

Developer/2000生成ガイド

リリース1

部品番号 : A37360-1

ORACLE®

Developer/2000生成ガイド リリース1

部品番号 : A37360-1

原本名 : A Guide to Developer/2000 Generation, 1.0

原本部品番号 : A34875-1

原本著者: Amy Coppola, Andy Page

Copyright © Oracle Corporation 1992, 1995

All rights reserved.

Printed in Japan.

制限付権利の説明

プログラムの使用、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。

本書の情報は、予告なしに変更されることがあります。本書に問題を見つけたら、当社にコメントをお送りください。オラクル社は、本書の無謬性を保証しません。

危険な用途への使用について

当社製品は、原子力、航空産業、大量輸送、又は医療の分野など、本質的に危険が伴うアプリケーションを用途として特に開発されておられません。当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いただきたく、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、当社および開発元である米国Oracle Corporation（その関連会社も含みます。）は一切責任を負いかねます。

ORACLEは、Oracle Corporationの登録商標です。

本文中の他社の商品名は、それぞれ各社の商標または登録商標です。



はじめに

このガイドを読む前に、以下の点を理解しておいてください。

- 本書の目的
- 本書の対象読者
- 本書を使用するための前提条件
- 本書の構成
- 本書の使い方
- 本書で使用する表記法
- 関連するマニュアル
- このマニュアルに対するご意見

本書の目的

このガイドでは、Designer/2000のジェネレータを紹介しています。このガイドを読むと、Designer/2000のリポジトリにある情報から独自のデータベース、フォーム、レポートを作成する方法がわかります。



詳細情報：このガイドはServer, Forms, Reportsの各ジェネレータの機能を細かく説明するものではありません。ジェネレータに関する総合的な情報はオンライン・マニュアルを参照してください。

本書の対象読者

このガイドは、Designer/2000のリポジトリに格納されている情報からDesigner/2000アプリケーションのサーバー側とクライアント側の構成要素を作成しなければならないアプリケーション開発者のためにかかれています。

本書を使用するための前提条件

このガイドでは、次のように仮定しています。

- 読者は使用するパーソナル コンピュータ (PC) とキーボードの操作について精通している。
- 読者はMicrosoft Windowsの環境について精通している。
- Designer/2000のソフトウェアがPCに正しくインストールされている。

本書の構成

このガイドは、4つの部分から構成されています。

- 第I部 「はじめに」では、Designer/2000製品と、サーバー、フォーム、レポートのジェネレータを紹介しています。
- 第II部 「サーバーの生成」では、Serverジェネレータを構成するユーティリティを紹介し、これらを使用してDDLを作成する方法、リポジトリとOracleデータ・ディクショナリを調整する方法、データベース・オブジェクトの定義をリポジトリにリバースする方法について説明しています。
- 第III部 「フォームの生成」では、作業環境とテンプレート・フォームの使用、メニューの生成、再生成、リバース・エンジニアリングなど、フォーム生成の主要な側面を紹介しています。
- 第IV部 「レポートの生成」では、作業環境とテンプレート・レポートの使用やリバース・エンジニアリングなど、レポート生成の主要な側面を紹介しています。

本書の使い方

Designer/2000ツールセットやツール間について詳しくない場合は、第1部「はじめに」を参照してください。以後の各部では、はじめに生成に関する概念を説明しています。これらの概念をすでに理解している場合は、各部の残りの部分に進んでください。これらの章では、生成作業を詳しく説明しており、数多くの作業例が含まれています。

本書で使用する表記法

メニュー選択は、メニューの名前とオプションの名前を分割する矢印で表します。例えば、「ファイル」メニューの「新規作成」オプションを表す場合は、次のようになります。

「ファイル→新規作成」を選択します。

ユーザーが入力する文字は、クーリエで表します。例えば、次のようになります。

REN0045と入力します。

関連するマニュアル

Oracle Designer/2000のオンライン・マニュアルをDesigner/2000のCD-ROMからインストールできます。

以下のマニュアルでは、Designer/2000製品に関する情報を印刷された形式で利用できます。

- 『Oracle Designer/2000 for Windowsインストレーション・ガイド』
部品番号：A15847
- 『Oracle Designer/2000 製品概要』
部品番号：A15849
- 『Oracle Designer/2000 ドキュメンテーション・セット』
部品番号：A37361-1
 - 『プロセス・モデリング・ガイド』
 - 『システム・モデリング・ガイド』
 - 『システム・デザイン・ガイド』
 - 『Developer/2000生成ガイド』
 - 『リポジトリ管理ガイド』

このマニュアルに対するご意見

弊社では、このマニュアルに関するご意見やご感想をお待ちしております。マニュアル末尾の「マニュアル・コメント用紙」をご使用になり、弊社マニュアル係までお寄せください。

なお、この用紙はマニュアルの記述に対するコメント用紙ですので、製品の操作における技術的な質問は、ユーザー登録を済ませていることをご確認の上で、弊社技術サポートまでお願いいたします。

目次

第 I 部	はじめに	
第 1 章	Designer/2000ジェネレータの紹介	1 - 1
	Designer/2000とは何か	1 - 2
	Designer/2000の構成要素	1 - 2
	Designer/2000ジェネレータとは何か	1 - 5
	Serverジェネレータ	1 - 5
	Formsジェネレータ	1 - 5
	Reportsジェネレータ	1 - 5
	ジェネレータへのアクセス	1 - 6
	ジェネレータを使用するための条件	1 - 7
第 2 章	生成のソース	2 - 1
	生成のソースとは何か	2 - 2
	リポジトリとは何か	2 - 3
	データベース・オブジェクト定義とは何か	2 - 4
	モジュール定義とは何か	2 - 5
	モジュール・データ取扱とは何か	2 - 5
	モジュール・ネットワークとは何か	2 - 6
	アプリケーション・システム	2 - 6
	作業環境とは何か	2 - 7
	作業環境レベル	2 - 7
	作業環境を使用するための指針	2 - 7
	テンプレートとは何か	2 - 8
	テンプレート・フォームとは何か	2 - 8
	テンプレート・レポートとは何か	2 - 8

第 II 部

サーバの生成

第 3 章

Serverジェネレータを使用するオブジェクト定義の作成と管理 3 - 1

Serverジェネレータの紹介 3 - 2

 Serverジェネレータとは何か 3 - 2

Oracle Serverジェネレータへのアクセス 3 - 3

DDLコマンドの生成 3 - 4

 DDLを生成するための前提条件 3 - 4

 使用するツール 3 - 4

 DDLを生成する前に 3 - 4

 DDLコマンドを作成するには 3 - 5

 何が生成されるか 3 - 9

 コマンドファイル名 3 - 10

データベース オブジェクトのリバース エンジニアリング 3 - 11

 データベース オブジェクトのリバース エンジニアリング
 の前提条件 3 - 11

 使用するツール 3 - 11

 データベース オブジェクトのリバース エンジニアリングの2つの
 ステージ 3 - 11

 データベース オブジェクトのリバース エンジニアリングを
 行うには 3 - 12

 何がリバース エンジニアされるか 3 - 14

リポジトリ オブジェクトの調整 3 - 15

 調整ユーティリティを使用するための前提条件 3 - 15

 使用するツール 3 - 15

 調整の処理 3 - 15

 リポジトリ オブジェクトを調整させるには 3 - 16

 何が調整されるか 3 - 19

第 III 部

フォームの作成

第 4 章

Oracle Forms ジェネレータの紹介 4 - 1

Oracle Formsとは何か 4 - 2

 フォーム アプリケーションの作成 4 - 2

Formsジェネレータとは何か 4 - 3

フォーム作成処理の概要 4 - 4

 モジュールの定義 4 - 4

 表および列取扱の定義 4 - 4

 作業環境の設定 4 - 4

 テンプレートの使用 4 - 4

 フォームの作成 4 - 4

	フォームの精製	4 - 5
	再生成	4 - 5
	リバース エンジニアリング	4 - 5
	Designer/2000とDeveloper/2000の用語	4 - 8
	作成したフォームの例	4 - 9
	Oracle仲買サービス デモ アプリケーションから組み合わせた フォーム	4 - 9
第 5 章	単一ブロックフォームの作成	5 - 1
	デフォルト設定を使用するフォームの作成	5 - 2
	モジュールの作成	5 - 2
	データ取扱細目の定義	5 - 3
	フォームの生成	5 - 6
	生成したフォームの機能	5 - 10
	① MDIウィンドウ	5 - 10
	モジュール・プロパティから生成されるプロパティ	5 - 11
	② ウィンドウ・タイトル	5 - 11
	表取扱細目および列取扱細目から生成されるプロパティ	5 - 11
	③ ブロック・タイトル	5 - 11
	④ 項目プロンプト	5 - 11
	⑤ 項目のサイズ	5 - 11
	⑥ メッセージ行	5 - 11
	生成したフォームの他のプロパティ	5 - 12
	モジュール定義、作業環境、テンプレートの精製とカスタマイズ	5 - 13
第 6 章	フォーム モジュール定義の精製	6 - 1
	フォーム モジュール定義の精製	6 - 2
	ウィンドウ プロパティの変更	6 - 4
	ウィンドウ タイトルの変更	6 - 4
	ウィンドウのサイズの変更	6 - 5
	ウィンドウの位置の変更	6 - 6
	生成したフォームの機能	6 - 6
	フォーム上のブロックのカスタマイズ	6 - 7
	ブロックのタイトルの変更	6 - 7
	表取扱フラグの設定	6 - 8
	フォーム上の項目のカスタマイズ	6 - 10
	列取扱の順序変更	6 - 10
	フィールド プロンプトの変更	6 - 11
	ポップリストの生成	6 - 11
	項目グループの生成	6 - 12
	列取扱フラグの設定	6 - 13
	生成したフォームの機能	6 - 16
	① ブロック タイトル	6 - 16
	② 項目の順序	6 - 16
	③ 項目プロンプト	6 - 16

④ 項目グループ	6 - 17
⑤ ポップリスト	6 - 17
複数行表示ブロックの定義	6 - 18
ブロック オーバーフロー スタイルの指定	6 - 19
行折り返しオーバーフロー スタイル	6 - 20
右にオーバーフローオーバーフロー スタイル	6 - 20
下にオーバーフローオーバーフロー スタイル	6 - 21
スプレッドテーブルオーバーフロー スタイル	6 - 21

第 7 章

作業環境を使用しての生成フォームのカスタマイズ	7 - 1
作業環境の紹介	7 - 2
作業環境とは何か?	7 - 2
作業環境の使用に関するガイドライン	7 - 2
作業環境の設定を変更する方法	7 - 3
プリファレンス・ナビゲータの起動	7 - 3
ジェネレータ作業環境のカテゴリ	7 - 4
CODING STYLE作業環境	7 - 4
COMMENTING作業環境	7 - 4
DBA作業環境	7 - 4
DESCRIPTIVE FLEX作業環境	7 - 4
END USER INTERFACE作業環境	7 - 4
ENVIRONMENT作業環境	7 - 4
FORM AND MENU ATTACHMENT作業環境	7 - 4
GENERATE OPTIONS作業環境	7 - 4
GENERATOR RUNTIME作業環境	7 - 4
JOURNALLING作業環境	7 - 4
LAYOUT作業環境	7 - 4
LAYOUT作業環境 - <Various>	7 - 5
LIST OF VALUES作業環境	7 - 5
MENU作業環境 - <Various>	7 - 5
REGENERATION作業環境	7 - 5
REVERSE ENGINEERING作業環境	7 - 5
TEMPLATE FORMS作業環境	7 - 5
レイアウト作業環境の設定と変更	7 - 6
モジュール・レベルによる項目グループ内の方向の変更	7 - 6
項目グループ・レベルにおける項目グループ・ブラシ幅の変更 ..	7 - 8
ブロック・デコレーションとタイトル文字位置の変更	7 - 10
生成されたフォームの機能	7 - 12
① ブロック・タイトルの文字位置	7 - 12
② 項目グループ内の方向	7 - 12
③ 項目グループ枠	7 - 12
④ ブロック枠	7 - 12

第 8 章

テンプレートを_usingしての生成フォームのカスタマイズ	8 - 1
テンプレート・フォームの紹介	8 - 2

テンプレート・フォームとは何か？	8-2
なぜテンプレートを使用するか？	8-3
使用するテンプレートの指定	8-4
STFFMB作業環境の変更	8-4
テンプレート・フォームの位置の指定	8-4
テンプレートの作成	8-5
標準テンプレートのコピー	8-5
背景としてのボイラープレートの作成	8-10
生成されたフォームの機能	8-13
ジェネレータ・オブジェクトの作成	8-14
生成されたフォームの機能	8-16
ユーザー・オブジェクトの作成	8-17
生成されたフォームの機能	8-19
テンプレートフォーム内のプロシージャを呼び出すための ボタン項目の生成	8-20
生成されたフォームの機能	8-23

第9章

複数の表取扱を持つフォームの生成	9-1
複数の表取扱を持つフォームの紹介	9-2
参照取扱の作成	9-3
参照表用の表取扱とは何か？	9-3
参照表取扱を作成するのはなぜか？	9-3
デフォルト結合生成ユーティリティの実行	9-4
参照取扱からのLOVの作成	9-6
生成されたフォームの機能	9-6
マスター・ディテール・フォームの生成	9-8
マスター・ディテール・フォームとは何か？	9-8
マスター・ディテール・フォームの生成	9-8
生成されたフォームの機能	9-10
配置の指定	9-11
別ウィンドウへのディテール・ブロックの配置	9-11
生成されたフォームの機能	9-12
各種配置オプション	9-14
1番目のブロック: 位置 = 常に新しいページ/新しいウィンドウ	9-14
2番目のブロック: 位置 = 同じウィンドウ	9-14
2番目のブロック: 位置 = 新しいページ/新しいウィンドウ	9-15
2番目のブロック: 位置 = 新しいページ	9-16
2番目のブロック: 位置 = 新しいポップアップ	9-17
3番目のブロック: 位置 = 同じポップアップ	9-17
ブロックの同期化の使用	9-18
ブロックの同期化とは何か？	9-18
各種のブロックの同期モード	9-18

第10章

モジュール間のナビゲート	10-1
ナビゲーション生成の概要	10-2

既存のモジュール結合と構造の変更	10 - 2
モジュール・ネットワークとFormsジェネレータ	10 - 2
メニュー・セキュリティとFormsジェネレータ	10 - 4
メニュー生成の前提条件	10 - 4
メニューの生成	10 - 5
生成されたメニューの機能	10 - 8
サブメニューの生成	10 - 9
生成されたメニューの機能	10 - 11
ナビゲート可能モジュールのポップリストの作成	10 - 12
生成されたメニューの機能	10 - 14
ナビゲーション・ボタンの生成	10 - 15
テンプレート・メニューを使ったメニューの生成	10 - 16
生成されたメニューの機能	10 - 18

第 11 章

再生成	11 - 1
再生成の紹介	11 - 2
再生成とは何か?	11 - 2
なぜ再生成が必要なのか?	11 - 2
再生成の必要性を減らす	11 - 2
フォームを再生成する方法	11 - 3
再生成時のリポジトリの更新	11 - 4
再生成時の新しいブロックと項目の追加	11 - 5
新しいブロックと項目の作成	11 - 5
新しいブロックのための新しいページ、ポップアップ、 ウィンドウの作成	11 - 5
再生成時の既存のブロックと項目の属性の変更	11 - 7
既存のブロックの属性の変更	11 - 7
既存の項目の属性の変更	11 - 7
コメントの変更	11 - 7
変更履歴、ジャーナルおよび列導出要件の変更	11 - 8
再生成とテンプレート	11 - 9
テンプレートから作成される既存の項目	11 - 9
既存のキャンバスとウィンドウ	11 - 9
新しく生成されたキャンバス	11 - 9
新しく生成されたウィンドウ	11 - 9
テンプレート・フォーム内の新しいユーザー・キャンバス	11 - 9
ユーザー・オブジェクト	11 - 9
ツールバー	11 - 9
再生成時の生成されたコードの更新	11 - 10
生成されたコードの再生成	11 - 10
非生成コード	11 - 10
再生成時のコードの保持	11 - 10
保持されたコードの表示	11 - 11
再生成時の新しいメニューの生成	11 - 13

第 12 章

フォームのリバース・エンジニアリング	12 - 1
リバース・エンジニアリングの紹介	12 - 2

リバース・エンジニアリングとは何か？	12 - 2
なぜリバース・エンジニアリングを使用するのか？	12 - 2
リバース・エンジニアリングの前に	12 - 3
フォームをリバース・エンジニアリングする方法	12 - 4
リバース・エンジニアリングしたモジュールの機能	12 - 6
リバース・エンジニアリング・プロセスで何が記録されるか？	12 - 7
リバース・エンジニアリングされるモジュール定義	12 - 7
リバース・エンジニアリングされる表取扱細目	12 - 8
リバース・エンジニアリングされる列取扱細目	12 - 10
リバース・エンジニアリングとリエンジニアリング	12 - 11
リエンジニアリングとは何で、なぜ行うのか？	12 - 11
リバース・エンジニアリング後の最初の生成	12 - 11
機能の保持	12 - 12

第 IV 部

レポートの生成

第 13 章

Oracle Reports ジェネレータの紹介	13 - 1
Oracle Reports ジェネレータとは何か？	13 - 2
Reports ジェネレータへのアクセス	13 - 2
生成されるレポート モジュールとは何か？	13 - 2
Oracle Reportsv2.5モジュール	13 - 3
SQL*Plusモジュール	13 - 4
レポート モジュールの生成方法は？	13 - 5
既存のレポートからのレポート モジュール定義の作成	13 - 5

第 14 章

レポート・モジュール定義の作成と生成	14 - 1
レポート モジュール定義の作成	14 - 2
ダイアグラムとモジュール定義の作成	14 - 2
表取扱細目の定義	14 - 4
列取扱細目の定義	14 - 5
レポート・モジュールの生成	14 - 7
生成されるレポートの構成要素 (Oracle Reports v2.5)	14 - 12
生成されるレポートの構成要素 (SQL*Plus)	14 - 16
モジュール定義の編集	14 - 18
モジュール定義のフォーマットの変更	14 - 18
パラメータ定義の作成	14 - 19
モジュール定義の削除	14 - 21
デフォルト サマリーの生成	14 - 22
参照表取扱のインクルード	14 - 23
参照自動生成	14 - 23
生成されるレポートの構成要素	14 - 26
手動による検索の作成	14 - 27
マスター・ディテール・レポート・モジュールの生成	14 - 29

生成されるレポートの構成要素	14 - 31
項目グループの作成、編集、削除	14 - 33
項目グループの編集	14 - 34
項目グループからの列取扱細目の削除	14 - 34
生成されたレポートの構成要素	14 - 35
ツリーウォーク結合の作成	14 - 36
ツリーウォーク結合の作成規則	14 - 36
ツリーウォーク結合の定義	14 - 37
生成されるレポートの構成要素	14 - 38
マトリックス・レポートの生成	14 - 40
生成されるレポートの構成要素	14 - 41
ドリルダウン詳細レポート・モジュールの生成	14 - 45
単独のモジュール定義を使用したドリルダウン レポートの生成	14 - 45
複数のモジュール定義の使用	14 - 47
関連情報のやりとり	14 - 47
モジュール定義の作成	14 - 48
モジュール・ネットワークの作成	14 - 50
レポート モジュール定義の生成	14 - 51

第 15 章

レポート・モジュールのデータ取扱の洗練	15 - 1
レポート・モジュールのデータ取扱細目	15 - 2
表の詳細	15 - 3
表テキスト	15 - 4
列細目	15 - 5
列表示	15 - 6
列テキスト	15 - 7
ブレイク・グループの作成	15 - 8
パラメータと列取扱細目との関連付け	15 - 9
生成されるパラメータ フォームの構成要素	15 - 10
サマリーの明示的な作成	15 - 11
ボイラープレート・テキストまたはグラフィックの追加	15 - 13
イメージまたはOLEコンテナのインポート	15 - 13
マルチメディア・ボタンの追加	15 - 14
ユーザー テキストの追加	15 - 15
オーバーフロー ページに対する列の追加と削除	15 - 16
列の手動選択	15 - 16
列の自動選択	15 - 17
レポート内のグループのカスタマイズ	15 - 20
別名の追加または変更	15 - 20
グループ タイトルの変更	15 - 21
ブレイク グループ タイトルの作成	15 - 22
レイアウト スタイルの変更	15 - 22
表示されるレコード数の制限	15 - 23
参照のための外部結合の使用	15 - 23
戻される行の制限	15 - 24

レポートのフィールドのカスタマイズ	15 - 25
エイリアスの追加または変更	15 - 25
表示と非表示	15 - 26
順序の制御	15 - 26
フィールド プロンプトの追加または変更	15 - 26
整列方法の変更	15 - 28
表示データ型の変更	15 - 28
DATEまたはMONEY書式 マスクの適用	15 - 29
フィールドの折り返しと切り捨て	15 - 29
フィールドの幅または高さの変更	15 - 30
項目グループのカスタマイズ	15 - 31
導出表現の作成	15 - 32
列テキストを使用したレポート トリガーの呼び出し	15 - 33

第 16 章

テンプレートと作業環境の使用	16 - 1
作業環境の初期値の変更	16 - 2
プリファレンス・ナビゲータの起動	16 - 2
作業環境のカテゴリ	16 - 3
作業環境レベルと設定の操作	16 - 5
使用する作業環境の指定	16 - 7
テンプレート・レポート	16 - 9
テンプレート レポートの適用	16 - 9
レイアウト モデル テンプレート レポート	16 - 9
パラメータ フォーム テンプレート レポート	16 - 10
ドリルダウン詳細テンプレート レポート	16 - 11
テンプレート レポートの作成	16 - 12
テンプレート レポートを使用したデータ・モデルのカスタマイズ	16 - 13
サイズと位置の設定	16 - 13
テンプレート・レポート内でのパラメータの定義	16 - 14
テンプレート・レポート内でのレポート・トリガーの実装	16 - 16
テンプレート レポートへのPL/SQLの追加	16 - 18
テンプレート・レポートへのフォーミュラ列の追加	16 - 21
テンプレート レポートを使用したレイアウトのカスタマイズ	16 - 22
パラメータ フォームのカスタマイズ	16 - 22
標準パラメータ・フィールドとプロンプトの作成	16 - 24
ジェネレータ・オブジェクトの作成	16 - 25
ユーザー オブジェクトの作成	16 - 26
レポートまたはパラメータ・フォームのページのカスタマイズ	16 - 28
フォントまたは可視属性の適用	16 - 28
文字単位を定義するフォントの変更	16 - 29

第 17 章

レポートのリバース・エンジニアリング	17 - 1
リバース・エンジニアリングの紹介	17 - 2
リバース・エンジニアリングとは何か?	17 - 2

なぜリバース・エンジニアリングを使用するのか?	17 - 2
サポートされるレポートの種類	17 - 2
リバース・エンジニアリングを行う前に	17 - 2
レポートのリバース・エンジニアリング方法	17 - 3
リバース・エンジニアリングオプションの変更	17 - 5
Oracle ReportsとSQL*ReportWriterのレポート	17 - 7
リバース・エンジニアリング・プロセス中に記録される情報	17 - 7
SQL*Plus、SQL*Report (RPT)、およびPro*Cのレポート	17 - 12
新しいモジュールと既存のモジュールの整理統合	17 - 13

第 I 部

はじめに

第1章

Designer/2000ジェネレータの 紹介

この章では、Designer/2000ジェネレータを紹介します。以下の項が含まれています。

- Designer/2000とは何か
- Designer/2000ジェネレータとは何か
- ジェネレータへのアクセス
- ジェネレータを使用するための要件

Designer/2000とは何か

Oracle Designer/2000は、業務要件を分析し、この要件に適合するクライアント/サーバーシステムを設計するための、一連のソフトウェアツールセットです。Designer/2000では、業務プロセスのモデリング、システム分析、ソフトウェア設計、コード生成がサポートされています。

Designer/2000は、マルチユーザーリポジトリを提供しており、オラクル社のクライアント/サーバー開発ツールセットであるDeveloper/2000と緊密に統合されています。このため、Designer/2000を使用すると、変化する業務要件に適合するスケラブルなクライアント/サーバーシステムを設計して素早く提供することができます。

Designer/2000の構成要素

Designer/2000は、様々なユーザーの主要な要求を反映するように、ツールセットにグループ化されています。

- **Process Modeller** (プロセス・モデラ)を使用すると、目標を達成するために使用した手法を再検討できるように、業務プロセスをモデル化することができます。業務プロセスを図で表現してその詳細を記録するための開発環境が提供されます。
- **Systems Modeller** (システム・モデラ)は、業務要件の詳細な説明を記録します。エンティティ、機能、システム内のデータの流れを図で表したモデルを作成するための開発環境が提供されます。
- **Systems Designer** (システム・デザイナ)は、業務要件に適合するシステムの設計を記録します。システムエンジニアと設計者のための開発環境が提供されます。
- **ジェネレータ製品** (Formsジェネレータ、Reportsジェネレータ、Serverジェネレータ)は、リポジトリに記録されている定義から、サーバー側の構成要素とクライアント側のアプリケーション (フォーム、レポート、メニュー) で構成される完全なシステムを作成します。
- **Repository Administrator** (リポジトリ・アドミニストレータ)は一連のリポジトリ管理プログラムで、リポジトリ・アドミニストレーション・ユーティリティ (リポジトリのインストール、アップグレード、保守用)、リポジトリ・オブジェクト・ナビゲータ (リポジトリを定義するためのグラフィカルブラウザ)、マトリックス・ダイアグラム (リポジトリ内のオブジェクトの関係を示す2次元の格子を作成)、リポジトリ・レポート (リポジトリの内容に関する一連の定義済みレポート) があります。

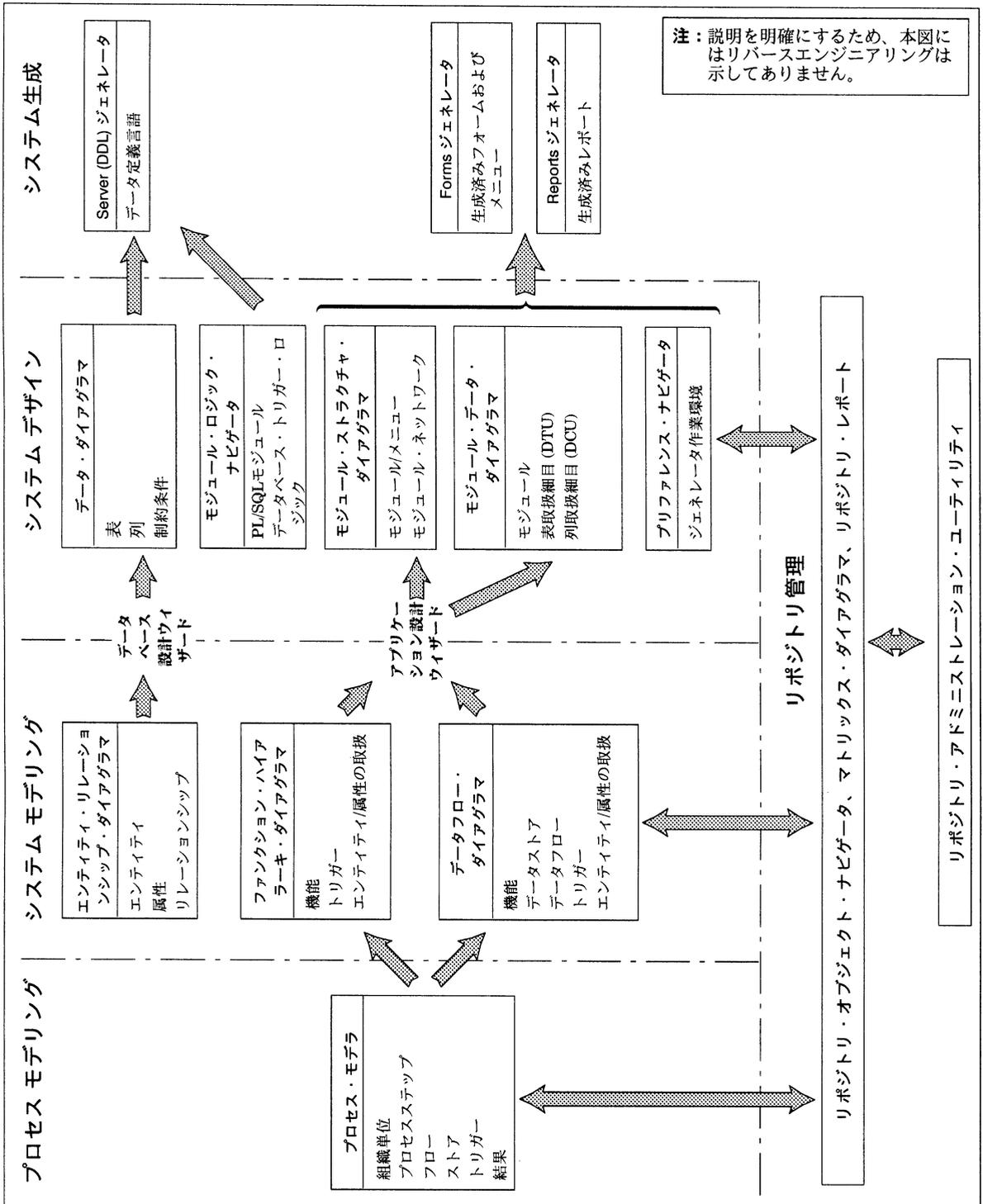
1回の操作で、Designer/2000のツールセット全体をクライアントPCにインストールできます。あるいは、プロセス・モデラ、システム・モデラ、システム・デザイナを別々にインストールすることもできます。ジェネレータ製品については、3つのジェネレータを別々にインストールできます。

プロセス・モデラ、システム・モデラ、あるいはシステム・デザイナをインストールすると、リポジトリ・アドミニストレータのツールセットも自動的にインストールされます。ただし、すでに存在している場合を除きます。

インストールプロセスの詳細については、『Oracle Designer/2000 for Windowsインストールレーション・ガイド』を参照してください。



詳細情報：オラクル社は、リポジトリ・アプリケーション・プログラム・インタフェース（API）も提供しています。これを使用すると、リポジトリの情報の作成と管理をプログラムで行うことができます。APIは、『Designer/2000オンライン マニュアル』の一部として表示されます。



Designer/2000ジェネレータとは何か

Designer/2000ジェネレータを使用すると、リポジトリに保持されている定義からデータベースとアプリケーションを作成することができます。

Designer/2000ジェネレータを使用すると、次のものを作成できます。

- リポジトリに保持されている定義からデータベース オブジェクト
- フォーム・モジュール
- レポート・モジュール

Serverジェネレータ

Serverジェネレータとは、業務システムのサーバー側構成要素の作成と管理を自動化するための一連のユーティリティです。サーバー ジェネレータは、次のユーティリティから構成されています。

- SQL DDL生成ユーティリティ
- 調整ユーティリティ
- リバース・エンジニアリング・ユーティリティ

Formsジェネレータ

Formsジェネレータは、リポジトリ、ユーザー作業環境、テンプレートに記録されている情報から新しいOracle Formsモジュールを作成します。

メニュー

Formsジェネレータは、フォームのメニューモジュールを作成することや、作成するフォームにメニュー モジュールを関連付けることもできます。

Reportsジェネレータ

Reportsジェネレータは、リポジトリ、ユーザー作業環境、テンプレート・レポートに記録されている情報を使用して、Oracle Reports version 2.5またはSQL*Plusのレポート・モジュールを作成します。

ジェネレータへのアクセス

ジェネレータは、以下のDesigner/2000ツールから起動できます。

- リポジトリ・オブジェクト・ナビゲータ
- データ・ダイアグラム (Serverジェネレータ)
- モジュール・ストラクチャ・ダイアグラム
- モジュール・データ・ダイアグラム
- プリファレンス・ナビゲータ (FormsジェネレータとReportsジェネレータ)
- モジュール・ロジック・ナビゲータ (Serverジェネレータ)

ジェネレータのオプションは、上記の各ツールの「ユーティリティ」メニューに表示されます。

使用するツールは何を行うかによって異なりますが、ほとんどの場合にモジュール・データ・ダイアグラムから作成します。

ジェネレータを使用するための条件

いずれかのジェネレータを使用するには、Designer/2000がコンピュータに正しくインストールされていなければなりません。この他に、ジェネレータを使用するための条件は以下の点によっても異なります。

- どのジェネレータを使用するか
- どのようにジェネレータを使用するか（例えば、Formsジェネレータを使用してフォームを作成するための条件は、Serverジェネレータを使用してデータベースのリバース・エンジニアリングを行うための条件と異なります。）

それぞれの条件の詳細は、このマニュアルの残りの部分とオンライン・マニュアルで説明します。

第2章

生成のソース

この章では、アプリケーションを作成するためにジェネレータが使用する情報を定義します。以下の項が含まれています。

- 生成のソースとは何か
- リポジトリとは何か
- データベース・オブジェクト定義とは何か
- モジュール定義とは何か
- モジュール・ネットワークとは何か
- 作業環境とは何か
- テンプレートとは何か

生成のソースとは何か

Serverジェネレータは、リポジトリに保持されているデータベースオブジェクト定義を使用して、業務システムのサーバー側構成要素の作成と管理を行います。

フォームとレポートのジェネレータは、いくつかのソースを組み合わせて生成プロセスを制御します。ソースには、次のようなものがあります。

- リポジトリに保持されている定義
 - データベース・オブジェクト定義
 - モジュール定義
 - モジュール間のリレーションシップ
- 作業環境
- テンプレート

この章では、これらのソースについてさらに詳しく説明します。

リポジトリとは何か

Designer/2000ジェネレータが行うことを理解するには、Designer/2000リポジトリが採用している概念の概要を知らなければなりません。これらを理解すれば、ジェネレータを使用することが困難でなくなります。

リポジトリは、Designer/2000の中核となる要素です。これによって、業務の操作に関する幅広い情報を標準的な形式で記録することができます。この情報を他のDesigner/2000ツールで処理すると、その業務の要件に適合するコンピュータシステムの定義を作成することができます。ジェネレータに関連するリポジトリの情報を、設計レベルの情報と呼びます。これは、次のように分類されます。

- データベース定義
- モジュール定義
- モジュールによるデータの使用
- モジュールによるモジュールの使用

データベース・オブジェクト定義とは何か

データベース定義では、業務システムが処理と格納を行えなければならない情報の種類を定義します。データベースはORACLE表と列によって記述されます。例えば、人事関係のデータベースには、次のような表が格納されます。

DEPARTMENTS表

ID	INTERNAL_COMPANY_ID	NAME	LOCATION
10	1	ACCOUNTING	NEW YORK
20	1	RESEARCH	DALLAS
30	2	SALES	CHICAGO
40	1	OPERATIONS	BOSTON

この表の定義とそこに含まれる列は、列に入力できる値や表間のリレーションシップなど、さまざまな詳細情報とともにリポジトリに格納されます。Serverジェネレータは、この情報を使用して、データベースの作成に必要なコマンドを生成します。

モジュール定義とは何か

業務システムが必要とする機能を実行するプログラム（つまり、データベースに情報を記録したりデータベースから情報を検索するプログラム）は、モジュール定義の形式で記述されます。FormsジェネレータとReporsジェネレータは、リポジトリに保持されているモジュール定義から、動作するプログラムを作成します。

モジュール・データ取扱とは何か

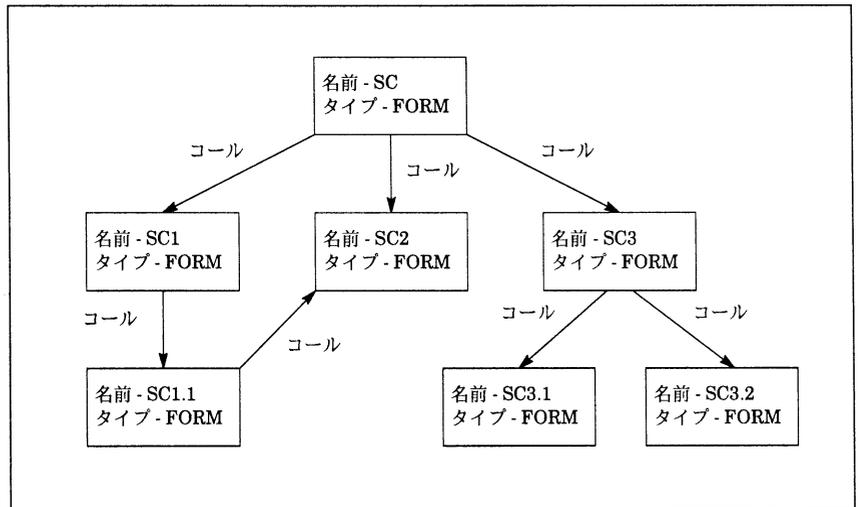
どのモジュールでも、データベース上にある情報の特定の項目を使用することができます。例えば、従業員の給料を変更するモジュールは、EMPLOYEES表のCURRENT_SALARY列を使用します。この、モジュールによるデータの取扱は、リポジトリにも記述されます。取扱情報のタイプは、次のとおりです。

取扱タイプ	説明
モジュール表取扱細目	表取扱細目は、モジュールが使用する表を指定し、その表が使用される特定の状況の詳細な情報を与えます。 表取扱細目は与えられた任意の表モジュールによって多数あります。
モジュール列取扱細目	列取扱細目は、モジュールが使用する列を指定し、その列が使用される特定の状況の詳細な情報を与えます。列取扱細目は、常に表取扱細目の一部となります。 与えられた任意の列の列取扱細目は多数あります。さらに列取扱細目には、同じ列の複数の取扱が含まれている場合があります。

モジュール・ネットワークとは何か

どのモジュールでも、他のモジュールを呼び出すことやインクルードすることができます。例えば、フォームを表すモジュールから別のフォームを呼び出すことができます。同様に、メニューを表すモジュールは、そのオプションを表す各モジュールを呼び出します。

この情報は、モジュール・ネットワークとしてリポジトリに記録され、指定されたモジュールが呼び出すモジュール呼び出し元のモジュールの詳細を記述します。



アプリケーション・システム リポジトリに記録される情報は、複数のアプリケーション・システムに分割されます。各アプリケーション・システムには、業務システム設計の特定の領域を記述するために使用するデータベースとモジュールの定義が含まれます。一般に、アプリケーション・システムは特定のプロジェクトまたはサブプロジェクトを定義します。例えば、システム設計を人事アプリケーション・システムと在庫管理アプリケーション・システムに分割することができます。

作業環境とは何か

作業環境とは、生成したフォームとレポートの外観と動作を制御するパラメータです。例えば、ブロック装飾作業環境 (BLKDEC) は、生成したフォームを囲むブロックに使用する線のスタイルを決定します。

すべての作業環境にデフォルト値 (出荷時設定) があり、あらかじめ作業環境を設定しなくてもフォームとレポートを生成することができます。しかし、特定の結果にするために、作業環境を明示的に設定することもできます。

作業環境レベル

作業環境を設定するレベルによって、変更が影響を与える対象をアプリケーション・システム内の全モジュール、1つのモジュール、モジュールの構成要素 (つまり表取扱細目) のいずれにするかが決定されます。

生成のときに、ジェネレータは最もローカルなレベルから始まる厳格な優先順位で作業環境を処理します。例えば、モジュールとアプリケーション・システムの両方のレベルで作業環境を設定すると、ジェネレータは常にモジュールレベルの作業環境を使用します。特定のレベルで作業環境を設定すると、ジェネレータはその値を使用します。この場合、それより高いレベルの同じ作業環境は検索されません。

どのレベルでも作業環境を設定しなかった場合、ジェネレータはデフォルト値を使用します。

作業環境を使用するための指針

作業環境を使用するための指針を以下に示します。

- 可能な限り高いレベルで作業環境を設定する。

作業環境をアプリケーション・システム全体に対して適用させたい場合は、アプリケーション・システム レベルで作業環境を設定します。

- 名前つき作業環境設定を使用する。

作業環境設定を使用すると、作業環境のグループを一括して管理することができます。名前つき設定は、作業環境階層内のどのレベルでも設定することができます。

- あるレベルで設定した作業環境の値をそれより低いレベルで変更しない。

例えば、オブジェクト (RENTALS表) の作業環境とRENTALSを使用するモジュールの作業環境を設定する場合、モジュール・レベルの作業環境によって常にオブジェクト・レベルの作業環境の設定が変更されます。これはおそらくユーザーの意図と異なるでしょう。

テンプレートとは何か

テンプレート・フォームとは何か テンプレート・フォームとはあらかじめ定義されているフォームで、Formsジェネレータがフォーム定義を作成するためのモデルとして使用できます。テンプレート・フォームを使用すると、さまざまな形式のオブジェクトとプロパティをあらかじめ定義することができます。

生成中は、テンプレート・フォームはオプションですが、テンプレートは非常に便利なので、常に使用することになるでしょう。

テンプレート・メニューとは何か テンプレートメニューとはあらかじめ定義されているメニューで、リポジトリで指定したメニュー構造に追加されます。メニューテンプレートを使用すると、メニュー、メニュー項目、メニューの機密保護の役割をあらかじめ定義することができます。

テンプレート・レポートとは何か テンプレート・レポートとはあらかじめ定義されているレポートで、Reportsジェネレータがレポート定義を作成するためのモデルとして使用できます。テンプレート・レポートには、2つの種類があります。

- レイアウト・モデル・テンプレート・レポート
- パラメータ・フォーム・テンプレート・レポート

レイアウト・モデル・テンプレート・レポートは、さまざまなレポートレイアウトオブジェクトとプロパティを定義します。ドリルダウンレポートの場合、詳細なレポートのために別々のレイアウト・モデル・テンプレートを定義することができます。

パラメータ・フォーム・テンプレート・レポートは、ユーザーがパラメータ値を入力できる実行時のフォームの外観を定義します。

データモデルオブジェクトは、レイアウト・モデル・テンプレート・レポートとパラメータ・フォーム・テンプレート・レポートの両方で定義することができます。

テンプレートオブジェクト テンプレートフォームとレポートには、2種類のテンプレートオブジェクトがあります。

- ユーザーオブジェクト
- ジェネレータオブジェクト

ジェネレータオブジェクトは、新しいオブジェクトの作成や生成したオブジェクトへの一連のプロパティの適用など、何らかの動作を行うようジェネレータに指示します。

ユーザーオブジェクトは、生成したフォームまたはレポートに、変更されないでコピーされます。

テンプレートの作成とテストが終了すると、アプリケーションシステムで基準となるユーザーインタフェースをサポートするテンプレートメニュー、フォーム、レポートの集合が得られます。

第 II 部

サーバの生成

第3章

Serverジェネレータを使用する オブジェクト定義の作成と管理

この章では、Serverジェネレータを使用してOracleデータ・ディクショナリとDesigner/2000リポジトリのデータベース オブジェクトの定義を作成・管理する方法を説明します。

この章には、以下の項が含まれています。

- Serverジェネレータの紹介
- Serverジェネレータへのアクセス
- DDLコマンドの生成
- データベース・オブジェクトのリバース・エンジニアリング
- リポジトリ・オブジェクトの調整

Serverジェネレータの紹介

Serverジェネレータとは何か

Serverジェネレータとは、業務システムのサーバー側構成要素の作成と管理を自動化するための一連のユーティリティです。Serverジェネレータは、次のユーティリティから構成されています。

- SQL DDL作成ユーティリティ
- リバース・エンジニアリング・ユーティリティ
- 調整ユーティリティ

SQL DDL作成ユーティリティとは何か

SQL DDL作成ユーティリティは、リポジトリに記録されている定義からデータベース・オブジェクトを作成するためのデータ定義言語（DDL）の文を生成します。リポジトリに記録されている定義からデータベース・オブジェクトを作成します。これらの文はコマンド・ファイルに書き込まれ、ターゲット・データベース上で実行することができます。生成したDDLには、あるユーザーが他のユーザーに権限を与えることを許可する文を含めることもできます。

リバース・エンジニアリング・ユーティリティとは何か

リバース・エンジニアリング・ユーティリティは、リポジトリ内のデータベース定義とモジュール データ取扱を既存のアプリケーションの要素から導き出します。リバースエンジニアリング操作を行った後で、リポジトリ内に保持されている新しい定義を将来のシステム開発の基礎として使用することができます。

調整ユーティリティとは何か

調整ユーティリティを使うと、アプリケーション・システムのリポジトリ定義とOracleデータ・ディクショナリ中の定義を比較して、リポジトリ定義と同じになるようOracleデータ・ディクショナリ中の定義を変更することができます。調整ユーティリティには、次の2つがあります。

- 相互参照ユーティリティ

このユーティリティは、リポジトリ定義とデータベース・オブジェクトを並べて表示するレポートを作成して、これらの違いを示します。このレポートは、CDRK55.LISと呼ばれます。

- ALTER DATABASEユーティリティ

このユーティリティは、データベース内の既存のオブジェクト定義を作成または変更するために必要なDDL文が含まれるDDLスクリプトファイルを作成します。スクリプトファイルは、CDRK55.SQLと呼ばれます。

接続文字列、リモートノードのユーザー名、パスワードを指定すれば、これらのユーティリティを使用してリモートノード上のデータベースを調整することもできます。

Oracle Serverジェネレータへのアクセス

Oracle Serverジェネレータは、以下のDesigner/2000ツールから起動することができます。

- データ・ダイアグラム
- モジュール・ストラクチャ・ダイアグラム
- モジュール・データ・ダイアグラム
- モジュール・ロジック・ナビゲータ
- リポトリ・オブジェクト・ナビゲータ

次のメニュー・オプションを使用すると、Serverジェネレータを構成するユーティリティにアクセスできます。

- ユーティリティ→SQL DDLの生成
- ユーティリティ→データベースのリバース・エンジニアリング
- ユーティリティ→調整

DDLコマンドの生成

DDLを生成するための前提条件

データベースまたはデータベース・オブジェクトを作成するには、次の条件が満たされていないければなりません。

- 適切な定義がリポジトリ内に存在しなければならない。
- 作成するオブジェクトの定義について作成フラグが設定されていない。

使用するツール

以下のものからDDL作成ユーティリティを起動します。

- データ・ダイアグラム
- モジュール・データ・ダイアグラム
- プリファレンス・ナビゲータ
- モジュール・ストラクチャ・ダイアグラム
- リポジトリ・オブジェクト・ナビゲータ

DDLを生成する前に

サーバー上でオブジェクトを作成する前に、設計を構成するデータベース定義の品質を確認してください。

リポジトリ・レポートは、いくつかの設計品質チェックレポートを作成します。これらのレポートを使用して、リポジトリに記録されているオブジェクトの詳細の完全性を検査してください。

以下のレポートを使用できます。

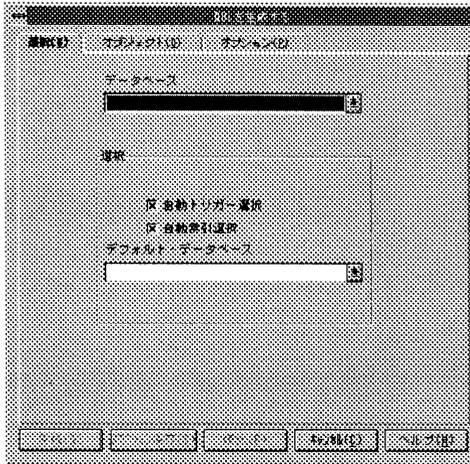
- テーブル品質制御
- データベース トリガー品質制御
- 作成状態品質制御
- データ構造品質制御
- 無効なデータベース オブジェクト
- パッケージ/プロシージャ品質制御
- グループ/ユーザー品質制御

また、以下の点も確認してください。

- Serverジェネレータを実行するデータベースに名前とバージョンがあるか
- データベースとオブジェクト定義の作成フラグが設定されているか
- PL/SQLモジュールが互いにリンクされている場合はリンクが適切か。
PL/SQLを作成するときは、INCLUDEリンクが意味を持ちます。CALLリンクはドキュメントのみのためのものです。

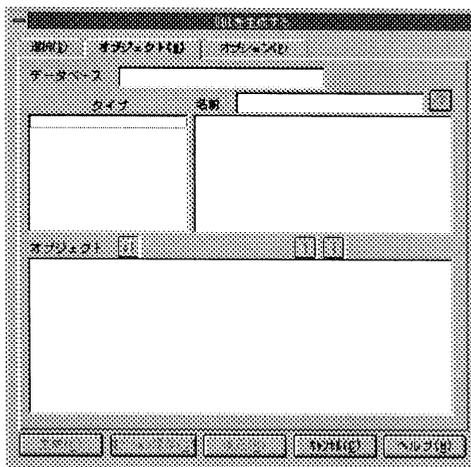
完全な品質保証チェックを行って、見つかった設計上の問題を解決しないと、サーバー生成の可能な完全なデータベース設計ができたことが保証されません。

- DDLコマンドを作成するには 1. 「ユーティリティ→SQL DDLの生成」を選択します。
「DDL生成」ダイアログ・ボックスの「選択」タブが表示されます。



2. データベース・オブジェクトを作成するデータベースの名前を指定します。
3. 「選択」グループ内のチェックボックスで、表そのものを作成するDDLに加えて、その表について定義されているトリガーや索引を作成するDDLも生成するかどうかを指定します。
4. (必要に応じて)「デフォルト・データベース」フィールドで、リポジトリ内で割り当てられていないデータベース・オブジェクトがあると仮定されるデータベースの名前を選択します。

5. 「DDL生成」ダイアログ・ボックスの「オブジェクト」タブを表示します。



6. DDLを生成するオブジェクトのタイプを指定します。

選択したタイプのオブジェクトの名前が「名前」フィールドに表示されます。

7. DDLを生成するオブジェクトを指定するために、次の方法のいずれかを使用して、「名前」フィールドから「オブジェクト」フィールドにオブジェクトを移動させます。

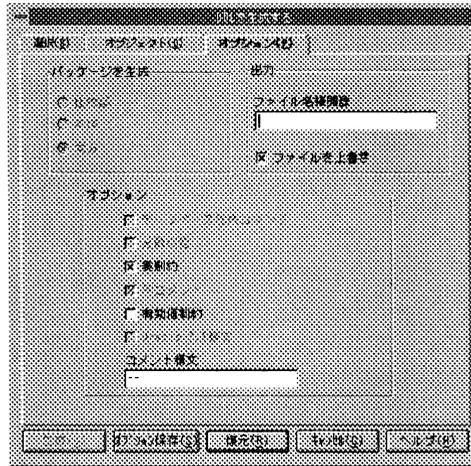
- 「名前」フィールドにあるすべてのデータベース・オブジェクトを「オブジェクト」フィールドに移動させるには、「すべて移動」矢印ボタンを選択します。
- 「名前」フィールドにある1つまたは複数のデータベース・オブジェクトを「オブジェクト」フィールドに移動させるには、移動させたいオブジェクトを選択して「下に移動」矢印ボタンを選択します。

オブジェクトを誤って「オブジェクト」フィールドに移動させた場合は、このリストボックスからオブジェクトを選択して「上に移動」矢印ボタンを選択します。

8. DDLを生成するすべてのオブジェクトを選択するまで、ステップ6と7を繰り返します。

ヒント：「タイプ」リストボックスから複数のオブジェクトのタイプを選択して（オブジェクト名は「名前」リストボックスには表示されません）「すべて移動」矢印ボタンを使用すると、選択したタイプのデータベース・オブジェクトがすべて「オブジェクト」フィールドに移動します。

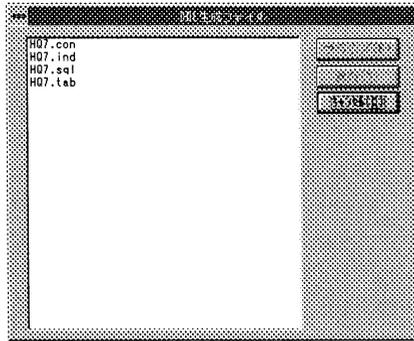
9. 「DDL生成」ダイアログ・ボックスの「オプション」タブを表示します。



10. リポジトリに保持されているパッケージ定義からパッケージ仕様やパッケージ本体を作成するには、「パッケージ生成」グループにある適切なボタンを選択します。
11. 出力ファイルの名前の接頭辞を「ファイル名接頭辞」フィールドに入力します。
出力ファイルの接頭辞がデフォルトで（「DDL生成」－「選択」タブで指定した）データベースの名前になるようにしたい場合は、このフィールドを空白にします。
12. カレントディレクトリにある、生成するファイルと同じ接頭辞のファイルを上書きしたくない場合は、「ファイルの上書き」チェックボックスのチェックを解除します。
13. 「オプション」グループで、適当なオプションの作業環境チェックボックスをチェックします。
14. オプションの作業環境を保存したい場合は、「オプションの保存」ボタンを選択します。
15. 「生成」ボタンを選択して、SQL DDL生成ユーティリティを実行します。

16. 生成が完了したら、メッセージ・ウィンドウを閉じます。

「DDL生成ファイル」ダイアログ・ボックスが表示され、SQL DDL生成ユーティリティによって作成されるコマンド ファイルが表示されます。



ユーティリティは、マスター・コマンド ファイル（接尾辞は.sql）とデータベース・オブジェクト・コマンド・ファイル（このファイルの数と名前は「DDL生成」-「オブジェクト」タブで選択したオブジェクトタイプによって異なります）を作成します。

17. （必要に応じて）コマンド ファイルを表示させるには、「ブラウズ」ボタンを選択します。
18. これで、次の方法のいずれかによってコマンド ファイルを実行できます。

- マスター・コマンド.sqlファイルを選択して実行する。

マスター・コマンド ファイルには、SQL DDL生成ユーティリティによって作成された他のスクリプト・ファイルを実行するためのエントリが含まれています。このファイルを実行すると、データベース・システムを一度に作成することができます。

- DDLスクリプトを別々に実行して、より制御された形式でデータベースを作成する。

DDLスクリプト間の相互依存のため、非常に単純な設計の場合を除いて、この方法で順番にデータベースを実現するとよいでしょう。データベースを順番に作成するときは、.dbファイルから始まる論理的な順番でDDLスクリプトを実行してください。このスクリプトには、データベースを作成するために必要なDDLコマンドが含まれています。次に、.tabファイルを実行して表を作成します。すると、トリガー、スナップショット、ビュー、PL/SQL、索引は表が存在するかどうか依存するので、これらのオブジェクトのDDLスクリプトを実行できるようになります。

注意: 生成したDDLスクリプトは、SQL*Plusプロンプトから直接実行することもできます。

何が生成されるか

SQL DDL生成ユーティリティは、以下のSQLコマンドを生成します。

コマンド	Oracle7	ORACLE RDBMS V6
CREATE DATABASE	Y	Y
CREATE TABLE	Y	Y
CREATE VIEW	Y	Y
CREATE INDEX	Y	Y
CREATE CLUSTER	Y	Y
CREATE SEQUENCE	Y	Y
CREATE TABLESPACE	Y	Y
CREATE ROLLBACK SEGMENT	Y	Y
CREATE USER	Y	N
CREATE ROLE	Y	N
CREATE PACKAGE	Y	N
CREATE PACKAGE BODY	Y	N
CREATE DATABASE LINK	Y	N
CREATE FUNCTION	Y	N
CREATE PROCEDURE	Y	N
CREATE SNAPSHOT	Y	N
CREATE SNAPSHOT LOG	Y	N
CREATE SYNONYM	Y	N
CREATE TRIGGER	Y	N
GRANT	Y	N

コマンドファイル名

作成されるデータベース オブジェクト コマンド ファイルの名前は、次のようになります。

オブジェクトのタイプ	作成されるファイル名
DATABASE	<出力ファイル名>.db
TABLE	<出力ファイル名>.tab
VIEW	<出力ファイル名>.vw
INDEX	<出力ファイル名>.ind
CLUSTER	<出力ファイル名>.cls
SEQUENCE	<出力ファイル名>.seq
TABLESPACE	<出力ファイル名>.tsp
ROLLBACK SEGMENT	<出力ファイル名>.rgr
USER GRANTS	<出力ファイル名>.ugr
ROLE GRANTS	<出力ファイル名>.rgr
PACKAGE	<出力ファイル名>.pck
FUNCTION	<出力ファイル名>.fnc
PROCEDURE	<出力ファイル名>.prc
SNAPSHOT	<出力ファイル名>.ss
SYNONYM	<出力ファイル名>.syn
TRIGGER	<出力ファイル名>.tgr
CONSTRAINTS	<出力ファイル名>.con

データベース オブジェクトのリバース エンジニアリング

データベース オブジェクトのリバース エンジニアリングの前提条件

データベース オブジェクトをリバース エンジニアリングしてリポジトリに格納するには、リバースエンジニアリングを行うオブジェクトが含まれるデータベースにアクセスできなければなりません。

使用するツール

以下のものからリバース エンジン ユーティリティを起動します。

- データ・ダイアグラム
- モジュール・データ・ダイアグラム
- モジュール・ストラクチャ・ダイアグラム
- リポジトリ・オブジェクト・ナビゲータ

データベース オブジェクトのリバース エンジニアリングの2つのステージ

多くの場合、リバース エンジニアリングは2つのステージで処理されます。

- ステージ1では、データベースオブジェクトをリバースエンジニアリングしてモジュール・データ取扱を作成します。
- ステージ2では、業務モデルをさらに詳しく再作成します。

この結果、完全にクロスリファレンスされ統合されたアプリケーション・システムが作成され、将来の拡張のための安定した基礎が得られます。

ステージ1

ステージ1には以下の処理が含まれます。

- データベース オブジェクトのリバース エンジニアリング
- モジュール データ取扱の作成
- Oracle Formsアプリケーションのリバース エンジニアリング

このステージが完了すると、リバースエンジニアリング処理のほとんどが終わったこととなります。それまで管理したり文書化したりできなかったであろう既存のアプリケーション・システムも、保守が可能になります。

ステージ2

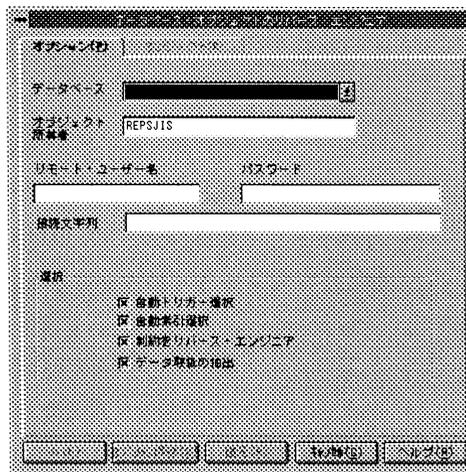
このステージはオプションですが、次のことを行うと業務モデルの詳細な情報が得られます。

- ファンクション・ハイアラキ・ダイアグラムを作成する
- (表からエンティティへのレトロフィット ユーティリティを使用して) 物理データベース定義テーブルからたたき台となるべきエンティティを導き出す
- リバース エンジンを行ったデータでトップダウンの業務モデルを合理化する

データベース オブジェクトの
リバース エンジニアリングを
行うには

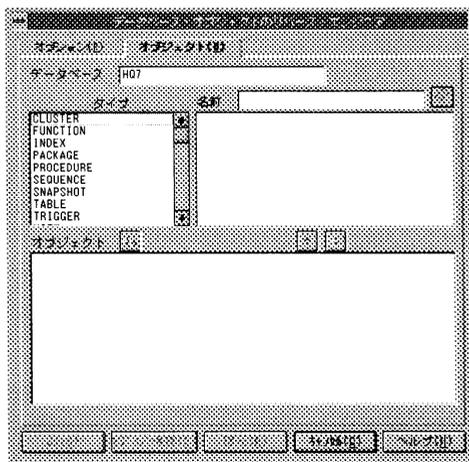
1 ユーティリティ→リバース エンジニア→データベースを選択します。

「データベース・オブジェクトのリバース・エンジニア」ダイアログ・ボックスの「オプション」タブが表示されます。



2. リバースエンジニアを行う定義を格納する、リポジトリ内で定義されたデータベースを指定します。
3. リバース エンジニアを行うオブジェクトを現在所有しているユーザー（デフォルトは現在のリポジトリのユーザー）を指定します。
4. リバースエンジニアを行うオブジェクトがリポジトリと同じデータベースにない場合は、次のようにします。
 - a. リバース エンジニアを行うオブジェクトの所有者のユーザー名とパスワードを「リモートユーザー名」と「パスワード」フィールドで指定します。
 - b. リバースエンジニアを行うオブジェクトが含まれるデータベースに接続するために必要な文字列を「接続文字列」フィールドで指定します。
5. リバース・エンジニアを行う表に関連するトリガー、索引、制約、データ取扱をリバースするには「選択」グループ内のチェックボックスを使用します。

6. 「データベース・オブジェクトのリバース・エンジニア」ダイアログ・ボックスの「オブジェクト」タブを表示させます。



7. リバースエンジニアを行うオブジェクトのタイプを指定します。
選択したタイプのオブジェクトの名前が「名前」フィールドに表示されます。
ヒント：期待したオブジェクトが表示されない場合は、「オプション」タブの「オブジェクト所有者」フィールドの名前がリモートデータベース上のオブジェクトを所有するユーザーになっているか確認してください。
8. リバースエンジニアを行うオブジェクトを指定するために、次の方法のいずれかを使用して、「名前」フィールドから「オブジェクト」フィールドにオブジェクトを移動させます。
- 「名前」フィールドにあるすべてのデータベース オブジェクトを「オブジェクト」フィールドに移動させるには、「すべて移動」矢印ボタンを選択します。
 - 「名前」フィールドにある1つまたは複数のデータベース オブジェクトを「オブジェクト」フィールドに移動させるには、移動させたいオブジェクトを選択して「下に移動」矢印ボタンを選択します。
- オブジェクトを誤って「オブジェクト」フィールドに移動させた場合は、このリストボックスからオブジェクトを選択して「上に移動」矢印ボタンを選択します。
9. リバースエンジニアを行うすべてのオブジェクトを選択するまで、ステップ7と8を繰り返します。
ヒント：「タイプ」リストボックスから複数のオブジェクトのタイプを選択して（オブジェクト名は「名前」リストボックスには表示されません）「すべて移動」矢印ボタンを使用すると、選択したタイプのデータベース オブジェクトがすべて「オブジェクト」フィールドに表示されます。
10. 「OK」をクリックすると、リバース エンジニア処理が開始します。
メッセージ ボックスが表示され、処理の進行状況が表示されます。

「Retrofit complete (レトロフィットが完了しました)」というメッセージが表示されると、リバース エンジニア処理が終了します。リバース エンジニアを行ってリポジトリに格納したデータベースオブジェクトの詳細については、リバース エンジニア処理で作成されるレポート「CDRF.LIS」を参照してください。

何がリバース エンジニアされるか

リバース・エンジニアリング・ユーティリティは、Oracleデータ・ディクショナリに格納されているオブジェクトをもとにリポジトリ内にオブジェクト定義を作成します。

Oracleユーザーのデータベースオブジェクトは、現在のユーザーがそのオブジェクトのSELECT権限を持っていれば、リバース エンジニアを行うことができます。

このユーティリティは、次のようなOracleデータベースオブジェクトをリバース エンジニアできます。

データベース オブジェクト	Oracle7	ORACLE RDBMS V6
TABLE	Y	Y
VIEW	Y	Y
INDEX	Y	Y
CLUSTER	Y	Y
SEQUENCE	Y	Y
TABLESPACE	Y	Y
PACKAGE	Y	N
FUNCTION	Y	N
PROCEDURE	Y	N
SNAPSHOT	Y	N
TRIGGER	Y	N

リポジトリ オブジェクトの調整

調整ユーティリティを使用するための前提条件 物理データベースをリポジトリオブジェクト定義と調整させるには、次のような条件が満たされなければなりません。

- リポジトリと比較するデータベースにアクセスできなければならない
- 調整させるデータベースがリポジトリ内で定義されていなければならない

使用するツール

以下のものから調整ユーティリティを起動します。

- データ・ダイアグラム
- モジュール・データ・ダイアグラム
- モジュール・ストラクチャ・ダイアグラム
- モジュール・ロジック・ナビゲータ
- リポジトリ・オブジェクト・ナビゲータ

調整の処理

調整は、2つのステージで処理されます。

- ステージ1では、相互参照ユーティリティを実行して（リポジトリに保持されている）データベースの論理定義とサーバー上に存在しているデータベース オブジェクトを比較します。

相互参照ユーティリティが作成するレポートを使用して、リポジトリに保持されている定義とデータベースの違いを調べます。

- ステージ2では、2つの定義の間に著しい違いがある場合に適切な作業を行います。

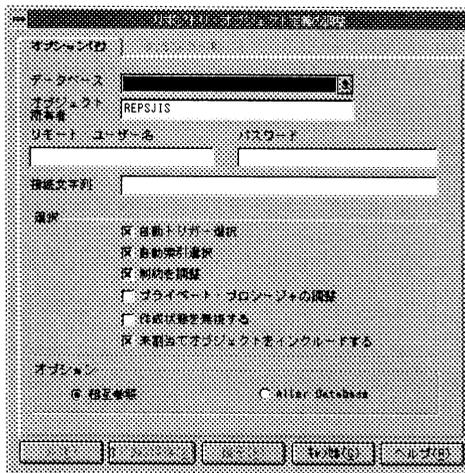
オブジェクトのリポジトリ定義の方が対応するデータベース オブジェクトより新しい場合は、ALTER DATABASEユーティリティを実行します。

オブジェクトの定義がリポジトリ内に存在していてデータベース上には存在していない場合は、SQL DDL生成ユーティリティを実行してオブジェクトを作成します。

データベースのオブジェクトの定義が対応するリポジトリ内の定義より新しい場合、あるいは、オブジェクトがデータベースに存在していて対応する定義がリポジトリ内にはない場合は、リバース エンジニア ユーティリティを実行します。

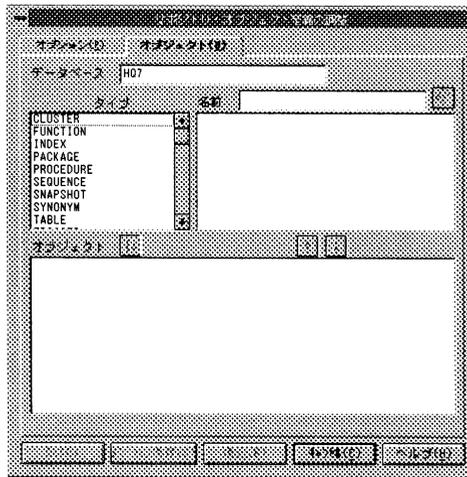
リポジトリ オブジェクトを調整させるには 1. ユーティリティ→調整を選択します。

「リポジトリ オブジェクト定義の調整」ダイアログボックスの「オプション」タブが表示されます。



2. データベースオブジェクトを調整するリポジトリ内で定義されたデータベースを指定します。
3. 調整するオブジェクトを現在所有しているユーザー（デフォルトは現在のリポジトリのユーザー）を指定します。
4. 調整するオブジェクトがリポジトリと同じデータベースにない場合は、次のようにします。
 - a. 調整するオブジェクトの所有者のユーザー名とパスワードを指定します。
 - b. 調整するオブジェクトが含まれるデータベースに接続するために必要な文字列を指定します。
5. 「選択」チェックボックスを使用して、調整するテーブルに関連するトリガー、索引、制約、列取扱を調整させます。

6. 「オプション」グループで、「相互参照」オプション ボタンまたは「ALTER DATABASE」オプション ボタンのいずれかを選択します。
 - 「相互参照」オプション ボタンを選択すると相互参照ユーティリティが実行され、リポジトリの定義とデータ・ディクショナリを並べて表示したレポートが作成され、これらの違いが示されます。
 - 「ALTER DATABASE」オプション ボタンを選択するとALTER DATABASEユーティリティが実行され、相互参照の結果として識別されるオブジェクトに適したALTER DDL文が含まれるスクリプトファイルが作成されます。
7. 「オブジェクト」タブを表示させます。



8. 調整するオブジェクトのタイプを指定します。
 選択したタイプのオブジェクトの名前が「名前」フィールドに表示されます。
 ヒント：期待したオブジェクトが表示されない場合は、「オプション」タブの「オブジェクト所有者」フィールドの名前がリモートデータベース上のオブジェクトを所有するユーザーになっているか確認してください。
9. 調整するオブジェクトを指定するために、次の方法のいずれかを使用して、「名前」フィールドから「オブジェクト」フィールドにオブジェクトを移動させます。
 - 「名前」フィールドにあるすべてのデータベース オブジェクトを「オブジェクト」フィールドに移動させるには、「すべて移動」矢印ボタンを選択します。
 - 「名前」フィールドにある1つまたは複数のデータベース オブジェクトを「オブジェクト」フィールドに移動させるには、移動させたいオブジェクトを選択して「下に移動」矢印ボタンを選択します。

オブジェクトを誤って「オブジェクト」フィールドに移動させた場合は、このリストボックスからオブジェクトを選択して「上に移動」矢印ボタンを選択します。

- 調整するすべてのオブジェクトを選択するまで、ステップ9と10を繰り返します。

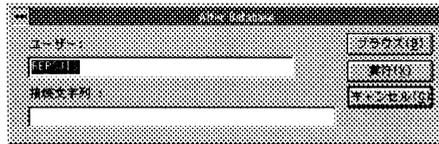
ヒント：「タイプ」リストボックスから複数のオブジェクトのタイプを選択して（オブジェクト名は「名前」リストボックスには表示されません）「すべて移動」矢印ボタンを使用すると、選択したタイプのデータベース オブジェクトがすべて「オブジェクト」フィールドに表示されます。

- 「OK」をクリックすると、選択した調整ユーティリティが実行されます。

相互参照ユーティリティは、CDRK55.LISという名前のレポートを作成します。リポジットリに保持されている定義とデータベース内のオブジェクトの違いを調べます。

ALTER DATABASEユーティリティは、CDRK55.SQLという名前のスクリプト ファイルを作成します。

ALTER DATABASEユーティリティを実行している場合は、「ALTER DATABASE」ダイアログ ボックスが表示されます。



- （必要に応じて）CDRK55.SQLファイルに含まれているSQLコマンドを調べるには、「ブラウズ」ボタンを選択します。
- （必要に応じて）CDRK55.SQLファイルのコマンドを実行するには、「実行」ボタンを選択します。

何が調整されるか

相互参照ユーティリティ

相互参照ユーティリティは、以下のデータベースオブジェクト定義をリポジトリの定義と比較して、違いを知らせるレポート（ファイル名はCDRK55.LIS）を作成します。

オブジェクトのタイプ	Oracle7	ORACLE RDBMS V6
CLUSTER	Y	Y
FUNCTION	Y	N
INDEX	Y	Y
PACKAGE	Y	N
PROCEDURE	Y	N
SEQUENCE	Y	Y
SNAPSHOT	Y	N
TABLE	Y	Y
TRIGGER	Y	N
VIEW	Y	Y

ALTERDATABASEユーティリティ

相互参照ユーティリティが実行した比較の結果から、ALTER DATABASEユーティリティはデータベース・オブジェクトを変更するための適切なALTER文が含まれるスクリプト ファイル（ファイル名はCDRK55.SQL）を作成します。

作成されるALTER文	Oracle7	ORACLE RDBMS V6
ALTER CLUSTER	Y	Y
ALTER INDEX	Y	Y
ALTER SEQUENCE	Y	Y
ALTER TABLE	Y	Y
ALTER TRIGGER	Y	N
ALTER VIEW	Y	Y
ALTER SNAPSHOT	Y	N

注記

調整ユーティリティを使用するときは、以下の点に注意してください。

- Oracleデータ・ディクショナリ中に存在するパッケージ、プロシージャ、ファンクションについては、その中の有効なもの（ステータスがvalidであるもの）のみが調整可能です。
- パッケージ、プロシージャ、ファンクションの調整では、PL/SQL文の解析のために時間がかかる場合があります。
- リモート サーバー上のオブジェクトの調整では、データベース リンクを通してデータを参照するために時間がかかる場合があります。

第 III 部

フォームの作成

第4章

Oracle Forms ジェネレータの 紹介

この章では、OracleFormsジェネレータを紹介し、いくつかの基本的な概念を説明します。

- Oracle Formsとは何か
- Oracle Formsジェネレータとは何か
- フォーム作成処理の概要
- Designer/2000とDeveloper/2000の用語
- 作成したフォームの例

Oracle Formsとは何か

OracleFormsはフォームアプリケーションを作成して実行するためのツールで、Oracleのクライアント/サーバー開発ツールセットであるDeveloper/2000の構成要素の1つです。フォームアプリケーションとは、ユーザーがデータベースの情報を照会、更新、入力、削除できるようにする対話形式のプログラムです。

フォームアプリケーションは、モジュールと呼ばれるアプリケーション構成要素から構成されます。Oracle Formsモジュールには、次のような3つのタイプがあります。

フォームモジュール オブジェクトとコードのルーチンの集まり。フォームモジュールで定義できるオブジェクトには、ウィンドウ、テキスト項目（フィールドとも呼びます）、チェックボックス、ボタン、警告、値のリスト、トリガーと呼ばれるPL/SQLコードのブロックがあります。多くの場合、フォームモジュールを単にフォームと呼びます。

メニューモジュール アプリケーションメニューを構成するメニュー（メインメニューオブジェクトとサブメニューオブジェクト）とメニュー項目コマンドの集まり。

注意: メニューモジュールはフォームアプリケーションの一部で、単体では実行できません。

ライブラリモジュール フォームアプリケーション内の他のモジュールから呼び出せる、ユーザーが名前をつけるプロシージャ、ファンクション、パッケージの集まり。

フォームアプリケーションは、1つのフォームモジュールだけから構成される場合もあります。メニューモジュールは単独ではフォームアプリケーションを構成しないので、メニューモジュールだけを実行することはできません。メニューモジュールを実行するには、あらかじめフォームモジュールと関連付けなければなりません。

フォームアプリケーションの作成 Oracle Forms Designerは、フォームを手作業で作成するために使用するツールです。しかし、業務ルールと条件がDesigner/2000のリポジトリに記録されていれば、Designer/2000のFormsジェネレータを使用してフォームアプリケーションを作成すると非常に効率よくなります。

Formsジェネレータとは何か

Formsジェネレータとは、Designer/2000のリポジトリに記録されているモジュール定義からOracle Formsアプリケーションを作成するプログラムです。

作成したフォームのブロックとフィールドは、リポジトリモジュールが使用する表と列によって決定されます。

ユーザー作業環境とテンプレートフォームは、作成したフォームの外観と動作を制御します。

Formsジェネレータは、リポジトリで定義されているモジュール・ネットワークからメニューモジュールも作成します。

フォームを作成したら、Oracle Forms Designerを使用して作成したフォームを変更することができます。行った変更が含まれるようにリポジトリのモジュール定義を更新するには、Formsジェネレータを使用してフォームを再生成します。

リポジトリ内で定義されているモジュールからFormsジェネレータで作成したものではないOracle Formsモジュールがある場合、Formsジェネレータを使用して対応するリポジトリモジュールを作成することができます。この処理は、リバースエンジニアリングと呼ばれます。

フォーム作成処理の概要

フォーム作成処理には、一般に次のステップのいずれかあるいはすべてが含まれます。

- モジュールの定義
- 表および列取扱の定義
- 作業環境の設定
- テンプレートの使用
- フォームの作成
- フォームの精製

再生成やリバースエンジニアリングのためにFormsジェネレータを使用することもできます。

モジュールの定義

Formsジェネレータはリポジトリのモジュールからフォームを作成するので、適切なモジュールがリポジトリ内に存在していなければなりません。リポジトリ内のモジュールを作成するには、アプリケーション設計ウィザードを実行するか手作業で行います。

表および列取扱の定義

Formsジェネレータは、リポジトリのモジュールが使用する表と列をもとにフォーム上にブロックとフィールドを作成します。

アプリケーション設計ウィザードを使用してモジュールを作成する場合、表および列取扱は自動的に定義されます。

手作業でモジュールを作成する場合は、表および列取扱を自分で定義しなければなりません。

作業環境の設定

作業環境を設定すると、作成したフォームの外観と動作を制御することができます。

テンプレートの使用

テンプレートを使用すると、作成したオブジェクトのプロパティを定義して、作成したフォームにあらかじめ定義したオブジェクトを含めることができます。

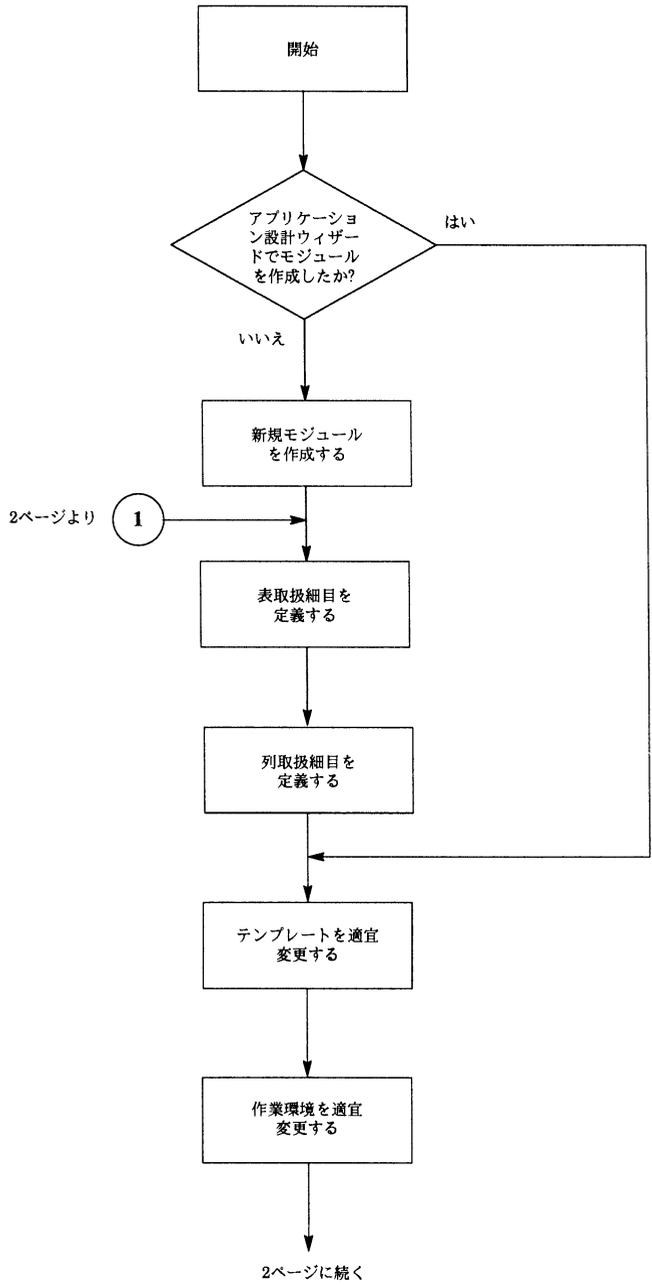
フォームの作成

以下のツールからフォームを作成することができます。

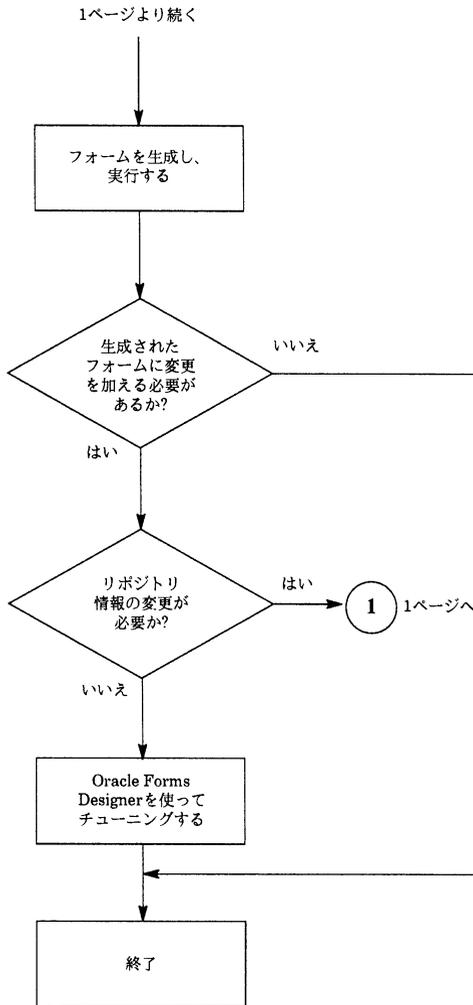
- モジュール・データ・ダイアグラム
- プリファレンス・ナビゲータ
- モジュール・ストラクチャ・ダイアグラム
- リポジトリ・オブジェクト・ナビゲータ

フォームを作成している途中で、フォームを実行して業務条件に適合しているか確認することができます。

- フォームの精製** 作成したフォームが条件に適合しない場合は、Designer/2000のツールを使用してリポジトリ内の適当な情報を変更します。
- 特に、データ取扱細目と作業環境設定は変更の必要があるでしょう。また、ジェネレータがテンプレートから取得する情報も変更する必要があるでしょう。
- 再生成** フォームを作成したら、Oracle Forms Designerを使用して作成したフォームを変更することができます。行った変更が含まれるようにリポジトリのモジュール定義を更新するには、Formsジェネレータを使用してフォームを再生成します。
- リバース エンジニアリング** リポジトリ内で定義されているモジュールからFormsジェネレータで作成したものではないOracleFormsモジュールがある場合、Formsジェネレータを使用して対応するリポジトリモジュールを作成することができます。この処理は、リバースエンジニアリングと呼ばれます。



フォーム生成の流れ (1/2ページ)



フォーム生成の流れ (2/2ページ)

Designer/2000とDeveloper/2000の用語

以下の表に、Designer/2000のリポジトリで定義とプロパティを記述するために使用する用語と、Oracle Formsジェネレータが作成したものを記述するためにDeveloper/2000のOracle Forms構成要素が使用する用語の間の関係を示します。

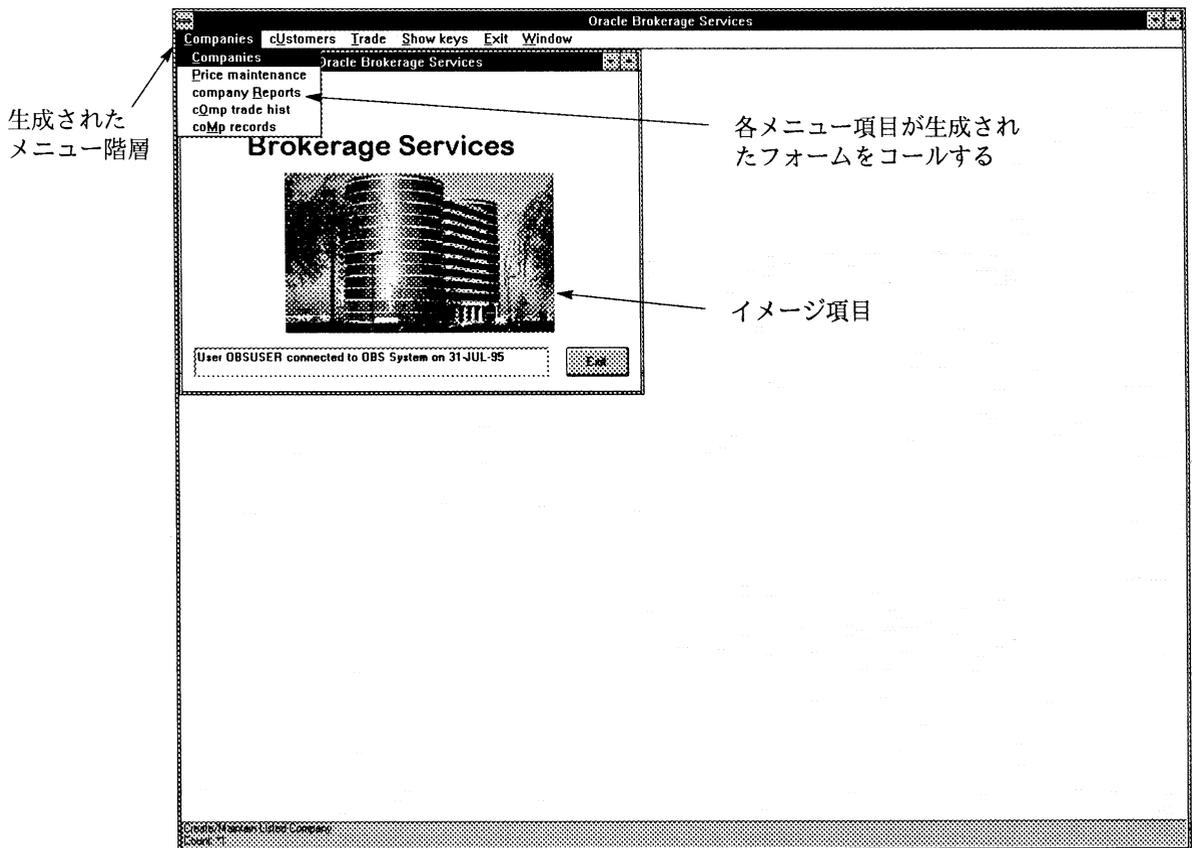
Designer/2000の用語	作成されるもの (Developer/2000の用語)	注意
(タイプがSCREENの) モジュール	フォーム アプリケーション	実行可能アプリケーション
モジュール ネットワーク	メニュー モジュール	他のプログラム (レポート アプリケーション) を起動するメニュー項目が含まれるサブメニューとメニューの階層
ページ	コンテキスト・キャンバス	項目がレイアウトされる領域。ウィンドウに同時に表示できるのは1つのコンテキスト・キャンバスだけ。 メニューを実行するには、メニューがフォームと関連づけられていなければなりません。
ポップアップ	スタック キャンバス	ウィンドウ内に現在表示されているキャンバスの一部またはすべてを覆うキャンバス。 注: ジェネレータは、オーバーフロー要件を実現するためと、テンプレート・フォームで定義されたスタック・キャンバスを含めるとするとき、スタック・キャンバスも作成します。
ビュー	キャンバスビュー	実行時にウィンドウ内に表示されるキャンバスの領域
基礎表取扱細目	ブロック	関連する項目をグループ化するためのメカニズム
列取扱細目	項目	ユーザーがアプリケーションと対話できるようにするインタフェースオブジェクト

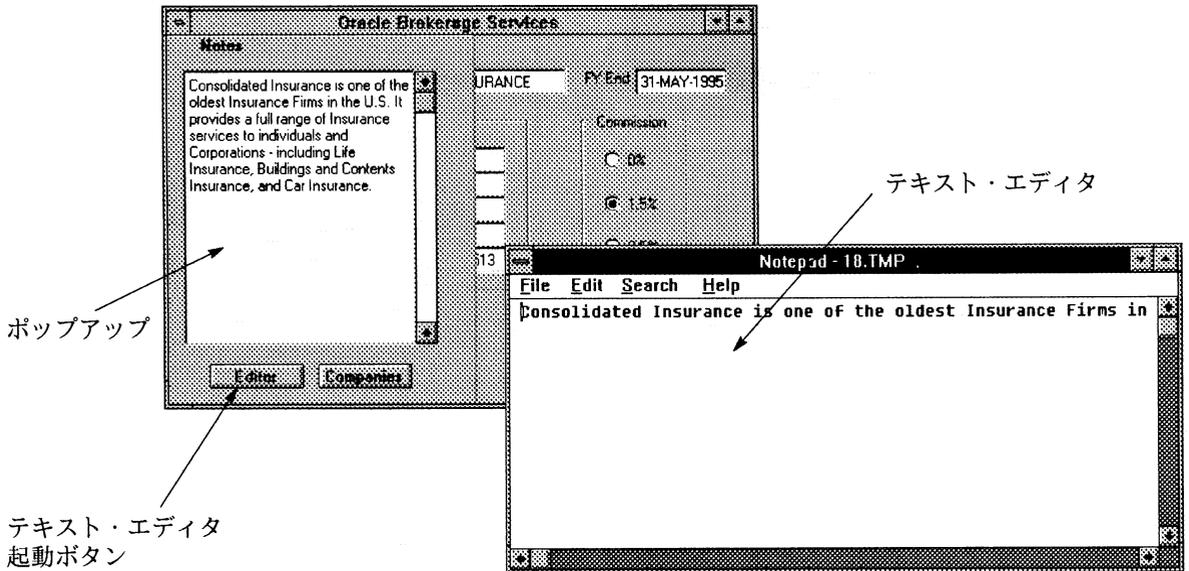
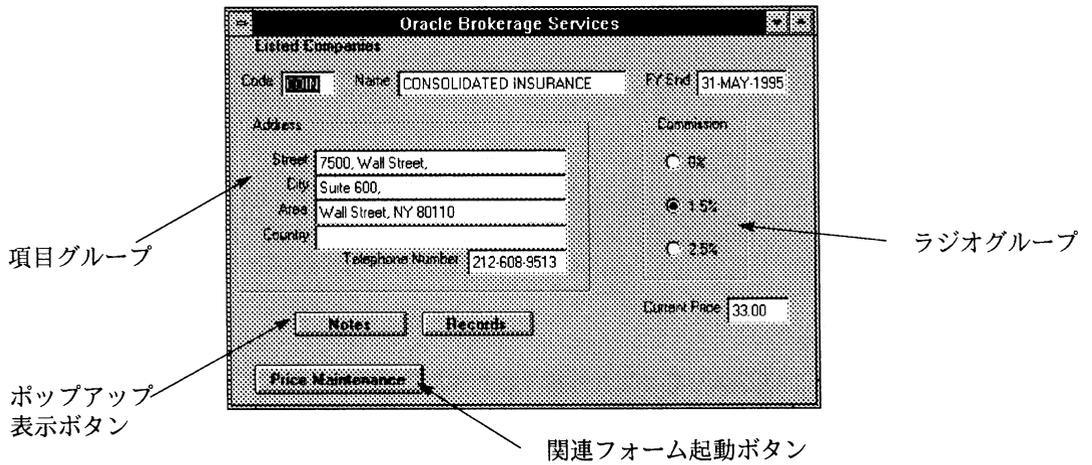
作成したフォームの例

以下の章では、作業例として詳細な指示があるので、Formsジェネレータを効率的に使用するためのスキルが得られます。作業例で作成するフォームは意図的に単純なものになっており、Formsジェネレータの特定の機能を示すために使用されま

す。
この項では、Oracle Formsジェネレータを使用して作成するフォームについての実際的な指示が与えられます。下に示すアプリケーションはいくつかのフォームモジュールから構成されており、すべてOracle Formsジェネレータを使用して作成したものです。

Oracle仲買サービス デモ アプリケーションから組み合わせたフォーム





Oracle Brokerage Services

Listed Company

Code: DBN Name: CONSOLIDATED INSURANCE Price: 93.00

Date	Open	Low	High	Close	\$20	Open	\$50
18-JUL-1995	39.00	33.00	39.00				
17-JUL-1995	39.00	39.00	39.00	39.00			
16-JUL-1995	36.00	34.00	39.00	39.00			
15-JUL-1995	30.00	28.00	36.00	36.00			
14-JUL-1995	34.00	30.00	34.00	30.00			
13-JUL-1995	32.25	32.25	34.00	34.00			

Trade

VBXコントロール項目

Oracle Brokerage Services

Customer Details

First Name: Last Name: Reference:

Address:

Street: City: Area: County: Telephone Number:

Employee:

Credit Details

Cash Balance: Allow Credit:

Portfolio Reserves Notes Statement

導出された項目

チェックボックス

Oracle Reports
レポート実行ボタン

第5章

単一ブロックフォームの作成

この章では、デフォルトのオプションを使用して単一ブロックフォームを作成する方法を示します。以下の項が含まれています。

- デフォルト設定を使用するフォームの作成
 - モジュールの作成
 - データ取扱細目の定義
 - フォームの作成
- 作成したフォームのプロパティ
- モジュール定義、作業環境、テンプレートの精製とカスタマイズ

デフォルト設定を使用するフォームの作成

通常は、戦略、分析、設計の作業に、システムのさまざまな構成要素についての識別されたモジュールがあります。

アプリケーション設計ウィザードを使用して機能をモジュールに変換した場合は、いくつかのモジュールがリポジトリに定義されているのでそこからフォームを作成できるでしょう。

作成するフォームについてモジュールが存在している場合は、この章の後の方にある「フォームの作成」で説明されているフォームの作成に進むことができます。

(おそらくは分析が完了していないために) アプリケーション設計ウィザードを使用していない場合は、モジュールを手作業で作成しなければなりません。

この章の作業例では、賃貸に関する情報の入力と表示を行うフォームを作成するためのモジュールを作成する方法を示します。

モジュールの作成

フォームを作成するためのリポジトリ・モジュールを手作業で作成するには、次のようにします。

1. モジュール・データ・ダイアグラマを起動します。
2. 接続したいOracleデータベースのユーザー名、パスワード、接続文字列を入力します (インストールによっては、このステップが不要な場合もあります)。
3. アプリケーション・システムとバージョン番号を選択します。
4. 「ファイル→新規作成」を選択して「新規ダイアグラム」ダイアログボックスを表示させて、新しいダイアグラムを作成します。



5. 「モジュールの作成」ボタンをクリックして、新しいモジュールを作成します。

「モジュールの作成」ダイアログ・ボックスが表示されます。

6. モジュールの短縮名、名前、目的を該当するフィールドに入力します。
作業例では、以下のように入力しています。

短縮名	REN0045
モジュール名	レンタル情報の登録・検索
目的	レンタルに関する情報の登録・検索

7. モジュール言語としてOracle Formsを指定します。
8. モジュールのタイプとしてSCREENを指定します。

ヒント：似たようなモジュールがすでに存在している場合は、「コピー・モジュール」ドロップダウン・リストを使用すると既存のモジュールを素早くコピーできます。

9. 「OK」をクリックすると、モジュールが作成され「モジュールの作成」ダイアログ・ボックスがクローズします。

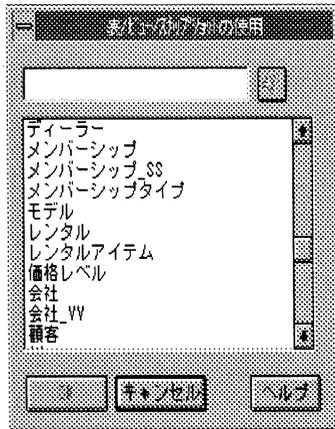
データ取扱細目の定義

モジュールを作成してタイプをSCREENと定義したら、モジュールのデータ取扱細目を指定しなければなりません。

データ取扱細目の定義

まず、モジュールが使用する表を指定します。

1. 「編集→作成→表取扱細目」を選択して、「表取扱細目の作成」ダイアログ・ボックスを表示させます。

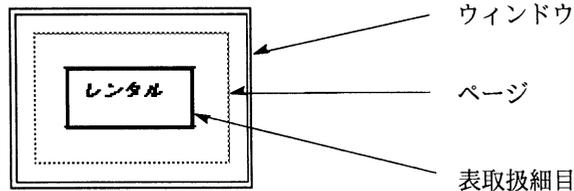


2. 「表取扱いの作成」ダイアログ・ボックスから1つまたは複数の表を選択します。

この例では、レンタルを選択しています。

3. 「OK」をクリックすると、モジュールの表取扱いが作成されます。

選択した表の取扱細目が、下図のようにモジュール・データ・ダイアグラムに表示されます。



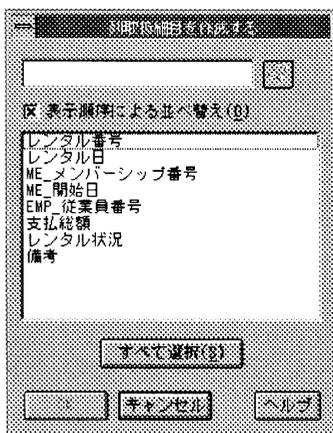
このダイアグラムには、表取扱い細目を示す四角形が作成されるだけでなく、ブロックのページ配置項目とウィンドウも示されます。このダイアグラムには、レンタルに基づくブロックが自分のウィンドウの自分のページに表示されることが示されています。

注意：列取扱細目を定義する前に、表取扱について適切な取扱フラグを設定するようにしてください。これらのフラグは、生成されたフォームで使用する表の情報について可能となる操作を定義し、その列取扱細目にも継承されます。列取扱細目を作成する前にフラグを設定すると、後で時間が節約されます。ただし、この章で示すデフォルトの作成のため、ここでは取扱フラグを設定しません。

列取扱細目の定義

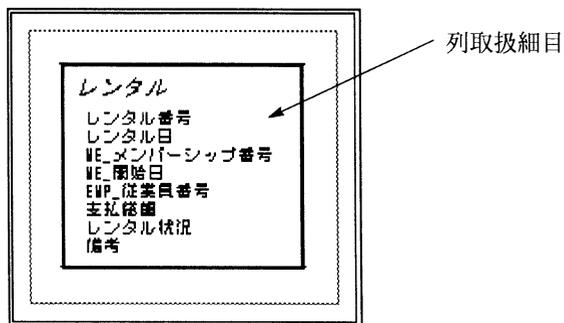
モジュールが使用する表を指定したら、モジュールが使用する列を表から指定します。下記の例では、レンタル表のすべての列について列取扱細目を作成します。

1. 表取扱細目をクリックして選択します。
2. 「編集→作成→列取扱細目」を選択して、「列取扱細目の作成」ダイアログ・ボックスを表示させます。



3. 列取扱細目を初めにアルファベット順で並べるか（リポジトリで定義した）表示順で並べるか決定します。
この例では、「表示順」チェックボックスのチェックが解除されています。
4. 取扱細目を作成する列を選択します。
この例では、「すべて選択」ボタンを使用してダイアログ・ボックス内のすべての列を強調表示させています。
5. 「OK」をクリックすると、選択した列の取扱細目が作成されます。

下図のように、列取扱細目がモジュール・データ・ダイアグラムに表示されます。



フォームの生成

モジュールの表取扱細目と列取扱細目を指定すると、モジュールからフォームを生成できるようになります。

表示されるメッセージは、指定した表と列の取扱によって異なります。この項で示すメッセージでは、この章で説明したレンタル表とその列の取扱細目が仮定されています。

1. モジュール・データ・ダイアグラムで「ユーティリティ→生成」を選択します。

下図のように、「フォームの生成」ダイアログ・ボックスが表示されます。



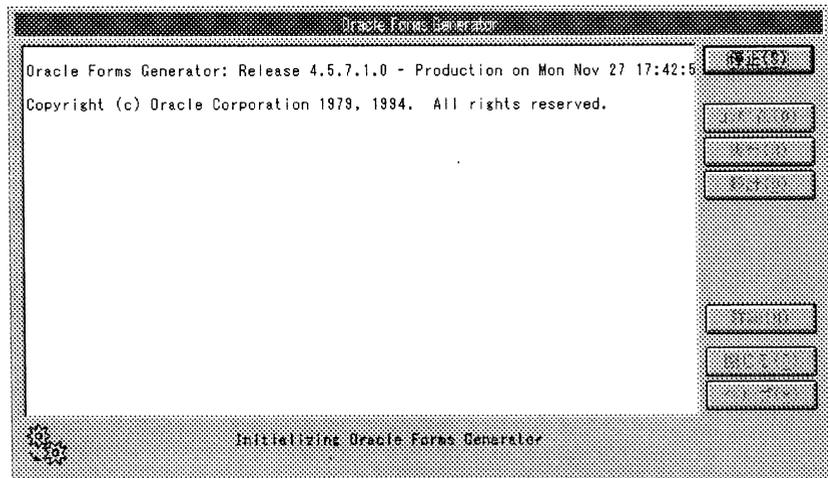
現在のモジュールは「生成」タブで強調表示されます。ダイアログ・ボックス内の他のタブを使用すると、作成したフォームをジェネレータが保存する場所など、Formsジェネレータの動作を制御できます。

「リポジトリの更新」チェックボックスがチェックされていることに注意してください。

この例では、変更は不要です。

2. 「開始」ボタンをクリックすると、「モジュール」フィールドに表示されているモジュールからのフォームの生成が開始します。

下図のように、「Oracle Formsジェネレータ」ダイアログボックスが表示されます。

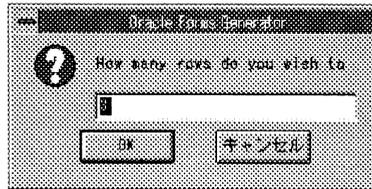


生成処理の間、ジェネレータはこのダイアログ・ボックスに一連のメッセージを表示します。これらのメッセージの一部は、ジェネレータが行っていることを示します。他のメッセージでは、ユーザーの応答が必要になります。

ジェネレータは、取扱フラグが設定されていないことを警告するメッセージを表示します。ジェネレータは、取扱の問合せ、挿入、削除フラグを設定します。このダイアログ・ボックスに表示される特定の警告またはエラー・メッセージに関する詳しい情報を調べるには、生成が完了した後でメッセージテキストをダブルクリックします。

注意：表取扱細目を定義するときに、通常はこれらのフラグをユーザーが設定していることに注意してください。

ジェネレータは、生成するフォームに表示させる情報の行数を指定するよう指示します。



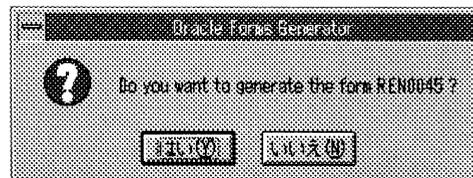
この例では、ジェネレータはページのデフォルトサイズを増やさずに表示できる最大の行数が3であると計算しています。

3. フィールドの値を1に変更して、「OK」をクリックします。

「リポジトリの更新」チェックボックスがチェックされていると、表取扱細目に対して指定する値が保持されます。

メッセージに応答すると、ジェネレータはOracle Formsのソースファイル（接尾辞は.fmb）を生成します。

次に、FormsジェネレータはOracle Formsの実行形式ファイルを生成するか確認します。

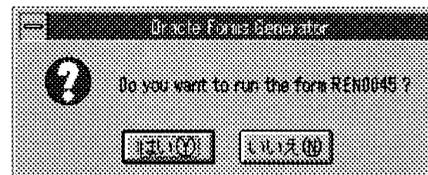


ヒント：「フォームの生成」ダイアログ・ボックスの「コンパイル」タブ上にある「フォーム/メニューのコンパイル」ラジオグループを使用すると、この時点でプロンプトが表示されるかどうかを指定することができます。

4. 「はい」をクリックします。

ジェネレータはOracleFormsの実行形式ファイル（接尾辞は.fmx）を生成します。

FMXファイルを生成すると、生成したフォームを実行できるようになります。



ヒント：「フォーム生成」ダイアログ・ボックスの「実行」タブ上にある「フォームの実行」ラジオグループを使用すると、この時点でプロンプトが表示されるかどうかを指定することができます。

- 「はい」をクリックすると、生成したフォームが実行されます。
下図のように、生成したフォームが表示されます。

レンタル情報の登録 - 仕様: #window1

アクション(A) 編集(E) ブロック(B) フィールド(F) レコード(R) 問合せ(Q) ウィンドウ(W) ヘルプ(H)

レンタル

租税額 借費 メンバーシップ 借組

レンタル状況 備考 レンタル日: 1995/11/28

レンタル番号:

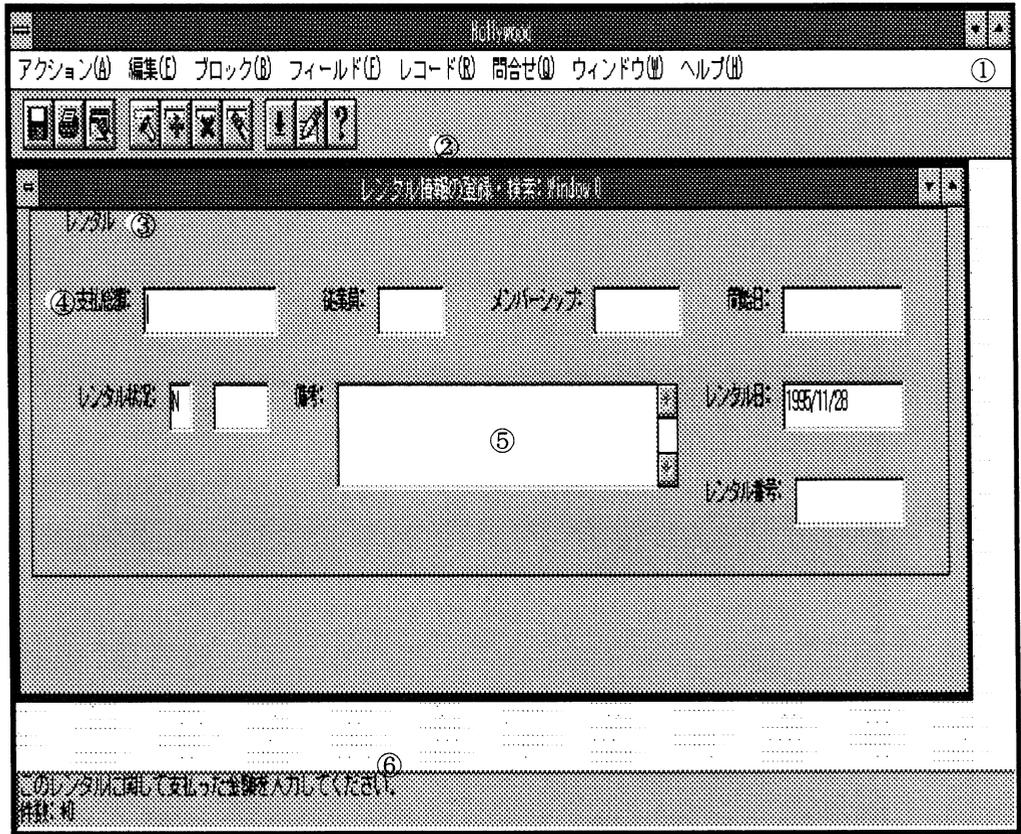
このレンタルに関して支払った金額を入力してください。
仕様: #4

以下の項では、フォームの構成要素のいくつかを詳しく説明します。

- 「アクション→終了」を選択すると、フォームがクローズして、「Oracle Formsジェネレータ」ダイアログ・ボックスに戻ります。
- 「閉じる」をクリックすると、「Oracle Forms ジェネレータ」ダイアログ・ボックスがクローズします。
- 「キャンセル」をクリックすると、「フォームの生成」ダイアログ・ボックスがクローズします。

生成したフォームの機能

前の項では、リポジトリに格納されているデフォルトの情報を使用してブロックが1つあるフォームを作成する方法を示しました。行わなければならないことは、モジュールの表および列取扱細目と、ブロックに表示させる行数を指定することだけでした。



① MDIウィンドウ

Oracle Formsは、ウィンドウ管理のMicrosoft Windows MDI（複数ドキュメントインタフェース）システムを完全にサポートしています。MDIアプリケーションは、アプリケーション・ウィンドウと呼ばれるデフォルトの親ウィンドウを表示します。MDIウィンドウは、ウィンドウ・マネージャによって実行時に作成されるので、モジュール・データ・ダイアグラムでは定義されません。

モジュール・プロパティから生成されるプロパティ

- ② ウィンドウ・タイトル ウィンドウ・タイトルは、ウィンドウ・タイトルとともにモジュールの名前から導き出されます。この場合、モジュールを作成するときにレンタル情報の登録・検索という名前を指定したので、これがウィンドウ・タイトルバーに表示される名前になります。

表取扱細目および列取扱細目から生成されるプロパティ

- ③ ブロック・タイトル ブロックのタイトルは、レンタルの表取扱細目のブロック・タイトルプロパティから導き出されます。レンタル表に基づくブロックのデフォルトのブロック・タイトルは、「レンタル」です。
- ④ 項目プロンプト 各項目の前のプロンプトは、その項目が基いている列取扱細目のプロンプト・プロパティによって決定されます。生成されたフォームの先頭の項目は、「支払総額」列の列取扱細目に基いています。「支払総額」列に基づく項目のデフォルトのプロンプトは「支払総額」です。また、「レンタル日」列に基づく項目のデフォルトのプロンプトは「レンタル日」です。
- ⑤ 項目のサイズ 項目のサイズは、その項目が基づいている列取扱細目の幅プロパティと高さプロパティによって決まります。「備考」項目の幅プロパティと高さプロパティは、それぞれ30および3です。
- ⑥ メッセージ行 メッセージ行には、現在カーソルがある項目を説明するヒントテキストが含まれます。列取扱細目「支払総額」のヒント・プロパティは、「このレンタルについて支払った金額」です。

生成したフォームの他のプロパティ

作成したフォームには、ユーザー作業環境とテンプレート・フォームから導き出されるその他のプロパティがあります。

作業環境の影響の例としては、レンタルブロックが細い実線で囲まれています。枠線のスタイルは、BLKBWD（ブロック ブラシ幅）作業環境とBLKDEC（ブロック装飾）作業環境を使用して変更することができます。テンプレートフォームは、アプリケーション・ウィンドウ・タイトル（Hollywood）がアプリケーション・システム名から導き出されると指定しています。

ユーザー作業環境については、第7章「作業環境を使用しての生成したフォームのカスタマイズ」で詳しく説明します。

テンプレート・フォームの影響の例としては、デフォルト・テンプレート・フォームで定義されているのでツールバーがMDIウィンドウ内で作成されています。

テンプレート・フォームについては、第8章「テンプレートを使用しての生成したフォームのカスタマイズ」で詳しく説明します。

モジュール定義、作業環境、テンプレートの精製とカスタマイズ

この章で説明したように、デフォルト オプションを使用してモジュールからフォームを生成することができます。しかし通常は、特定の要件に適合するようにデフォルトオプションを変更します。このような変更には、次のようなものがあります。

- リポジトリ内のデフォルト モジュール定義（モジュールのデータ取扱）の変更
- デフォルトの作業環境設定の変更
- デフォルト・テンプレートの変更

デフォルトのジェネレータオプションに対するこれらの変更について、以下の章で説明します。

第6章

フォーム モジュール定義の精製

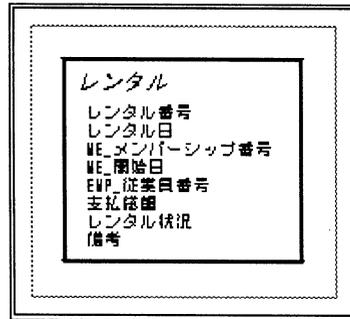
前の章の例では、デフォルト設定を使用してフォームを作成する方法を示しました。この章では、これらのデフォルト設定の一部を変更する方法を説明します。次のような項が含まれています。

- フォーム モジュール定義の精製
- ウィンドウ プロパティの変更
- フォーム上のブロックのカスタマイズ
- フォーム内のフィールドのカスタマイズ
- 複数行のブロックの定義
- ブロック オーバーフロー スタイルの指定

フォーム モジュール定義の精製

前の章の例では、レンタルの表取扱細目とレンタル表内のすべての列の列取扱細目で画面モジュールを作成しました。

モジュールは、下図のようにモジュール・データ・ダイアグラムに表示されました。



モジュールを定義したら、Forms ジェネレータはデフォルト設定を使用してモジュールからフォームを生成しました。生成中に、ジェネレータは問合せ、挿入、削除の取扱を設定し、生成したフォームに表示する行の数（1を指定）を指定するようユーザーに要求しました。

生成したフォームは、次のようになります。

The screenshot shows a window titled "レンタル" (Rental) with a menu bar containing "アクション(A)", "編集(E)", "ブロック(B)", "フィールド(F)", "レコード(R)", "問合せ(Q)", "ウィンドウ(W)", and "ヘルプ(H)". Below the menu bar is a toolbar with icons for save, undo, redo, delete, and help. The main area contains a form with the following fields:

租税額:	<input type="text"/>	借費:	<input type="text"/>	メンバーシップ:	<input type="text"/>	開始:	<input type="text"/>
レンタル状況:	<input type="text" value="N"/>	備考:	<input type="text"/>		レンタル日:	<input type="text" value="1995/11/28"/>	
					レンタル番号:	<input type="text"/>	

このレンタルに関して支払った金額を入力してください。
単位: 円

リポジトリ内の情報を変更すると、生成したフォームの表示を改良して、より使いやすくすることができます。

ウィンドウ プロパティの変更

この例では、生成したフォームが含まれるウィンドウのサイズと位置を変更する方法を示します。

注意：フォームが含まれるウィンドウのサイズと位置は、ウィンドウに表示される最初のブロックの基礎となる表取扱細目のプロパティです。

ウィンドウ タイトルの変更

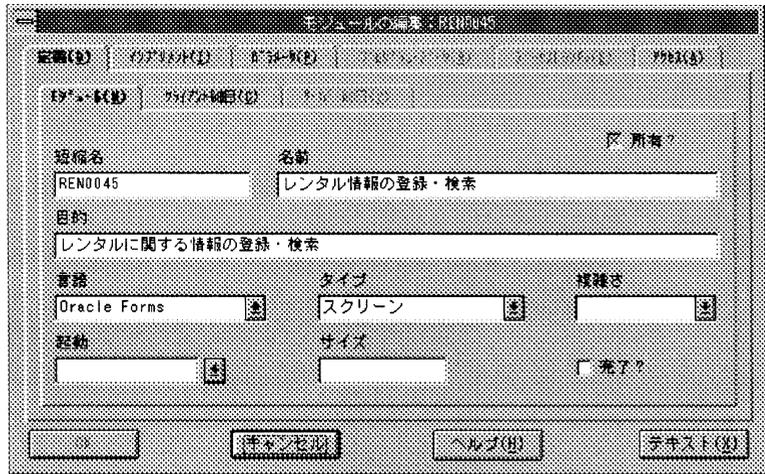
フォームが含まれるウィンドウのタイトルは、モジュール名から導き出されず。モジュール・データ・ダイアグラマに現在表示されているモジュールの名前を変更するには、次のようにします。

1. 「編集→要素→モジュール」を選択して、モジュールのリストが含まれるダイアログ・ボックスを表示させます。



現在のモジュールがすでに選択されています。

2. 「OK」をクリックすると、現在のモジュールのプロパティが含まれている「モジュールの編集」ダイアログ・ボックスが表示されます。

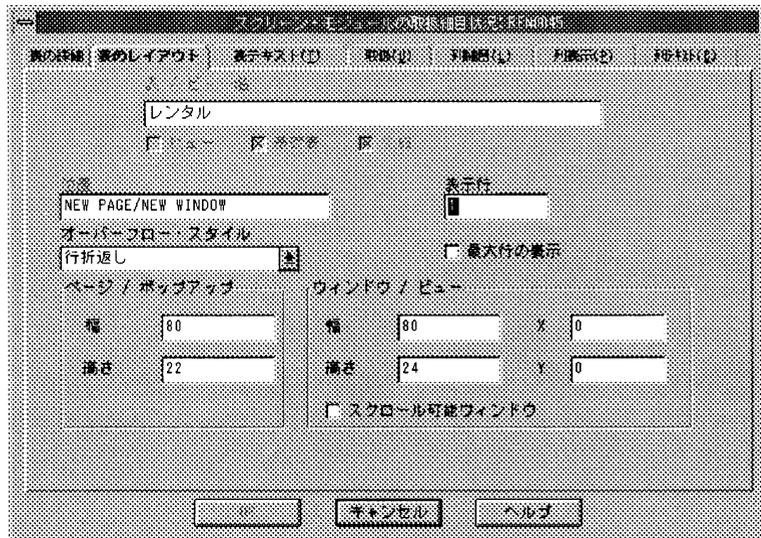


3. 「名前」フィールドの値をレンタル詳細に変更します。
4. 「OK」をクリックすると、「モジュールの編集」ダイアログ・ボックスがクローズします。

ウィンドウのサイズの変更

レンタルブロックが含まれるウィンドウのデフォルトサイズは、作成したフォームを表示するために必要なサイズより大きくなっています。

1. レンタルの表取扱細目を囲む枠をダブルクリックして選択し、「スクリーン・モジュールの取扱細目状況」ダイアログ・ボックスを表示させます。
2. 「表のレイアウト」タブを表示させます。



3. 「ウィンドウ/ビューの高さ」フィールドの値を10に変更します。

ウィンドウの位置の変更

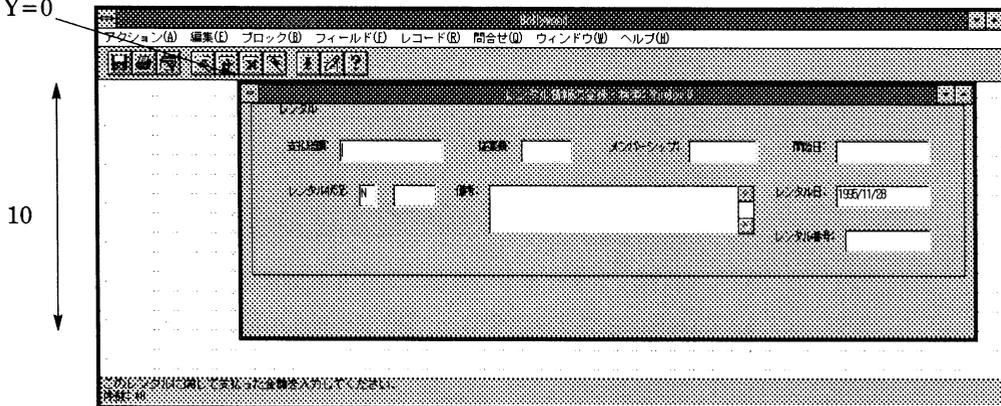
フォームが含まれるウィンドウのデフォルト位置は、MDIウィンドウの左上隅です。フォームが含まれるウィンドウの水平位置を変更するには、次のようにします。

1. 「ウィンドウ/ビュー」グループの「X」フィールドの値を15に変更します。

生成したフォームの機能

モジュールから生成したフォームは、次のようになります。

X=15, Y=0



生成したフォームが含まれるウィンドウを変更した効果は、次のようになります。

- ウィンドウ タイトルが変更された
- ウィンドウの高さは10行
- ウィンドウはMDIウィンドウの左端から離れている

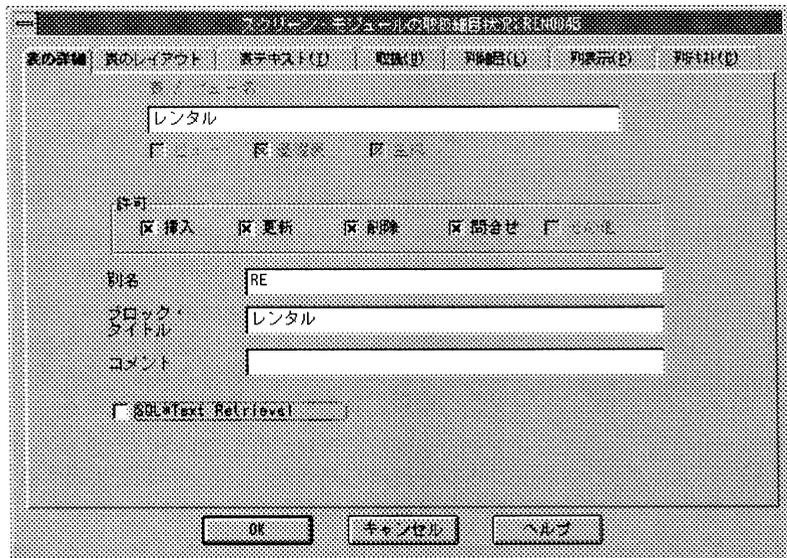
フォーム上のブロックのカスタマイズ

この例では、表取扱細目から作成したブロックのタイトルを変更する方法と、生成したフォームを使用してユーザーがデータベースに対して実行できる操作を変更する方法を示します。

ブロックのタイトルの変更

デフォルトのブロックタイトルは、そのブロックが基づいている表取扱細目の名前から導き出されます。しかし、ブロックのタイトルをより詳しいものにすることもできます。

1. レンタルの表取扱細目を囲む枠をダブルクリックして選択し、「スクリーン・モジュールの取扱細目状況」ダイアログ・ボックスを表示させます。
2. 「テーブルの詳細」タブを表示させます。



3. 「ブロック タイトル」フィールドの値を「レンタルー各レンタルの詳細情報」に変更します。

表取扱フラグの設定

前の章では、作成中に次のメッセージが表示されます。

```
WARNING: Usage of レンタル has no usage flags set;  
query, insert and delete usages set
```

(警告：レンタルに取扱フラグが設定されていません。問合せ、挿入および削除を設定しました)

取扱フラグは、ブロックの基礎となる表内のレコードに対して、生成したフォームのユーザーが実行を許可される操作を決定します。表取扱細目について4つの取扱フラグがあります。

- 挿入
- 更新
- 削除
- 照会

列取扱細目は、表取扱細目の設定から取扱フラグを継承します。

モジュール・データ・ダイアグラムでモジュールを生成して表取扱細目を含めると、デフォルトでは表のレコードに対して操作が許可されなくなります。したがって、表取扱細目について取扱フラグが設定されません。

表取扱フラグのデフォルト設定を変更せずに生成したモジュールからフォームを生成すると、ジェネレータは挿入、削除、問合せの表取扱フラグを設定します。

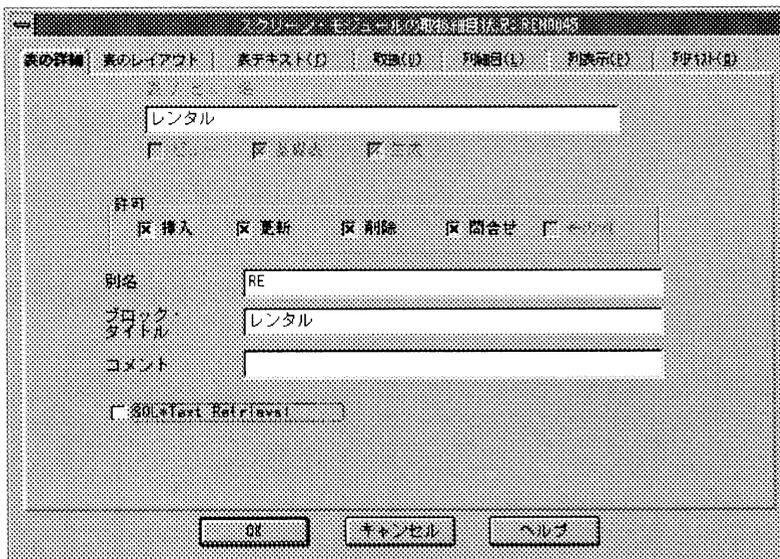
生成したフォーム上の特定の項目を使用してユーザーが実行できる操作をより細かく制御するには、適切な列取扱フラグを設定しなければなりません。特に、ユーザーがレコードを更新できるようにするには、表取扱フラグと適切な列取扱フラグの両方を設定しなければなりません。

フラグを正しく設定していないと（つまり、自動的に設定される列について更新フラグを設定した場合）、フォームを生成するとき、ジェネレータは警告を出してフラグ設定を変更します。

注意：アプリケーション設計ウィザードは、機能/エンティティ マッピングを使用して、生成したフォームのブロックに許可される操作を決定します。アプリケーション設計ウィザードが生成したモジュールからフォームを生成している場合は、おそらく取扱フラグを手作業で設定する必要はないでしょう。

1. レンタルの表取扱細目を囲む枠をダブルクリックして選択し、「スクリーン・モジュールの取扱細目状況」ダイアログ・ボックスを表示させます。

2. 「表の詳細」タブを表示させます。



3. 「更新」チェックボックスをチェックして、ユーザーがレンタル表を更新できるようにします。

次の項では、列取扱細目の取扱フラグを変更します。

フォーム上の項目のカスタマイズ

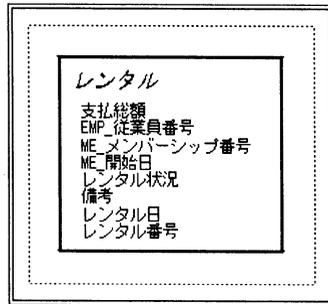
列取扱の順序変更

前に生成したモジュールでは、レンタル表にあるすべての列の取扱細目を指定しました。列の取扱は、アルファベット順に編成されていました。

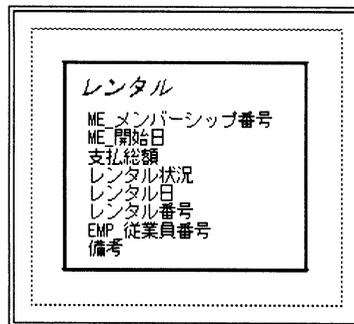
列定義は、表示順にすることもできます。列取扱細目を生成するときに、アルファベット順ではなく表示順にすることができます。

生成したフォームの項目の順番は、列取扱の順番によって決定されます。列取扱の順序を変更して、生成したフォームの項目の順序を変更するには、次のようにします。

1. 列取扱が読みやすくなるように、ダイアグラムの表示を大きくします。



2. 「支払総額」列の取扱をクリックして選択します。
3. 「支払総額」列の取扱をドラッグして、ME_開始日とレンタル状況の間にドロップします。
4. 同じ方法で、下に示す順番になるように列取扱を再編成します。



フィールドプロンプトの変更 通常フォーム内の項目には、項目を何に使用するかを示すプロンプトがあります。生成したフォームの項目は、その基礎となる列取扱細目のプロンプトプロパティからプロンプトを導き出します。特定の列取扱細目のプロンプトを変更して、特定のモジュールの要件に適合させることができます。

1. レンタルの表取扱細目を囲む枠をダブルクリックして選択し、「スクリーン・モジュールの取扱細目状況」ダイアログ・ボックスを表示させます。
2. 「列表示」タブを表示させます。



3. ME_メンバーシップ番号の取扱に関して、「プロンプト」フィールドの値を「メンバーシップ」から「メンバー番号」に変更します。

ポップリストの生成

特定の列について有効な値が比較的小数の場合、対応する列取扱をラジオグループまたはポップリストに指定すると、ユーザーが入力できる値を制限することができます。

レンタル状況列の値は、支払状況ドメインで定義される値のいずれかでなければなりません。支払状況ドメインについては、3つの可能な値が定義されています。

列取扱細目をポップリストとして表示するには、次のようにします。

1. 「スクリーン・モジュールの取扱細目状況」ダイアログ・ボックスの「列表示」タブで、レンタル状況の取扱詳細を表示させます。



2. 「表示データ型」ポップリストから「ポップリスト (略称)」を選択します。
3. 「幅」フィールドの値を10に変更します。

項目グループの生成

多くの場合、生成したフォームの2つ以上の項目を1つの項目グループとして扱うと便利です。グループ内の項目は、それぞれに対してレイアウトされます。項目グループは、1つの項目の場合と同じように、生成したフォームに配置されます。項目グループには、独自の枠とプロンプトを与えることができます。

項目グループを使用すると、生成したフォームで関連する情報が互いに保持され、最も意味がある方法で与えられること保証されます。

1. 「スクリーン・モジュールの取扱細目状況」ダイアログ・ボックスの「列表示」タブで、ME_メンバーシップ番号の取扱詳細を表示させます。



2. 「項目グループ」の「名前」フィールドに、「メンバー詳細情報」と入力します。
3. 「項目グループ」の「プロンプト」フィールドに、「メンバーシップ詳細」と入力します。
4. 「列取扱」フィールドのME_開始日をクリックします。
5. 「項目グループ」の名前ドロップダウン・リストから、「メンバー詳細情報」を選択します。

これで、ME_メンバーシップ番号とME_開始日が同じ項目グループに属するようになり、適切にレイアウトされます。

列取扱フラグの設定

生成したフォームを使用してユーザーがデータベースのレコードに対して実行できる操作は、表取扱および列取扱の取扱フラグによって異なります。

デフォルトでは、列取扱フラグは表取扱細目の取扱フラグの設定を継承します。

前の章では、ジェネレータは表取扱細目について挿入、削除、問合せの取扱フラグを設定しました。列取扱細目については、表示、挿入、選択の取扱フラグを設定しました。唯一の例外は、レンタル番号の列取扱細目です。この列は順番に設定されるので、更新取扱は不適当です。任意入力取扱フラグも、任意入力として定義された列について設定されています。

以下の例では、取扱フラグのデフォルト設定を変更する方法を示します。

注意：アプリケーション設計ウィザードは、機能/エンティティ マッピングを使用して、生成したフォームのブロックに許可される操作を決定します。アプリケーション設計ウィザードが生成したモジュールからフォームを生成している場合は、おそらく取扱フラグを手作業で設定する必要はないでしょう。

1. 「スクリーン・モジュールの取扱細目状況」ダイアログ・ボックスを表示させて、「取扱」タブをクリックします。



2. 下図のように、表示、挿入、更新、選択の各フラグを設定します。

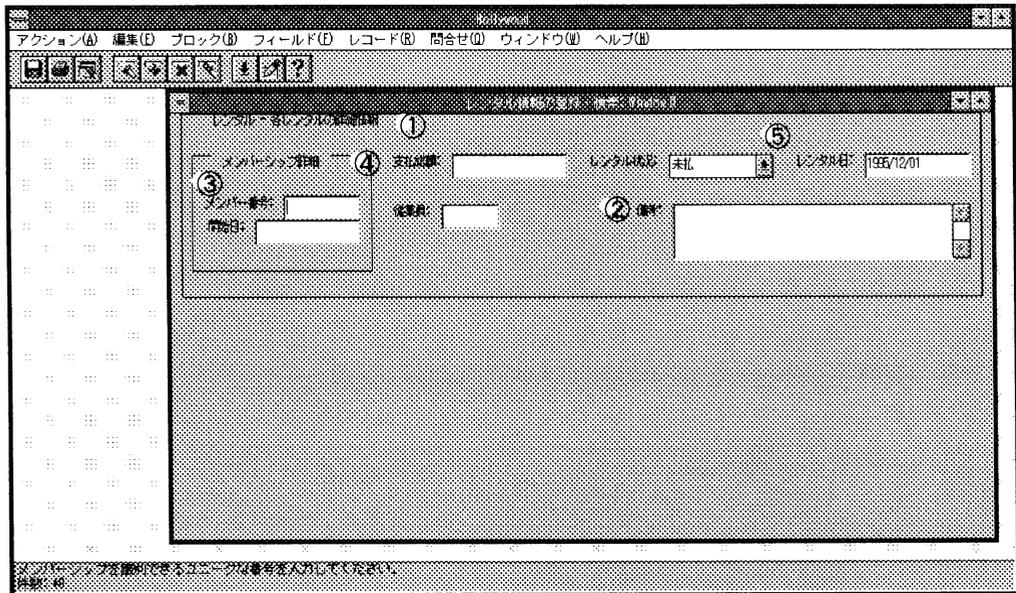
	表示	挿入	更新	選択
ME_メンバーシップ番号	✓	✓	✓	✓
ME_開始日	✓	✓	✓	✓
支払総額	✓	✓	✓	✓
レンタル状況	✓	✓	✓	✓
レンタル日	✓	✓	✓	✓
レンタル番号				
EMP_従業員番号	✓	✓		✓
備考	✓	✓	✓	✓

ユーザーとの打ち合わせから、生成するフォームにはレンタル番号が不要であることがわかりました。したがって、レンタル番号の表示取扱フラグの設定を解除します。表示取扱フラグの設定を解除すると、他の取扱フラグの設定がすべて解除されます。レンタル番号はレンタル表の主キーであり、存在しないとユーザーが新しいレコードを挿入できないので、この列は表示させずに残します。

従業員番号の更新取扱フラグは、従業員表の外部キーなので設定されていません。レンタル表と従業員表間のリレーションシップは、転送不可と定義されています。このようにすると、トランザクションの詳細を元々入力していた従業員の従業員番号を変更できなくなります。

生成したフォームの機能

モジュールから生成したフォームは、次のようになります。



- ① **ブロック タイトル** ブロックタイトルは、モジュール名に「-各レンタルの詳細」というテキストを追加したので変更されました。
- ② **項目の順序** 項目の順序は、列取扱細目の順序を変更したので変更されました。
- ③ **項目プロンプト** ME_メンバーシップ番号の列取扱細目に基づく項目のプロンプトは、メンバー番号に変更されました。

④ 項目グループ

ME_メンバーシップ番号とME_開始日に基づく項目は、項目グループとしてグループ化されました。項目グループには独自の枠とタイトルがあり、その内部の項目は右揃えで垂直にレイアウトされます。ブロック内の他の項目は、項目グループの周りに配置されます。

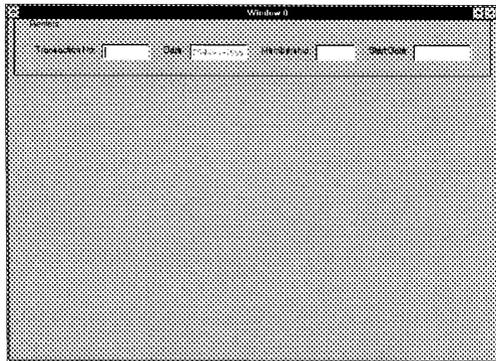
⑤ ポップリスト

レンタル状況に基づく項目の有効な値を示すために、ポップリストが生成されました。

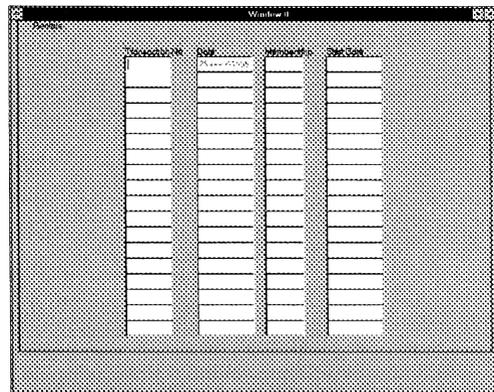
複数行表示ブロックの定義

これまでの例では、生成するブロックには1つの行が含まれていました。1つの行が含まれるブロックを、単一行表示ブロックと呼びます。

次の2つのフォームは、同じモジュールから生成したものです。左側のフォームを生成するときは、必要な行の数として1を指定しました。右側のフォームを生成するときは、必要な行の数として18を指定しました。

A screenshot of a software window titled "Window 0". It contains a form with four input fields labeled "Transaction ID", "Date", "Amount", and "Status". Each field has a small "OK" button to its right. The form is a single horizontal line.

単一行表示ブロック

A screenshot of a software window titled "Window 0". It contains a multi-line form with four columns labeled "Transaction ID", "Date", "Amount", and "Status". Each column has 18 rows of input fields. The form is a grid of 18 rows and 4 columns.

複数行表示ブロック

2つ以上の行が含まれるブロックを、複数行表示ブロックと呼びます。ジェネレータは、単一行表示ブロックと複数行表示ブロックを別々に扱うことができます。例えば、複数行表示ブロックのプロンプトは項目の上に、単一行表示ブロックでは項目の横に表示させることができます。

ブロック オーバーフロー スタイルの指定

多くの場合、項目の数が多いため1行では表示できなくなります。ジェネレータは、残りの項目（オーバーフロー項目）を配置する位置を、「スクリーン・モジュールの取扱細目状況」ダイアログ・ボックスの「表のレイアウト」タブで指定するオーバーフロー スタイルに基づいて決定します。

デフォルトのオーバーフロースタイルは「行折り返し」で、オーバーフロー項目は次の行に配置されます。これまでの例では、生成したフォームにこのオーバーフロー スタイルを使用していました。

あるいは、1行に表示させる項目（コンテキスト項目）とオーバーフロー領域に配置させる項目（オーバーフロー項目）を指定することもできます。一般に、コンテキスト項目は基礎表の主キーから導き出されるか、基礎表の記述子としてマークされます。

複数行表示ブロックのオーバーフロー領域には、現在のコンテキスト項目のレコードが含まれます。

注意：コンテキスト項目用のスペースに収まらないコンテキスト項目は、オーバーフロー領域に移動されます。オーバーフロースタイルが右にオーバーフローまたはスプレッドテーブルの場合、コンテキスト項目と関連するオーバーフロー領域の間にあるページ上で使用可能なスペースの分割は、OFAPCN（コンテキストに予約した利用可能な幅のオーバーフロー領域のパーセンテージ）作業環境によって制御されます。

行折り返し
オーバーフロー スタイル

「行折り返し」は、デフォルトのオーバーフロースタイルです。オーバーフロー項目は、次の行に配置されます。

単一行表示ブロック

複数行表示ブロック

複数行表示ブロックのデフォルトではフィールドの上にプロンプトが配置されるので、1行のフィールドよりも多くのスペースがあります。しかし、上の例では、1行に収まらない項目が1つあります。

右にオーバーフロー
オーバーフロー スタイル

オーバーフロースタイルを「右にオーバーフロー」に設定すると、コンテキスト項目の右にオーバーフロー領域が表示されます。

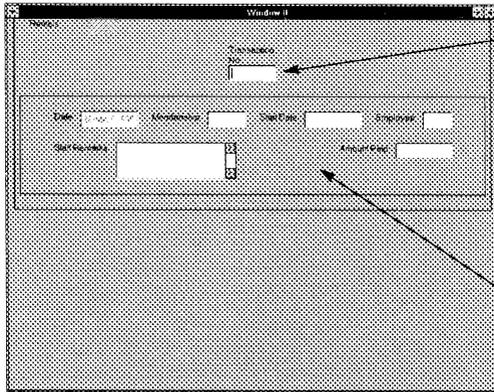
単一行表示ブロック

複数行表示ブロック

複数行表示ブロックのオーバーフロー領域には、現在のコンテキスト項目のレコードが含まれます。

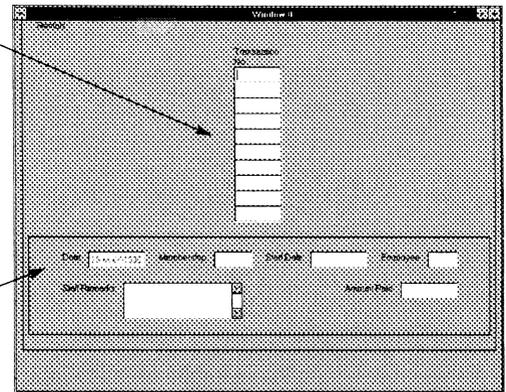
下にオーバーフロー
オーバーフロースタイル

オーバーフロースタイルを「下にオーバーフロー」に設定すると、コンテキスト項目の下にオーバーフロー領域が表示されます。



コンテキスト項目

オーバーフロー領域



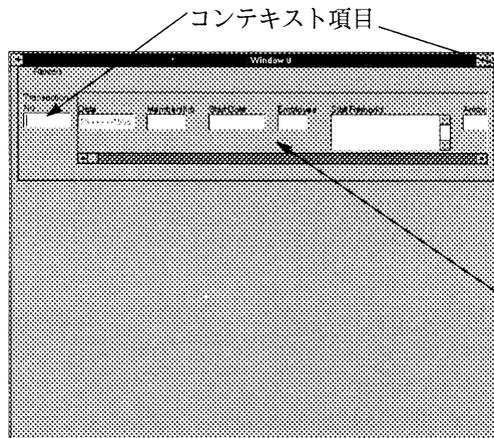
単一行表示ブロック

複数行表示ブロック

複数行表示ブロックのオーバーフロー領域には、現在のコンテキスト項目のレコードが含まれます。

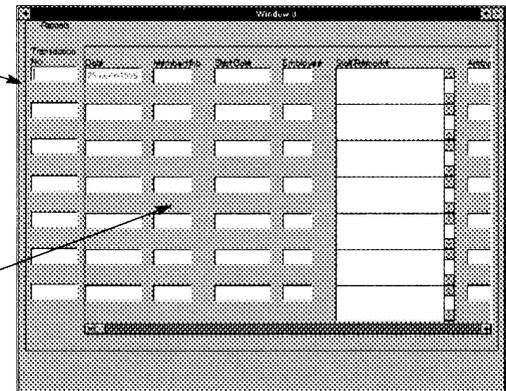
スプレッドテーブル
オーバーフロースタイル

オーバーフロースタイルを「スプレッドテーブル」に設定すると、オーバーフロー領域はスプレッドテーブルとして表示され、コンテキスト項目の右に配置されます。



コンテキスト項目

スプレッドテーブル



単一行表示ブロック

複数行表示ブロック

第7章

作業環境を使用しての生成 フォームのカスタマイズ

この章では、作業環境を使って生成されたフォームの外観と動作を変更する方法について説明します。この章は次の節からできています。

- 作業環境の紹介
- ジェネレータ作業環境のカテゴリ
- レイアウト作業環境の設定と変更
- 生成されたフォームの機能

作業環境の紹介

作業環境とは何か？

作業環境は、生成されるフォームの外観と動作を制御するのに使用されるパラメータです。たとえば、ブロック・デコレーション作業環境（BLKDEC）は、生成されるフォーム上のブロックを囲むのに使用される線のスタイルを決めます。

すべての作業環境にはデフォルト値（出荷時設定）があり、最初に作業環境を何も設定しなくてもフォームを生成することができます。しかし、特定の結果を得るために作業環境を明示的に設定することができます。

作業環境のレベル

作業環境を設定するレベルによって、変更がアプリケーション内のすべてのモジュール、1つのモジュール、モジュールのコンポーネント（例、表取扱細目）のいずれに影響を与えるかが決まります。

生成の際、ジェネレータは最もローカルなレベルから順に、厳密な優先順位で作業環境を処理します。たとえば、作業環境がモジュールとアプリケーション・システムの両方のレベルで設定された場合、ジェネレータは常にモジュールレベルの作業環境を使用します。作業環境があるレベルで設定されていることが見つかった場合、ジェネレータはその値を使用します。同じ作業環境の検索をより高いレベルで続けることはありません。

作業環境がどのレベルでも設定されていない場合、ジェネレータは出荷時設定値を使用します。

作業環境の使用に関するガイドライン

作業環境の使用に関するガイドラインがいくつかあります。

- 作業環境は適用可能な最高レベルで設定します。

作業環境をアプリケーション・システムに対してグローバルに適用したい場合、作業環境をアプリケーション・システム・レベルで設定します。

- 名前付きの作業環境設定を使用します。

作業環境設定を使用すると、複数の作業環境を1つの場所で保守することができます。名前付き設定は、作業環境階層の任意の適切なレベルで参照できます。

- あるレベルで設定した作業環境のレベルをより低いレベルでも設定することによって不注意に上書きしないように留意します。

たとえば、オブジェクト（レンタル表）とレンタルを使用するモジュールに関する作業環境を設定する場合、モジュールレベルの作業環境は常にオブジェクトレベルの作業環境値の上に上書きされます。これは、意図した結果ではないでしょう。

- 作業環境の設定を変更する方法** 作業環境の設定を変更することによって生成されるモジュールの外観と動作を制御するには、プリファレンス・ナビゲータと呼ばれる Designer/2000 ツールを使用します。
- モジュール仕様が完全に定義されていない場合、プリファレンス・ナビゲータを使ってプロトタイプのために作業環境を設定することができます。データベース設計が完全で、各モジュール仕様が完全に定義されている場合、プリファレンス・ナビゲータを使って作業環境を設定することもできます。
- プリファレンス・ナビゲータの起動** Designer/2000 プログラム・グループの中の適切なアイコンをダブルクリックすることによってプリファレンス・ナビゲータを起動します。モジュール・データ取扱を定義するためにモジュール・データ・ダイアグラムを使用している場合、ツール→プリファレンス・ナビゲータを選択することによってプリファレンス・ナビゲータを起動することができます。

ジェネレータ作業環境のカテゴリ

関連する作業環境をカテゴリにグループ化することができます。Formsジェネレータに適用される作業環境のカテゴリを次に示します。

CODING STYLE作業環境	これらの作業環境を使用すると、生成されるコードのスタイルを制御してプロジェクト標準に準拠させることができます。
COMMENTING作業環境	これらの作業環境を使用すると、生成されるフォームの中のブロック、フィールド、トリガーに追加されるコメントのレベルを制御することができます。
DBA作業環境	これらの作業環境は、オンライン参照表の命名などのデータベースの管理と制御を取り扱います。
DESCRIPTIVE FLEX作業環境	これらの作業環境は、Oracleアプリケーションの付加フレックス・フィールドの生成を制御します。これらの作業環境を使用するには、次のことが必要です。 <ul style="list-style-type: none">• Oracle アプリケーション リリース10Gを用意します。• Application Object Library (AOL) ライブラリが連結されたテンプレートを作成します。
END USER INTERFACE作業環境	これらの作業環境は、生成されるフォームがエンド・ユーザーによって実行されたときにその動作を制御するのに使用します。
ENVIRONMENT作業環境	これらの作業環境を使用すると、キャラクタ・モード環境とグラフィカル環境のためのシステムを開発することができます。
FORM AND MENU ATTACHMENT作業環境	これらの作業環境を使用すると、フォーム生成時にメニューが生成されなかったフォームに対するメニューの結合を制御することができます。
GENERATE OPTIONS作業環境	これらの作業環境を使用すると、生成されるコードの内容を制御することができます。
GENERATOR RUNTIME作業環境	これらの作業環境を使用すると、生成プロセスの際にジェネレータの動作を制御することができます。ほとんどの作業環境では、ジェネレータが尋ねる質問に対する答えを事前定義できます。
JOURNALLING作業環境	これらの作業環境を使用すると、ユーザー／作成日付／最終変更情報を識別するためにジェネレータによって追加されたコードを制御し、ジャーナル表の中のユーザー・データの変更を記録することができます。
LAYOUT作業環境	これらの作業環境を使用すると、ボイラープレートと線幅に関するクリア・パターンの生成を制御できます。

LAYOUT作業環境 –
<Various>

各種のインタフェース・オブジェクトのレイアウトを制御するのに使用する複数の作業環境グループ。次のものがあります。

Block	Image	Radio Group
Button	Item	Single Area
Canvas	Item Group	Spread Table
Checkbox	Over flow	Stub
Combo Box	Page	Text Item
Context	Poplist	Text List
Custom Item	Popup	Window

LIST OF VALUES作業環境

これらの作業環境を使用すると、有効な値のリストをインプリメントすることによって、既知の値と範囲の列に関する妥当性チェックを制御することができます。

MENU作業環境 –
<Various>

メニューを生成するための作業環境が入った複数の作業環境グループ。次のものがあります。

End User	Generator Runtime	Templates
Generation Options	Roles	

REGENERATION作業環境

これらの作業環境を使用すると、ジェネレータが再生成プロセス中に自身のコードを変更するのを防ぐことができます。

REVERSE ENGINEERING作業環境

これらの作業環境を使用すると、既存のフォームをリバース・エンジニアリングするときに作成されたモジュール定義を制御することができます。

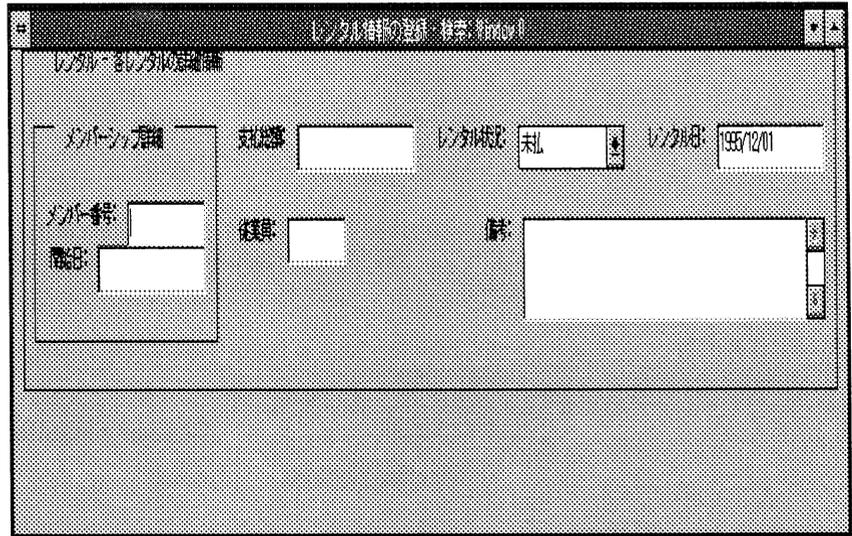
TEMPLATE FORMS作業環境

これらの作業環境を使用すると、フォーム生成時にどのテンプレート・フォームを使用するかを指定することができます。

レイアウト作業環境の設定と変更

この節では、プリファレンス・ナビゲータを使って生成されるフォームのレイアウトを制御する作業環境の一部を設定および変更する方法を示します。

次に示すようなフォームを生成したとします。



このフォームの外観と動作を改善するために、いくつかのレイアウト作業環境を設定することができます。

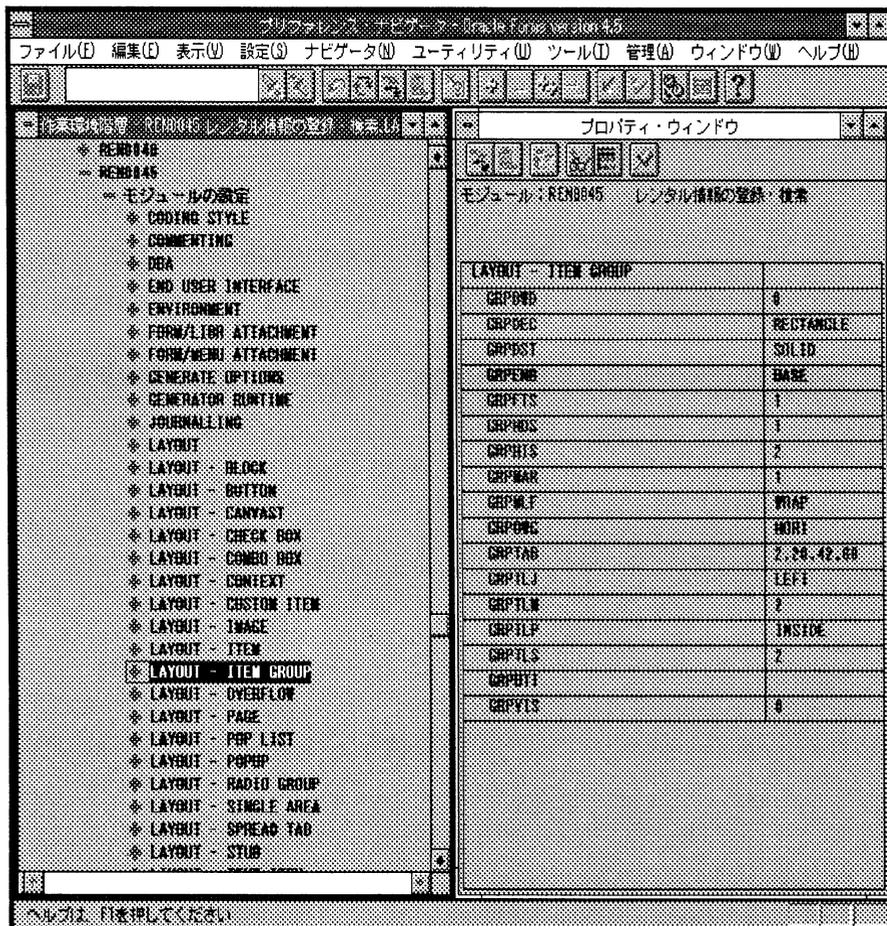
モジュール・レベルによる項目グループ内の方向の変更

デフォルトでは、項目グループ内の項目は垂直にレイアウトされます。項目グループ内の項目のレイアウトは、GRPOWG（グループ内の項目グループの方向）作業環境によって制御されます。

個々のモジュールに関して生成されるすべての項目グループの中で項目が水平にレイアウトされるようにしたいとします。作業環境をモジュール・レベルで設定しなければなりません。

1. 次のいずれかの方法でプリファレンス・ナビゲータを起動します。
 - Designer/2000プログラムの中で適切なアイコンをダブルクリックします。
 - モジュール・データ取扱を定義するためにモジュール・データ・ダイアグラムを使用している場合、ツール→プリファレンス・ナビゲータを選択します。
2. 現在のアプリケーション・システムにあるモジュールを見るために、「モジュール」をダブルクリックして作業環境階層ウィンドウの中のツリーを拡張します。
3. 作業環境を変更したいモジュールをダブルクリックします。

4. モジュール・レベルで設定された作業環境を表示するために「モジュールの設定」をダブルクリックします。
5. このカテゴリにある作業環境の現在の設定をプロパティ・ウィンドウに表示するために「LAYOUT-ITEM GROUP」作業環境グループをダブルクリックします。



GRPOWG（グループ内の項目グループの方向）作業環境によって、項目グループ内の項目のレイアウトが決まります。この作業環境の出荷時設定は VERT（垂直）です。

6. GRPOWGをクリックして選択します。

GRPOWG作業環境の現在の設定が作業環境設定フィールドに表示されます。

7. 作業環境設定ドロップダウン・リストからHORIを選択します。

項目グループ・レベルにおける項目グループ・ブラシ幅の変更

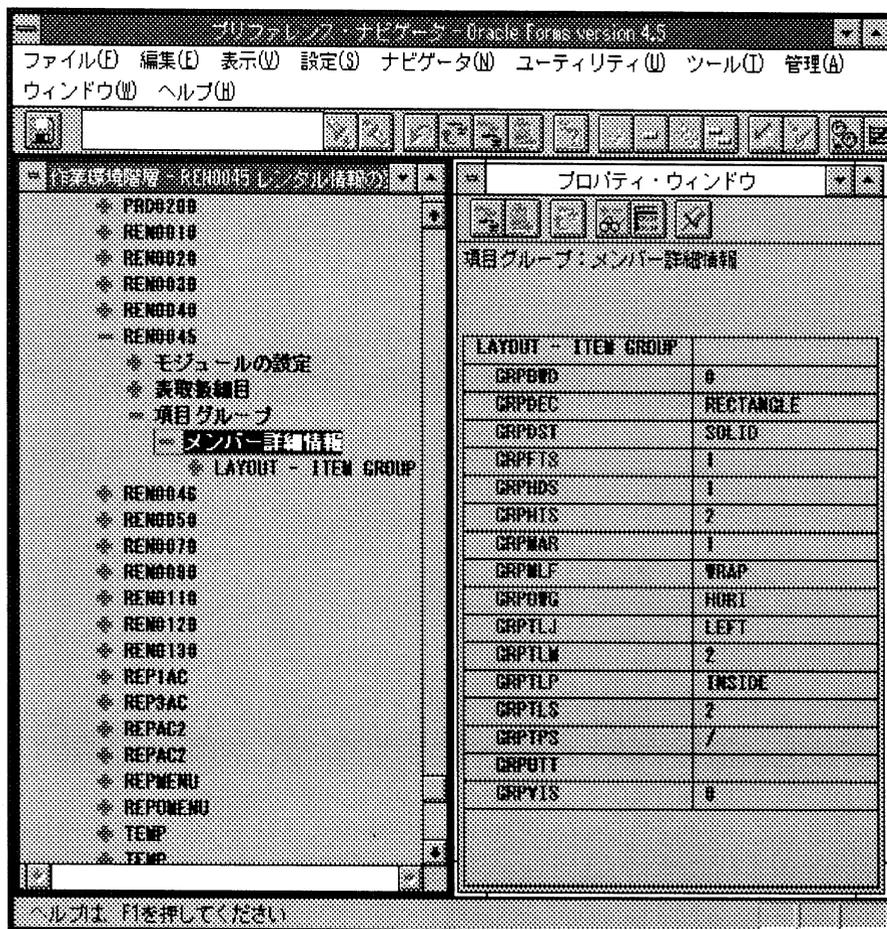
デフォルトでは、項目グループを囲む枠は可能な最も細い線（0と見なされる）です。項目グループ枠の幅は、GRPBWD（項目グループ・ブラシ幅）作業環境によって制御されます。

モジュールの中の特定の項目グループの枠を太くしたいとします。これには、GRPBWD作業環境を項目レベルで設定します。

1. 作業環境階層内のツリーが現在作業中のモジュールに関する作業環境レベルを示している状態で、このモジュール内で定義された項目グループを表示するために「項目グループ」をダブルクリックします。
2. 枠幅を設定したい項目グループをダブルクリックします。

前の章の実際の例では、メンバー詳細項目グループが定義されています。

利用可能なのは、「LAYOUT - ITEM GROUP」という作業環境カテゴリ1つだけです。このカテゴリ内の作業環境がプロパティ・ウィンドウに表示されます。



GRPBWD作業環境によって、項目グループを囲む枠の幅が決まります。この作業環境の出荷時設定は0で、最も細い線幅を示します。

3. GRPBWDを選択するためにクリックします。

GRPBWD作業環境の現在の設定が作業環境設定フィールドに表示されます。

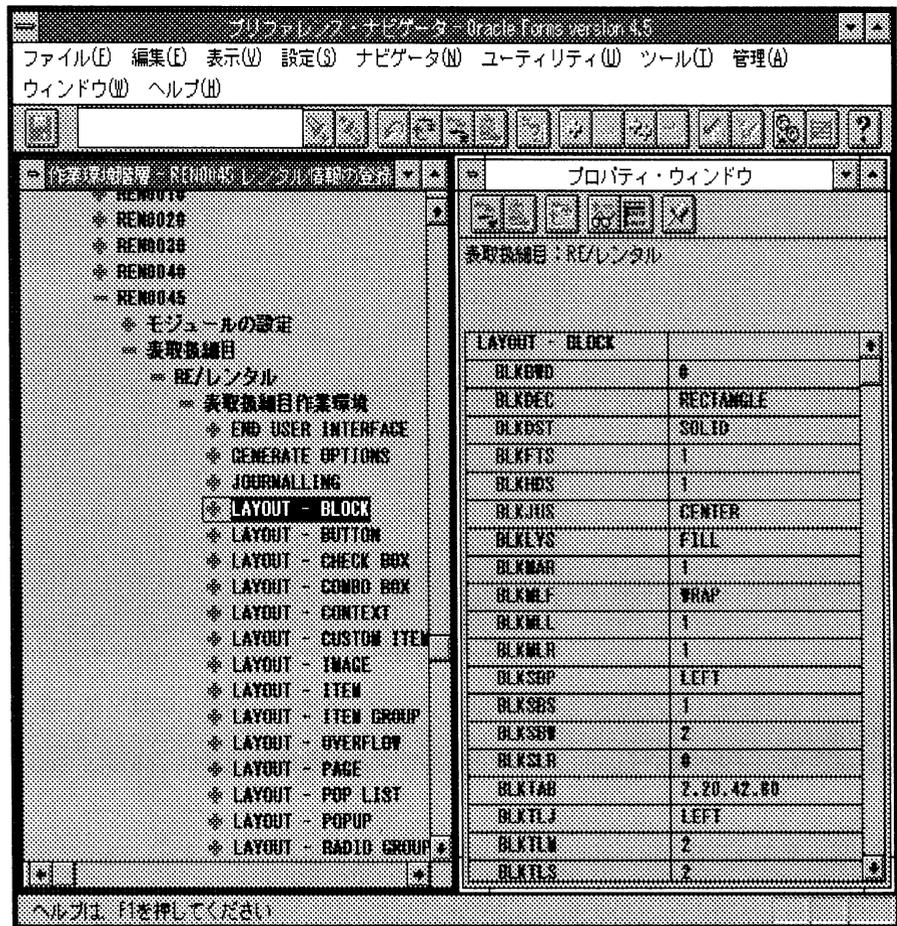
4. 作業環境設定ドロップ・ダウン・リストから2を選択します。

ブロック・デコレーションと
タイトル文字位置の変更

デフォルトでは、ブロックは枠としての1本の線を持っており、ブロック・タイトルは左揃えになっています。

特定の表取扱細目に基づいてブロック・タイトルを中央揃えにし、ブロックをレベル表示にしたいとします。

1. 作業環境階層のツリーが現在作業中のモジュールに関する作業環境レベルを示している状態で、このモジュールに関する表取扱細目を表示するために「表取扱細目」をダブルクリックします。
2. ブロック・タイトルの文字位置を設定したい表取扱細目をダブルクリックします。
3. このレベルで設定できる作業環境カテゴリを見るために「表取扱細目環境」をダブルクリックしてください。
4. このカテゴリにある作業環境の現在の設定をプロパティ・ウィンドウに表示するために「LAYOUT-BLOCK」作業環境グループをクリックします。



BLKBWD（ブロック・ブラシ幅）作業環境によって、ブロックを囲む枠の幅が決まります。

5. BLKBWDの設定を4に変更します。

BLKDEC（ブロック・デコレーション）作業環境によって、ブロックの回りに描かれる枠のスタイルが決まります。

6. BLKDECの設定をRAISED RECTANGLEに変更します。

BLKTLJ（ブロック・タイトル文字位置）作業環境によって、ブロック・タイトルの位置が決まります。この作業環境の出荷時設定はLEFTです。

7. BLKTLJ作業環境の設定をCENTERに変更します。
8. 作業環境の設定に施した変更を保存するためにファイル→保存を選択します。

生成されたフォームの機能

モジュールを生成すると次に示すようなフォームができあがります。

The screenshot shows a form window titled 'BLKTLJ作業環境' (BLKTLJ Work Environment). The form contains several input fields and labels. Callout 1 points to the title 'BLKTLJ作業環境'. Callout 2 points to the 'メンバー番号' (Member Number) and '所属' (Affiliation) fields. Callout 3 points to the '項目' (Item) dropdown menu. Callout 4 points to the main content area of the form.

- | | |
|------------------|---|
| ① ブロック・タイトルの文字位置 | BLKTLJ作業環境を変更した結果、ブロック・タイトルは中央揃えになります。 |
| ② 項目グループ内の方向 | GRPOWG作業環境をHORIに変更した結果、項目グループ内の方向は水平になります。 |
| ③ 項目グループ枠 | GRPBWD作業環境を4に変更した結果、項目グループの枠は4ポイント幅になります。 |
| ④ ブロック枠 | BLKDECとBLKBWDの作業環境をそれぞれ設定することによって、ブロック枠のスタイルと太さが変化しました。 |

第8章

テンプレートを使用しての生成フォームのカスタマイズ

この章では、テンプレートがどのように生成されたフォームに影響を与えるかを説明します。この章は次の節からできています。

- テンプレート・フォームの紹介
- 使用するテンプレートの指定
- テンプレートの作成
- バックドロップとしてのポイラープレートの作成
- ジェネレータ・オブジェクトの作成
- ユーザー・オブジェクトの作成
- テンプレートフォーム内のプロシージャを呼び出すためのボタン項目の生成

テンプレート・フォームの紹介

テンプレート・フォームとは何か？

テンプレート・フォームは、Formsジェネレータが生成プロセス中に使用する既存のフォームです。テンプレート・フォームは、一般に自分のフォームに関する標準要件であるオブジェクトを定義するのに使用します。その情報をリポジトリ・モジュール定義の一部として指定する必要はありません。

生成されるフォームは、テンプレート・フォームから豊富な可視要素（ボタンやデコレーションなど）、プロパティ（メニュー・スタイルなど）、コード（イベント・トリガー）を継承および再利用できます。他の例には、標準のブロック、ツールバー、可視属性、カスタム項目があります。したがって、標準テンプレート・フォームのセットを使用することは、ジェネレータの異なる実行によって生成されるフォームに関して首尾一貫したルック・アンド・フィールを保証するための戦略の一部となりえます。

ジェネレータは、フォームを生成するときに使用できるいくつかの標準テンプレートといっしょにインストールされます。これらテンプレートは、そのまま使用するか、Oracle Forms Designerを使って自分の要件に合わせてカスタマイズすることができます。必要に応じて、独自のテンプレート・フォームを作成することもできます。

テンプレート・フォームによる生成はオプションですが、テンプレートには、あまりに多くの機能があるので、おそらく常に1つを指定することになるでしょう。デフォルトでは、ジェネレータは付属のテンプレートの1つを使用します。

テンプレート・フォームには2種類のオブジェクトがあります。

- ユーザー・オブジェクト
- ジェネレータ・オブジェクト

ジェネレータは、オブジェクトの名前に基づいてオブジェクトがユーザー・オブジェクトかジェネレータ・オブジェクトかを判断します。CG\$で始まり、認識名が続くオブジェクト名はジェネレータ・オブジェクトで、そうでないオブジェクトはユーザー・オブジェクトです。

ユーザー・オブジェクト ジェネレータは、テンプレート・フォームのユーザー・オブジェクトを生成されたフォームに変更せずにコピーします。

ジェネレータ・オブジェクト ジェネレータは、テンプレート・フォームの中のジェネレータ・オブジェクトを使って次のことを行います。

- 生成されたオブジェクトの属性を設定します。
- 生成されたフォームの中に特殊オブジェクトを作成します。

ジェネレータ・オブジェクトの名前は、CG\$で始まり、認識名が続かなければなりません。

付属のテンプレート ジェネレータには、Microsoft Windows環境専用の2つの標準テンプレートが付属しています。

標準テンプレートには、次の標準機能があります。

- アイコン・ボタンとコンテキスト・ヘルプ付きのツールバー
- ボタンを使ったモジュール間のナビゲーションを可能にするためのCG\$FF1の使用
- Microsoft Windows環境のための可視属性
- アプリケーション・タイトル・バーの動的設定
- 実座標の使用

なぜテンプレートを使用するか？

テンプレート・フォームを使用すると、ジェネレータの異なる実行によって生成されるフォームに関して首尾一貫したルック・アンド・フィールを保証することができます。テンプレート・フォームを使用すると、次のようなさまざまなオブジェクトやプロパティを事前定義することができます。

- 座標システムと文字セル・サイズ
- ヘッダーとフッター
- 生成されるオブジェクトに適用するための名前付きの可視属性
- ツールバー
- キャンパス上の会社のロゴ
- 問合せ実行(Execute Query)などのForms機能を実行するためのボタン
- 制御ブロック
- VBXコントロール
- OLEコンテナ
- フォーム・トリガーとプログラム単位
- ライブラリ連結

トリガーの実行スタイル

Formsジェネレータは、トリガーが単に上書きするのではなく、フォームの中のより高いレベルの同じ名前付きトリガーの前後で実行できるようにOracle Forms Version 4.5の機能をサポートしています。

これによって、ブロックまたは項目レベルで実行される生成されたトリガーの前後で実行されるテンプレート内のモジュール・レベル・トリガーを組み込むことができます。

使用するテンプレートの指定

生成時にジェネレータは、STFFMB（テンプレート・フォームの名前）作業環境によって指定されたテンプレートを使用します。この章で施された変更は、Formsジェネレータに付属の標準テンプレートの1つに施されています。

STFFMB作業環境の変更

Formsジェネレータが生成時に使用するテンプレート・フォームを変更するには、

1. プリファレンス・ナビゲータを起動します。
2. そこから生成するモジュールに関するモジュール作業環境をオープンします。
3. STFFMB作業環境の設定を新しく使用するテンプレート・フォームの名前に変更します。

ヒント: STFFMB作業環境はテンプレート作業環境の1つです。

この章の例では、STFFMB作業環境はtplate1.fmbに設定されています。

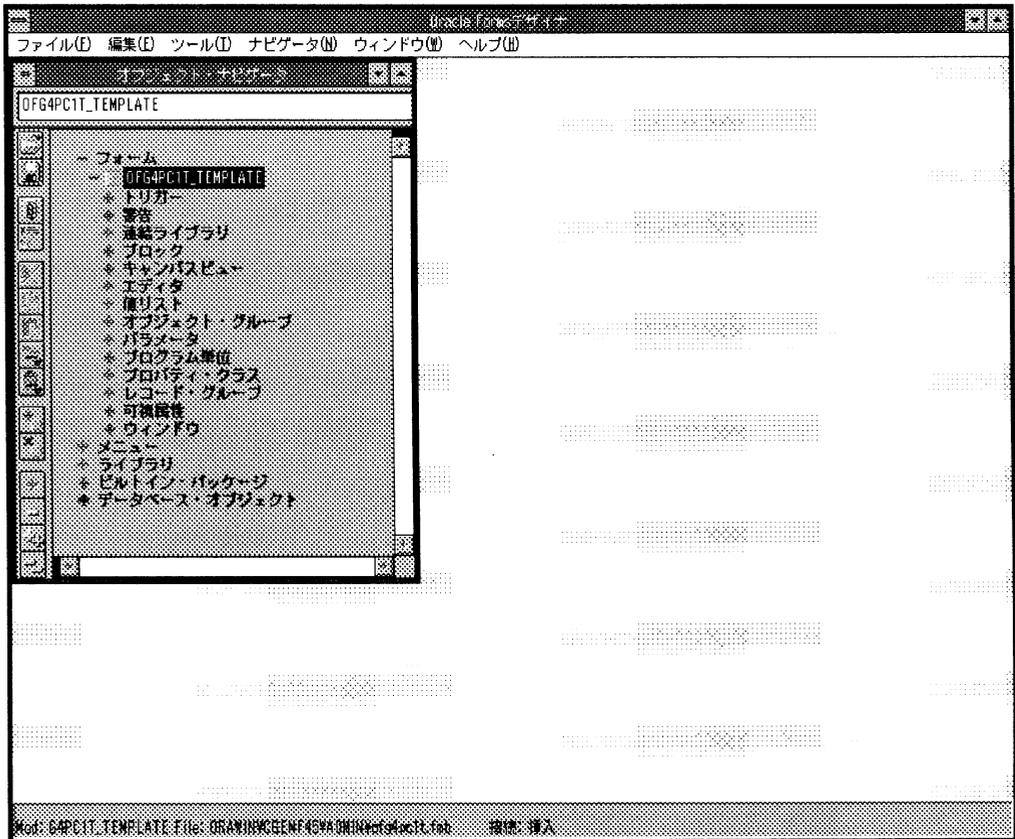
テンプレート・フォームの位置の指定

テンプレート・フォームの位置を指定するには、「フォームの生成」ダイアログ・ボックスの「テンプレート」タブに関連する細目を入力します。

テンプレートの作成

標準テンプレートのコピー テンプレートは初めから作成するより付属のテンプレートの1つをコピーする方が速いことがあります。

1. モジュール・データ・ダイアグラマが表示された状態で、Oracle Forms Designerをオープンするためにツール→Oracle Forms Designerを選択します。
2. ファイル→オープンを選択し、ofgpc1t.fmbテンプレートをオープンします。
このファイルの位置はインストレーションによって異なります。通常はC:\ORAWIN\CGENF45\ADMINにあります。



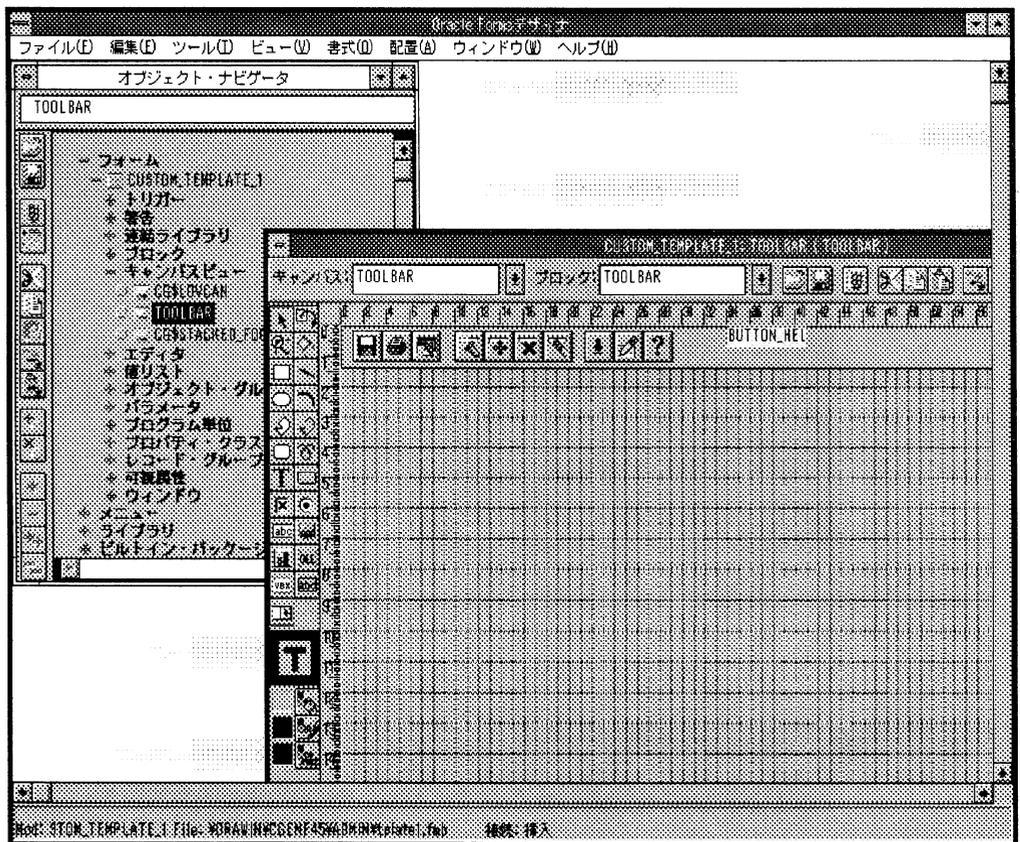
3. 付属のテンプレートは変更しないでください。ファイル→別名保存を選択し、付属のテンプレートofg4pc1t.fmbを別の名前（例、tplate1.fmb）で保存することによって、自分のコピーを作成します。

注意: 付属のテンプレートは変更しないことを推奨します。常に付属のテンプレートをコピーし、コピーを変更してください。

4. 混乱を避けるために、既存のフォーム・モジュール (OFG4PC1T_TEMPLATE) の名前をもっと意味のあるもの (例、CUSTOM_TEMPLATE_1) に変更します。
5. コピーしたテンプレートをOracle Forms Designerの中でオープンした状態で、オブジェクト・ナビゲータのキャンバスビューをダブルクリックします。

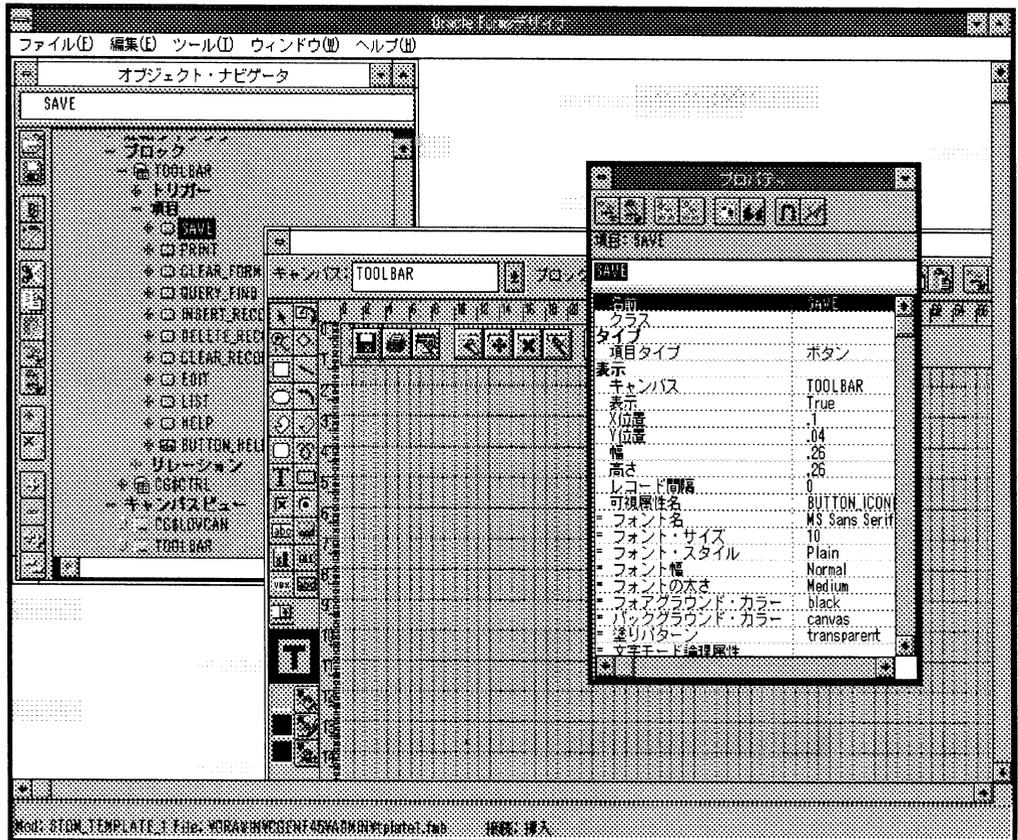
3つのキャンバスビューが定義されています。TOOLBARキャンバスはユーザー・オブジェクト、そしてCG\$LOVCANキャンバスとCG\$STACKED_FOOTERキャンバスはジェネレータ・オブジェクトです。

6. TOOLBARキャンバスが入ったレイアウト・エディタを表示するためにTOOLBARの横にあるアイコンをダブルクリックします。



TOOLBARキャンバスには、このテンプレートを使って生成されたフォームの中でツールバーとして表示されるアイコン・ボタンが入っています。

- 「プロパティ」ウィンドウを表示するためにTOOLBARキャンバスの中にある最初のボタン項目をダブルクリックします。

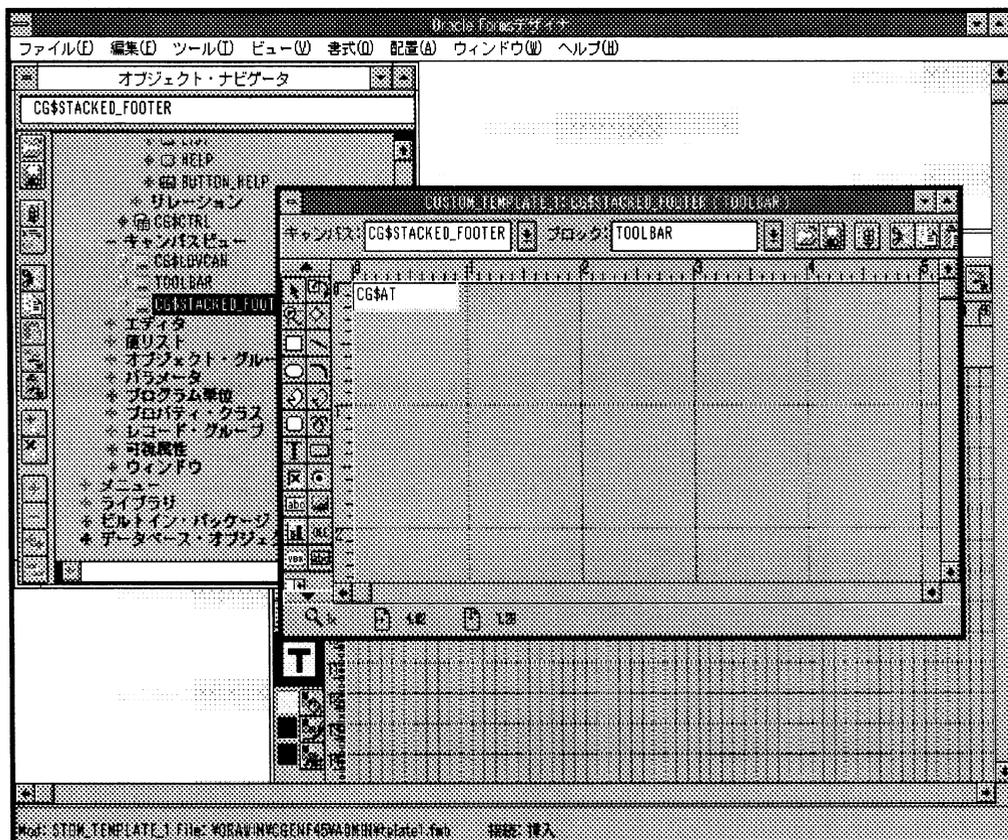


ボタンの名前はSAVEで、その外観はBUTTON_ICON可視属性によって制御されます。その名前から、キャンバス、ボタン、名前付きの可視属性がすべてユーザー・オブジェクトであることがわかります。ユーザー・オブジェクトなので、ジェネレータはそれらを生成されたフォームにそのままコピーします。

- BUTTON_HELP項目のプロパティを表示するためにその項目をクリックします。

BUTTON_HELPボタンの「表示」プロパティはfalseに設定されていることに注意してください。この項目は、カーソルがツールバー上のボタンに一定時間とどまっているときに表示されます。この機能をサポートするために、トリガーとプログラム単位の形のユーザー・オブジェクトがあります。

- オブジェクト・ナビゲータの中のCG\$STACKED_FOOTERキャンバスの横にあるアイコンをダブルクリックします。



CG\$STACKED_FOOTERはジェネレータ・オブジェクトです。この場合、キャンバス上の唯一の項目はCG\$ATと呼ばれるテキスト項目です。CG\$AT項目があることにより、ジェネレータはリポジトリからアプリケーション・システム・タイトルを取り出します。

ジェネレータが生成時にリポジトリから取り出した情報によって何を行うかは、ジェネレータ項目の表示プロパティの設定によって異なります。

- 表示プロパティがtrueに設定されている場合、ジェネレータはリポジトリから取り出した値が入っているポイラープレートを作成します。
- 表示プロパティがfalseに設定されている場合、ジェネレータは非表示項目を作成します。項目のヒント・テキスト・プロパティは、リポジトリから取り出した値に設定されます。このヒント・プロパティはトリガーやプログラム単位で使用することができます。

標準テンプレートには、CG\$AT項目から生成された非表示項目に関するヒント・テキストを使ってウィンドウ・タイトルを設定するためのフォームレベルPRE-FORMトリガーにユーザー・コードがあります。

この章の残りの部分で、現在表示されているテンプレート・フォームに変更を施していきます。

背景としてのボイラープレートの作成

テンプレート・フォームの使い方の1つとして、生成したフォームの各ページ上の背景として使用するためのコンテキスト・キャンバスを提供することがあります。

1. Oracle Forms Designerで、ofgpc1t.fmbからコピーしたテンプレートの中に新しいキャンバスを作成します。

ヒント: オブジェクト・ナビゲータの中のキャンバスビューをクリックし、ナビゲータ→作成を選択します。

2. 新しく作成したキャンバスの名前をCG\$PAGEに変更します。

CG\$PAGEと呼ばれるテンプレート・キャンバス上のオブジェクトが、生成されたフォーム内のすべてのコンテキスト・キャンバスに表示されます。

3. レイアウト・エディタを表示するためにオブジェクト・ナビゲータの中のCG\$PAGEの横にあるアイコンをダブルクリックします。
4. キャンバス上でテキスト・ツールを使って適切なボイラープレート・テキストを入力します。

実際の例では、次のテキストが入力されています。

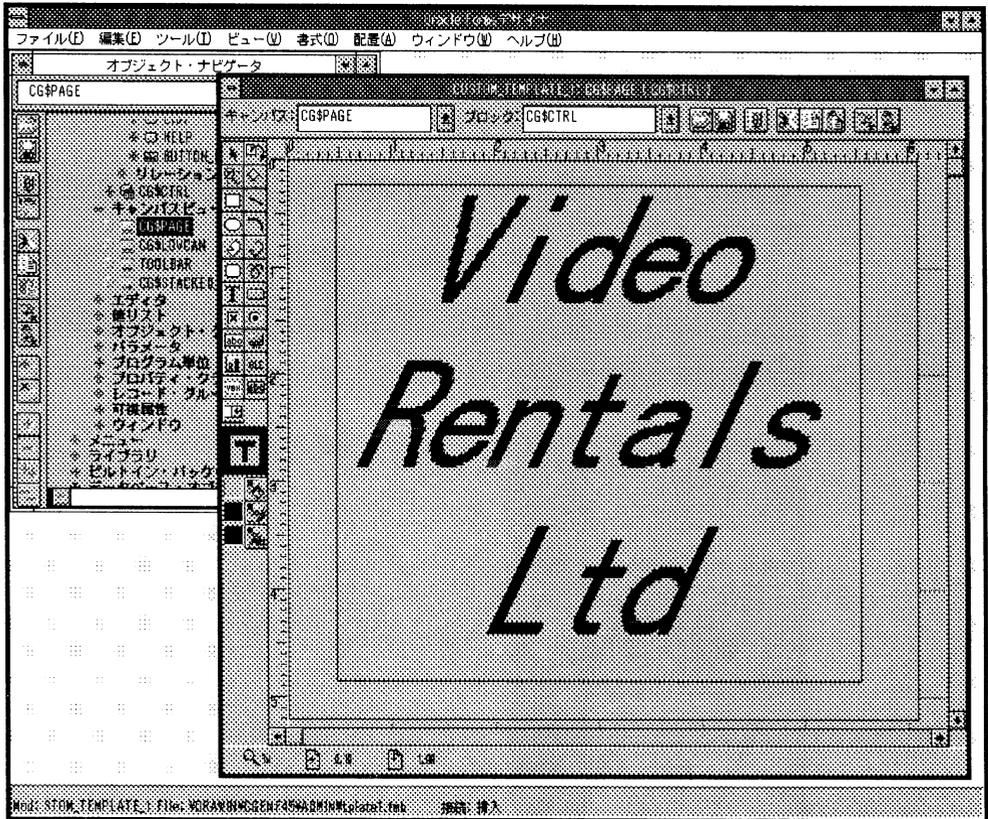
Video
Rentals
Ltd

5. 書式→フォントを選択し、テキストのプロパティを必要に応じて変更します。

実際の例では、テキストのプロパティが次のように変更されています。

フォント名:	Arial
スタイル:	ボールドイタリック
サイズ:	100

6. 書式→文字位置を選択し、テキストの文字位置を必要に応じて変更します。
実際の例では、文字位置が「中央揃え」に変更されています。
7. テキストを生成されたフォーム内の表示したい位置に移動します。



8. キャンバスの背景色を必要に応じて変更します（多くの場合、明るい色が適します）。
 ヒント: キャンバスの背景色を変更するには、オブジェクト・ナビゲータの中のキャンバス名をクリックし、ツール→プロパティを選択し、「バックグラウンド・カラー」プロパティを適切に設定します。
9. ボイラープレート・テキストの背景色を必要に応じて変更します（キャンバス背景色に近くして目立たなくするか、異なる色にして目立たせます）。この例では、「バックグラウンド・カラー・プロパティ」は「塗りなし」に設定されています。
 ヒント: ボイラープレート・テキストの背景色を変更するには、ボイラープレート・テキストを選択し、「塗りカラー」ツールバー・ボタンをクリックし、必要な色を選択します。
10. ボイラープレート・テキスト枠を必要に応じて変更します（キャンバスの背景色に近くして目立たなくするか、異なる色にして目立たせます）。この例では、「枠なし」に設定されています。
 ヒント: ボイラープレート・テキストの枠色を変更するには、ボイラープレート・テキストを選択し、「線カラー」ツールバー・ボタンをクリックし、必要な色を選択します。

生成されたフォームの機能 モジュールを生成すると次に示すようなフォームができあがります。

CG\$PAGEキャンバスに入力したボイラープレート・テキストが、生成されたオブジェクトの下に表示されます。

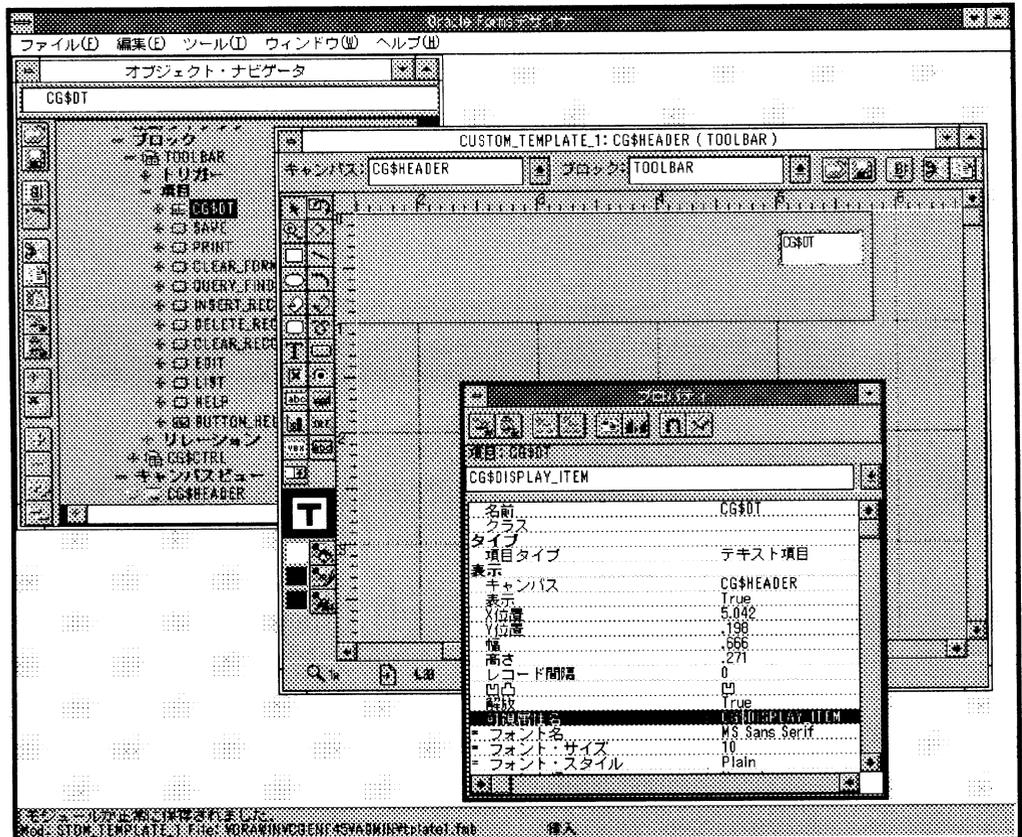
生成されたフォームはCG\$PAGEキャンバスの寸法になることに注意してください。

ジェネレータ・オブジェクトの作成

テンプレート上にジェネレータ項目があることにより、ジェネレータは生成されるフォームの中に適切なコードを作成します。

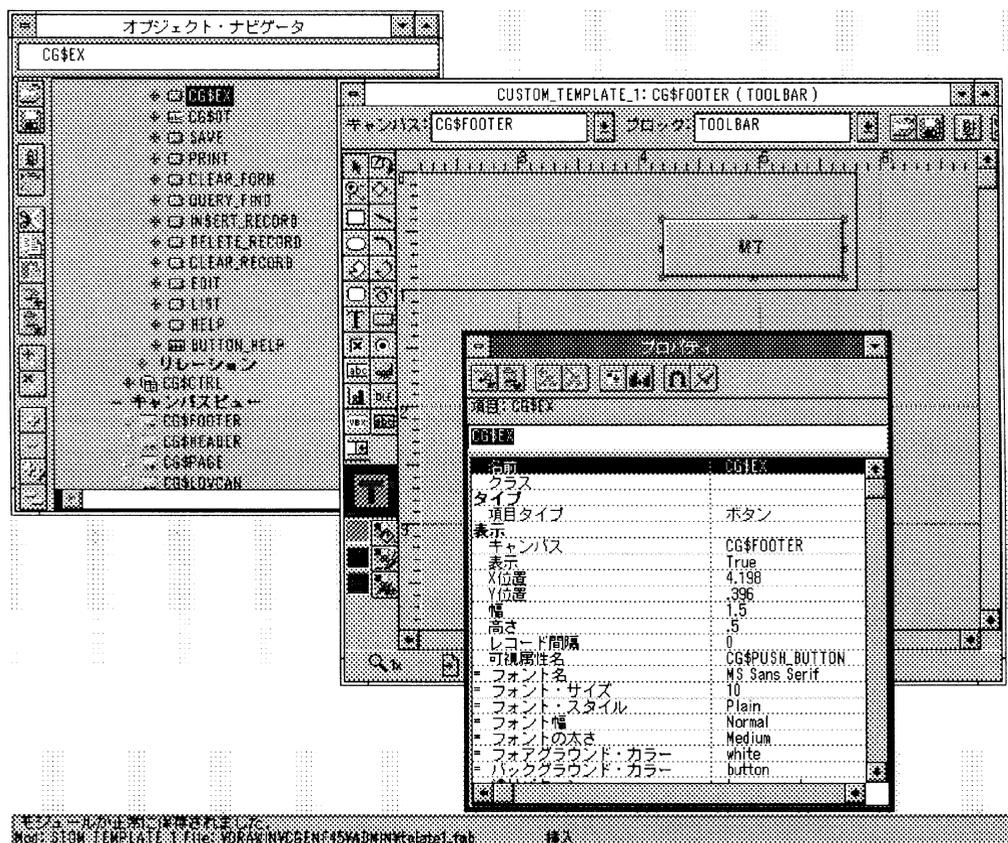
1. Oracle Forms Designerで新しいキャンバスを作成します。
2. 新しいキャンバスはCG\$HEADERという名前にします。
3. CG\$HEADERキャンバスをレイアウト・エディタに表示します。
4. キャンバスをサイズ変更して、CG\$PAGEキャンバスより幅が少し狭く、高さが1インチになるようにします。
5. CG\$HEADERキャンバスの右隅に次のプロパティを使って新しいテキスト項目を作成します。

名前:	CG\$DT
可視属性名:	CG\$DISPLAY_ITEM



6. CG\$FOOTERという新しいキャンバスを作成し、それをレイアウト・エディタに表示します。
7. CG\$FOOTERキャンバスをCG\$HEADERキャンバスと同じサイズにします (すなわち、CG\$PAGEより幅が狭い)。
8. CG\$FOOTERキャンバスの右隅に幅約1.5インチ、高さ約0.5インチのボタンを作成します。
9. ボタンに次のプロパティを与えます。

名前:	CG\$EX
可視属性名:	CG\$PUSH_BUTTON
ラベル:	終了



10. オブジェクト・ナビゲータを使って作成した項目がCG\$CTRLブロックの中にあるかを確認します。ない場合、中に移動します。

生成されたフォームの機能 モジュールを生成すると、次に示すようなフォームができあがります。

レンタル各レンタルの詳細情報

メンバー番号: 開始日:

従業員: 番号:

支払総額:

レンタル状況: 未払

レンタル日: 1996/12/01

終了

ジェネレータは、CG\$HEADERキャンパスのCG\$DT項目を生成されたページの中にマージすることによって、現在の日付を示す表示フィールドを作成しました。

ジェネレータは、CG\$EXボタンを生成されたページの中にマージすることによって、フォームの下部に「終了」ボタンを作成しました。ジェネレータは、「終了」ボタンに必要な処理を実行するためのWHEN-BUTTON-PRESSEDトリガーを作成しました。

ユーザー・オブジェクトの作成

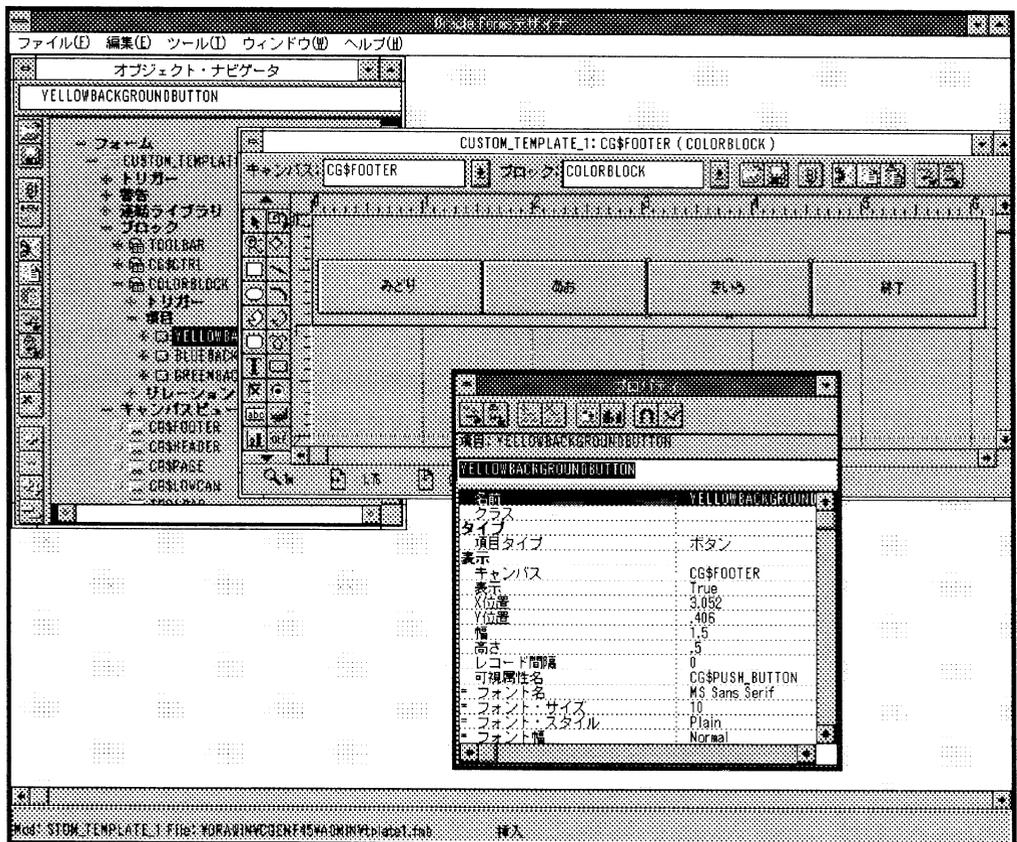
1. OracleFormsDesignerで、オブジェクト・ナビゲータの中の「ブロック」をクリックし、ナビゲータ→作成を選択し、「新規ブロック・オプション」ダイアログ・ボックスで次のものを選択することによって新しいブロックを作成します。

実表:	<NONE>
ブロック名:	COLORBLOCK
キャンバス:	CG\$FOOTER

2. CG\$FOOTERキャンバスをレイアウト・エディタに表示します。
3. 幅約1.5インチ、高さ約0.5インチのボタンを3つ作成します。
4. ボタンをキャンバス内で水平に整列します。
5. ボタンに次のプロパティを与えます。

名前:	GREENBACKGROUND-BUTTON	BLUEBACKGROUND-BUTTON	YELLOWBACKGROUND-BUTTON
可視属性名:	CG\$PUSH_BUTTON	CG\$PUSH_BUTTON	CG\$PUSH_BUTTON
ラベル:	みどり	あお	きいろ

6. オブジェクト・ナビゲータを使って作成した項目がCOLORBLOCKブロックの中にあるかを確認します。ない場合、中に移動します。



7. 可視属性を3つ作成します。名前とバックグラウンド・カラー属性は次のようになります。

名前:	バックグラウンド・カラー:
GREENBACKGROUND	r70g100b70
BLUEBACKGROUND	r70g100b100
YELLOWBACKGROUND	r100g100b70

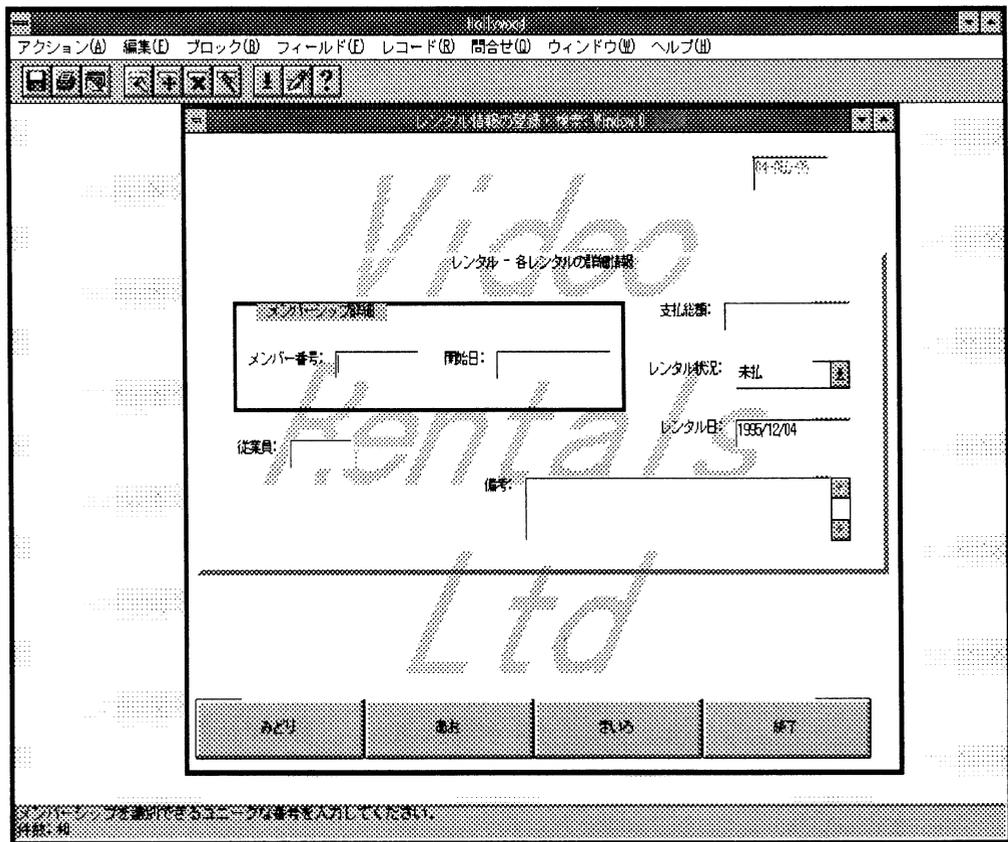
つぎに可視属性を使用する3つの項目のためのWhen-Button-Pressedトリガーを作成します。

8. トリガーを連結する項目の下の「トリガー」ノードにカーソルを移動します。
9. トリガーのリストを表示するためにナビゲータ→作成を選択し、When-Button-Pressedトリガーを選択します。
10. PL/SQLエディタで、指定された項目について下記のトリガー・コードを入力します。

BLUEBACKGROUNDBUTTON	set_canvas_property('CG\$PAGE_1', visual_attribute, 'BLUEBACKGROUND');
GREENBACKGROUNDBUTTON	set_canvas_property('CG\$PAGE_1', visual_attribute, 'GREENBACKGROUND');
YELLOWBACKGROUNDBUTTON	set_canvas_property('CG\$PAGE_1', visual_attribute, 'YELLOWBACKGROUND');

11. トリガーをコンパイルするために「コンパイル」ボタンをクリックします。
12. プロシージャが正常にコンパイルされたら、PL/SQLエディタを終了するために「クローズ」ボタンをクリックします。

生成されたフォームの機能 モジュールを生成すると次に示すようなフォームができあがります。



ジェネレータは、3つのユーザー定義ボタンを生成されたページ内にマージすることによって、フッター上に3つのボタンを作成しました。各色ボタンをクリックすると、CG\$PAGE_1キャンパスの色が変化します。

テンプレートフォーム内のプロシージャを呼び出すためのボタン項目の生成

テンプレート上にボタンを定義すると同様に、PL/SQLプロシージャを実行するために列取扱細目に基づいたボタンを定義することができます。プロシージャは次のいずれかに存在していなければなりません。

- サーバー
- テンプレート・フォーム
- 連結されたライブラリ

プロシージャをどこに作成したらよいか分からない場合、次のガイドラインを参考にしてください。

- プロシージャがSQLを使用する場合、サーバー上になければなりません。
- プロシージャがSQLを使用せず、汎用の（すなわち、複数の生成されたフォームで使用される）場合、テンプレート・フォーム内でプログラム単位として定義するか、ライブラリ内で定義し、そのライブラリをテンプレート・フォームと連結します。
- プロシージャがSQLを使用せず、特定のモジュール専用の場合、ライブラリ内で定義し、MODLIB作業環境を使って生成されたフォームと連結するライブラリを識別します。（注意: この機能が利用できるのは、Designer/2000リリース1.1以降です。）

次の例では、押すとテンプレート・フォーム内に保持されているPL/SQLが呼び出されるボタンを生成する方法を示します。まず、CG\$PAGE_1キャンパスのカラーをグレーに設定するためにテンプレート内にプロシージャを作成します。つぎにモジュールによってすでに使用されている列の2番目の取扱を作成し、それを使ってプロシージャを呼び出します。

1. Oracle Forms Designerで、次のような名前とバックグラウンド・カラー・プロパティを持つ可視属性を作成します。

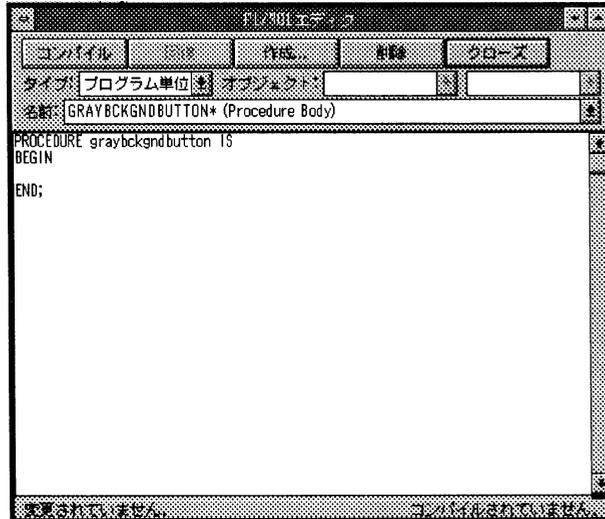
名前:	バックグラウンド・カラー:
GRAYBACKGROUND	gray

この可視属性を使用する新しいプロシージャを作成します。

2. 「プログラム単位」ノードを選択します。
3. ナビゲータ→作成を選択し、「新規プログラム単位」ダイアログ・ボックスの「名前」フィールドにgraybckgndbuttonとタイプします。

4. 「新規プログラム」ダイアログ・ボックスをクローズするために「OK」ボタンをクリックします。

PL/SQLエディタが自動的に表示されます。

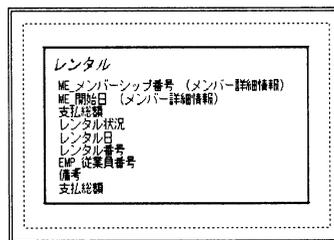


5. PL/SQLエディタの中のBEGINという語の下に次のようにタイプします。

```
set_canvas_property('CG$PAGE_1',  
visual_attribute,'GRAYBACKGROUND');
```

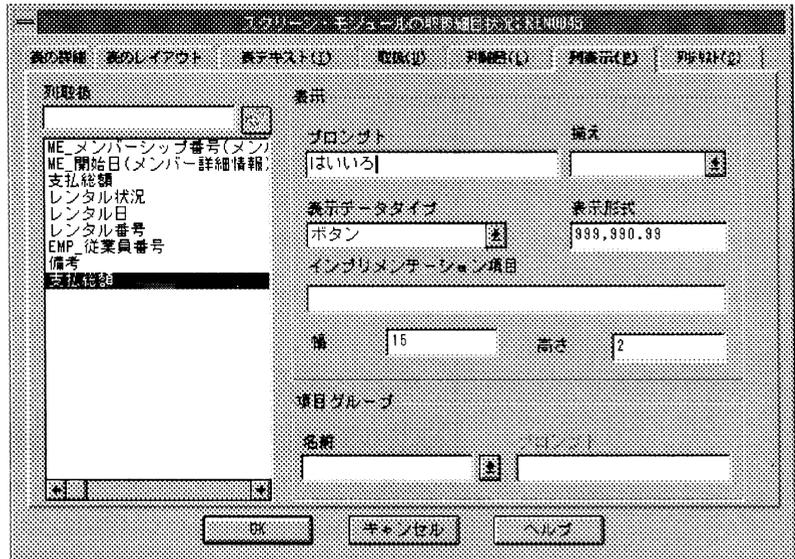
6. PL/SQLエディタを終了するために「クローズ」ボタンをクリックします。
7. モジュール・データ・ダイアグラマで、既存の列取扱細目の別の取扱を作成します。

この例では、支払総額の別の取扱を作成しました。新しい取扱は第2取扱と呼ばれます。



8. 「スクリーンモジュールの取扱細目状況」ダイアログ・ボックスを表示します。
9. 「取扱」タブの「表示」チェックボックスをチェックし、支払総額の第2列取扱細目に関する他のすべての取扱フラグのチェックを解除します。
10. 支払総額の第2取扱における「列表示」タブで、「表示データ型」ドロップダウン・リストからボタンを選択します。

11. 「幅」フィールドの値を15に変更します。
12. 「高さ」フィールドの値を2に変更します。
13. 「プロンプト」フィールドの既存の入力を「はいいろ」と置き換えます。
「列表示」タブは次のようになります。



14. 支払総額の第2の取扱に関する「列テキスト」タブで、「タイプ」ドロップダウン・リストからPL/SQLブロックを選択し、次のようにタイプします。

COMPLEX:graybckgndbutton

注意: COMPLEXという語は大文字でなければなりません。

15. 「スクリーン・モジュールの取扱細目状況」ダイアログ・ボックスをクローズするために「OK」ボタンをクリックします。

生成されたフォームの機能 モジュールを生成すると次に示すようなフォームができあがります。

レンタル - 各レンタルの詳細情報

メンバー番号: 開始日:

支払総額:

レンタル状況: 未払

従業員: レンタル日: 1995/12/04

備考:

メンバーシップを印刷できるユーザは番号を入力してください。
件数: 件

ジェネレータは、支払総額の第2列取扱からボタン項目を生成しました。「はいいいろ」ボタンをクリックすると、CG\$PAGE_1キャンパスの色が変化します。

第9章

複数の表取扱を持つフォームの生成

この章では、複数の表取扱を持つモジュールを作成しフォームを生成する方法を示します。この章は次の節からできています。

- 複数の表取扱を持つフォームの紹介
- 参照取扱の作成
- マスター・ディテール・フォームの作成
- 配置の指定
- 各種配置オプション
- ブロックの同期化の使用

複数の表取扱を持つフォームの紹介

複数の表からの情報をフォームに表示したい場合がよくあります。

ジェネレータは、モジュールの表取扱細目と表取扱間に存在する結合から生成されるフォームにブロックを導出します。表取扱には2種類あります。

- 基礎表取扱
- 参照表取扱

生成されるフォーム内の各ブロックは基礎表取扱から導出されます。ブロックの中の項目は、基礎表内の列からまたは参照表内の列から導出されます。参照表から導出された列は、基礎表の中の情報と関連するデータを表示します。

関連する情報を示すためにフォーム上の別のブロックが必要な場合、適切な基礎表取扱の間の結合を定義します。結合は、表の間に存在する外部キーに基づいています。結合を明示的に描くか、デフォルト結合生成ユーティリティを使うことによって、モジュール・データ・ダイアグラムの中のダイアグラム上に新しい結合を作成することができます。

参照取扱の作成

参照表用の表取扱とは何か？ 参照表取扱は、生成されたフォームの中のブロックが参照目的のみに使用する表の取扱細目です。参照表の情報は、生成されるブロックに表示できますが、変更はできません。

記述子列とは何か？ リポジトリで表や列を定義するとき、表の中の1つまたは複数の列を各レコードに関する記述情報を提供する記述子列として指定することができます。通常は、参照表の記述子列を生成するフォームに組み込みます。

参照表取扱を作成するのはなぜか？ 複数の表の情報を1つのブロックに表示したい場合がよくあります。たとえば、次に示す表取扱細目からフォームを生成するとします。

レンタル
レンタル番号
レンタル日
ME_メンバーシップ番号
ME_開始日
EMP_従業員番号
支払総額
レンタル状況
備考

レンタル番号	レンタル日	ME_メンバーシップ番号	ME_開始日
従業員番号	支払総額	レンタル状況	
備考			

生成されたフォームは、トランザクションを入力したスタッフ・メンバーの従業員番号を示します。

しかし、従業員名も見たいとします。この情報は別の表に入っています。この情報をフォームに組み込むには、外部キーに基づいて2つの表の間の結合を指定しなければなりません。

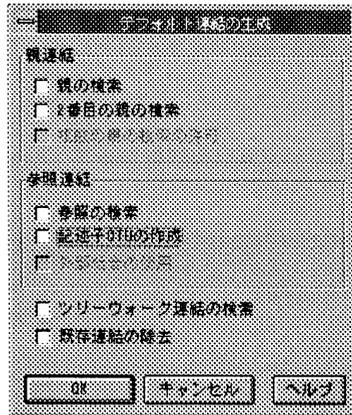
参照表取扱細目を作成するには、次の条件が満たされていなければなりません。

- ダイアグラム上に既存の基礎表取扱細目が少なくとも1つあること

- 基礎表と参照表の間に外部キー制約が存在していること

デフォルト結合生成ユーティリティの実行 参照表の列に基づいて項目を組み込むには、

1. ユーティリティ→デフォルト結合の生成を選択します。「デフォルト結合の生成」ダイアログ・ボックスが表示されます。



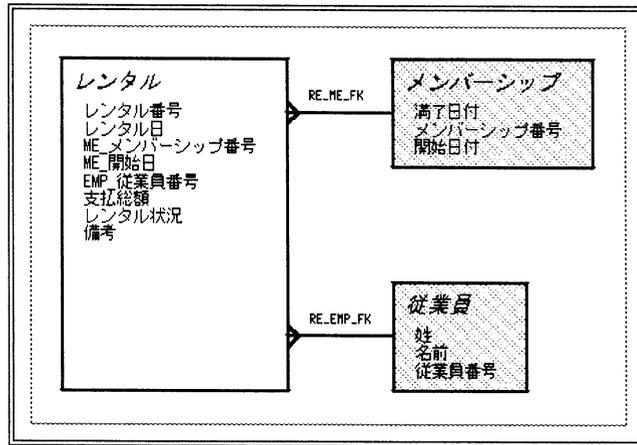
2. 「記述子DTUの作成」チェックボックスをチェックし、「OK」ボタンをクリックすると、デフォルト結合生成ユーティリティが実行されます。

次の条件が満たされている場合、ユーティリティは参照表取扱細目を作成します。

- 基礎表取扱の各外部キー構成要素表に列取扱細目が存在していること
- 参照表取扱細目の中に、記述子として定義された、外部キーの一部ではない列が1つもしくは複数存在すること

参照表取扱細目はダイアグラムの中で基礎の横に表示されます。次の列取扱細目が作成されます。

- 外部キーを構成する列の非表示取扱
- 記述子列として定義された列の表示取扱



この例では、デフォルト結合生成ユーティリティがメンバーシップと従業員の2つの参照表取扱を識別しました。

EMP_従業員番号という列取扱細目、従業員への外部キーに基づいて、ユーティリティは従業員という参照取扱を作成しました。

姓と名前という列が記述子として定義されているため、表示されたこれらの列の列取扱細目が作成されました。

レンタルという表取扱細目にあるME_メンバーシップ番号およびME_開始日という列取扱細目のため、ユーティリティはメンバーシップという参照細目を作成しました。これらの列はメンバーシップへの外部キーを形成します。これらの列細目のいずれかがレンタルという表取扱細目に存在していた場合、参照細目は作成されません。

記述子列から自動的に作成される列取扱細目とともに、参照表用の表取扱にある他の列に基づいてより細かな列取扱を手動で作成できます。

3. 従業員という参照表用の表取扱を選択するためにそれを囲んでいる枠をクリックします。
4. 編集→作成→列取扱細目を選択します。列取扱細目の作成ダイアログ・ボックスが表示されます。
5. 列取扱細目を作成するためにイニシャルをダブルクリックします。
6. 姓と名の列取扱細目の間にイニシャル列取扱細目をドラッグし、マウス・ボタンを放します。

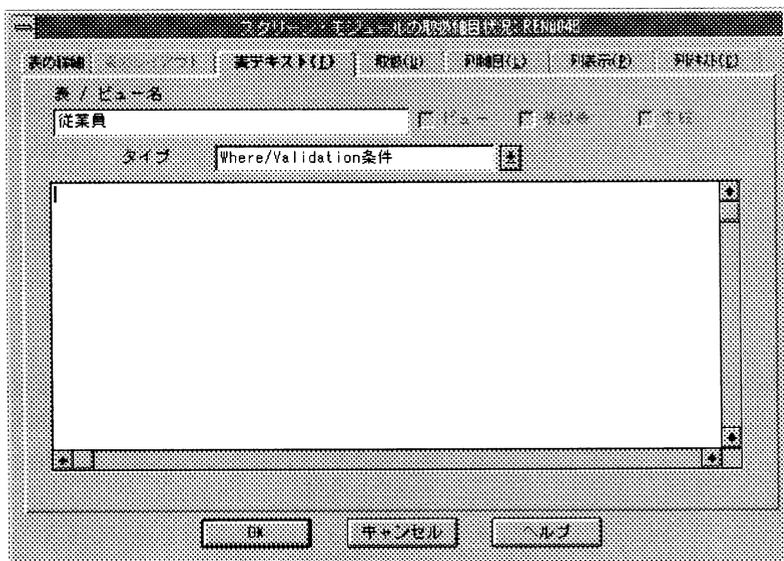
参照取扱からのLOVの作成

参照取扱の列に基づいて項目を組み込む場合、それらの項目について一覧表 (LOV) を利用可能に指定することができます。LOVには、参照表取扱にある任意の列を組み込むことができます。LOVは、選択取扱フラグがセットされている参照表の列取扱に基づいた項目について利用可能です。

1. 従業員という参照取扱にある従業員番号をダブル・クリックします。「スクリーン・モジュールの取扱細目状況」ダイアログ・ボックスが表示されます。
2. 「取扱」タブで、従業員番号と姓という取扱に関する選択と一覧のチェックボックスを選択します。

一覧表の情報を制限するために、表の参照取扱に対する制限条件を入力します。たとえば、レンタル情報を入力する従業員はアシスタントだけの場合、LOVに表示される従業員を制限するとユーザーの時間を節約できます。

3. 「スクリーン・モジュールの取扱細目状況」ダイアログ・ボックスの「表テキスト」タブを表示します。



4. テキスト・タイプをWhere/Validation条件 (デフォルト) に設定し、次の内容をテキスト・フィールドにタイプします。

```
position='ASSISTANT'
```

5. ダイアログ・ボックスをクローズするために「OK」ボタンをクリックします。

生成されたフォームの機能

モジュールを生成すると次に示すようなフォームができあがります。

The screenshot shows a 'Receipt Information' window with several input fields: Transaction No., Date, Membership, Start Date, Expiry Date, Employee, Last Name, Other Initials, First Name, Amount Paid, Status, and a text area for Staff Remarks. An 'Employees' search overlay is active, showing a list of employees with columns for 'Employee' and 'Last Name'. The search criteria is '20%'.

Employee	Last Name
2005	DRIVER
2006	AHUMUGAMA
2007	PALLOTIA
2008	LUTHER
2009	DIMON
2010	HOYOS
2011	DE SWERT

参照列取扱に基づく項目は外部キー列に基づく項目の後に表示されることに注意してください。

- 満了日付という記述子列取扱に基づく項目は、番号と開始日付に基づく項目の後に表示されます。
- 姓と名前という記述子列取扱に基づく姓と名前の項目は、従業員番号に基づく項目の後に表示されます。
- イニシャルという参照列取扱に基づく項目は、姓と名前に基づく項目の間に表示されます。
- 問合せ入力モードでは、従業員項目で利用可能な一覧表には、アシスタントであるすべての従業員の従業員番号と姓が組み込まれます。

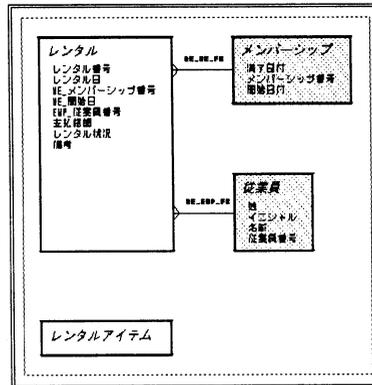
マスター・ディテール・フォームの生成

マスター・ディテール・フォームとは何か？

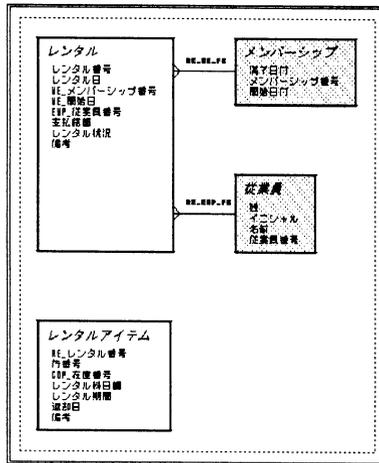
マスター・ディテール・フォームには少なくとも2つのブロックが入っています。2番目のブロック（ディテール・ブロック）には、1番目のブロック（マスター・ブロック）に示されるレコードに関連した情報が入っています。ディテール・ブロックは、マスター・ブロックへの外部キーによって関連づけられ、外部キーの列はディテール・ブロック内になければなりません。

マスター・ディテール・フォームの生成

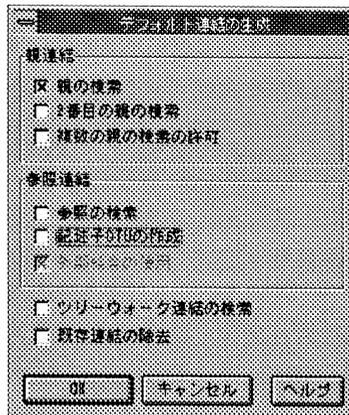
1. 編集→作成→表取扱細目を選択し、表のリストからレンタルアイテムを選択します。



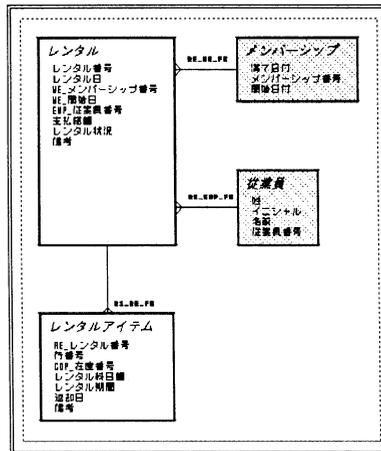
2. レンタルアイテム表取扱を選択した状態で、編集→作成→列取扱細目を選択します。「列取扱細目の作成」ダイアログボックスが表示されます。
3. 「表示順序による並べ替え」チェックボックスをチェックします。
4. 「列取扱細目の作成」ダイアログ・ボックスに示されるすべての列を選択するために「すべて選択」ボタンをクリックします。
5. 「OK」ボタンをクリックすると、列取扱細目が作成されます。



- ユーティリティ→デフォルト結合の生成を選択します。「デフォルト結合の生成」ダイアログ・ボックスが表示されます。
- 「親の検索」チェックボックスをチェックします。



8. 「OK」 ボタンをクリックすると、ユーティリティが起動されます。



レンタルアイテムという取扱が外部キーRI_RE_FKによってレンタルという取扱とリンクされます。

生成されたフォームの機能

モジュールを生成すると次に示すようなフォームができあがります。

The screenshot shows a form titled 'レンタル' (Rentals) with the following fields:

- レンタル番号: [Text Field]
- レンタル日: [Date Field] (Value: 05/12/04)
- メンバーシップ: [Text Field]
- 開始日: [Date Field]
- 終了日: [Date Field]
- 従業員: [Text Field]
- VE_レンタル: [Text Field]
- 支払総額: [Text Field]
- レンタル状態: [Text Field]
- 備考: [Text Field]

Below the main form is a section for 'レンタルアイテム' (Rentals Items) with fields:

- 在庫番号: [Text Field]
- レンタル開始日: [Date Field]
- レンタル期間: [Text Field]
- 開始日: [Date Field]
- 備考: [Text Field]

レンタルアイテムにある列の列取扱細目に基づく項目が、フォームの2番目のブロックに表示されます。2番目のブロックは1番目のブロックと同じキャンバス上にあります。

配置の指定

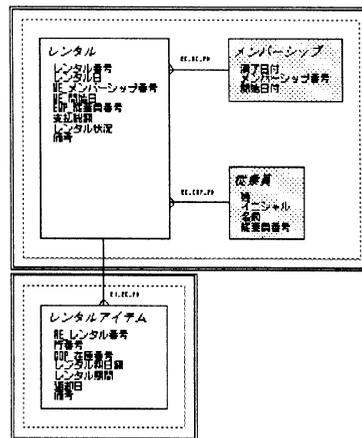
参照取扱から導出された項目は、関連する基礎表取扱の項目と常に同じキャンバス、同じウィンドウ内に存在します。

デフォルトでは、十分なスペースがある場合、ジェネレータはマスター・ブロックとディテール・ブロックを同じウィンドウ内の同じキャンバス上に位置づけます。しかし、表取扱細目をダイアグラム内の適切な配置項目の中に位置づけることによって、マスター・ブロックとディテール・ブロックが別々のキャンバスまたは別々のウィンドウもしくはその両方に表示されるように指定することもできます。

別ウィンドウへのディテール・ブロックの配置

1. 「新規ウィンドウ」描画ツールバー・ボタンを選択します。
2. ラバーバンドがレンタルアイテム表取扱を囲むようにドラッグすることによって、ダイアグラム上にウィンドウ配置項目を描画します。

ダイアグラムが変更され、レンタルアイテム表取扱が別のウィンドウ配置項目の中に表示されます。

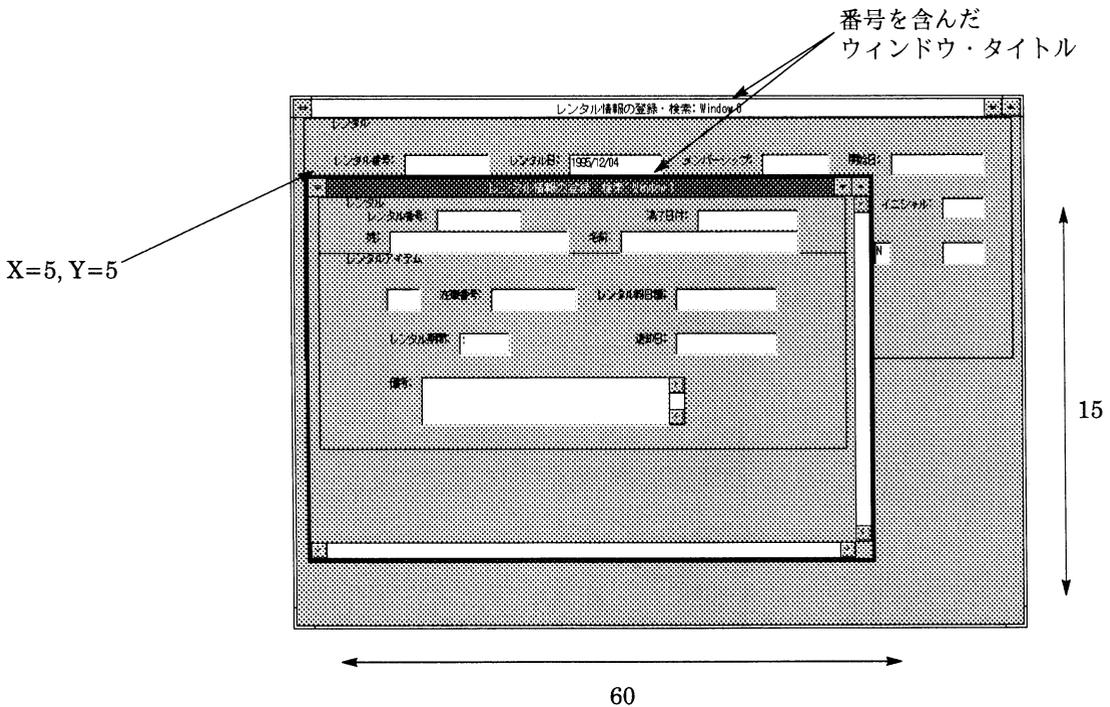


3. レンタルアイテムという表取扱細目をダブルクリックします。「スクリーン・モジュールの取扱細目状況」ダイアログ・ボックスが表示されます。

4. 「表のレイアウト」タブを次のように設定します。

ページ/ポップアップ 幅: 60
ページ/ポップアップ 高さ: 15
ウィンドウ/ビュー 幅: 60
ウィンドウ/ビュー 高さ: 15
ウィンドウ/ビュー X: 5
ウィンドウ/ビュー Y: 5

生成されたフォームの機能 モジュールを生成すると次に示すようなフォームができあがります。



2番目のブロック（レンタルアイテム）が新しいウィンドウに表示されます。ここでは、マスター・ブロックに示されるレンタルに関する詳細情報が入っています。

マスター・ブロックの項目のいくつかは2番目のウィンドウの詳細・ブロックの上に表示されることに注意してください。これらの項目は「コンテキスト項目」と呼ばれ、マスター・ブロックに関してディテール・ブロックのコンテキストを確立します。

デフォルトでは、ジェネレータは次のディテール・ブロックの上にコンテキスト項目を作成します。

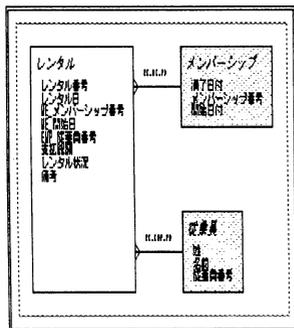
- マスター・ブロックの基礎表の主キー列
- マスター・ブロックの基礎表に関して定義された任意の記述子

上の例では、レンタル番号項目がコンテキスト項目として表示されます。これは、この項目がレンタル表の主キーだからです。満了日付、名前、姓の各項目は、記述子として定義された列の参照取扱に基づいているために組み込まれません。

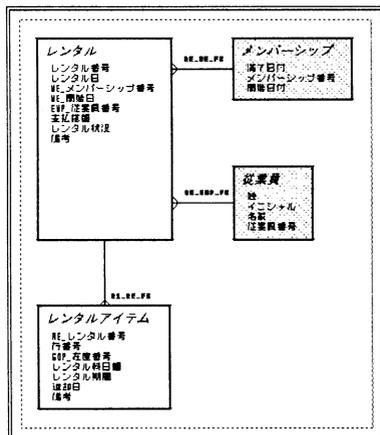
コンテキスト項目は、ディテール・ブロックの上の「ブロック・スタブ」に表示されます。ブロック・スタブに関して別のプロパティを指定し、デフォルトを使用するかわりに独自のコンテキスト項目を定義することができます。ブロック・スタブの作成と内容を制御するには、Layout - StubグループとLayout - Contextグループの作業環境を使用します。

各種配置オプション

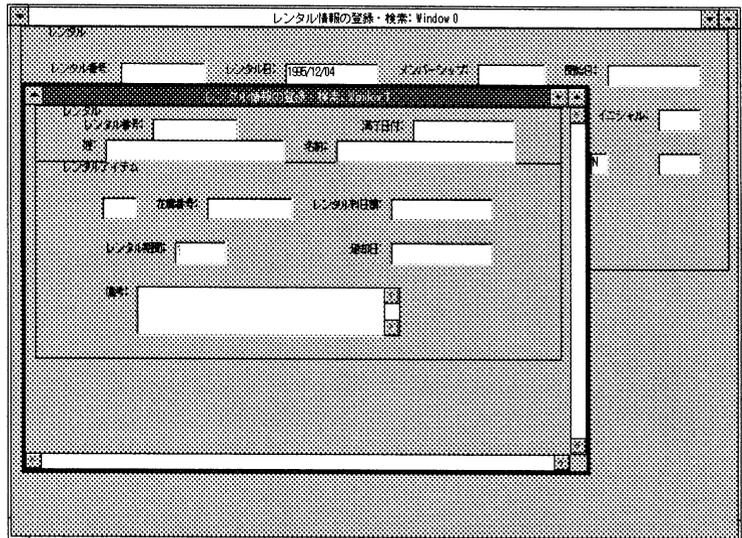
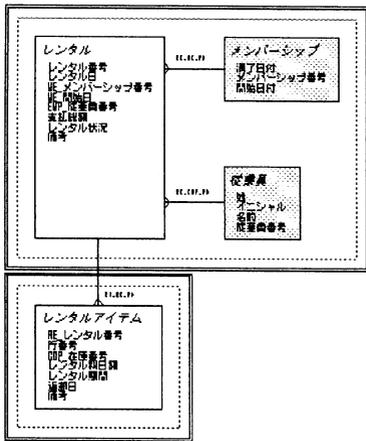
1番目のブロック: 位置 = 常に新しいページ/新しいウィンドウ



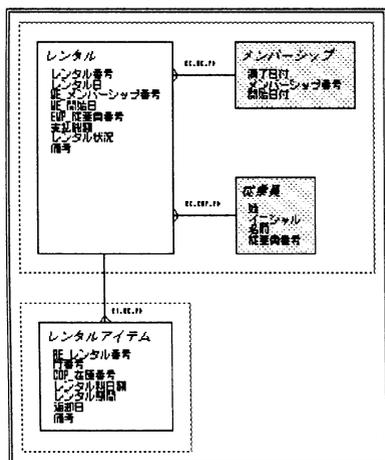
2番目のブロック: 位置 = 同じウィンドウ



2番目のブロック: 位置 = 新しいページ/新しいウィンドウ

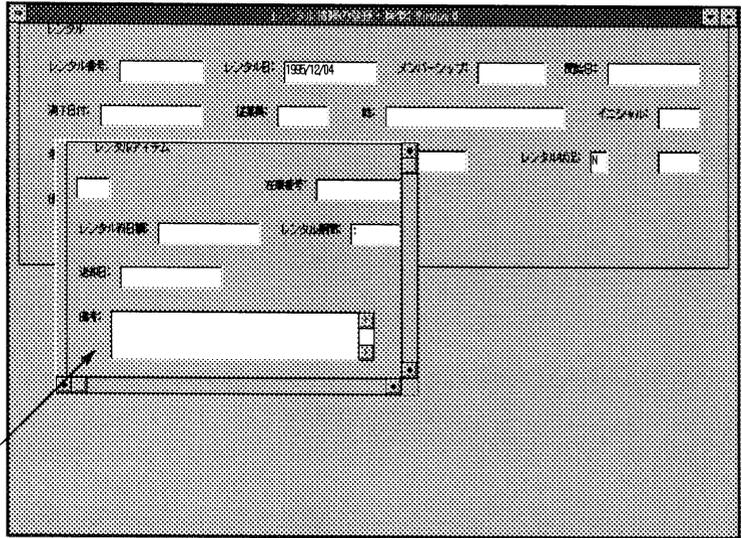
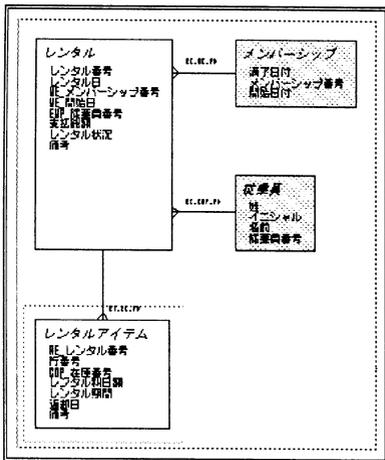


2番目のブロック: 位置 = 新しいページ



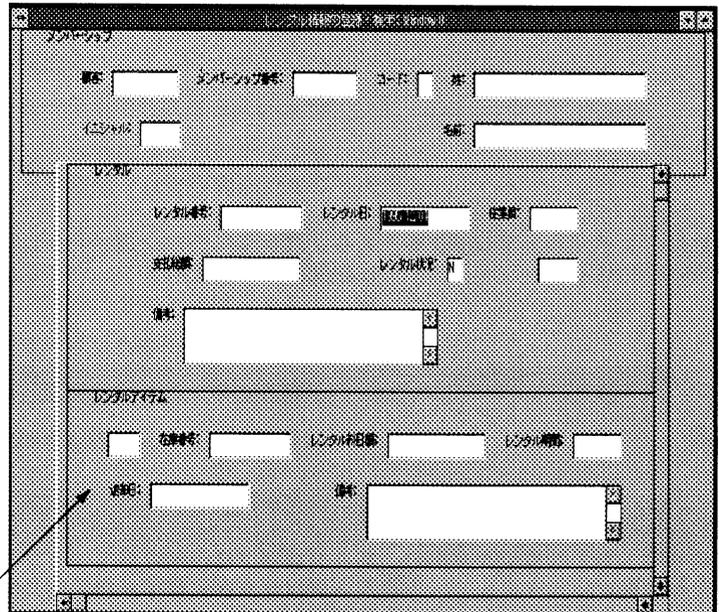
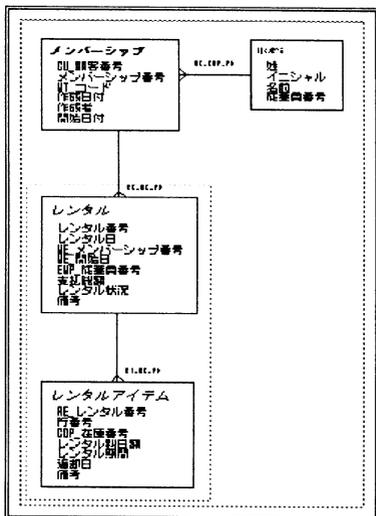
これらの2ページは、同一ウィンドウ(Window0)に表示されます。レンタルアイテムを基礎表にもつブロックが含まれる2ページ目が、1ページ目を上書きします。

2番目のブロック: 位置 = 新しいポップアップ



このブロックは、スタック・キャンバス（フォームの用語）またはポップアップ（ジェネレータの用語）に表示されます。

3番目のブロック: 位置 = 同じポップアップ



これらのブロックは、スタック・キャンバス（フォームの用語）またはポップアップ（ジェネレータの用語）に表示されます。

ブロックの同期化の使用

ブロックの同期化とは何か？ 一般にフォームがマスター・ブロックとディテール・ブロックからできている場合、ディテール・ブロックに示される情報とマスター・ブロックに示される情報との関連性を保っておきたいと考えるのが普通です。このためのプロセスをブロックの同期化と呼びます。

Oracle Formsジェネレータは、複数のキャンバスとウィンドウの間で生成されたブロックに表示されるデータの完全な同期をサポートしています。この同期は、ブロックが生成される表取扱細目の間に指定された結合に基づいています。ジェネレータは、リポジトリに記録された外部キー制約も利用します。

生成されるブロックの同期には主に2つの機能があります。

- ユーザーには、マスター・ブロック内のレコードを変更する前にディテール・ブロック内の変更をコミットするように指示するプロンプトが表示されます。
- ユーザーがマスター・ブロック内のレコードを変更した場合、ディテール・ブロック内のレコードも（ディテール・ブロックを再問合せするか、消去することによって）変更されます。

ブロックの同期の他の機能は、作業環境と生成したフォームのユーザーの選択によって制御されます。

各種のブロックの同期モード 次の3つのブロックの同期モードのいずれかで実行されるフォームを生成できます。

- 常にディテール・ブロックを問い合わせる
- ディテール・ブロックを問い合わせない
- 入ったときだけディテール・ブロックを問い合わせる

ユーザーがブロックの同期モードを変更できるようにするために、テンプレート・フォーム内の項目を定義することもできます。

常にディテール・ブロックを問い合わせる あるブロックに別のレコードを表示すると、そのすべてのディテール・ブロックは自動的に消去されます。現在表示されており、問合せ範囲内にあるディテール・ブロックが再問合せされます。問合せの範囲は次のように制限できます。

- 現在のページまたはポップアップに表示されているディテール・ブロック
- 現在のウィンドウに表示されているディテール・ブロック
- 現在のフォームに表示されているディテール・ブロック

ブロックが消去されると、そのすべてのディテール・ブロックも消去されます。

ディテール・ブロックを問い合わせない あるブロックに表示されているレコードを変更すると、そのすべてのディテール・ブロックが自動的に自動的に消去されますが、再問合せは行われません。

入ったときだけディテール・別のブロックのディテールであるブロックに初めて入ったときに、問合せが自動的に実行されます。しかし、行が返されず、ブロックが再びナビゲートされた場合、問合せは反復されません。

第10章

モジュール間のナビゲート

この章では、他の生成されたフォームと生成されたフォームのレポートを起動する方法について説明します。この章は次の節からできています。

- ナビゲーション生成の概要
- メニューの生成
- サブメニューの生成
- ナビゲート可能モジュールのポップリストの作成
- ナビゲーション・ボタンの生成
- テンプレート・メニューを使ったメニューの生成

ナビゲーション生成の概要

一般に構築するアプリケーションには、最終システムの各種機能分野を具現化する多数のモジュールが含まれます。

これら各種モジュールの関係をまとめてモジュール・ネットワークと呼びます。

アプリケーション設計ウィザードを使って初期のアプリケーション設計を作成した場合、簡単なモジュール・ネットワークが作成されます。

既存のモジュール結合と構造の変更

モジュール・ネットワークを手動で定義したり、アプリケーション設計ウィザードによって作成されたモジュール・ネットワークを変更したい場合があります。たとえば、次のようなことが考えられます。

- 画面モジュールをあるメニューから別のメニューへ移動します。
- 新しいモジュールを構造内に結合します。
- 既存のモジュールを構造から取り除きます。
- 既存のモジュールを複数のサブモジュールに分割します。

モジュール・ネットワークを作成および変更するのに使用する Designer/2000 ツールは、モジュール・ストラクチャ・ダイアグラムです。

モジュール・ネットワークと Formsジェネレータ

生成フォームのユーザーが、関連するフォームにナビゲートしたり、レポートを起動したり、PL/SQLルーチンを実行したりできるように Formsジェネレータは、モジュール定義とモジュール・ネットワーク内のモジュール定義の関係を 사용합니다。

Formsジェネレータは次のものを作成できます。

- トリガーと仮想キー
- 呼出可能モジュール (CG\$FF) のリスト - OFG4PC2Tはこの機構を使用します
- メニュー生成

トリガーと仮想キー

モジュールが別のモジュールを呼び出す場合、ジェネレータは、呼び出されるモジュールに基づいてフォームまたはレポートを呼び出すためのコードを生成します。この動作はCALLFM作業環境によって制御されます。

CALLFMがYに設定されている場合、ジェネレータは各Key-Fnトリガー (nは0から9までの番号) のコードを追加します。最初に呼び出されるフォームまたはレポートに使用されるトリガーは仮想Key-F0、2番目のトリガーは仮想Key-F1、というようになります。仮想キーを物理キーにマップするには Oracle*Terminalを使用します。

呼出可能モジュールのリスト 生成フォームにおいては、現在のフォームから呼出可能なフォームのリストを表示したい場合がよくあります。

このようなリストが入ったフォームを生成するには、テンプレート・フォームのヘッダーまたはフッターの中にCG\$FFと呼ばれるポップリストを作成します。フォームを生成すると、リスト内にそのフォームから呼出可能なすべてのフォームとレポートのリストが表示されます。

OFG4PC2Tテンプレートを使用する場合、他の生成されたモジュールを呼び出すためのボタンを生成することができます。OFG4PCT2テンプレートには、CG\$FFポップリスト項目と5つの汎用ボタンが入っています。テンプレート内のコードは、リスト内の各値をボタンに置き換えます。リスト内の項目の数が5つを超えている場合、5番目のボタンにはその他というラベルが付き、他のモジュールを表示するためのリストを呼び出します。

注意: モジュール・ネットワーク内に子モジュールが定義されているモジュールから生成するときだけ、OFG4PC2T.FMBをテンプレートとして定義します。

メニュー生成

Formsジェネレータは、モジュール・ネットワーク内の情報からメニュー階層を作成することができます。メニューは次の2つの方法で生成できます。

- モジュール・ネットワークの部分を生成するために、タイプがメニューまたはスクリーンのスタート・モジュールを指定します。
- フォームを生成する部分としてメニューを暗示的に生成します。

モジュール・ストラクチャ・ダイアグラムを使って入力された情報に加え、Formsジェネレータは、メニューを作成するための作業環境や「テンプレート・メニュー」に入った情報も使用します。

生成されたメニュー上の項目は次のものを起動できます。

- フォーム
- レポート
- サブメニュー
- 標準メニュー機能 (例、終了)

メニュー・セキュリティと Formsジェネレータ

Formsジェネレータは、メニュー項目の生成の中でメニュー項目を起動するのに必要なロールを作成したり、メニュー項目を適切なロールと関連づけたり、ユーザーにロールを付与することができます。ロールは、リポジトリに定義されたモジュール・ネットワーク内のモジュールへのアクセスによってグループから導出されます。

開発者に生成したメニューをテストするのに必要なロールが付与されていない場合があります。生成されたメニューのセキュリティを使用禁止にするには、MNUEUS作業環境をNに設定します。

Formsセキュリティ表にロールが存在しない場合、ジェネレータがメニュー項目を存在しないロールと関連づけないようにするために、MNUSAS作業環境をYに設定します。

注意: ロールを生成したり、ユーザーをロールに割り当てるには、ロール管理権限を持っていないけません。この権限がなくてメニューを生成したい場合、MNUSASをYに設定します。

メニュー生成の前提条件

Formsジェネレータを使ってメニューを作成する前に、次の条件が満たされているかを確認してください。

- Oracle Forms表がデータベース上に存在し、それにアクセスできること。
- DBA_ROLESとDBA_ROLE_PRIVSに関するSELECT権限を持っていること。

生成したメニューがフォームやレポートを呼び出す場合、適切な環境変数（それぞれFORMS45_PATHとREPORTS25_PATH）を実行可能ファイルの場所に設定します。

メニューの生成

いくつかのツールを使ってモジュールからメニューを生成することができます。しかし、モジュール・ストラクチャ・ダイアグラマを使うと、生成されたメニューの構造を簡単に可視化できます。

1. プリファレンス・ナビゲータを使ってアプリケーション・レベルで MNUSAS を Y に設定します。
ヒント: MNUSAS は「MENU - ROLES」グループの作業環境にあります。
2. モジュール・ストラクチャ・ダイアグラマを起動します。
ヒント: モジュール・データ・ダイアグラマを使用する場合、ツール→モジュール・ストラクチャ・ダイアグラマを選択します。
3. 新しいダイアグラマを作成するためにファイル→新規作成を選択します。
4. 編集→ネットワークのインクルードを選択します。「データベースからモジュール・ネットワークのインクルード」ダイアログ・ボックスが表示されます。

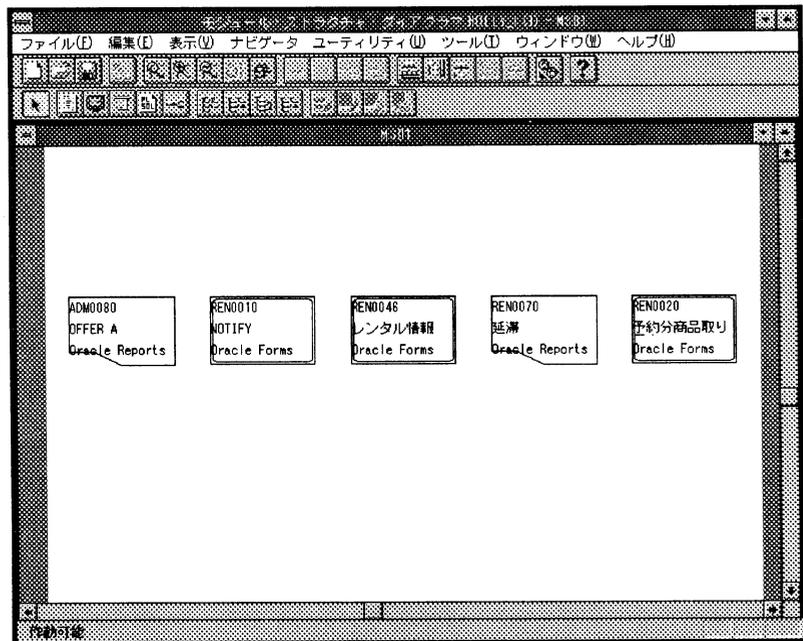


アプリケーション設計ウィザードを使って機能をモジュールにマップした場合、アプリケーション・システム内のモジュールは、おそらくすでに階層に編成されています。

階層内では、各モジュールの結合状況は次のいずれかになります。

- ルート（子モジュールはあるが、親モジュールがないモジュール）
 - フル（子と親の両方のモジュールがあるモジュール）
 - リーフ（親モジュールはあるが、子モジュールがないモジュール）
 - ソロ（親モジュールも子モジュールもないモジュール）
5. モジュール・ネットワークに組み込みたいモジュールを一つずつ（モジュールをダブルクリックする）か、一度に複数（[Control]キーを押したまま複数選択し、OKをクリックする）選択します。

「データベースからモジュール・ネットワークのインクルード」をクローズすると、モジュールがダイアグラム上に表示されます。



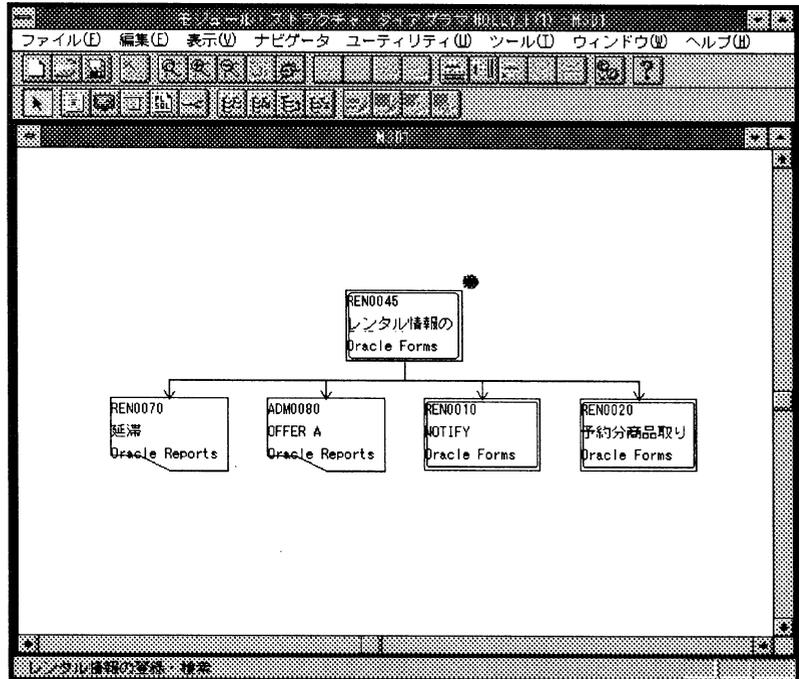
6. 「選択モジュールの位置変更」ツールバー・ボタンを使ってモジュール・ネットワークを作成します。

ヒント: モジュールを移動するには、

- a. 「選択モジュールの位置変更」ボタンをクリックします。
- b. 移動したいモジュール上でマウス・ボタンを押したままにします。

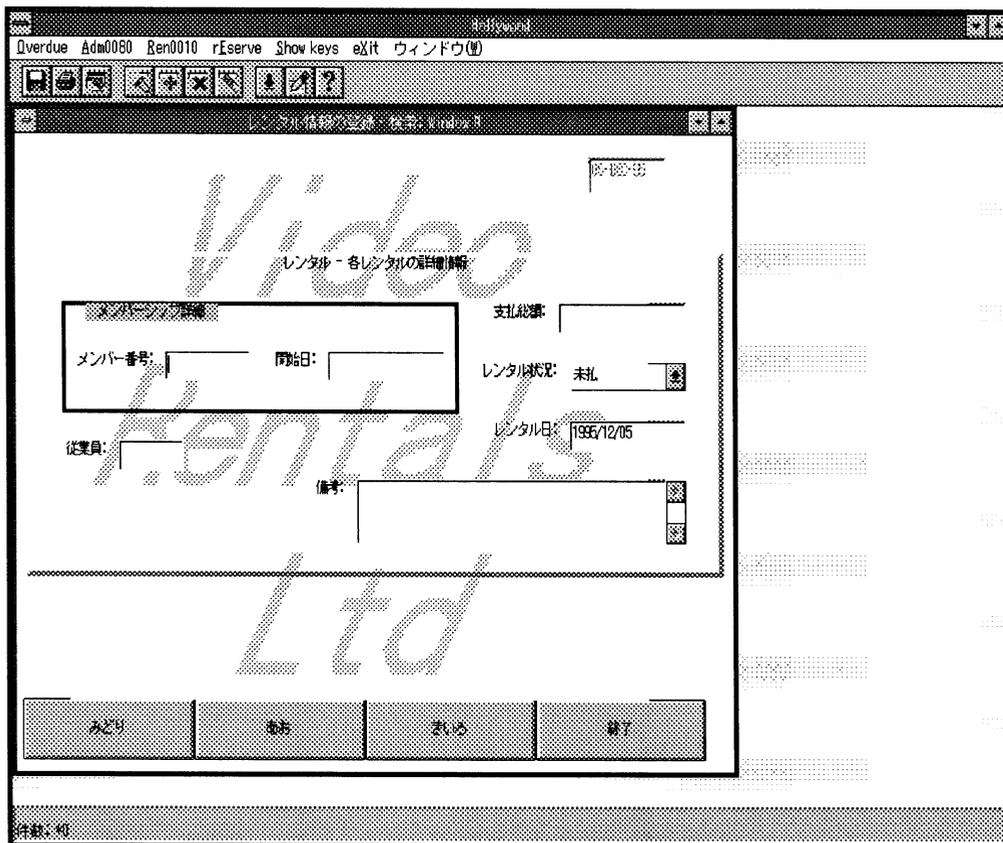
- c. カーソルを移動したい位置へドラッグします（選択したモジュールを別のモジュールの子モジュールにするには、モジュールを親モジュールの下にドラッグします。モジュールを新しいネットワークのルートにするには、モジュールをダイアグラムの空きスペースにドラッグします）。
- d. マウス・ボタンを放します。

次の例では、モジュールREN0045はネットワークのルートとして定義されています。モジュールREN0070、ADM0080、REN0010、REN0020はリーフ・モジュールです。



7. ルート・モジュールをクリックして選択します。
8. ユーティリティ→生成を選択します。「フォームの生成」ダイアログ・ボックスが表示されます。
9. 「生成」タブの「生成オプション」ドロップダウン・リストからForm and Associated Menuを選択します。
10. 生成を始めるために「開始」ボタンをクリックします。
ジェネレータはメニュー・モジュールを作成してから、フォームを生成します。
11. 生成されたフォームを実行します。

生成されたメニューの機能 モジュールを生成すると次に示すようなメニューができあがります。



トップ・レベル・メニュー上の項目 ジェネレータは、モジュール・ネットワーク内のルート・モジュールの下にあるモジュールからトップ・レベル・メニュー上の項目を導出します。メニュー項目の名前は、モジュールの短縮タイトルから導出されるか、これがない場合、モジュール短縮名の最初の15文字になります。

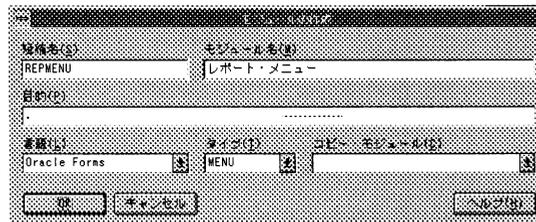
デフォルトでは、ジェネレータはメニュー項目「短縮名」の一意な文字を大文字に変換します。この動作はMNUULC作業環境によって制御されます。大文字にされた文字は、メニュー上のすべての項目について大文字にされた文字の間で一意になります。この一意の文字は、メニュー項目を選択するのに使用できます。一意な文字が存在しない場合、ジェネレータは「短縮名」の最初の文字を大文字にします。

標準メニュー項目 モジュール・ネットワーク内のモジュールから導出される項目に加え、キー一覧と終了がトップ・レベル・メニュー上の項目として表示されます。これらの標準項目を生成されたメニューに組み込むには、それぞれMNUSKYとMNUEXIの作業環境を使用します。

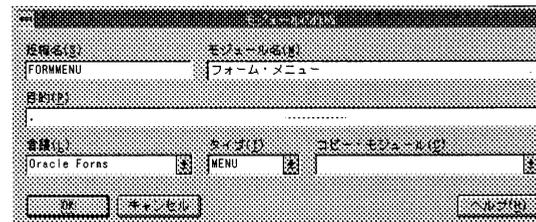
サブメニューの生成

別々のメニューで呼び出されるモジュールをグループ化したい場合があります。たとえば、画面オプションとレポート・オプションが別々のメニューにあると、メニューがもっと直観的に使えるようになるでしょう。

1. モジュール・ストラクチャ・ダイアグラムにモジュール・ネットワークが表示された状態で、レポート・モジュールの親になるモジュールを定義するために「メニュー・モジュールの作成」ツールバー・ボタンを選択します。
2. ダイアグラム上の空の領域をクリックします。
「メニュー・モジュールの作成」ダイアログ・ボックスが表示されます。
3. メニュー・モジュールについて適切な短縮名とモジュール名を入力します。

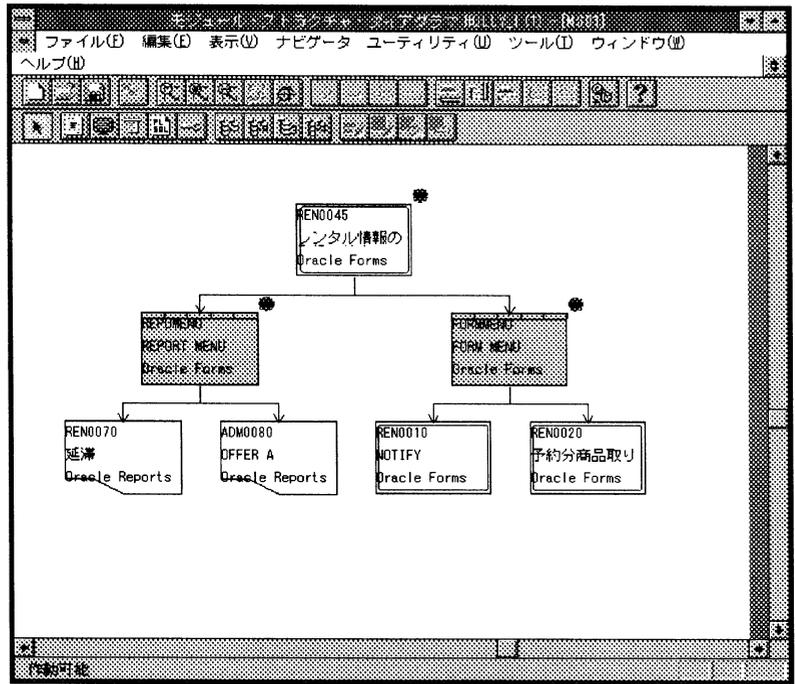


4. 生成されたメニューに表示したいレポート・モジュールをモジュール・ネットワーク内の新しいモジュールの下へ移動します。
5. 「メニュー・モジュールの作成」ボタンを使って、画面モジュールの親になるメニュー・モジュールを作成します。
6. メニュー・モジュールについて適切な短縮名とモジュール名を入力します。



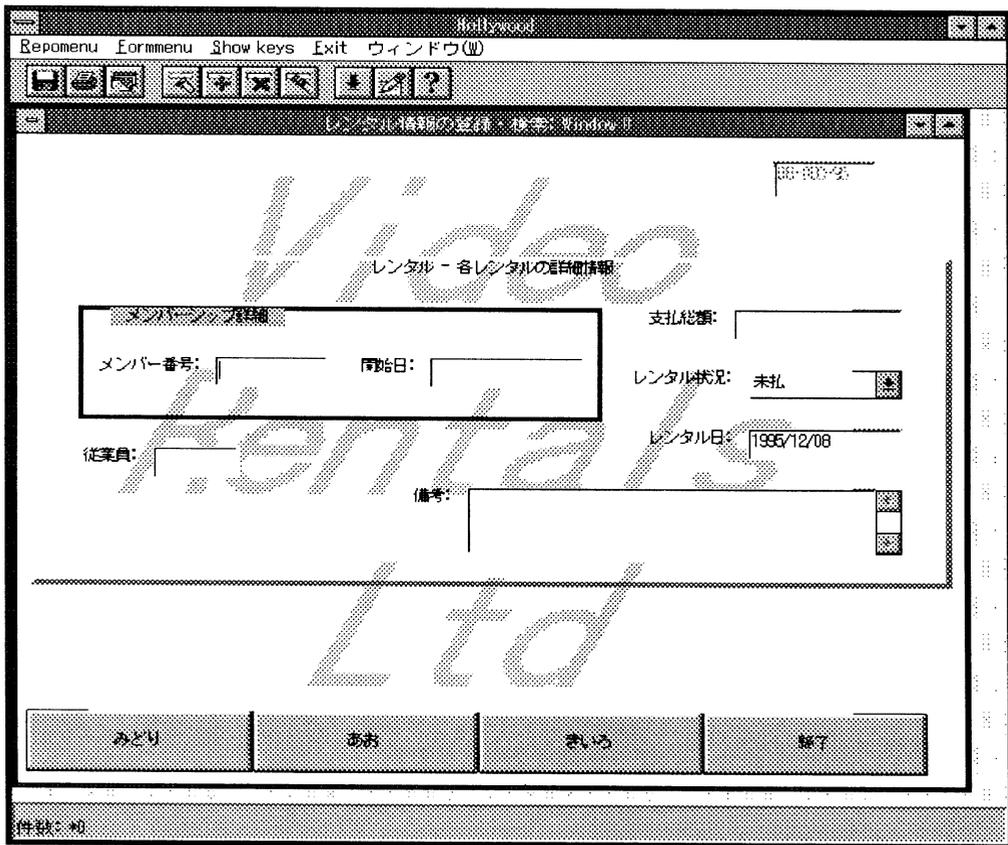
7. 生成されたメニューに表示したいスクリーン・モジュールをモジュール・ネットワーク内の新しいモジュールの下へ移動します。
8. 2つのメニュー・モジュール（およびそれらの子モジュール）をもとのルート・スクリーン・モジュールの下へ移動します。

モジュール・ネットワークは次のようになります。



9. ルート・モジュールを選択するためにクリックします。
10. ユーティリティ作成を選択します。「フォームの生成」ダイアログ・ボックスが表示されます。
11. 「生成オプション」ドロップダウン・リストからForm and Associated Menuを選択します。
12. 「開始」ボタンをクリックすると生成が行われます。
13. 生成されたフォームを実行します。

生成されたメニューの機能 モジュールを生成すると次に示すようなメニューができあがります。



トップ・レベル・メニュー上に2つのメニュー項目が作成されました。
REPOMENUは、レポートのモジュールを示すサブメニューです。
FORMMENUは、スクリーンのモジュールを示すサブメニューです。

ナビゲート可能モジュールのポップリストの作成

モジュール・ネットワーク内の他のモジュールにアクセスするための方法の1つはメニューです。このほかに、ユーザーが起動できるフォームやレポートのポップリストを表示する方法があります。

ポップリストは、モジュール・ネットワーク内で現在のフォームが生成されたモジュールのすぐ下にあるスクリーンまたはレポートのモジュールを示すだけであることに注意してください。ここで示す例では、前の節で作成したメニュー・モジュールは取り除かれています。

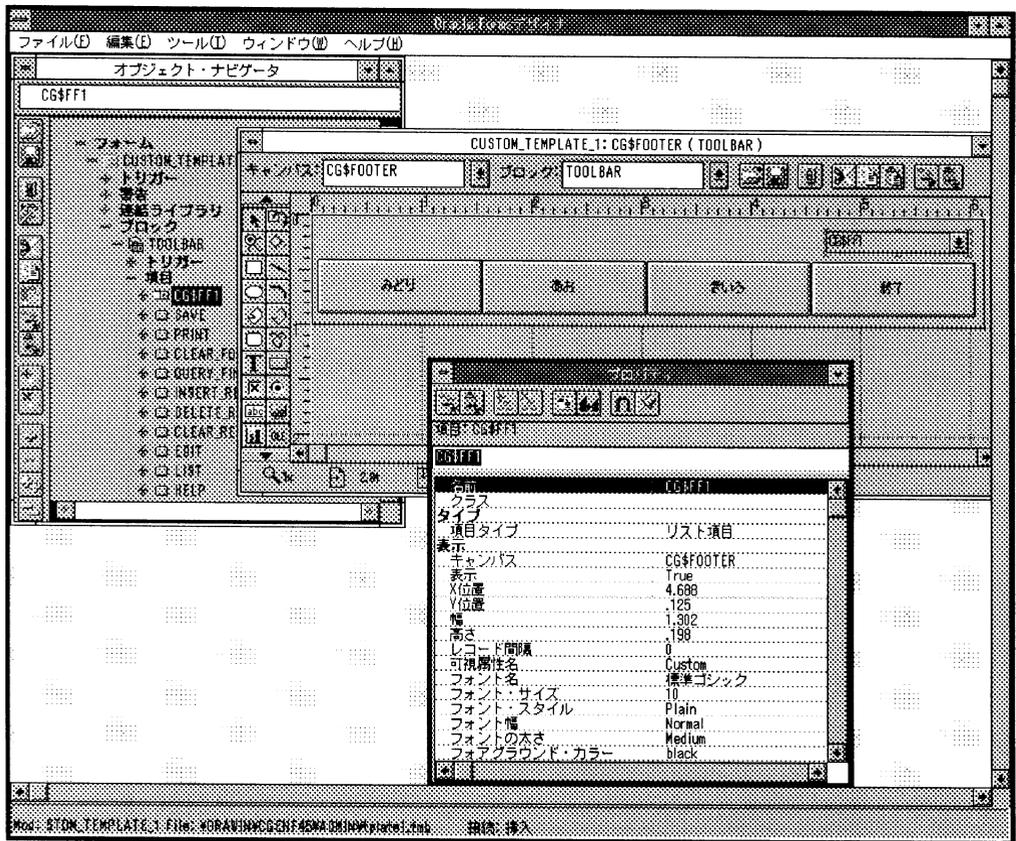
このようなリストが入ったフォームを生成するには、テンプレート・フォームのヘッダーまたはフッターの中にCG\$FF1と呼ばれるポップリスト項目を作成します。フォームを生成すると、リスト内にそのフォームから呼出可能なすべてのフォームとレポートのリストが表示されます。

1. モジュール・ストラクチャ・ダイアグラムにモジュール・ネットワークが表示された状態で、Oracle Forms Designerをオープンするためにツール→Oracle Forms Designerを選択します。
2. ファイル→オープンを選択して、生成するモジュールが使用するテンプレートをオープンします。

このファイルの場所はインストレーションによって異なります。通常はC:\ORAWIN\CGENF45\ADMINにあります。

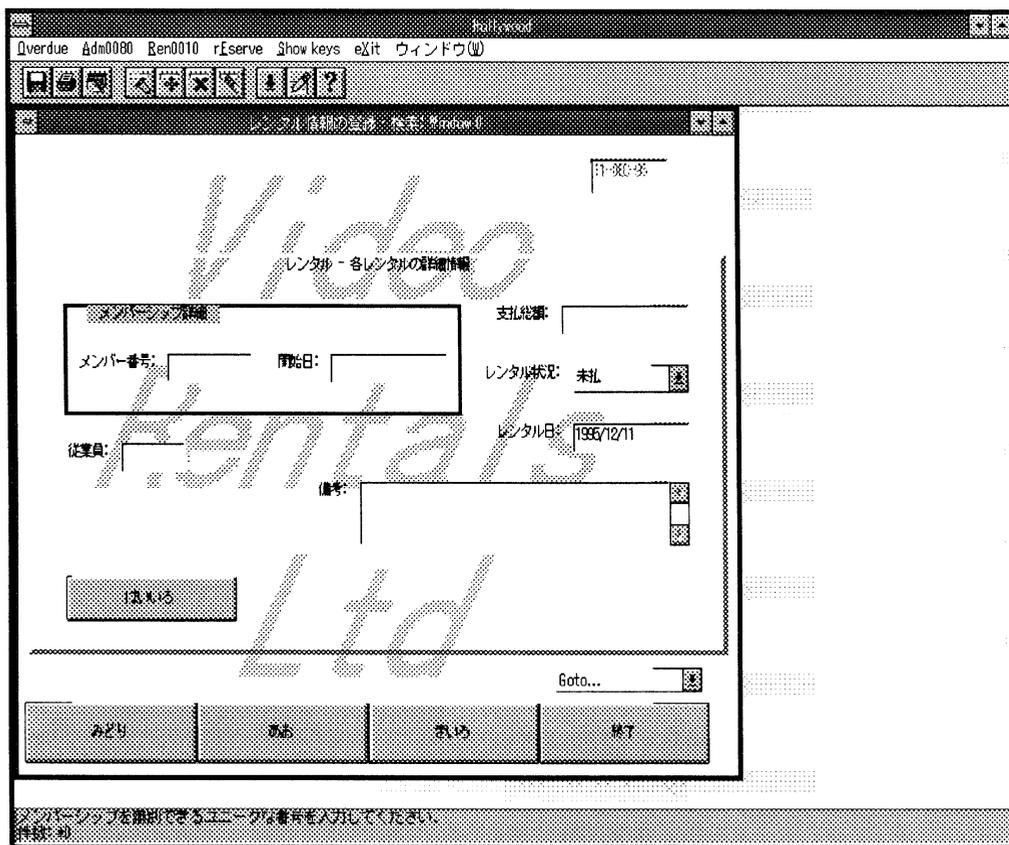
注意: モジュールが付属のテンプレート・フォームを使用している場合、そのテンプレート・フォームをコピーし、コピーを変更します。

3. CG\$FOOTERキャンバスで、(ポップリストのリスト形式で) CG\$FF1というリスト項目を作成します。
4. リスト項目のデフォルト値としてGoto...とタイプします。



5. テンプレート・フォームの変更を保存し、Oracle Forms Designerを終了し、モジュール・ストラクチャ・ダイアグラマに戻ります。
6. ルート・モジュールを選択するためにクリックします。
7. ユーティリティ→生成を選択します。「フォームの生成」ダイアログ・ボックスが表示されます。
8. 「生成オプション」ドロップダウン・リストからForm and Associated Menuを選択します。
9. 生成を始めるために「開始」ボタンをクリックします。
10. 生成されたフォームを実行します。

生成されたメニューの機能 モジュールを生成すると次に示すようなメニューができあがります。



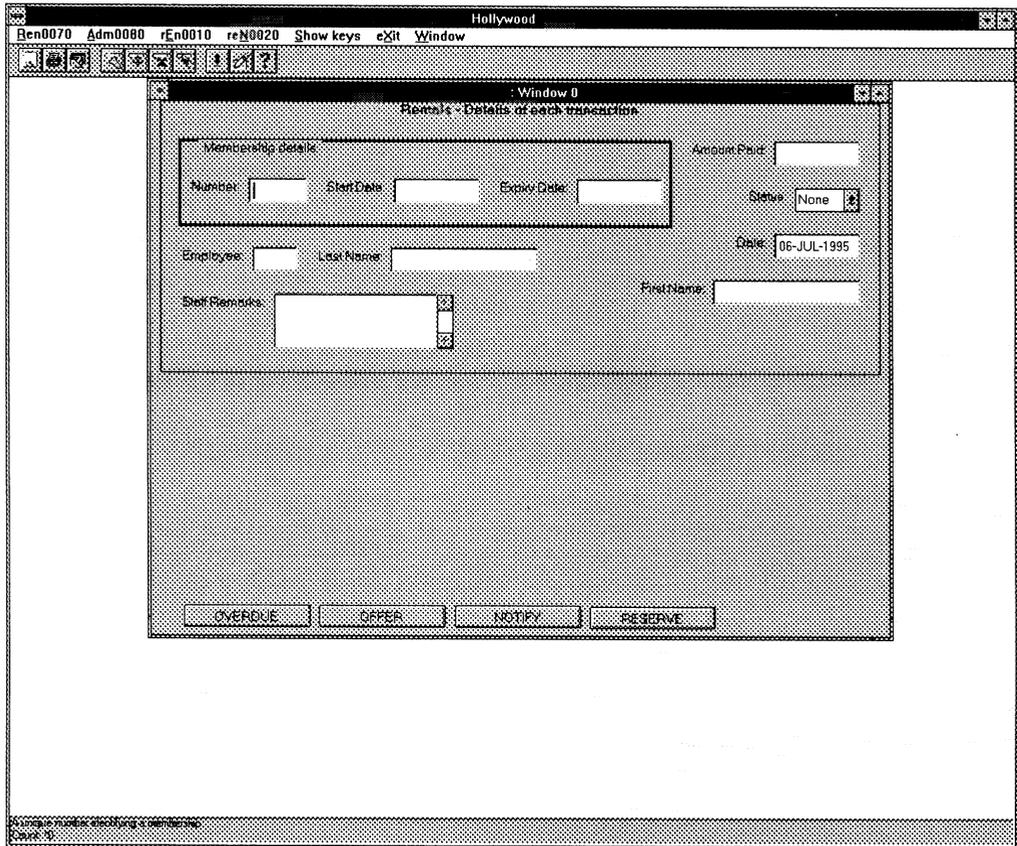
Oracle Forms Designerのポップリストに与えたデフォルト値が、ポップリストの最初のエン트리として表示されます。

リスト内のエント리는、モジュール・ネットワーク内の現在のモジュールの下にあるモジュールです。

ナビゲーション・ボタンの生成

Oracle Formsジェネレータに付属のOFG4PC2T標準テンプレートを使用すると、リスト項目のかわりに他のフォームやレポートを呼び出すためのボタンを生成することができます。

テンプレート内のフッター上の最大5つのボタンが、各ナビゲーション・オプションについて使用可能になり、それによってラベルが付けられます。テンプレート内のコードは、リスト内の値をボタンに置き換えます。リスト内の項目の数が5つを超えている場合、5番目のボタンにはその他というラベルが付き、他のモジュールを表示するためのリストを呼び出します。



注意: モジュール・ネットワーク内に子モジュールが定義されているモジュールから生成するときだけ、OFG4PC2T.FMBテンプレートを使用します。生成されるキャンバス上にボタンを組み込むには、生成されるページの幅を指定します。これは、テンプレート内のCG\$FOOTER_1キャンバスの幅より広くします。

テンプレート・メニューを使ったメニューの生成

Oracle Forms Designerを使ってテンプレート・メニュー・アプリケーションを作成し、Formsジェネレータで生成したメニューとマージすることができます。

テンプレート・メニューを作成し、生成されたメニューとマージするには、

1. モジュール・ストラクチャ・ダイアグラマがモジュール・ネットワークを表示している状態で、Oracle Forms Designerをオープンするためにツール→Oracle Forms Designerを選択します。
2. 新しいメニュー・モジュールを作成します。
3. モジュール・ネットワーク内のモジュールから導出したオプションに加えて生成されたメニュー上に表示したいメニュー項目を指定します。
4. 適切なコマンド・テキストを入力します。

たとえば、SET_CANVAS_PROPERTY Formsビルトインを使ってColorsメニューによってフォームのバックグラウンド・カラーを変更したいとします。Colorsメニューのメニュー・オプションには、次のコマンド・テキストが考えられます。

```
set_canvas_property('CG$PAGE_1',visual_attribute,
'BLUE_BACKGROUND');
```

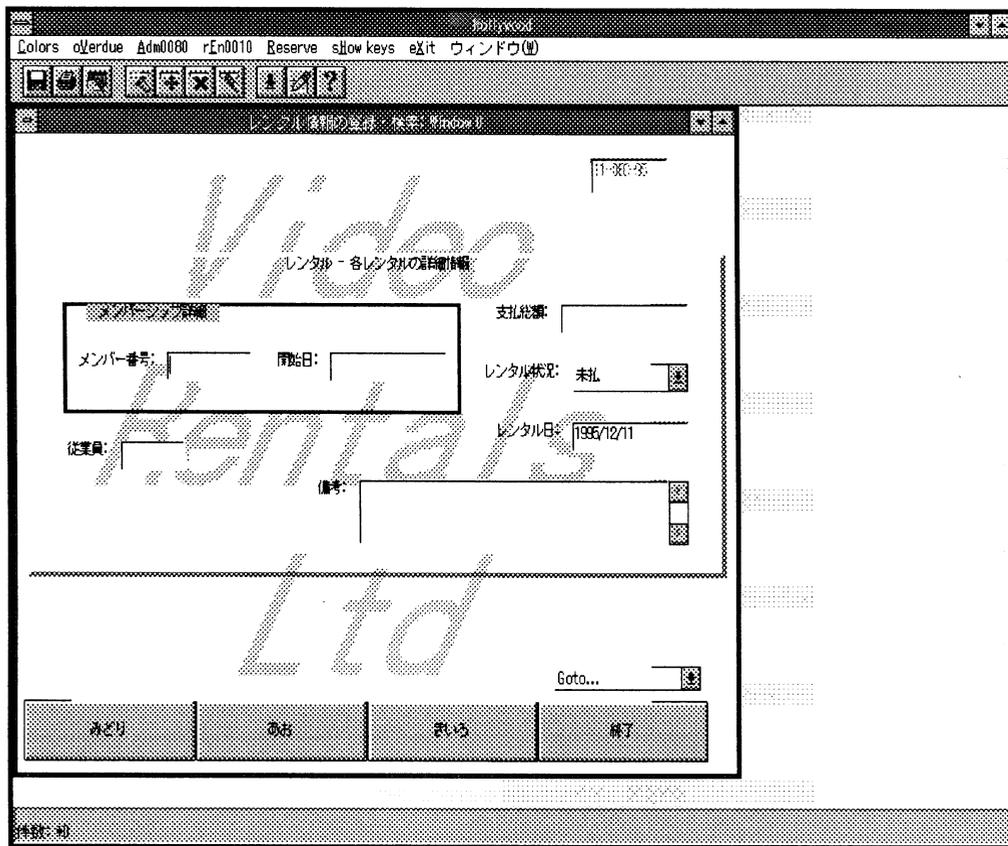
注意: この例では、Colorsメニュー上のオプションによって呼び出される可視属性はテンプレート・フォームで定義されます。



5. テンプレート・メニューを保存し、モジュール・ストラクチャ・ダイアグラムに戻ります。
6. ツール→プリファレンス・ナビゲータを選択します。
7. 「MENU-TEMPLATE」作業環境グループのSTMMMB作業環境の値を作成したテンプレート・メニュー・モジュールの名前に変更します。
8. プリファレンス・ナビゲータ内の変更を保存し、モジュール・ストラクチャ・ダイアグラムに戻ります。
9. ルート・モジュールを選択するためにクリックします。
10. ユーティリティ→生成を選択します。「フォームの生成」ダイアログ・ボックスが表示されます。
11. 「テンプレート」タブで、「位置」がファイル、「パス」が作成したテンプレート・メニューの場所に設定されていることを確認します。
12. 「生成」タブで、「生成オプション」ドロップダウン・リストからForm and Associated Menuを選択します。
13. 「開始」ボタンをクリックすると生成が行われます。

14. 生成されたフォームを実行します。

生成されたメニューの機能 モジュールを生成すると次に示すようなメニューができあがります。



Oracle Forms Designerで作成したテンプレート・メニューは、Formsジェネレータによって生成されたメニューとマージされます。

Colorsメニュー上の各メニュー項目はSET_CANVAS_PROPERTY Formsビルトインを呼び出し、テンプレート・フォームについて定義された可視属性を指定します。

生成されたメニュー項目に対するテンプレート・メニュー項目の位置は、MNUSGA作業環境によって制御されます。

第11章

再生成

この章では、再生成の利点とプロセスについて説明します。この章は次の節からできています。

- 再生成の紹介
- フォームを再生成する方法
- 再生成時のリポジトリの更新
- 再生成時の新しいブロックと項目の追加
- 再生成時の既存のブロックと項目の属性の変更
- 再生成とテンプレート
- 再生成時の生成されたコードの更新
- 再生成時の新しいメニューの生成

再生成の紹介

再生成とは何か？

再生成とは、Oracle Forms Designerを使って定義または変更された細目を残したまま、リポジトリに記録された新しいまたは異なる要件を取り込むために、生成されたOracleFormsアプリケーション（すなわち、OracleFormsジェネレータによって作成されたもの）を再び生成するプロセスです。

このプロセスは次のステップから構成されます。

- リポジトリの更新
- 新しい表取扱細目と列取扱細目によって指定された新しいブロックと項目の追加
- 生成されたトリガー・コードの更新
- 新しいメニューの生成

なぜ再生成が必要なのか？

Designer/2000のツールを使用すると、生成後の変更がほとんど必要ないフォームが生成できるような情報を記録することが可能です。

しかし、リポジトリを使って可能なすべての機能要件を定義することはできません。特に次のいずれかの要件がある場合、生成されたフォームをOracle Forms Designerで変更することが必要になります。

- レイアウトの微調整
- 特殊なユーザー・インタフェース要件
- アプリケーション固有のビジネス・ロジック

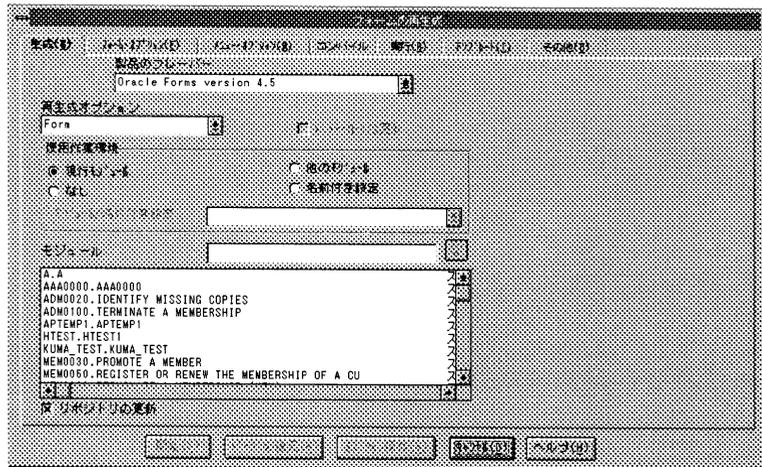
再生成の必要性を減らす

再生成の必要性を減らすには、

- 生成されたフォームに組み込みたいオブジェクトやコードは、テンプレートを使って組み込みます。
- アプリケーション固有のビジネス・ロジックを実装するには、データベース・トリガーを定義します。
- 妥当性チェックと複雑な導出に関する事前定義された関数を起動するには、COMPLEX:<function call>機能を使用します。

フォームを再生成する方法

1. モジュール・データ・ダイアグラマを起動します。
2. ユーティリティ→フォームの再生成を選択します。
「フォームの再生成」ダイアログ・ボックスが表示されます。



3. 再生成したいFormsモジュールを指定します。
4. 「開始」ボタンをクリックすると、再生成が行われます。

再生成時のリポジトリの更新

再生成の際、ジェネレータは最初にフォームの既存の定義をリポジトリ内にリバース・エンジニアし、モジュール定義と取扱情報を必要に応じて更新します。

注意: ジェネレータは、モジュール・ネットワーク定義またはモジュール・アクセス細目を更新しません。

再生成時の新しいブロックと項目の追加

新しいブロックと項目の作成 ジェネレータは、既存のブロックに対応していない任意の基礎表用の取扱細目について新しいブロックを作成します。ジェネレータが新しいブロックを作成する場合、次の列取扱細目に基づいて新しい項目も作成します。

- 新規基礎表取扱
- 参照表取扱

非基礎表項目

ジェネレータは再生成時に新しい非基礎表項目を生成します。

- 既存の基礎表取扱について新しい参照表取扱が定義されている場合
- 新しい基礎表取扱にサマリー項目を指定する列取扱細目が入っている場合

ジェネレータは、再生成時に作成した非基礎表項目を他のすべてのページの後にあるオーバーフロー・ページに配置します。フォーム内の各ブロックは、自身のオーバーフロー・ページを持っている場合があり、そこには1つまたは複数の非基礎表項目が入っています。

オーバーフロー・ページに配置された項目の位置を変更するには、Oracle Forms Designerを使用します。

新しいブロックのための新しいページ、ポップアップ、ウィンドウの作成

ジェネレータは、最初に新しく生成されたブロックを常に新しいウィンドウの新しいページに配置します。以後新しく生成されたブロックは、表取扱細目について指定された位置に従って配置されます（通常の生成における生成ブロックの位置に関する規則に準じます）。

ジェネレータについて次のことに注意してください。

- 新しく生成されたブロックを既存のページまたはポップアップ上に配置することはありません。
- 新しく生成されたページやブロックを既存のウィンドウ内に配置することはありません。

新しく生成されたページ、ポップアップ、ウィンドウに名前を付ける場合、ジェネレータは、既存のオブジェクトと重複しない最も小さな順序番号を使用します。たとえば、フォームにCG\$PAGE_1、CG\$PAGE_3、CG\$WINDOW_9が入っている場合、最初の新しいブロックは、CG\$WINDOW_1というウィンドウ内のCG\$PAGE_2に作成されます。

ウィンドウ、ページ、ポップアップの名前によって、ジェネレータがテンプレート・オブジェクトを適用する方法が異なります（通常の生成における生成ブロックの位置に関する規則に準じます）。

オーバーフロー・ページ

ジェネレータが新しい非基礎表項目を保持するためにオーバーフロー・ページを作成する場合、最初のオーバーフロー・ページの名前はCG\$OVERFLOW_1、2番目はCG\$OVERFLOW_2というようになります。

オーバーフロー・ページは、そこにいれる項目にとって最低限十分な大きさになります。ビュー・サイズはキャンパス・サイズと同じです。あるブロックに関するオーバーフロー・ページは、そのブロックの他の項目が配置されるページと同じウィンドウに配置されます。

再生成時の既存のブロックと項目の属性の変更

既存のブロックの属性の変更 再生成の際、ジェネレータは既存のブロックの次の属性を変更できます。

- 表所有者属性 (TABOWN作業環境の値に基づく)
- 列セキュリティ属性 (COLSEC作業環境の値に基づく)

既存の項目の属性の変更 再生成の際、ジェネレータは既存の項目の次の属性を変更できます。

- 最大値および最小値の属性 (DVHILO作業環境の値に基づく)
- 項目に関する一覧表用のSQL文 (項目に関して既存の非生成SQL文がある場合を除き、外部キー項目と既知の値のリストに対して妥当性検査される項目について再生成中にメンテナンスされます)
- コピー項目属性

既存のポップ・リストとテキスト・リスト ジェネレータは、再生成中にポップリストやテキスト・リストの位置や寸法を変更しません。しかし、リストの値は、それが基づいている列またはそのドメインに関するリポジトリに記録された有効な値から再移入されます。

既存のラジオ・グループとチェックボックス ジェネレータは、既存のラジオ・グループとチェックボックスを再生成しません。

既存のボタン ジェネレータは、コードを再作成しますが、ボタンそのものは変更しません。

既存のOLEコンテナ・カスタム項目 OLEコンテナ・カスタム項目のプロパティは再生成時に変化しません。OLEコンテナ・カスタム項目に対して生成されたトリガー・コードは、必要に応じて再生成されます。

既存のVBXコントロールカスタム項目 再生成時にジェネレータは、(FLDCMT作業環境の設定には関係なく) テンプレート実現項目のコメント・マッピング行を生成されたVBXコントロールカスタム項目のコメント内に組み込みます。これは、VBXコントロールカスタム項目を再生成するときにVBプロパティ・マッピング・コメントがジェネレータにとって利用可能なことを意味します。範囲下限や範囲上限などのテキスト項目プロパティが再生成できるので、VBXコントロールについて関連してプロパティの正しい再生成が簡単になります。

したがって、再生成時に列取扱細目に対して指定されたインプリメンテーションは、表示データ型がVBXコントロールの列取扱細目をテンプレート・フォーム内のVBXコントロールカスタム項目とリンクするのに必要ないため、ジェネレータによって無視されます。必要な変更は、Forms Designerを使って生成されたフォームのVBXコントロールに対して直接行う必要があります。

新しい基礎表VBXコントロールがフォームに追加され、フォームを再生成したい場合、Forms Designerを使ってVBXコントロールのコメントにマッピング行を追加しなければなりません。これを行わないと、VBXコントロールに関するプロパティが正しく再生成されません。

コメントの変更 ジェネレータは、自身のブロックレベルまたはフォームレベルのコメントを次のようにして変更します。

- MODIFIEDセクションへの行の追加

- ジェネレータによる警告の追加
- 必要に応じた表および列取扱情報の変更
- 必要に応じた参照整合性コメントの変更

**変更履歴、ジャーナルおよび
列導出要件の変更**

再生成時、ジェネレータはリポジトリ内の表と列の定義に施された変更を検出した場合、次のことを行います。

- 列に関する変更履歴をメンテナンスするためにコードを追加または削除します。

変更履歴列に関する表示項目を入力可能と定義した場合、再生成時にジェネレータは警告を発行し、項目を入力不可にします。

- ジャーナルをサポートするためにコードを追加または削除します。

JNNTRQ作業環境の値は再生成に影響を与えません。ジェネレータは、最初に必要に応じて注記を入力するための非基礎表項目を探します。この項目が見つからない場合、警告を発行してから作成します。

- リポジトリの列導出情報が変更されている場合、前に生成されたトリガーを削除し、新しい情報で再び生成します。

すべての生成されたトリガーを保存し、新しい導出コードの生成を停止するには、REGNCD作業環境をNに設定します。

再生成とテンプレート

テンプレートから作成される既存の項目	ジェネレータは、テンプレートから導出される既存の項目を取り除いたり、場所を変更したりしません。 しかし、動的ジェネレータ項目（すなわち、次のページ・ボタン、前のページ・ボタン、問合せモード・ボタン、問合せモード・チェックボックス、問合せモード・テキスト項目）から導出された項目のコードは、動作と外観が再生成したフォームの内容ときちんと同じになるように再生成されます。
既存のキャンバスとウィンドウ	これらのオブジェクトは変更されません。テンプレート・キャンバスが適用されることはありません。
新しく生成されたキャンバス	新しいキャンバスが生成された場合、ジェネレータはテンプレートのヘッダーとフッターまたはベース・キャンバスを通常の生成時とまったく同様に使用します。
新しく生成されたウィンドウ	新しいウィンドウが生成された場合、ジェネレータはテンプレートのスタック・ヘッダーとフッター・キャンバスを通常の生成時と同様に使用します。
テンプレート・フォーム内の新しいユーザー・キャンバス	新しいユーザー・キャンバス（すなわち、生成されたフォーム内にかつてコピーされなかったもの）は、生成されたフォーム内にコピーされません。
ユーザー・オブジェクト	ユーザー・オブジェクトは再生成時にコピーされません。
ツールバー	テンプレートのツールバー・キャンバスは、新しいウィンドウを生成するときだけ使用されます。CG\$WINDOW_n（nはつぎに利用可能なウィンドウ番号）を生成するとき、ジェネレータは、テンプレート・フォーム内でCG\$HORIZONTAL_TOOLBAR[_n]またはCG\$VERTICAL_TOOLBAR[_n]というツールバー・キャンバスを探し、新しいウィンドウのツールバーとして連結します。

再生成時の生成されたコードの更新

生成されたコードの再生成	<p>再生成時にジェネレータは、データベース設計情報の変更または次の結果をとまなう作業環境値の変更を組み込むために、最も前に生成されたコードを追加、削除、または変更することができます。</p> <ul style="list-style-type: none">• 参照整合性規則の変更• 制約定義の追加または削除• 使用するブロックの同期のタイプ• オンライン・ヘルプ・サブシステムの組み込み• 項目位置の変更 <p>生成されたコード・グループは、コード・グループ識別子コメント（例、<code>/*CGRI\$CHECK_ON_DELETE*/</code>）によって識別されます。このコメントが再生成中に見つかったら、ジェネレータはその中（BEGINとENDの間、BEGINとENDを含む）のコードを削除し、コード・グループを再作成するときに、そのコードを残った識別子コメントの位置に追加します。</p> <p>作業環境を使用して再生成の範囲をコントロールすることができます。</p> <p>ジェネレータは次の生成されたコードは変更できません。</p> <ul style="list-style-type: none">• サマリー導出コード• 非データベース項目導出コード• ユーザー、日付、「コンテキスト項目」を移入するためのコード
非生成コード	<p>ジェネレータは、再生成時に非生成コードを変更しません。したがって、ジェネレータを使って、生成されたコード・ブロックの周囲に追加したコードに影響を与えることなく、生成されたコードをメンテナンスできます。</p>
再生成時のコードの保持	<p>生成されたコード・グループを変更し、再生成時にコードを保持したい場合、一意コード・グループ識別子コメントにテキストを追加します。追加するテキストは次のいずれかでなければなりません。</p> <ul style="list-style-type: none">• 英数字• 下線 (_)• ドル (\$) <p>生成されたコードに対する変更を保持する必要がなくなった場合、一意コード・グループ識別子コメントに追加したテキストを取り除き、フォームを再生成するだけでかまいません。</p> <p>注意: コメントの後にテキストを追加する以外の方法でコメントを変更した場合（例、<code>/*CGRI\$CHECK_ON_DEL*/</code>）、コード・グループは削除されませんが、同じタイプの新しいコード・グループが作成されます。</p> <p>新しいプロシージャまたはファンクションは、最初の行にその名前が入っているだけのコメントを付けて作成する必要があります。たとえば、</p>

```

/* CGTE$CHECK_CONSTRAINT_VIO */
FUNCTION CGTE$CHECK_CONSTRAINT_VIO(
    P_SERVER_ERR IN NUMBER
    ,P_SERVER_MSG IN VARCHAR2)
RETURN BOOLEAN IS
    constraint_name VARCHAR2(61); /* Violated constraint */
BEGIN
    ...
END;

```

コード・グループと同じ方法でコメントを変更する場合、プロシージャまたはファンクションは再生成時に再作成されません。

ジェネレータは、コードを保持しなかった場合に施される変更を詳しく説明したレポートを作成することができます。これには、「フォームの再生成」ダイアログ・ボックスの「再生成レポート」オプションを設定します。

保持されたコードの表示

コードを保持しなかった場合に施される変更を見ることができると便利です。ジェネレータが施すはずだった変更をリストしたレポートを作成するには、「フォームの再生成」ダイアログ・ボックスの「再生成レポート」オプションを設定します。

次のレポート例は、ユーザーの要求によって保持されている生成されたコードに施されるはずだった変更を詳しく記述しています。前に'<'が付いているコードは、今回の実行でジェネレータが作成するはずだったコードです。前に'>'が付いているコードは、保持されているユーザーが変更したコードです。

```

Trigger ON-CLEAR-DETAILS.CGBSSON_CLEAR_DETAILS
< /* CGBSSON_CLEAR_DETAILS */
< /* clear all detail blocks for the given master block */
< BEGIN
<   IF (FORM_FAILURE AND :SYSTEM.COORDINATION_OPERATION IN ('MOUSE',
<     'DUPLICATE_RECORD')) THEN
<     RAISE FORM_TRIGGER_FAILURE;
<   END IF;
<   IF (:SYSTEM.MASTER_BLOCK = :SYSTEM.TRIGGER_BLOCK) THEN
<     CGBS$.CLEAR_MASTER_DETAIL( :SYSTEM.MASTER_BLOCK,
<       :SYSTEM.COORDINATION_OPERATION );
<   END IF;
< END;
---
> /* CGBSSON_CLEAR_DETAILS_modified */
> /* clear all detail blocks for the given master block */
> BEGIN
>   IF (FORM_FAILURE AND :SYSTEM.COORDINATION_OPERATION IN ('MOUSE',
>     'DUPLICATE_RECORD')) THEN
>     RAISE FORM_TRIGGER_FAILURE;
>   END IF;
>   IF (:SYSTEM.MASTER_BLOCK = :SYSTEM.TRIGGER_BLOCK) THEN
>     CGBS$.CLEAR_MASTER_DETAIL( :SYSTEM.MASTER_BLOCK,
>       :SYSTEM.COORDINATION_OPERATION );
>   /* My alternative code */
>   else
>     do_something_different(:SYSTEM.MASTER_BLOCK);
>   /* end of modification */
>   END IF;
> END;

```

再生成時の新しいメニューの生成

フォームを再生成するときに新しいメニューを生成するには、「フォームの再生成」ダイアログ・ボックスのRegenerate Form and its Associated Menuオプションを選択します。

注意: ジェネレータはメニュー・モジュールを再生成しません。再生成されたフォームのメニューは、リポジトリ内の情報だけに基づいています。

第12章

フォームのリバース・エンジニアリング

この章では、リバース・エンジニアリングのプロセスについて説明します。この章は次の節からできています。

- リバース・エンジニアリングの紹介
- フォームをリバース・エンジニアリングする方法
- リバース・エンジニアリング・プロセスで何が記録されるか？
- リバース・エンジニアリングとリエンジニアリング

リバース・エンジニアリングの紹介

リバース・エンジニアリングとは何か？ リバース・エンジニアリングは、既存のフォームを読み込み、対応するリポジトリ・モジュール定義を作成または更新するプロセスです。

具体的には、リバース・エンジニアリング・ユーティリティは、フォームから次の情報を抽出し、リポジトリに記録します。

- モジュールの定義と説明
- 表および列の取扱サマリー
- 基礎表取扱と基礎列取扱

ユーティリティは、リバース・エンジニアリング・プロセス中にジェネレータが下す決定に関する情報を提供するログも作成します。

リバース・エンジニアリング時にジェネレータは、フォーム定義からデータベース設計を推論しようとはしません。フォームが使用するオブジェクトの定義がリポジトリ内に存在しない場合、Designer/2000 データベースのリバース・エンジニアリングユーティリティを使って、データベースから細目を読み込み、リポジトリに記録します。

注意: Oracle Formsメニュー・モジュールはリバース・エンジニアリングできません。

なぜリバース・エンジニアリングを使用するのか？ リバース・エンジニアリングは次のような目的で行われます。

- 既存のシステムをリポジトリ内に文書化するため
- 影響分析を使用可能にするため
- 従来システムをメンテナンスするため
- リポジトリの定義を実世界と調和させるため
- リエンジニアリングのための基礎を提供するため

リバース・エンジニアリングの前に フォームをリバース・エンジニアリングするには、以下の条件を満たすことが必要です。

- リバース・エンジニアリングするフォームにアクセスできること。(フォームは、Oracle Forms Version 4.5を使って構築されたものでなければなりません。これより前のバージョンのOracle Formsで作成されたフォームをリバース・エンジニアリングするには、まずそのフォームをVersion 4.5のフォームに変換しなければなりません)。
- リポジトリ内のアプリケーション・システムへの書込権限を持っていること。
- フォームが参照するすべてのデータベース・オブジェクトに関してリポジトリ定義がされていること。

リバース・エンジニアリングする前に次のものをリポジトリに記録することをお奨めします。

- すべての主キーと外部キー定義
- 列定義に関する通常の表示プロンプト。これは、ジェネレータがもとのフォームのプロンプトをリバース・エンジニアリングせず、かわりに関連する列定義の通常の表示プロンプトを使用するからです。

フォームをリバース・エンジニアリングする方法

Oracle Forms Designerを使って作成し、リポジトリ内にリバース・エンジニアリングする簡単なマスター/ディテール・フォームの例を次に示します。

ディーラー		
番号	ディーラー	地域
100	ピックアップ自動車	440

オーダー						
番書	数量	日付	参照番号	モデル	仕入価格	販売価格
85238	1	92-08-09	2105	22	28342	29453
87930	5	92-07-31	975	14	8215	8739
87930	6	92-07-31	975	14	5115	17000
87940	9	92-07-31	975	11	5390	6500
98760	2	92-11-27	60702	1	13294	15322

検索条件の入力 検索の実行 終了

Serverジェネレータを使って、フォームが参照するデータベース・オブジェクト（この例ではディーラー表とオーダー表）をリポジトリ内にリバース・エンジニアリングしたら、フォームそのものをリバースエンジニアするために次のことを行います。

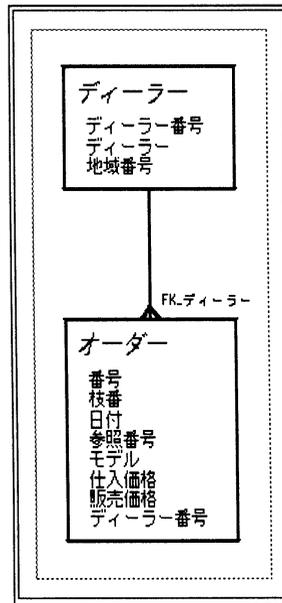
1. モジュール・データ・ダイアグラマを起動します。

- ユーティリティ→リバース・エンジニア→フォームを選択します。
「フォームのリバース・エンジニア」ダイアログ・ボックスが表示されます。



- 既存のFormsモジュールの位置を指定します。
- リバース・エンジニアリングした結果をいれるリポジトリ・モジュール（新しいモジュールまたは既存のモジュール）の名前を入力します。
- リバース・エンジニアリングするフォームの名前を指定します（フォームの位置としてファイルを指定した場合、「ファイル」フィールドを使ってフォームを識別します。データベース・オブジェクトを指定した場合、「データベース」タブを使ってフォームを識別します）。
- 「開始」ボタンをクリックすると、リバース・エンジニアリング処理が行われます。
- リバース・エンジニアリング処理が完了したら、メッセージ・ダイアログ・ボックスをクローズします。
- 新しいダイアグラムを作成し、「新規ダイアグラム」ダイアログ・ボックスからリバース・エンジニアリングしたモジュールを選択します。

ダイアグラムにリバース・エンジニアリングしたモジュールのデータ取扱細目
目が表示されます。



リバース・エンジニアリングしたモジュールの機能 リバース・エンジニアリングしたモジュールは次の機能を持っています。

- フォームが使用する各基礎表の取扱細目（すなわち、表に基づく各ブロックについて1つの取扱細目）
- それぞれの表取扱細目の列取扱細目はデフォルト・プロパティを持っています。
- すべてのトリガーとプログラム単位コードで使用される表と列のサマリー取扱（連結されたファイル内のコードが参照する表と列の取扱はありません）
- 基礎表取扱の取扱サマリー

プロパティがどのように導出されるかについては、次の節「リバース・エンジニアリング・プロセスで何が記録されるか？」を参照してください。

リバース・エンジニアリング・プロセスで何が記録されるか？

リバース・エンジニアリング時にジェネレータは、次の表と列取扱のモジュール定義とプロパティを作成および変更します。

リバース・エンジニアリングされるモジュール定義

リポジトリ・モジュール定義	割り当てられる値/ソース
短縮名	「フォームのリバース・エンジニア」ダイアログ・ボックスで指定されたモジュール短縮名が設定されます。
名前	最初はモジュール短縮名と同じと見なされます。しかし、既存のモジュールの名前と重複する場合、新しい名前を入力を求められます。
目的	タイトルを持つフォームの最初のウィンドウのタイトルに設定されます。タイトルのあるウィンドウが見つからない場合、モジュール短縮名と同じになります。
言語	Oracle Forms
タイプ	スクリーン (SCREEN)
フォーマット	マスター・ディテール (MASTER DETAIL)
タイトル先頭	タイトルを持つフォームの最初のウィンドウのタイトルに設定されます。タイトルのあるウィンドウが見つからない場合、空白になります。
コメント	新しいモジュールを作成する場合、フォームレベルのコメントがモジュール記述行にコピーされます。
注記	ADNTRT 作業環境がYに設定されている場合、次のテキスト文字列がモジュールの注記行にコピーされます。 Oracleフォーム・ジェネレータ・バージョン<version>を使って<user>により<date>にリバースエンジニアリングされました。

プロセスが既存のリポジトリ・モジュールを更新する場合、ジェネレータはタイプと言語の値だけを入力または更新します。言語があらかじめSQL*Formsに設定されている場合、言語の値は更新されないことに注意してください。

リバース・エンジニアリングされる表取扱細目

新しく作成される表取扱細目のプロパティ

表取扱細目のプロパティ	割り当てられる値／ソース
順序	前のブロックにマップされ、1だけインクリメントされる表取扱細目の順序番号。前のブロックがない場合、番号は1になります。必要に応じて、以後の表取扱細目の順序番号は新しい表取扱のスペースを確保するためにインクリメントされます。
削除フラグ	ブロックのDELETE属性に合わせて設定されます。
挿入フラグ	ブロックのINSERT属性に合わせて設定されます。
問合せフラグ	ブロックのQUERY属性に合わせて設定されます。
更新フラグ	ブロックのUPDATE属性に合わせて設定されます。
表示タイトル	表定義に関する表示タイトルの値を取ります。これはリポジトリに記録されています。
WHERE/Validation 条件	<p>ブロックに関して記録されたデフォルトのWHERE句からコピーされます。ORDER BY句がある場合、表取扱から省略されます。かわりに列取扱細目についてORDER BYが設定されます。</p> <p>参照項目を含むWHERE句のリバース・エンジニアリングは、REWHFD作業環境で制御されます。REWHFDがNに設定されていると、警告メッセージを出し、WHERE句はリバースされません。REWHFDがYに設定されている場合は、警告を出した上でWHERE句をリバースし、次の文章がコメントに付記されます。</p> <p style="text-align: center;">"Rev engineered WHERE clause contains references to items"</p>
コメント	<p>"Reverse Engineered by <user> on <date>"に設定されます。</p> <p>参照項目を含むWHERE句がリバースされた場合、さらにコメントが付記されます。</p>
タイプ	基礎表
表示行	「スクリーン・モジュールの取扱細目状況」ダイアログ・ボックスの「表レイアウト」タブに記録されています。

ジェネレータは、ブロック内のCopyFrom属性を読み込み、その情報をこの表の外部キーと比較し、この表取扱細目から前の表取扱細目（新しく作成されたかどうかに関係なく）への適切な結合を作成します。

既存の表取扱細目について変更されたプロパティ

表取扱細目のプロパティ	割り当てられる値／ソース
更新フラグ	ブロックのUPDATE属性に合わせて設定されます。
挿入フラグ	ブロックのINSERT属性に合わせて設定されます。
削除フラグ	ブロックのDELETE属性に合わせて設定されます。
選択フラグ	ブロックのSELECT属性に合わせて設定されます。
WHERE句	ブロックに関して記録されたデフォルトのWHERE句からコピーされます。 注意： WHERE句に参照項目が含まれているかどうかに関わらず、常にコピーされます
型	基礎表
表示行	「スクリーン・モジュールの取扱細目状況」ダイアログ・ボックスの「表のレイアウト」タブに記録されています。

表取扱細目の順序プロパティは更新されません。ブロックの順序と対応していない場合、ジェネレータは警告を発行します。

リバース・エンジニアリングされる列取扱細目

列取扱細目のプロパティ	割り当てられる値/ソース
順序	前の項目にマップされ、1だけインクリメントされる列取扱細目の順序番号。前の項目がない場合、番号は1になります。必要に応じて、以後の列取扱細目の順序番号は新しい列取扱のスペース確保のためにインクリメントされます。
表示データ型	<p>基礎を形成する列の項目タイプとデータタイプをチェックすることによって設定されます。たとえば、数値列に基づくテキスト項目の場合、列取扱細目の表示データタイプはNUMBERに設定されます。</p> <p>基礎テーブルの表示項目またはユーザー領域では、これはUNKNOWNに設定されます。</p> <p>ラジオ・グループ、ポップアップ・リスト、テキスト・リストでは、ジェネレータは表示データタイプを次のように設定します。</p> <p>リストまたはグループ内の入力またはボタンのラベルが、下のカラムの許容値と同じ場合、表示データタイプはPOP-LIST、T-LIST、RADIO GROUPのいずれか適切なものに設定されます。</p> <p>そうでない場合、すべてのラベルが1語で6文字より短い場合、表示データタイプはPOP-LIST (ABGR)、T-LIST (ABBR)、RADIO GRP (ABBR) のいずれかに設定されます。</p> <p>そうでない場合、表示データタイプはPOP-LIST (MEAN)、T-LIST (MEAN)、RADIO GRP (MEAN) のいずれかに設定されます。</p>
表示書式	項目に関する表示形式マスクに設定されます。
表示幅	項目の表示高さに設定されます。
表示高	項目の表示高さに設定されます。
表示フラグ	項目が表示される場合に設定されます。
挿入フラグ	項目のInsert Allowed属性に基づきます。
更新フラグ	項目のUpdate Allowed属性に基づきます。
オプション(任意入力)フラグ	項目のRequired属性に基づきます。
選択フラグ	項目のQuery Allowed属性に基づきます。
ヒント・テキスト	項目が入力可能または問合せモードで入力可能で、ヒント・テキストを持っている場合、項目のヒント・テキストからコピーされます。
プロンプト	<p>新しい列取扱細目では、項目が表示される場合、列定義に関するデフォルト・プロンプトに設定されます。</p> <p>既存の詳細なカラムの使用状況では、プロンプトは変更されません。</p> <p>唯一の例外はチェックボックス項目です。ここではプロンプトがチェックボックスのラベルに設定されます。</p>
Order By	項目が入ったブロックにデフォルトのORDER BY文がある場合、列取扱細目のORDER BY番号が、ORDER BY文の中の列の位置を反映するように設定されます。
デフォルト	項目のデフォルトが列のデフォルトと同じでない場合、それに設定されます。列のデフォルトと同じ場合、変更されません。

リバース・エンジニアリングとリエンジニアリング

リエンジニアリングとは何
で、なぜ行うのか？

リエンジニアリングは、リバース・エンジニアリング時にリポジトリに記録され
た情報からフォームを生成するプロセスです。

リエンジニアリングによって、スタイル、レイアウト、ドキュメンテーション、
アプリケーション・コードを標準化できます。

リバース・エンジニアリング
後の最初の生成

この章の前の方で示したフォームのリバース・エンジニアリングを行ったのち、
新しいモジュール定義から新しいフォームを生成したい場合があります。

リエンジニアリングされたフォームを次に示します。

The screenshot shows a software application window titled "Microsoft Access" with a menu bar (アクション(A), 編集(E), ブロック(B), フィールド(F), レコード(R), 問合せ(Q), ウィンドウ(W), ヘルプ(H)) and a toolbar. The main window contains a form with the following elements:

- Two text input fields labeled "ディラー番号:" and "ディラー:".
- A text input field labeled "電話番号:".
- A table with 7 columns and 10 rows. The column headers are: 番号, 姓, 日付, 電話番号, モジュール, 仕入価格, 販売価格.
- A status bar at the bottom with the text: "ディラー番号の値を入力します。" and "件数: 0".

リエンジニアリングによって作成したモジュールからフォームを生成した場合、

- 列定義に関する項目プロンプトが回復されます。
- 現在のユーザー作業環境とテンプレートに従ってレイアウトが変更されず（リエンジニアリングされたフォームのマスター・ブロック内の項目の上ではなく横に表示されるプロンプトに注意）。
- Oracle Forms Designerを使ってもとの機能の一部を再インプリメントする必要があるかもしれません（問合せ入力、問合せ実行、終了の各ボタンはリエンジニアリングされたフォームには表示されません）。
- 参照情報は取扱細目としてリバース・エンジニアリングされません。

機能の保持

もとのフォームの機能の一部をリエンジニアリングされたアプリケーションに残したい場合があります。この機能をリエンジニアリングされたアプリケーションに組み込むには、2つの方法があります。

- 必要な機能をリエンジニアリング時に使用されるテンプレートにコピーします。
- Oracle Forms Designerを使って、必要な機能をもとのフォームからリエンジニアリングされたフォームにコピーします。

第 IV 部

レポートの生成

第13章

Oracle Reports ジェネレータ の紹介

この章では、Oracle Reports ジェネレータを紹介し、その基本概念の一部について説明します。

- Oracle Reports ジェネレータとは何か?
- 生成されるレポート モジュールとは何か?
- レポート モジュールの生成方法は?

Oracle Reports ジェネレータとは何か？

Oracle Reports ジェネレータは、製品に相当する品質を備えたOracleレポートバージョン2.5とSQL*Plusレポートを作成するためのDesigner/2000ツールです。

このReports ジェネレータは、リポジトリ、Oracle Reports、またはSQL*Plus作業環境内に記録されている情報とテンプレート レポートを使用してレポート モジュールを作成します (SQL*Plusモジュールの作成では、ジェネレータはテンプレート・モジュールを使用しません)。

注意:生成されるモジュールのソースについては、このガイドの第2章を参照してください。

Oracle Reports ジェネレータは、Oracle ReportsおよびSQL*Plusの作業環境、デフォルト・テンプレート・レポートの初期設定を備えているため、環境設定の編集やテンプレート・レポートの編集を先に実行しなくてもレポートモジュールを作成できます。ただし、カスタムレポートを生成する場合は、初期設定の変更やテンプレート・レポートの編集が必要な場合があります。

Reports ジェネレータへのアクセス

Oracle Reports ジェネレータは、次のDesigner/2000ツールから利用できます。

- モジュール・データ・ダイアグラム
- プリファレンス・ナビゲータ
- モジュール・ストラクチャ・ダイアグラム
- リポジトリ・オブジェクト・ナビゲータ

どのツールを使用してもレポートモジュール定義を生成できますが、次に示すように、現在使用中のツールを使用してレポート モジュール タスクを完了すると便利です。

- レポート モジュール定義の定義か再定義を実行した後、モジュール・データ・ダイアグラムからモジュール定義を生成できます。
- 初期環境設定を変更する場合は、プリファレンス・ナビゲータからレポートモジュール定義を生成し、生成されるレポートに対して新しい設定がどのように影響するかを観察できます。
- モジュール・ストラクチャ・ダイアグラムを使用して1つのモジュール・ネットワーク内に複数のレポート モジュールを入れる場合は、このツールを使用してレポート モジュール間のリンクを確立するレポートモジュール定義を生成できます。
- テーブル定義の変更後にリポジトリ・オブジェクト・ナビゲータからレポートモジュール定義を生成できます。

生成されるレポート モジュールとは何か？

生成されるレポート モジュールは、Oracle ReportsのrdfファイルかSQL*Plusの.sqlスクリプト、またはデータベース内に記憶されているものです。

Oracle Reportsv2.5モジュール 生成されるOracle Reports バージョンv2.5のレポートは、次の要素から構成されます。

- データ・モデル
- レイアウト・モデル
- パラメータ・フォーム

データ・モデル 生成されるレポートは、次のデータ定義オブジェクトで構成されます。

- グループ
- リンク
- 問合せ
- 列
- PL/SQLファンクション
- パラメータ

これらのオブジェクトを総称してレポートのデータ・モデルと呼びます。データ・モデルは、レポートによって使用されるデータと、データとレポート構造間の階層と関係を定義します（データ・モデル・オブジェクトの一般的な情報については、Oracle Reports 2.5のマニュアルを参照してください）。

レイアウト

Reports ジェネレータは、生成されたデータ・モデル、作業環境設定、レポート・レポートを使用して、生成されるレポートのレイアウトを作成します。このレイアウトによって、生成されるレポート内でのオブジェクトの位置が決まります。

レポートのレイアウトを構築するうえでもっとも重要な要素は、グループ階層とレイアウト・スタイルです。

グループ階層

ジェネレータは、データモデルの生成中に作成されたグループ階層を使用してレポートの本体を作成します（レポートの本体についての詳細は、第14章を参照してください）。

ジェネレータは、各グループのためにグループ反復フレームとグループフレームを作成します。

グループ反復フレーム グループは、レポートの情報源ですが、グループ反復フレームは、グループ内のすべての表示列取扱細目のためのプレースホルダーです。

ジェネレータは、一連のグループ反復フレームとしてレポート・レイアウトを作成し、各グループ反復フレームをグループ・フレームで囲みます。グループ反復フレームには、グループ・フレームとグループ反復フレームのネストが含まれることがあります。

ジェネレータは、表示列取扱細目のためのフィールドとフィールドプロンプトも作成します。フィールドは、グループ反復フレーム内

に作成されます。フィールドプロンプトは、グループ反復フレーム内に作成するか、対応するフィールドと整列している別個のフレーム内に作成されることがあります。

グループ フレーム グループ フレームはフレーム本体を含み、フレーム ヘッダーとフレーム フッターも含まれることがあります。グループ フレーム本体は、グループ反復フレームを含みますが、それは各グループの表示フィールド オブジェクトを囲みます。グループ フレーム本体は、グループ タイトルのためのプロンプト フレームを含むことがあります。

レイアウト スタイル

生成されるレポート内の各グループに対して、モジュールのフォーマットを基にしたレイアウトスタイルが割り当てられます (モジュール定義フォーマットの詳細については、14章を参照してください)。Oracle Reports ジェネレータは、次のレイアウト スタイルをサポートしています。

- フォーム (Form)
- タビュラ (Tabular)
- アドレス (Address)
- アクロス (Across)



詳細情報: レイアウト スタイルの詳細については、Oracle Reports ジェネレータ オンライン マニュアルを参照してください。

レイアウト スタイルは、フィールドとボイラープレート オブジェクトのグループ フレーム内での配置方法に影響します。

フィールド フィールドは、データベースから検索された値を含み、レポート内でのその値の外観を定義します。ジェネレータは、表示列として指定されたデータモデル内の各列のグループ反復フレーム内に1つのフィールドを作成します。

ボイラープレート ボイラープレートは、生成されるレポート内に現れるテキストまたはグラフィカル オブジェクトです。

SQL*Plusモジュール

SQL*Plusの .sqlファイルは、SQL*Plusを作成するためのSQLステートメント、フォーマット ステートメント、コメント、次のSQL*Plusコマンドを含みます。

- SET TERMOUT OFF
- SPOOL OFF
- EXIT

フォーマット ステートメントとコメントを作成したくない場合、または .sqlファイルに含まれているSQL*Plusコマンドを変更したい場合は、SQL*Plus作業環境の初期設定を変更します。詳細については、第16章を参照してください。

レポート モジュールの生成方法は？

データの構造を計画した後、レポートモジュールを作成するために生成できるモジュール定義をリポジトリ内に作成できます。

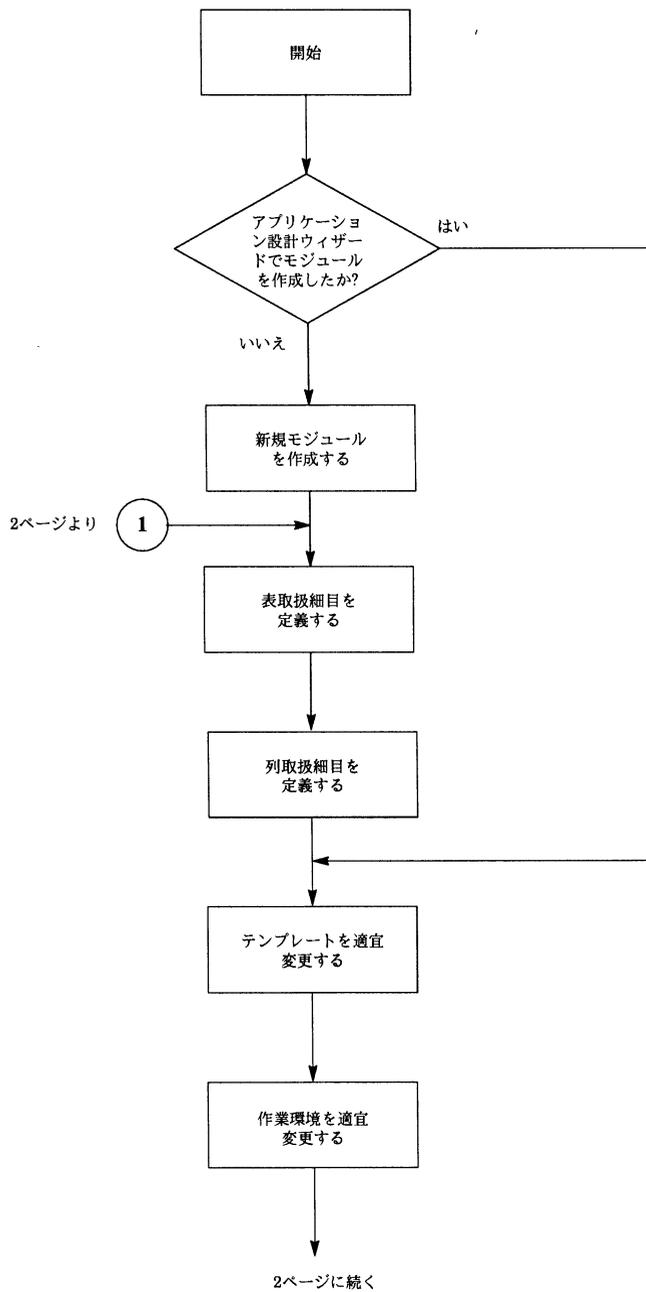
次の手順に従って、生成プロセス用のレポート定義を準備します。

1. モジュール・データ・ダイアグラムを使用して、レポート・モジュール定義のためのモジュール・データ・ダイアグラムを作成します。ダイアグラムを作成するときは、現在のアプリケーション・システムからレポート・モジュール定義を選択するか、新しいモジュール定義を作成する必要があります。
または、アプリケーション設計ウィザードを実行してリポジトリ内にモジュール定義を作成します。
2. モジュール・データ・ダイアグラムを使用して、モジュール定義の表取扱細目と列取扱細目を作成します。
3. モジュール・データ・ダイアグラムを使用して、表取扱細目間の結合を定義するか、デフォルト結合を生成します。
4. サマリーまたはパラメータを作成するか、表取扱細目または列取扱細目をカスタマイズする場合は、モジュール・データ・ダイアグラムを使用してデータ取扱を編集します。
5. 1つのモジュール・ネットワーク内で複数のモジュール定義を結合する場合は、モジュール・ストラクチャ・ダイアグラムを使用します。
6. 必要であれば、プリファレンス・ナビゲータを使用して初期環境設定を変更します。または、初期環境設定を使用してレポート・モジュール定義を生成します。
7. 必要であれば、Oracle Reports Designerを使用してテンプレート・レポートを編集します。または、デフォルト・テンプレートを使用してレポート・モジュール定義を生成します。
8. レポート・モジュール定義を生成し、生成されたレポート・モジュールを実行します。
9. 作業環境、テンプレート・レポート、データ取扱をカスタマイズした後、レポート・モジュール定義を再度生成します。

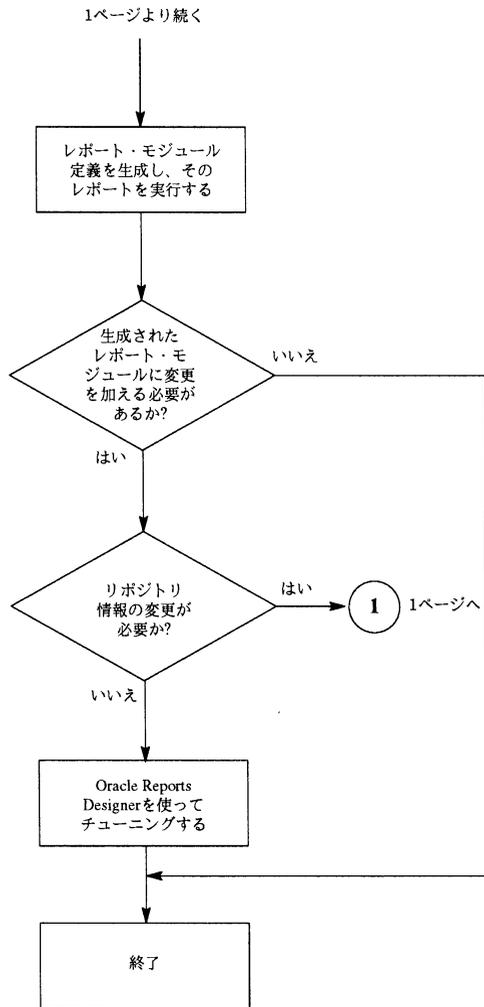
この後の2ページに記載されているダイアグラムは、レポート・モジュール生成プロセスのフローチャートを示しています。

既存のレポートからのレポートモジュール定義の作成

既存のOracle Reports、SQL*ReportWriter、SQL*Plus、SQL*Report (RPT)、またはPro*Cレポートがある場合は、それらのレポートをリポジトリ内にリバース・エンジニアリングできます。レポートのリバース・エンジニアリングについての詳細は、第17章を参照してください。



レポート生成の流れ (1/2ページ)



レポート生成の流れ (2/2ページ)

第14章

レポート・モジュール定義の作成と生成

この章では、レポートモジュール定義の作成方法と、Oracle Reports2.5またはSQL*Plusレポートモジュールを作成するための定義の生成方法について説明します。この章では、次の処理の実行方法を具体的に説明します。

- リポジトリ内にレポートモジュール定義を作成する
- Oracle Reports2.5とSQL*Plusレポートモジュールを生成する
- モジュール定義を編集してモジュールの詳細やフォーマットを変更するかパラメータを作成する
- デフォルトサマリーフィールドを生成する
- レポートモジュール定義に参照情報を追加する
- マスター・ディテール・レポートを生成する
- 項目グループの作成、編集、または削除を実行する
- ツリーウォーク結合を作成する
- マトリックス・レポートを生成する
- ドリルダウン詳細レポートを生成する

レポート モジュール定義の作成

通常モジュール定義は、Designer/2000における戦略、分析、または設計フェーズ中に作成されます。さらに、アプリケーション設計ウィザードを実行することでモジュール定義のデフォルト セットが作成される場合もあります。

アプリケーション・システムにモジュール定義がすでに存在する場合は、この節を飛ばし、存在しているモジュール定義からレポートを生成できます。

レポート・モジュールを生成するために使用したいモジュール定義がアプリケーション・システムに含まれていない場合は、Designer/2000システム・デザイナー・ツールを使用してモジュール定義を作成できます。

注意: この章では、モジュール・データ・ダイアグラムを使用したモジュール定義の作成方法について説明します。

レポート・モジュールを生成するために使用できるモジュール定義の作成には、次の4つの段階があります。

1. ダイアグラムを作成することで、リポジトリ内にモジュール定義を作成します。
2. 最低1つの表取扱をダイアグラムに追加します。
3. 表取扱細目に最低1つの列取扱細目を追加します。
4. 必要であれば、表取扱細目間の結合を作成するか、デフォルト結合を生成します。

ダイアグラムとモジュール定義の作成

モジュール・データ・ダイアグラムでは、ダイアグラムはリポジトリ内の一定のモジュール定義の作成と修正を実行するためのグラフィカル・ユーザー・インターフェースを備えています。

リポジトリ内にすでに存在するレポート・モジュールを生成するためにダイアグラムを作成する必要はありませんが、次のタスクを完了するためには、ダイアグラムを作成する必要があります。

- モジュール・データ・ダイアグラムを使用して、リポジトリ内にモジュールを定義する
- モジュール定義に対して表取扱細目を追加したり、モジュール定義から表取扱細目を削除する
- 表取扱細目に列取扱細目を追加したり、表取扱細目から列取扱細目を削除する
- モジュール定義内に項目グループを作成したり、モジュール定義から項目グループを削除する
- 表取扱細目間の結合を作成する、デフォルト結合を生成する、またはモジュール定義内の結合を削除する

ダイアグラムに対して要素の追加、変更、または削除を実行すると、変化はただちにリポジトリに反映されます。したがって、モジュール定義をリポジトリに保存するためにダイアグラムを保存する必要はありません。

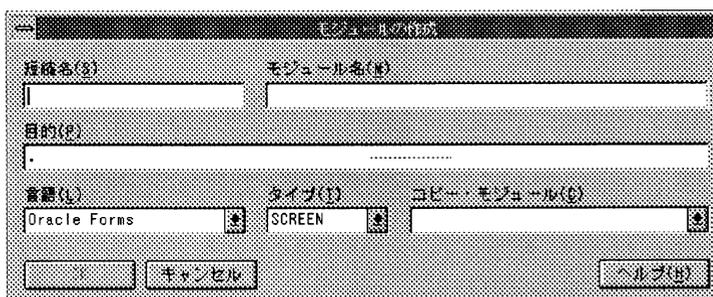
次の手順に従って、ダイアグラムとモジュール定義を作成します。

1. モジュール・データ・ダイアグラマを起動します。
2. ユーザー名、パスワード、接続したいOracleデータベースの接続文字列を入力します（インストールによっては、この手順が必要ない場合があります）。
3. アプリケーション・システムとバージョン番号を選択します。
4. 「ファイル→新規作成」を選択して、「新規ダイアグラム」ダイアログボックスを表示します。



5. 「モジュール作成」ボタンをクリックして、新しいモジュールを作成します。

「モジュール作成」ダイアログボックスが表示されます。



6. モジュールの短縮名、モジュール名、目的を入力し、「言語」としてOracle ReportsかSQL*Plusを、「タイプ」としてREPORTを選択します。

または、「コピー・モジュール」ドロップダウン リストからモジュール定義の名前を選択することでそのモジュールをコピーし、その後必要であればそのモジュールの「言語」と「タイプ」を変更できます。

注意: モジュール定義を編集することで、生成されるレポートで使用される上部、下部、または省略タイトルなどのモジュールに関する追加情報を指定で

きます。詳細については、この章で後ほど説明する「モジュール定義の編集」を参照してください。

7. 「OK」をクリックしてモジュール定義の作成またはコピーを実行し、ダイアグラムを作成します。

表取扱細目の定義

リポジトリ内にレポート・モジュール定義を作成し、そのモジュール定義のためのダイアグラムを作成した後、表取扱細目を追加して、生成されるレポートモジュール内にグループを作成するために使用する表を識別します。

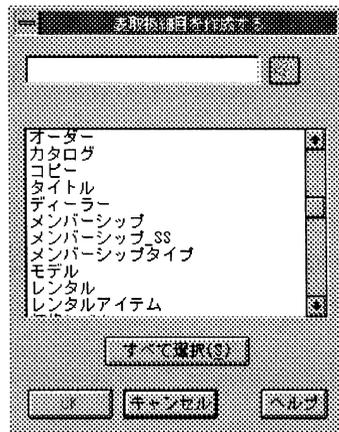
1. 「表取扱細目」描画ツールバー ボタンを選択します。



または、「編集→作成→表取扱細目」を選択します。

2. ボックスの下にあるダイアグラム上にカーソルを置き、マウス ボタンをクリックします。

このアプリケーションシステムのために定義されているすべてのテーブルを示す「表取扱細目」ダイアログボックスが表示されます。次の図は、サンプル・アプリケーション・システムの表を示しています。



3. 「表取扱細目の作成」ダイアログボックスから表を選択し、「OK」をクリックします。たとえばレンタル表を選択します。

選択した表取扱細目がモジュール・データ・ダイアグラム上に表示されます。基礎表取扱細目としてダイアグラム上の最初の表取扱細目が作成されます。

列取扱細目の定義

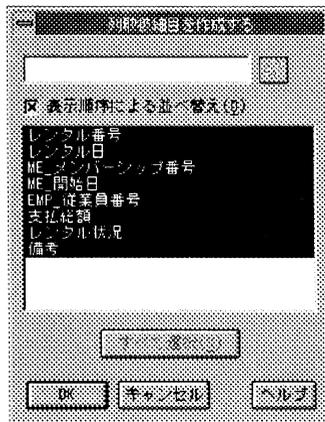
モジュールが使用する表を指定した後、モジュールが使用する列を表から指定します。

1. 現在選択されていない場合は、ダイアグラム上の表取扱細目を選択します。
2. 「列取扱細目」描画ツールバー ボタンを選択します。



または「編集→作成→列取扱細目」を選択します。

「列取扱細目」ダイアログ・ボックスが表示されます。

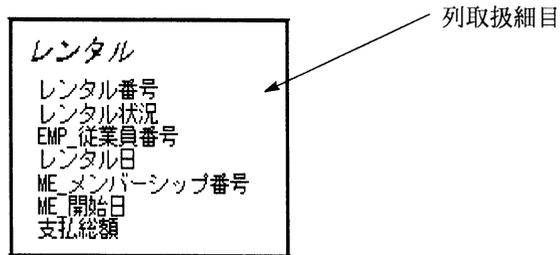


3. ダイアログ・ボックスから列を選択するか、「すべて選択」ボタンを使用してすべての列を選択します。

注意: 表取扱細目に関する列取扱細目をアルファベット順に表示したい場合は、「表示順序による並べ替え」チェックボックスの選択を解除します。このチェックボックスが選択されている場合、列は表示順で表示されます。

4. 「OK」をクリックします。

次のような列取扱細目がモジュール・データ・ダイアグラム内に表示されます。



5. 列取扱細目の順序は、1つを選択し表取扱細目の別の位置までそれをドラッグすることで変更できます。列取扱細目の順序を変更すると、生成されるレポート上でのフィールドの表示順序が変化します。
6. 「ファイル→上書き保存」を選択してダイアグラムを保存します。

モジュール定義に対して実行した変更を保存するためにダイアグラムを保存する必要はありません。ただし、モジュールの編集回数が多い場合は、モジュールを編集するたびにダイアグラムを作成するよりも保存されているダイアグラムをオープンするほうが便利です。

レポート・モジュールの生成

モジュールによって使用される表と列を特定した後、モジュールからレポートを生成できます。次のDesigner/2000ツールを使用してレポート・モジュールを生成できます。

- モジュール・データ・ダイアグラマ
- プリファレンス・ナビゲータ
- モジュール・ストラクチャ・ダイアグラマ
- リポジトリ・オブジェクト・ナビゲータ

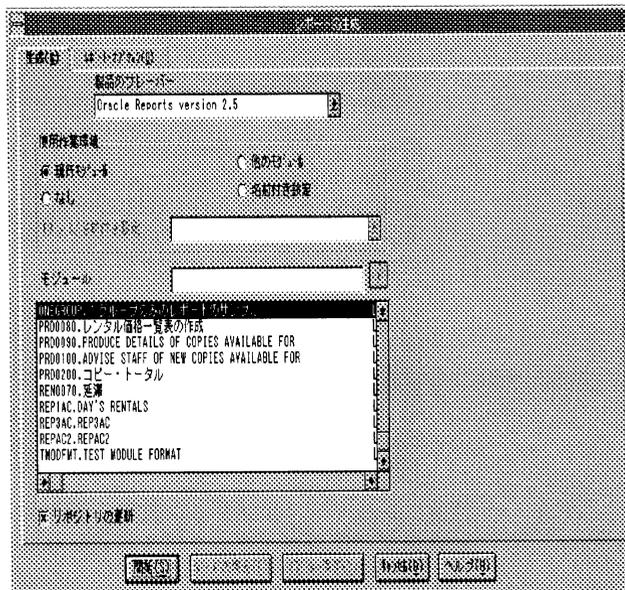
注意: この章では、モジュール・データ・ダイアグラマを使用したレポート・モジュールの生成方法について説明します。

1. 「生成」 ツールバー ボタンを選択します。



または「ユーティリティ→生成」を選択します。

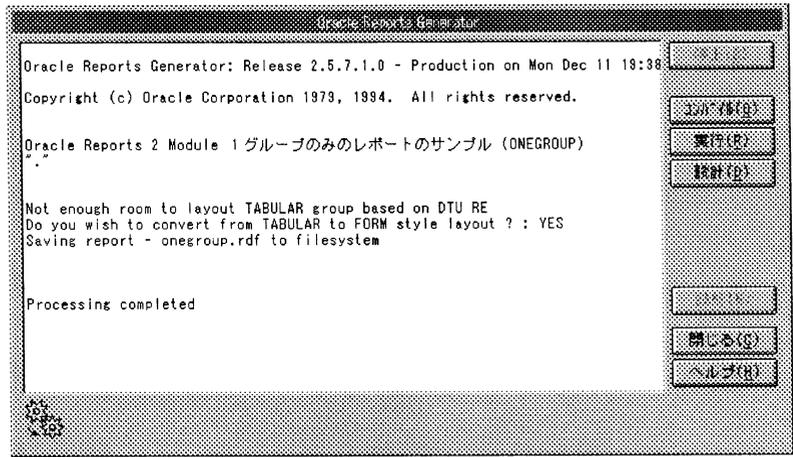
「レポートの生成」ダイアログ・ボックスが表示されます。



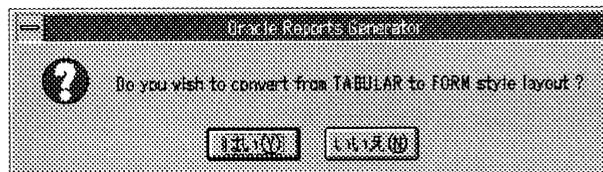
2. 生成するレポート・モジュールの種類に従って、「製品のフレーバー」フィールドからOracle Reports Version2.5かSQL*Plus Reportを選択します。
3. 別のモジュールかアプリケーション・システムの環境設定、または名前付き環境設定セットを使用して生成する場合は、「使用作業環境」グループから適切なオプションを選択します。環境設定セットの詳細については、第16章を参照してください。

- レポート・モジュールを生成すると、生成プロセス中に実行された変更によってリポジトリが更新されます。リポジトリを更新したくない場合は、「リポジトリの更新」チェックボックスの選択を解除します。
- 「開始」ボタンをクリックすると、「モジュール」フィールドで選択されているモジュールからのレポートの生成が開始されます。

「Oracle Reports ジェネレータ」ダイアログ・ボックスが表示されます。ジェネレータがレポート・モジュールを生成している間、生成プロセスにおけるメッセージがこのダイアログ・ボックスに表示されます。



生成プロセス中に追加情報を与えなければならない場合があります。たとえばOracle Reportsのレポート・モジュールを生成する場合、ジェネレータは各マスター・グループに対してタブュラ・レイアウト・スタイルを使用します。レポートがグループ内のフィールドを表示できるほど大きくない場合、ジェネレータは次のダイアログ・ボックスを表示します。



注意: レポート・オプション設定をカスタマイズすることで、ジェネレータがこれらのダイアログ・ボックスを表示しないように設定できます。詳細については、次の節を参照してください。

- 生成プロセスが完了したら、次の処理を実行できます。
 - 「実行」ボタンをクリックして、生成されたレポートを実行する
 - 「コンパイル」ボタンをクリックして、生成されたレポートをコンパイルする
 - 「設計」ボタンをクリックして、Oracle Reports Designerを起動する

- ダイアログ・ボックスに表示されているメッセージを選択し、「詳細」ボタンを選択することで、メッセージに関する情報を取得する

生成オプションのカスタマイズ 生成されるレポート・モジュール、ログ・ファイル、テンプレート・レポートのロケーションなどの生成オプションを制御するデフォルト設定を変更できます。

注意:生成されるレポート・モジュールは、Reportsジェネレータのテンプレートと環境設定でもカスタマイズできます。詳細については、第16章を参照してください。

次の手順に従ってレポート オプションを変更します。

1. 「レポートの生成」ダイアログ・ボックスから「レポート・オプション」タブを選択します。

次のオプションが表示されます。

2. 生成されたレポートをデータベースに保存する場合は、ドロップダウンリストから「データベース」を選択します。
3. レポートをファイルに保存すると、デフォルトで現在の作業ディレクトリにレポートが作成されます。ディレクトリを変更する場合は、「ディレクトリ」フィールドにパスを入力します。「ブラウズ」ボタンを使用してディレクトリを検索できます。

注意: Reports ジェネレータは、このオプションを使用してパスを検索します。したがって、レポート名は入力しないでください。ジェネレータは、入力された内容をディレクトリ名と解釈します。

4. 生成プロセス中に発生したエラーと警告を記録するログ・ファイルを作成する場合は、「ログ・ファイル名」フィールドにログ・ファイル名を指定します。

「ブラウザ」ボタンを使用してファイルを検索できます。ログ・ファイルが指定されない場合、ジェネレータはログ・ファイルを作成しません。

生成プロセス中にエラーか警告が発生した場合、ジェネレータは指定されたディレクトリにログ・ファイルを作成します。ディレクトリが指定されない場合、ジェネレータは現在の作業ディレクトリにログ・ファイルを作成します。エラーや警告がまったく発生しなかった場合、ジェネレータはログ・ファイルを作成しません。

5. デフォルトでは、ジェネレータはLAYREP、PARREP、DETREPの各環境設定を使用してテンプレートの場所と名前を判断します（作業環境とテンプレートの詳細については、第16章を参照してください）。

別のディレクトリに保存されているテンプレートを使用する場合は、「テンプレート」グループの「ディレクトリ」フィールドにそのディレクトリを指定します。

注意: Reports ジェネレータは、このオプションを使用してパスを検索します。したがって、テンプレート名は入力しないでください。ジェネレータは、入力された内容をディレクトリ名と解釈します。ジェネレータが提供しているテンプレートではなく、独自のテンプレートを使用する場合は、プリファレンス・ナビゲータを使用してLAYREP、PARREP、DETREPの各環境設定に指定されているテンプレート名を変更してください。テンプレート・フィールドに指定したディレクトリと一致するように環境設定のパス名を変更する必要はありません。ジェネレータは、環境設定内のパス設定を無視し、指定されたパスを使用します。

6. モジュール・データ・ダイアグラムにログインするときに使用したユーザー名以外のユーザー名でレポートを実行する場合は、次の書式で「実行時ユーザー」フィールドにユーザー名、パスワード、接続文字列を入力します。

ユーザー名/パスワード@接続文字列

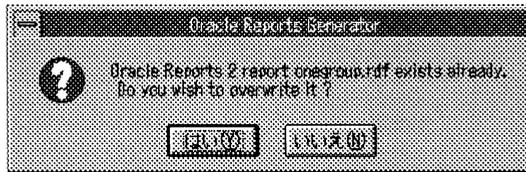
注意: ORACLE.INIファイルにLOCALパラメータが指定されている場合は、接続文字列を指定する必要はありません。

7. モジュール名の長さを制限する場合は、「最大モジュール名」フィールドに文字数を指定します。

ジェネレータは、モジュール名を使用してレポート名を作成するため、モジュール名の長さを制限することでアプリケーションシステム間および異なる開発プラットフォーム間での整合性が保たれます。

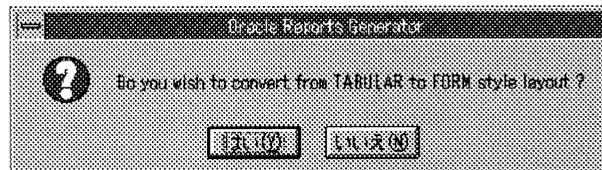
注意: モジュール名の長さを制限しない場合は、このフィールドに0（ゼロ）を入力します。

8. 数値データ型であり、主キー・外部キー・一意キーの一部ではないレポート上のフィールドのサマリーをグループとレポート・レベルで生成する場合は、「デフォルト・サマリーの生成」チェックボックスを選択します（サマリーについては、第15章を参照してください）。
9. 生成プロセス中にジェネレータが次のダイアログ・ボックスを表示しないようにする場合は、「既存定義の上書き」オプションから「はい」または「いいえ」を選択します。



常にレポートを上書きする場合は、このグループで「はい」オプションを選択します。モジュール定義と同名のレポートモジュールが存在するときには生成プロセスを停止したい場合は、「いいえ」を選択します。

10. 生成プロセス中にジェネレータが次のダイアログ・ボックスを表示しないようにする場合は、「TABULARからFORMレイアウトへの変換」オプションから「はい」または「いいえ」を選択します。



レイアウト・スタイルを変更する代わりにタブュラー・レイアウト・スタイルのグループのフィールドを常に折り返す場合は、グループから「いいえ」オプションを選択します。確認なしでグループを変換する場合は、「はい」を選択します。

注意: グループのフィールドを折り返すことを選択した場合は、MINMSP環境の初期設定を変更することで折り返す量を制御できます。この環境設定については、オンライン・マニュアルを参照してください。

11. 設定したオプションを保存する場合は、「オプション保存」ボタンを選択します。これらの設定は、次回のレポート生成時にデフォルトとして「生成オプション」タブに表示されます。
12. 「開始」ボタンをクリックして、レポート モジュール定義を生成します。

生成されるレポートの構成要素 (Oracle Reports v2.5)

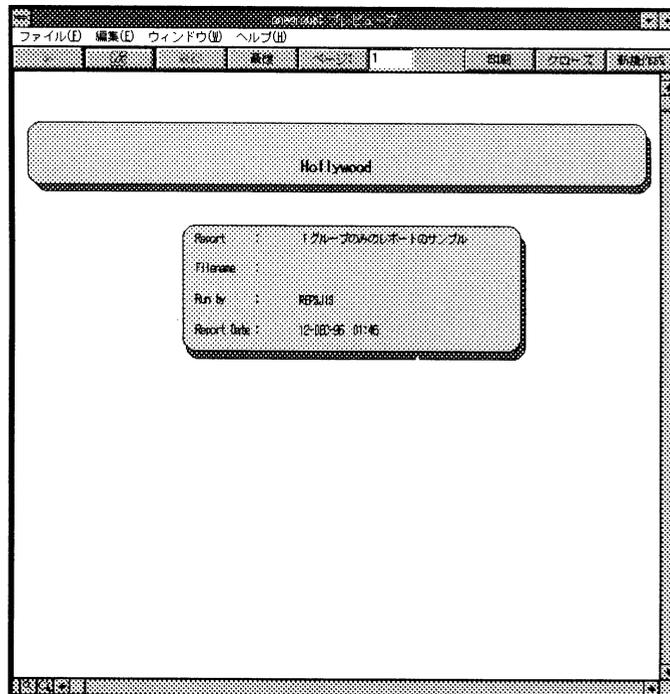
レポートの「製品のフレーバー」として「OracleReportv2.5」を選択した場合、レポート モジュールに対して.rdfファイルが作成されます。このレポート モジュールを実行すると、次のレポート セクションが表示されます。

- ヘッダー・ページ
- レポート本体
- トレーラ・ページ

この節では、生成されるレポート・モジュールの各セクションについて説明します。

ヘッダー ページ

次の図は、サンプル・ヘッダー・ページを例示しています。

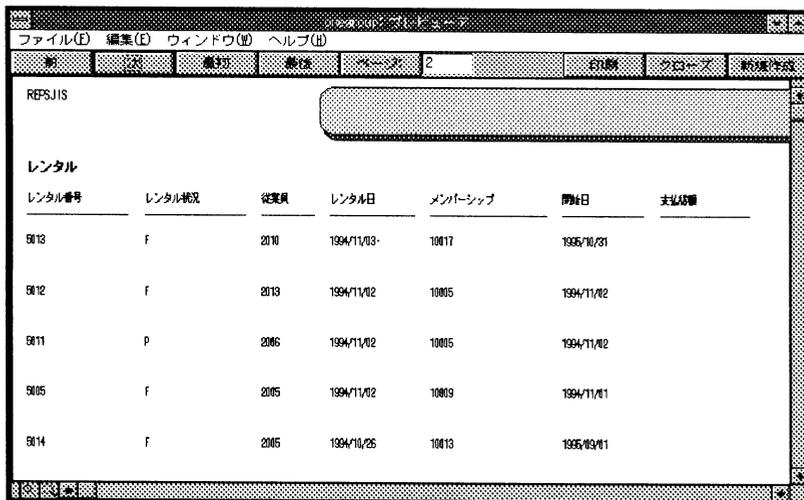


ジェネレータは、レイアウト・モデル・テンプレート・レポートを使用してヘッダー・ページを作成します。このテンプレートが、レポートのヘッダー・ページ

の内容と、ヘッダー・ページに表示されるデータの位置、フォント、フォント・サイズを決定します。テンプレート・レポートの変更については、第16章を参照してください。

レポート本体

「次」ボタンを選択すると、レポートの本体が表示されます。次の図は、レポート本体のサンプルを示しています。



The screenshot shows a window titled 'REPSJIS' with a menu bar (File, Edit, Window, Help) and a toolbar. Below the toolbar is a table with the following data:

レンタル番号	レンタル状況	従業員	レンタル日	メンバーシップ	閉鎖日	支払金額
5013	F	2010	1994/11/03	10017	1995/10/31	
5012	F	2013	1994/11/02	10005	1994/11/02	
5011	P	2006	1994/11/02	10005	1994/11/02	
5005	F	2005	1994/11/02	10009	1994/11/01	
5014	F	2005	1994/10/25	10013	1995/03/01	

グループ レポート内の情報は、グループ単位で整理されます。このレポートには1つのグループが含まれており、これは、1つの基礎表取扱細目から導出されています（グループ内の情報は、複数の表取扱細目から導出できることに注意してください。詳細については、後ほど説明する「参照情報のインクルード」を参照してください）。

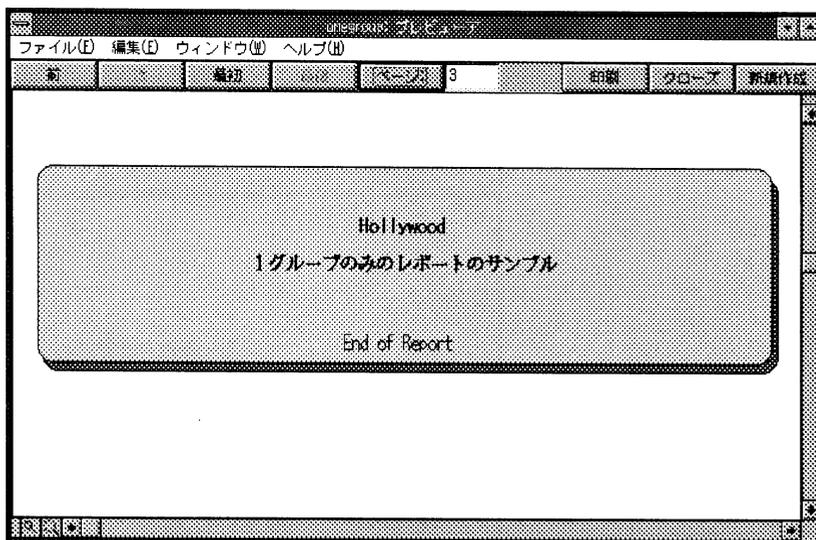
グループ・タイトル ジェネレータは、「表示タイトル」を使用してグループ・タイトルを作成します。「表示タイトル」は、表取扱細目を編集することで変更できます。グループ・タイトルの変更については、第15章を参照してください。

レイアウト・モデル・テンプレート・レポートは、グループ・タイトルのフォントを決定します。環境設定は、タイトルの位置とフォーマットを決定します。テンプレートと環境設定によるグループ・タイトルのカスタマイズについては、第16章を参照してください。

- グループ・レイアウト・スタイル** ジェネレータは、モジュール定義のフォーマットを使用して各グループにレイアウト・スタイルを割り当てます。サンプル・モジュール定義は、マスター・ディテール・フォーマットになっており、これはレポート・モジュールに割り当てられるデフォルト・フォーマットです。このグループは、マスター・ディテール・フォーマットのモジュールから生成され、子グループがないため、タビュラー・レイアウト・スタイルが適用されます。
- モジュールのフォーマットとデフォルト・レイアウトスタイルの詳細は、この章で後ほど説明する「モジュール定義の編集」を参照してください。
- フィールド** フィールドには、データベースの列から検索された値が含まれています。生成されるレポート上に表示すべき取扱として列取扱細目が指定されている場合、ジェネレータは、列取扱細目ごとに1つのフィールドを作成します。
- フィールドは、表取扱細目内での列取扱細目の表示順序と同じ順序で表示されます。ダイアグラム上で列取扱細目を移動することでフィールドの順序を変更できます。
- フィールドは、列取扱細目の編集、テンプレートレポートの編集、または環境設定の変更を行うことでカスタマイズできます。詳細は、第15章と第16章を参照してください。
- フィールド・プロンプト** ジェネレータは、列名を使用してフィールド・プロンプトを作成します。フィールド・プロンプトは、列取扱細目を編集することで変更できます。詳細は、このガイドの第15章を参照してください。
- 余白** レイアウト・モデル・テンプレート・レポートによって、余白の内容がぎまります。テンプレート・レポートのカスタマイズについては、第16章を参照してください。

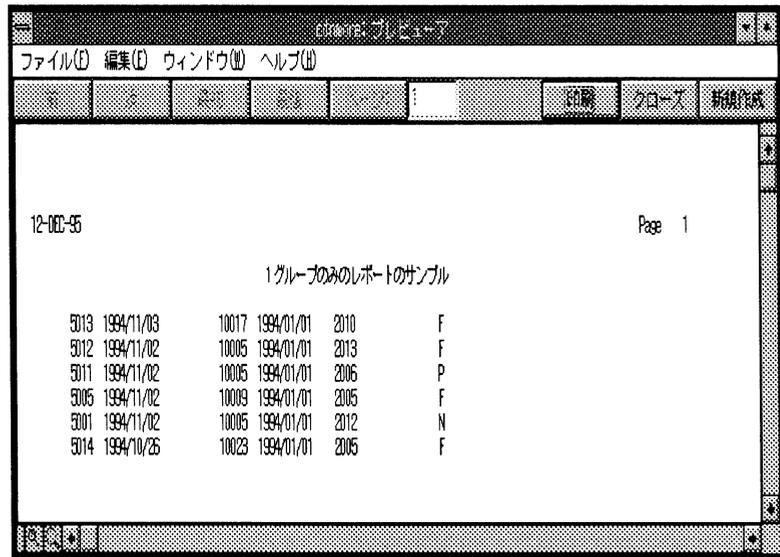
トレーラ・ページ

Oracle Reports Previewerで「最後」ボタンをクリックすると、トレーラ・ページが表示されます。トレーラ・ページのサンプルは次のとおりです。



生成されるレポートの構成要素 (SQL*Plus)

SQL*Plusレポートを生成する場合は、「レポートの生成」ダイアログ・ボックスの「製品のフレーバー」ドロップダウン・リストから「SQL*Plus Report」を選択します。次の図は、レポート・モジュール定義から生成されたSQL*Plusレポートのサンプルを示しています。

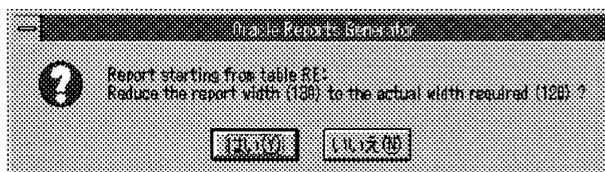


レポートの幅と高さ

レポート・モジュール定義からSQL*Plusレポートを生成すると、ジェネレータはレポートに情報を表示できるだけの幅があるかどうかを判断します。

Oracle Reportsのレポート生成プロセスとは異なり、ジェネレータはSQL*Plusレポートのデフォルト幅と高さを決定するためにテンプレート・レポートを使用することはありません。代わりに、WIDTHとHEIGHT環境設定が使用されます。WIDTH環境設定の初期値は80文字であり、HEIGHT環境設定の初期値は60文字です（環境設定の詳細については、第16章を参照してください）。

- 生成するレポートの幅がSQL*Plusレポートの指定幅よりも狭ければ、次のようなダイアログ・ボックスが表示されます。



レポートの幅を表示されている幅に減らすことも指定した幅を使用することもできます。

- 生成されるレポート内の情報がデフォルト幅よりも広いレポート幅を必要とする場合、レポート・モジュール定義は生成されません。

レポート・モジュールを生成するためには、次のいずれかの方法で、モジュール定義を変更する必要があります。

- 生成されるフィールドの表示幅を減らす。詳細は、第16章を参照してください。
- モジュール定義のフォーマットを変更する。詳細は、この章で後ほど説明する「モジュール定義の編集」を参照してください。
- WIDTH環境設定の初期値を増やす

モジュールのフォーマット デフォルトでは、SQL*Plusレポートにはコントロール・ブレイク・フォーマットが割り当てられます。モジュール定義の編集によって、このモジュール・フォーマットをマスター・ディテールに変更できます。詳細は、この章で後ほど説明する「モジュール定義の編集」を参照してください。

SQL*Plusコマンド ファイル デフォルトでは、次のコマンドを含むSQL*Plusコマンド ファイルが作成されず。

- SPOOL OFF
- TERMOUT OFF
- EXIT

環境設定を使用して、生成されるコマンドファイル内のコマンドのカスタマイズやジェネレータによるコマンドファイルの作成を抑制できます。



詳細情報: SQL*Plus環境設定の全リストについては、Oracle Reports ジェネレータのオンライン・マニュアルを参照してください。

モジュール定義の編集

レポート・モジュール定義を編集して、モジュールの短縮名や名前、言語、タイプを変更したり、モジュールのフォーマットやパラメータ、ランタイムパスのソース、コマンド行の構文、上部タイトル、下部タイトル、短縮タイトルの追加または変更を行えます。

次の手順に従って、モジュールを編集します。

1. 「編集→要素→モジュール」を選択します。
2. 適切なタブを選択します。
 - 短縮名や名前、目的、言語、タイプを変更する場合は、「定義」タブから「モジュール」タブを選択します。
 - フォーマットやランタイムパスのソース、コマンド行の構文、上部タイトル、下部タイトル、短縮タイトルを変更する場合は、「定義」タブから「クライアント細目」タブを選択します。モジュール定義のフォーマットの変更については、次の節を参照してください。
 - パラメータを変更する場合は、「パラメータ」タブを選択します。詳細は、この節で後ほど説明する「パラメータ定義の作成」を参照してください。

モジュール定義のフォーマットの変更

レポート・モジュールを生成するとき、ジェネレータは常にモジュールのフォーマットを使用してモジュール内のグループのレイアウト・スタイルを決定します。

次の表は、レポート・モジュールのフォーマットに割り当てられるレイアウト・スタイルを示します。

レポート モジュールのフォーマット	生成されるレポート内のデフォルトレイアウトスタイル
コントロール ブレーク	全グループについてタビュラレイアウトスタイル
マスター・ディテール	子グループがあるグループではフォーム・レイアウト・スタイル 子グループがないグループではタビュラ・レイアウト・スタイル
マトリックス	アクロス・マスター階層内のすべてのグループではアクロス・レイアウト・スタイル ダウンマスター/セル階層内のすべてのグループではタビュラ・レイアウト・スタイル
ラベル	すべてのグループについてアドレス・レイアウト・スタイル

注意: 表取扱細目を編集することで一定のグループのデフォルト・レイアウト・スタイルを上書きできます。詳細は、第15章を参照してください。

サンプル

次の図は、モジュール定義からデフォルト・モジュール・フォーマットで生成されたレポート・モジュールを示しています。

Last Name	Street	Town
PACHARE	1304 L ST	SACRAMENTO
HAN	6622 BAYSHORE HIGHWAY	BURLINGAME
LARGA	621 VINEYARD WAY	SAN MATEO
BRANDT	45 OLD COUNTY RD	BELMONT
BACHALLET	311 GREENWICH ST	SAN FRANCISCO
VALENCIA	899 STROUD ST	SAN FRANCISCO
COLLINE	306 MARSH ST	SAN FRANCISCO
BASEL	2756 BAYSHORE HIGHWAY	BURLINGAME
STEINHAUSE	622 CYRIL MAGNIN	SAN FRANCISCO
GELDOF	23 MARKET ST - APT 6	SAN FRANCISCO
TAPAZI	116-POST ST	SAN FRANCISCO
RINTZELMAN	143 DAVIS DR	BELMONT

モジュール定義フォーマットがデフォルトからラベル・モジュール・フォーマットに変更された場合は、モジュール定義を生成すると次のレポートが作成されます。

Last Name:	PACHARE
Street:	1304 L ST
Town:	SACRAMENTO
Postal Code:	95914
Last Name:	HAN
Street:	6622 BAYSHORE HIGHWAY
Town:	BURLINGAME
Postal Code:	94010
Last Name:	LARGA
Street:	621 VINEYARD WAY
Town:	SAN MATEO
Postal Code:	94404
Last Name:	BRANDT
Street:	45 OLD COUNTY RD
Town:	BELMONT
Postal Code:	94002

注意: 列取扱細目を編集することで、フィールド・プロンプトを削除できます。詳細は、第15章を参照してください。

パラメータ定義の作成

モジュール定義を編集することで、モジュールのパラメータを作成できます。パラメータの作成は、列取扱細目の編集やテンプレート・レポートの修正でも実行

できますが、モジュールレベルでパラメータを作成すると、パラメータ名と演算子以外の情報も指定できます。

パラメータを作成する場合は、パラメータ名と、ドメインとデータ型のどちらかを指定する必要があります。パラメータの追加情報も指定できますが、その情報は、パラメータの作成に必ず必要なわけではありません。

1. 「パラメータ」タブにパラメータ名を入力します。
2. あるドメインに属するパラメータを設定したい場合は、ドロップダウンリストからドメインを選択します。

ドメインを指定すると、ドメインに対して指定されているプロパティの情報によってタブの「データ型」と「長さ」フィールドが更新されます。

ヒント: ドロップダウン・リストにドメイン名が表示されない場合は、リポジトリ・オブジェクト・ナビゲータ内でドメインを作成します。



詳細情報: ドメインの作成については、リポジトリ・オブジェクト・ナビゲータのオンライン・マニュアルを参照してください。

3. ドメインを指定しない場合は、パラメータのデータ型を指定する必要があります。
4. 「パラメータ」タブのその他のフィールドに値を指定できますが、この情報は、パラメータの作成で必ず必要なわけではありません。



詳細情報: モジュール定義とパラメータの編集については、モジュール・データ・ダイアグラムのオンライン・マニュアルを参照してください。

5. 「OK」を選択すると、パラメータが作成されます。

これで、パラメータを列取扱細目に関連付けるか、テンプレートレポートを使用してさらにカスタマイズできます。

パラメータと列の関連付けと、環境設定とテンプレートを使用したパラメータのカスタマイズについては、第15章と第16章を参照してください。

モジュール定義の削除

レポート・モジュール定義を削除する場合は、リポジトリ・オブジェクト・ナビゲータを使用します。

1. モジュール・データ・ダイアグラマから「ツール→リポジトリ・オブジェクト・ナビゲータ」を選択します。
2. 設計/インプリメンテーション・オブジェクトを表示します。

注意: すでにオープンされているアプリケーション・システムで、「モジュール」や「表」等がオブジェクト・ツリー上に存在しない場合は、設計/インプリメンテーション・オブジェクトが表示されていません。その場合は、「ファイル→開く」を選択し、「フィルタ」ダイアログ・ボックスで「設計/インプリメンテーション・オブジェクトの表示」チェックボックスをチェックして、再度そのアプリケーション・システムをオープンしてください。

3. モジュール・ノードを展開し、削除するモジュール定義名を選択します。
4. 「編集→削除」を選択します。



詳細情報: オブジェクトの削除については、リポジトリ・オブジェクト・ナビゲータのオンライン・マニュアルを参照してください。

デフォルト サマリーの生成

デフォルト サマリー生成機能を使用して、生成されるレポート内にレポート レベルかグループ レベルのサマリーを自動的にインクルードできます。

注意: ページ レベルかレコード レベルでサマリーを作成する場合、または SUM以外の演算子を使用してサマリーを作成する場合は、サマリーを明示的に作成する必要があります。手順については、第15章を参照してください。

次の手順に従うと、デフォルト サマリーが自動的に生成されます。

1. レポートモジュール定義のためのダイアグラムを作成するか、オープンします。

モジュール定義は、数値列を基にした列取扱細目を含んでいなければなりません。さらに、列は、主キー・外部キー・一意キーの各制約のキー列であってはなりません。

2. 「ユーティリティ→生成」を選択します。
3. 「レポート・オプション」タブを選択します。
4. 「デフォルト・サマリーの生成」チェックボックスを選択します。
5. 「OK」をクリックして、レポート モジュール定義を生成します。

ジェネレータは、主キー、外部キー、または一意キーの一部ではない各数値フィールドに対してグループ レベルとレポート レベルのサマリーを作成します。ジェネレータは、サマリーフィールドの列取扱細目をダイアグラム内の表取扱細目に追加することにも注意してください。

参照表取扱のインクルード

この章ですでに説明した「レポートモジュールの生成」で、1つの基礎表取扱細目を含むサンプル・レポート・モジュールを提示しました。このモジュールは、基礎表取扱細目で使用されている表を基にした1つのグループを含むレポートを生成します。

グループには、他の表から導出されたフィールドもインクルードできます。

たとえば前節で提示したレポートは、ビデオの貸し出しに関する詳細を表示しており、各貸し出しを承認した従業員の従業員番号が含まれています。生成されるレポートに従業員の名前を追加する場合、レポート・モジュールは、その情報を含む表に基いた参照表取扱細目を持っていなければなりません。

生成されるレポートに参照情報をインクルードする方法は2つあります。

- デフォルト結合生成ユーティリティを使用して、参照取扱細目を自動的に作成する
- 参照情報用の表取扱細目と列取扱細目を作成する

参照自動生成

参照表取扱細目が基礎表取扱細目を作成する場合は、基礎表取扱細目に外部キー列のための列取扱細目が含まれていなければなりません。



前述のサンプル・モジュール定義では、基礎表取扱細目には、次の外部キー列のための列取扱が含まれています。

- EMP_従業員番号
- ME_メンバーシップ番号
- ME_開始日

従業員表とメンバーシップ表の列詳細を表示すると、外部キー列以外の列に対して記述子シーケンスが定義されていることが示されます。

列名	記述子シーケンス	AutoGen Type
従業員番号		
姓	10	
職務		
上司従業員番号		
名前	20	
イニシャル		

列名	記述子シーケンス	AutoGen Type
メンバーシップ番号		
開始日付		
EMP_従業員番号		

注意: 列に定義されている記述子シーケンスを表示するには、「編集→要素→表/ビュー/スナップショット」を選択し、「列表示」タブを選択します。

このサンプルは、デフォルトの参照取扱を生成するための次の前提条件を例示しています。

- 基礎表取扱は、外部キー構成要素ごとに列取扱細目を持っていないといけない
- 注意: 外部キーが複数の列で構成されている場合、基礎表取扱は、各外部キー列の列取扱細目を持っていないといけない。
- 参照表取扱細目を作成するために使用される表は、記述子列として定義される外部キーの一部ではない列を最低1つを持っていないといけない

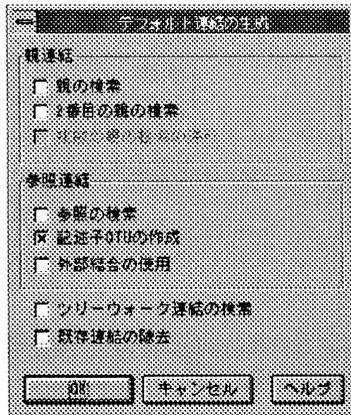
次の手順に従うと、参照表取扱細目が自動的に作成されます。

1. 「デフォルト結合の生成」 ツールバー・ボタンを選択します。



または、「ユーティリティ→デフォルト結合の生成」を選択します。

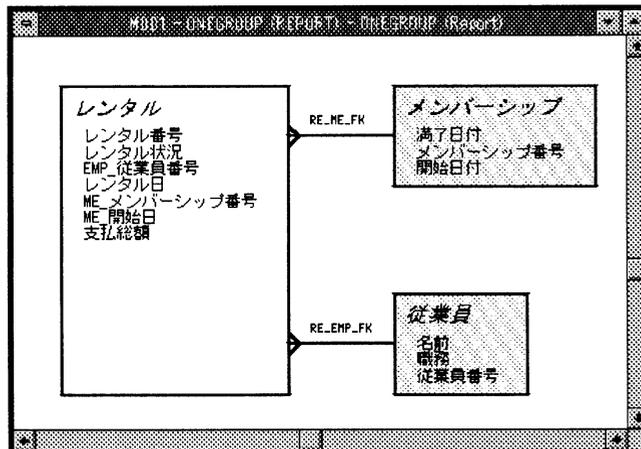
次に示す「デフォルト結合の生成」ダイアログ・ボックスが表示されます。



2. 次のチェックボックスを選択します。

- 「親の検索」
- 「参照の検索」
- 「記述子DTUの作成」

3. 「OK」をクリックします。ユーティリティは、次に示すような参照表取扱細目を作成します。



このサンプルでは、デフォルト結合生成ユーティリティは2つの参照表取扱細目を作成しています。作成された列取扱細目とその作成理由を次の表に示します。

表取扱細目	列取扱細目	作成理由
メンバーシップ	満了日付	記述子列
	メンバーシップ番号	複合外部キーの一部
	開始日付	複合外部キーの一部
従業員	姓	記述子列
	名前	記述子列
	従業員番号	外部キー

4. レポート・モジュール定義を生成し、生成されたレポート・モジュールを実行します。

生成されるレポートの構成要素

前節のサンプル・レポート・モジュール定義を実行すると、参照情報を含むレポートが生成されます。

The screenshot shows a report window titled 'REPSJIS' with a menu bar (File, Edit, Window, Help) and a toolbar. The main content is a table with the following data:

レンタル番号	レンタル状況	従業員	名前	職種	レンタル日	メンバーシップ	開始日	満了日	課
5810	F	2810	今日子	ASISTANT	1994/11/03	10017	1995/10/01		31
5812	F	2810	英子	庶務	1994/11/02	10005	1994/11/02		01
5811	P	2806	ひらみ	ASISTANT	1994/11/02	10005	1994/11/02		01
5805	F	2805	五郎	ASISTANT	1994/11/02	10009	1994/11/01		31
5814	F	2805	五郎	ASISTANT	1994/10/26	10013	1995/09/01		01

グループ

レポートには、「メンバーシップ」表取扱細目と「従業員」表取扱細目からの情報が含まれます。

グループタイトル

グループタイトルは、基礎表取扱と参照表取扱の表示タイトルの組み合わせであり、スラッシュ記号で区切られます。

表示タイトルは、取扱細目を編集することで変更できます。詳細は、第15章を参照してください。タイトルを区切るために使用する文字は、FTCHRS環境設定の初期値を変更することで変更できます。環境設定のデフォルト値の変更については、第16章を参照してください。

フィールド

レポートには、各レンタルの詳細が含まれており、会員権の発効日と満期日、レンタルを承認した従業員の姓名が含まれています。

参照列取扱を基にしたフィールドは、外部キー列を基にしたフィールドの後に表示されます。

- 姓、名前、記述子列を基にしたフィールドは、外部キー列EMP_従業員番号を基にしたフィールドの後に表示される
- 満了日付記述子列を基にしたフィールドは、複合外部キーを構成する列であるME_メンバーシップ番号とME_開始日を基にした項目の後に表示される

デフォルト結合生成ユーティリティが外部キーの反対側にある参照表内の列の列取扱細目を作成しているにもかかわらず、それらの列は生成されたレポートにフィールドとして表示されていません。ジェネレータは、これらの取扱を非表示取扱として作成します。生成されるレポート上での列取扱細目の非表示と表示については、第15章を参照してください。

手動による 検索の作成

参照表取扱細目は、手動でも作成できます。この節では、前節の「参照の自動生成」でサンプルとして示された表取扱細目を使用して説明します。

1. レポート・モジュールを含むダイアグラムをオープンするか、モジュール定義用の新しいダイアグラムを作成します。

注意: モジュール定義には、最低1つの基礎表取扱細目が含まれていなければなりません。

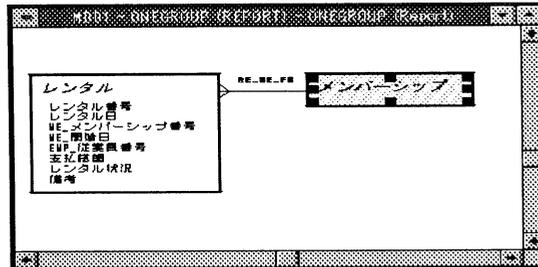
2. 基礎表取扱細目の隣に表取扱細目を作成します。「表取扱細目の作成」ダイアログ・ボックスが表示されます。

注意: 四角形の左上角の位置によって、新しい取扱の配置場所が決定されます。四角形の左上角は、新しい取扱が隣に配置される取扱のダイアグラムに比べ高すぎても低すぎてもいけません。

3. 参照情報として使用する列を含む表を選択し、「OK」をクリックします。

注意: 表に外部キーと最低1つの記述子列が含まれていることを確認してください。

参照表取扱細目と参照結合が作成されます。

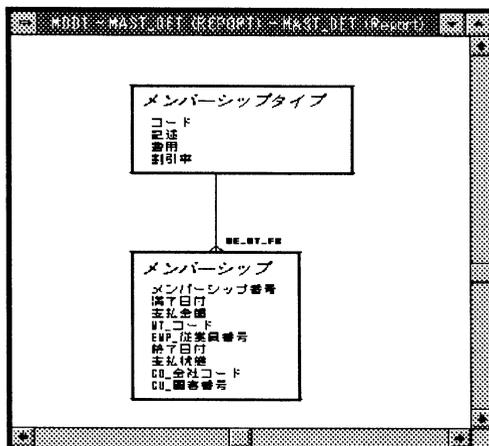


4. 生成されるレポートに表示されるフィールドの列取扱細目を追加します。
注意: 外部キー結合の反対側にその外部キー列の列取扱細目を作成する必要はありません。
5. レポート・モジュール定義を生成し、生成されたレポート・モジュールを実行します。

マスター・ディテール・レポート・モジュールの生成

マスター・ディテール・レポートには、複数のグループが含まれています。マスターグループのすべての値に対して関連する値、ディテール・グループ、その他のグループのいずれかが検索されます。

この節では、生成時にマスター・ディテール・レポート・モジュールを作成するレポート・モジュール定義の作成方法について説明します。次の図は、マスター・ディテール・レポート・モジュール定義のサンプルを示しています。



次の手順に従って、マスター・ディテール・レポートを作成します。

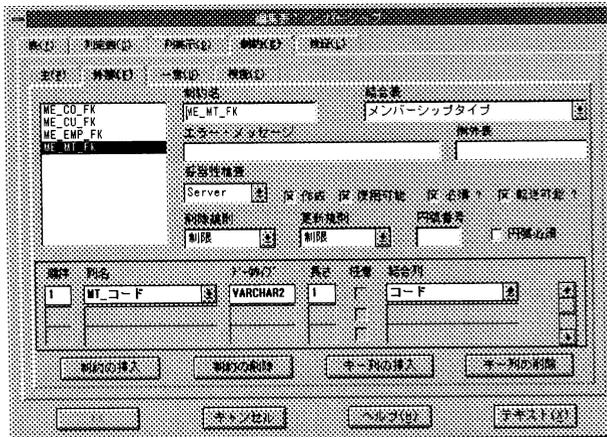
1. レポートモジュールを含むダイアグラムをオープンするか、モジュール定義用の新しいダイアグラムを作成します。
2. ダイアグラム上に基礎表取扱細目を作成します。ジェネレータはこの基礎表取扱細目を使用して、生成されるレポート・モジュールのマスター・グループのための情報を検索します。

この例では、マスターグループの基礎表取扱細目は、メンバーシップタイプ表を基にしています。

3. 最初の基礎表取扱細目の下に、基礎表取扱細目をもう1つ作成します。ジェネレータはこの基礎表取扱細目を使用して、生成されるレポートモジュールのディテール・グループのための情報を検索します。

注意: ディテール・グループのための基礎表取扱細目は、マスター・グループのための基礎表取扱細目の下に配置しなければなりません。

選択する表は、マスターとなる基礎表取扱細目の表に対して外部キー制約を持っていなければなりません。この例では、メンバーシップ表が使用されます。メンバーシップ表に対して定義されているキー制約を次の図に示します。



注意: キー制約を表示するには、「編集→要素→表/ビュー/スナップショット」を選択して表を指定し、その後「キー制約」タブを選択します。

4. 各表取扱細目の列取扱細目を作成します。ディテール・グループの表取扱細目には、外部キーの列取扱細目が含まれていなければなりません。
この例では、MT_コード列が外部キー列になります。
5. 基礎表取扱間マスター・ディテール結合を作成します。これは、次の2つの方法で実行できます。

- 描画ツールバー上の「結合の作成」描画ツールバー・ボタンを選択する



基礎表取扱細目のどれかをクリックし、別の表取扱細目までマウスを移動し、その取扱の上でクリックする。2つの表取扱細目のクリック順序は問いません。

- デフォルト結合生成ユーティリティを使用する。「ユーティリティ→デフォルト結合の生成」を選択し、「親の検索」チェックボックスを選択する

2つの表取扱細目の間に単独の外部キー制約が存在する場合、結合が作成されます。

注意: このサンプルでは、会社と顧客表のための参照結合と表取扱細目も生成できます。これは、メンバーシップ表取扱細目に、外部キー列CU_顧客番号とCO_顧客番号のためのキー列が含まれているからです。参照取扱については、この章ですでに説明した「参照取扱のインクルード」を参照してください。

6. レポート・モジュール定義を生成し、生成されたレポート・モジュールを実行します。

生成されるレポートの構成要素

前節で示したサンプル・レポート・モジュール定義は、次のレポートを生成します。

REPSJIS Page 2 of 3

メンバーシップタイプ

コード: G 記述: ゴールド 費用: 2,500.00 割引率: 20.00

メンバーシップ

メンバーシップ番号	満了日付	支払金額	コード	従業員番号	終了日付	支払状態
10008	95/11/01	2,000.00	G	2003		P
10013	95/09/01	2,000.00	G	2007		P
10016	93/16/31	2,500.00	G	2008		F
10017	95/10/31	2,500.00	G	2006		F
10019	94/16/30	2,500.00	G	2014		F
10020	95/16/30	2,500.00	G	2014		F
10023	94/11/30	2,500.00	G	2005		F

コード: S 記述: シルバー 費用: 1,500.00 割引率: 10.00

メンバーシップ

メンバーシップ番号	満了日付	支払金額	コード	従業員番号	終了日付	支払状態
10005	95/11/01	1,500.00	S	2005		F
10006	04/06/01	1,500.00	S	2002		F

グループ

最初の基礎表取扱細目からの情報を含むマスター グループが会員タイプごとに1回表示されます。2番目の基礎表取扱からの情報を含むディテール・グループは、各グループの会員リストを表示します。

注意: デフォルトでは、マスター・グループのフィールドは、ページ幅全体に渡って表示されます。このサンプルレポートでは、フレームの左側にフィールドを揃えるようにデフォルトが変更されています。デフォルトの整列方法を変更するには、CENTER環境設定の初期設定をLEFT_RIGHTからLEFTに変更します。レポート モジュールの環境設定については、第16章を参照してください。

レイアウト スタイル

ジェネレータは、マスター・グループに対してフォーム・レイアウト・スタイルを、ディテール・グループに対してタビュラ・レイアウトを適用しています。グループのデフォルト・レイアウトは、データ取扱を編集することで変更できます。グループのレイアウトの変更については、第15章を参照してください。

詳細グループの位置

ディテール・グループはマスター・グループの下に表示され、2文字分インデントされます。ディテール・グループとマスター・グループの位置は、FRAME環境設定の初期値を変更することで変更できます。



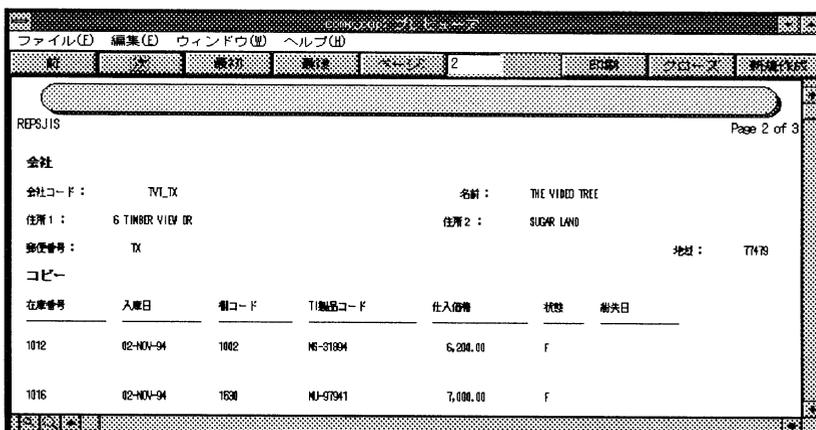
詳細情報: FRAME環境設定については、Oracle Reports ジェネレータのオンライン・マニュアルを参照してください。

項目グループの作成、編集、削除

複数のフィールドとそれらのフィールド・プロンプトを項目グループと呼ばれる1つのオブジェクトにグループ化できます。項目グループの作成によって、関連性のある情報を生成されるレポート上にまとめて表示できます。

注意: 項目グループを作成できるのは、フォーム・レイアウト・スタイルを持っているグループの中だけです。

たとえば次の図は、マスター・ディテール・レポートを示しています。マスター・グループはフォーム・レイアウトスタイルで表示され、項目グループを持っていません。



在庫番号	入庫日	品コード	TI製品コード	仕入価格	状態	納期日
1012	02-NOV-94	1002	NS-31994	6,000.00	F	
1016	02-NOV-94	1030	NJ-97941	7,000.00	F	

注意: デフォルトでは、マスター・グループのフィールドは、ページ幅全体に渡って表示されます。このサンプルレポートでは、CENTER環境設定の初期設定がLEFT_RIGHTからLEFTに変更されているため、フィールドはフレームの左側に揃えられています。レポートモジュールの環境設定については、第16章を参照してください。

次の手順に従って、列取扱細目を項目グループにグループ化します。

1. 項目グループに入れたい列取扱細目をすべて選択します。
2. 標準ツールバーから「列取扱細目のグループ化」アイコンを選択します。



または、「ユーティリティ→列取扱細目のグループ化」を選択します。

3. 作成する項目グループ名を「項目グループの選択/作成」ダイアログ・ボックスに入力し、「OK」をクリックします。

項目グループに列取扱細目を追加する場合は、ダイアログ・ボックスから項目グループ名を選択し、「OK」をクリックします。

項目グループにグループ化されたことを示すため、列取扱細目の名前が変化します。

4. グループのラベルを作成したり、デフォル・トレイアウト・スタイルを変更する場合は、項目グループを編集します。次の節「項目グループの編集」を参照してください。
5. レポート モジュール定義を生成し、生成されたレポート モジュールを実行します。

項目グループの編集

項目グループの作成後は、項目グループ名の変更、項目グループラベルの追加、レイアウト スタイルの変更を実行できます。

次の手順に従って、項目グループを編集します。

1. 「編集→要素→項目グループ」を選択します。
「項目グループの編集」ダイアログ・ボックスが表示されます。
2. 項目グループ名を選択し、「OK」をクリックします。
「項目グループ」ダイアログ・ボックスが表示されます。
3. 項目グループ名の変更や項目グループラベル（プロンプト）の追加、レイアウト スタイルの変更を実行します。
注意: レポートモジュール定義内の項目グループには、フォームとアドレスという2つのスタイルだけが適用されます。
4. 「OK」を選択して、変更を保存します。

項目グループからの列取扱細目の削除

列取扱細目とアイコンを選択することで項目グループから列取扱細目を削除できます。



または、「ユーティリティ→グループからの除去」を選択します。

生成されたレポートの構成要素

次の図は、生成されるレポートが列取扱細目のグループ化によってどのように変化するかを示しています。

在庫番号	入庫日	41コード	T1項目コード	単入価格	状態	船出日
1012	02-MX-94	1002	M6-31894	6,200.00	F	
1015	02-MX-94	1008	MU-0741	7,000.00	F	

項目グループ内の列取扱細目フィールドは、二次元フィールドとして表示されます。項目グループのレイアウトは変更可能です。

- 項目グループにタイトルを追加する場合は、列取扱細目を編集します。詳細は、第15章を参照してください。
- 項目グループか項目グループ タイトルの間隔や整列方法を変更する場合は、ITEM GROUP作業環境グループ初期設定を変更します



詳細情報: Item Group環境設定の詳細については、Oracle Reports ジェネレータのオンライン・マニュアルを参照してください。

- 項目グループの前景、背景、境界線の種類、幅を変更する場合は、レイアウトモデルテンプレートレポートを編集します。詳細は、第16章を参照してください。

ツリーウォーク結合の作成

レポートモジュール定義に、階層データを持つ表を基にした表取扱細目が含まれている場合、階層順に行を表示できます。これを実行するには、モジュール定義の中にツリーウォーク結合を作成します。

ツリーウォーク結合を含むモジュール定義を生成すると、ジェネレータは、生成される問い合わせに対してSTART WITH句とCONNECT BY句を追加します。問い合わせの構文は、モジュール定義を生成する前に環境設定の初期値を変更することで制御できます。

この節では、ツリーウォーク結合に関する次の情報について説明します。

- ツリーウォーク結合の作成規則
- 表取扱細目のためのツリーウォーク結合の手動による定義方法
- デフォルト結合生成ユーティリティを使用したツリーウォーク結合の作成方法
- ツリーウォーク結合のために生成される問い合わせの制御方法

ツリーウォーク結合の作成規則

ツリーウォーク結合の追加または生成を実行する前に、次の規則に注意してください。

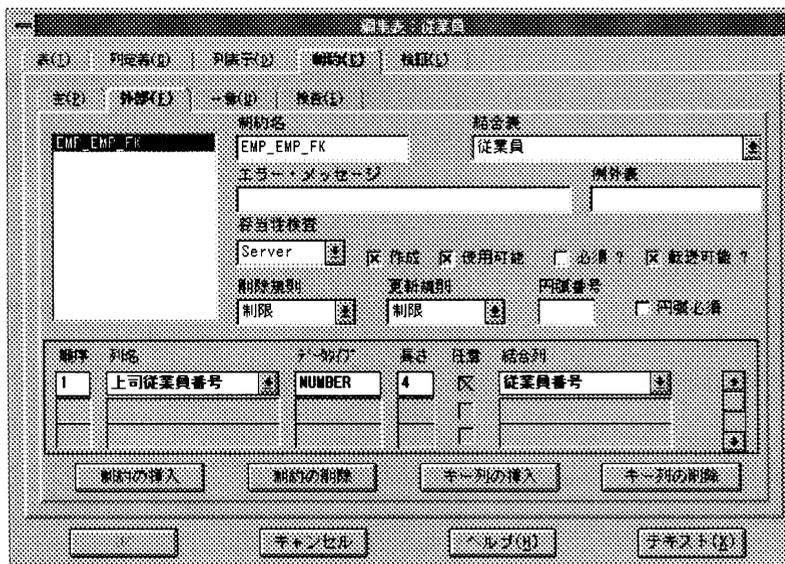
- ツリーウォーク結合を実装するために使用する外部キーは、再帰型外部キーでなければならない
- 再帰型外部キーは、キー構成要素を1つだけ持っていなければならない
- 基礎表取扱細目を使用してツリーウォーク結合を実装する。参照表取扱細目のためのツリーウォーク結合は作成できない
- 表取扱細目を同じ問い合わせの中でマスターまたはディテールの表取扱細目にリンクしない
- 表取扱細目を参照表取扱細目に結合しない
- これまでに表取扱細目のための結合を定義し、モジュール定義を生成している場合、それらの結合は、ダイアグラムから削除されている場合でも、依然としてリポジトリ内で表取扱細目のために定義されている

ツリーウォーク結合の定義

この節では、ダイアログ上での作成か、デフォルト結合生成ユーティリティの使用によるツリーウォーク結合の定義方法について説明します。

1. 再帰型外部キーを含む表のための基礎表取扱細目を作成します。

たとえば次の図は、EMP表上に定義されているEMP_SELF_KEY制約に対するキー制約を示しています。結合表もEMP表であり、EMP_SELF_KEYがMGR列に定義され、EMPNO列で結合されています。



2. 列取扱細目を作成します。

注意: 再帰型外部キーの両側の列取扱を定義しない場合、デフォルト結合生成ユーティリティの実行時かモジュール定義の生成時に、定義されていない側の列取扱細目が作成され、リポジトリ内にリバースされます。

3. ダイアグラム上に描画するかデフォルト結合生成ユーティリティを実行することでツリーウォーク結合を定義します。
 - 結合を描画する場合は、「結合」ツールを選択します。表取扱細目の上にカーソルを置き、1回クリックし、表取扱細目のどこかにカーソルを移動し、もう1回クリックします。
 - 結合を自動的に作成する場合は、「ユーティリティ→デフォルト結合の生成」を選択し、「ツリーウォーク結合の検索」チェックボックスを選択します。
4. レポート モジュール定義を生成します。

モジュール定義に対してパラメータをまったく作成しなかった場合、ジェネレータは次のようなメッセージを表示します。

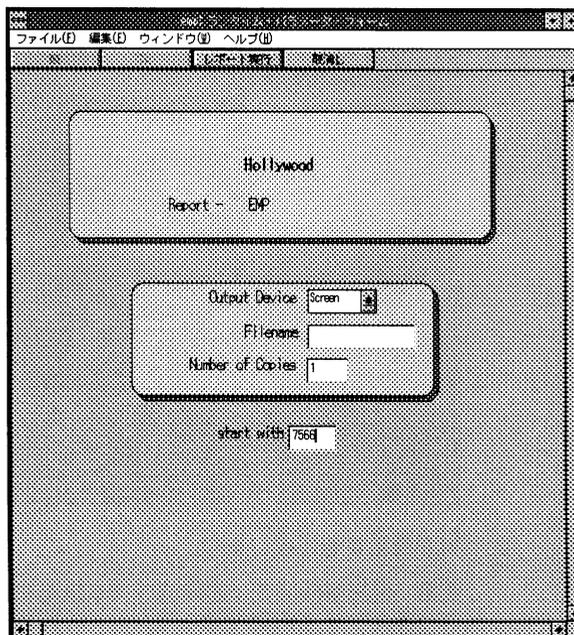
```
CGEN-02057: WARNING: No START WITH parameter exists  
for treewalk DTU EMP, a default START WITH parameter  
will be created.
```

(CGEN-02057:警告: ツリーワークDTU EMPのSTART WITHパラメータが存在しません。デフォルトSTART WITHパラメータが作成されます。)

ジェネレータは、パラメータ・フォーム・テンプレート・レポートを使用して、生成されるレポート・モジュールに含まれるパラメータ・フォームを作成します。

生成されるレポートの構成要素

生成されたレポートモジュールを実行すると、次のようなパラメータ・フォームが表示されます。



このパラメータ・フォームには、レポートを実行するためにユーザーが入力しなければならないパラメータ・フィールドが含まれています。

サンプル・パラメータ・フォームでは、パラメータ・フィールドの幅は4キャラクタであり、これはツリーワーク結合のソース列の列幅です。ユーザーは、レポート内のデータを表示するためのルートとして使用される行に相当する従業員番号を指定する必要があります。たとえばユーザーが7566と入力し、レポートを実行すると、次のようなレポートが表示されます。

従業員番号	名前	職名	上司
7076	AWNG	CLERK	7009
7082	FORD	ANALYST	7066
7066	JONES	MANAGER	7009
7088	SCOTT	ANALYST	7066
7009	SMITH	CLERK	7082

ルートとして、ユーザーがパラメータフィールドに入力した従業員番号が使用されています。

ツリーウォーク結合を含むレポート・モジュール定義の生成は、次の方法で制御できます。

- ツリーウォーク結合の列取扱細目のために生成されるデフォルトパラメータの長さか演算子をカスタマイズする場合は、TWPRLNまたはTWPROP環境設定の設定を変更する
- 生成される問合せのSELECT文かパラメータのデフォルト値にSTART WITH句を追加するかどうかを制御する場合は、TWSWSS環境設定の設定を変更する
- START WITH句にパラメータ名、演算子、列を追加するかどうかを制御する場合は、TWPRSW環境設定の設定を変更する
- ソースまたはキー列にCONNECTBY句のPRIORキーワードが追加されるかどうかを指定する場合は、TWTNFT環境設定の設定を変更する



詳細情報: 特定の環境設定については、Oracle Reports ジェネレータのオンライン・マニュアルを参照してください。

- 特定のパラメータを使用して、ジェネレータがデフォルトを作成しないようにする場合は、ツリーウォーク結合を定義するために使用される列取扱細目とパラメータを関連付けます。パラメータと列取扱細目の関連付けについては、第15章を参照してください。

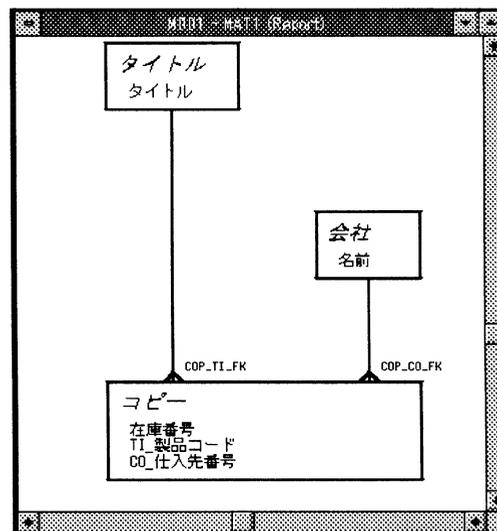
マトリックス・レポートの生成

マトリックス・レポートは、1行のラベル、1列のラベル、行ラベルと列ラベルの両方に関するグリッド形式の情報を持っています。

マトリックス・レポート・モジュールを生成するためには、少なくとも次のデータがモジュール定義に含まれていなければなりません。

- 同じグループ階層に属する2つのマスター・グループを表す2つの基礎表取扱細目。これらの基礎表取扱は、マスター・グループでなければならないことに注意。つまりこれらの取扱は、ダイアグラム上で上位にある表取扱細目とは結合できない
- 両方のマスター・グループのディテール・グループである1つの基礎表取扱細目

次の図は、サンプルモジュール定義を示します。



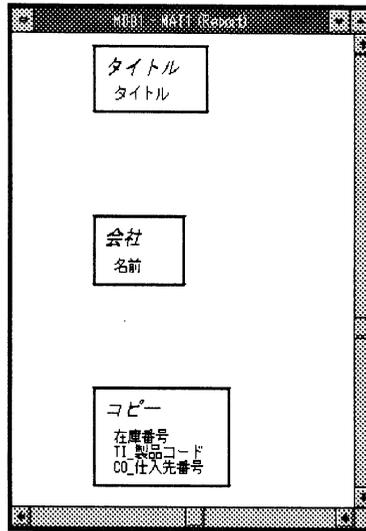
サンプル・モジュール定義は、次のようになっています。

- タイトル表取扱細目を使用して、ラベル行か横方向グループを生成する
- 会社表取扱細目テーブルを使用して、ラベル列か縦方向グループを生成する
- コピー表取扱細目を使用して、レポートのグリッドかセル内の情報を生成する

次の手順に従って、レポート・モジュール定義を作成し、マトリックス・レポート・モジュールを生成します。

1. 横方向、縦方向、セル・グループを表す3つの表取扱細目を作成します。

ヒント: 表取扱細目間の結合を作成したときに横方向グループの位置が変化しないようにするため、次の図に示すように横方向グループをダイアグラムの上部に、セル・グループを下部に移動しておきます。



2. 横方向グループのための表取扱細目とセル・グループのための表取扱細目の間に結合を描画します。
3. 縦方向グループのための表取扱細目とセル・グループのための表取扱細目の間に結合を描画します。横方向グループのための表取扱細目が、レポート内で垂直方向に情報を表示するグループのための表取扱細目の左側に表示されていることを確認します。
ダイアグラムは、この節で前に示したようになるはずです。
4. レポート・モジュール定義を生成します。

生成されるレポートの構成要素

次の図は、サンプル・レポート・モジュール定義から生成されたマトリックス・レポートを示します。

matrix1: Previewer			
Prev	Next	First	Last
Page: 5		Print	Close
File Name:	E. T.	AMADEUS	UNFORGIVEN
	<u>Number of Copies</u>	<u>Number of Copies</u>	<u>Number of Copies</u>
THE VIDEO TREE			2
MOVING PICTURES INC		1	
SCI-FI DISTRIBUTORS	4		
WILD WEST FILMS			3
GAMES GALORE			
VIVA VIDEO	2		
PACIFIC HYPER			

グループ・タイトルとセル

このレポートでは、映画の名前がレポートの横方向に、映画を配給する企業がレポートの左側に縦方向に表示されています。レポートのセルには、各企業によって配給されている映画の本数が含まれています。

注意: 映画の本数を表示するために、コピー表取扱細目の在庫番号列取扱細目に COUNT関数を持つレコード・レベル・サマリーが定義されています。サマリーの定義については、第15章を参照してください。

フィールド

TI_製品コードとSU_仕入先番号外部キー列を基にした列取扱細目のためのフィールドが作成されなかったことに注意してください。デフォルトでは、ジェネレータは表示列を基にしたすべての列取扱細目を表示します。ただしサンプルのレポート・モジュール定義では、これらのフィールドのための列取扱細目は、非表示フィールドになるように設定されています。生成されるレポートへのフィールドの非表示と表示については、第15章を参照してください。

フィールド・プロンプト

ラベル「Number of Copies」が、セル・グループ内のフィールドのためのフィールドプロンプトです。列取扱細目のためのプロンプトを削除することで、これらのラベルを生成されるレポートから削除できます。フィールド・プロンプトについては、第15章を参照してください。

レイアウト・スタイル

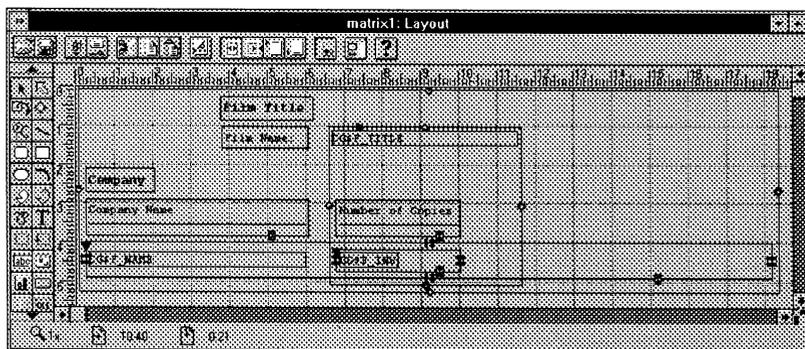
ジェネレータは、親グループ階層内の各グループと、子グループ階層内の各グループに対して次のデフォルト レイアウトを割り当てます。

グループが属する階層	適用されるレイアウト
横方向グループ階層	アクロス
縦方向グループ階層	タビュラ
セルグループ階層	タビュラ

マトリックス・レポートでは、グループのレイアウト・スタイルは変更できません。

グループ フレーム

レイアウト内にフレームを作成するとき、ジェネレータはグループを組み合わせ、横方向と縦方向のグループの反復フレームを結合させるマトリックス・フレームを作成します。セルグループ階層の反復フレームは、横方向と縦方向のグループの反復フレームの交点内に囲い込まれます。



ヒント: Oracle Reports Designer を使用してマトリックス・レポートの反復フレームのサイズを変更する場合は、横方向の親グループの反復フレームをマトリックス・フレームの下に拡張しなければならないことに注意してください。同様に、縦方向の親グループの反復フレームは、マトリックス・フレームの右側に拡張しなければなりません。

次の表は、ジェネレータが各反復フレームに割り当てる出力方向を示しています。

反復フレーム	出力方向
横方向	横
縦方向	縦
セル	縦

データ モデル

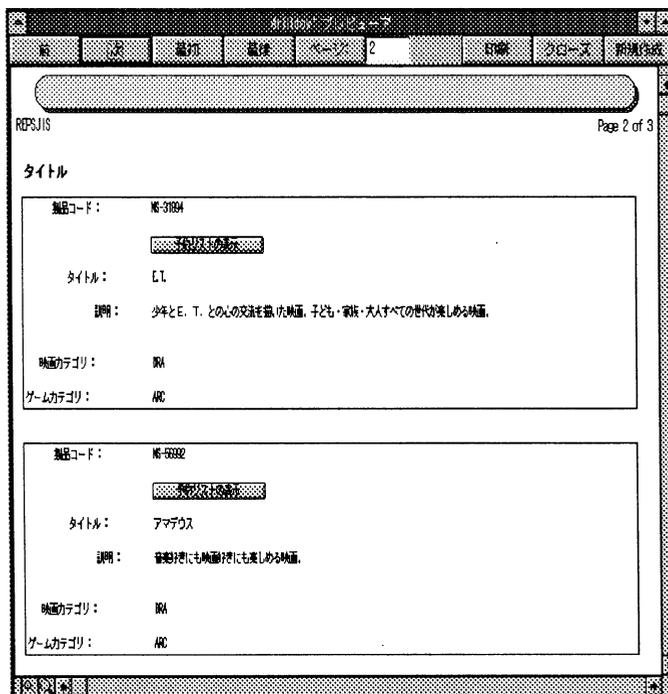
サンプルのレポートでは、ジェネレータは3つの表取扱細目を使用して生成データ・モデル内に4つのグループを作成しています。

グループ名	目的
G_CROSS	レポートの上と横を作成する2つのサブグループを含むスーパーグループかクロスプロダクトグループ。スーパーグループは、サブグループ内のカラムを参照できる。スーパーグループは、問い合わせを直接参照しない
G_WITHIN1	レポートの片側に表示されるフィールドを保持するスーパーグループ内の先頭サブグループ
G_WITHIN2	スーパーグループ内の第2サブグループ
G_MATRIX	レポートのセルに情報を提供するために使用される列を含む

ドリルダウン詳細レポート・モジュールの生成

ユーザーがレポート上のボタンをクリックすると、常に別のレポートを表示するレポートを生成できます。

たとえば次の生成レポートモジュールでは、予約状況テーブルから戻されたすべての行にボタンが表示されています。ユーザーが「List of Bookings」ボタンをクリックすると、その映画を予約している会員に関する情報を含む別のレポートが表示されます。



ドリルダウン詳細レポート・モジュールは、次の2つの方法で作成できます。

- 単独のモジュール定義を使用し、ダイアグラム上で新しいレポートを定義する
- モジュール・ストラクチャ・ダイアグラムで、複数のモジュール定義間の結合を定義する

単独のモジュール定義を使用したドリルダウンレポートの生成

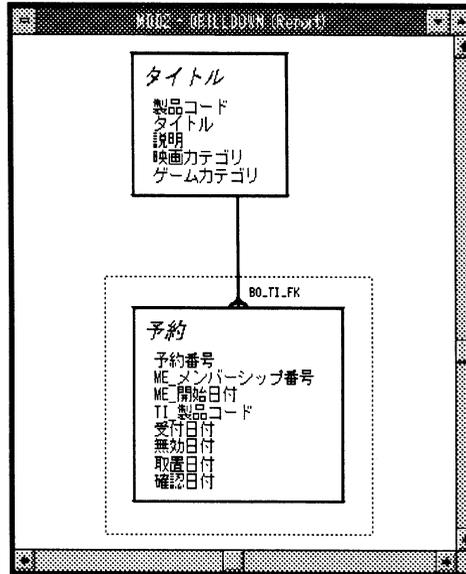
1つのモジュール定義を使用してドリルダウン詳細レポートを生成する場合は、レポートの基になる表取扱細目に対して新しいレポートを定義します。

1. 描画ツールバーの「新規ページ/レポート アイコン」 ボタンを選択します。



2. マウスボタンを押して、ドリルダウン・レポートを生成するために使用する基礎表取扱細目の周囲に境界線を描いてからマウスボタンを離します。

表取扱細目の周囲に境界線が描かれます。



3. モジュール定義を生成します。

ジェネレータは、2つのレポートを作成し、「Oracle Reports ジェネレータ」ダイアログ・ボックスにレポート名を表示します。他のレポートから呼び出されるレポートの名前は、そのレポートを定義している最初の基礎表取扱細目に基づいていることに注意してください。

ジェネレータは、ドリルダウン詳細テンプレート・レポートを使用して、ドリルダウン・レポートのために生成されるモジュールのレイアウトを作成します。テンプレートを使用したドリルダウン・レポートのカスタマイズについては、第16章を参照してください。

4. レポート・モジュールを実行します。

この節で前に示したレポートに類似したレポートが表示されます。

ヒント: 最初の表取扱細目のレイアウト・スタイルは、デフォルトであるフォーム・レイアウト・スタイルからアドレス・レイアウト・スタイルに変更されています。グループのレイアウト・スタイルの変更については、第15章を参照してください。

複数のモジュール定義の使用

2つのレポート・モジュール定義間の結合を定義することによって、ユーザーがレポート上のボタンを選択したとき常に別のレポートを呼び出すようなレポートを生成できます。

複数のモジュール定義を使用して、別のレポートを呼び出すレポートを生成するためには、次の4段階の手順に従います。

1. レポート間の関連情報をどのようにやりとりするように生成したいかを決めます。
2. モジュール定義と、必要であれば、生成されるレポート間で関連情報をやりとりするためのパラメータを作成します。
3. モジュール定義とそれを含むモジュール・ネットワークを作成し、各モジュールを生成します。
4. レポート・モジュール定義を生成し、生成された呼び出しレポートを実行します。

関連情報のやりとり

レポートの中に他のレポートの情報に関連する情報を作成するためには、レポート・モジュール定義が次のどれかの方法で相互に関連付けられていなければなりません。

- レポートを呼び出すレポート上に出力パラメータとして、またレポートから呼び出されるレポート上に入力パラメータとして、同じ名前と等号(=)演算子を持つパラメータを定義する
- 呼び出し側と呼び出される側のレポート・モジュール定義に、同じ表を基にした表取扱細目を作る。呼び出される側のモジュールでは、基礎表取扱細目はダイアグラム上の最初の表取扱細目である。さらにこのテーブルは、それに対して定義された一次または独自キー制約がある
- 呼び出される側のモジュール定義の最初の基礎表取扱細目に、パラメータと関連付けられている列取扱細目を最低1つ入れる。さらに、呼び出し側のモジュールに、呼び出される側のモジュール定義内の表上に定義されている外部キー制約に対応する列取扱細目を入れる

これらの条件が複数適用される場合、ジェネレータは、前述の説明順序に従って、レポート間での関連情報の受け渡し方法を決定します。

これらの条件がどれも適用されない場合、ジェネレータは、生成される呼び出し側レポート内にボタンを作成しますが、呼び出される側のレポートに表示される情報は、呼び出し側レポート内の行に関連する情報だけに限定されません。

モジュール定義の作成

ジェネレータが呼び出し側レポートと呼び出される側のレポート間でどのように情報をやりとりするかを決めたら、モジュール定義と、必要であればパラメータを作成し、そのやりかたを具現化します。

マッチング・パラメータの使用

パラメータを使用してレポート間で関連情報を渡す場合は、モジュール定義を作成した後、マッチング・パラメータを作成します。

1. 呼び出し側レポートのレポートモジュール定義を作成し、表取扱細目と列取扱細目を追加します。
2. 呼び出される側のレポートのレポートモジュール定義を作成し、表取扱細目と列取扱細目を追加します。
3. 呼び出される側のモジュール定義を編集して、そのモジュールのためのパラメータを作成します（パラメータの作成については、この章ですでに説明した「モジュール定義の編集」を参照してください）。

呼び出される側のモジュール定義の「パラメータ」タブで、次を実行します。

- パラメータ名の入力
 - 「演算子」列のドロップダウン・リストからの等号(=)の選択
 - 「In/Out」列のドロップダウン・リストからの「入力」の選択
4. 呼び出し側モジュール定義を編集して、そのモジュールのためのパラメータを作成します
呼び出し側モジュール定義の「パラメータ」タブで、次を実行します。
 - 呼び出される側のモジュールに指定したパラメータ名と同じパラメータ名の入力
 - 「演算子」列のドロップダウン・リストからの等号(=)の選択
 - 「In/Out」列のドロップダウン・リストからの「出力」の選択
 5. 入力パラメータを呼び出し側モジュール定義内の列取扱細目に関連付け、出力パラメータを呼び出される側のモジュール定義内の列取扱細目に関連付けます。パラメータと列取扱細目の関連付けについては、第15章を参照してください。

主キーまたは一意キー制約の使用

呼び出される側のモジュール定義内の最初の表取扱細目が、呼び出し側モジュール内の表取扱細目と同じ表を基にしている場合、ジェネレータは、生成されるレポート間で関連情報を渡すために必要な列取扱細目とパラメータを自動的に作成できます。

1. 他のレポートによって呼び出されるレポートのレポートモジュール定義を作成し、表取扱細目を最低1つ追加します。ダイアグラム上の最初の基礎表取扱細目が、モジュール定義間の関係を確立するためにモジュールによって使用されるものであることを確認します。

主キーまたは一意キー制約のキー列に対して列取扱細目を定義できます。列取扱細目が定義されない場合、生成プロセス中にジェネレータがそれらを定義します。

2. 他のレポートを呼び出すレポートのためのレポート・モジュール定義を作ります。呼び出される側のレポートの最初の表取扱細目と同じ表を基にする基礎表取扱細目を確実に作成してください。

主キーまたは一意キー制約のキー列に対して列取扱細目を定義できます。列取扱細目が定義されない場合、生成プロセス中にジェネレータがそれらを定義します。

外部キー制約の使用

ジェネレータは、外部キー制約を使用して呼び出し側と呼び出される側のレポート間で関連情報をやりとりできます。

1. 他のレポートから呼び出されるレポートのレポートモジュール定義を作成します。
2. 使用したい外部キー制約を含む表のための基礎表取扱細目を作成します。この基礎表取扱細目がダイアグラム上の最初の表取扱細目であることを確認します。
3. 主キーまたは一意キー制約のキー列に列取扱細目を定義します。
4. 呼び出される側のモジュール定義を編集して、そのモジュールのパラメータを作成します（パラメータの作成については、この章ですでに説明した「モジュール定義の編集」を参照してください）。

呼び出される側のモジュール定義の「パラメータ」タブで、次を実行します。

- パラメータ名の入力
 - 「演算子」列のドロップダウン・リストからの等号(=)の選択
 - 「In/Out」列のドロップダウン・リストからの「入力」の選択
5. パラメータを各キー列の列取扱細目と関連付けます。パラメータと列取扱細目の関連付けについては、第15章を参照してください。
 6. 他のレポートを呼び出すレポートのレポートモジュール定義を作成します。
 7. 外部キー制約の反対側にある表を基にした基礎表取扱細目を作成します。
 8. 外部キー制約のキー列に列取扱細目を定義します。

モジュール・ネットワークの作成

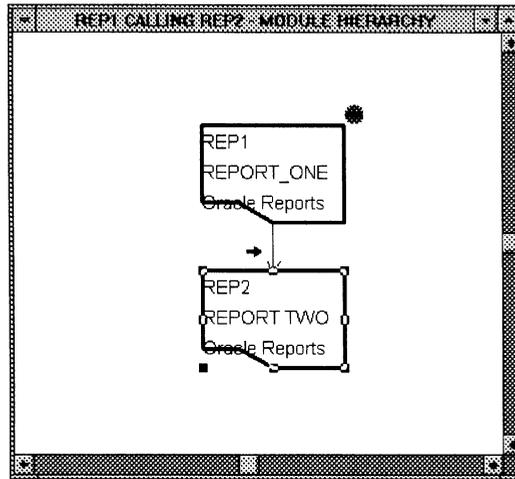
モジュール定義を生成する前に、それらを1つのモジュール・ネットワークにまとめて結合する必要があります。モジュール・ネットワークは、モジュール・ストラクチャ・ダイアグラムを使用して作成します。

1. 「ツール→モジュール・ストラクチャ・ダイアグラム」を選択して、モジュール・ストラクチャ・ダイアグラムを起動します。
2. 「新規作成」ツールバーボタンをクリックするか「ファイル→新規作成」を選択して、新しいダイアグラムを作成します。
3. 「編集→ネットワークのインクルード」を選択します。「データベースからモジュール・ネットワークをインクルード」ダイアログ・ボックスが表示されます。
4. ダイアログ・ボックスから選択することで、モジュール・ネットワークにインクルードするモジュール定義を指定します。
5. 「OK」をクリックします。モジュールがダイアグラム内に表示されます。
6. モジュールを結合するため、描画ツールバーから「モジュール移動」ボタンを選択します。



7. レポートから呼び出されるレポートモジュールをクリックし、カーソルをモジュールの最上部に移動します。

次の図に類似したモジュール階層が表示されます。



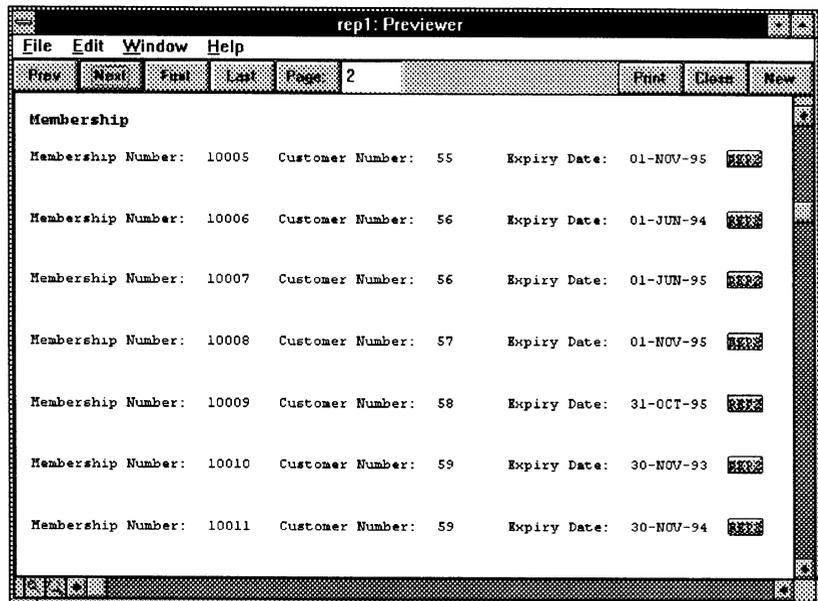
レポート モジュール定義の生成

レポート モジュール定義の作成と結合を実行した後、レポート モジュール定義を生成し、呼び出し側レポートを実行できます。

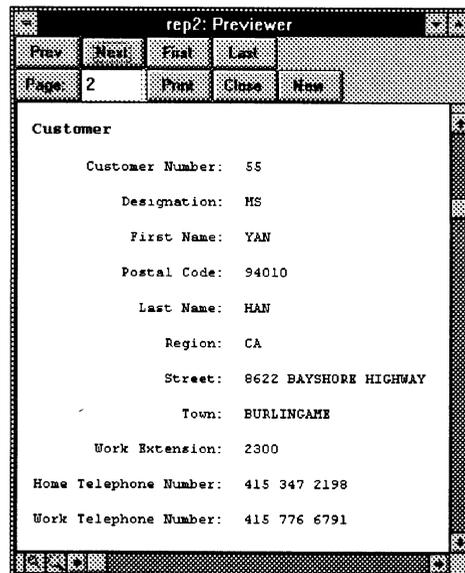
1. モジュール・ストラクチャ・ダイアグラムかモジュール・データ・ダイアグラムから、呼び出される側のレポート・モジュール定義を生成します。

注意: モジュール定義の生成順序には重要な意味があります。呼び出される側のモジュールを先に生成してください。

2. 呼び出し側モジュール定義を生成し、生成されたレポートを実行します。呼び出し側レポートの例を次の図に示します。



ユーザーがボタンをクリックすると、呼び出される側のレポートが表示されます。このレポートには、呼び出し側レポートの会員番号に相当する顧客情報が含まれています。



第15章

レポート・モジュールのデータ取扱の洗練

この章では、モジュール定義のためのデータ取扱を編集することで、レポートモジュール定義を洗練させる方法について説明します。データ取扱を変更することで、次の処理を実行できます。

- ブレーク・グループの作成
- パラメータと列取扱細目の関連付けによるパラメータ・フォームの生成
- サマリー・フィールドの明示的な定義
- ボイラープレート・テキストまたはグラフィックの追加
- グループとグループ・タイトルのカスタマイズ
- フィールドとフィールド・プロンプトのカスタマイズ

注意: 生成されるレポートは、環境設定とテンプレートレポートを使用することもカスタマイズできます。詳細は、第16章を参照してください。

レポート・モジュールのデータ取扱細目

モジュール定義内に表取扱細目と列取扱細目を作成した後、それらの取扱細目を編集して、生成されるレポートの一定の側面を制御できます（レポートモジュールの作成、表取扱細目、列取扱細目については、第14章を参照してください）。

取扱細目は、ダイアグラム内で表取扱細目をダブルクリックすると表示されません。

注意: 取扱細目の表示は、ダイアグラム内の表取扱細目を選択し、「編集→要素→取扱細目」を選択し、「表取扱細目の編集」ダイアログ・ボックスから表取扱細目を選択することでも実行できます。

モジュール・データ・ダイアグラムは、「レポートモジュールの取扱細目:<モジュール名>」ダイアログ・ボックスを表示します。

取扱についての情報は、次の5つのカテゴリに分類されます。

- 表の詳細
- 表テキスト
- 列細目
- 列表示
- 列テキスト

各カテゴリは、「レポートモジュールの取扱細目:<モジュール名>」ダイアログ・ボックスでタブによって表現されます。この節では、各取扱細目タブについて説明します。

表の詳細

「表の詳細」タブでは、次の処理を実行できます。

- 表別名の変更
- 生成されるレポート内で表取扱細目に対応するグループの上部に表示される表示タイトルの変更
- 生成されるグループのレイアウト・スタイルの変更
- 生成されるレポートの論理ページ上に表示する最大レコード数の指定
- 取扱が参照表取扱細目の場合の表取扱細目の外部結合の指定

さらに「表の詳細」タブは、次の情報を表示します。

- 表取扱細目の基となっている表またはビューの名前。表取扱細目がビューを基にしている場合は、「ビュー」チェックボックスが選択される
- ドリルダウン詳細レポート用の新しいレポートを作成するために表取扱細目が使用されているかどうか
- 表取扱細目が外部結合と関連付けられているかどうか。表取扱細目が基礎表取扱細目の場合、「外部結合」チェックボックスは、選択することも選択を解除することもできない

表テキスト



「表テキスト」タブでは、次の処理を実行できます。

- グループに対するWHERE句または妥当性検査の指定
- グループの説明やメモの追加

列細目

The screenshot shows a software window titled "レポート・モジュールのデータ取扱いの洗練" (Report Module Data Transfer Wash). The "列細目" (Column Detail) tab is active. On the left, a list of columns is shown, with "レンタル番号" (Rental Number) selected. The right side of the window contains configuration options for the selected column: "名前" (Name) is set to "レンタル番号"; "ブレイク" (Break) is unchecked; "Order By" is set to "20"; "方向" (Direction) is set to "降順" (Descending); "サマリー" (Summary) is checked; "タイプ" (Type) is set to "標準" (Standard); and "ソース" (Source) is set to "標準" (Standard). A "コメント" (Comment) field is also present. At the bottom, there are buttons for "OK", "キャンセル" (Cancel), and "ヘルプ" (Help).

「列細目」タブでは、次の処理を実行できます。

- 列取扱細目の別名の変更
- 生成されるレポート上に列取扱細目を表示するかどうかの指定
- 選択済みの列取扱細目でレポートがブレイクするかどうかの指定
- 生成されるレポート上でのORDER BY句による行の表示方法の制御
- 列取扱細目から生成されるグループとパラメータとの関連付け
- レポート、ページ、グループ、レコードサマリーの作成

列表示



「列表示」タブでは、次の処理を実行できます。

- 列取扱細目の別名変更
- オーバーフロー ページ上に列を表示するかどうかの指定
- 列取扱細目から生成されるフィールドのラベルの変更
- フィールドの整列方法の指定
- 生成されるレポート上のフィールドにデータを表示するために使用するデータ型の変更
- 生成されるフィールドに対するDATAまたはMONEY書式マスクの適用
- フィールドの折り返しまたは切り捨て
- フィールドの高さまたは幅の変更
- 生成されるレポート上での列取扱細目のグループ化

列テキスト



「列テキスト」タブでは、次の処理を実行できます。

- フィールドの導出表現の指定
- フィールドの強調条件の指定
- 生成されるレポート内でのユーザー テキストの指定
- フィールドの説明やメモの追加

ブレイク・グループの作成

デフォルトでは、基礎表取扱は、それぞれ別個のグループを指定します。ただし、表取扱細目に対してブレイク・グループを作成することによって、特定の列でレポートがブレイクするように指定できます。取扱細目にブレイクグループを追加することで、単独の表取扱細目によってレポート上に複数のグループを指定できます。

1. ブレイクを指定したいグループの表取扱細目のための「列細目」タブを表示します（タブの表示方法については、この章ですでに説明した「レポート・モジュールのデータ取扱細目」を参照してください）。
2. 「列取扱」リスト ボックスから列取扱細目を選択します。
3. 「ブレイク」内に最初のグループのための識別子を入力します。あらゆる文字を入力できます。この文字は、ブレイクグループ内に現れる列取扱細目を識別するだけです。
4. 列取扱細目を選択し「ブレイク」内に同じ識別子を指定することによって、ブレイク グループに他の列取扱細目を追加します。

注意: ブレイク・グループ内の列取扱細目は、表取扱細目の中で隣接していなければなりません。ブレイク グループに入れたい列が隣接していない場合は、列取扱細目を選択しドラッグすることで、表取扱細目の中での順序を変更してください。

5. 「OK」をクリックし、その後レポートモジュール定義を生成します。

ブレイクグループに属する列は、レポート上で別々のグループとして表示されます。

パラメータと列取扱細目との関連付け

パラメータと列取扱細目を関連付けることで、パラメータ・フォームを生成できます。パラメータは、パラメータ・フォームとレイアウト・モデル内のフィールドと、データ・モデル内の他の列とPL/SQLプロシージャからも参照できます。

注意: ツリーウォーク結合を定義する列取扱細目に作成されるパラメータについては、第14章を参照してください。

次の手順に従って、パラメータと列取扱細目を関連付けます。

1. 「列表示」タブを表示します (タブの表示方法については、この章ですでに説明した「レポートモジュールのデータ取扱細目」を参照してください)。
2. 次のどれかを実行します。
 - 「引数」グループの「名前」ドロップダウン・リストからパラメータを選択する。リストに表示されるパラメータは、モジュールのために作成されているパラメータです (パラメータの作成については、第14章を参照してください)。
 - 「名前」フィールドに名前を入力することで新しいパラメータを作成する
 - パラメータ・フォームを含むテンプレート レポート内に作成したパラメータ名を指定する。テンプレート内の一定のパラメータについては、第16章を参照してください。
3. 同じ手順で、他の列取扱細目に対してパラメータを追加します。1つのパラメータを複数の列取扱細目で参照できます。
4. 「演算子」ドロップダウン・リストからパラメータ演算子を選択します。ツリーウォーク結合を含む列取扱細目にパラメータを追加する場合は、**START WITH**演算子を選択します。

注意: **START WITH**の選択は、**START WITH**句をSQL文に追加するようにジェネレータに指示するだけです。ジェネレータは、実際は等号演算子(=)を**SELECT**文の**START WITH**句に追加します。

5. 「OK」を選択して、パラメータを保存します。
6. パラメータに一覧表がある場合、または妥当性検査を必要とする場合は、参照コード表にデータを入力します。



詳細情報: 参照コード表へのデータ入力については、Oracle Reportsジェネレータのオンライン・マニュアルを参照してください。

7. レポート モジュール定義を生成し、生成されたレポート モジュールを実行します。

生成されるパラメータ フォームの構成要素 レポートモジュールを実行すると、列取扱細目と関連付けられたパラメータが表示されます。

パラメータの順序 パラメータに順序番号を指定した場合、ジェネレータはレポート内のパラメータを順序番号に従って表示します。モジュール定義内のパラメータの編集については、第14章を参照してください。

または、CG\$PF<number>キーワードを使用してテンプレート・レポート内にパラメータ・フィールドを作成した場合、ジェネレータは、テンプレート・レポート内のフィールドの位置に従ってパラメータを表示します。テンプレート・レポートについては、第16章を参照してください。

オーバーフロー・ページ パラメータを表示するために複数のページが必要な場合、ジェネレータはオーバーフロー・ページを作成します。このページは、「PARAMETER」作業環境グループの初期値を変更することでカスタマイズできます。



詳細情報:パラメータの「PARAMETER」作業環境グループとキーワードについては、Oracle Reportsジェネレータのオンラインマニュアルを参照してください。

データ・モデル ジェネレータは、適切なWHERE句を問合せに付加します。

パラメータに対してレコード・レベルの集合関数が指定されている場合は、WHERE句ではなくHAVING句が問合せ内に作成されます。

パラメータに対してLIKE演算子を選択した場合は、SQL文の中の列に対してTO_CHAR関数が付加されます。

LIKE以外の演算子を選択した場合は、SQL文の中の列に対してTO_NUMBER関数が付加されます。

これでジェネレータは、LIKE演算子を使用してすべてのレコードまたはその一部を検索できます。

サマリーの明示的な作成

第14章では、いかにしてレポートおよび生成レポート内のグループ・レベル・サマリーを自動生成するかについて説明しました。ただし、生成されるレポートの中にさらに複雑なサマリーを入れたい場合は、列取扱細目に対してサマリーを指定することでそれらを作成する必要があります。

サマリーを明示的に作成することで、次の処理を実行できます。

- サマリーの関数、タイプ、ソース、表示属性の完全制御
- 非数値列取扱細目を使用したサマリーの作成
- 他の表取扱細目の列取扱細目を使用したサマリーの作成（クロスグループサマリー）
- ページまたはレコード・レベルでのサマリーの作成
- SUM以外の演算子を使用したサマリーの作成
- サマリー フィールドのラベルの指定

次の手順に従って、サマリーを明示的に作成します。

1. レポートモジュール定義のためのダイアグラムをオープンするか、新しいダイアグラムを作成します。
2. グループ、ページ、またはレポートレベルでサマリーを作成する場合は、レポートモジュール定義には、次の2つの列取扱細目が含まれていなければなりません。

- サマリーを求める列を指定する列取扱細目
- 生成レポート中にサマリー表示項目を作成するための列取扱細目

モジュール定義にこれらの列取扱細目が含まれていない場合は、追加してください。

3. サマリーを含むことになる表取扱細目の「列細目」タブを表示します（タブの表示方法については、この章ですでに説明した「レポートモジュールのデータ取扱細目」を参照してください）。

4. 「機能」ドロップダウン リストから関数を選択します。

SQL*Plusレポートのための列取扱を定義している場合は、AVG、COUNT、MAX、MIN、またはSUMのいずれかを選択します。

注意: ジェネレータは、実行中のページレベルサマリーはサポートしません。

5. 「タイプ」ドロップダウン・リストから実行するサマリーのレベルを選択します。

6. 「ソース」ドロップダウン・リストからソース列取扱の別名を任意に選択します。他の表取扱細目の列取扱細目の別名もソースとして一覧表示されることに注意してください。サマリーは、妥当な列取扱細目または導出された使用法のすべてに関して作成できます。

「ソース」から別名を選択しない場合、ソースは同じ表取扱細目の中にあると見なされます。

7. 「OK」をクリックして、サマリーを保存します。
8. レポート モジュール定義を生成し、生成されたレポート モジュールを実行します。

生成されるレポートの中にサマリーフィールドが作成されます。ページ レベルのサマリーを作成した場合は、生成されるレポートの余白にサマリーフィールドが追加されます。

9. サマリーフィールドを表示するために、それが基になっている列の長さを超える長さが必要な場合、サマリーのために作成される列取扱細目の表示幅を増やすことができます（詳細については、この章で後ほど説明する「レポート内のフィールドのカスタマイズ」を参照してください）。

ボイラープレート・テキストまたはグラフィックの追加

列取扱細目を編集して、生成されるレポート内に次の種類のボイラープレート・テキストとグラフィックを表示できます。

- イメージとOLEオブジェクト
- マルチメディア ボタン
- テキスト

イメージまたはOLEコンテナのインポート

列取扱細目を編集することで、レポートにイメージを追加できます。

次の手順に従って、イメージまたはOLEオブジェクトをレポートにインポートします。

1. イメージまたはOLEオブジェクトを含むフィールドに該当する表取扱細目のための「列細目」タブを表示します（タブの表示方法については、この章ですでに説明した「レポート・モジュールのデータ取扱細目の表示」を参照してください）。
2. 「表示データ型」ドロップダウン・リストから「イメージ (IMAGE)」または「OLEコンテナ (OLE CONTAINER)」を選択します。

注意: 列取扱細目が基にしている列がイメージをサポートしていない場合は、生成される列はファイル読み込みとしてマークされます。

3. ファイルに保存されているイメージまたはOLEオブジェクトを使用する場合は、イメージの完全なディレクトリ・パスとファイル名をユーザー・テキストとして列取扱細目に追加します。詳細については、この節で後ほど説明する「ユーザー・テキストの追加」を参照してください。
4. 「OK」をクリックして列取扱細目を保存し、その後レポート・モジュールを生成します。

フィールドを作成することで、生成されるレポートにイメージまたはOLEオブジェクトがインポートされます。

イメージまたはOLEオブジェクト・ファイル名を含むユーザー・テキストが追加されている場合は、生成されるレポート内の各行にイメージまたはOLEオブジェクトが表示されます。連結したファイルからこれらのイメージまたはオブジェクトを作成するため、イメージのためのボイラープレート連結ファイル・オブジェクトが作成されます。このボイラープレート連結ファイル・オブジェクトは、列取扱細目を基にしている他のフィールドと同じ属性（位置、幅、高さなど）が適用されます。

マルチメディア・ボタンの追加

生成されるファイルの中に、テキスト・ファイル、イメージ、ビデオクリップ、またはサウンドクリップを表示するボタンを作成できます。

1. マルチメディア・ボタンのためのフィールドに該当する表取扱細目のための「列表示」タブを表示します（タブの表示方法については、この章ですでに説明した「レポートモジュールのデータ取扱細目」を参照してください）。

2. 「表示データ型」ドロップダウンリストから「ボタン (BUTTON)」を選択します。

注意: 列取扱細目が基にしている列がイメージをサポートしていない場合は、生成される列はファイル読み込みとしてマークされます。

3. ボタンによってファイルに保存されているイメージ、サウンドクリップ、またはビデオクリップを表示する場合は、完全なディレクトリ・パスとファイル名をユーザーテキストとして列取扱細目に追加します。詳細については、この節で後ほど説明する「ユーザーテキストの追加」を参照してください。

4. 「表示フォーマット」に、イメージ (IMAGE)、サウンド (SOUND) またはビデオ (VIDEO) を入力します。

5. 「OK」をクリックして列取扱細目を保存し、その後レポート・モジュールを生成します。

フィールドを作成することで、生成されるレポート内にボタンが作成されます。

イメージ、サウンド、またはビデオ ファイル名を含むユーザー テキストが追加されている場合は、生成されるレポート内の各行にボタンが表示されません。連結したファイルからこれらのボタンを作成するため、ボタンのためのポイラプレート連結ファイル・オブジェクトが作成されます。このポイラプレート連結ファイル・オブジェクトには、列取扱細目を基にしている他のフィールドと同じ属性（位置、幅、高さなど）が適用されます。

ユーザー テキストの追加

レポート・モジュール定義の列取扱細目にユーザーテキストを指定できます。列取扱細目にユーザー テキストを追加することによって、次の処理を実行できます。

- 生成されるレポート内へのボイラープレート・テキストの作成
- 連結したファイルによるイメージ、OLEオブジェクト、またはマルチメディア・ボタンのレポートへのインポート

次の手順に従って、レポート・モジュール定義にユーザー テキストを追加します。

1. ユーザーテキストを含むフィールドに該当する表取扱細目のための「列テキスト」タブを表示します（タブの表示方法については、この章ですでに説明した「レポートモジュールのデータ取扱細目の表示」を参照してください）。
2. 「タイプ」ドロップダウンリストから「ユーザ・テキスト」を選択します。
3. 「タイプ」の下のテキスト ボックスにユーザー テキストを入力します。

または、イメージ用連結ファイルを追加する場合は、イメージの完全なディレクトリ・パスとファイル名を指定します。

アメリカ英語以外の言語を使用してユーザーテキストを指定する場合は、レポート モジュールを生成する前にNATLNG作業環境の値を変更する必要があります。



詳細情報: NATLNGと他の作業環境については、Oracle Reports ジェネレータのオンライン・マニュアルを参照してください。

4. イメージ、OLEオブジェクト、またはマルチメディア・ボタンのための連結ファイルを作成する場合は、前節の「マルチメディア・ボタンの追加」または「イメージまたはOLEオブジェクトのインポート」の説明に従って、列取扱細目を編集します。
5. 「OK」をクリックして、ユーザー テキストを保存します。
6. レポート モジュール定義を生成します。

フィールドを作成することで、生成されるレポート内にテキストがインポートされます（ジェネレータがどのようにイメージ、OLEオブジェクト、またはマルチメディアボタンを使用するかについては、前の節を参照してください）。

オーバーフロー ページに対する列の追加と削除

デフォルトでは、マスターグループがデータを表示するために複数のページを必要とする場合、ジェネレータはすべての列をオーバーフロー ページに表示します。すべての列をオーバーフロー ページに表示したくない場合は、表示する列を指定するか、オーバーフロー ページに表示する列をジェネレータに自動的に判断させることができます。

注意: オーバーフロー ページへの列表示を制御できるのは、レイアウトスタイルがフォームまたはアドレスのグループだけです。タビュラーまたはアクロスレイアウトスタイルのグループでは、すべての列がオーバーフロー ページに表示されます。

列の 手動選択

レポートモジュール定義を生成する前に、オーバーフロー ページに表示する列を手動で選択できます。

1. 「列表示」タブを表示します。
2. 生成されるレポートのオーバーフロー ページに表示する各列の「コンテキスト」チェックボックスを選択します。
3. プリファレンス・ナビゲータを使用して、CXDFLT作業環境の値がデフォルト値のNになっていることを確認します。プリファレンス・ナビゲータについては、第16章を参照してください。
4. プリファレンス・ナビゲータを使用して、CXDPPTH作業環境の初期値を変更します。CXDPPTHの値は、コンテキスト取扱が適用されるグループ階層内のマスター グループのレベル数を示します。

注意: CXDPPTH作業環境の初期値である0を変更しないと、すべての列がオーバーフロー ページに表示されます。

CXDPPTHに0、1、または2を設定したときに作成されるレポート出力については、この節で後ほど説明するサンプルを参照してください。

列の自動選択

次の手順に従うと、すべてのページで繰り返される列をジェネレータが自動的に判断します。

1. プリファレンス・ナビゲータを使用して、CXDFLT作業環境の値をYに変更します。プリファレンス・ナビゲータについては、第16章を参照してください。
2. プリファレンス・ナビゲータを使用して、CXDPATH作業環境の初期値を変更します。CXDPATHの値は、コンテキスト取扱が適用されるグループ階層内のマスターグループのレベル数を示します。

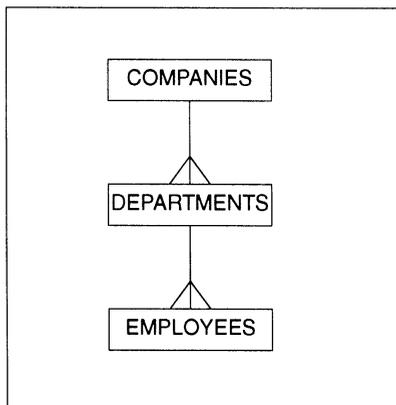
CXDPATHに0、1、または2を設定したときに作成されるレポート出力については、この後のサンプルを参照してください。

注意: CXDPATH作業環境の初期値である0を変更しないと、すべての列がオーバーフローページに表示されます。

レポートが生成されたとき、主キーの一部を構成する列か、記述子列として定義されている列だけが表示されます。

例

次の図は、3つのテーブル取扱のエンティティ関係図を示しています。



次の列取扱細目のための「コンテキスト」取扱が選択されていると仮定します。

- COMPANIESテーブルのCOMPANY NAME
- DEPARTMENTSテーブルのDEPARTMENT NUMBER

CXDPTH=2

CXDPTHに2が設定されている場合、ジェネレータは詳細グループフレームの第2レベルに対してコンテキスト取扱を適用します。つまり、DEPARTMENTSレコードのためのEMPLOYEESレコードが複数のオーバーフローページを必要とする場合、ジェネレータは、オーバーフローページを表示するときにDEPARTMENTSレコードとCOMPANIESレコードのためのコンテキスト取扱を適用します。これは、EMPLOYEESレコードがグループ階層の第2レベルではないからです。

したがって、DEPARTMENT NUMBERとCOMPANY NAMEのためのコンテキスト取扱が選択されているため、オーバーフローページにはこれらのフィールドだけが表示されます。

CXDPTH=1

CXDPTHに1が設定されている場合、ジェネレータは詳細グループフレームの第1レベルだけにコンテキスト取扱を適用します。つまり、DEPARTMENTSレコードのためのEMPLOYEESレコードが複数のオーバーフローページを必要とする場合、ジェネレータは、オーバーフローページを表示するときにCOMPANIESレコードのためのコンテキスト取扱だけを適用します。これは、DEPARTMENTレコードがグループ階層の第1レベルではないからです。

したがって、オーバーフロー ページにはCOMPANY NAMEだけが表示されま
す。DEPARTMENTS取扱のために指定されたコンテキスト取扱は無視され、
DEPARTMENTSレコードのすべてのフィールドがオーバーフローページに表示
されます。

ヒント: オーバーフロー ページにCOMPANIESレコードのフィールドをまっ
たく表示しない場合は、CXDFLTに初期値であるNを設定し、CXDPATHに1を設定
し、マスターグループ内の列取扱細目に対してコンテキスト取扱を選択しませ
ん。

CXDPATH=0

CXDPATHが0の場合、コンテキスト取扱は無視され、COMPANIESと
DEPARTMENTSレコードのすべてのフィールドがオーバーフローページに表示
されます。

レポート内のグループのカスタマイズ

ユーザーが表取扱細目を作成すると、モジュール・データ・ダイアグラマがデータ取扱に対してデフォルト値を割り当てます。これらのデフォルト値は、表定義またはモジュール定義に対して指定された値を基にしています。

ユーザーがレポート モジュール定義を生成すると、ジェネレータはこれらの値（と作業環境とテンプレート）を使用して生成されるレポート内にグループを作成します。レポートモジュールを生成する前にデータ取扱を編集することで、次の処理を実行できます。

- 別名またはグループの変更または作成
- グループ タイトルの変更
- ブレーク グループのタイトルの作成
- グループのレイアウト スタイルの変更
- 論理ページに表示されるグループのレコード数の制限
- 参照グループのための外部結合の使用
- グループに戻される行の制限

別名の追加または変更

グループのデフォルトの別名は、表定義内に指定されている表別名から導出されます。別名が指定されていない場合は、表名から導出されます。

モジュール定義に、同じ表を基にした複数の表取扱細目が含まれている場合は、別名の末尾に番号が追加されます。番号は、ダイアグラム上の表取扱細目の位置に従って、順に追加されます。

次の手順に従って、グループの別名を変更します。

1. グループに該当する表取扱細目のための「表の詳細」タブを表示します（タブの表示方法については、この章ですでに説明した「レポートモジュールのデータ取扱細目の表示」を参照してください）。
2. 別名を変更します。

グループタイトルの 変更

ユーザーがレポートを生成すると、ジェネレータは表示タイトルを使用してグループタイトルを作成します。グループが複数の表取扱細目を基にしている場合、タイトルは生成されるレポート内で結合されます。

グループのデフォルト表示タイトルは、表定義内に指定されている表示タイトルから導出されます。表示タイトルが指定されていない場合は、表名から導出されます。

次の手順に従って、グループタイトルを変更します。

1. グループに該当する表取扱細目のための「表の詳細」タブを表示します（タブの表示方法については、この章ですでに説明した「レポートモジュールのデータ取扱細目の表示」を参照してください）。

2. タイトル名を変更します。

注意: グループタイトルを作成しない場合は、タイトルを削除し、FTDFLT作業環境の初期値をNに変更します（この作業環境値を変更しないと、表取扱細目の基になっている表名からタイトルが作成されます）。

3. マトリックスグループのタイトルを作成する場合は、FTCELL作業環境の初期値をYに変更します。

4. 「OK」を選択し、その後レポート・モジュールを生成します。

指定したタイトルが、生成されるレポートにコピーされます。グループ内に複数の表取扱細目が表示される場合、または複数のグループを使用してグループ・フレームを作成している場合、タイトルはスラッシュ(/)によって分離されます（FTCHRS作業環境を変更することでグループフレームタイトルを分離する文字列を変更している場合は、その文字列が使用されます）。

注意: 指定したタイトルが使用されない場合は、FTCRET作業環境に初期値であるYが設定されていることを確認します。

5. グループタイトルの外観は、作業環境の値を変更するか、レイアウトモデルテンプレートを使用してフォントまた可視属性を適用することでカスタマイズできます。詳細は、第16章を参照してください。

ブレイク グループ タイトルの作成

ブレイク グループを作成した場合は、ブレイク グループ構造体の中の各グループに対してタイトルを指定できます（ブレイクグループの作成については、この章ですでに説明した「ブレイク グループの作成」を参照してください）。

次の手順に従って、ブレイク グループ タイトルを作成します。

1. ブレイクグループに該当する表取扱細目のための「表の詳細」タブを表示します（タブの表示方法については、この章ですでに説明した「レポートモジュールのデータ取扱細目の表示」を参照してください）。
2. 「表示タイトル」にタイトル名を入力します。タイトルは、縦棒文字(|)で区切ります。

FTBRKS作業環境の初期値が別の文字列に変更されている場合は、その文字列を使用してタイトルを区切ります。

注意: ブレイクグループタイトルを作成しない場合は、タイトルの代わりに縦棒文字を入力します。

3. 「OK」を選択し、その後レポート・モジュールを生成します。

指定したタイトルが、生成されるレポートにコピーされます。ブレイク グループに該当しないタイトルは無視されます。

4. グループ タイトルの外観は、作業環境の値を変更するか、レイアウト モデル テンプレート レポートを使用してフォントまたは可視属性を適用することでカスタマイズできます。詳細は、第16章を参照してください。

レイアウトスタイルの変更

ユーザーがモジュール内に表取扱細目を作成すると、ジェネレータはモジュールフォーマットを使用してグループのレイアウトスタイルを決定します。グループのレイアウトスタイルによって、グループ フレーム本体でのフィールドとプロンプトの表示方法が決まります。

デフォルトでは、複数のマスターグループを持つ詳細グループがモジュールに含まれていない限り、すべてのモジュールに対してマスター・ディテールモデルフォーマットが割り当てられます（モジュールフォーマットの変更については、第14章を参照してください）。

注意: マトリックスレポートを生成する場合は、デフォルトレイアウトを変更しないでください。マトリックスレポートの生成では、この取扱に対して行われた変更は無視されます。

次の手順に従って、グループのレイアウトスタイルを変更します。

1. グループに該当する表取扱細目の「表の詳細」タブを表示します（タブの表示方法については、この章ですでに説明した「レポートモジュールのデータ取扱細目の表示」を参照してください）。
2. 「レイアウト・スタイル」ドロップダウン・リストからレイアウト・スタイルを選択します。
3. 「OK」を選択し、その後レポート・モジュールを生成します。

表示されるレコード数の制限

デフォルトでは、生成されるレポートの論理ページに合わせて、グループのレコードが表示されます。

次の手順に従って、表示されるレコードの最大数を指定します。

1. グループに該当する表取扱細目の「表の詳細」タブを表示します（タブの表示方法については、この章ですでに説明した「レポートモジュールのデータ取扱細目の表示」を参照してください）。
2. 「最大レコード/ページ」に最大レコード数を入力します。
3. 「OK」を選択し、その後レポート・モジュールを生成します。

参照のための 外部結合の使用

参照グループに外部結合 (OUTER JOIN) を適用すると、結合表とマッチする行だけでなく、すべての行が問合せにより返されます。

デフォルトでは、次の条件に該当する場合、参照表取扱細目に対して外部結合が作成されます。

- 複数の列取扱細目が、列定義内でオプション（任意入力）として定義されている列を基にしている
- 同じ列に複数の外部キーが定義されている

次の手順に従って、外部結合を指定します。

1. 参照グループに該当する表取扱細目の「表の詳細」タブを表示します（タブの表示方法については、この章ですでに説明した「レポートモジュールのデータ取扱細目の表示」を参照してください）。
2. 「外部結合」チェックボックスを選択します。

注意: 外部結合は、基礎表取扱細目には指定できません。

戻される行の制限

表取扱細目について生成されるSQL文にWHERE句を追加できます。

次の手順に従って、WHERE句を追加します。

1. グループに該当する表取扱細目の「表テキスト」タブを表示します（タブの表示方法については、この章ですでに説明した「レポートモジュールのデータ取扱細目」を参照してください）。
2. 「タイプ」ドロップダウンリストから「Where/Validation条件」を選択します。
3. 「タイプ」の下のテキストボックスにWHERE句を入力します。
4. 「OK」をクリックしてSQL文を保存し、その後レポート・モジュール定義を生成します。

レポートのフィールドのカスタマイズ

ユーザーが列取扱細目を作成すると、モジュール・データ・ダイアグラマがデータにデフォルト値を割り当てます。これらのデフォルト値は、列定義またはモジュール定義のために指定された値を基にしています。

ユーザーがレポート モジュール定義を生成すると、ジェネレータはこれらの値（と作業環境とテンプレート）を使用して、列のために生成されるフィールドを作成します。レポートモジュールを生成する前にデータ取扱を編集することで、次の処理を実行できます。

- フィールドの別名の変更または作成
- フィールドの表示と非表示
- フィールドに表示される値の順序の制御
- フィールド プロンプトの追加または変更
- フィールドの整列方法の変更
- フィールドの表示データ型の変更
- フィールドに対するDATEまたはMONEY書式の適用
- フィールドの幅または高さの変更
- 項目グループに対するフィールドの追加または削除、項目グループ名の変更、項目グループに対するフィールド プロンプトの追加または変更

注意: サマリーフィールド、ブレイクグループ、パラメータ、ボイラープレートテキスト、グラフィック、ボタンの作成方法と、オーバーフローページに表示されるフィールドの制御方法については、この章の前半の関連する節を参照してください。

エイリアスの追加または変更

フィールドのデフォルトの別名は、列定義に指定されている列別名から導出されます。別名が指定されていない場合は、列名から導出されます。

モジュール定義に、同じ表を基にした複数の表取扱細目が含まれている場合は、別名の末尾に番号が追加されます。番号は、ダイアグラム上の表取扱細目の位置に従って、順に追加されます。

次の手順に従って、グループの別名を変更します。

1. フィールドに該当する列取扱細目のための「列細目」タブを表示します（タブの表示方法については、この章ですでに説明した「レポートモジュールのデータ取扱細目」を参照してください）。
2. 別名を変更します。

表示と非表示

デフォルトでは、表取扱細目に列取扱細目が追加されると、ジェネレータは常に列定義の設定を使用して、取扱のために生成されるフィールドを生成されるレポート上に表示するかどうかを判断します。デフォルトを変更して、生成されるレポート上に列取扱のためのフィールドが作成されないようにできます。

次の手順に従って、生成されるレポートにフィールドを表示するか非表示にします。

1. フィールドに該当する詳細カラム使用法の「列細目」タブを表示します（タブの表示方法については、この章ですでに説明した「レポートモジュールのデータ取扱細目の表示」を参照してください）。
2. 「表示」チェックボックスを選択するか、選択を解除します。
3. 「OK」をクリックし、その後レポートモジュール定義を生成します。

順序の制御

列に対してORDER BY句を指定して、生成されるレポート内でフィールドに戻される値の順序を制御できます。

次の手順に従って、ORDER BY句を指定します。

1. ORDER BY句を指定する列の「列細目」タブを表示します（タブの表示方法については、この章ですでに説明した「レポートモジュールのデータ取扱細目の表示」を参照してください）。
2. ORDER BY句で最初の列になる列取扱細目に対して1を、2番目の列になる列取扱細目に対して2を入力します。
3. 「方向」ドロップダウン・リストから昇順か降順を選択します。
4. 「OK」をクリックし、その後レポートモジュール定義を生成します。

フィールド プロンプトの追加または変更

ジェネレータは、列取扱細目に指定されたプロンプトを使用して、フィールドのフィールド プロンプトを作成します。

次の手順に従って、フィールド プロンプトを作成します。

1. プロンプトを変更または追加したい列取扱細目を含む表取扱細目をダブルクリックし、「列表示」タブを表示します。（タブの表示方法については、この章ですでに説明した「レポートモジュールのデータ取扱細目の表示」を参照してください）。
2. プロンプト名を変更します。

注意: フィールドのプロンプトを作成したくない場合は、プロンプトを削除します。

3. プロンプトを複数行に分割する場合は、プロンプトを分割したい位置に円記号(¥)を挿入します。SPLCHRの初期値が他の分割プロンプトマーカーに変更されている場合は、そのマーカーを使用してプロンプトを分割します。

ヒント: デフォルトでは、生成されるレポート内でスペースが節約できる場合だけジェネレータはプロンプトを分割します。すべてのプロンプトを分割する場合は、プリファレンス・ナビゲータを使用してSPLPMT作業環境の初期値をYに変更します。

4. フィールドの前後に表示されるプロンプトを作成する場合は、プロンプトを入力し、縦棒(|)で区切ります。PFPCHR作業環境の初期値が別のマーカーに変更されている場合は、そのマーカーを使用してプロンプト部分を区切ります。

ヒント: デフォルトでは、フィールドの前に表示されるプロンプト部分の後ろにコロン(:)が挿入されます。この文字は、PRSPTRとSUPSEP作業環境の初期値を変更することで変更できます。

5. 「OK」をクリックし、その後レポート・モジュール定義を生成します。

指定したプロンプトが、生成されるレポートにコピーされます。

6. プロンプトの外観は、作業環境の値を変更するかフォントまたは可視属性に適用されるレイアウト・モデル・テンプレートを使用することでカスタマイズできます。

フィールドに隣接するプロンプトでは、マーカーの前にあるプロンプト部分が生成されるフィールドの前に表示され、マーカーの後ろにあるプロンプト部分が生成されるフィールドの後ろに表示されます。プロンプトの縦棒の後ろにスペースが入っている場合、生成されるフィールドの後ろに表示されるプロンプトの中にもスペースが入ります。

フィールドの上部にあるプロンプトでは、マーカーの前にあるプロンプト部分が生成されるフィールドの上部に表示され、マーカーの後ろにある部分が生成されるフィールドの下部に表示されます。縦棒の後ろにスペースが入っている場合、これらのスペースは無視され、スペース以外の部分が生成されるフィールドの下部に表示されます。

整列方法の変更

デフォルトでは、列取扱細目の位置は、列の位置から導出されます。レポートを生成すると、列取扱細目に指定された整列方法を使用してグループ内のフィールドの位置が揃えられます。

次の手順に従って、フィールドの位置を揃えます。

1. フィールドに該当する列取扱細目の「列表示」タブを表示します（タブの表示方法については、この章ですでに説明した「レポートモジュールのデータ取扱細目の表示」を参照してください）。
2. 「列取扱」リスト ボックスから列取扱細目を選択します。
3. 「揃え」ドロップダウン・リストから整列方法を選択します。
4. 「OK」をクリックし、その後レポート・モジュール定義を生成します。

表示データ型の変更

デフォルトでは、フィールドの表示データ型は、列定義に対して指定された表示データ型から導出されます。列に対して表示データ型が定義されていない場合は、列のデータ型から導出されます。

次の手順に従って、デフォルト・データタイプを変更します。

1. フィールドに該当する列取扱細目の「列表示」タブを表示します（タブの表示方法については、この章ですでに説明した「レポートモジュールのデータ取扱項目表示」を参照してください）。

2. 「表示データ型」ドロップダウン・リストからデータ型を選択します。

注意: 表示データ型がボタン、イメージ、またはOLEコンテナであるフィールドの作成については、この章の前半の関連する節を参照してください。

3. 「OK」をクリックし、その後レポート・モジュール定義を生成します。

DATEまたはMONEY書式マスクの適用 生成されるレポートで日付または通貨の値を表示するフィールドに対して、書式マスクを適用できます。

次の手順に従って、DATEまたはMONEY書式マスクを適用します。

1. 変更したいフィールドの列取扱細目を含む表取扱細目の「列表示」タブを表示します（タブの表示方法については、この章ですでに説明した「レポートモジュールのデータ取扱細目の表示」を参照してください）。
2. 「表示データ型」ドロップダウン・リストからフィールドのデータ型を選択します。フィールドの日付書式マスクは、DATE表示データ型で指定できます。フィールドの金額書式マスクは、MONEYまたはそれ以外のすべての数値データ型で指定できます。
3. 「表示型式」に何らかのOracle Report書式マスクを入力します。

注意: 書式マスクを適用しない場合、デフォルトの日付と金額の書式マスクであるDD-MON-YYと9,999,990.00が使用されます。デフォルト書式マスクは、DTEMSKとMNYMSK作業環境によって指定されます。

4. 「OK」をクリックし、その後レポートモジュール定義を生成します。

生成されるレポートのフィールドに日付と金額の書式マスクが適用されます。

フィールドの折り返しと切り捨て

デフォルトでは、グループフレーム内にフィールドを表示するだけのスペースがない場合、ジェネレータは、プロンプトの分割、行内でのフィールド移動、またはフィールドの上部へのフィールドプロンプトの表示を行うことでフィールドを表示しようとします。どの方法を使用してもフィールドが表示できない場合、ジェネレータは、フィールドの表示長が列取扱細目の表示幅の80パーセントを超える限り、利用できる幅に合うようにフィールドを切り捨てます。

デフォルトを変更して、ジェネレータがフィールドの表示ではなく切り捨てを実行するように設定できます。

グループフレーム内に十分なスペースがなければフィールドを切り捨てる場合は、次の手順に従います。

1. フィールドに該当する列取扱細目のための「列表示」タブを表示します（タブの表示方法については、この章ですでに説明した「レポートモジュールのデータ取扱細目の表示」を参照してください）。

2. 「折返し」ドロップダウンリストから「切捨て」を選択します。

注意: この値は、フィールドの水平および垂直のサイズプロパティにも影響します。詳細は、この章で後ほど説明する「フィールドの幅または高さの変更」を参照してください。

3. 「OK」をクリックし、その後レポート・モジュール定義を生成します。

ヒント: 利用できる表示幅が一定のパーセント未満しかない場合でもフィールドの切り捨てを実行させたくない場合は、「折返し」のデフォルト設定を変更しないでください。代わりに、「折返し」と設定されていることを確認し、TRCPCT作業環境の値を変更してください。



詳細情報: 作業環境については、Oracle Reportsジェネレータのオンライン・マニュアルを参照してください。

フィールドの幅または高さの変更

デフォルトでは、フィールドの幅は、列定義のために指定された表示幅から導出されます。列に対して表示データ型が定義されていない場合は、列幅から導出されます。フィールドの高さは、1に設定されます。

次の手順に従って、フィールドの幅または高さを変更します。

1. フィールドに該当する列取扱細目の「列表示」タブを表示します (タブの表示方法については、この章ですでに説明した「レポートモジュールのデータ取扱の表示」を参照してください)。
2. 高さと幅を指定します。
3. 「OK」をクリックし、その後レポート・モジュール定義を生成します。

ユーザーがレポート・モジュール定義を生成すると、ジェネレータは「レイアウト・スタイル」に設定されている値を使用して、生成されるフィールドが固定サイズであるか可変の水平/垂直サイズであるかを判断します。

グループのレイアウトスタイル	水平サイズ	垂直サイズ
フォーム、アドレス、またはタブビュラ	固定	「折返し」使用法に「折返し」または<ヌル>が設定されている場合は可変
		「折返し」使用法に「切捨て」が設定されている場合は固定
アクロス	「折返し」使用法に「折返し」または<ヌル>が設定されている場合は可変	固定
	「折返し」使用法に「切捨て」が設定されている場合は固定	

ジェネレータは、グループ反復フレームに対してこれらのOracleプロパティを設定します (これらのプロパティを参照する場合は、Oracle Reports Designerでレイアウト・エディタを使用します)。

「折返し」については、この節ですでに説明した「フィールドの折り返しまたは切り捨て」を参照してください。

項目グループの カスタマイズ

第14章で、ダイアグラム上での項目グループの作成と編集方法について説明しました。項目グループの作成と編集は、データ取扱を編集することでも実行できます。

注意: 項目グループのレイアウトスタイルを変更するためには、ダイアグラム上で項目グループを編集する必要があります。レイアウトスタイルは、データ取扱からは変更できません。詳細は、第14章を参照してください。

次の手順に従って、項目グループを変更または作成します。

1. 項目グループに対する追加または削除を実行したいフィールドに該当する列取扱細目のための「列表示」タブを表示します（タブの表示方法については、この章ですでに説明した「レポート モジュールのデータ取扱細目の表示」を参照してください）。
2. 項目グループに列取扱細目を追加する場合は、「名前」ドロップ ダウンリストから項目グループを選択します。
3. 項目グループに対して他の列取扱細目を追加するか削除します。
4. 項目グループにプロンプトを指定する場合は、「プロンプト」フィールドにプロンプトを入力します。
5. 「OK」をクリックし、その後レポート・モジュール定義を生成します。

導出表現の作成

レポート内で導出表現を参照できます。レポート内で導出表現を参照することで、外部列別名を参照するサマリーを作成できます。

次の手順に従って、生成されるレポート内で参照できる導出表現を作成します。

1. 導出表現を参照するグループの列取扱細目のための「列テキスト」タブを表示します（タブの表示方法については、この章ですでに説明した「レポートモジュールのデータ取扱細目の表示」を参照してください）。
2. 「Type」ドロップダウン・リストから「導出表現(Derivation Expression)」を選択します。
3. 導出表現を入力します。

ヒント：導出表現にサマリー関数を入れ、列取扱細目の明示的な別名を参照することで複雑なサマリーを作成できます。たとえば、次の導出表現を入力します。

```
SUM(DCU_X) * 10
```

レポートモジュールを生成すると、DCU_Xを合計するサマリー列が作成され、新しいサマリー列を参照するようにPL/SQLコードが修正されます。

4. 「OK」を選択して導出表現を保存し、その後レポートモジュールを生成します。

生成プロセス中に、ジェネレータは、Oracle Reports 2.5の列のデータ型を表示データ型によって指示されたデータ型に設定します。これによって列のデータ型と導出が指定されている列との間で競合が発生する場合は、警告が表示されます。

ジェネレータは、列幅に次の2つの値の大きいほうの値を設定します。

- 下敷きとなっている列の幅
- 表示高で乗算された表示幅の値

ジェネレータは、大きいほうの値を選択して、もつづいているデータベースの列が提供する幅よりも大きな幅を必要とする導出表現がある項目を処理します。

列テキストを使用したレポートトリガーの呼び出し

レポートモジュールの列取扱細目からレポート・トリガーを呼び出せます。列取扱細目からレポート・トリガーを呼び出すことで、個々のフィールドの属性を列取扱細目の中に直接定義できます。

次の手順に従って、列取扱細目からレポート・トリガーを呼び出します。

1. 列取扱細目を含む表取扱細目を選択します。
2. 「列テキスト」タブを表示します（タブの表示方法については、この章ですでに説明した「レポートモジュールのデータ取扱細目の表示」を参照してください）。
3. 「タイプ」ドロップダウン・リストから「強調条件 (Highlighting Condition)」を選択します。
4. 「タイプ」ドロップダウン・リストの下のボックスに、PL/SQL名と入力値を入力します。たとえば、次のような条件を入力できます。

```
CGFMT$ change_fld_font('Courier', 12);
```

この条件は、次のPL/SQLトリガーを呼び出します。

```
PROCEDURE change_fld_font (p_font in char, p_size in
number) IS
-- Use the given font and size to alter the font
-- that is used to display the field.
-- Parameters:
--   p_font:the name of the font; for example, 'Courier'
--   p_size:the size of the font; for example, 10
BEGIN
srw.attr.mask := SRW.SZ_ATTR +-- Sets up mask to define
                    SRW.FACE_ATTR-- formatting attributes.
srw.attr.face := p_font;      -- Adds given font to the
                                -- mask.
srw.attr.sz :=   p_size;      -- Adds given size to the
                                -- mask.
srw.set_attr := (0, srw.attr);-- Applies format mask to
                                -- the object.

END;
```

5. 「OK」をクリックしてトリガーを保存し、その後レポート・モジュール定義を生成します。

第16章

テンプレートと作業環境の使用

この章では、作業環境の初期値を変更しテンプレート・レポートをカスタマイズすることで、生成されるレポート モジュールを制御する方法について説明します。この章では、特に次の事柄について説明します。

- Oracle Reports v2.5とSQL*Plus生成レポートのための作業環境のカテゴリ
- 作業環境レベルと設定
- テンプレート・レポートの作成とアタッチ
- テンプレート・レポートを使用して生成されるレポート モジュールのデータ モデルとレイアウトの変更

注意: SQL*Plusレポート モジュールを作成する場合、ジェネレータはテンプレート レポートを使用しません。

作業環境の初期値の変更

ユーザーがレポートを生成すると、Reportsジェネレータは常に作業環境を使用して生成プロセスを制御します。各設定は初期値または「出荷時設定値」が設定されているため、レポートを生成する前にユーザーが作業環境を設定する必要はありません。

ただし、作業環境の値を変更することでデータモデル、レイアウト、パラメータフォームを変更し、生成されるレポートをカスタマイズできます。

プリファレンス・ナビゲータの起動

作業環境の設定は、プリファレンス・ナビゲータを使用して変更します。プリファレンス・ナビゲータはグラフィックベースのDesigner/2000システム デザイナ ツールであり、作業環境を階層リストで表示します。



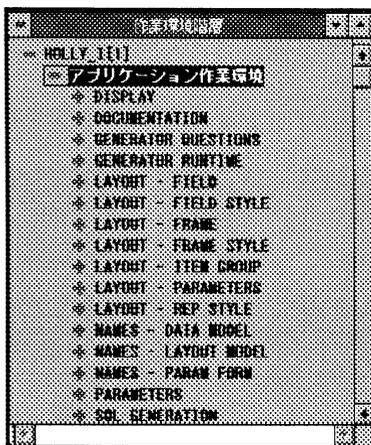
詳細情報: プリファレンス・ナビゲータについては、プリファレンス・ナビゲータのオンライン・マニュアルを参照してください。

1. モジュール・データ・ダイアグラマで作業している場合は、「ツール→プリファレンス・ナビゲータ」を選択することでプリファレンス・ナビゲータを起動できます。
または、プログラムマネージャからアイコンをダブルクリックすることでプリファレンス・ナビゲータを起動できます。
2. プログラム・マネージャからプリファレンス・ナビゲータを起動した場合は、ユーザー名、パスワード、接続文字列の入力と、アプリケーションシステムの選択を求められます。
適切な情報を入力し、「OK」をクリックします。
3. 「製品のフレーバー」ダイアログ・ボックスが表示されたら、「製品」フィールドからOracle Reportsジェネレータを選択し、「フレーバー」フィールドからOracle Reports Version 2.5かSQL*Plus Reportsを選択します。
「作業環境階層」と「プロパティ・ウィンドウ」ウィンドウが表示されます。
4. 「作業環境階層」ウィンドウで、アイコンをクリックするか名前をダブルクリックして、「アプリケーション作業環境」か「モジュール作業環境」ノードを展開します。
作業環境のカテゴリが表示されます。

作業環境のカテゴリ

作業環境は「作業環境階層」ウィンドウでカテゴリ別に分類されます。

Oracle Reports



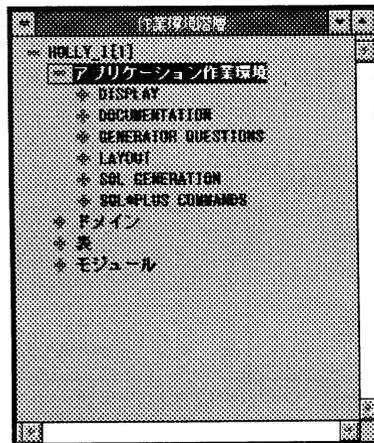
Oracle Reports Version 2.5の作業環境のカテゴリと、各カテゴリによって制御されるレポート オブジェクトを次の図に示します。

作業環境のカテゴリ	カスタマイズの対象
DISPLAY	レポート下部のタイトル、社名、フィールド書式マスク、レポートの言語
DOCUMENTATION	文書フォーマットの文字列
GENERATION QUESTIONS	生成プロセス中に表示される質問（デフォルトの応答を指定することで生成プロセス中のジェネレータによるプロンプト表示を抑止できる）
GENERATION RUNTIME	ドメイン、ジャーナル列、START WITHパラメータ、キー列内の列の生成
LAYOUT-ITEM GROUP	生成されるレポートでの項目グループの表示
LAYOUT-FIELD	フィールドとフィールド プロンプトの間隔
LAYOUT-FIELD STYLE	フレーム内のフィールドとフィールド プロンプトの位置
LAYOUT-FLAME	枠の周囲の境界線、フレーム間の間隔、改ページ
LAYOUT-FLAME STYLE	サマリーのレイアウト スタイル、フレームとマトリックスセルのタイトル、ボイラープレートの線、グループの折り返し、フレーム内のオブジェクトの位置
LAYOUT-PARAMETERS	パラメータ フィールドのサイズと位置

作業環境のカテゴリ	カスタマイズの対象
LAYOUT-REP STYLE	レイアウト モデル内へのレポート本体の作成、オーバーフロー ページに表示されるフィールド、詳細およびシプリング グループの位置
NAMES-DATA MOD- EL	データ モデル内に生成されるオブジェクトの名前
NAMES-LAYOUT MODEL	レイアウト モデル内に生成されるオブジェクトの名前
NAMES-PARAM FORM	パラメータ フォーム内に生成されるオブジェクトの名前
PARAMETERS	値、評価、エラー メッセージのパラメータ リスト
SQL GENERATION	SQLとPL/SQLオブジェクト
TEMPLATE	テンプレート レポートの種類と位置

SQL*Plus

「製品フレーバー」としてSQL*Plusを選択した場合は、次のカテゴリが表示されます。



SQL*Plusの作業環境のカテゴリと、各カテゴリによって制御されるレポート オブジェクトを次の図に示します。

作業環境のカテゴリ	カスタマイズの対象
DISPLAY	レポート下部のタイトル、社名、フィールド フォーマット マスク、レポートの言語
DOCUMENTATION	文書フォーマットの文字列
GENERATION QUESTIONS	生成プロセス中表示される質問（デフォルトの応答を指定することで生成プロセス中のジェネレータによるプロンプト表示を抑止できる）
LAYOUT	フィールドのサイズとプロンプトの配置

作業環境のカテゴリ	カスタマイズの対象
SQL GENERATION	SQLとPL/SQLオブジェクト
SQL*Plus COMMANDS	SQL*Plusコマンド ファイルの内容

作業環境レベルと 設定の操作

作業環境は、次の3つの方法でレポート モジュールに適用できます。

- アプリケーション・システム・レベルで作業環境を設定する
- モジュール・レベルで作業環境を設定する
- 名前付きの作業環境設定を作成し、それらをアプリケーション・システム
かモジュールに適用する

一般に、作業環境は、アプリケーション・システム・レベルで変更されます。このレベルで作業環境を設定すると、アプリケーション・システム内のモジュールから生成されるすべてのレポートの外観が揃います。

ただし、特定のモジュールのデータ モデル、レイアウト、またはパラメータ フォームをカスタマイズしたい場合は、モジュールレベルで作業環境を変更できます。

作業環境設定は作成して保存できます。名前付き設定を作成することで、同じ作業環境をアプリケーション・システム内の複数のレポートモジュールにただちに適用できます。名前付き作業環境設定は、他のアプリケーション・システム内のモジュールにも適用できます。

設定の変更

作業環境は、アプリケーションかモジュール・レベルで変更できます。レポートモジュールを生成するときにジェネレータが使用する設定は、ユーザーが使用する作業環境レベルとユーザーが選択した生成オプションによって決まります。

- アプリケーション・レベルで作業環境の初期値を変更した場合、そのアプリケーション・システムでレポート モジュールを生成すると、その値が使用される
- モジュール・レベルで作業環境の初期値を変更した場合、ユーザーがアプリケーション レベルで初期環境を変更しているかどうかに関係なく、ジェネレータはモジュール レベルの設定を使用する
- レポートを生成する前に、「レポートの生成」ダイアログ・ボックスから作業環境レベルを選択できます。詳細は、この章で後ほど説明する「使用する作業環境の指定」を参照してください。

次の手順に従って、作業環境を変更します。

1. 「作業環境階層」ウィンドウで、アイコンをクリックするか名前をダブルクリックして、「アプリケーション作業環境」か「モジュール作業環境」ノードを展開します。
2. モジュールレベルの作業環境を変更する場合は、モジュール名のノードを展開してから「モジュール作業環境」ノードを展開します。
3. 作業環境カテゴリのノードを展開し作業環境をクリックすることで、変更したい作業環境を選択します。
4. 「プロパティ」ウィンドウで、作業環境を選択します。

アプリケーションかモジュール名の下に入力フィールドが表示されます。

5. このフィールドに、新しい環境を入力するか、ドロップダウン・リストから値を選択します。

注意: コンテキストヘルプ・アイコンを選択し作業環境をクリックすることで、作業環境とその設定についての詳細情報を取得できます。

作業環境アイコンが、初期値が変更されたことを示すように変化します。

6. 「ファイル→上書き保存」を選択して、変更を保存します。

ユーザーがレポートを生成すると、レポート出力に対して実行した変更が適用されます。

名前付き設定の作成

次の手順に従って、すべてのモジュールに適用できる作業環境設定を作成します。

1. 「設定→作成」を選択し、セット名を入力し、「OK」をクリックします。「名前付き設定」ウィンドウに名前付きセットのノードが作成されます。
2. 名前付き設定のノードを展開し、設定を変更します。
3. この名前付き設定を他のアプリケーション・システムで使用する場合は、アプリケーション・システムを変更してから「設定→コピー」を選択します。名前付き設定のアイコンが、この名前付き設定は他のアプリケーション・システムと共用されていることを示すように変化します。

名前付き設定のアタッチ

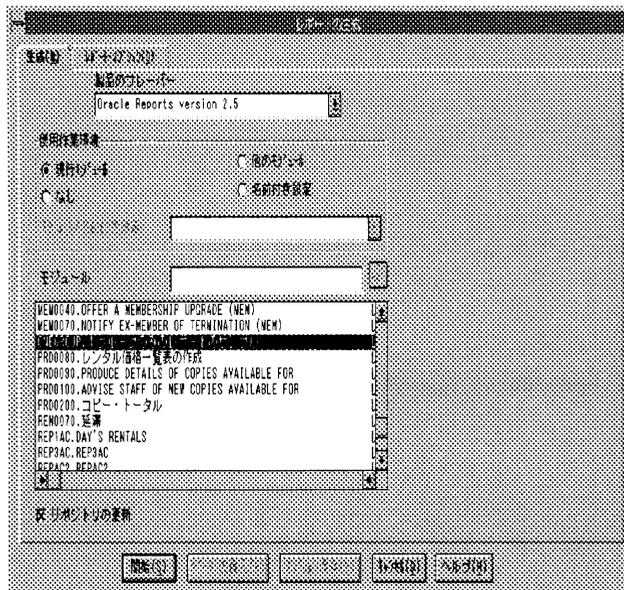
名前付き設定を作成した後、モジュールからそれを参照できます。

1. 「モジュール」ノードを展開し、モジュールを選択します。
2. 「設定→参照」を選択し、名前付き設定を選択します。

注意: モジュールから名前付き設定を参照しても、モジュールを生成するときにその設定を使用すると指定したわけではありません。レポートを生成する前に、「レポートの生成」ダイアログ・ボックスから「名前付き設定」オプションを選択する必要があります。詳細は、次の節を参照してください。

使用する作業環境の指定

レポートモジュールを生成するときは、「レポートの生成」ダイアログ・ボックスの「使用作業環境」グループから「現行モジュール」オプションを選択します。



この選択を変更しない場合、初期設定が使用されます。ただし、次の設定の集合を使用してモジュールを生成できます。

- 名前付き作業環境設定
- 別のモジュールのモジュール設定
- アプリケーションレベルの設定

次の手順に従って、使用する作業環境を指定します。

1. 「ユーティリティ→生成」を選択します。
2. 「レポートの生成」タブで、「使用作業環境」グループから使用したい作業環境の種類を選択します。

注意: アプリケーションレベルの設定を使用する場合は「なし」を選択します。「なし」を選択すると、モジュールレベルで作業環境に対して実行した変更が無効になります。

テンプレート・レポート

テンプレート作業環境の初期値が変更されなかった場合、ジェネレータは、レポート・モジュールまたはアプリケーション・システムに適用されているデフォルトのレイアウトモデルとパラメータフォームを使用します。必要に応じて、ドリルダウン詳細テンプレートレポートも使用します。

これらのデフォルトテンプレートレポートを修正するか、独自のテンプレートレポートを作成しそれらをレポートモジュールかアプリケーションシステムに適用することで、生成されるレポートをカスタマイズできます。

注意: SQL*Plusレポートモジュールを作成するとき、ジェネレータはテンプレートレポートを使用しません。

この節では、次の処理の実行方法について説明します。

- テンプレートレポートの適用
- テンプレートレポートの作成
- テンプレートレポートを使用したデータモデルのカスタマイズ
- テンプレートレポートを使用したレイアウトのカスタマイズ

テンプレートレポートの適用

デフォルトでは、次のテンプレートレポートがアプリケーションシステムに適用されています。

- CGBMPT.RDF。ビットマップレポートのレイアウト用であり、ページの向きは垂直方向
- CGBMPF.RDF。ビットマップパラメータフォーム用
- CGBMDT.RDF。ビットマップドリルダウン詳細レポート用（レポート内レポート）

別のデフォルトテンプレートレポートか独自のテンプレートレポートを使用する、またはレポートレベルでレポートにテンプレートを適用する場合は、「TEMPLATE」作業環境グループを使用してテンプレートを適用できます。

レイアウトモデルテンプレートレポート

レイアウトモデルテンプレートレポートを作成または修正した後、モジュールかアプリケーションシステムにそれを適用し、その後レポートモジュールを生成します。

注意: デフォルトレイアウトテンプレートレポートであるCGBMPT.RDFを修正した場合は、PARREP作業環境の初期値を変更していない限り、モジュールにそれを適用する必要はありません。

次の手順に従って、レイアウトモデルテンプレートレポートを適用します。

1. プリファレンス・ナビゲータを起動し、「TEMPLATE」ノードを選択します
(Oracle Reports作業環境でプリファレンス・ナビゲータを使用することの詳細は、この章ですでに説明した「作業環境の初期設定の変更」を参照してください)。
2. 「プロパティ」ウィンドウからLAYREP作業環境を選択します。
3. テンプレート・レポートの名前とフルパスを入力します。

 キャラクタモードのレポートを生成する場合は、ジェネレータが備えているデフォルト キャラクタモード テンプレートであるCGCHPT.RDFやCGCHLS.RDFなどのキャラクタモードレイアウトモデルテンプレートを適用してください。

 ページの向きが水平方向のレポートを生成する場合は、ジェネレータが備えているデフォルト水平テンプレートであるCGBMLS.RDFやCGCHLS.RDFなどの水平レイアウト モデルテンプレート レポートを適用してください。
4. レイアウト モデルテンプレート レポートがデータベースに保存されている場合は、プリファレンス・ナビゲータを使用して、LAYDEV作業環境の値をDATABASEに変更します。
5. レポート モジュールを生成するユーザーが、適用したレイアウト モデルテンプレートレポートに対する正当なアクセス権限を持っていることを確認します。

パラメータ フォーム テンプレート レポート

パラメータ フォーム テンプレート レポートを作成または修正した後、モジュールかアプリケーション・システムにそれを適用し、その後レポートモジュールを生成します。

注意: デフォルト パラメータ フォーム テンプレート レポートであるCGBMPF.RDFを修正した場合は、PARREP作業環境の初期値を変更していない限り、モジュールにそれを適用する必要はありません。

次の手順に従って、パラメータ フォーム テンプレート レポートをアタッチします。

1. プリファレンス・ナビゲータを起動し、「TEMPLATE」ノードを選択します
(Oracle Reports作業環境でプリファレンス・ナビゲータを使用することの詳細は、この章ですでに説明した「作業環境の初期設定の変更」を参照してください)。
2. 「プロパティ」ウィンドウからPARREP作業環境を選択します。
3. テンプレート・レポートの名前とフルパスを入力します。

LAYREPに指定したテンプレート レポートと同じテンプレート レポートをPARREPに指定することで、レイアウト モデルテンプレート レポートをパラメータ フォームのためのテンプレートとして使用できます。

キャラクタモードのレポートを生成する場合は、ジェネレータが備えているデフォルトキャラクタモードテンプレートであるCGCHPT.RDFなどのキャラクタモードレイアウトモデルテンプレートを適用してください。

注意: レイアウト・モデルに対して指定したのと同じ種類のテンプレート・レポートをパラメータ・フォームに対して指定してください。つまり、レイアウト・モデルにキャラクタ・モードのテンプレート・レポートを指定した場合は、パラメータ・フォームにもキャラクタ・モードのテンプレート・レポートを指定します。パラメータ・フォームに別の種類のテンプレート・レポートを指定した場合、ジェネレータは、レイアウト・モデルのテンプレート・レポートをパラメータ・フォームのテンプレート・レポートとして使用します。

4. パラメータ・フォームのテンプレート・レポートがデータベースに保存されている場合は、プリファレンス・ナビゲータを使用して、PARDEV作業環境の値をDATABASEに変更します。
5. レポートモジュールを生成するユーザーが、適用したレイアウト・モデル・テンプレート・レポートに対する正当なアクセス特権を持っていることを確認します。

ドリルダウン詳細テンプレート レポート

ドリルダウン詳細テンプレート・レポートを作成または修正した後、モジュールかアプリケーション・システムにそれを適用し、その後レポート・モジュールを生成します。

注意: デフォルトのドリルダウン詳細 テンプレート・レポートであるCGBMDT.RDFを修正した場合は、DETREP作業環境の初期値を変更していない限り、モジュールにそれを適用する必要はありません。

次の手順に従って、ドリルダウン・テンプレート・レポートを適用します。

1. プリファレンス・ナビゲータを起動し、「TEMPLATE」ノードを選択します (Oracle Reports作業環境でプリファレンス・ナビゲータを使用することの詳細は、この章ですでに説明した「作業環境の初期設定の変更」を参照してください)。
2. 「プロパティ」ウィンドウからDETREP作業環境を選択します。
3. テンプレート・レポートの名前とフルパスを入力します。

LAYREPに指定したのと同じテンプレート・レポートをDETREPに指定することで、レイアウト・モデルのテンプレート・レポートをドリルダウン詳細レポートのテンプレートとして使用できます。

キャラクタモードのレポートを生成する場合は、ジェネレータが備えているデフォルトのキャラクタモード・テンプレートであるCGCHPT.RDFやCGCHLS.RDFなどのキャラクタモードのレイアウトモデル・テンプレートを適用してください。

4. テンプレート・レポートがデータベースに保存されている場合は、プリファレンス・ナビゲータを使用して、DETDEV作業環境の値をDATABASEに変更します。

5. レポートモジュール定義を生成するユーザーが、適用したテンプレートに対する正当なアクセス権限を持っていることを確認します。

テンプレート レポートの作成

ジェネレータが提供しているテンプレート・レポートを修正または使用したくない場合は、独自のテンプレート・レポートを作成し、レポート・モジュールとともにそれを使用できます。

次の手順に従って、テンプレート・レポートを作成します。

1. モジュール・データ・ダイアグラマから「ツール→Oracle Reports Designer」を選択します。
2. 新しいレポートを作成します。
3. Oracle Reports Designerでテンプレート・レポートを保存して.rdfファイルを作成するか、データベースにファイルを保存します。
4. この章ですでに説明した「テンプレート・レポートの適用」に従って、新しいテンプレート・レポートをレポート・モジュールに適用します。
5. レポート・モジュール定義を生成します。

ジェネレータは、ユーザーが作成したテンプレートを使用してレポートを生成します。

テンプレートレポートを使用したデータ・モデルのカスタマイズ

テンプレート・レポートを使用して、Oracle Reports v2.5のデータ・モデルをカスタマイズできます。テンプレート・レポートを編集することで、次の処理を実行できます。

- 生成されるレポート内でのデータ・モデルのサイズと位置のカスタマイズ
- パラメータの定義
- レポート・トリガーの追加
- PL/SQLプロシージャの追加とPL/SQLライブラリのアタッチ
- フォーミュラ列の追加

サイズと位置の設定

テンプレート・レポートを使用して、データ・モデルの標準サイズと位置を指定できます。サイズと位置の指定によって、生成されるすべてのレポートのデータ・モデルに対して標準的な外観を確立できます。

次の手順に従って、データ・モデルのサイズと位置を設定します。

1. モジュール・データ・ダイアグラマから「ツール→Oracle Reports Designer」を選択します。
2. テンプレート・レポートをオープンします。
3. データ・モデル・エディタを呼び出します。
4. データ・モデルのサイズと位置を指定します。
5. テンプレート・レポートを保存しクローズします。
6. この章ですでに説明した「テンプレートレポートの適用」に従って、新しいテンプレート・レポートをレポート・モジュールに適用します。
7. レポート・モジュール定義を生成します。

ジェネレータは、生成されるレポートに対してこのデータ・モデルのサイズと位置を適用します。

テンプレート・レポート内の パラメータの定義

第14章では、モジュールの定義を編集することでパラメータを作成する方法について説明しました。パラメータは、テンプレート・レポートに追加してから列取扱細目に関連付けることでも作成できます。

テンプレート・レポート内にパラメータを作成することで、次のような拡張パラメータ機能を提供できます。

- カスタム デフォルト値
- アプリケーション標準アクセス
- クライアント側インターフェースとしてOracle Formsのフォームを使用するカスタム・パラメータ・フォーム
- アクセス制御

次の手順に従って、テンプレート・レポート内にパラメータを作成します。

1. モジュール・データ・ダイアグラマから「ツール→Oracle Reports Designer」を選択します。
2. パラメータ・フォームを含むテンプレート・レポートをオープンします。
注意: 複数のテンプレート・レポート内に同じパラメータを作成しないでください。
3. データ・モデルを展開します。
4. 「ユーザー・パラメータ」下に新しいパラメータを作成します。
注意: Reportsジェネレータが提供しているデフォルトのテンプレート・レポートには、サンプル・パラメータが含まれています。これらのパラメータを参考にしてください。
5. テンプレート・レポートを保存しクローズします。
6. 必要であればレポート・モジュールにテンプレート・レポートを適用します。
7. パラメータを列取扱細目と関連付けます。手順については、第15章を参照してください。
注意: 列取扱細目が基づいている列とパラメータのデータ型が同じであることを確認してください。違う場合は、モジュール定義を編集することでパラメータのデータ型を変更します。
8. レポート・モジュール定義を生成します。

パラメータの例

当日の日付をデフォルト値としてパラメータ内に表示する場合は、テンプレート・レポート内にパラメータを作成し、次の値をパラメータのプロパティシートに入力します。

名前: CG\$P_SYSDATE
タイプ: Date
幅: 20

パラメータ・フォームを使用してレポートに対するアクセスを制限する場合は、テンプレート・レポート内にパラメータを作成し、次の値をパラメータのプロパティシートに入力します。

名前: CG\$P_ACCESS
タイプ: Character
幅: 10
初期値: *****

データベース上の表からパラメータのデフォルト値を検索する場合は、次のようになります。

1. 次のパラメータを作成します。

名前: CG\$P_SEQUENCE
タイプ: Number
幅: 20

2. データベース上の値を含む各パラメータ・フィールドに対してパラメータを作成します。

名前: CG\$P_PARAMETER1
タイプ: Character
幅: 40
名前: CG\$P_PARAMETER2
タイプ: Character
幅: 40
名前: CG\$P_PARAMETER3
タイプ: Character
幅: 40
名前: CG\$P_PARAMETER4
タイプ: Character
幅: 40

3. CG\$GET_PARAMETER_VALUES PL/SQLプロシージャを使用して、パラメータに順序番号を渡します。

テンプレート・レポート内でのレポート・トリガーの実装

第15章では、列取扱細目に対して列テキストを作成することで、生成されるレポート内にレポート・トリガーを実装する方法について説明しました。トリガーは、それらをテンプレート・レポートに追加することでも実装できます。

テンプレート・レポートにレポート・トリガーを実装することで、すべてのレポートモジュールに共通するレポート・トリガーを組み込みめます。

次の手順に従って、テンプレート・レポートにレポート・トリガーを追加します。

1. モジュール・データ・ダイアグラマから「ツール→Oracle Reports Designer」を選択します。
2. パラメータ・フォームを含むテンプレート・レポートをオープンします。
3. 「レポート・トリガー」ノードを展開します。
4. 作成したいグローバル・トリガーを呼び出します。

注意: レポートの生成では、「Before Form」と「After Form」トリガーがもっとも役に立つトリガーです。参考として、デフォルトのテンプレート・レポートによって提供されているレポート・トリガーをサンプルとして使用してください。

5. テンプレート・レポートを保存しクローズします。
6. 必要であればレポート・モジュールにテンプレート・レポートを適用し、その後レポート・モジュール定義を生成します。

レポート・トリガーの例

当日の日付をデフォルト値としてパラメータ内に表示する場合は、次のようになります。

1. テンプレート内で「Before Form」グローバル・レポート・トリガーに対して次の構文を追加します。

```
function BeforePForm return boolean is
begin
  -- initialize the SYSDATE parameter
  -- to current date
  CGPF$SET_SYSDATE_PARAM;
  return (TRUE);
end;
```

2. SYSDATEパラメータを当日の日付に初期化するCGPF\$SET_SYSDATE_PARAM PL/SQLプロシージャを作成します。または、このコードをレポート・トリガーに追加できます。

パラメータ・フォームを使用してレポートに対するアクセスを制限する場合は、テンプレート内で「AfterForm」グローバル・レポート・トリガーに対して次の構文を追加します。

```
function AfterPForm return boolean is
begin
  -- Check access rights of user
  IF (CG$CHECK_ACCESS_RIGHTS = FALSE)
  THEN
    return (FALSE);
  END IF;

  return (TRUE);
end;
```

データベース上の表からパラメータのデフォルト値を検索する場合は、次のようにします。

1. テンプレート内で「Before Form」グローバル・レポート・トリガーに対して次の構文を追加します。

```
function BeforeReport return boolean is
begin
  -- Get parameter values from database
  CG$GET_PARAMETER_VALUES;

  return (TRUE);
end;
```

2. CG\$GET_PARAMETER_VALUES PL/SQLプロシージャを使用して、パラメータに順序番号を渡します。

テンプレート レポートへの PL/SQLの追加

第15章では、列取扱細目に対して列テキストを作成することで、生成されるレポート内にPL/SQLコードを実装する方法について説明しました。

列取扱細目にPL/SQLコードを追加することで、特定のフィールドに対するPL/SQLプロシージャがサポートされます。

PL/SQLプロシージャのこのサポートは、テンプレート・レポートにPL/SQLプログラム単位を追加するか、もしくはPL/SQLライブラリをアタッチすることで、PL/SQLに含まれる可能性があるすべてのオブジェクトに対して拡張できます。テンプレート・レポートにPL/SQLライブラリをアタッチした場合、そのテンプレートを使用するすべてのモジュールがそのライブラリを利用できます。

テンプレート・レポートにPL/SQLプロシージャを追加するかPL/SQLライブラリをアタッチする場合は、次のようにします。

1. モジュール・データ・ダイアグラマから「ツール→Oracle Reports Designer」を選択します。
2. パラメータ・フォームを含むテンプレート・レポートをオープンします。
3. 「プログラム単位」ノードを展開してPL/SQLプロシージャを作成するか、「PL/SQLライブラリ」ノードを展開してライブラリをアタッチします。

注意: Reportsジェネレータが持っているデフォルトのテンプレート・レポートには、サンプルPL/SQLプロシージャとライブラリが含まれています。これらを参考にしてください。

4. テンプレート・レポートを保存しクローズします。
5. 必要であればレポート・モジュールにテンプレート・レポートをアタッチし、その後レポート・モジュール定義を生成します。

PL/SQLプロシージャとライブラリの例 当日の日付をデフォルト値としてパラメータ内に表示する場合は、次の構文をテンプレート内のプログラム単位に追加します。または、このプロシージャをライブラリとしてテンプレート・レポートにアタッチします。

```
PROCEDURE CGPF$SET_SYSDATE_PARAM IS
-- Initialize the SYSDATE parameter to
current date
BEGIN
    -- set parameter
    :CG$P_SYSDATE := SYSDATE;
    -- validate parameter
    IF (:CG$P_SYSDATE IS NULL)
    THEN
        srw.message(1000, 'Unable to set Parameter
CG$P_SYSDATE');
    END IF;
END;
```

パラメータ・フォームを使用してレポートに対するアクセスを制限し、個々の妥当性検査を実行し、該当するエラー・メッセージとエラー番号を与える場合は、次の構文をテンプレート内のプログラム単位に追加します。または、このプロシージャをライブラリとしてテンプレート・レポートにアタッチします。

```
FUNCTION CG$CHECK_ACCESS_RIGHTS RETURN BOOLEAN IS
BEGIN
    -- Test the parameter against the password
    IF (:P_ACCESS != 'password')
    THEN
        srw.message(1000, 'Invalid Password. ');
        return (FALSE);
    END IF;

    return (TRUE);
END;
```

データベース上の表からパラメータのデフォルト値を検索する場合は、次のようにします。

1. 次のPL/SQLプロシージャを作成します。

```
PROCEDURE
CG$GET_PARAMETER_VALUES IS
BEGIN
  -- Retrieves parameter values from
  -- database table using the given
  -- sequence number
  SELECT      tmp_value INTO :CG$P_PARAMETER1
  FROM        tmp_report_parameters
  WHERE       tmp_name = 'PARAM1'
  AND        tmp_seq = :CG$P_SEQUENCE;

  SELECT      tmp_value INTO :CG$P_PARAMETER2
  FROM        tmp_report_parameters
  WHERE       tmp_name = 'PARAM2'
  AND        tmp_seq = :CG$P_SEQUENCE;

  SELECT      tmp_value INTO :CG$P_PARAMETER3
  FROM        tmp_report_parameters
  WHERE       tmp_name = 'PARAM3'
  AND        tmp_seq = :CG$P_SEQUENCE;

  SELECT      tmp_value INTO :CG$P_PARAMETER4
  FROM        tmp_report_parameters
  WHERE       tmp_name = 'PARAM4'
  AND        tmp_seq = :CG$P_SEQUENCE;
END;
```

2. パラメータ・フォーム内の4つのパラメータに値を渡すCG\$P_SEQUENCEパラメータに順序番号を渡します。

テンプレート・レポートへの
フォーミュラ列の追加

生成されるレポートにフォーミュラ列を組み込んで、データベースから戻された値を表示できます。たとえば、レポートのヘッダー、トレーラ、余白に表示する値をデータベースから戻すことができます。

次の手順に従って、テンプレート・レポートにフォーミュラ列を追加します。

1. モジュール・データ・ダイアグラマから「ツール→Oracle Reports Designer」を選択します。
2. テンプレート・レポートをオープンします。
3. データ・モデル・エディタを呼び出します
4. 適切なタイプのフォーミュラ列を作成します。
5. フォーミュラ列のためのプログラム単位を作成します。
注意: デフォルトのテンプレート・レポート内のCG\$C_USER列を参考にしてください。
6. テンプレート・レポートを保存しクローズします。
7. 必要であればレポート・モジュールにテンプレート・レポートをアタッチし、その後レポート モジュール定義を生成します。

テンプレートレポートを使用したレイアウトのカスタマイズ

テンプレートレポートを修正して、生成されるレポートとパラメータ・フォームの外観を変更できます。この節では、次の処理の実行方法について説明します。

- パラメータ・フォームのカスタマイズ
- パラメータ・フォームの標準パラメータ・フィールドとプロンプトの作成
- レポートまたはパラメータ・フォーム内でのジェネレータ・オブジェクトの作成
- レポートまたはパラメータ・フォーム内でのユーザー・オブジェクトの作成
- レポートまたはパラメータ・フォームのページのカスタマイズ
- フォントまたは可位属性の適用
- 文字単位を定義するフォントの変更

パラメータ フォームのカスタマイズ

「TEMPLATE」作業環境グループの初期値が変更されない場合、ジェネレータは、デフォルトのテンプレート・レポートであるCGBMPF.RDFに指定されているパラメータ・フォームを使用します。

表取扱細目を表示するために追加したパラメータに対してジェネレータ・オブジェクトを作成することで、パラメータ・フォーム内のパラメータのサイズと位置を調整し、プロンプトの可視属性を変更できます（列取扱細目は、「レポート・モジュールの取扱細目」ダイアログ・ボックスの「列細目」タブの「表示」チェックボックスがチェックされている場合に表示されます）。

1. モジュール・データ・ダイアグラマから「ツール→Oracle Reports Designer」を選択します。
2. パラメータ・フォームを含むテンプレート・レポートをオープンします。
3. パラメータ・フォーム・エディタを呼び出します。
4. レポート・モジュールに追加した各パラメータのフィールドを作成し、CG\$PF<number>キーワードを使用してフィールドに名前をつけます。CG\$PF<number>内の番号をパラメータの順序番号に置き換えます（パラメータのシーケンス番号は、「モジュールの編集」ダイアログ・ボックスの「パラメータ」タブ内に定義されます）。

注意: レポート・モジュール内の表示パラメータの数が不明な場合、またはパラメータの正確な位置を指定したくない場合は、CG\$PF<number>フィールドに加えて、またはそれに代えて標準フィールドを指定できます。ジェネレータは、表示パラメータに対してCG\$PF<number>が定義されていない場合は、常に標準フィールドを使用します。

5. パラメータ・フォーム・エディタを使用して、各パラメータのフォントとその他のフォーマット属性を指定します。
6. 作成した各CG\$PF<number>フィールドのフィールド・プロンプトのボイラプレート・テキスト・オブジェクトを作成します。

CG\$PP<number>キーワードを使用して、オブジェクトに名前を付けます。CG\$PP<number>内の番号をパラメータの順序番号に置き換えます。

注意: レポート・モジュール内の表示パラメータの数が不明な場合、またはパラメータの正確な位置を指定したくない場合は、CG\$PP<number>フィールドに加えて、またはそれに代えて標準フィールドを指定できます。ジェネレータは、表示パラメータに対してCG\$PP<number>が定義されていない場合は、常に標準フィールドを使用します。

7. パラメータ・フォーム・エディタを使用して、各プロンプトのフォントとその他のフォーマット属性を指定します。
8. テンプレート・レポートを保存しクローズします。
9. 必要であれば、レポート・モジュールにテンプレートレポートを適用し、その後レポート・モジュール定義を生成します。

ジェネレータは、CG\$PF<number>フィールドを使用して、レポート・モジュールの表示列取扱細目に指定されているパラメータを作成します。ジェネレータは、CG\$PP<number>ボイラープレート・オブジェクトを使用して、プロンプトを作成します。

次に、標準パラメータとプロンプトが作成されていれば、ジェネレータはこれらのオブジェクトを使用して、CG\$PF<number>フィールドとCG\$PP<number>オブジェクトが作成されなかった表示列取扱細目のすべてのパラメータのフィールドとプロンプトを作成します。

標準パラメータとプロンプトが指定されなかった場合、ジェネレータはパラメータのフィールドを作成し、パラメータの順序番号に従ってそれらをパラメータ・フォーム上に表示します。これらのパラメータのデフォルト間隔と位置を調整する場合は、「LAYOUT-PARAMETER」作業環境グループを使用します。これらのパラメータのフォントと可視属性を変更する場合は、レイアウト・モデルのテンプレート・レポートにCG\$PARAMETERオブジェクトを追加します。



詳細情報: 「LAYOUT - PARAMETER」作業環境グループについては、Oracle Reportsジェネレータのオンライン・マニュアルを参照してください。

レポートの生成後、ジェネレータは、未使用のパラメータ、CG\$PFNフィールド、CG\$PPNボイラープレートオブジェクトをテンプレート・レポートから削除します。

標準パラメータ・フィールド とプロンプトの作成

表示列取扱細目に対して追加されたすべてのパラメータに対して `CG$PF<number>` フィールドと `CG$PP<number>` ボイラープレート・オブジェクトを追加することで、パラメータフォーム内のフィールドの位置やフォーマットをカスタマイズできます。

ただし、レポート・モジュール内の表示パラメータの数が不明な場合、またはパラメータの正確な位置を指定したくない場合は、`CG$PF<number>` フィールドと `CG$PP<number>` ボイラープレート・オブジェクトに加えて、またはそれらに代えて標準フィールドとボイラープレート・オブジェクトを指定できます。

ジェネレータは、表示パラメータに対して `CG$PF<number>` フィールドや `CG$PP<number>` ボイラープレート・オブジェクトが定義されていない場合は、常に標準フィールドと標準ボイラープレート・オブジェクトを使用します。

`CG$PF<number>` フィールドまたは標準パラメータが指定されなかった場合、ジェネレータはパラメータのフィールドを作成し、パラメータの順序番号に従って、それらをパラメータ・フォーム上に表示します。

これらのパラメータのデフォルト間隔と位置を調整する場合は、「LAYOUT - PARAMETER」作業環境グループを使用します。これらのパラメータのフォントと可視属性を変更する場合は、レイアウトモデルテンプレートレポートに `CG$PARAMETER` オブジェクトを追加します。



詳細情報: 「LAYOUT - PARAMETER」作業環境グループについては、Oracle Reportsジェネレータのオンラインマニュアルを参照してください。

次の手順に従って、標準パラメータ・フィールドとプロンプトを作成します。

1. モジュール・データ・ダイアグラマから「ツール→Oracle Reports Designer」を選択します。
2. パラメータ・フォームを含むテンプレート・レポートをオープンします。
3. パラメータ・フォーム・エディタを呼び出します。
4. パラメータ・フォーム内にフィールドを作成することで、標準フィールドを作成します。`CG$PFN` キーワードを使用してフィールドに名前を付けます。フィールドが、生成されるレポート内に表示される情報を処理できるだけの大きさを持っていることを確認します。
5. パラメータ・フォーム・エディタを使用して、パラメータのフォントとその他のフォーマット属性を指定します。ジェネレータは、該当する `CG$PF<number>` フィールドがないすべてのパラメータ・フィールドに対して、ユーザーが定義したフォーマットを適用します。

6. ボイラープレート・テキスト・オブジェクトを作成することで、標準プロンプトを作成します。CG\$PPNキーワードを使用して、オブジェクトに名前を付けます。オブジェクトが、生成されるレポート内に表示されるプロンプトを処理できるだけの大きさを持っていることを確認します。
7. CG\$PARAMETERジェネレータ・オブジェクトを作成することで、フォントとその他のフォーマット属性を指定します。
8. テンプレート・レポートを保存しクローズします。
9. 必要であればレポート・モジュールにテンプレート・レポートを適用し、その後レポート・モジュール定義を生成します。

CG\$PF<number>フィールドが指定されている場合、ジェネレータはこれらのフィールドを最初に使用して、レポート・モジュールの表示列取扱細目に指定されているパラメータを作成します。ジェネレータは、CG\$PP<number>ボイラープレート・オブジェクトを使用して、プロンプトを作成します。

次に、ジェネレータは標準パラメータとプロンプトを使用して、CG\$PF<number>フィールドとCG\$PP<number>オブジェクトが作成されなかった表示列取扱細目内のすべてのパラメータのフィールドとプロンプトを作成します。

レポートの生成後、ジェネレータは、CG\$PFNフィールドとCG\$PPNボイラープレート・オブジェクトをテンプレート・レポートから削除します。

ジェネレータ・オブジェクトの作成

ジェネレータ・オブジェクトは、可変名を持つテンプレート内のオブジェクトです。ジェネレータは、ジェネレータ・オブジェクトを使用して、生成されるオブジェクトの属性の設定、生成されるレポートとパラメータ・フォーム内での特殊オブジェクトの作成、他の生成オブジェクトに対するオブジェクトの割り当てを行います。

テンプレート・レポートにジェネレータ・オブジェクトを追加することで、次の処理を実行できます。

- アプリケーション・システム名、表示タイトル、バージョン番号、社名などのアプリケーション・システムについての詳細の追加
- モジュールの上部タイトル、下部タイトル、プロジェクトコード、タスク・コード、名前、所有者、目的、短縮名などのモジュールについての詳細の追加
- パラメータの入力フィールドやプロンプトの位置の設定
- 現在のOracleユーザーの表示

すべてのジェネレータ・オブジェクトの名前は接頭語CG\$で始まり、オブジェクトのタイプを識別する2文字または3文字のキーワードが続きます。たとえば、アプリケーション・システム名に割り当てられる接頭語はCG\$ANです。

名前の残りの部分は、個々のオブジェクトを識別します。たとえば、ヘッダーとトレーラ・ページにモジュール名を追加する場合は、レイアウト・モデル・テンプレート・レポートのヘッダー部分にCG\$MN_TITLEと呼ばれるジェネレータ・

オブジェクトを、トレーラ部分にCG\$MN_TRAILERと呼ばれるジェネレータ・オブジェクトを作成します。

テンプレート・レポート内にジェネレータ・オブジェクトのキーワードを作成することで、生成されるレポートとパラメータ・フォーム内にジェネレータ・オブジェクトを作成できます。

次の手順に従って、ジェネレータ・オブジェクトを作成します。

1. モジュール・データ・ダイアグラマから「ツール→Oracle Reports Designer」を選択します。
2. テンプレート・レポートをオープンします。ジェネレータ・オブジェクトは、レイアウト・モデル、パラメータ・フォーム、ドリルダウン詳細レポート・レポートの中に作成できます。
3. 適切なエディタを呼び出します。ジェネレータ・オブジェクトは、テンプレート・レポートのレイアウト・モデル、データ・モデル、またはパラメータ・フォーム モデルの中に作成できます。

注意: ジェネレータ・オブジェクトは、レポートのヘッダー、トレーラ、余白の中に作成できますが、レイアウト・モデルのレポート本体の中には作成しないでください。ジェネレータは、レポート・モジュールを生成する前に、これらのオブジェクトをレイアウト・モデルから取り除きます。

4. テキスト・ボイラープレート・オブジェクトを作成し、ジェネレータ・オブジェクトのキーワードを入力します。パラメータ入力フィールドの位置を指定している場合、または生成されるレポート内に現在のユーザーを表示するためのオブジェクトを作成している場合は、テキスト・ボイラープレートの代わりにフィールドを作成します。テキスト・ボイラープレートまたはフィールドを作成するときは、次の点に注意してください。
 - 完全な接頭語を使用する。さもないとジェネレータ・オブジェクトはユーザー・オブジェクトとして扱われ、レポート内に直接コピーされる
 - ボイラープレート・テキスト・オブジェクトまたはフィールドのサイズは、生成されるレポート内に表示される情報を処理できるだけの大きさをなければならない
5. テンプレート・レポートを保存しクローズします。
6. 必要であればレポート・モジュールにテンプレート・レポートを適用し、その後レポート・モジュール定義を生成します。

生成されるレポート内にジェネレータ・オブジェクトが作成されます。

注意: ジェネレータは、有効な値が見つからないすべてのジェネレータ・オブジェクトを削除します。

ユーザー オブジェクトの作成 ユーザー・オブジェクトは、ジェネレータが生成されるレポート内に修正なしでコピーできるオブジェクトです。特定のユーザー・オブジェクトが生成されるレポート内に組み込まれる条件は、ユーザー・オブジェクトのタイプと、ジェネレータによるユーザー・オブジェクトの使用法によって決まります。

注意: ジェネレータは、名前がCG\$で始まらないすべてのオブジェクトをユーザー・オブジェクトと見なします。

テンプレート・レポート内にオブジェクトを作成するかそれを変更することで、生成されるレポートとパラメータ・フォームにユーザー・オブジェクトを追加できます。

次の手順に従って、ユーザー・オブジェクトを作成するか修正します。

1. モジュール・データ・ダイアグラマから「ツール→Oracle Reports Designer」を選択します。
2. テンプレート・レポートをオープンします。ユーザー・オブジェクトは、レイアウト・モデル、パラメータ・フォーム、ドリルダウン詳細テンプレート・レポートの中に作成できます。
3. 適切なエディタを呼び出します。ユーザー・オブジェクトは、テンプレート・レポートのレイアウト・モデル、データ・モデル、またはパラメータ・フォームの中に作成するか、修正できます。

注意: ユーザー・オブジェクトは、レポートのヘッダー、トレーラ、余白の中に作成できますが、レイアウト・モデルのレポート本体の中には作成しないでください。ジェネレータは、レポート・モジュールを生成する前に、これらのオブジェクトをレイアウト・モデルから取り除きます。

4. 修正するユーザー・オブジェクトを選択するか、新しいオブジェクトを作成します。
5. テンプレート・レポートを保存しクローズします。
6. 必要であればレポート・モジュールにテンプレート・レポートを適用し、その後レポート・モジュール定義を生成します。

ユーザー・オブジェクトが、テンプレート・レポートから生成されるレポートかパラメータ・フォームにコピーされます。

レポートまたはパラメータ・フォームのページのカスタマイズ

レイアウト・モデルとドリルダウン・テンプレート・レポートを修正することで、レポートのヘッダー・ページとトレーラ・ページ、論理ページ・サイズと物理ページ・サイズを定義できます。また、パラメータ・フォームのテンプレート・レポートを修正することで、パラメータ・フォームのページサイズとページ数を定義できます。

次の手順に従って、レポートまたはパラメータ・フォームのページ定義を変更します。

1. モジュール・データ・ダイアグラマから「ツール→Oracle Reports Designer」を選択します。
2. レイアウト・モデル、ドリルダウン詳細、またはパラメータ・フォームのテンプレート・レポートをオープンします。
3. レイアウト・モデルまたはパラメータ・フォームのエディタを呼び出します。
4. レポートまたはパラメータ・フォームのページ属性を変更し、その後テンプレート・レポートを保存しクローズします。
5. 必要であればレポート・モジュールにテンプレート・レポートを適用し、その後レポート・モジュール定義を生成します。

フォントまたは可視属性の適用

レイアウト・モデルのテンプレート・レポートを使用して、次のオブジェクトのフォントと可視属性を定義できます。

- フィールドの前後に表示されるプロンプト（フィールド前プロンプトとフィールド後プロンプト）
- フィールド
- フレーム・タイトル
- パラメータ・プロンプト
- 項目グループを囲むフレーム

注意: 生成されるレポートとパラメータ・フォーム内の文字単位を定義するために使用するフォントも変更できます。

次の手順に従って、フォントまたは可視属性を定義します。

1. モジュール・データ・ダイアグラマから「ツール→Oracle Reports Designer」を選択します。
2. デフォルトのレイアウト・モデル・テンプレートであるCGBMPT.RDFをオープンします。CGBMLS.RDF、CGCHPT.RDF、CGCHLS.RDFなどのレイアウト・モデルのテンプレート・レポートや独自のレイアウト・モデル用テンプレート・レポートを使用している場合は、そのレポートをオープンします。

注意: デフォルトのドリルダウン詳細レポートをオープンするか、独自のドリルダウン詳細レポートを作成することで、ドリルダウン詳細レポートのフォントと可視属性も定義できます。

3. レイアウト・モデルのテンプレート・レポートまたはドリルダウン詳細テンプレート・レポートのレイアウト・モデル内のレポート本体を開きます。

4. カスタマイズしたいレイアウト・オブジェクトの各タイプに対してボイラープレート・テキスト・オブジェクトを作成し、名前を付けます。



詳細情報: テンプレート・オブジェクトの全リストについては、Oracle Reportsジェネレータのオンライン マニュアルを参照してください。

5. レイアウト・オブジェクトに適用するフォントと可視属性を定義します。次に示すすべてのフォーマッティング・スタイルを適用できます。

- フォント
- フォント・サイズ
- フォントの重み
- フォントの色
- 前景の色
- 背景の色
- 境界線の幅
- 境界線の色

たとえば、プロンプトに対して10ポイントのGaramondフォントを、フレーム・タイトルに対して12ポイントの太字のArialフォントを指定できます。

6. テンプレート・レポートを保存しクローズします。
7. 必要であればレポート・モジュールにテンプレート・レポートを適用し、その後レポート・モジュール定義を生成します。

生成されるレポートのレイアウトが作成されるときに、ユーザーが指定したフォントと属性が適用されます。

文字単位を定義するフォントの環境設定の値は、文字単位で計測されます。

の変更

AVGWID	MGRFIG
DGRFIG	PARSPC
FIEPRG	PRFIEG
GRFIND	TWPRLN
GRPTGP	VFIESP

デフォルトでは、ジェネレータはこれらの作業環境の値の計測単位としてシステム・フォントの高さと幅を使用します。システム・フォントを文字単位として使用したくない場合は、別のフォントとフォント設定を使用できます。

注意: キャラクタ・モード・レポートでは、別のフォントを使用できません。すべてのキャラクタ・モード・レポートは、文字単位で計測されます。

次の手順に従って、文字単位を定義するフォントを変更します。

1. モジュール・データ・ダイアグラマから「ツール→Oracle Reports Designer」を選択します。
2. デフォルトのレイアウト・モデル・テンプレートであるCGBMPT.RDFをオープンします。CGBMLS.RDFなどのレイアウト・モデルのテンプレート・レポートや独自のレイアウト・モデル用テンプレート・レポートを使用している場合は、そのレポートをオープンします。
注意: デフォルトのドリルダウン詳細レポートをオープンするか、独自のドリルダウン詳細レポートを作成することで、ドリルダウン詳細レポートの文字単位も定義できます。
3. レイアウト・モデルのテンプレート・レポートまたはドリルダウン詳細テンプレート・レポートのレイアウト・モデル内のレポート本体を開きます。
注意: レイアウト・モデルのテンプレート・レポートまたはドリルダウン詳細テンプレート・レポートのレイアウト・モデル内のレポート本体に文字単位を定義する必要があります。レポート本体にオブジェクトを定義することで、生成されるレポートにそれらがコピーされることを防ぎます。
4. CG\$SIZINGと呼ばれるボイラープレート・テキスト・オブジェクトを作成します。
5. フォント、フォントサイズ、フォントの重みを定義します。
6. テンプレート・レポートを保存しクローズします。
7. 必要であればレポート・モジュールにテンプレート・レポートを適用し、その後レポート・モジュール定義を生成します。

ユーザーがレポート・モジュール定義を生成すると、ジェネレータは常にCG\$SIZINGによって定義されたフォントの幅と高さを使用して文字単位のサイズを判断します。

第17章

レポートのリバース・エンジニアリング

この章では、レポートのリバース・エンジニアリングプロセスについて説明します。この章では、特に次の情報について説明します。

- リバース・エンジニアリングの紹介
- レポートをリバース・エンジニアリングするための手順
- Oracle Reports v2.5とSQL*ReportWriterレポートのリバース・エンジニアリングプロセス
- SQL*Plus、SQL*Report (RPT)、Pro*Cレポートのリバース・エンジニアリングプロセス
- 新しいモジュールと既存のモジュールの整理統合

リバース・エンジニアリングの紹介

リバース・エンジニアリングとは何か? リバース・エンジニアリングとは、既存のレポートを使用して、対応するリポジトリ モジュール定義の作成または更新を実行するプロセスです。

具体的には、リバース・エンジニアリング・ユーティリティは次の情報をレポートから抽出し、それをリポジトリに記録します。

- モジュール定義と記述
- 表および列の取扱サマリー
- 基本表取扱と基本列取扱

ユーティリティは、リバース・エンジニアリングプロセス中にジェネレータが判断した内容を示すログ・ファイルも作成します。

なぜリバース・エンジニアリングを使用するのか? リバース・エンジニアリングを行う理由はいくつかあります。

- 既存のシステムをリポジトリ内に文書化するため
- 影響分析を可能にするため
- 既に存在する システムをメンテナンスするため
- リポジトリ定義を現実の状態と一致させるため

サポートされるレポートの種類 次の種類のレポートをリバース・エンジニアリングできます。

- Oracle Reports バージョン2.5および2.0
- SQL*ReportWriter V1.1
- SQL*Plus
- SQL*Report (RPT)
- Pro*C

リバース・エンジニアリングを行う前に レポートをリバース・エンジニアリングするためには、次のものが必要になります。

- レポートに対するアクセス権
- リポジトリ内のアプリケーション・システムに対する書き込みアクセス権
- レポートによって参照されるすべてのデータベース・オブジェクトのリポジトリ定義

注意: リバース・エンジニアリング・ユーティリティは、レポート定義からデータベース設計を推論しません。リポジトリによって使用されているオブジェクトの定義がリポジトリ内に存在しない場合は、Designer/2000の「データベースのリバース・エンジニアリング」ユーティリティを使用してデータベースからオブジェクト定義を取得し、それらをリポジトリに記録してください。

レポートのリバース・エンジニアリング方法

次の手順に従って、レポートのリバース・エンジニアリングを実行します。

1. モジュール・データ・ダイアグラマまたはリポジトリ・オブジェクト・ナビゲータから「ユーティリティ→リバース・エンジニア→レポート」を選択します。

「レポートのリバース・エンジニア」ダイアログ・ボックスが表示されます。



2. 「製品のフレーバー」ドロップダウン・リストから、リバースするレポートの種類を選択します。
3. 「位置」フィールドを使用して、リバースされるレポートがファイル内にあるのかデータベース内にあるのかを指定します。
4. 既存のモジュールを更新する場合は、「接続先モジュール」フィールドからモジュールの短縮名を選択します。

注意: 既存のモジュールの更新については、「新しいモジュールと既存のモジュールの整理統合」の項を参照してください。

新しいモジュールを作成したり、複数のレポートをリバースする場合は、接続先モジュールを選択しないでください。ジェネレータは、レポート名を接続先モジュールとして使用します。

5. レポートをリバースするために使用する作業環境のタイプを選択します（作業環境のタイプについては、第16章を参照してください）。
6. リバースするレポートの名前を指定します。

レポートの位置としてファイルを選択した場合は、「ファイル」フィールドを使用してレポートを選択します。レポートの位置としてデータベースを選択した場合は、「データベース」タブを使用してレポートを選択します。

一覧表示されるレポートはフィルター処理できます。フィルター・フィールドに検索規準を入力し、フィルターボタンを選択します。たとえばSALで始まるすべてのレポートを一覧表示する場合はSAL%と入力します。

7. リポジトリをリバースされた情報で更新する場合は、「リポジトリの更新」チェックボックスを選択します。

ジェネレータがリポジトリに記録する情報は、リバースされるレポートの種類によって決まります。詳細については、この章で後ほど説明する「Oracle ReportsとSQL*ReportWriterのレポート」と「SQL*Plus、SQL*Report (RPT)、およびPro*Cのレポート」を参照してください。

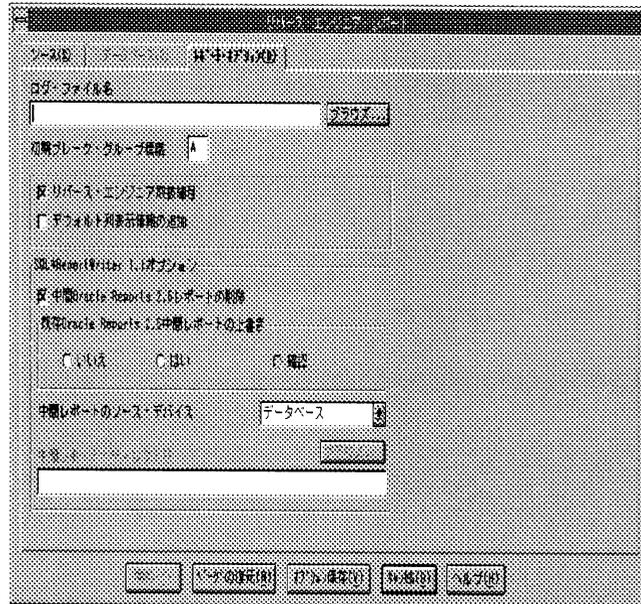
ジェネレータによるリポジトリの更新を抑止する場合は、チェックボックスの選択を解除します。ジェネレータはリポジトリを更新しませんが、「Oracle Reports ジェネレータ」ダイアログ・ボックスを使用して、リポジトリで更新されなかったプロパティを表示します。ログ・ファイルが指定されている場合は、ログ・ファイルにも変更が書き込まれます。

8. データベースに保存されているレポートをリバースする場合は、「データベース」タブの「所有者」フィールドにレポートの所有者の名前を指定します。
9. レポートをリバースするために使用するレポート・オプションをカスタマイズする場合は、「レポート・オプション」タブを選択します。ログ・ファイルの作成、取扱細目のリバース、SQL*ReportWriter 1.1のレポートのために生成される中間Oracle Reportsv 2.5の保存などのオプションを設定できます。詳細については、次の「リバース・エンジニアリング・オプションの変更」を参照してください。
10. 「OK」をクリックして、リバース・エンジニアリング・プロセスを開始します。

ジェネレータは「Oracle Reportsジェネレータ」ダイアログ・ボックスを表示し、レポートをリバースします。「リポジトリの更新」チェックボックスが選択されている場合、ジェネレータは、情報をリポジトリに記録します。

リバース・エンジニアリング オプションの変更

レポートをリバースするために使用するオプションを変更する場合は、「レポートのリバース・エンジニアリング」ダイアログ・ボックスの「レポート・オプション」タブを表示します。



ログ・ファイル

リバース・エンジニアリング・プロセスのログ・ファイルを作成する場合は、ログ・ファイル名を入力します。リバース・エンジニアリング中にエラーや警告が発生した場合、ジェネレータは、現在の作業ディレクトリにログ・ファイルを作成します。エラーも警告も発生しなかった場合、ログ・ファイルは作成されません。

ブレイク・グループ

ブレイク・グループをリバースする場合は、レポート内でブレイク・グループを示すために使用する先頭文字を指定します。

データ取扱細目

リバースするモジュールの表取扱細目と列取扱細目の情報を作成する場合は、「取扱細目のリバース・エンジニア」チェックボックスを選択します。表と列の細目には、複数ページ、サマリー列、ブレイク・グループなどのオブジェクトが含まれます。

このオプションの選択を解除すると、ジェネレータは取扱サマリーだけをリバースします。

リバースするモジュールの列表示の情報を作成する場合は、「デフォルト列表示情報の追加」チェックボックスを選択します。このオプションは、レポートを初めてリバースするときだけ選択してください。

注意：「取扱細目のリバース・エンジニア」チェックボックスを選択しなかった場合、「デフォルト列表示情報の追加」チェックボックスは薄く表示されます。

SQL*ReportWriterのレポート SQL*ReportWriterのレポートをリバースしたときに生成される中間Oracle Reports v2.5を保存する場合は、「Drop Intermediate Oracle Reports 2.5 Reports」チェックボックスの選択を解除します。

これらのレポートの保存を選択した場合は、同名の中間レポートが存在する場合、ジェネレータにそのレポートを上書きしてほしいかどうかを指定します。既存の中間レポートを上書きする場合は「はい」を選択します。同名の中間レポートが存在する場合は常にリバース・エンジニアリング・プロセスを中止する場合は「いいえ」を選択します。既存の中間ファイルの上書きを実行するかどうかの確認を求めて欲しい場合は、「確認」を選択します。

中間ファイルをレポートに保存するかデータベースに保存するかを指定します。「ファイル」オプションを選択した場合は、中間レポートを保存するディレクトリを指定します。ディレクトリは、「ブラウズ」ボタンを使用して検索できます。ディレクトリを指定しなかった場合、中間レポートは現在の作業ディレクトリの中に作成されます。

Oracle ReportsとSQL*ReportWriterのレポート

リバース・エンジニアリング・プロセス中、ジェネレータはOracle ReportsとSQL*ReportWriterのレポートの次の要素や情報を作成します。

- リバースしている各レポートのための新しいモジュール定義の作成。モジュール定義がすでに存在する場合は、新しいモジュール定義と古いモジュール定義を整理統合する
- レポート内のグループ構造を使用して、問合せの階層と表取扱細目の順序を定義する
- 各問合せの中の選択項目を識別し、選択項目のための表取扱と列取扱を作成する。これらの列取扱の表示取扱を自動選択する。また、結合条件や副問合せを使用して、モジュールに表示されない表と列の取扱を作成する
- 次のデータ・モデル・オブジェクトを抽出する
 - 列取扱サマリー
 - パラメータ
 - ブレーク・グループ

注意: 列取扱サマリーのリバース・エンジニアリングは、リバース・エンジニアリング・プロセスを開始する前に「デフォルト列表示情報の追加」チェックボックスを選択した場合だけ実行されます。ブレーク・グループのリバース・エンジニアリングは、リバース・エンジニアリング・プロセスを開始する前に「初期グループ・ブレーク標識」ボックスにブレークグループインジケータを指定した場合だけ実行されます。

- リバース・エンジニアリング・プロセスを開始する際に「リポジトリの更新」チェックボックスが選択された場合は、レポートのリバース・エンジニアリング情報をリポジトリに記録する
- SQL*ReportWriterのリバース・エンジニアリング・プロセスを開始する前に「中間Oracle Reports 2.5レポートの削除」チェックボックスが選択されなかった場合は、Oracle Reports 中間レポートを保存する

注意: Reports ジェネレータは、データ・モデルとレイアウト・モデル内のすべてのオブジェクトをリバースするわけではありません。

リバース・エンジニアリング・プロセス中に記録される情報

SQL*ReportWriterまたはOracle Reportsのレポートをリバースすると、(必要に応じて) 次の要素が作成され、これらの要素のプロパティがリポジトリに記録されます。

- レポート・モジュール定義

- 表取扱細目
- 列取扱細目
- パラメータ

これらのプロパティは、リポジトリ・オブジェクト・ナビゲータを使用して参照できます。

新しいモジュール定義

ジェネレータは、リバース・エンジニアリング・プロセスによって新しいモジュール定義が作成された場合は常に次のプロパティをリポジトリに記録します。

プロパティ	割り当て値またはソース
短縮名	元のレポート名
名前	元のレポート名
目的	「Oracle Reportsレポートジェネレータによってリバース・エンジニアリングされました」
言語	Oracle ReportsまたはSQL*ReportWriter
タイプ	REPORT
フォーマット	マスター・ディテール
メモ	「Oracleレポートモジュールに対して保持済み」

新しい表取扱細目

ジェネレータは、リバース・エンジニアリング・プロセスによって新しい表取扱細目が作成された場合は常に次のプロパティをリポジトリに記録します。

プロパティ	割り当て値またはソース
Seq	値は詳細テーブル使用法によって指定される
Select	値は詳細テーブル使用法によって指定される

注意: 取扱細目間の結合はリバースされません。

新しい列取扱細目

ジェネレータは、リバース・エンジニアリング・プロセスによって新しい列取扱細目が作成された場合は常に次のプロパティをリポジトリに記録します。

プロパティ	割り当て値またはソース
Seq	値はモジュールによって指定される
「表示」チェックボックス	列が選択項目かサマリー列で使用されている場合に選択される
Select	列が選択項目で使用されている場合は「Y」、サマリーなどの二次使用法がある場合は「N」
データ型	列のデータ型
幅	レポートをリバースする際に「レポートのリバース・エンジニアリング」ダイアログ・ボックスの「デフォルト列表示情報の追加」チェックボックスが選択されている場合は、Oracle Reports v2.5のデフォルト値に設定。「デフォルト列表示情報の追加」チェックボックスが選択されていない場合は、デフォルト列表示幅か最大列幅に設定
高さ	レポートをリバースする際に「レポートのリバース・エンジニアリング」ダイアログ・ボックスの「デフォルト列表示情報の追加」チェックボックスが選択されている場合は、Oracle Reports v2.5のデフォルト値。「デフォルト列表示情報の追加」チェックボックスが選択されていない場合は、デフォルト表示高に設定
プロンプト	レポートをリバースする際に「レポートのリバース・エンジニアリング」ダイアログ・ボックスの「デフォルト列表示情報の追加」チェックボックスが選択されている場合は、Oracle Reports v2.5デフォルト値。「デフォルト列表示情報の追加」チェックボックスが選択されていない場合は、デフォルト・プロンプトに設定
サマリー関数	値は列取扱によって指定され、Oracle Reports v2.5列がサマリー関数の場合だけ設定される
サマリータイプ	値は列取扱細目によって指定され、Oracle Reports v2.5列がサマリー関数であり、タイプを判別できる場合だけ設定される
折り返し	値は、列取扱細目によって指定される
ブレイク	列がブレイク・グループ内にある場合は、「レポートのリバース・エンジニア」ダイアログ・ボックスに指定されているブレイクインジケータに設定
コメント	「Oracle Reportsジェネレータにより作成されました」

新しいモジュールパラメータ

ジェネレータは、リバース・エンジニアリング・プロセスによって新しいモジュール・パラメータが作成された場合は常に次のプロパティをリポジトリに記録します。

プロパティ	割り当て値またはソース
名前	Oracle Reports v2.5パラメータ名

Seq	値はモジュールによって指定される
データ型	列のデータ型
幅	Oracle Reports v2.5デフォルト値がある場合はその値に設定。 デフォルト値がない場合は、最大長に設定
演算子	= (等号)
プロンプト	レポートをリバースする際に「レポートのリバース・エンジニア」ダイアログ・ボックスの「デフォルト列表示情報の追加」チェックボックスが選択されている場合は、Oracle Reports v2.5のデフォルト値に設定。 「デフォルト列表示情報の追加」チェックボックスが選択されていない場合は、デフォルト プロンプトに設定
説明	「Oracle Reports ジェネレータによりリバース・エンジニアリングされました」

レイアウト・モデル・オブジェクト

ジェネレータは、Oracle Reports v2.5レポートのレイアウトモデル内のオブジェクトをリバースしませんが、Oracle Reports v2.5のデフォルト情報、フレーム・タイトル、フィールド・プロンプトを次のように処理します。

Oracle Reportsのデフォルト情報

ジェネレータは、Oracle Reportsデフォルト画面で入力された情報をリバースします。

フィールド プロンプトとフレーム タイトル

ボイラープレートは、それが表示されるレポートだけに関連付けられ、列や表には関連付けられません。したがって、ジェネレータはこのテキストを表取扱細目や列取扱細目に関連付けることができず、リバースは実行されません。この結果、次のようになります。

- 列取扱細目のプロンプトは、Oracle Reportsのデフォルト・プロンプトか表定義内のデフォルト表示プロンプトに設定される
- 各基礎表取扱細目のタイトルは、テーブル定義上の通常の表示タイトルに設定される
- モジュールの下部タイトルは、空白のままである

データ・モデル・オブジェクト

ジェネレータは、Oracle Reports v2.5のデータ モデルの大半をリバースしますが、次の例外があります。

リバース・エンジニアリング・プロセスは、導出表現をリバースしません。可能であれば、ジェネレータは、導出表現内で使用されている列を反映する取扱を追加します。

ジェネレータは、リバースを実行しない導出表現やSQLサマリー列に対して警告を発行します。

結合

結合はリバースされません。ただし、モジュール内の多くの結合は、デフォルト結合生成ユーティリティを使用して再作成可能です。

レコード・レベルのSQLサマリー

レコード・レベルのSQLサマリーはリバースされません。

SQL*Plus、SQL*Report (RPT)、およびPro*Cのレポート

リバース・エンジニアリング・ユーティリティは、ソース・ファイルを解析し、レポートのSQL問合せ文を識別します。ジェネレータは、各問合せに対して次の処理を実行します。

- 表取扱および列取扱
- レポート・モジュールのためにサマリー データをリポジトリ内に作成するか更新する

ジェネレータはSQL*Plus、SQL*Report(RPT)、およびPro*Cのレポートの表取扱細目を作成しませんが、リバース・エンジニアリング中に生成されるレポート・モジュールを使用して取扱細目を作成できます。

新しいモジュール定義

ジェネレータは、リバース・エンジニアリング・プロセスによって新しいモジュール定義が作成された場合は常に次のプロパティをリポジトリに記録します。

プロパティ	割り当て値またはソース
短縮名	元のレポート名
名前	元のレポート名
目的	「Oracle Reportsジェネレータによってリバース・エンジニアリングされました」
言語	SQL*Plus、Pro*C、またはOracle Reports
タイプ	REPORT
フォーマット	マスター・ディテール
メモ	「<日付>付けて<ユーザー> Oracle Reportsジェネレータによって後処理済み」

新しいカラムとテーブルの使用法

ジェネレータは、リバース・エンジニアリング・プロセスによって新しい列取扱や表取扱が作成された場合は常に次のプロパティをリポジトリに記録します。

- 表示
- 選択 (SELECT)
- 更新 (UPDATE)
- オプション

これらのプロパティに割り当てられる値は、問合せの取扱タイプによって決まります。

新しいモジュールと既存のモジュールの整理統合

リポジトリ内にすでに存在するモジュールのリバース・エンジニアリングを指定すると、古いモジュールと新しいモジュールの整理統合を求められます。

整理統合を選択すると、ジェネレータは既存のモジュールと新しいモジュールを結合し、表取扱細目と列取扱細目を追加します。ジェネレータは、既存の設計情報の整合性を保持し、不要な表取扱細目と列取扱細目だけを削除します。表取扱細目全体が取り除かれる場合は、次の項目が削除されます。

- ユーザーが定義した導出表現
- ユーザー・テキスト
- ファイルとイメージ列
- レコードレベルのSQLサマリー

ジェネレータは、既存の表取扱細目と列取扱細目の情報を更新しません。

マニュアル・コメント用紙

Developer/2000生成ガイド, リリース1
部品番号: A37360-1

日本オラクル株式会社は、本書の品質と有用性に関する読者の皆様からのご意見やご感想をお待ちしております。皆様からのご意見は、本書の改訂の際に重要な参考資料とさせていただきます。

ご意見につきましては、章名、項目名、およびページ番号を以下にご記入の上、下記弊社担当係まで、郵送またはファックスでお願いいたします。

宛先:

〒102 東京都千代田区紀尾井町4-1
ニューオータニ ガーデンコート
日本オラクル株式会社
マニュアル係
FAX: 03-5213-6465

なお、この用紙はマニュアルの記述に対するコメント用紙ですので、製品の操作における技術的な質問は、ユーザー登録番号をご確認の上、弊社技術サポートまでお願い致します。

貴重なお時間をさいて頂き、ありがとうございました。

