサマリーの作成
- レポート・レイアウトの作成
- パッケージへの SELECT 文の移動
- PL/SQL ライブラリへのパッケージの移動

REF カーソルの使用方法の詳細は、Reports Builder オンライン・ヘルプを参照してください。



---

# XML トランスポートابل・データ・ソースを使用したレポートの作成

Reports Builder では、任意のデータ・ソースを使用できます。この章では、Oracle9i Reports に付属の XML トランスポートابل・データ・ソースを使用する方法について説明します。

## トランスポートابل・データ・ソースについて

企業の様々なデータ・ソースから、公開の必要な情報を得ることがよくあります。これらのデータ・ソースは、SQL ベース（リレーショナル・データベース）の場合もあれば、XML や OLAP などの非 SQL ベースの場合もあります。意味のある情報を公開するために、1 つ以上のデータ・ソースからのデータを結合する必要がある場合も少なくありません。たとえば、傾向とパフォーマンスを比較するために、リレーショナル・データベースのデータをマルチディメンション・データベースのデータと結合する場合などです。

Oracle9i Reports では、あらゆるデータ・ソースのデータに対し、集計、要約、形式設定、スケジューリングなどの機能を利用できます。PDS（トランスポートابل・データ・ソース）アーキテクチャを利用すれば、ユーザー固有のデータ・ソースにも、Oracle9i Reports で利用可能なデータ・ソース（XML、JDBC、テキストおよび Express）にも接続できます。

トランスポートابل・データ・ソースの詳細は、Reports Builder オンライン・ヘルプおよび PDS API の Javadoc に関するマニュアルを参照してください。

## 使用例

米国と海外に倉庫を置いて、国際的な取引を行っているとします。これらの倉庫では、各現場でローカルに業務データを保存する、分散型管理システムを運用しています。倉庫の在庫は、現地の責任者が管理しています。ところが、計画を行うために、本社のチームがすべての倉庫の最新データを含む在庫データ（SQL 形式）にアクセスする必要が生じました。倉庫のデータは、XML ストリームとしてのみ入手できます。この例では、ローカル・データベースのデータ（つまり倉庫データ）と XML 入力からのデータを結合して、Web レポートを作成する方法を習得します。

この例では、静的XMLファイルを使用します。レポートは、受注入力システムの業務間インタフェースを使用して、XMLによる入力にオンラインでアクセスします。

表 4-1 XML PDS の例で説明する機能

機能	項
SQL 問合せを手動で作成する。	4.2.1 項「新規レポートの SQL 問合せの作成」
データ・ウィザードを使用して XML 問合せを作成する。	4.2.2 項「XML データ・ソースへのアクセスに使用する XML 問合せの作成」
SQL 問合せと XML 問合せ間にデータ・リンクを作成する。	4.2.3 項「2 つの問合せ間のデータ・リンクの作成」
レポート・ウィザードを使用してレポートのレイアウトを作成する。	4.2.4 項「レポート・ウィザードを使用したレポートのレイアウトの作成」
フォーマット・トリガーとプロシージャを使用して行カラーを交互に適用する。	4.2.5 項「レポートの行カラーの交互適用」
グループ・フィルタを使用して XML データをソートする。	4.2.6 項「グループを使用した XML データのフィルタリング」

4.1 この例の前提条件

このマニュアルの例を作成するには、サンプル・ファイルが必要です。また、Oracle9i データベース付属のサンプル・スキーマにアクセスする必要があります。

4.1.1 サンプル・ファイル

サンプル・ファイルが手元にない場合は、この例を完了するために必要なファイルを『Oracle9i Reports Developer スタート・ガイド』から入手し、使用するマシンにインストールできます。

サンプル・ファイルを手入手してインストールするには：

- 1. 『Oracle9i Reports Developer スタート・ガイド』を起動します。
- 2. 「インデックス」をクリックし、"Building a Report using an XML Pluggable Data Source" という例を検索します。
- 3. ファイル XML\_PDS.zip を、使用するマシンの一時ディレクトリにダウンロードします ("d:\temp" など)。



4. このファイルを、ディレクトリ構造を維持したまま、使用するマシンのサンプル・ファイル用のディレクトリに展開します (d:\orawin90\examples など)。

この zip ファイルには、次のファイルが含まれています。

**表 4-2 XML PDS を使用したサンプル・レポートの作成に必要なファイル**

ファイル	説明
Examples\XML_PDS\result\inventory_report.pdf	ペーパー・レポートの最終 PDF バージョン。
Examples\XML_PDS\result\inventory_report.rdf	ペーパー・レポートの最終 RDF バージョン。
Examples\XML_PDS\scripts\XMLPDS_SQL.txt	このレポートで使用する各種 SQL 文。
Examples\XML_PDS\scripts\warehouse_inventory.xml	レポート内の問合せで使用する XML データ・ソース。
Examples\XML_PDS\scripts\warehouse_inventory.xsd	レポートの XML データ・ストリーム。

---

**注意：** index.html ファイルと assets ディレクトリは、『Oracle9i Reports Developer スタート・ガイド』Web サイトの一部として使用される場合があります。これらのファイルは、削除または移動しないでください。

---

### 4.1.2 サンプル・スキーマへのアクセス

Oracle9i データベース付属のサンプル・スキーマに対するアクセス権があるか不明な場合は、データベース管理者に問い合わせてください。この例を完了させるには、スキーマの "Order Entry" 部分に対するアクセス権が必要です。通常、このスキーマにログインするにはユーザー ID とパスワード "oe/oe" を指定し、データベースの名前を入力します。

## 4.2 SQL 問合せと XML 問合せを使用したレポートの手動作成

レポートは、レポート・ウィザードを使用して作成するか、または独自に作成できます。このレポートを作成するには、SQL 問合せと XML 問合せという 2 つの問合せを作成する必要があります。

### 4.2.1 新規レポートの SQL 問合せの作成

SQL 問合せを作成するときは、Oracle9i データベースに付属するサンプル・スキーマの Order Entry 部分にアクセスする必要があります。アクセス権がない場合は、データベース

管理者に問い合せてください。通常は、接続文字列 "oe/oe@< データベース名 >" を使用してログインできます。

### SQL 問合せを作成するには：

1. Reports Builder で、「ファイル」⇒「新規」⇒「レポート」を選択します。
2. 「新規レポート」ダイアログ・ボックスで、「新規レポートを手動で作成」ラジオ・ボタンを選択して「OK」をクリックします。  
オブジェクト・ナビゲータに、新規のレポートが "MODULE 2" などの名前で表示されます。新規レポートのデータ・モデル・ビューも表示されます。
3. データ・モデル・ビューのツール・パレットで、「SQL 問合せ」アイコンをクリックします。
4. キャンバス上に問合せを作成する領域を描画します。
5. 「SQL 文」フィールドに、次のコードを入力します。

```
select      W.WAREHOUSE_ID,
            W.WAREHOUSE_NAME,
            L.CITY,
            L.STATE_PROVINCE,
            C.COUNTRY_NAME
from        WAREHOUSES W
            HR.LOCATIONS L
            HR.COUNTRIES C
where       ( W.LOCATION_ID = L.LOCATION_ID(+) )
and         ( L.COUNTRY_ID = C.COUNTRY_ID(+) )
order by   C.COUNTRY_NAME, W.WAREHOUSE_NAME
```

---

---

**注意：** 提供されているテキスト・ファイル、xmlpds\_sql.txt からコードをコピーして貼り付けることもできます。テキスト・エディタでこのファイルを開き、List of Warehouse 問合せを「SQL 文」テキスト・ボックスにコピーします。

---

---

6. 「OK」をクリックします。

---

---

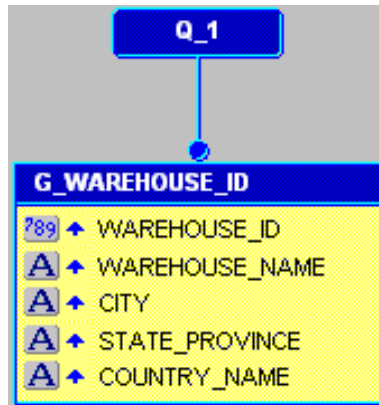
**注意：** 「接続」ダイアログ・ボックスが表示された場合は、ユーザー ID、パスワードおよびサンプル・スキーマを含んでいるデータベースの名前を入力します。

---

---

次のようにデータ・モデルがデータ・モデル・ビューに表示されます。

図 4-1 XML PDS の例の SQL 問合せで使用するデータ・モデル



7. レポートを `inventoryreport_xml_<自分のイニシャル>.rdf` という名前で保存します。

これで、レポートのデータを取得する SQL 問合せの作成が完了しました。

## 4.2.2 XML データ・ソースへのアクセスに使用する XML 問合せの作成

この項では、XML データ・ソースにアクセスするための問合せを作成します。提供されている結果のレポートを参照すると、作成した問合せが正しいかどうかを確認できます。ただし、データ定義ファイルへのパスを、提供されているサンプル・ファイルの場所の名前で更新する必要があります。

### XML 問合せを作成するには：

1. データ・モデル・ビューで、「挿入」⇒「問合せ」を選択して別の問合せを作成します。
2. データ・ウィザードが表示されたら、「ようこそ」ページで「次へ」をクリックします。
3. 「問合せ」ページで、「次へ」をクリックします。
4. 次のページで、「XML Query」をクリックし、「次へ」をクリックします。
5. 「データ・ソース定義」ページで、「問合せ定義」をクリックします。
6. 「XML 問合せの定義」ダイアログ・ボックスで、「Data Definition」の下に「Browse」をクリックして、提供されている XSD ファイル、`warehouse_inventory.xsd` を検索します。
7. ファイルを検出したらそれを選択し、「開く」をクリックします。

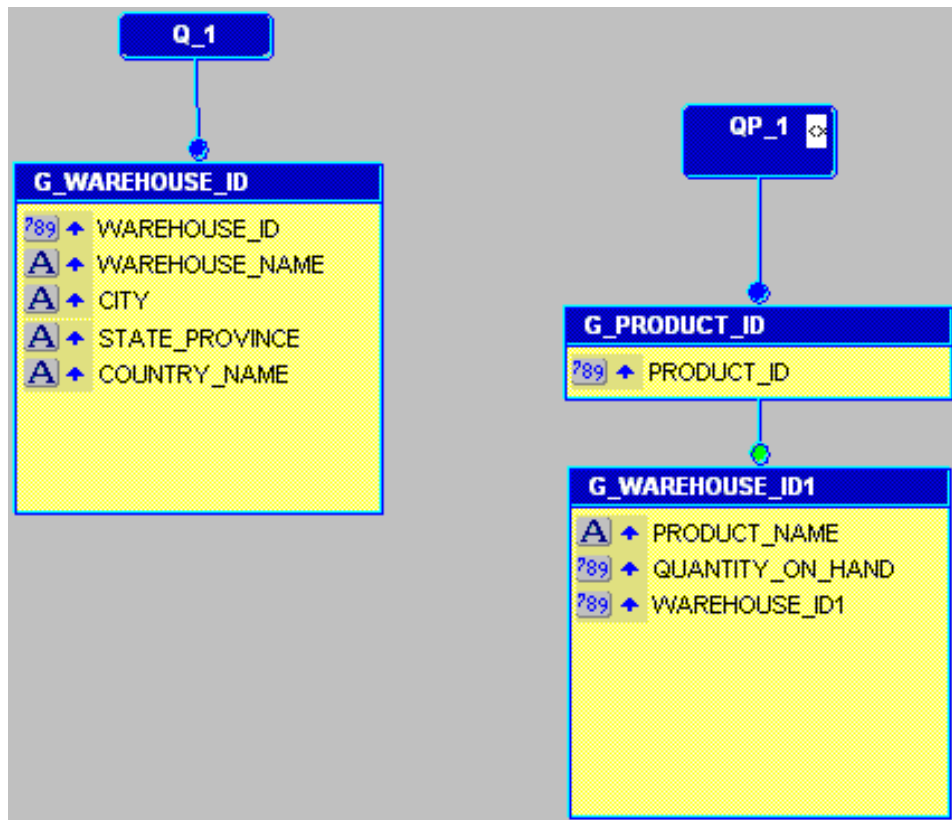
8. 「XML 問合せの定義」ダイアログ・ボックスで、「Data Source」の下に「Browse」をクリックして、データを含む XML ファイル、`warehouse_inventory.xml` を検索します。

作成したデータ定義を提供されているデータ定義と比較する場合は、データ定義の場所を、作成したファイルの場所で置き換えてください。

9. ファイルを検出したらそれを選択し、「開く」をクリックします。
10. 「XML 問合せの定義」ダイアログ・ボックスで、「OK」をクリックします。
11. データ・ウィザードで「次へ」をクリックします。
12. 「グループ」ページで、`PRODUCT_ID` をクリックし、次に右矢印ボタンをクリックします。
13. 「完了」をクリックします。

次のように、作成した 2 つの問合せがデータ・モデル・ビューに表示されます。

図 4-2 XML 問合せと SQL 問合せを含む XML PDS の例のデータ・モデル



14. レポートを `inventoryreport_xml_<自分のイニシャル>.rdf` という名前で保存します。

これで、提供されている XML データ・ソースにアクセスするための XML 問合せが作成されました。

### 4.2.3 2つの問合せ間のデータ・リンクの作成

次に、SQL 問合せと XML 問合せをリンクして、企業データと各ローカル倉庫のデータにアクセスできるようにする必要があります。

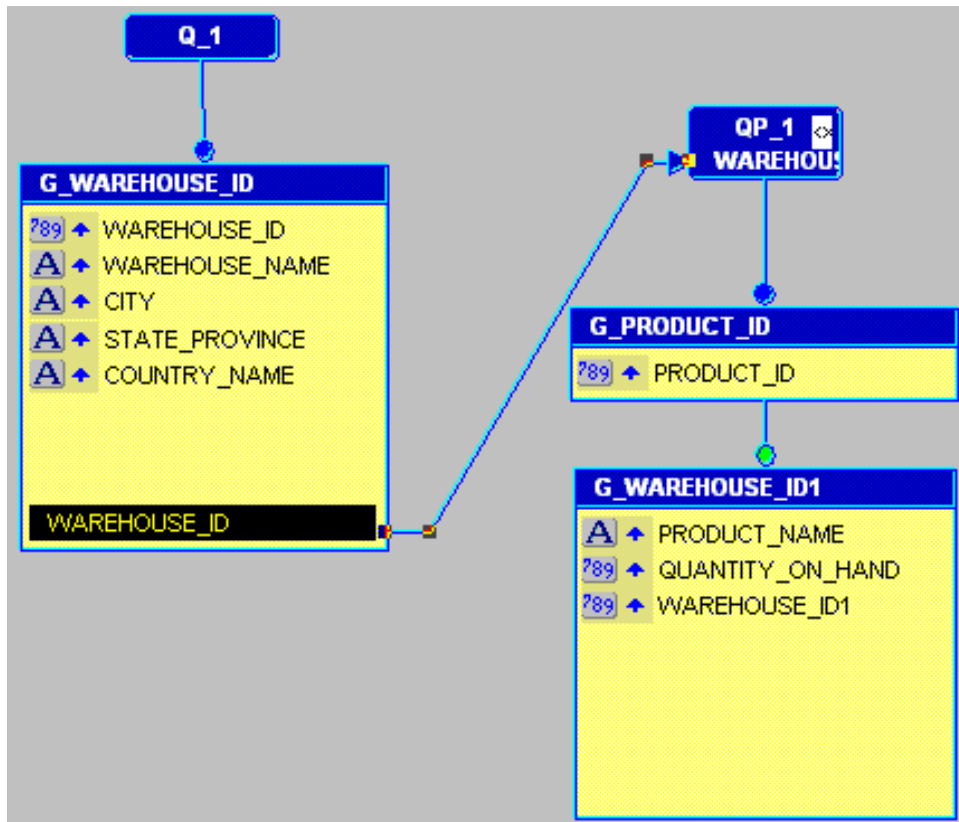
#### データ・リンクを作成するには：

1. レポートのデータ・モデル・ビュー内で、ツール・パレットの「データ・リンク」アイコンをクリックします。

2. 最初の問合せ（Q\_1）で、「WAREHOUSE\_ID」列をクリックします。
3. カーソルをそのまま 2 番目の問合せ（Q\_2）の「WAREHOUSE\_ID1」列までドラッグします。

その結果、データ・モデルは次のようになります。

図 4-3 SQL 問合せと XML 問合せ間にデータ・リンクが作成されたデータ・モデル



WAREHOUSE\_ID 列が Q\_1 の下部でハイライト表示され、そこから伸びた線が WAREHOUSE\_ID1 列を指しているのがわかります。リンクのプロパティ・インスペクタを閉じます。

4. レポートを inventoryreport\_xml\_<自分のイニシャル>.rdf という名前で保存します。

これで、2 つの問合せの WAREHOUSE\_ID 列間にデータ・リンクが作成されました。

## 4.2.4 レポート・ウィザードを使用したレポートのレイアウトの作成


レポートを実行するには、事前にレイアウトを定義する必要があります。それには、レポート・ウィザードを使用する方法が最も簡単です。

### ペーパー・レイアウトを作成するには：

1. レポートのデータ・モデル・ビュー内でキャンバスを右クリックし、メニューから「レポート・ウィザード」を選択します。
2. 「レポート・タイプ」ページで、「ペーパー・レイアウトのみ作成」ラジオ・ボタンを選択し、「次へ」をクリックします。
3. 「スタイル」ページで、「グループ上」ラジオ・ボタンを選択します。
4. 「グループ」ページで、「G\_WAREHOUSE\_ID」グループと「G\_WAREHOUSE\_ID1」グループが、「表示グループ」リスト内で下向きとしてリストされていることを確認します。
5. 「フィールド」ページで、「>>」ボタンをクリックして、すべてのフィールドを表示します。
6. 「テンプレート」ページで、「Blue」を選択した後に「完了」をクリックします。

次のようなレポートがペーパー・デザイン・ビューに表示されます。

図 4-4 XML PDS のサンプル・レポートのペーパー・デザイン・ビュー



Warehouse Id6		Warehouse Name Sydney
State Province	New South Wales	Country Name Austral
6	1820	69
6	2414	59
6	2417	29
6	2395	56
6	2396	57
6	2371	71
6	2492	41
6	2270	64
6	1742	31
6	2596	51
6	2430	65
6	2319	11

7. レポートを inventoryreport\_xml\_<自分のイニシャル>.rdf という名前で保存します。

これでペーパー・レポートのレイアウトが完成しました。

4.2.5 レポートの行カラーの交互適用

レポートの作成が完了したので、次にサマリー列を使用して行に異なるカラーを交互に適用することで、ユーザーが見やすいレポートにします。

4.2.5.1 行を数えるためのサマリー列の作成

サマリー列を作成するには：

1. レポートのデータ・モデル・ビュー内で、ツール・パレットの「サマリー列」アイコンをクリックします。



ペーパー・デザイン・ビューがまだ表示されている場合は、ツール・パレットの「データ・モデル」アイコンをクリックしてデータ・モデル・ビューを表示します。

2. XML 問合せグループ (G\_WAREHOUSE\_ID1) 内をクリックして、サマリー列を作成します。
3. プロパティ・インスペクタで、新規のサマリー列に LineNo という名前を付けます。
4. 「列タイプ」プロパティが「サマリー」に設定されていることを確認します。
5. 「データ型」プロパティのドロップダウン・ボックスで、「Number」を選択します。
6. 「サマリー」の下で、「ファンクション」ドロップダウン・ボックスから「カウント」を選択します。
7. 「ソース」プロパティのドロップダウン・ボックスから、「PRODUCT\_NAME」を選択します。
8. 新規サマリー列のプロパティ・インスペクタを閉じます。

#### 4.2.5.2 行カラーを変更するプロシージャの作成

プロシージャを作成するには：

1. オブジェクト・ナビゲータで、作成したレポート名の「プログラム・ユニット」ノードをクリックし、「作成」アイコンをクリックします。
2. 「名前」フィールドに「linecolors」と入力します。
3. 「Types」の下で「Procedure」ラジオ・ボタンが選択されていることを確認し、「OK」をクリックします。
4. エディタで次のコードを入力します（既存のテキストは太字で表記されています）。

```
PROCEDURE LineColors IS
BEGIN
  if (:LineNo mod 2 = 0)
  then
    srw.set_text_color('TextColor');
  else srw.set_text_color('black');
  end if;
END;
```

---

**注意：** xmlpds\_sql.txt ファイルに含まれているプロシージャから、このコードをコピーして貼り付けることもできます。その場合は、Line Colors Procedure 以下のテキストをコピーしてください。

---

5. 「Compile」をクリックしてプロシージャをコンパイルします。

エラーが表示された場合は、コードが正しいこと、および [4.2.5.1 項「行を数えるためのサマリー列の作成」](#) でサマリー列を作成済みであることを確認します。

6. 「閉じる」をクリックします。

### 4.2.5.3 プロシージャを呼び出す各フィールドに対するフォーマット・トリガーの作成

**フォーマット・トリガーを作成するには：**

1. オブジェクト・ナビゲータで、レポート名の下の「ペーパー・レイアウト」ノードを開き、「メイン・セクション / ボディ / M\_G\_WAREHOUSE\_ID\_GRPFR/R\_G\_WAREHOUSE\_ID/M\_G\_WAREHOUSE\_ID1\_GRPFR/R\_G\_WAREHOUSEID1」までナビゲートします。
2. 最初のフィールド「F\_PRODUCT\_ID」を選択し、プロパティ・インスペクタを開きます。

---

**注意：** 特定のフィールドが見つからない場合は、オブジェクト・ナビゲータの上部にある「検索」フィールドを使用してください。

---

3. 「拡張レイアウト」の下で、「フォーマット・トリガー」プロパティの横のフィールドをクリックします。
4. 表示されたコード・エディタで次のコードを入力します（既存のテキストは太字で表記されています）。

```
function F_PRODUCT_IDformatTrigger return Boolean is
begin
    LineColors;
    return (TRUE);
end;
```

5. 残っている次の 3 つのフィールドに対してステップ 3～5 を実行します。
  - F\_PRODUCT\_NAME
  - F\_QUANTITY\_ON\_HAND
  - F\_WAREHOUSE\_ID1
6. レポートを保存します。

7. ツールバーの「ペーパー・レイアウトを実行」アイコンをクリックして、ペーパーを対象としたレポートを実行します。

これで、レポートに対する行カラーの交互適用が完了しました。

## 4.2.6 グループを使用した XML データのフィルタリング

XML ファイルに大量のデータが含まれている場合は、データのソートやフィルタリングが必要になることがあります。グループ・フィルタと階層を作成することで、ソートとフィルタリングが可能となります。

### 4.2.6.1 繰返し枠でのグループ・フィルタの作成

**グループ・フィルタを作成するには：**

1. オブジェクト・ナビゲータで、「ペーパー・レイアウト」ノードをダブルクリックします。
2. 「ペーパー・レイアウト」ノードの下で、「メイン・セクション」ノードをダブルクリックします。
3. 「メイン・セクション」ノードの下で、「ボディ」ノードをダブルクリックします。
4. 「R\_G\_WAREHOUSE\_ID」グループを右クリックし、ポップアップ・メニューから「プロパティ・インスペクタ」を選択します。

---

**注意：** 繰返し枠が見つからない場合は、オブジェクト・ナビゲータの上部にある「検索」フィールドを使用してください。

---

5. プロパティ・インスペクタの「グループ」の下で、「フィルタ・タイプ」プロパティのドロップダウン・リストから「PL/SQL」を選択します。  
「PL/SQL フィルタ」プロパティが表示されます。
6. 「PL/SQL フィルタ」プロパティの横にあるフィールドをクリックしてコード・エディタを表示します。
7. 表示されたエディタで次のコードを入力します（既存のテキストは太字で表記されています）。

```
function G_WAREHOUSE_IDGroupFilter return boolean is
begin
    if (:LineNo mod 2 = 0)
    then
        srw.set_background_fill_color('TableCell');
    end if;
```

```
        if (:LineNo mod 2 = 1)
        then      srw.set_background_fill_color('Totals');
        end if;
    return (TRUE);
end;
```

8. 「Compile」 ボタンをクリックしてコードをコンパイルします。
9. エラーが表示された場合は、テキストが正確にコピーされているか確認してください。

---

---

**注意：** PL/SQL のコンパイル方法がわからない場合は、PL/SQL のリファレンス・マニュアルを参照してください。

---

---

10. コードがコンパイルされた後、「閉じる」をクリックします。
11. レポートを保存します。

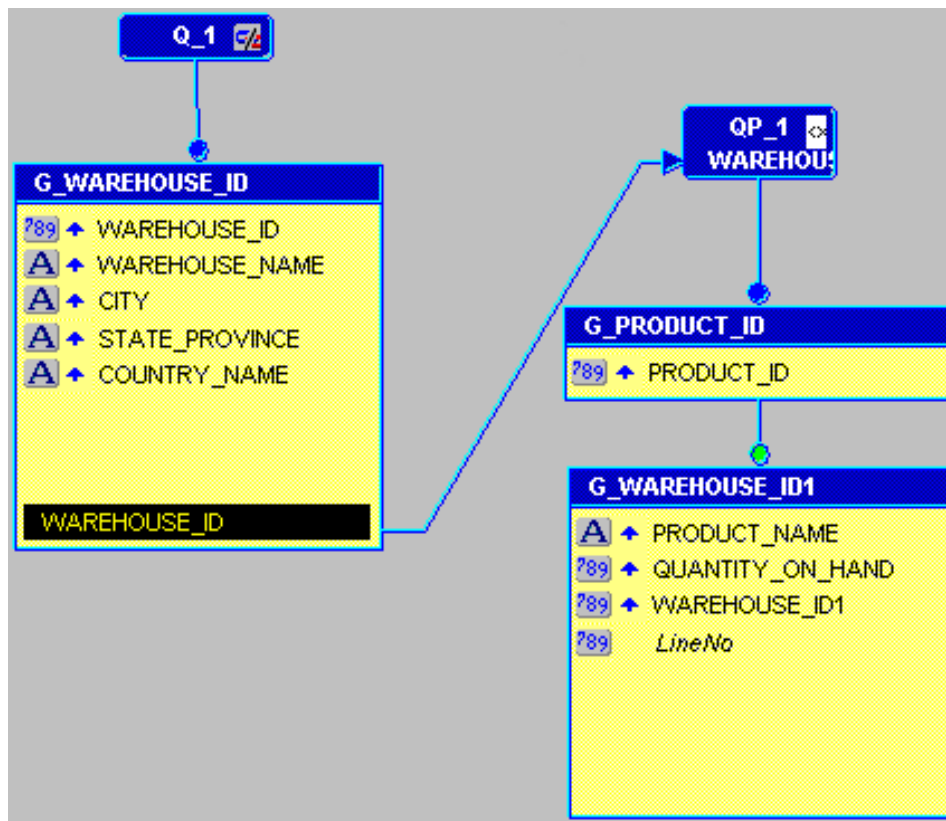
### 4.2.6.2 XML 問合せに対する階層の作成

**階層を作成するには：**

1. レポートのデータ・モデル・ビューで、XML 問合せを検索します。
2. XML 問合せ内で、「PRODUCT\_ID」列をクリックし、そのまま問合せ名とグループの間にドラッグします。

作成したデータ・モデルは次のようになります。

図 4-5 階層を持つデータ・モデル



**注意：** PRODUCT\_ID 列を問合せの他の部分より上にドラッグすることで、XML 問合せの階層が作成されています。

3. ペーパーを対象としてレポートを実行します。

提供されている RDF (*Examples¥XML\_PDS¥source¥inventoryreport.rdf*) を実行し、その結果を Reports Builder で参照することもできます。

4. レポートを `inventoryreport_xml_<自分のイニシャル>.rdf` という名前で保存します。

これで、XML データをソートするグループ・フィルタが完成しました。

### 4.3 ペーパーを対象としたレポートの実行

ペーパー・レポートを実行するには：

- 1. オブジェクト・ナビゲータで、作成したレポート（inventoryreport\_xml\_<自分のイニシャル>.rdf）が選択されていることを確認します。
- 2. 「ペーパー・レイアウトを実行」アイコンをクリックして、ペーパーを対象としてレポートを実行します。
- 3. 次のようなレポートがペーパー・デザイン・ビューに表示されます。

図 4-6 XML PDS のサンプル・レポートの最終的なペーパー・デザイン・ビュー

YOUR Inc. COMPANY		
Warehouse Name	Sydney	
City	Sydney	State Province
Country	Australia	
	Product Id	Product Name
	1733	PS 220V /UK
	1734	Cable RS232 10/A
	1737	Cable SCSI 10/FW
	1738	PS 110V /US
	1739	SDRAM - 128 MB
	1740	TD 12GB/DAT
	1742	CD-ROM 500/16x
	1745	Cable SCSI 20/WVD
	1748	PS 220V /EUR
	1749	DIMM - 256MB
	1750	DIMM - 2GB
	1755	32MB Cache /NM

## 4.4 まとめ

以上ですべての作業が完了しました。ペーパー・レポートに XML データ・ソースを使用できました。この章で学習した内容は次のとおりです。

- SQL 問合せの新規作成
- データ・ウィザードを使用した XML 問合せの作成
- SQL 問合せと XML 問合せ間のデータ・リンクの作成
- レポート・ウィザードを使用したレポートのレイアウトの作成
- フォーマット・トリガーとプロシージャによる、レポートへの行カラーの交互適用
- グループ・フィルタと階層を使用した XML データのフィルタリング

XML をデータ・ソースとして使用する方法の詳細は、『Oracle9i Reports Developer スタート・ガイド』または Reports Builder オンライン・ヘルプを参照してください。





---

# テキスト・トランスポートابل・データ・ソース を使用するレポートの作成

Reports Builder では、任意のデータ・ソースを使用できます。この章では、文字区切りテキストをデータ・ソースとして使用する方法を説明します。

## トランスポートابل・データ・ソースについて

企業の様々なデータ・ソースから、公開の必要な情報を得ることがよくあります。これらのデータ・ソースは、SQL ベース（リレーショナル・データベース）のこともあれば、XML や OLAP などの非 SQL ベースのこともあります。意味のある情報を公開するために、1 つ以上のデータ・ソースからのデータを結合する必要がある場合が少なくありません。たとえば、傾向とパフォーマンスを比較するために、リレーショナル・データベースのデータをマルチディメンション・データベースのデータと結合する場合などです。

Oracle9i Reports では、あらゆるデータ・ソースのデータに対して、集計、要約、形式設定、スケジューリングなどの機能を利用できます。PDS（トランスポートابل・データ・ソース）アーキテクチャを利用すれば、ユーザー固有のデータ・ソースにも、Oracle9i Reports で利用可能なデータ・ソース（XML、JDBC、テキストおよび Express）にも接続できます。

トランスポートابل・データ・ソースの詳細は、Reports Builder オンライン・ヘルプおよび PDS API の Javadoc に関するマニュアルを参照してください。

## 使用例

米国国勢調査局のデータを CSV（カンマ区切り値）形式でダウンロードし、レポートを生成するとします。この例では、このデータを問い合わせるペーパー・レポートを作成します。

表 5-1 このテキスト PDS の例で説明する機能

機能	項
テキスト・ファイルをトランスポートابل・データ・ソースとして認識するように Reports Builder を構成する。	5.2 項「textpds.conf ファイルの設定」
レポート・ウィザードを使用してテキスト・データ・ソースに基づくペーパー・レポートを作成する。	5.3 項「レポート・ウィザードを使用したレポートの作成」

5.1 この例の前提条件

このマニュアルの例を作成するには、サンプル・ファイルが必要です。サンプル・ファイルが手元にない場合は、この例を完了するために必要なファイルを『Oracle9i Reports Developer スタート・ガイド』から入手し、使用するマシンにインストールできます。

サンプル・ファイルを入手してインストールするには：

- 1. 『Oracle9i Reports Developer スタート・ガイド』を起動します。
- 2. 「インデックス」をクリックし、"Building a Report Using a Text Pluggable Data Source" という例を検索します。
- 3. ファイル textpds.zip を、使用するマシンの一時ディレクトリにダウンロードします ("d:\temp" など)。
- 4. このファイルを、ディレクトリ構造を維持したまま、使用するマシンのサンプル・ファイル用のディレクトリに展開します (d:\orawin90\examples など)。

この zip ファイルには、次のファイルが含まれています。

表 5-2 このテキスト PDS の例を作成するために必要なファイル

ファイル	説明
Examples\TextPDS\result\censusreport.rdf	レポート用に完成した RDF。
Examples\TextPDS\result\censusreport.pdf	完成したレポートの PDF バージョン。
Examples\TextPDS\scripts\census_csv.txt	米国国勢調査局の Web サイトから入手した文字区切りデータ。
Examples\TextPDS\scripts\config.txt	TextPDS.conf ファイルのコード。

5.2 textpds.conf ファイルの設定

テキスト・ファイルをトランスポートابل・データ・ソースとして使用するには、事前にそのテキスト・ファイル内の値の定義でテキスト PDS 構成ファイル (textpds.conf) を設定す

する必要があります。構成ファイルにこの書式情報を追加すると、Reports Builder はエントリを有効な書式として認識できるようになります。

---

**注意：** 変更を有効にするには、構成ファイルを編集した後で Reports Builder を起動する必要があります。

---

### textpds.conf ファイルを設定するには：

1. UltraEdit などのテキスト・エディタで、textpds.conf ファイルを開きます。このファイルは、*Reports\_Home/reports/conf* ディレクトリにあります。

---

**注意：** *Reports\_Home* は、使用しているコンピュータで Reports Builder がインストールされている場所です。

---

2. 別のテキスト・エディタ・ウィンドウで、提供されている config.txt ファイルを開きます。このファイルは、*Examples/TextPDS/Scripts* ディレクトリにあります。
3. config.txt 内のテキストを切り取り、textpds.conf の `</textPDS>` エントリの前に貼り付けます。これによって、textpds.conf ファイルに次のようなエントリが追加されます。

図 5-1 textpds.conf エントリのスナップショット

```
<!--Data definition for Text PDS example -->
  <fileFormat  name="CensusCSV" comment="#" deli
    <columnInfo>
      <column    name="Category"
      <column    name="Subject"      type
      <column    name="value"        type
      <column    name="percentage"
    </columnInfo>
  </fileFormat>

</textPDS>
```

---

---

**注意：** このエントリによって、Reports Builder はテキスト・ファイルを PDS として認識できるようになります。使用する PDS をレポート・ウィザードで選択する際に、テキスト・ファイルが選択肢の 1 つとして表示されます。ここでは、このファイルの各列のプロパティも定義します。

---

---

4. textpds.conf ファイルを保存します。

これで、文字区切りテキスト・データ・ソースの textpds.conf ファイルが設定されました。

## 5.3 レポート・ウィザードを使用したレポートの作成

レポートは、レポート・ウィザードを使用して作成するか、または独自に作成できます。この例の単純なレポートを作成する場合は、レポート・ウィザードを使用できます。

テキスト・トランスポータブル・データ・ソースを使用する前に、テキスト・ファイルの内容を調べる必要がある場合があります。UltraEdit やメモ帳などのテキスト・エディタで census\_csv.conf ファイルを開き、この例で使用するデータを確認できます。

### 簡単なレポートを作成するには：

1. textpds.conf ファイルは更新済みなので、Reports Builder を起動します。

**ヒント：** textpds.conf ファイルを変更するときに Reports Builder が開いていた場合は、Reports Builder を一度シャットダウンして再起動してください。

2. 「新規レポート」ダイアログ・ボックスで、「レポート・ウィザードを使う」ラジオ・ボタンを選択して「OK」をクリックします。
3. 「ようこそ」ページが表示されたら、「次へ」をクリックします。
4. 「レポート・タイプ」ページで、「ペーパー・レイアウトのみ作成」ラジオ・ボタンを選択し、「次へ」をクリックします。
5. 「タイトル」ページで、レポートのタイトルを入力し、「グループ上」ラジオ・ボタンを選択します。
6. 「次へ」をクリックします。
7. 「データ・ソース」ページで、「Text Query」をクリックし、「次へ」をクリックします。
8. 「データ・ソース定義」ページで、「問合せ定義」をクリックします。  
「問合せ定義」ダイアログ・ボックスが表示されずにエラー・メッセージが表示される場合は、構成ファイル (textpds.conf) に追加したコードに間違いがないか確認します。
9. 「Data Definition」で、ドロップダウン・リストから「CensusCSV」書式を選択します。

---

**注意：** CensusCSV 書式がこのリストに表示されるのは、この書式を textpds.conf ファイルに追加したためです。

---

10. 「Data Source」で、「Browse」をクリックして *Examples/TextPDS/Scripts* ディレクトリにある `census_csv.txt` ファイルを検索します。  
ディレクトリにこのファイルがない場合は、ドロップダウン・リストで「TXT」を選択したか確認します。
11. このファイルを選択し、「開く」をクリックします。
12. 「Location」フィールドに `census_csv.txt` ファイルが表示されたら、「OK」をクリックします。  
データ・ソース定義がテキスト・ボックスに表示されます。
13. 「次へ」をクリックします。
14. 「グループ」ページで、左側のリストで「CATEGORY」をクリックし、次に右矢印をクリックして、このグループを「グループ・フィールド」リストに移動します。
15. 「次へ」をクリックします。
16. 「フィールド」ページで、「>>」ボタンをクリックして、すべての使用可能フィールドを表示します。
17. レポート・ウィザードで「テンプレート」ページが表示されるまで「次へ」をクリックします。
18. 「テンプレート」ページで、「Beige」を選択し、「完了」をクリックします。  
次のようなレポートがペーパー・デザイン・ビューに表示されます。

図 5-2 最終的なテキスト PDS サンプル・レポートのペーパー・デザイン・ビュー

YOUR Inc.  
COMPANY

Category HISPANIC OR LATINO AND RACE		
Subject	Value	Percentage
Total population	33871648	100
Hispanic or Latino (of any race)	10966556	32.4
Mexican	8455926	25
Puerto Rican	140570	.4
Cuban	72286	.2
Other Hispanic or Latino	2297774	6.8
Not Hispanic or Latino	22905092	67.6
White alone	15816790	46.7
Category HOUSEHOLDS BY TYPE		
Subject	Value	Percentage
Total households	11502870	100
Family households (families)	7920049	68.9
With own children under 18 years	4117036	35.8
Married-couple family	5877084	51.1
With own children under 18 years	2989974	26

19. レポートを censusreport\_< 自分のイニシャル >.rdf という名前で保存します。

5.4 まとめ

以上ですべての作業が完了しました。ペーパー・レポートにテキスト・トランスポートブル・データ・ソースを使用することができました。この章で学習した内容は次のとおりです。

- textpds.conf ファイルの設定
- レポート・ウィザードを使用してテキスト・データ・ソースに基づくペーパー・レポートを作成

テキストをデータ・ソースとして使用方法の詳細は、『Oracle9i Reports Developer スタート・ガイド』、または Reports Builder オンライン・ヘルプを参照してください。このオンライン・ヘルプにアクセスするには、Reports Builder で「ヘルプ」⇒「ヘルプの目次」を選択します。





---

## Express データを使用したレポート作成

この章で説明するレポートは、Express データの Report Builder 機能を学習する目的で設計されています。ここでは、ある製品区分について地域および販売経路別の年間売上予測と実績をまとめた Express レポートを作成します。

このレポートを作成するため、まずレポート・ウィザードを使用して初期データ・モデルとレポート・レイアウトを作成します。次に、データ・モデルと Express クエリーを修正します。最後に、ペーパー・レイアウト・ビューとペーパー・デザイン・ビューでレポートの外観を拡張します。

### Express について

Express では、多次元データ・モデルを使用したオンライン分析処理（OLAP）を実現できます。このモデルは、企業データ（売上、マーケティング、財務変数など）を交差させて、その傾向やパターンを分析できるように最適化されています。

### 使用例

この例では、売上レポートを作成します。キューブにはデータが入っていて、この中から抽出するデータを決定します。キューブの各面は、カテゴリ（製品など）に含まれる可変データのリストを表します。このカテゴリとその値リストをあわせてディメンションと呼びます。各ディメンションの部分を選択して、選択部分と他のディメンションとの相互作用を分析します。このような分析をメジャーと言います。

売上を分析するメジャーの例では、時間、製品、地域区分、経路の各ディメンションからデータを選択します。Express では、問合せを作成するときに、表示する情報の範囲を広げること（たとえば全域で販売された製品の年間直接売上および間接売上）も、狭めること（たとえばカリフォルニア州で販売されたテレビの月次直接売上）もできます。

このレポートの作成ステップを示します。

表 6-1 この Express の例で説明する機能

機能	項
レポート・ウィザードを使用して Express クエリーを定義し、レポートの最初のドラフトを作成する。	6.2 項「レポート・ウィザードによる Express レポートの作成」
ディメンション値を指定して Express クエリーの効率を高める。	6.3 項「Express クエリーの修正」
データ・モデル・ビューを使用して、サマリーおよび計算合計値を追加する。	6.4 項「データ・モデルへのサマリー列およびカスタム・メジャーの追加」
サマリーおよび計算合計値をレポートのレイアウトに追加する。レポートの外観を拡張する。	6.5 項「レポート・レイアウトの拡張」

**Express データの操作に関するヒント** レポートの作成を開始する前に、Express データの操作に関するヒントを確認してください。

**注意：** Express データに関するヒントの詳細は、Reports Builder オンライン・ヘルプの「Express PDS」を参照してください。

6.1 この例の前提条件

このマニュアルの例を作成するには、サンプル・ファイルが必要です。また、Oracle Express データ・ソースへのアクセス権も必要です。

6.1.1 サンプル・ファイル

サンプル・ファイルが手元にない場合は、この例を完了するために必要なファイルを『Oracle9i Reports Developer スタート・ガイド』から入手し、使用するマシンにインストールできます。

サンプル・ファイルを手入手してインストールするには：

- 1. 『Oracle9i Reports Developer スタート・ガイド』を起動します。
- 2. 「インデックス」をクリックし、"Building a Report using an Express Data Source" という例を検索します。
- 3. ファイル Express.zip を、使用するマシンの一時ディレクトリにダウンロードします ("d:\temp" など)。
- 4. このファイルを、ディレクトリ構造を維持したまま、使用するマシンのサンプル・ファイル用のディレクトリに展開します (d:\orawin90\examples など)。

この zip ファイルには、次のファイルが含まれています。

表 6-2 Express サンプル・レポートの作成に必要なファイル

ファイル	説明
<code>Examples\Express\result\exprs.rdf</code>	この章を終えた時点で作成されている最終的なレポート。

### 6.1.2 OLAP Server へのアクセス

この Express レポートを作成する前に、OLAP Server と連携するように Report Builder を設定してください。

**注意：** Express データの設定の詳細は、Reports Builder オンライン・ヘルプの「Express データ・ソースの構成について」を参照してください。

## 6.2 レポート・ウィザードによる Express レポートの作成

レポート・ウィザードはレポート作成を開始するうえで有効な手段です。レポート・ウィザードを使用するだけでも、ユーザーの要件を満たす Express レポートを作成できます。レポート・ウィザードで作成したレポートが不十分な場合は、データ・モデル・ビュー、ペーパー・デザイン・ビューおよびペーパー・レイアウト・ビューを使用して、レポートを調整できます。このレポートでは、最初にレポート・ウィザードを使用します。この項に示すステップに従って、初期レポートを作成します。

**注意：** 詳細は、Reports Builder オンライン・ヘルプの「標準的なレポートの作成」を参照してください。

この例では、各製品区分に関する地域、経路別の月次売上予測および実績を提示するレポートを作成します。Express へのクエリーでは 2 つのメジャーを使用します。それぞれのメジャーは製品、時間、地域および経路のディメンションで構成されます。

1. レポート・ウィザードの「ようこそ」ページが表示されたら、「レポート・ウィザードを使用する」を選択し、「OK」をクリックします。
2. 「レイアウト」ページで、「ペーパー・レイアウトのみ作成」を選択し、「次へ」をクリックします。
3. 「スタイル」ページで、タイトルに「Sales Report」と入力し、レポート・スタイルとして「グループ別マトリックス」を選択します。
4. 「次へ」をクリックします。

5. 「データ・ソース・タイプ」ページで「Express Server Query」を選択し、「次へ」をクリックします。
6. 「データ」ページで「問合せ定義」をクリックします。

---

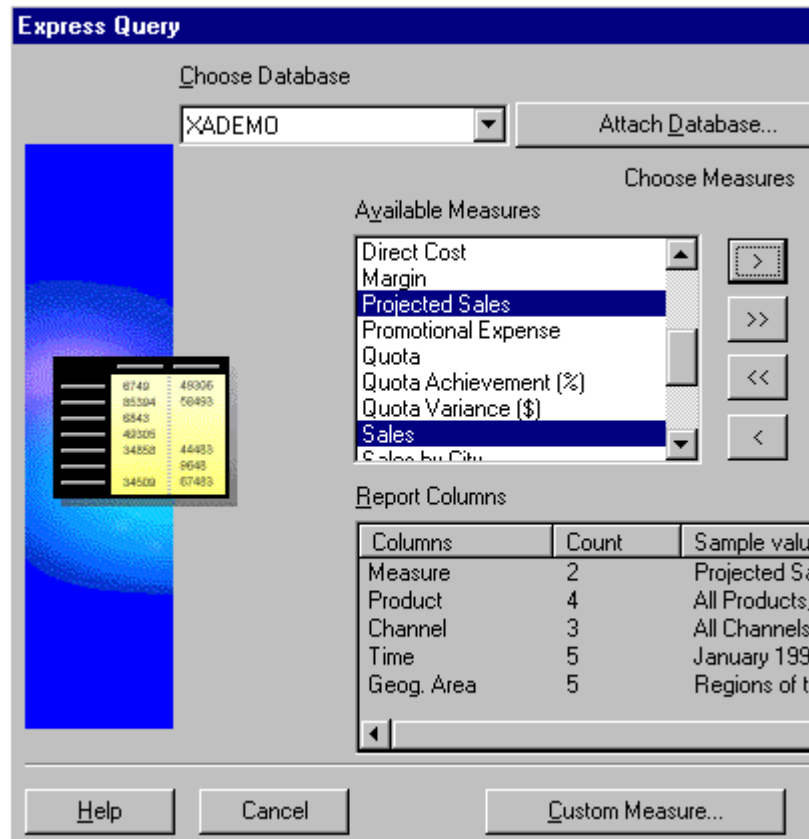
---

**注意：** まだ Oracle OLAP Server に接続していない場合は、「接続」ダイアログ・ボックスが現れます。アクセスする Oracle OLAP Server インスタンスを選択して、「OK」をクリックします。

---

---

7. このセッション中にアタッチするデータベースのパスと名前を選択するため、「Express Query」ダイアログ・ボックスで「Attach Database」を選択します。
8. 「Attach Database」ダイアログ・ボックスで、「/oec632/」というラベルの付いたディレクトリを選択します。次に「xademo.db」を選択します。これは、Oracle OLAP Server 付属のサンプル・データベースです。
9. 「Open」をクリックすると、現在のセッションにデータベースが接続されます。
10. 「Express Query」ダイアログ・ボックスで [Ctrl] キーを押しながらマウスをクリックして、「Available Measures」リストから「Sales」と「Projected Sales」を選択します。
11. 右矢印をクリックすると、「Sales」と「Projected Sales」が「Selected Measures」リスト・ボックスに移動します。「Express Query」ダイアログ・ボックスの外観は、次の図のようになります。



12. Express Query の選択内容を受け入れるには、「OK」をクリックします。この後のステップで、「Sales」および「Projected Sales」のメジャーに関連するディメンション値を修正するときにこのダイアログ・ボックスに戻ります。
13. 「データ」ページで「次へ」をクリックします。
14. 「グループ」ページで、「使用可能フィールド」リスト・ボックスの「PRODUCT」を選択し、さらに右矢印をクリックしてこのフィールドを「マトリックス・グループ・フィールド」リスト・ボックスに移動します。
15. 「Level1」を選択した後に「TIME」を選択し、さらに右矢印をクリックして「マトリックス・グループ・フィールド」ボックスに次のように表示されるようにします。



- 16. 「次へ」をクリックします。
- 17. 「行」 ページで「GEOG\_AREA」をクリックします。
- 18. 右矢印をクリックします。
- 19. 「次へ」をクリックします。
- 20. 「列」 ページで、「使用可能フィールド」 リスト・ボックス内の「CHANNEL」 をクリッ  
クします。
- 21. 右矢印をクリックします。
- 22. 「次へ」をクリックします。
- 23. 「セル」 ページで、「使用可能フィールド」 リスト・ボックスから「PROJECTED  
SALES」を選択し、「Add One」 ボタンをクリックしてこのフィールドを「マトリッ  
ク・セル・フィールド」 リスト・ボックスに移動します。
- 24. 「SALES」 も同じように移動させます。
- 25. 「次へ」をクリックします。
- 26. 「合計」 ページで「次へ」 ボタンをクリックします。  
サマリー合計の追加は、この後のステップで行います。
- 27. 「ラベル」 ページで、ラベルと幅を次のように変更します。

フィールド	ラベル	幅
SALES	Actual Sales	7
PROJECTED_SALES	Projected Sales	7
GEOG_AREA	Region	10
PRODUCT	Product:	10
TIME	Time:	10
CHANNEL	Channel	7

この後のステップで新しいレイアウト列を追加するので、ラベルの幅はここで変更して  
ください。現行のデフォルト幅では、列が次のページに渡ってしまいます。

- 28. 「次へ」をクリックします。

29. 「テンプレート」 ページで、「事前定義テンプレート」 が選択されていない場合はクリックし、リスト・ボックスで「Gray」を選択します。
30. 「完了」をクリックします。次の図のように、レポート出力が自動的にペーパー・デザイン・ビューに表示されます。



## Sales Report

Product: All Products

Time: January 1997

Region	Projected Sales		Projected Sales	
Regions of the World	9262779	8769594	4068757.5	406
Areas in the Americas	2768417.25	2620113.5	1206539.625	120
Australia	782333.875	740387	340566.09375	340
Europe	3982743.75	3773573	1779844	177
Asia	1729284.5	1635521.25	741807.6875	741

31. 「ファイル」⇒「別名保存」を選択します。レポートを任意のディレクトリに保存し、レポート名を xprs\_910.rdf とします。

---

**注意：** レポートの設計段階では、レポートを異なるファイル名で頻繁に保存することをお勧めします。そうすれば、エラーが生じたり、変更した内容に問題がある場合でも、前に保存したファイルに簡単に戻ることができるので、その時点からの修正を加えるだけで済みます。

---

## 6.3 Express クエリーの修正

この項に示すステップに従って、Express クエリーを修正します。ここまでの作業で、ある製品カテゴリに関する地域、経路別の月次売上予測および実績を示すレポートを開発しました。しかし実際は、ある製品区分について経路、地域別の年間売上予測および実績が知りたいとします。これを実現するには、表示するディメンションの値を制限します。

---

**注意：** 詳細は、Reports Builder オンライン・ヘルプの「データの選択」を参照してください。

---

この例では、「Express クエリー」ダイアログ・ボックスで次のディメンション値を指定します。

- 1997 年の売上予測および実績
  - 地域（アジア、アメリカ大陸など）
  - 製品区分（アクセサリ、オーディオ区分など）
1. ペーパー・デザイン・ビューで、「ツール」⇒「レポート・ウィザード」を選択します。
  2. 「データ」ページで、「問合せ定義」を選択します。
  3. 「Edit Query」ダイアログ・ボックスで、「Selector」を選択します。
  4. 「Selector」ダイアログ・ボックスで、ディメンション・オプションの「Time Period」をクリックします。
  5. ツールバーから、リスト・ツールを選択するために「List」ボタンをクリックします。
  6. 「List」ダイアログ・ボックスで、「Available Time Period」リスト・ボックスから「1997」を選択します。
  7. 「Select」をクリックします。それまでの選択内容に代わって「1997」が表示されます。
  8. 「OK」をクリックします。
  9. 「Selector」ダイアログ・ボックスで、ディメンション・オプションの「Geographical Area」をクリックします。
  10. ツールバーから、レベル・ツールを選択するために「Level」ボタンをクリックします。
  11. 「Select by Level」ダイアログ・ボックスで、「level」リスト・ボックスの「Continents/Regions」を選択します。
  12. 「OK」をクリックします。
  13. 「Selector」ダイアログ・ボックスで、ディメンション・オプションの「Product」をクリックします。
  14. 「Level」ボタンをクリックします。
  15. 「Select by Level」ダイアログ・ボックスで、「level」リスト・ボックスの「Divisions」を選択します。
  16. 「OK」をクリックします。
  17. 「Selector」ダイアログ・ボックスで「OK」をクリックします。
  18. 「Express Query」ダイアログ・ボックスで「OK」をクリックします。
  19. 「グループ」ページで、「マトリックス・グループ・フィールド」リスト・ボックスの「TIME」を選択します。この Express クエリーでは 1997 年度の集約データのみを取り出すので、「TIME」をブレイク・グループとして使用する必要はなくなりました。



20. 「<」 ボタンをクリックします。「マトリックス・グループ・フィールド」 リスト・ボックスに表示されるディメンションは「PRODUCT」のみになります。
21. 「スタイル」 ページで、タイトルを「1997 Sales Report」に変更します。
22. 「完了」を選択します。このレポートは、次の図のようになります。

YOUR *Inc.* COMPANY

1997 Sales Report

Product: Audio Division

	Channel	All Channels	Direct	
Region		Projected Sales	Actual Sales	Projected Sales
Areas in the Americas		5499198	7905061	3627530
Australia		1539263	2225799	1005164.6875
Europe		7315498.5	10423398	4897673
Asia		3088618.5	4461758	2020369.25

23. レポートを xprs\_920.rdf という名前で保存します。
24. 必要に応じて、このレポートを、前に xprs\_910.rdf という名前で保存したレポートと比較できます。

その際、売上予測と実績に注目します。新しいレポートの各セルには、ある製品区分に関する地域と経路の 1997 年度年間売上が示されています。一方、前のレポートでは、ある製品区分に関する地域と経路の月次売上データが表示されます。

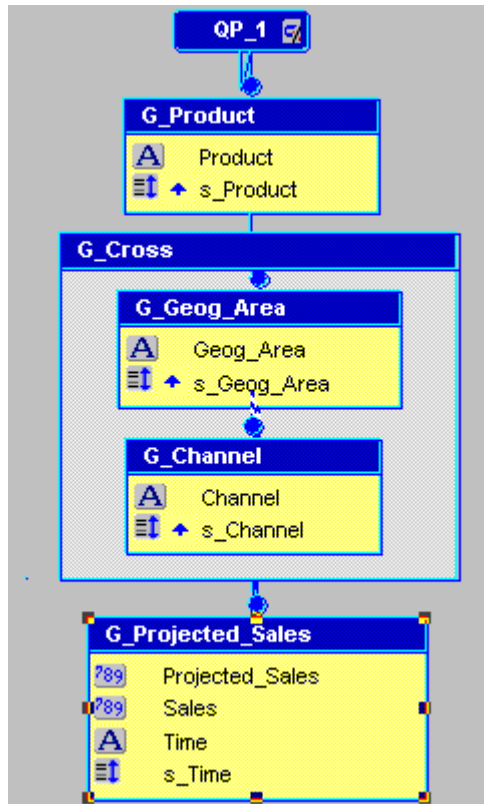
## 6.4 データ・モデルへのサマリー列およびカスタム・メジャーの追加

この項に示すステップに従って、ある製品区分に関する経路別サマリー合計が含まれるようにデータ・モデルを調整します。さらに、売上実績に対して売上予測がどのくらい正確だったかを知りたいとします。そのために、売上が売上予測を上回ったパーセントを計算するカスタム・メジャーを作成します。

まず、データ・モデル・ビューでサマリー・ツールを使用してサマリー列を作成します。

次に、「Express クエリー」ダイアログ・ボックスでカスタム・メジャー・ツールを使用してカスタム・メジャーを作成します。

作業を始める前に、データ・モデルについて概説します。



データ・モデル・ビューを見ると、S\_GEOG\_AREA や S\_CHANNEL などの列が追加されていることがわかります。これらはディメンション・ソート列です。ディメンション・ソート列はデータ・モデルにのみ表示され、ディメンションをアルファベット順ではなく論理順序でソートするときに使用する索引です。列を新規グループに移動するときは、関連するソート列もそのグループに移動させる必要があります。

この後のステップでは、「Express クエリー」ダイアログ・ボックスでソート・ツールを使用してディメンション値をソートします。

## 6.4.1 データ・オブジェクト名の変更

1. データ・モデル・ビューが表示されていない場合は、オブジェクト・ナビゲータで対象レポートのノードの下にある「データ・モデル」ボタンをダブルクリックします。
2. QP\_1 を選択します。
3. 「ツール」⇒「プロパティ・インスペクタ」を選択します。

Express クエリーを変更する場合は、「問合せ」ノードの下の「Express クエリー」プロパティを選択します。

4. 「一般情報」ノードで、「名前」プロパティを「QP\_SALES」に変更します。
5. [Enter] キーを押すかプロパティの外側をクリックすると、値が受け入れられます。プロパティ・インスペクタを閉じます。
6. ステップ 2～5 を繰り返し、G\_PROJECTED SALES グループの「名前」プロパティを G\_SALES\_DATA に変更します。
7. レポートを xprs\_931.rdf という名前で保存します。

## 6.4.2 サマリー列の作成

この例では、G\_Cross グループにサマリー列を 2 つ追加します。各サマリー列では、ある製品区分に関する経路ごと（すべての経路、直接、間接）の売上予測および実績の合計を計算します。

1. データ・モデル・ビューで「サマリー列」アイコンをクリックし、次に G\_Cross グループをクリックします。
2. 「ツール」⇒「プロパティ・インスペクタ」を選択します。
3. 売上予測には次のプロパティを設定します。

ノード	プロパティ	値
一般情報	名前	CS_PjSalesPerChannel
列	積順序	G_CHANNEL
サマリー	ソース	PROJECTED_SALES
サマリー	リセット位置	G_CHANNEL

4. [Enter] キーを押すかプロパティの外側をクリックすると、値が受け入れられます。
5. プロパティ・インスペクタを閉じます。
6. ステップ 1～4 を繰り返して、売上実績のサマリー列を作成します。次のプロパティを設定します。

ノード	プロパティ	値
一般情報	名前	CS_SalesPerChannel
列	積順序	G_CHANNEL
サマリー	ソース	SALES
サマリー	リセット位置	G_CHANNEL

7. レポートを xprs\_932.rdf という名前で保存します。

### 6.4.3 カスタム・メジャーの作成

この例では、各製品区分について地域ごとに売上実績が売上予測を上回ったパーセントを計算するカスタム・メジャーを作成します。作業としては、「Express クエリー」ダイアログ・ボックス内のカスタム・メジャー・ツールを使用して「Increase」という新規メジャーを作成します。

1. データ・モデル・ビューで、「QP\_Sales」問合せオブジェクトをダブルクリックして「Express クエリー」ダイアログ・ボックスを開きます。
2. 「Express クエリー」ダイアログ・ボックスの下部にある「カスタム・メジャー」を選択します。
3. 「新規」を選択して、「カスタム・メジャー - 新規」ダイアログ・ボックスを開きます。
4. 「名前」ボックスに「INCREASE」と入力します。
5. 「説明」ボックスに「Increase」と入力します。
6. 「カテゴリ」ボックスの「演算子」の下にある「テンプレート」を選択します。「選択肢」の下にテンプレートのリストが表示されます。
7. 左カッコを選択し、「挿入」をクリックします。「式」ボックスに左カッコが表示されます。
8. 「カテゴリ」ボックスの「Express Objects」の下にある「メジャー」を選択します。
9. 「F.SALES」を選択し、「挿入」を選択します。
10. 次の表のとおり to 式を作成します。

カテゴリ	サブカテゴリ	選択	または次のように入力
演算子	数値	マイナス符号	-
Express Objects	メジャー	F.WHATIF	F.WHATIF
演算子	テンプレート	右カッコ	)
演算子	数値	スラッシュ	/

カテゴリ	サブカテゴリ	選択	または次のように入力
Express Objects	メジャー	F.WHATIF	F.WHATIF
演算子	数値	アスタリスク	*

11. 「式」ボックスで、アスタリスクの後ろに「100」を入力します。

12. ここまで終了すると、式は次の図のようになります。

**Custom Measures -- New**

Name: INCREASE Description: Increase

Database: Express Demo Database (XADEMO)

Expression: ( F.SALES - F.WHATIF ) / F.WHATIF \* 100

Category: Express Objects, Dimensions, Measures, Relations, Operators, Comparison, Logical, **Numeric**, Template, Functions, All, Conversion, Date and Time

Choices: -, \*, \*\*, /, **+**

Insert

+ Addition

13. 「OK」をクリックします。「カスタム・メジャー」ダイアログ・ボックスの「カスタム・メジャー」テキスト・ボックスに、「Increase」が表示されます。

14. 「閉じる」をクリックします。
15. 「Express クエリー」ダイアログ・ボックスで、「利用可能なメジャー」ボックスをスクロールします。今度は、「Increase」がアルファベット順に表示されます。「Increase」を選択し、「Add One」ボタンをクリックします。「選択済みのメジャー」ボックスの「Projected Sales」と「Sales」の下に「Increase」が表示されます。
16. 「OK」をクリックしてデータ・モデルに戻ります。  
作成したカスタム・メジャー INCREASE が、グループ G\_SALES\_DATA に表示されます。
17. 「ペーパー・レイアウトを実行」ボタンをクリックしてペーパー・デザイン・ビューにレポートを表示します。作成したサマリー列もカスタム・メジャーもこのレポートでは使用できません。これは、レポート・レイアウトにフィールドとして追加されていないためです。レポート・レイアウトへの追加は次の作業で行います。
18. レポートを xprs\_933.rdf という名前で保存します。

## 6.5 レポート・レイアウトの拡張

この項ではレポート・レイアウトの再配置、6.3 項「[Express クエリーの修正](#)」で作成したサマリー列とカスタム・メジャー列の追加、およびオブジェクトの書式設定を行って、レポートの外観を拡張します。これらの変更作業には、ペーパー・レイアウト・ビューとペーパー・デザイン・ビューを使用します。

### 6.5.1 レポートへのサマリー・フィールド挿入

1. オブジェクト・ナビゲータで、レポートのノードの下にある「ペーパー・レイアウト」アイコンをダブルクリックしてペーパー・レイアウト・ビューを表示します。
2. オブジェクト・ナビゲータとペーパー・レイアウト・ビューが横に並ぶように画面を調整してください。「ペーパー・レイアウト」、「メイン・セクション」、「本体」および残りのネストされたノード (M\_G\_PRODUCT\_GRPFR や R\_G\_PRODUCT ノードなど) を拡張します。
3. オブジェクト・ナビゲータの「検索」フィールドに「M\_G\_CROSS\_GRPFR」と入力して、このオブジェクトを検索します。ペーパー・レイアウト・ビューで、マスター・クロス・マトリックス枠が選択されます。
4. 選択された枠を、下に約 1/4 インチ拡張します。
5. オブジェクト・ナビゲータで「F\_CHANNEL」を選択します。
6. 「親枠を選択」ボタンをクリックして、親枠である R\_G\_CHANNEL を選択します。

---

---

**注意：**「親枠を選択」ボタンはツールバーの右端にあるので、このボタンが表示されていない場合はペーパー・レイアウト・ウィンドウのサイズを変更する必要があります。

---

---

7. 選択された枠を下に約 1/4 インチ拡張します。
8. 「フィールド」ボタンをクリックします。
9. 四角形をクリックして、そのまま「F\_PROJECTED\_SALES」フィールドのすぐ下の領域にドラッグします。これによってフィールド・オブジェクトが挿入されます。
10. 「ツール」⇒「プロパティ・インスペクタ」を選択します。
11. プロパティ・インスペクタで、「フィールド」ノードの下「ソース」プロパティを「CS\_PjSalesPerChannel」に設定します。
12. オブジェクト・ナビゲータでオブジェクトを選択し、その名前を「 F\_PjSalesPerChannel」に変更します。
13. このフィールドを配置して、書式を次のように変更します。
  - 「塗りカラー」ボタンをクリックして、塗りカラーをライト・イエローに変更します。
  - 「テキスト・カラー」ボタンをクリックして、テキスト・カラーをダーク・ブラウンに変更します。
  - 「線カラー」ボタンをクリックして、フィールド周囲の境界線をダーク・ブラウンにします。

---

---

**注意：** オブジェクトをレイアウトに配置する際に、必要に応じて格子指定をオンまたはオフに切り替えることができます。そのためには、「表示」⇒「格子指定」を選択します。チェック・マークはそのオプションがオンになっていることを示します。

---

---

14. ステップ 8～9 を繰り返して、F\_SALES の下にも新規オブジェクトを配置します。
15. プロパティ・インスペクタで、「フィールド」ノードの下「ソース」プロパティを「CS\_SalesPerChannel」に設定します。
16. オブジェクト・ナビゲータでオブジェクトを選択し、その名前を「 F\_SalesPerChannel」に変更します。

---

---

**注意：** 塗りカラーやテキスト・カラー、境界線はさきほど作成した「F\_PjSalesPerChannel」フィールドと同じカラーにします。

---

---

17. 「テキスト」ボタンをクリックします。
18. 四角形をクリックしてそのままドラッグし、F\_GEO\_AREA のすぐ下の領域に置きます。
19. 四角形の中に、「Totals:」と入力します。
20. 「中央に整列」ボタンをクリックして、テキスト・オブジェクトを中央に揃えます。さきほど作成したサマリー・フィールドと同じ書式に変更します。
21. 「ペーパー・デザイン」アイコンをクリックしてペーパー・デザイン・ビューで変更内容を確認します。
22. レポートを xprs\_941.rdf という名前で保存します。

## 6.5.2 レポートへのカスタム・メジャー・フィールド挿入

この項では、レポート・レイアウトにフィールド・オブジェクトを挿入することにより、[6.4.3 項「カスタム・メジャーの作成」](#)で作成したカスタム・メジャーを表示する列を追加します。

作業としては、まずレポートのレイアウトに新規列を追加し、その列にフィールド・オブジェクトを挿入します。

**ヒント：** 新規フィールド・オブジェクトの頻度は、F\_PROJECTED\_SALES および F\_SALES と同じである必要があります。フィールド・オブジェクトの頻度が異なると、レポートを実行できません。

1. ペーパー・デザイン・ビューで、「ペーパー・レイアウト」アイコンをクリックしてペーパー・レイアウト・ビューを表示します。ペーパー・レイアウト・ビューとオブジェクト・ナビゲータが横に並ぶようにしてください。
2. オブジェクト・ナビゲータで、[Ctrl] キーを押しながら「M\_G\_PRODUCT\_GRPFR」と「R\_G\_PRODUCT」をクリックします。

**ヒント：** M\_G\_PRODUCT\_GRPFR は、基準となるマスター・グループです。この時点では、R\_G\_PRODUCT の下に隠れていて見えません。そのため、ペーパー・レイアウト・ビューでは選択されているグループが1つのみに見えますが、実際には両方の枠が選択されています。

3. ペーパー・レイアウト・ビューで、選択された枠の幅を約 43/4 インチに拡張します。

---

**注意：** オブジェクトのサイズ変更や移動ができないときは、「フレックス・オン」ボタンをクリックしてフレックス・モードをオンにするか、「フレックス・オフ」ボタンをクリックしてフレックス・モードをオフにします。

---



4. オブジェクト・ナビゲータで、「M\_G\_CROSS\_GRPFR」をクリックします。
5. ペーパー・レイアウト・ビューで、選択された枠の幅を約 4 3/4 インチに拡張します。
6. 「F\_GEOG\_AREA」オブジェクトを選択し、次に「親枠を選択」をクリックして親枠 R\_G\_GEOG\_AREA を選択します。選択された枠の幅を約 4 3/4 インチに拡張します。
7. 「F\_CHANNEL」を選択し、「親枠を選択」ボタンをクリックして親枠 R\_G\_CHANNEL を選択します。
8. 選択された枠の幅を約 4 3/4 インチに拡張します。
9. もう一度「F\_CHANNEL」を選択して、このオブジェクトの幅を約 4 3/4 インチに拡張します。
10. 「フィールド」ボタンをクリックします。
11. ボックスをクリックし、そのまま F\_SALES オブジェクトの右にドラッグします。
12. 「ツール」⇒「プロパティ・インスペクタ」を選択します。
13. 次のプロパティを設定します。

ノード	プロパティ	値
一般情報	名前	F_Increase
フィールド	ソース	INCREASE

14. 「ペーパー・レイアウトを実行」アイコンをクリックしてレポートを実行します。F\_Increase が INCREASE を参照する頻度が所属グループよりも低いことを示すエラーが表示されます。このためレポートは実行できません。

エラーの原因を確認するため、オブジェクト・ナビゲータで F\_INCREASE を検索してください。F\_INCREASE が、R\_G\_PROJECTED\_SALES よりも上位レベル（低い頻度）に配置されているはずですが、INCREASE は、売上実績が売上予測を上回ったパーセントを計算する列です。このレポートを実行するには、F\_INCREASE を F\_PROJECTED\_SALES および F\_SALES と同じ頻度にして、値の計算に必要なデータを参照できるようにする必要があります。

15. 「OK」をクリックしてエラー・メッセージを閉じます。
16. 「ペーパー・レイアウト」アイコンをクリックしてペーパー・レイアウト・ビューを表示します。
17. フィールド「F\_INCREASE」を選択し、削除します。
18. 「F\_SALES」を選択し、次に「親枠を選択」ボタンをクリックして親枠 R\_G\_PROJECTED\_SALES を選択します。
19. 選択された枠の幅を約 4 3/4 インチに拡張します。
20. ステップ 10 ～ 13 を繰り返して、フィールド・オブジェクトを作成します。

21. F\_PROJECTED\_SALES および F\_SALES が同じ頻度であることを確認するため、F\_Increase オブジェクトを選択したままオブジェクト・ナビゲータで「F\_INCREASE」を検索します。
22. 「F\_Increase」フィールドの書式を次のように変更します。
  - 「塗りカラー」 ボタンをクリックして、塗りカラーをライト・イエローに変更します。
  - 「テキスト・カラー」 ボタンをクリックして、テキスト・カラーをダーク・ブラウンに変更します。
  - 「線カラー」 ボタンをクリックして、フィールド周囲の境界線をダーク・ブラウンにします。
  - 「太字」 ボタンをクリックして、目立つようにテキストを太字にします。
23. 「テキスト」 ボタンをクリックします。
24. 列タイトルを追加するため、四角形をクリックし、そのまま F\_Increase の上にドラッグします。
25. 四角形の中に、「Increase」と入力します。
26. この列にテキスト・オブジェクトを配置し、左にある「Actual Sales」フィールドと同じ書式に変更します。

---

**注意：**「表示」メニューの「格子指定」をオフに切り替えると、テキスト・オブジェクトをフィールド全体に拡張できるようになります。書式設定を適用するときは、目的のテキスト・オブジェクトが選択されていることを確認してください。選択されていないと書式設定は無効になります。

---

27. 「ペーパー・レイアウトを実行」アイコンをクリックしてレポートを表示します。
28. レポートを xprs\_942.rdf という名前で保存します。

### 6.5.3 ディメンション値のソート

レポートで配布経路のソート順を変更するとします。この例では、「Express クエリー」ダイアログ・ボックスのセレクトアを使用して、経路ディメンションのソート基準を変更します。デフォルト経路階層（最上位から最下位）の順序で表示するのではなく、階層の最下位経路から最上位経路の順にデータを表示します。階層は、「All Channels」が先頭で「Indirect」が最後になるようにデータベースに事前定義されています。

1. データ・モデル・ビューで、問合せオブジェクト「QE\_SALES」をダブルクリックします。
2. 「Express クエリー」ダイアログ・ボックスで「セレクトア」を選択します。

3. 「ディメンション」リストで、「Distribution Channel」を選択して「ソート」ボタンをクリックします。
4. 「セレクションのソート」ダイアログ・ボックスで、次の値を選択します。

基準	選択
基準	階層
順序	下から上へ
階層	Standard

5. 「セレクションのソート」ダイアログ・ボックスで「OK」をクリックします。
6. 「セレクト」ダイアログ・ボックスで「OK」をクリックします。
7. 「Express クエリー」ダイアログ・ボックスで「OK」をクリックします。
8. 「ペーパー・レイアウトを実行」アイコンをクリックしてレポートを表示します。
9. レポートを xprs\_943.rdf という名前で保存します。

#### 6.5.4 ペーパー・デザイン・ビューでの書式変更

1. ペーパー・デザイン・ビューで、[Shift] キーを押しながら「Projected Sales」と「Actual Sales」の下列、および「Projected Sales total」と「Sales total」の両フィールドをクリックします。
2. 「通貨」ボタンをクリックして、書式マスクを通貨に変更します。
3. 「右に整列」ボタンをクリックして、値を右揃えにします。
4. 「10 進桁を追加」を 2 回クリックして、小数点以下 2 桁を追加します。
5. 「Increase」の下列をクリックします。
6. 「パーセント」ボタンをクリックして、書式マスクをパーセンテージに変更します。
7. 「中央に整列」ボタンをクリックして、値を中央揃えにします。

レポートは次の図と同じようになります。



1997 Sales Report

Product: Audio Division			
	Channel	Indirect	
Region	Projected Sales		Actual Sales      Increase
Areas in the Americas	\$1871668.13		\$4277531.00      129%
Australia	\$534098.31		\$1220634.25      129%
Europe	\$2417825.50		\$5525725.00      129%
Asia	\$1068249.13		\$2441388.50      129%
Totals:	\$5891841.06		\$13465278.75

8. レポートを xprs\_944.rdf という名前で保存します。

6.6 まとめ

以上ですべての作業が完了しました。Express のサンプル・レポートが完成しました。この章で学習した内容は次のとおりです。

- レポート・ウィザードを使用した、データ・モデルおよびレイアウトの定義
- ディメンション値の制限による Express クエリーの変更
- データ・モデル・ビューを使用した、レポートへのサマリーおよびカスタム・メジャー列の追加
- ペーパー・レイアウト・ビューを使用したフィールド挿入およびレイアウト再配置
- ペーパー・デザイン・ビューを使用したレポート外観の拡張

---

# 用語集

## HTML (Hypertext Markup Language)

Hypertext Markup Language の略称。Internet 上の Web サーバーの他のドキュメントに対するコンテンツとリンクを指定するのに使用される、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。

## Java (Java)

プラットフォームに依存しないアプレットの形式でインターネット向けのプログラミングをサポートするコンピュータ言語。

## JSP (JavaServer Page)

サーブレット機能に対する拡張で、Web ページとの簡単なプログラム・インタフェースを実現する。JSP は特殊なタグと埋め込み Java コードを持つ HTML ページである。埋め込まれた Java コードは Web またはアプリケーション・サーバーで実行され、HTML ページに動的な機能を提供する。JSP は、最初の要求時にサーバーの JVM で実行されると、サーブレットにコンパイルされる。

## <Oracle\_Home>

ディレクトリベースのオペレーティング・システムにおける、Oracle ディレクトリ階層の最上位ディレクトリの別名。Oracle 製品のルート・ディレクトリを示す環境変数。

## Oracle9i Application Server

Oracle9i Application Server (Oracle9iAS) は、ネットワーク・アプリケーションを展開するための戦略的プラットフォームである。アプリケーション・ロジックをアプリケーション・サーバーに移動し、ネットワーク・クライアントを配置することで、複雑さが緩和され、管理が容易になり、開発と配布が簡素化されるため、組織は大幅な省力化を実現できる。Oracle9iAS のみが、ビジネスに不可欠なプラットフォームを提供する。このプラットフォームによって、データベース Web パブリッシングが容易になるほか、従来のクライアント / サーバーからネットワーク・アプリケーション・アーキテクチャに移行しつつ、過去の資産を完全に統合できる。

## **Oracle9i Developer Suite**

Oracle9i Developer Suite (Oracle9iDS) は、Oracle の優れたアプリケーション開発ツールとビジネス・インテリジェンス・ツールを結合して 1 つの統合製品としたものである。Java や XML などのインターネット標準に基づくこの製品は、Oracle9i Application Server および Oracle9i Database 向けのアプリケーションを作成するための、高い完成度と生産性を持つ開発環境を提供する。

## **Oracle9iAS Portal**

Oracle9iAS Portal は、スケーラブルかつセキュアで拡張可能な HTML アプリケーションおよび Web サイトを作成するための HTML ベースの開発ツールである。Oracle9iAS Reports Services では、Oracle9iAS Portal を使用して、様々な情報を保存することにより、Web 上で発行されたレポートに対するエンド・ユーザーのアクセスを制御する。この保存する情報とは、レポート要求、保護されているサーバー、およびレポート出力の印刷に使用する Oracle9iAS Reports Services のプリンタに関するものである。

## **Oracle9iAS Reports Services**

Reports Developer アプリケーションのランタイム環境。Oracle9iAS Reports Services は、各レポートを全社規模で実行、配布、公開する。Reports Services を使用してレポートを展開することで、柔軟性が増し、時間が節約され、処理能力も高まる。

## **Oracle9iAS Reports Servlet**

Java ベースの Web サーバーと Oracle9iAS Reports Runtime の間のインタフェース。Web ブラウザから動的にレポートを実行できるようにする。

## **Oracle9iDS Reports Builder (rwbuilder)**

レポート定義を作成、開発および管理する。

## **PDF (Portable Document Format)**

ドキュメントを、その作成に使用されたアプリケーション・ソフトウェア、ハードウェアおよびオペレーティング・システムに依存することなく表示できるようにする Adobe Acrobat 専用のファイル形式。PDF ファイルでは、テキスト、グラフィックおよびイメージのあらゆる組合せを含むドキュメントを、デバイスや解像度に依存しない形式で記述できる。

## **PL/SQL**

SQL 言語に対する Oracle 独自の拡張。アプリケーションの作成に適したプロシージャ構造やその他の構造を SQL に追加する。

## **RDF ファイル (RDF file)**

単一のレポート定義をバイナリ形式で保存したファイル。RDF ファイルは、レポートを実行および編集するために使用される。

## **SELECT 文 (SELECT statement)**

1 つ以上の表またはビューからフェッチする行と列を指定する SQL 文。

## **SQL**

リレーショナル・データベースに対して、情報を格納または取り出すための標準インタフェース。SQL は、Structured Query Language の略称。

## **SQL スクリプト (SQL script)**

SQL 文を含むファイル。これを実行すると、データベースの管理をすばやく容易に実施できる。Oracle 製品には SQL スクリプトがいくつか付属している。

## **SQL ファイル (SQL file)**

テキスト形式 (ASCII や EBCDIC など) で保存された問合せを含むファイル。

## **SQL 文 (SQL statement)**

Oracle に対する SQL 命令。SELECT 文は SQL 文の一種である。

## **URL (Uniform Resource Locator)**

URI の一種である URL は、インターネットを通じて利用できるリソースの場所を簡潔な文字列で表したものである。また、クライアントが Oracle9iAS への要求をエンコードするために使用するテキスト文字列の書式でもある。

## **Web サーバー (Web server)**

Web サイトで実行されるサーバー・プロセス (HTTP デモン) で、リモートの Web ブラウザからの HTTP 要求に応じて Web ページを送信する。

## **Web ソース・ビュー (Web source view)**

レポートの HTML/JSP ソースを表示する。このビューでレポート・ブロック・ウィザードおよびグラフ・ウィザードを使用することにより、Web ページに動的なコンテンツを追加できる。熟練した Java 開発者であれば、このビューで直接 Web ソースを編集できる。

## **Web ブラウザ (Web browser)**

エンド・ユーザーが、Web サーバーを通じて提供されるコンピュータ上の HTML ドキュメントおよびプログラムを読み込むために利用するプログラム。

## **XML**

eXtensible Markup Language の略語。SGML を使用してデータの定義と構造化を行うメタ言語である。Reports Builder は、XML の出力をサポートすることで、Web パブリッシングおよびサードパーティ製アプリケーションとの電子データ交換を実現している。XML を使用して、実行時に他のレポート定義とマージできるレポート定義や、個別に実行できるレポート定義を作成することもできる。

### **アイコン (icon)**

ウィンドウやツールの図形表示。

### **イメージ (image)**

アプリケーションに保管およびロードできるビットマップ・オブジェクト。クライアントは、インポートされたイメージを変更できない。

### **イントラネット (intranet)**

内部的な TCP/IP ネットワークであり、これに対するアクセスは（ファイアウォールにより）企業や組織内の人物に制限されている。イントラネットが組織内で提供するサービスはインターネットが提供するものと同様であるが、必ずしもインターネットに接続されているとは限らない。企業内で情報やアプリケーションの配布のため、内部ネットワーク上に 1 つ以上の Web サーバーをセットアップするのはイントラネットの一般的な例である。

### **ウィザード (wizard)**

一般的に実行されるタスクを、順を追って簡単に実行するためのインタフェース。Report Builder でのウィザードは次のとおり。

- レポート・ウィザード: 基本的なペーパー・レポートまたは Web レポートを作成するための手順を示す。ウィザードの各ページでは、最初のレポートを作成するための情報の入力が必要とされる。
- データ・ウィザード: (Data Wizard) 多重問合せデータ・モデルで使用する問合せを簡単に定義または変更できるようにする。
- グラフ・ウィザード: 真の 3 次元グラフなどの様々なチャートやグラフを追加する。Report Builder では、Oracle BI graph Bean と共にインプリメントされている。
- レポート・ブロック・ウィザード: 静的 HTML ページにデータを追加できるようにする。

### **ウィンドウ (window)**

デスクトップ上でアプリケーションを表示する矩形領域。各ウィンドウには、ユーザーがアプリケーションと対話するための領域がある。ウィンドウは、オープン、サイズ変更、移動、アイコンへの縮小、全画面での表示などが可能である。

### **エディタ (editor)**

プログラム単位の作成やアプリケーションの設計など、一連の特定のタスクを実行する作業領域。



## オブジェクト (object)

1. レイアウト上に配置できるアイテム。オブジェクトの例としては、四角形、線、楕円、円弧、ポリゴン、折れ線、丸い四角形、手書き、グラフ、テキスト、記号、テキスト・フィールドなどがある。
2. Oracle8 におけるオブジェクト・タイプのインスタンス。オブジェクト表の行、またはリレーショナル表の列オブジェクトに含まれる行の一部がオブジェクトとなる。

## 構文 (syntax)

コマンド、修飾子およびパラメータを組み合わせて有効なコマンド文字列を形成するための規則。

## スタイル・シート (style sheet)

HTML ドキュメントの書式をきわめて柔軟に設定できるようにする、HTML の拡張書式。スタイル・シートを利用する HTML ドキュメントは、スタイル・シートをサポートするブラウザで表示する必要がある。

## ダイアログ・ボックス (dialog box)

操作を完了するために必要な情報の入力を求める、部分的な画面またはウィンドウ。

## チェックボックス (checkbox)

小さな四角形として表示されるインタフェース要素であり、ユーザーはこれをオンまたはオフに切り替えることができる。

## ツールバー (toolbar)

製品のコマンドを実行するアイコン・ボタンの集合。通常、ウィンドウ上部に水平に表示されるか、ウィンドウの右側または左側に垂直に表示される。

## ツール・パレット (tool palette)

ツールの集合。

## データ・ソース (data source)

問合せから返されるデータのソース。表、ビュー、シノニム、スナップショット、ビューとして格納された問合せなどのデータベース・オブジェクトを含む。

## データベース (database)

1. ユニットとして扱われる、1 組のディクショナリ表とユーザー表。
2. (Oracle Express) データを編成、格納および操作するオブジェクトを含んでいる単一のファイル (通常は拡張ファイルを伴う)。Express には、このようなオブジェクトの例として、変数、ディメンション、式、モデル、プログラムなどがある。

## データ・モデル (data model)

データベースからどの値をフェッチすべきか、どの値を計算すべきか、データをレポート内でどのように順序付けるべきかを定義するリレーショナル・モデル。データ・モデルを定義する Report Builder オブジェクトは、問合せおよびグループ、列、パラメータ、リンクである。

## データ・モデル・ビュー (Data Model view)

レポート内のデータを構造的に表示する。レポート出力にオブジェクトは表示されないが、構造によってレイアウト・スタイルが決まり、データ・オブジェクトによってレイアウト・オブジェクトに表示される値が提供される。

## テンプレート (template)

共通のスタイルと標準を含むフレームワークを定義したもの。グラフィックを含む場合もある。テンプレートによって提供される標準書式を基に、専門的な水準のルック・アンド・フィールを備えたレポートを迅速に簡単に開発できる。

## 問合せ (query)

データベースの 1 つ以上の表またはビューから取り出すデータを指定する SQL SELECT 文。

## 表 (tabel)

関連する情報の集合を行と列から成る 2 次元のグリッド形式にして名前を付けたもの。リレーショナル・データベースやサーバーに格納される。

## 表形式 (tabular)

ページの最上部にラベルを表示し、その下にデータ列を表示したデフォルト・レイアウト。

## フィールド (field)

1. その中でデータの入力、編集および削除が可能なインタフェース要素。2. 特定の問合せ列のデータ表示方法を定義するレイアウト・オブジェクト。

## ペーパー・デザイン・ビュー (Paper Design view)

ペーパー・レポートの出力を表示し、レイアウトに対する間隔設定、フィールドの書式設定、カラー設定、テキスト編集などの一般的で簡単な変更の多くを、ペーパー・レイアウト・ビューを開くことなく実行できるようにする。

## ペーパー・レイアウト・ビュー (Paper Layout view)

ペーパー・レポート内のレイアウト・オブジェクトを表示し、あらゆるレイアウト・オブジェクトに対して多くの変更を実行できるようにする。すべてのレイアウト・オブジェクトには、プロパティ・インスペクタで変更可能なプロパティが設定されている。レイアウト・オブジェクトの階層は、データ・モデルによって決まる。

## **レイアウト (layout)**

オブジェクトの作成、変更、配置または削除を行うエディタの領域。

## **列 (column)**

1. データの特定のドメインを表す、データベース表内の垂直方向の領域。列には列名 (ENAME など) があり、特定のデータ型 (CHAR など) を持っている。たとえば、従業員情報の表で従業員名のすべてが 1 つの列を構成するなど。レコード・グループ列は、データベースの列を意味する。
2. 問合せの SELECT リスト内の列のそれぞれの式に対し自動的に作成されるか、集計や式を実行したりプレースホルダーとして稼働するために手動で作成されるデータ・モデル・オブジェクト。



---

---

# 索引

## B

---

Before Report トリガー、作成 , 1-6

## H

---

HTML ファイル、問合せの作成 , 1-14

## J

---

Java Importer、使用 , 1-5

JavaBean、Web レポートでの使用 , 1-13

JavaBean、インポート , 1-5

JavaBean、ペーパー・レポートでの使用 , 1-4

Java クラス、インポート , 1-5

## R

---

REF カーソル

問合せ、作成 , 3-6

問合せ、リンクの作成 , 3-11

入力、定義 , 3-4

REPORTS\_CLASSPATH、更新 , 1-4

## S

---

SELECT 文、パッケージへの移動 , 3-17

SQL、SQL と XML による問合せの作成 , 4-3

SQL、問合せの手動作成 , 4-4

SQL 問合せ、手動作成 , 4-4

## T

---

textpds.conf、編集 , 5-3

## W

---

Web レポート、実行 , 1-22, 2-6

Web レポート、バーコード JavaBean の使用 , 1-13

## X

---

XML、SQL と XML による問合せの作成 , 4-3

XML データ・ソース、XML 問合せの作成 , 4-5

XML データ・ソース、について , 4-1

XML 問合せ、作成 , 4-5

## あ

---

一括処理 , 2-3

## か

---

行カラー、行カラーの交互適用 , 4-10

交互の行カラー、適用 , 4-10

## さ

---

作成

REF カーソル問合せ , 3-6

REF カーソル問合せ間のリンク , 3-11

パッケージ本体 , 3-18

ブレイク・グループ , 3-9

レイアウト , 3-15

サマリー

列、追加 , 3-13

式列、作成 , 1-7, 1-16

式列、データ・モデルへの追加 , 1-7

書式、行カラーの交互適用 , 4-10

使用

レポート・ウィザード, 6-3

## た

---

追加

サマリー列, 3-13

定義

REF カーソル型, 3-4

パッケージ仕様部, 3-4

テキスト、textpds.conf ファイルの構成, 5-3

テキスト、テキスト・データ・ソースを使用したレポートの作成, 5-4

テキスト、データ・ソースとして使用, 5-1

テキスト・データ・ソース、構成, 5-3

テキスト・データ・ソース、使用, 5-1

テキスト・データ・ソース、レポートの作成, 5-4

テキスト・トランスポートابل・データ・ソース、構成, 5-3

テキスト・トランスポートابل・データ・ソース、使用, 5-1

テキスト・トランスポートابل・データ・ソース、レポートの作成, 5-4

データ、フィルタリング, 4-13

データ・ソース、XML について, 4-1

データ・リンク、作成, 4-7

問合せ、SQL 問合せの手動作成, 4-4

問合せ、XML と SQL の使用, 4-3

問合せ、XML 問合せの作成, 4-5

問合せ、作成, 1-14

トランスポートابل・データ・ソース、XML の使用, 4-1

トランスポートابل・データ・ソース、テキストの使用, 5-1

## は

---

配布 XML 定義、編集, 2-5

配布、XML 定義の編集, 2-5

配布、使用, 2-1

バーコード

バーコード JavaBean、について, 1-1

バーコード JavaBean、初期化, 1-19

バーコード、Web レポートでの使用, 1-13

バーコード、ペーパー・レポートでの使用, 1-4

パッケージ

SELECT 文の移動, 3-17

仕様、定義, 3-4

本体、作成, 3-18

ライブラリへの移動, 3-20

パッケージ、作成, 1-5

複数の宛先、レポートの配布, 2-1

ブレイク・グループ、作成, 3-9

プログラム単位エディタ、使用, 3-7

ペーパー・レイアウト、作成, 1-12, 4-9

ペーパー・レポート、バーコード JavaBean の使用, 1-4

ペーパー・レポート、レイアウトの作成, 4-9

## ら

---

ライブラリへのパッケージの移動, 3-20

リンク、REF カーソル問合せ間の作成, 3-11

レイアウト、紙出力用に作成, 1-12, 4-9

レイアウト、作成, 3-15

列、サマリー, 3-13

レポート

Express, 6-1

レポート、Web に対する実行, 1-22, 2-6

レポート、一括処理, 2-3

レポート、手動作成, 4-3

レポート、複数の宛先への配布, 2-1

レポート・ウィザード

使用, 6-3